Francisco Alexandre Lourenço Maia



Learning-based Policies for Stochastic Dynamic Inventory-Routing Problems

Submitted to Faculdade de Engenharia da Universidade do Porto in partial fulfillment of the requirements for the degree of Doctor of Philosophy in Industrial Engineering and Management, supervised by Gonçalo Figueira, Assistant Professor of Faculdade de Engenharia da Universidade do Porto, and Fábio Neves-Moreira, Assistant Professor of Faculdade de Economia da Universidade do Porto

Department of Industrial Engineering and Management
Faculdade de Engenharia da Universidade do Porto
2026

This research was partially supported by the PhD grant 2022.13466.BD awarded by the Portuguese Foundation for Science and Technology and by the European Union's Horizon 2020 research and innovation programme under grant agreement No 952060 (TRUST-AI).

"O homem primeiro tropeça, depois anda, depois corre, um dia voará."

José Saramago, *Memorial do Convento*

Acknowledgments

First of all, I want to thank my supervisors, Professor Gonçalo Figueira and Professor Fábio Neves-Moreira, for their continuous support and guidance, words of encouragement, and valuable insights over the past four years. The critical thinking and rigor you fostered in me will last throughout my career and life.

I want to express my gratitude to Professor Marlin Ulmer, Florentin Hildebrandt, and all the other colleagues who kindly welcomed me for three months at the Otto-von-Guericke University in Magdeburg, Germany. Your support and insights proved invaluable for the third and final stage of this PhD thesis. Additional thanks go to Mário Amorim Lopes for his support.

I am deeply grateful to all the professors, colleagues, and friends I have met during this PhD journey, in particular Nuno Marques, Sérgio Castro, Daniela Fernandes, and all the others that I'll never forget. Your friendship is one of the most important things I take away from this PhD experience.

I want to thank the Faculty of Engineering of the University of Porto and INESC TEC, namely the Department of Industrial Engineering and Management and the Center for Industrial Engineering and Management, respectively, for all the administrative support and excellent working environment provided - a heartfelt thank you to Professor Maria Antónia Carravilla for her dedication and kindness to the doctoral students.

Finally, I extend my thanks to my family and friends for all their support and motivation, especially my parents, Francisco Maia and Maria Isilda, and my sister Carolina Maia. Without you, this would not have been possible. Thank you for the principles you taught me, always fostering love, honesty and hard work.

Abstract

Supply chains compete to ensure the timely and cost-effective distribution of goods from suppliers to customers, with their success depending on how well operational activities are organized and interconnected. The coordination between inventory replenishment and transportation planning stands out, as it significantly impacts future inventory levels, stock-out risks, and overall logistics and distribution expenses. In this context, stochastic dynamic inventory-routing problems (SDIRPs) are complex combinatorial optimization problems that combine inventory management and vehicle routing in uncertain and dynamic environments. Inventory and routing decisions must be continuously adapted as new information is revealed, to react to and anticipate future events.

This thesis advances the research on SDIRPs, contributing to the fields of inventory-routing problems and sequential decision-making under uncertainty. In particular, it draws on insights from the inventory management and vehicle routing literatures and explores how machine learning can support the design of anticipatory and efficient policies. Three main contributions stand out, each resulting in a research article.

The first contribution is a comprehensive review of the SDIRP literature, classifying the existing studies in two main dimensions (problem settings and decision policies), identifying research gaps and emerging trends, and pointing to future directions. The review highlights opportunities to account for extended network configurations, multiple sources of uncertainty, perishability, multi-objective formulations, and pickup and delivery operations, to bridge the gap to more realistic models. It further discusses how different problem aspects, including scale, complexity, and real-time considerations, shape the design of decision policies, and highlights learning-based policies as a promising direction.

The second contribution addresses the classical SDIRP, where one supplier replenishes a set of customers facing stochastic demand over a multi-period horizon using a capacitated vehicle. The proposed policies follow a three-stage decision process: (i) a priority score is computed for each customer using a rule learned via genetic programming, (ii) delivery quantities are determined using an (s,S) inventory replenishment rule calibrated by a genetic algorithm; (iii) vehicle routes are optimized using a mathematical solver. Computational experiments show that the resulting policies outperform all benchmarks while being orders of magnitude faster than integrated inventory-routing approaches based on direct lookahead approximation.

Finally, the third contribution considers an SDIRP inspired by the operations of a Portuguese pharmaceutical distributor, where a central warehouse replenishes a set of pharmacies facing daily stochastic demand, using a fleet of vehicles with limited capacity and subject to a maximum route duration and a service time at each pharmacy. Within the framework of approximate dynamic programming, the proposed policy relies on a parametric cost function approximation that incorporates an inventory target level strategy to capture future demand, an inventory synchronization strategy to anticipate future routing limitations, and the learning of adaptive policy parameters. Computational experiments show that both strategies are effective individually, with their integration yielding superior performance across all practice-oriented benchmarks.

Resumo

As cadeias de abastecimento competem entre si para garantir a distribuição atempada e eficiente de produtos entre fornecedores e clientes, estando o seu desempenho dependente da organização e interligação das várias atividades operacionais. A coordenação entre o reabastecimento de stocks e as operações de transporte destaca-se, com impacto significativo na disponibilidade futura de stock e nos custos logísticos associados. Neste contexto, os problemas dinâmicos e estocásticos de roteamento de veículos e gestão de inventário (SDIRPs) constituem problemas de otimização combinatória complexos, nos quais as decisões de inventário e de roteamento devem ser continuamente adaptadas à medida que novas informações são reveladas, para reagir e antecipar eventos futuros.

Esta tese contribui para o avanço da investigação em SDIRPs, abrangendo as áreas de gestão de inventário, roteamento de veículos e problemas de decisão sequenciais sob incerteza. Em particular, explora os contributos de ambas as literaturas e como *machine learning* pode apoiar o desenvolvimento de políticas de decisão proativas e eficientes. Destacam-se três contribuições principais, que resultaram em três artigos científicos.

A primeira contribuição é uma revisão da literatura em SDIRPs, que classifica os estudos existentes em duas dimensões principais (configurações do problema e políticas de decisão), identifica lacunas de investigação, tendências emergentes e direções futuras. A revisão evidencia oportunidades para considerar redes de distribuição mais complexas, perecibilidade, formulações multi-objetivo e operações de *pickup and delivery*, contribuindo para modelos mais realistas. A revisão discute ainda como a escala, a complexidade e o tempo disponível para a tomada de decisão influenciam o desenho das políticas de decisão, destacando o recurso a *machine learning* como uma direção promissora.

A segunda contribuição aborda o clássico SDIRP, em que um fornecedor reabastece um conjunto de clientes que enfrentam procura estocástica ao longo de um horizonte multi-período. As políticas de decisão propostas seguem um processo de três etapas: (i) cálculo de uma prioridade de entrega através de uma regra aprendida via *genetic programming*; (ii) determinação das quantidades de entrega usando uma regra do tipo (s,S) calibrada com recurso a um algoritmo genético; (iii) otimização da rota do veículo usando um *solver* matemático. Os resultados computacionais mostram que as políticas propostas superam todos os benchmarks considerados, sendo várias ordens de grandeza mais rápidas do que as políticas integradas baseadas em *direct lookahead approximation*.

Por fim, a terceira contribuição considera um SDIRP inspirado nas operações de um distribuidor farmacêutico português, em que um armazém central reabastece diariamente um conjunto de farmácias com procura estocástica, utilizando uma frota de veículos com capacidade limitada sujeita a restrições operacionais. No contexto de *approximate dynamic programming*, a política proposta baseia-se numa *parametric cost function approximation* que incorpora uma estratégia de níveis de stock alvo que capta a procura futura, uma estratégia de sincronização de stocks que antecipa limitações futuras de roteamento e a aprendizagem de parâmetros adaptativos. Os resultados computacionais mostram que ambas as estratégias são eficazes individualmente, com a sua integração a resultar num desempenho superior a todos os *benchmarks* alinhados com práticas da indústria.

Contents

List of Figures xv

List of Tables xvii

List of Acronyms xix

1 Introduction 1
- 1.1 Research objectives and questions . . . 3
- 1.2 Thesis structure . . . 6
- Bibliography . . . 7

2 Literature review on stochastic dynamic inventory-routing problems 11
- 2.1 Introduction . . . 12
- 2.2 Research methodology . . . 14
  - 2.2.1 Material collection . . . 14
  - 2.2.2 Descriptive analysis . . . 15
  - 2.2.3 Category selection . . . 15
- 2.3 Problem formulation . . . 16
  - 2.3.1 Problem description . . . 16
  - 2.3.2 General Markov decision process model . . . 17
- 2.4 Problem settings . . . 21
  - 2.4.1 General features . . . 22
  - 2.4.2 Inventory features . . . 25
  - 2.4.3 Routing features . . . 27
- 2.5 Decision policies . . . 30
  - 2.5.1 Myopic policies . . . 31
  - 2.5.2 Policy search . . . 33
  - 2.5.3 Direct lookahead approximation . . . 36
  - 2.5.4 Value function approximation . . . 41
  - 2.5.5 Hybrid policies . . . 42
  - 2.5.6 Learning and simulation in policy design . . . 44
- 2.6 Theoretical and empirical instances . . . 47
- 2.7 Conclusions and future directions . . . 48
  - 2.7.1 Bridging the gap to realistic problems . . . 49
  - 2.7.2 Designing sequential decision policies . . . 51
- Bibliography . . . 54

3 Policy design for the stochastic dynamic inventory-routing problem 63
3.1 Introduction . . . 64
3.2 Related works . . . 65
3.2.1 Stochastic dynamic inventory-routing models . . . 65
3.2.2 Sequential decision policies . . . 66
3.2.3 Evolutionary algorithms . . . 67
3.3 Problem definition . . . 68
3.3.1 Problem description . . . 68
3.3.2 Illustrative example . . . 69
3.3.3 Markov decision process . . . 70
3.4 Solution method . . . 72
3.4.1 Three-stage decision policy . . . 73
3.4.2 Learning strategies . . . 75
3.4.3 Fitness function . . . 77
3.5 Experimental setting . . . 78
3.5.1 Instance setting . . . 78
3.5.2 Performance metrics . . . 79
3.5.3 Training configurations . . . 80
3.5.4 Benchmarks . . . 80
3.5.5 Implementation details . . . 82
3.6 Computational results . . . 82
3.6.1 Improving inventory replenishment . . . 82
3.6.2 Learning priority rules and calibrating the RPA parameters . . . 84
3.6.3 Perfect-hindsight comparison . . . 85
3.6.4 Scenario-level analysis . . . 85
3.7 Conclusion . . . 87
Bibliography . . . 88
Appendix 3.A Constructive routing heuristic . . . 92
Appendix 3.B Setup cost estimates comparison . . . 92
Appendix 3.C Solution quality . . . 93
Appendix 3.D Trade-off between performance and runtime . . . 94
Appendix 3.E Gurobi optimality gaps . . . 94

4 Policy design for a real-world pharmaceutical distribution problem 97
4.1 Introduction . . . 98
4.2 Related works . . . 101
4.2.1 Pharmaceutical distribution . . . 101
4.2.2 Stochastic dynamic inventory-routing . . . 103
4.3 Problem definition . . . 105
4.3.1 Problem description . . . 105
4.3.2 Illustrative example . . . 105
4.3.3 Sequential decision process . . . 107
4.4 Solution method . . . 110
4.4.1 Motivation and overview . . . 110

4.4.2 Cost function approximation . . . 110
4.4.3 Bayesian optimization algorithm . . . 111
4.5 Experimental setting . . . 112
4.5.1 Instance setting . . . 112
4.5.2 Benchmarks . . . 114
4.5.3 Implementation details . . . 115
4.6 Computational results . . . 116
4.6.1 Solution quality . . . 116
4.6.2 The value of local inventory synchronization . . . 117
4.6.3 Instance data analysis . . . 119
4.7 Conclusion . . . 120
Bibliography . . . 121
Appendix 4.A Bayesian optimization algorithm . . . 125
Appendix 4.B The value of (adaptive) scenario-dependent policy parameters . . . 126

5 Conclusions and future work **127**
5.1 Fulfillment of the research objectives and answers to the research questions 127
5.2 Future work . . . 133
Bibliography . . . 134

List of Figures

1.1 Research methodology. 4

2.1 Number of publications per year. 15
2.2 Decision-making process behind an SDIRP. 17
2.3 Conventional MDP for a periodic SDIRP. 21
2.4 Conventional MDP for a real-time SDIRP. 22

3.1 Example of a potential state, two alternative decisions (based on two different priority rules), corresponding post-decision states, demand realization and next states. 70
3.2 Training procedure of the proposed three-stage policy. 75
3.3 Total cost performance (MYOPIC as the baseline) vs. average runtime per scenario configuration. 94

4.1 Example of a state, two possible decisions, corresponding post-decision states, demand realization, and next states. 106
4.2 Training procedure of our CFA policy. 112
4.3 Distribution network including the central warehouse and 24 pharmacies in southern Portugal. 113
4.4 Average expected demand across the pharmacy network for each weekday (Monday-Sunday). 114
4.5 Average inventory target deviation by CFA_{QD} and CFA. 118
4.6 Average proximity-weighted inventory imbalance by CFA_{QD} and CFA. . . . 118
4.7 Average holding costs by CFA_{QD} and CFA. 118
4.8 Average stockout costs by CFA_{QD} and CFA. 118

List of Tables

2.1 Material collection procedure. . . . 14
2.2 Notation summary. . . . 17
2.3 Problem settings taxonomy. . . . 22
2.4 Problem settings classification (general features). . . . 26
2.5 Problem settings classification (inventory and routing features). . . . 30
2.6 Sequential decision policies taxonomy. . . . 31
2.7 Learning domains and subdomains taxonomy. . . . 44
2.8 Decision policies classification. . . . 47
2.9 Summary of instances across reviewed papers. . . . 49

3.1 Notation summary. . . . 69
3.2 Terminal and operator sets. . . . 76
3.3 Training, validation and test instance settings. . . . 79
3.4 Parameterization of the GP algorithm. . . . 80
3.5 Parameterization of the GA algorithm. . . . 81
3.6 Overall performance per policy (MYOPIC as the baseline). . . . 83
3.7 Trade-off between rule's complexity and total cost performance (LIF-RPA as the baseline). . . . 84
3.8 Total cost performance for $N = 5$ and $T = 10$ (MYOPIC as the baseline). . . . 85
3.9 Total cost performance (C_{mean}) across scenario groups (MYOPIC as the baseline). . . . 86
3.10 Setup cost configurations comparison (customer-supplier distance as the baseline). . . . 93
3.11 Overall performance per policy (MYOPIC as the baseline). . . . 93
3.12 Gurobi optimality gaps (on average) for MYOPIC and DLA. . . . 95

4.1 Overall comparison with the proposed CFA policy. . . . 117
4.2 Scenario-wise comparison with the proposed CFA policy. . . . 119
4.3 Learned policy parameters for each experimental scenario. . . . 126

List of Acronyms

ADP	Approximate dynamic programming
AI	Artificial intelligence
ALNS	Adaptive large neighborhood search
BO	Bayesian optimization
BOF	Big orders first
CFA	Cost function approximation
CO	Combinatorial optimization
DLA	Direct lookahead approximation
GA	Genetic algorithm
GP	Genetic programming
IRP	Inventory-routing problem
LIF	Lowest inventory first
LSF	Lowest storage first
MDP	Markov decision process
ML	Maximum-level
OU	Order-up-to-level
P&D	Pickup & delivery
PFA	Policy function approximation
PH	Perfect-hindsight
RH	Rolling horizon
RL	Reinforcement learning
RPA	Revised power approximation
SDIRP	Stochastic dynamic inventory-routing problem
SDP	Stochastic dynamic programming
SDVRP	Stochastic dynamic vehicle routing problem
SL	Supervised learning
TSP	Traveling salesman problem
VFA	Value function approximation
VMI	Vendor-managed inventory
VRP	Vehicle routing problem

Chapter 1

Introduction

Globalization, technological advances, and increasing uncertainty are among the factors that contribute to the continuous reshaping of supply chain operations. These are characterized by multiple stakeholders with overlapping objectives, complex information flows, fluctuations in demand and supply, and transportation challenges. From upstream (i.e., suppliers) to downstream (i.e., retailers and last-mile customers), these challenges underscore the need for greater visibility and coordination among all stakeholders to support more informed decision-making and improved operational performance. Supply chain digitalization has therefore been increasingly recognized as an important means of enhancing the availability, quality, and exchange of information (Zhao et al., 2023).

Over the past decades, vendor-managed inventory (VMI) has become a successful supply chain management strategy in which the vendor (or supplier) takes full responsibility for monitoring and replenishing the inventory of one or multiple customers (or retailers), deciding when and how to replenish them in line with mutually agreed contractual terms (Waller et al., 1999). From the customer's perspective, this strategy saves resources allocated to purchasing and inventory control activities, while helping manage the risks of overstocking and stockouts. From the vendor's perspective, access to updated, real-time inventory levels and accurate demand patterns at customer locations helps anticipate unexpected demand fluctuations that might lead to significant supply disruptions.

Beyond effective inventory replenishment decisions, the vendor must also ensure that deliveries are performed efficiently. In fact, deciding when and how much to deliver to a given customer directly depends on the available transportation resources, including fleet size, vehicle capacity, and delivery time windows. Treating these decisions in isolation can therefore lead to poorly coordinated plans. For instance, efficient delivery routes may fail to support effective inventory replenishment plans, and vice versa, leading to increased inventory and transportation costs (Campbell et al., 1998).

The need to better align replenishment decisions with fleet operations lies at the core of the inventory-routing problem (IRP), a complex combinatorial optimization (CO) challenge that combines inventory management and vehicle routing in order to minimize the expected holding, stockout and routing costs over a finite or infinite planning horizon (Archetti and Speranza, 2016). The supplier must determine when to visit each customer, how much to replenish, and how to combine customers into feasible vehicle routes. Building on the classical vehicle routing problem (VRP), the first works on inventory-routing date back to the 1980s, capturing the interest of academics and practitioners, with applications ranging from industrial gas distribution to grocery and medical supply logistics.

Uncertainty is a common obstacle in inventory replenishment and distribution operations, arising from demand fluctuations, supply availability issues, delivery delays, and

price variability. Among the different sources of uncertainty, customer demand has received particular attention in the inventory-routing literature, contributing to a better understanding of how inventory dynamics affect both inventory and routing decisions over time. That said, demand uncertainty may be represented in several ways, ranging from stochastic modeling approaches (e.g., Rave et al., 2025), based on known probability distributions, to alternative fuzzy (e.g., Avila-Torres and Arratia-Martinez, 2024) or interval-based representations (e.g., Agra et al., 2018). In some scenarios, no future information may be provided at all (Bianchi et al., 2009).

In several real-world settings, demand information is not only uncertain but also dynamic, i.e., gradually revealed over time. Therefore, there is a growing need for inventory-routing decisions to be dynamic rather than static, and proactive rather than purely reactive. A static solution spanning the entire planning horizon is no longer appropriate. Instead, we need a policy that guides the decisions over time, based on observed demand. The joint inventory-routing optimization in a stochastic dynamic environment refers, therefore, to the stochastic dynamic inventory-routing problem (SDIRP), a complex sequential decision problem with both practical and theoretical relevance that is gaining increasing attention, as new publications emerge (e.g., Lagos, 2026).

> SDIRPs are inventory-routing problems in which (i) uncertain (stochastic) information, e.g., customer demand, is gradually revealed over time; (ii) planning decisions can be repeatedly adapted over the time horizon to react to and anticipate future developments (dynamicity).

Significant advances have been made in understanding, modeling, and solving the SDIRP. Initial research focused on static, deterministic formulations, often via mathematical programming models repeatedly solved over time without anticipating future demand (e.g., Trudeau and Dror, 1992) or, instead, approximating it with single-point forecasts (e.g., Elbek and Wøhlk, 2016). Stochastic programming formulations have emerged to better capture uncertainty through a finite set of scenarios, each representing a sequence of potential future demand realizations (e.g., Mirzapour Al-e hashem et al., 2019). Nevertheless, several challenges emerge, including significant dependence on the quality of lookahead approximations, whether deterministic or stochastic, the curse of dimensionality, and the difficulty of capturing the sequential evolution of decisions over time.

As a sequential decision problem, the SDIRP can be represented as a Markov decision process (MDP), a structured mathematical modeling framework for sequential decision-making under uncertain and complex environments (Puterman, 1994). Accordingly, it can be defined by a sequence of states, decisions, and information realizations. A state contains all the information needed to make a decision (e.g., inventory positions, unit holding and stockout costs), while a decision specifies which customers to visit, how much to replenish them, and the corresponding vehicle routes. Information realizations reflect the stochastic and dynamic behavior of demand at each customer over the planning horizon, often inferred from updated inventory levels that become available, for example, at regular, periodic intervals (e.g., daily, Brinkmann et al., 2019).

The main objective of solving an MDP is to find a policy, i.e., a parametric or non-parametric function, strategy, or rule, that maps a state to a decision, while respecting the problem constraints and minimizing the expected costs (Bellman, 1957). Nevertheless, obtaining an optimal policy for the SDIRP is a challenging task due to its combinatorial, uncertain, and dynamic nature. According to the unified framework for sequential decisions under uncertainty established by Powell (2022), two fundamental strategies stand out: policy search and lookahead approximations, which differ in how they approximate the future consequences of current decisions and in how they use the stochastic information to guide those decisions.

Rolling horizon (RH) approaches have been thoroughly investigated for decades in stochastic dynamic environments (Sethi and Sorger, 1991), ranging from purely reactive to more proactive strategies. For instance, simple inventory (e.g., order-up-to-level) and routing (e.g., nearest neighbor) heuristics may be useful in practice, but fail to capture the inherent inventory-routing interdependencies and anticipate future uncertainty, leading to less effective decisions. On the other hand, direct lookahead approximation (DLA) policies address these limitations by jointly optimizing inventory and routing decisions over a finite lookahead horizon. However, important challenges arise when designing a lookahead model, including representing uncertainty and selecting an appropriate lookahead horizon, which affects both the computational effort required to solve the model and whether a decision is more or less myopic.

These challenges, particularly the trade-off between solution quality and computational effort, motivate the exploration of alternative policy design strategies. For instance, policy search methods offer a promising basis for developing anticipatory and efficient policies that yield effective decisions within short inference times, without the need to solve large and computationally demanding optimization models. In this context, the intersection of the fields of approximate dynamic programming, machine learning and CO, combined with insights from the inventory management and vehicle routing literatures, offers a promising avenue for advancing the literature on SDIRPs in the light of prescriptive analytics.

1.1. Research objectives and questions

The main goal of this PhD project is to advance the field and understanding of the SDIRP and explore the potential of learning algorithms and policy search methods to support the design of anticipatory and efficient policies. To achieve this goal, this thesis is organized around three main research objectives (RO), as shown in Figure 1.1. First, a comprehensive review of SDIRPs is conducted to identify research gaps and emerging trends, highlighting which problem variants have been studied, whether they align with the complexities of real-world settings, and how inventory and routing decisions have been addressed. Building on these insights, the second objective focuses on the design of anticipatory and efficient inventory-routing policies for the classical SDIRP, using a set of instances derived from well-established literature benchmarks (Coelho et al., 2014a). Finally, the third objective addresses a real-world problem motivated by the operations of a Portuguese pharmaceutical distributor, focusing on developing practice-oriented inventory-routing policies.

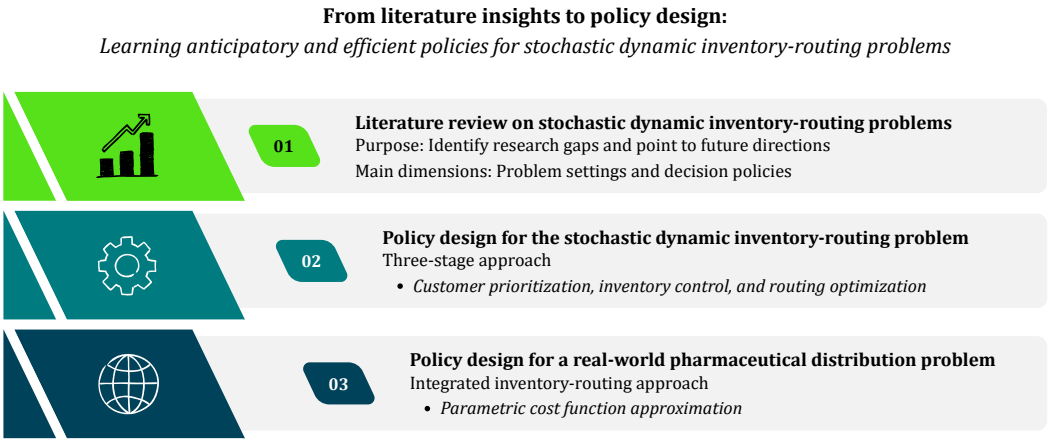


Figure 1.1: Research methodology.

RO 1: Literature review on stochastic dynamic inventory-routing problems

The SDIRP is a complex CO problem that has attracted significant interest from both the academic and industrial communities, contributing to the development of the fields of inventory-routing, sequential decision-making, and stochastic dynamic optimization. In this context, it becomes paramount to understand how the literature has been addressing this problem over the last decades, i.e., which problem settings have been explored, and how inventory and routing decisions have been addressed. This objective responds to the need for an updated literature review focused on the SDIRP, as the last comprehensive review on dynamic inventory-routing dates back to Baita et al. (1998). Since then, the field has advanced significantly, with the emergence of new problem variants and solution approaches (notably those incorporating machine learning), translating into a substantial increase in publications over the last decade.

Despite the numerous literature reviews dedicated to the IRP, most tend to focus on specific application contexts, such as sustainability (Soysal et al., 2019), perishability (Shaabani, 2022), industrial applications (Andersson et al., 2010), or planning horizons (Moin and Salhi, 2007). From a broader perspective, Coelho et al. (2014b) reviewed the past thirty years of research on IRPs and proposed a classification based on their structural variants and the availability of demand information. However, the discussion on SDIRPs remains limited. Furthermore, Archetti and Bertazzi (2021) addressed the recent developments in routing and inventory-routing, including modeling uncertainty, while Roldán et al. (2017) focused on stochastic lead times and demand. Despite all these contributions, none provide a comprehensive discussion centered on dynamic decision-making, which completely changes the difficulty of the problem and therefore requires special attention.

This objective focuses on investigating the SDIRP literature across two interconnected dimensions: problem settings and decision policies, identifying critical gaps and emerging challenges. Furthermore, the proposed literature review aims to assess how various problem aspects, including scale, complexity, and real-time considerations, shape the design of decision policies, thereby mapping promising research directions. The following research questions (RQ) are to be answered:

- RQ 1.1: Which problem settings are most explored in the SDIRP literature, and which remain underexplored despite their practical relevance?
- RQ 1.2: Which policy classes have been explored in the SDIRP literature, and how do different problem aspects, including scale, complexity and real-time considerations, shape the design and suitability of decision policies?
- RQ 1.3: How can learning mechanisms support decision policies for the SDIRP?

RO 2: Policy design for the stochastic dynamic inventory-routing problem

Dynamic IRPs require coordinated and updated inventory replenishment and vehicle routing decisions over time. The prospect of looking into the future and anticipating upcoming uncertainties, along with the joint inventory-routing optimization, makes DLA the most widely used policy class in the literature, where deterministic and stochastic lookahead models are repeatedly solved in an RH fashion. Still, the computational effort required to solve these lookahead models increases with the length of the lookahead horizon, becoming even more challenging as the problem's complexity and scale grow, given the larger state and decision spaces. On the other hand, myopic RH approaches are limited in their ability to account for future uncertainty, leading to less effective inventory-routing decisions.

This objective focuses on solving the classical SDIRP, where one supplier replenishes a set of customers facing stochastic demand with a single product over a multi-period horizon, using a capacitated vehicle. It aims to investigate how insights from the inventory management and vehicle routing literatures can be combined with policy search, machine learning and CO to support the design of anticipatory and efficient policies. These should account for relevant inventory-routing interdependencies and offer a competitive alternative to more integrated approaches.

To evaluate the robustness and effectiveness of the proposed policies, a set of problem instances based on well-established literature benchmarks is to be considered, with the goal of minimizing the total inventory holding, stockout, and routing costs (Coelho et al., 2014a). In particular, it is interesting to explore the trade-off between solution quality and runtime, and how it changes under different problem conditions, including endpoint structure, planning horizon, vehicle capacity, holding and stockout costs. The following research questions are to be answered:

- RQ 2.1: How can insights from both inventory management and vehicle routing be combined to support the design of anticipatory and efficient policies?
- RQ 2.2: How do the proposed policies perform in terms of solution quality and computational effort?
- RQ 2.3: Under which problem conditions are the proposed policies most advantageous?

RO 3: Policy design for a real-world pharmaceutical distribution problem

This objective addresses an SDIRP inspired by the operations of a Portuguese pharmaceutical distributor. A central warehouse replenishes a set of pharmacies periodically (i.e., daily) with a particular medicine, using a fleet of vehicles with limited capacity. Pharmacies face stochastic demand revealed at the end of each day through inventory updates. Inventory and routing decisions should be planned at the beginning of each day. Operational constraints include a service time that accounts for handling and unloading activities at pharmacy locations, and a maximum route duration per vehicle to reflect the driver's availability. The main trade-off lies between inventory holding and stockout costs, while routing acts as a resource constraint.

Finding effective decisions can be quite challenging, as the problem combines operational constraints, inventory management across multiple pharmacies, and dynamic multi-period routing. Accordingly, this objective aims to investigate how insights from the inventory management and vehicle routing literatures, together with the potential of cost function approximation (CFA) as a policy class, can support the design of an anticipatory and efficient inventory-routing policy. The proposed policy is to be compared with practice-oriented approaches and evaluated under different conditions that may be encountered in practice. The following research questions are to be answered:

- RQ 3.1: What operational challenges arise when addressing a real-world pharmaceutical distribution problem?
- RQ 3.2: How to leverage the potential of CFA to design an anticipatory and efficient policy?
- RQ 3.3: How does the proposed policy perform compared with standard industry practices in terms of solution quality and robustness?

1.2. Thesis structure

This thesis is organized as a collection of papers, with each chapter corresponding to a paper that aims to answer the research questions of each of the three objectives.

Chapter 2 provides a detailed understanding and organizes the literature on SDIRPs. It presents an in-depth literature review that focuses on two main domains, i.e., problem dimensions and decision policies, pointing out the most promising forthcoming research paths, including those explored in the following chapters.

Chapter 3 contributes to the SDIRP literature from a prescriptive perspective, proposing anticipatory and efficient inventory-routing policies to solve the classical SDIRP. These policies draw on insights from both inventory management and vehicle routing, along with the concepts of policy search, machine learning, and CO. Their performance is evaluated against reactive and DLA approaches, focusing on inventory holding, stockout, and routing costs, as well as the trade-off between solution quality and computational effort.

Chapter 4 addresses an SDIRP motivated by the operations of a real-world pharmaceutical distributor, involving the daily replenishment of a set of pharmacies in southern

Portugal facing stochastic demand patterns. The chapter focuses on designing anticipatory and efficient inventory-routing policies for this real-world setting, combining concepts of the inventory management literature with the potential of CFA as a policy class. Their performance is evaluated against standard industry practices, with particular focus on inventory holding and stockout costs.

Finally, Chapter 5 summarizes the main findings and contributions of this doctoral research, ultimately highlighting promising avenues for future research.

Bibliography

Agra, A., Christiansen, M., Hvattum, L.M., Rodrigues, F., 2018. Robust optimization for a maritime inventory routing problem. Transportation Science 52, 509–525. https://doi.org/10.1287/trsc.2017.0814.

Andersson, H., Hoff, A., Christiansen, M., Hasle, G., Løkketangen, A., 2010. Industrial aspects and literature survey: Combined inventory management and routing. Computers & Operations Research 37, 1515–1536. https://doi.org/10.1016/j.cor.2009.11.009.

Archetti, C., Bertazzi, L., 2021. Recent challenges in routing and inventory routing: E-commerce and last-mile delivery. Networks 77, 255–268. https://doi.org/10.1002/net.21995.

Archetti, C., Speranza, M.G., 2016. The inventory routing problem: the value of integration. International Transactions in Operational Research 23, 393–407. https://doi.org/10.1111/itor.12226.

Avila-Torres, P.A., Arratia-Martinez, N.M., 2024. Fuzzy inventory-routing problem with priority customers: Pa avila-torres, nm arratia-martinez. Soft Computing 28, 7947–7961. https://doi.org/10.1007/s00500-024-09721-z.

Baita, F., Ukovich, W., Pesenti, R., Favaretto, D., 1998. Dynamic routing-and-inventory problems: a review. Transportation Research Part A: Policy and Practice 32, 585–598. https://doi.org/10.1016/S0965-8564(98)00014-7.

Bellman, R., 1957. Dynamic Programming. Princeton University Press, Princeton, NJ, USA.

Bianchi, L., Dorigo, M., Gambardella, L.M., Gutjahr, W.J., 2009. A survey on metaheuristics for stochastic combinatorial optimization. Natural Computing 8, 239–287. https://doi.org/10.1007/s11047-008-9098-4.

Brinkmann, J., Ulmer, M.W., Mattfeld, D.C., 2019. Dynamic lookahead policies for stochastic-dynamic inventory routing in bike sharing systems. Computers & Operations Research 106, 260–279. https://doi.org/10.1016/j.cor.2018.06.004.

Campbell, A., Clarke, L., Kleywegt, A., Savelsbergh, M., 1998. The inventory routing problem, in: Fleet Management and Logistics. Springer, pp. 95–113. https://doi.org/10.1007/978-1-4615-5755-5_4.

Coelho, L.C., Cordeau, J.F., Laporte, G., 2014a. Heuristics for dynamic and stochastic inventory-routing. Computers & Operations Research 52, 55–67. https://doi.org/10.1016/j.cor.2014.07.001.

Coelho, L.C., Cordeau, J.F., Laporte, G., 2014b. Thirty years of inventory routing. Transportation Science 48, 1–19. https://doi.org/10.1287/trsc.2013.0472.

Elbek, M., Wøhlk, S., 2016. A variable neighborhood search for the multi-period collection of recyclable materials. European Journal of Operational Research 249, 540–550. https://doi.org/10.1016/j.ejor.2015.08.035.

Mirzapour Al-e hashem, S.M., Rekik, Y., Hoseinhajlou, E.M., 2019. A hybrid l-shaped method to solve a bi-objective stochastic transshipment-enabled inventory routing problem. International Journal of Production Economics 209, 381–398. https://doi.org/10.1016/j.ijpe.2017.06.020.

Lagos, F., 2026. The stochastic and dynamic inventory routing problem with fine time granularity: A reinforcement learning approach. Computers & Operations Research 185, 107298. https://doi.org/10.1016/j.cor.2025.107298.

Moin, N.H., Salhi, S., 2007. Inventory routing problems: a logistical overview. Journal of the Operational Research Society 58, 1185–1194. https://doi.org/10.1057/palgrave.jors.2602264.

Powell, W.B., 2022. Reinforcement Learning and Stochastic Optimization: A Unified Framework for Sequential Decisions. John Wiley & Sons, Hoboken, NJ. https://doi.org/10.1002/9781119815068.

Puterman, M.L., 1994. Markov decision processes: discrete stochastic dynamic programming. John Wiley & Sons.

Rave, A., Fontaine, P., Kuhn, H., 2025. Cyclic stochastic two-echelon inventory routing for an application in medical supply. European Journal of Operational Research 325, 81–99. https://doi.org/10.1016/j.ejor.2025.02.032.

Roldán, R.F., Basagoiti, R., Coelho, L.C., 2017. A survey on the inventory-routing problem with stochastic lead times and demands. Journal of Applied Logic 24, 15–24. https://doi.org/10.1016/j.jal.2016.11.010.

Sethi, S., Sorger, G., 1991. A theory of rolling horizon decision making. Annals of Operations Research 29, 387–415. https://doi.org/10.1007/BF02283607.

Shaabani, H., 2022. A literature review of the perishable inventory routing problem. The Asian Journal of Shipping and Logistics 38, 143–161. https://doi.org/10.1016/j.ajsl.2022.05.002.

Soysal, M., Çimen, M., Belbağ, S., Toğrul, E., 2019. A review on sustainable inventory routing. Computers & Industrial Engineering 132, 395–411. https://doi.org/10.1016/j.cie.2019.04.026.

Trudeau, P., Dror, M., 1992. Stochastic inventory routing: Route design with stockouts and route failures. Transportation Science 26, 171–184. https://doi.org/10.1287/trsc.26.3.171.

Waller, M., Johnson, M.E., Davis, T., 1999. Vendor-managed inventory in the retail supply chain. Journal of Business Logistics 20, 183.

Zhao, N., Hong, J., Lau, K.H., 2023. Impact of supply chain digitalization on supply chain resilience and performance: A multi-mediation model. International Journal of Production Economics 259, 108817. https://doi.org/10.1016/j.ijpe.2023.108817.

CHAPTER 2

Literature review on stochastic dynamic inventory-routing problems

Stochastic dynamic inventory-routing: A comprehensive review

Francisco Maia[*] · **Gonçalo Figueira**[*] · **Fábio Neves-Moreira**[†]

Published in *Computers & Operations Research*, 2026
https://doi.org/10.1016/j.cor.2026.107383

Highlights

- Provides a comprehensive review of the stochastic dynamic inventory-routing problem;
- Offers an in-depth classification of decision policies;
- Identifies underexplored problem settings to align with real-world applications;
- Investigates how different problem aspects shape the design of decision policies;
- Points out current trends, challenges and future research directions.

Abstract The stochastic dynamic inventory-routing problem (SDIRP) is a fundamental problem within supply chain operations that integrates inventory management and vehicle routing while handling the stochastic and dynamic nature of exogenous factors unveiled over time, such as customer demands, inventory supply and travel times. While practical applications require dynamic and stochastic decision-making, research in this field has only recently experienced significant growth, with most inventory-routing literature focusing on static variants. This paper reviews the current state of research on SDIRPs, identifying critical gaps and highlighting emerging trends in problem settings and decision policies. We extend the existing inventory-routing taxonomies by incorporating additional problem characteristics to better align models with real-world contexts. As a result, we highlight the need to account for further sources of uncertainty, multiple-supplier networks, perishability, multiple objectives, and pickup and delivery operations. We further categorize each study based on its policy design, investigating how different problem aspects shape decision policies. To conclude, we emphasize that large-scale and real-time problems require more attention and can benefit from decomposition approaches and learning-based methods.

[*]INESC TEC, Faculdade de Engenharia, Universidade do Porto, Rua Dr. Roberto Frias, Porto, 4200-465, Portugal
[†]INESC TEC, Faculdade de Economia, Universidade do Porto, Rua Dr. Roberto Frias, Porto, 4200-464, Portugal

Keywords Literature review · Stochastic dynamic inventory-routing · Sequential decision-making · Machine learning

2.1. Introduction

In recent decades, vendor-managed inventory (VMI) has emerged as a cooperating strategy for managing and reducing inventory costs, where information models, containing on-hand inventories, customer demands and lead times, are paramount to making informed and proper decisions. The vendor is responsible for replenishing the inventory held by one or multiple customers, bringing numerous benefits such as reducing inventory carrying costs and increasing service levels (Van den Bogaert and Van Jaarsveld, 2022). Therefore, decisions must consider when to visit customers, how much to replenish them, and in what routes (Campbell et al., 1998).

Building on VMI principles, the need for better inventory and transportation logistics coordination has led to the inventory-routing problem (IRP), an extension of the traditional vehicle routing problem (VRP). The IRP seeks to minimize the total inventory and transportation costs over a given planning horizon while ensuring a desired service level. Its relevance has driven significant research in multiple sectors, including industrial gas distribution (e.g., Adelman, 2004), healthcare (e.g., Niakan and Rahimi, 2015), retail (e.g., Aghezzaf, 2008), military (e.g., McKenna et al., 2020), bike-sharing (e.g., Brinkmann et al., 2020) and waste collection (e.g., Markov et al., 2020) logistics.

Since the first publication in 1983 by Bell et al. (1983), numerous works have addressed the deterministic variant (e.g., Bertazzi et al., 2002). However, real-world operations are inherently uncertain and dynamic, with the primary sources of variability stemming from customer demands (e.g., Solyalı et al., 2012) and travel times (e.g., Rodrigues et al., 2019), the latter being most predominant in maritime transportation. This variability is typically represented by random variables with known probability distributions (e.g., Federgruen and Zipkin, 1984), although it can also be characterized through fuzzy values (e.g., Niakan and Rahimi, 2015) or interval values in robust optimization approaches (e.g., Agra et al., 2018). In contrast, when no future information at all is provided, decisions are made on the fly based on real-time service requests (e.g., Bertazzi et al., 2025).

Besides uncertainty, the dynamic nature of the problem further increases the complexity of decision-making, as unknown factors, such as customer demands, are gradually disclosed over time and not fully known in advance (Coelho et al., 2014a). For instance, inventory levels may be revealed at regular periods (e.g., daily, Kleywegt et al., 2004) or episodically (e.g., Giesen et al., 2010), such as when a vehicle arrives at a customer location and a pickup or delivery should occur. As a result, continuous replanning is essential to leverage the newly disclosed information, whether driven by periodic reviews or real-time updates. The joint optimization of inventory and routing decisions under stochastic dynamic environments refers, therefore, to the stochastic dynamic inventory-routing problem (SDIRP).

The SDIRP is a complex combinatorial optimization problem within sequential decision-making. As a dynamic problem, a practical approach is no longer focused on computing a single and static optimal solution but rather a decision policy that guides the evolution

of decisions based on the observed information realizations (e.g., customer demands). The SDIRP can then be formulated as a Markov decision process (MDP) by defining states, actions, information realizations and (negative) rewards reflecting inventory and routing costs. Accordingly, we can compute policies that minimize the expected costs over the planning horizon. Given the inherent curse of dimensionality, which makes exact solutions computationally infeasible, these policies can be effectively obtained through approximate dynamic programming or reinforcement learning (see Powell, 2022b).

Despite an expanding body of research on the SDIRP, no updated literature review offers a comprehensive, in-depth analysis of the current state of the art, focusing on sequential decision-making. Existing IRP reviews tend to focus on specific application contexts, such as sustainability (Soysal et al., 2019), perishability (Shaabani, 2022), industrial applications (Andersson et al., 2010), or planning horizons (Moin and Salhi, 2007). From a broader perspective, Coelho et al. (2014b) surveyed thirty years of research and proposed a classification based on problem structure and information availability. However, their discussion on SDIRPs remains limited. Subsequent studies, such as Markov et al. (2018) and Archetti and Bertazzi (2021), addressed recent developments in routing and inventory-routing, including modeling uncertainty, while Roldán et al. (2017) focused on stochastic lead times and demands. Nevertheless, none provide a structured classification centered on dynamic decision-making.

While Baita et al. (1998) provided an early review of dynamic routing-and-inventory problems, their work predates significant advancements in the field. Our review is the first since Baita et al. (1998) to focus exclusively on the stochastic dynamic variant of the IRP. We explore the literature along two interconnected dimensions: problem settings and decision policies. Besides identifying promising gaps in problem settings, we investigate how different problem characteristics, such as scale and complexity, influence the design of decision policies. We highlight opportunities for enriching current policies by drawing on insights from other stochastic and dynamic optimization communities. The contributions of this paper are fourfold:

1. It proposes an in-depth classification of the decision policies adopted in each paper, according to the classes of policies discussed by Powell (2022a), decomposition approach, optimization method and potential learning domains;
2. It extends the problem taxonomies proposed by the existing IRP reviews with new dimensions: general (e.g., objective function), inventory (e.g., perishable items) and routing features (e.g., time windows, logistical context, split delivery);
3. It investigates how various problem aspects, including scale, complexity and real-time considerations, shape the design of decision policies;
4. It identifies current trends, emerging challenges, and promising research directions regarding problem settings and decision policies for the SDIRP.

The remainder of this paper is organized as follows. Section 2.2 depicts the research methodology. Thereafter, we formulate the SDIRP as an MDP in Section 2.3, distinguishing periodic from real-time or episodic decisions. Section 2.4 investigates each problem

setting, including general (2.4.1), inventory (2.4.2) and routing (2.4.3) features. Decision policies are explored in Section 2.5, including the study of each policy class (2.5.1-2.5.5), and a proposed taxonomy to classify the policy's learning nature, if applicable (2.5.6). Section 2.6 exploits the existing problem instances, both real-world and theoretical. Conclusions and future directions are pointed out in Section 2.7.

2.2. Research methodology

Our systematic literature review follows the methodology proposed by Mayring et al. (2004), which contains four steps: (i) material collection (Section 2.2.1), (ii) descriptive analysis (Section 2.2.2), (iii) category selection (Section 2.2.3), and (iv) material evaluation: problem settings (Section 2.4) and decision policies (Section 2.5).

2.2.1 Material collection

Table 2.1 depicts the collection procedure followed in this paper. In order to access a wide range of academic literature, we utilized the following research databases: Scopus, Web of Science, and Google Scholar. The original query includes *"Inventory" AND "Routing"* in the paper's title, as well as *"Stochastic" OR "Uncertain" OR "Dynamic" OR "Online" OR "Rolling Horizon"* in the paper's keywords. We obtained 144 publications (137 journal articles + 7 preprints) from the Scopus database and 82 from the Web of Science database (December 16th, 2024). Conference proceedings, book series, master and PhD dissertations, unavailable publications, and papers written in a different language than English were excluded from this stage.

Table 2.1: Material collection procedure.

Stage	Description	Resulting papers
Stage 1	Initial query search	141 methodological (7 preprints) + 3 reviews
Stage 2	Exclusion criteria	36 methodological (2 preprints)
Stage 3	Snowballing procedure	46 methodological (3 preprints)

After assessing the abstract and other relevant sections of each paper, we applied the following exclusion criteria: (1) lack of stochasticity; (2) lack of dynamicity; (3) location IRPs; (4) pricing problems; (5) arc routing problems; (6) maritime logistics. Criteria (3) to (5) address operational challenges that do not align with the focus of the SDIRP. Criterion (6) arises from the unique constraints and complexities of the maritime IRP, where shipping operations involve continuous movements, i.e., vehicles require multiple periods to collect and deliver goods (Blom et al., 2020). After this stage, 36 methodological papers, including two preprints from 2024, have resulted. Afterwards, we applied a snowballing procedure (Pillac et al., 2013) to complement the previous list of papers using Google Scholar. This procedure consists of reviewing the references and citations of the articles (i.e., Coelho et al., 2014a) on our list to identify additional works and reduce the likelihood of missing works of significant interest. This stage led to ten additional methodological pa-

pers, contributing to a final sample of 46 methodological papers (including three preprints). Furthermore, the works in which we could not find enough information in the respective model or solution method are not included.

2.2.2 Descriptive analysis

Descriptive analysis aims to provide an overview of the chronological evolution of publications within SDIRPs. Figure 2.1 shows that the number of publications has been increasing considerably over time. We could divide them into three main periods. In the first 15 years (from 1987 to 2002), the works were sporadic, with no more than one publication per year and several consecutive years with no publication. Research activity slightly increased in the following ten years (2003 to 2013), with some years having two publications. Finally, from 2014 onward, the number of works significantly increased, with peaks of five and six publications per year in recent years. This trend indicates a growing interest in the field, motivated by new emerging challenges and advancements in solution approaches.

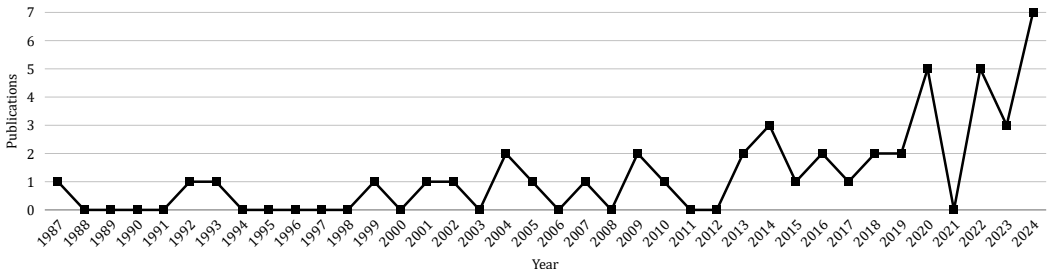


Figure 2.1: Number of publications per year.

2.2.3 Category selection

Our review follows two broad taxonomies: problem settings and decision policies. Tables 2.3 and 2.6 illustrate both taxonomies in a high-level outlook. Problem settings branch into three dimensions: general, inventory and routing. Regarding general features, our classification extends the structural variants proposed by Coelho et al. (2014b) by categorizing the objective function and uncertainty sources. Concerning inventory decisions, we classify stockouts into lost sales or backorders, the number of items, the existence of perishable items and the inventory policy itself. Regarding vehicle routing, we label the fleet composition and size, the consideration of time windows and split deliveries. Besides, we distinguish pickup and delivery operations from emergency transshipments, considered instantaneous and typically scheduled after demands have been revealed to face the occurrence of stockouts (Coelho et al., 2014a). Our classification framework also recognizes the problem domain or application sector.

Given the IRP's stochastic and dynamic nature, it is not practical to compute optimal policies for large instances in real-time. Authors, therefore, approximate future behaviors. Resorting to Powell (2022a) and Soeffker et al. (2022), we consider five main classes of policies: myopic, policy function approximation, cost function approximation, direct

lookahead approximation and value function approximation. Furthermore, we explore the learning procedure behind each policy, if applicable, and the periodicity of decisions.

2.3. Problem formulation

We start by providing the problem description and notation (Section 2.3.1). Then, we model the SDIRP as a Markov decision process (MDP), which provides a structured framework for optimizing decisions over time while accounting for uncertainty (Section 2.3.2). This helps to establish a baseline to discuss the problem settings in Section 2.4 and decision policies in Section 2.5.

2.3.1 Problem description

The SDIRP is a complex combinatorial optimization problem, enticing the attention of several research communities motivated by a vast spectrum of applications and decision-making complexity. It is an NP-hard problem since it includes the VRP as a special instance (Coelho et al., 2012). Similarly to the static IRP, the problem is defined with a graph $\mathcal{G} = (\mathcal{N}, \mathcal{E})$, in which the supplier denoted by node 0 is responsible for delivering products $p \in \mathcal{P}$ to a set of N geographically dispersed customers: $\mathcal{N}' = \mathcal{N} \setminus \{0\} = \{1, ..., N\}$. The connection between nodes is defined by a set of edges $\mathcal{E} = \{(i, j) : i, j \in \mathcal{N}, i \neq j\}$. For modeling purposes, long-term planning horizons are typically split into T consecutive discrete time periods: $t \in \mathcal{T} = \{0, 1, ..., T-1\}$.

We consider that the demand of product $p \in \mathcal{P}$ that each customer $i \in \mathcal{N}'$ faces per period $t \in \mathcal{T}$ is a random variable D_{ipt}. These demands are gradually revealed at the end of each period t and are only known probabilistically. The initial inventory levels of each customer i (I_{ip0}) and supplier (I_{0p0}) are provided at the beginning of the planning horizon ($t = 0$). Each customer and supplier incurs a unit holding cost h_{ip} at each period, and each customer i has a maximum storage capacity $\bar{U}_i$. The supplier has unlimited capacity and is periodically replenished with product p at the beginning of every period, according to a given rate r_{pt}.

A set of vehicles $k \in \mathcal{K} = \{1, ..., K\}$ with capacity Q_k is responsible for completing the planned deliveries, incurring a cost c_{ij} for traversing edge (i, j). Every vehicle must start from and return to the supplier empty. The continuous variable q_{ipkt} represents the delivery quantity of product p to customer i by vehicle k during period t.

As a result of stochastic demands, inventory shortages may arise at the end of each period after demands are disclosed. Let l_{ipt} be the unmet demand of product p at customer i in period t, and z_{ip} be the unit shortage cost of product p at customer i. We use the binary decision variables x_{ijkt} and y_{ikt} for routing decisions, where x_{ijkt} indicates whether the edge (i, j) is traversed and y_{ikt} indicates if customer i is visited by vehicle k in period t.

Finally, the continuous variables I_{0pt+1} and I_{ipt+1} define the inventory of product p at the supplier and each customer i at the end of period t. The objective is to minimize the total inventory holding, stockout and routing costs for the given planning horizon. Table 2.2 summarizes the model's notations. Figure 2.2 illustrates the decision-making process, demonstrating how demand is revealed and how inventories are updated over time.

Table 2.2: Notation summary.

	Sets
$\mathcal{N}$	Set of customers and supplier
$\mathcal{E}$	Set of edges (i, j)
$\mathcal{T}$	Planning horizon, indexed by t
$\mathcal{P}$	Set of products, indexed by p
$\mathcal{K}$	Set of vehicles, indexed by k
	Parameters
c_{ij}	Transportation cost for traversing edge (i, j), i.e., distance-dependent
Q_k	Capacity of vehicle k
I_{ip0}	Inventory level of product p at location i at the beginning of the planning horizon
$\bar{U}_i$	Inventory upper limit (capacity) at customer i
r_{pt}	Production rate of product p at the supplier in period t
h_{ip}	Unit holding cost of product p at location i
z_{ip}	Unit shortage cost of product p at customer i
D_{ipt}	Demand of product p in period t at customer i
	Decision variables
x_{ijkt}	Equal to 1, if edge $(i, j) \in \mathcal{E}$ is traversed by vehicle k in period t
y_{ikt}	Equal to 1, if location i is visited by vehicle k in period t
I_{ipt+1}	Inventory level of product p at location i at the end of period t
q_{ipkt}	Delivery quantity of product p for customer i by vehicle k in period t
l_{ipt}	Lost sale of product p at customer i in period t after the demand is revealed

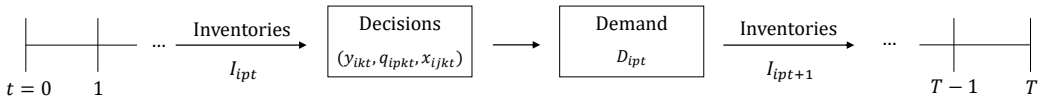


Figure 2.2: Decision-making process behind an SDIRP.

2.3.2 General Markov decision process model

The modeling of the SDIRP has evolved from static representations to more dynamic frameworks. Early research focused on static deterministic formulations, in which mathematical programming models were repeatedly solved over time, without explicitly accounting for future uncertainties or relying solely on single-point forecasts. Subsequently, scenario-based formulations were introduced to better capture uncertainty, often resorting to Monte Carlo simulation and other scenario-generation techniques. While these formulations improved upon earlier rolling horizon (RH) approaches by incorporating stochastic lookaheads, they fail to fully capture the sequential evolution of decisions over time. In this context, representing the SDIRP as an MDP offers a structured framework for dynamic decision-making under uncertainty, leveraging the fields of approximate dynamic programming and machine learning to develop adaptive policies expected to deliver superior performance.

Accordingly, we formulate the SDIRP as an MDP, i.e., a discrete-time stochastic control process and a valuable tool for modeling sequential decision problems in complex and uncertain contexts (Puterman, 2014). A sequential decision process is defined by a sequence of states $\{S_t\}_{t=0}^{T}$, decisions $\{a_t\}_{t=0}^{T-1}$ and information realizations $\{\omega_{t+1}\}_{t=0}^{T-1}$, where each decision follows a policy π so that $a_t = A^{\pi}(S_t)$ and each ω_{t+1} is drawn from a set of possible realizations Ω_t. Information realizations express the uncertain changes over the planning horizon $\mathcal{T}$ that become known at a particular time. For the sake of concreteness, we adopt a one-to-many network topology and assume that uncertainty arises exclusively from customer demands. While more papers consider periodic information updates, i.e., demands are fully realized at the end of each period $t \in \mathcal{T} = \{0, 1, ..., T-1\}$, some works assume that demands are unveiled upon the arrival of a vehicle at a customer site (e.g., Giesen et al., 2010). The MDP for an SDIRP can be represented as follows.

2.3.2.1 State (S_t)

A state S_t contains all the information required to select a decision at time t. It may include static and dynamic information related to both the supplier and customers, such as the current inventory levels (I_{ipt}) and further contextual information (Φ_t). This contextual information may contain the maximum storage capacity of each customer ($\bar{U}_i$); the distances between customers and customer-supplier (c_{ij}); the holding and stockout costs (h_{ip}, z_{ip}); and the H historical demand observations of each customer i for a given product p ($\hat{D}_{ipt} = \{D_{ipt-1}, ..., D_{ipt-H}\}$). Therefore, Φ_t may be defined as $\Phi_t = (\bar{U}_i, c_{ij}, h_{ip}, z_{ip}, \hat{D}_{ipt})_{i \in \mathcal{N}', (i,j) \in \mathcal{E}, p \in \mathcal{P}}$.

Regarding periodic models, the planning horizon is typically split into several identical time periods so that a decision point t corresponds to the beginning of period t, i.e., deliveries are made "first thing in the morning" (Crama et al., 2018). Therefore, it is common to assume that a vehicle starts and ends its route at a central depot during each period (refer to Section 2.3.2.5). For instance, Kleywegt et al. (2004) control vehicle routes and customer replenishments with daily planning, and McKenna et al. (2020) consider that a decision point occurs every six hours. For the sake of concreteness, let us introduce $I_t = (I_{ipt})_{i \in \mathcal{N}, p \in \mathcal{P}}$ as a $|N+1|$ dimensional vector that contains the supplier's and customers' inventory levels at the beginning of period t. Therefore, the state can be summarized as $S_t = (I_t, \Phi_t)$, where S_0 represents the initial state.

Concerning real-time models, a decision point t refers to when a vehicle arrives at a customer location (refer to Section 2.3.2.6). Thus, the decision state may include additional information when compared to a periodic model, such as the vehicle's current location (ψ_t) and load (χ_t). For example, Brinkmann et al. (2020) approach a bike-sharing system problem in which a decision point occurs whenever the vehicle arrives at a station. The new state might be $S_t = (I_t, \Phi_t, \psi_t, \chi_t)$, in which $\psi_t \in \{0, 1, ..., N\}$ and $\chi_t \in \{0, 1, ..., Q\}$.

2.3.2.2 Decision (a_t)

A decision at time t is defined by $a_t \in \mathcal{A}(S_t)$ and has four components disclosing inventory and routing decisions. The inventory component of each customer i is modeled by the

replenishment quantity q_{ipt} and by the inventory after delivery or pickup I_t^a. Let us consider $q_t = (q_{ipt})_{\ i \in \mathcal{N}',\ p \in \mathcal{P}}$, so that $I_t^a = I_t + q_t$. The routing component is defined by the binary variables y_{ikt} and x_{ijkt}. Let us introduce $y_t = (y_{ikt})_{\ i \in \mathcal{N},\ k \in \mathcal{K}}$ and $x_t = (x_{ijkt})_{\ (i,j) \in \mathcal{E},\ k \in \mathcal{K}}$. The combination of S_t and a_t leads to a post-decision state $S_t^a = (S_t, a_t)$, where inventories and further period-dependent information regarding the supplier and customers are updated, according to the initial inventories and delivery quantities. A decision $a_t = (q_t, I_t^a, y_t, x_t)$ at period t is valid if the following constraints are satisfied:

$$I_{0pt}^a = I_{0pt} + r_{pt} - \sum_{i \in \mathcal{N}'} \sum_{k \in \mathcal{K}} q_{ipkt}, \quad \forall\ p \in \mathcal{P},\ t \in \mathcal{T} \tag{2.1}$$

$$I_{ipt}^a = I_{ipt} + \sum_{k \in \mathcal{K}} q_{ipkt}, \quad \forall\ i \in \mathcal{N}',\ p \in \mathcal{P},\ t \in \mathcal{T} \tag{2.2}$$

$$\sum_{p \in \mathcal{P}} (I_{ipt} + \sum_{k \in K} q_{ipkt}) \leq \bar{U}_i, \quad \forall\ i \in \mathcal{N}', t \in \mathcal{T} \tag{2.3}$$

$$\sum_{p \in \mathcal{P}} q_{ipkt} \leq Q_k\ y_{ikt}, \quad \forall\ i \in \mathcal{N}', k \in \mathcal{K}, t \in \mathcal{T} \tag{2.4}$$

$$\sum_{i \in \mathcal{N}'} \sum_{p \in \mathcal{P}} q_{ipkt} \leq Q_k\ y_{0kt}, \quad \forall\ k \in \mathcal{K}, t \in \mathcal{T} \tag{2.5}$$

$$\sum_{j \in \mathcal{N}: j \neq i} x_{jikt} = y_{ikt}, \quad \forall\ i \in \mathcal{N}, k \in \mathcal{K}, t \in \mathcal{T} \tag{2.6}$$

$$\sum_{j \in \mathcal{N}: j \neq i} x_{ijkt} = y_{ikt}, \quad \forall\ i \in \mathcal{N}, k \in \mathcal{K}, t \in \mathcal{T} \tag{2.7}$$

$$\sum_{i \in \mathcal{B}} \sum_{j \in \mathcal{B}: j \neq i} x_{ijkt} \leq \sum_{i \in \mathcal{B}} y_{ikt} - y_{\ell kt}, \quad \forall\ \mathcal{B} \subseteq \mathcal{N}', |\mathcal{B}| \geq 2, k \in \mathcal{K}, t \in \mathcal{T}, \ell \in \mathcal{B} \tag{2.8}$$

$$\sum_{k \in \mathcal{K}} y_{ikt} \leq 1, \quad \forall\ i \in \mathcal{N}', t \in \mathcal{T} \tag{2.9}$$

$$I_{ipt}^a \geq 0, \quad \forall\ i \in \mathcal{N}, p \in \mathcal{P}, t \in \mathcal{T} \tag{2.10}$$

$$q_{ipkt} \geq 0, \quad \forall\ i \in \mathcal{N}', p \in \mathcal{P}, k \in \mathcal{K}, t \in \mathcal{T} \tag{2.11}$$

$$y_{ikt} \in \{0,1\}, \quad \forall\ i \in \mathcal{N}, k \in \mathcal{K}, t \in \mathcal{T} \tag{2.12}$$

$$x_{ijkt} \in \{0,1\}, \quad \forall\ (i,j) \in \mathcal{E}, k \in \mathcal{K}, t \in \mathcal{T} \tag{2.13}$$

Constraints (2.1) and (2.2) update the inventory levels at the supplier and each customer i at each period t. Constraints (2.3) ensure that the inventory level of each customer i does not exceed the maximum level $\bar{U}_i$. Constraints (2.4) link the delivery and visit decision variables. Constraints (2.5) guarantee that the total delivery quantities assigned to the vehicle k do not exceed its capacity Q_k. Constraints (2.6) and (2.7) ensure the vehicle's flow conservation. Constraints (2.8) are subtour elimination constraints. Constraints (2.9) limit the number of visits to each customer i in each period t. Constraints (2.10) - (2.13) ensure the domains of decision variables.

2.3.2.3 Stochastic information (ω_{t+1}) and transition function (Γ)

The exogenous noise capturing the uncertainty in customer demands, typically represented by information realizations $\omega_{t+1} = (D_t^{\omega})$ with $D_t^{\omega} = (D_{ipt}^{\omega})_{\, i \in \mathcal{N}', p \in \mathcal{P}}$, is independent of both S_t and a_t. The combination of post-decision state S_t^a and demand realization ω_{t+1} originates a new state $S_{t+1} = \Gamma(S_t^a, \omega_{t+1})$ according to a deterministic transition function (Γ) representing stochasticity. In this new state, inventories are updated, and costs are computed. Regarding periodic models, we assume that the delivery quantities determined at decision point t, i.e., the beginning of period t, are used to satisfy the demands only realized at $t+1$, i.e., the end of period t. The state variables are updated as follows:

$$I_{0pt+1} = I_{0pt}^a, \quad \forall\ p \in \mathcal{P}, t \in \mathcal{T} \tag{2.14}$$

$$I_{ipt+1} = I_{ipt}^a - D_{ipt}^{\omega} + l_{ipt}, \quad \forall\ i \in \mathcal{N}', p \in \mathcal{P}, t \in \mathcal{T} \tag{2.15}$$

$$l_{ipt} \geq 0, \quad \forall\ i \in \mathcal{N}', p \in \mathcal{P}, t \in \mathcal{T} \tag{2.16}$$

The resulting cost has three components. The first two relate to inventory costs (holding and stockout), and the third one to routing costs:

$$C(S_t^a, \omega_{t+1}) = \sum_{i \in \mathcal{N}} \sum_{p \in \mathcal{P}} h_{ip}\ I_{ipt+1} + \sum_{i \in \mathcal{N}'} \sum_{p \in \mathcal{P}} z_{ip}\ l_{ipt} + \sum_{(i,j) \in \mathcal{E}} \sum_{k \in \mathcal{K}} c_{ij}\ x_{ijkt} \tag{2.17}$$

2.3.2.4 Policy (π)

A solution for the SDIRP is a policy $\pi \in \Pi$, i.e., any parametric or non-parametric method, that assigns a feasible decision $a_t = A^{\pi}(S_t)$ to every state S_t. The optimal policy π^* satisfies the Bellman equation depicted in Equation (2.18), minimizing the expected amount of unsatisfied demand, inventory holding and routing costs over a given planning horizon $\mathcal{T}$. For each state S_t, the expected total costs comprise an immediate cost $C(S_t, a_t)$ plus the expected sum of costs of future states given S_t and a_t.

$$A^{\pi^*}(S_t) = \arg \min_{a_t \in \mathcal{A}(S_t)} \left\{ C(S_t, a_t) + \mathbb{E}\left[\sum_{t'=t+1}^{T-1} C\left(S_{t'}, A^{\pi^*}(S_{t'})\right) \mid (S_t, a_t) \right] \right\} \tag{2.18}$$

The problem, therefore, consists of finding a feasible policy π that solves the objective function (2.19), given the initial state S_0.

$$\min_{\pi \in \Pi} \mathbb{E}\left\{ \sum_{t \in \mathcal{T}} C_t(S_t, A^{\pi}(S_t)) \mid S_0 \right\} \tag{2.19}$$

2.3.2.5 Example of a periodic model

Figure 2.3 shows an MDP for a periodic SDIRP. The state S_t represents the inventory levels at each customer and supplier at the beginning of the planning period t. The decision a_t depicts a delivery plan corresponding to a sequence of customers to visit during the time period t and the amount of product to deliver. The vehicle is planned to visit customer *2*

first, delivering two units, then customer *3*, providing one unit, and finally return to the central depot empty. After deliveries are completed and the vehicle has returned to the supplier, the system evolves to a post-decision state S_t^a, representing the updated inventories. Then, as soon as the real demands regarding the current period t are revealed (i.e., two units demanded at customer *1* and one unit demanded at customer *2* and *3*), the system transits to a new state S_{t+1}, inventories are updated, and costs are computed.

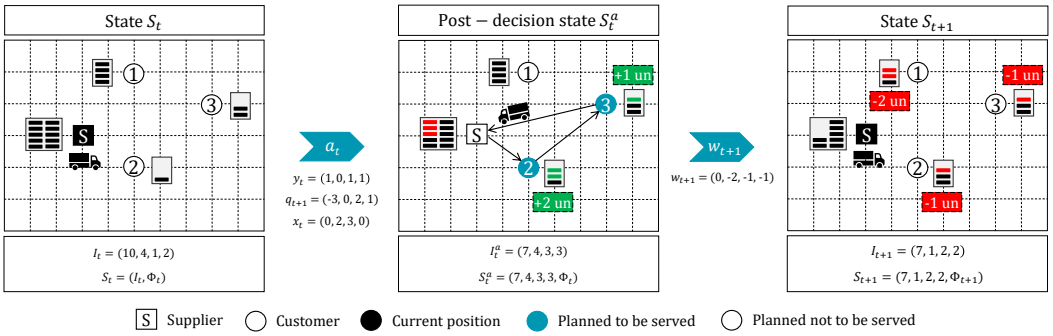


Figure 2.3: Conventional MDP for a periodic SDIRP.

2.3.2.6 Example of a real-time model

Figure 2.4 depicts an MDP for a real-time SDIRP. In the proposed model, a new decision is taken whenever the vehicle arrives at a customer location, performing a delivery or pickup. The delivery plan consists of identifying the quantities to deliver or pick up at the current customer and the next customer to visit. At decision point t, the vehicle departs from customer *2* to customer *3*, first delivering a unit of product to customer *2*. At this moment, the inventory of customer *2* is updated, and the system transits to S_t^a. When the vehicle arrives at customer *3* and inventories are revealed, the system evolves to a new state S_{t+1} and a new decision is needed to determine the pickup or delivery quantities at customer *3*, as well as the next location. On the other hand, a_t may already comprise the pickup or delivery quantities of customer *3* when the vehicle departs from customer *2*. Moreover, and unlike this example, a delivery plan might incorporate a sequence of customers to be visited instead of a single one. Therefore, information realizations can be achieved when the vehicle arrives at the last customer of the planned sequence or whenever a single customer is visited. In this scenario, the previously scheduled sequence of customers might be replanned.

2.4. Problem settings

According to the problem settings taxonomy detailed in Table 2.3, we describe each proposed category (Sections 2.4.1.1-2.4.3.7) and classify the selected papers (Tables 2.4 and 2.5).

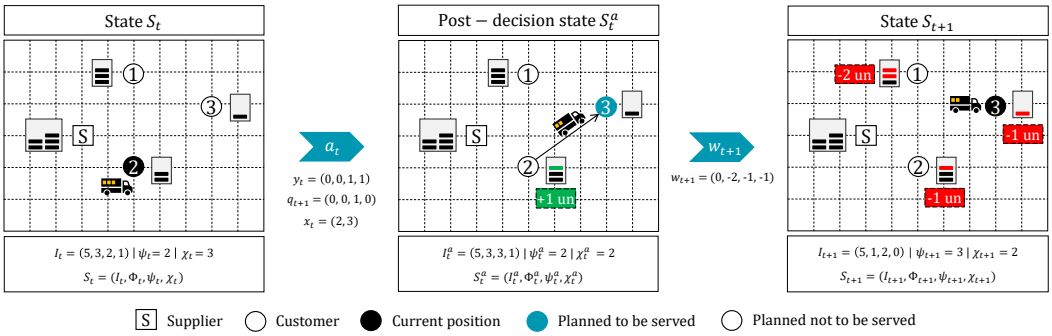


Figure 2.4: Conventional MDP for a real-time SDIRP.

Table 2.3: Problem settings taxonomy.

Class	Dimension	Categories			
General	*Endpoint structure* (2.4.1.1)	One-to-one	One-to-many	Many-to-one	Many-to-many
	Uncertainty source (2.4.1.2)	Demands	Environment		
	Time horizon (2.4.1.3)	Finite	Infinite		
	Objective function (a) (2.4.1.4)	Single objective	Multiple objective		
	Objective function (b) (2.4.1.4)	Inventory costs	Routing costs	Additional costs	
	Application sector (2.4.1.5)	General	Specific application		
Inventory	*Stockouts* (2.4.2.1)	Lost sales	Backorders		
	Items (2.4.2.2)	Single	Multiple		
	Perishability (2.4.2.3)	Perishable items	Non-perishable items		
	Inventory policy (2.4.2.4)	Maximum-level	Order-up-to-level		
Routing	*Fleet composition* (2.4.3.1)	Homogeneous	Heterogeneous	Uncapacitated	
	Fleet size (2.4.3.2)	Single vehicle	Multiple vehicles	Unconstrained	
	Time windows (2.4.3.3)	Allowed	Not allowed		
	Type of routes (2.4.3.4)	Direct	Multiple		
	Logistical context (2.4.3.5)	Delivery	Pickup	Pickup & delivery	
	Emergency transshipment (2.4.3.6)	Allowed	Not allowed		
	Split delivery (2.4.3.7)	Allowed	Not allowed		

2.4.1 General features

This section details the general dimensions of the problem, i.e. those that are not related just to inventory or routing, including the endpoint structure, source and type of uncertainty, time horizon, objective function and application sector.

2.4.1.1 Endpoint structure

The topology of a general IRP reveals the number of suppliers and customer sites under consideration. Around 80% of the collected papers follow a one-to-many structure (e.g., Roldán et al., 2016; Cubillos et al., 2022), in which a single supplier is responsible for replenishing a set of customers using a fleet of vehicles. This predominance suggests a need to investigate further topologies. For instance, Jaillet et al. (2002) include various satellite facilities where drivers might be refilled during delivery. Elbek and Wøhlk (2016) solve a waste collection problem considering a many-to-many topology, assuming two treatment facilities (i.e., paper and glass). Moreover, Markov et al. (2020) address a recyclable waste collection problem for large containers considering intermediate facilities. On the other

hand, Mirzapour Al-e hashem et al. (2019) consider a stochastic transshipment-enabled IRP where a set of suppliers serves a single assembly plant (i.e., many-to-one). Finally, Violi et al. (2023) assess a three-echelon agri-food supply chain composed of a central warehouse replenished by a set of production plants and a set of retailers supplied by the warehouse (i.e., many-to-one-to-many).

2.4.1.2 Uncertainty source

The primary source of uncertainty comes from customer demands (e.g., Crama et al., 2018), although additional uncertainties should also be considered to reflect practical conditions, including stochastic travel times or network disruptions (e.g., Morales et al., 2018), supply (e.g., Hasturk et al., 2024), purchasing prices (e.g., Cuellar-Usaquén et al., 2024), and time windows (i.e., environmental factors). Assuming customer demands are partially unknown, they can be modeled using a probability distribution function or machine learning-based point forecasts. For instance, Coelho et al. (2014a) consider stationary and non-stationary (i.e., seasonal and correlated) demands using information from the previous 50 historical periods. However, there are circumstances under which customer demands are entirely unknown and only revealed at the end of a given period (e.g., Bertazzi et al., 2025). Jia et al. (2025) propose a fully data-driven approach for the online IRP that does not rely on any distributional assumption.

Regarding alternative sources of uncertainty, McKenna et al. (2020) tackle the military IRP where stochasticity is linked to the potential delivery failures caused by hostile actions by non-friendly forces, mechanical failures, and extreme weather conditions. Cuellar-Usaquén et al. (2024) and Hasturk et al. (2024) identify supply volumes as an additional source of uncertainty, which remains underexplored in the SDIRP literature. Moreover, Cuellar-Usaquén et al. (2024) further consider that the suppliers' purchase prices are revealed periodically.

2.4.1.3 Time horizon

We categorize this time dimension into finite and infinite planning horizons. While a finite planning horizon focuses on optimizing decisions within a limited time frame, an infinite horizon emphasizes long-term operational efficiency without looking at a predefined endpoint. A common strategy to dynamically solve finite planning horizons is to employ an RH approach, which may utilize lookahead models to consider both long-term and immediate costs. For example, Federgruen and Tzur (1996) derive an efficient procedure to detect minimal forecast horizons addressing five classes of dynamic planning problems. Dror and Ball (1987) propose a procedure for reducing a long-term IRP to a single-period problem by defining single-period costs that reflect long-term costs, safety stock levels and the subset of customers to fulfill during that period. The effects of a single-period decision in the later ones are accounted for by penalizing earlier deliveries to retailers with sufficient inventory and providing incentives for delivering to customers not required by the constraints. Jaillet et al. (2002) reduce a distribution problem from an annual time base to a biweekly rolling planning period using incremental cost approximations. The authors

adopt the RH framework of Bard et al. (1998), considering a biweekly planning period where solely the customers scheduled for the first week are routed. Then, the two-week planning horizon is shifted by a week, and the process repeats. Elbek and Wøhlk (2016) handle planning periods of eleven days. Moreover, Crama et al. (2018) apply four different solution methods in an RH fashion during a planning horizon of thirty periods.

When handling infinite planning horizons, the focus shifts from detailed scheduling decisions to an overall distribution strategy. These problems are typically formulated as MDPs (i.e., discrete-time stochastic control processes). For instance, Berman and Larson (2001) apply stochastic dynamic programming to derive a policy that minimizes the expected costs, including early and late deliveries, product shortfalls and non-empty returns to the central depot over an infinite planning horizon. Kleywegt et al. (2002) settle the problem using value function approximations to maximize the expected discounted value, including sales revenues, production, routing, inventory costs, and stockout penalties. Hvattum et al. (2009) reduce the dimensions of the infinite MDP by designing a finite scenario tree approximation for each state. At last, McKenna et al. (2020) solve a military IRP modeled as an MDP employing approximate policy iteration methods in which decisions might be taken sequentially over time and under uncertainty.

2.4.1.4 Objective function

The main goal of an SDIRP is to minimize the total expected costs over a given planning horizon. Typically, these costs comprise inventory holding, stockout and transportation costs, as depicted in the objective function (2.19). However, there might be additional cost or revenue components. For example, Hvattum et al. (2009) consider a revenue for each completed delivery. Moreover, transportation costs might comprise a fixed cost according to vehicle usage (e.g., Elbek and Wøhlk, 2016), and additional penalties due to early deliveries might be considered (e.g., Dror and Ball, 1987). In waste collection problems, stockouts are not applicable, but rather an overflow waste cost (e.g., Markov et al., 2020). For instance, Mes et al. (2014) assign a cost to the action of emptying the containers. Furthermore, Cui et al. (2023) introduce a service violation index to assess the likelihood of service violations at the inventory level and develop a robust model to minimize this index.

Building on this, most SDIRP papers intend to optimize a single objective function. As an exception, Mirzapour Al-e hashem et al. (2019) propose a multi-objective model that aims to (1) minimize the expected value of the supply chain costs, including holding, shortage, transportation and disposal costs, and (2) minimize the total quantity of the greenhouse gas emission produced. The scarcity of multi-objective studies, therefore, highlights an opportunity for further investigation.

2.4.1.5 Application sector

The SDIRP comprises a wide range of real-world domains. Berman and Larson (2001), Jaillet et al. (2002) and Adelman (2004) focus on industrial oil, gas and fuel distribution problems under stochastic customer demands. Dror and Ball (1987) develop a model for a propane-distribution company. Trudeau and Dror (1992) handle the distribution of heating

oil to multiple customers. Kleywegt et al. (2002) and Kleywegt et al. (2004) address a partnership between a producer and distributor of air products. Violi et al. (2020) tackle the challenges of managing an agri-food supply chain. Raba et al. (2020) focus on optimizing replenishment strategies for silos in multiple farms. McKenna et al. (2020) center their work on military combat environments that the United States Army faces, in which air-based resupply operations are critical. Jia et al. (2025) address the problem of spare parts. Furthermore, bike-sharing systems (e.g., Brinkmann et al., 2019, 2020) and waste collection problems (e.g., Nolz et al., 2014; Mes et al., 2014; Markov et al., 2020) are increasingly recognized for their practical importance within the field.

2.4.2 Inventory features

This section explores the characteristics of the inventory management subproblem, including the type of stockout, the number of items and their perishability, and the replenishment inventory policy.

2.4.2.1 Stockouts

Inventory shortages inevitably arise when facing stochastic customer demands. We have identified three strategies to deal with out-of-stocks: backlogging (e.g., Reiman et al., 1999), lost sales (e.g., Giesen et al., 2010) and emergency transshipments (e.g., Achamrah et al., 2022). For the latter, we refer to Section 2.4.3.6. Backlogging enables postponing the unmet demand to subsequent periods of the planning horizon (e.g., Li et al., 2016). For example, Cui et al. (2023) and Jia et al. (2025) consider that any excess demand is backlogged and a shortage cost is applied. Cavagnini et al. (2022) evaluate the inclusion of emergency transshipments in addition to classical backorders. In the context of collector-managed IRP (e.g., Mes et al., 2014), these concepts do not apply, but rather penalties linked to overfilled inventories and emergency collection costs. Regarding infectious medical waste collection, Nolz et al. (2014) assume that the unitary excess inventory costs apply when the container's maximum storage capacity of a given pharmacy is exceeded. These costs represent the risk associated with each box in excess. Furthermore, Markov et al. (2020) consider that a container overflow incurs a penalty, which implies backordering inventory decisions where the number of backorder days is restricted to one.

2.4.2.2 Items

Incorporating multiple items or products in our models leads to a significant increase in computational effort due to additional complexities, including demand variations and storage constraints. As a result, around 85% of the papers focus on models with a single item (e.g., Cubillos et al., 2022). The following exceptions stand out. Vonolfen et al. (2013) focus on a large-scale VMI scenario for supermarkets that entails many products. Elbek and Wøhlk (2016) address a multi-period collection problem of recyclable materials, including paper and glass. In situations where multiple products are involved, product substitution can be a viable recourse action facing the occurrence of stockouts (e.g., Achamrah et al., 2022). The lack of multi-item formulations indicates a promising area for future research.

Table 2.4: Problem settings classification (general features).

Authors (year)	Endpoint structure	Uncert. source	Time horizon	Objective function				Application sector
				Objective	Transp. costs	Invent.	Addit. costs	
1. Dror and Ball (1987)	1-M	D	F	Single	✓	✓	✓	Energy
2. Trudeau and Dror (1992)	1-M	D	F	Single	✓	✓	✓	Energy
3. Minkoff (1993)	1-M	D	∞	Single	✓	✓	-	General
4. Reiman et al. (1999)	1-M	D, T	∞	Single	✓	✓	-	General
5. Berman and Larson (2001)	1-M	D	∞	Single	✓	✓	✓	Energy
6. Jaillet et al. (2002)	M-M	D	F	Single	✓	✓	-	Energy
7. Kleywegt et al. (2002)	1-M	D	∞	Single	✓	✓	✓	Energy
8. Kleywegt et al. (2004)	1-M	D	∞	Single	✓	✓	✓	Energy
9. Adelman (2004)	1-M	D	∞	Single	✓	✓	-	Energy
10. Giesen et al. (2005)	1-M	D	F	Single	✓	✓	-	General
11. Jarugumilli and Grasman (2007)	1-M	D	F	Single	✓	✓	-	General
12. Hvattum et al. (2009)	1-M	D	∞	Single	✓	✓	✓	General
13. Hvattum and Løkketangen (2009)	1-M	D	∞	Single	✓	✓	✓	General
14. Giesen et al. (2010)	1-M	D	F	Single	✓	✓	-	General
15. Bertazzi et al. (2013)	1-M	D	F	Single	✓	✓	-	General
16. Vonolfen et al. (2013)	1-M	D	∞	Single	✓	✓	-	Groceries
17. Nolz et al. (2014)	1-M	D	F	Single	✓	✓	-	Waste collection
18. Coelho et al. (2014a)	1-M	D	F	Single	✓	✓	✓	General
19. Mes et al. (2014)	1-M	D	F	Single	✓	✓	✓	Waste collection
20. Bertazzi et al. (2015)	1-M	D	F	Single	✓	✓	-	General
21. Roldán et al. (2016)	1-M	D	F	Single	✓	✓	-	General
22. Elbek and Wøhlk (2016)	M-M	D	F	Single	✓	-	✓	Waste collection
23. Li et al. (2016)	1-M	D	F	Single	✓	✓	-	Groceries
24. Morales et al. (2018)	1-M	D, T	F	Single	✓	✓	-	General
25. Crama et al. (2018)	1-M	D	F	Single	✓	-	✓	General
26. Mirzapour Al-e hashem et al. (2019)	M-1	D	F	Multi	✓	✓	✓	Medicines
27. Brinkmann et al. (2019)	1-M	D	F	Single	-	✓	-	Bike-sharing
28. Markov et al. (2020)	M-M	D	F	Single	✓	✓	✓	Waste collection
29. Brinkmann et al. (2020)	1-M	D	F	Single	-	✓	-	Bike-sharing
30. Violi et al. (2020)	M-1-M	D	F	Single	✓	✓	✓	Agri-food
31. McKenna et al. (2020)	1-M	T	∞	Single	-	-	✓	Military
32. Raba et al. (2020)	1-M	D	F	Single	✓	✓	-	Animal-feed
33. Cubillos et al. (2022)	1-M	D	F	Single	✓	✓	-	General
34. Achamrah et al. (2022)	1-M	D	F	Single	✓	✓	✓	General
35. Neves-Moreira et al. (2022)	1-M	D	F	Single	✓	✓	-	General
36. Cavagnini et al. (2022)	1-M	D	F	Single	✓	✓	✓	General
37. Cui et al. (2023)	1-M	D	F	Single	-	-	✓	Energy
38. Li et al. (2023)	1-M	D	F	Single	✓	✓	-	Energy
39. Violi et al. (2023)	M-1-M	D	F	Single	✓	✓	-	Agri-food
40. Cuellar-Usaquén et al. (2024)	M-1	S, P, D	F	Single	✓	-	✓	Groceries
41. Ortega et al. (2024)	1-M	D	F	Single	✓	✓	-	Groceries
42. Chiu et al. (2024)	1-M	D	F	Single	✓	✓	-	General
43. Jia et al. (2025)	1-M	D	∞	Single	✓	✓	-	Spare parts
44. Greif et al. (2024)	1-M	D	F	Single	✓	✓	-	General
45. Hasturk et al. (2024)	1-M	S, D	∞	Single	✓	✓	✓	Energy
46. Liang et al. (2024)	M-M	D	F	Single	-	-	✓	Bike-sharing

Legend: (1-M) - One-to-many | (M-1) - Many-to-one | (M-1-M) - Many-to-one-to-many | (M-M) - Many-to-many | D - Demand | F - Finite | Multi - Multiple | PP - Purchasing price | S - Supply | T - Travel time

2.4.2.3 Perishability

Shaabani (2022) is the author of the latest literature review focusing exclusively on the perishable IRP. This characteristic is particularly important in certain applications like food supply chains, pharmaceuticals, and blood distribution. Accordingly, perishable items might increase the complexity of inventory and routing decisions through shelf-life and expiration date constraints, for example. Despite not being the main focus of this review, we highlight three SDIRP papers that address this variant. Crama et al. (2018) handle a

perishable IRP with dynamic stochastic demands, in which the oldest units are sold first, following a first-in-first-out approach. Violi et al. (2020) and Violi et al. (2023) address an agri-food supply chain, including perishable products with demand uncertainty. Cuellar-Usaquén et al. (2024) solve a problem faced by agri-food e-commerce platforms in purchasing several perishable products, collecting them from multiple producers and delivering them to a single warehouse. The scarcity of such studies points to a clear research gap in addressing perishability within SDIRPs.

2.4.2.4 Inventory policy

According to the existing IRP literature, we consider two main inventory replenishment policies: maximum-level (ML) and order-up-to-level (OU). An ML policy empowers the supplier to freely decide on the quantities of a specific product to deliver to a set of customers, taking into account the available resources of each customer, such as the current inventory level, capacity and expected demand (e.g., Hvattum et al., 2009). The delivery quantity is such that the inventory level at each customer after delivery and consumption is not greater than its maximum level, i.e., capacity (Archetti et al., 2011). Moreover, customers are not required to be replenished up to their full capacity. On the other hand, an OU policy states that every time a customer is visited, their inventory is refilled to its total capacity (e.g., Cubillos et al., 2022).

The following studies highlight both inventory policies. Coelho et al. (2014a) initially propose a reactive approach that employs an (s,S) policy, where s represents the reorder point of a given customer and S its inventory capacity, translating into an OU policy. Afterwards, the proactive approach adopts an ML policy where delivery quantities are computed by an adaptive large neighborhood search (ALNS) heuristic, respecting the vehicle and customer capacities. Roldán et al. (2016) employ a fixed quantity policy in which each customer consistently receives a fixed quantity, defined as a fraction of its maximum inventory level, along with an (s,S) policy for different values of s. Raba et al. (2020) explore distinct refilling strategies that minimize the total costs. Regarding waste collection problems, the inventory policy entails emptying the containers whenever visited, being classified as an OU policy (e.g., Markov et al., 2020; Nolz et al., 2014).

2.4.3 Routing features

This section addresses the features of the vehicle routing subproblem, namely the fleet composition and size, consideration of time windows, type of routes, logistical context, allowance of emergency transshipment or direct shipping, and split delivery.

2.4.3.1 Fleet composition

The fleet composition describes the homogeneity of a given fleet of vehicles. Most selected SDIRP papers consider homogeneous fleets of vehicles with the same capacity (e.g., McKenna et al., 2020), simplifying part of the problem. However, heterogeneous fleets are more than common in many industries, such as food and medical goods (e.g., Mirzapour Al-e hashem et al., 2019) distribution. Additionally, the carrying capacity of these vehicles

may be directly related to short-term or longer-distance deliveries. As an exception, Li et al. (2016) and Cui et al. (2023) model a set of vehicles with unlimited capacity. This can be useful to reduce the complexity of decisions and shift the focus toward demand fluctuations and associated uncertainties.

2.4.3.2 Fleet size

The fleet size refers to the number of vehicles responsible for deliveries and pickups at suppliers and customers. We found that approximately two-thirds of the selected studies utilize multiple vehicles (e.g., Cavagnini et al., 2022), rather than relying on a single vehicle (e.g., Cubillos et al., 2022). Using a multi-vehicle fleet enhances delivery flexibility and efficiency by reducing delivery times and allowing for larger and more complicated networks, which is not possible with just one vehicle. Furthermore, Minkoff (1993), Kleywegt et al. (2002) and Jaillet et al. (2002) assume an unlimited number of vehicles, and Coelho et al. (2014a) consider an extra capacity by including direct transportation from the central depot in case the single-vehicle capacity is insufficient.

2.4.3.3 Time windows

Although time windows have been extensively approached in static deterministic (e.g., Alinaghian et al., 2021) and stochastic (e.g., Liu and Lee, 2011) IRP variants, they still hold significant potential for further exploration in road-based SDIRPs. Morales et al. (2018) adapt the formulation proposed by Archetti et al. (2007) and include the time window constraints based on the model developed by Pérez Kaligari and Guerrero Rueda (2015). Markov et al. (2020) address a recyclable waste collection problem for large containers comprising a central depot, multiple customers and multiple dumps playing the role of intermediate facilities for unloading the collected waste. For depots and dumps, time windows represent working hours, while for containers, they can be set to avoid collections during rush hours, school times, and other particular periods. Finally, Ortega et al. (2024) incorporate intra-day inventory depletion into an SDIRP where replenishment and routing decisions under time window restrictions are scheduled in every subperiod.

2.4.3.4 Type of routes

The routing subproblem is commonly addressed by constructing vehicle routes that cover multiple customer locations (e.g., Neves-Moreira et al., 2022), which corresponds to solving the classical VRP. In contrast, a limited number of studies, primarily focused on the inventory management subproblem, adopt a simplified routing structure based on direct deliveries (e.g., McKenna et al., 2020; Cavagnini et al., 2022), although this does not reflect much of reality. Additionally, Reiman et al. (1999) propose a dynamic choice between a prespecified tour of all retailers and direct shipping, and Kleywegt et al. (2004) extend the direct delivery routes obtained by the algorithm presented in Kleywegt et al. (2002) to multiple routing.

2.4.3.5 Logistical context

The logistical context identifies whether the problem handles only pickups, deliveries, or both pickup & delivery (P&D). Most of the selected papers primarily focus on delivering between a central depot and a designated set of customers. However, there are several contexts where pickups assume the leading role, such as waste collection problems. Therefore, a central depot usually operates as the destination point for the collected items from multiple pickup locations (e.g., Nolz et al., 2014; Mes et al., 2014; Elbek and Wøhlk, 2016; Markov et al., 2020).

P&D arise during the routing stage, letting a specific product be picked up at a particular customer location and delivered to another one in the same period. For example, bike-sharing systems represent a specific extension of the IRP that incorporates both P&D operations (e.g., Liang et al., 2024; Brinkmann et al., 2020). Simultaneous P&D implies that a given customer can receive and send products in the same period (e.g., Neves-Moreira et al., 2022). Despite its relevance, it remains underexplored in the SDIRP research. On the other hand, non-simultaneous P&D limits customers to either deliveries or pickups in a given period. Mirzapour Al-e hashem et al. (2019) propose a transshipment-enabled model to effectively control transportation costs, reduce greenhouse gas emissions, and manage uncertainties.

2.4.3.6 Emergency transshipment

In IRPs, transshipments are often operated as recourse actions to face the occurrence of stockouts through direct emergency shipping (i.e., from suppliers) or lateral emergency transshipments (e.g., Cavagnini et al., 2022). Subcontracted carriers are typically responsible for these actions (Coelho et al., 2014a). Besides lateral transshipments and lost sales, Achamrah et al. (2022) also consider product substitutions as recourse actions. Moreover, Markov et al. (2020) introduce the notion of emergency collections in which the inventory is reset to zero. As the latest example identified in the SDIRP literature, Hasturk et al. (2024) consider that, in case the vehicle capacity is not enough given the stochasticity of demands, emergency shipments may occur at a given cost.

2.4.3.7 Split delivery

Split delivery constitutes a further source of flexibility. For instance, in retail industries, high-demand products may need to be delivered to a store from multiple warehouses by different vehicles on the same day. Another example concerns emergency supplies, in which hospitals may receive multiple deliveries of emergency medical supplies in a given period of time, relying on different suppliers. However, because split delivery introduces additional complexity into the existing models, it is uncommon to notice this variant in the current SDIRP literature. As an exception, Hvattum et al. (2009) and Hvattum and Løkketangen (2009) admit this possibility by using a decision variable that allows a customer to be visited by more than one vehicle in each planning period.

Table 2.5: Problem settings classification (inventory and routing features).

Authors (year)	Inventory				Routing						
	Stockout	Item	Perish.	Inventory policy	Fleet comp.	Fleet size	Time wind.	Type routes	Log. cont.	Emg. ship.	Split deliv.
1. Dror and Ball (1987)	LS	S	-	OU	Hom	M	-	M	D	-	-
2. Trudeau and Dror (1992)	LS	S	-	OU	Hom	M	-	M	D	-	-
3. Minkoff (1993)	LS	S	-	ML	Hom	∞	-	M	D	-	-
4. Reiman et al. (1999)	BL	S	-	ML	Hom	S	-	Dir, M	D	-	-
5. Berman and Larson (2001)	LS	S	-	ML	Hom	S	-	M	D	-	-
6. Jaillet et al. (2002)	LS	S	-	ML	Hom	∞	-	M	D	-	-
7. Kleywegt et al. (2002)	LS	S	-	ML	Hom	M	-	Dir	D	-	-
8. Kleywegt et al. (2004)	LS	S	-	ML	Hom	M	-	M	D	-	-
9. Adelman (2004)	LS	M	-	ML	Hom	M	-	M	D	-	-
10. Giesen et al. (2005)	LS	S	-	ML	Hom	S	✓	M	D	-	-
11. Jarugumilli and Grasman (2007)	LS	S	-	ML	Hom	S	-	M	D	-	-
12. Hvattum et al. (2009)	LS	S	-	ML	Hom	M	-	M	D	-	(✓)
13. Hvattum and Løkketangen (2009)	LS	S	-	ML	Hom	M	-	M	D	-	(✓)
14. Giesen et al. (2010)	LS	S	-	ML	Hom	S	-	M	D	-	-
15. Bertazzi et al. (2013)	LS	S	-	OU	Hom	S	-	M	D	-	-
16. Vonolfen et al. (2013)	LS	M	-	ML	Hom	M	-	M	D	-	-
17. Nolz et al. (2014)	LS	S	-	OU	Hom	S	-	M	P	-	-
18. Coelho et al. (2014a)	LS	S	-	ML, OU	Hom	S	-	M	D	✓	-
19. Mes et al. (2014)	BL	S	-	OU	Hom	M	-	M	P	-	-
20. Bertazzi et al. (2015)	LS	S	-	OU	Hom	S	-	M	D	-	-
21. Roldán et al. (2016)	LS	S	-	ML, OU	Hom	S	-	M	D	-	-
22. Elbek and Wøhlk (2016)	BL	M	-	OU	Hom	S	-	M	P	-	-
23. Li et al. (2016)	BL	M	-	ML	∞	S	-	M	D	-	-
24. Morales et al. (2018)	-	S	-	ML	Het	M	✓	M	D	-	-
25. Crama et al. (2018)	LS	S	✓	ML	Hom	M	-	M	D	-	-
26. Mirzapour Al-e hashem et al. (2019)	LS, BL	M	-	ML	Het	M	-	M	P&D	-	-
27. Brinkmann et al. (2019)	LS	S	-	ML	Hom	S	-	M	P&D	-	-
28. Markov et al. (2020)	BL	S	-	OU	Het	M	✓	M	P	✓	-
29. Brinkmann et al. (2020)	LS	S	-	ML	Het	M	-	M	P&D	-	-
30. Violi et al. (2020)	LS	S	✓	ML	Hom	M	-	M	D	-	-
31. McKenna et al. (2020)	-	S	-	ML	Hom	M	-	Dir	D	-	-
32. Raba et al. (2020)	LS	S	-	ML	Hom	M	-	M	D	-	-
33. Cubillos et al. (2022)	LS	S	-	OU	∞	S	-	M	D	-	-
34. Achamrah et al. (2022)	LS	M	-	ML	Hom	M	-	M	D	✓	-
35. Neves-Moreira et al. (2022)	-	M	-	ML	Het	M	-	M	P&D	-	-
36. Cavagnini et al. (2022)	LS, BL	S	-	ML	Hom	M	-	Dir	D	✓	-
37. Cui et al. (2023)	BL	S	-	ML	∞	S	-	M	D	-	-
38. Li et al. (2023)	BL	S	-	ML	∞	S	-	M	D	-	-
39. Violi et al. (2023)	LS	S	✓	ML	Hom	M	-	M	D	-	-
40. Cuellar-Usaquén et al. (2024)	-	M	✓	ML	Hom	M	-	M	D	-	-
41. Ortega et al. (2024)	LS	S	-	ML	Hom	M	✓	M	D	-	-
42. Chiu et al. (2024)	LS	S	-	ML	Hom	M	-	M	D	-	-
43. Jia et al. (2025)	BL	S	-	ML	Hom	M	-	M	D	-	-
44. Greif et al. (2024)	LS	S	-	OU	Hom	S	-	M	D	-	-
45. Hasturk et al. (2024)	-	S	-	ML	Hom	M	-	M	D	✓	-
46. Liang et al. (2024)	LS	S	-	ML	Hom	M	-	M	P&D	-	-

Legend: BL - Backlogging | D - Delivery | Dir - Direct | Het - Heterogeneous | Hom - Homogeneous | LS - Lost sales | M - Multiple | ML - Maximum-level | OU - Order-up-to-level | P - Pickup | P&D - Pickup & delivery | S - Single

2.5. Decision policies

Planning across an entire planning horizon with a proactive optimization strategy can be computationally demanding, particularly for longer planning horizons. A practical solution is to decompose the problem into smaller subproblems, computing inventory and routing decisions over shorter time periods. Rolling horizon (RH) approaches have been widely employed for decision-making in stochastic dynamic environments for decades (Sethi and Sorger, 1991). According to the literature on sequential decision-making, the most straightforward class of policies is referred to as myopic. These are applied reactively within RH

frameworks, focusing only on the immediate costs without considering any future information about subsequent periods or states.

The unified framework of Powell (2022b) categorizes four additional types of policies: policy function approximation (PFA), cost function approximation (CFA), direct lookahead approximation (DLA), and value function approximation (VFA). As policy search methods, PFAs are parametric or non-parametric functions that map states to decisions, whereas CFAs are parameterized optimization models that result from modified cost functions (e.g., objective function or constraints). As policies based on lookahead approximations, DLAs guide current-period decisions by optimizing a deterministic or stochastic lookahead model that extends over a finite horizon, while VFAs estimate the value of moving to a subsequent state.

Table 2.6 summarizes the aforementioned policies, which are discussed in detail in Sections 2.5.1-2.5.4, and their hybridization in Section 2.5.5. Section 2.5.6 investigates the learning mechanisms underlying each policy class. At the end of the section, Table 2.8 provides a structured classification of the reviewed studies, including the modeling framework, decision periodicity, policy class, optimization approach, and learning domain.

Table 2.6: Sequential decision policies taxonomy.

Strategy	Decision policy		Code
Myopic	Rolling horizon reoptimization (2.5.1)		RO
Policy search	Policy function approximation (2.5.2.1)		PFA
	Cost function approximation (2.5.2.2)		CFA
Lookahead approximation	Direct lookahead approximation	Rolling horizon with deterministic lookaheads (2.5.3.1)	DLA_a
		Rolling horizon with stochastic lookaheads (2.5.3.2)	DLA_b
		Rollout algorithm (2.5.3.3)	DLA_c
	Value function approximation (2.5.4)		VFA

Source: adapted from Powell (2022b).

2.5.1 Myopic policies

Myopic or reactive policies ground their decisions exclusively on the current model's information, so no predicted knowledge or direct representations of future decisions are considered. Therefore, only the immediate costs are evaluated. However, adjustable parameters might be introduced to improve the overall performance.

A myopic policy can be represented as:

$$A^{RO}(S_t) = \underset{a_t \in \mathcal{A}(S_t)}{\arg\min}\, C(S_t, a_t), \tag{2.20}$$

where S_t is the current state, $\mathcal{A}(S_t)$ represents the feasible decision space, and $C(S_t, a_t)$ indicates the immediate inventory and routing costs of taking decision a_t in state S_t.

The design of a myopic policy may follow an integrated or a decoupled approach. Integrated single-period policies, often formulated as mixed-integer programs, jointly optimize

inventory and routing decisions for the current period. In contrast, decoupled policies decompose the problem by solving inventory and routing subproblems separately. In both settings, selecting which customers to visit is at the intersection of inventory and routing. For example, classic inventory replenishment rules, based on reorder points, may guide this customer selection and, consequently, condition the vehicle routes. Ultimately, these routes can remain consistent throughout the planning horizon (e.g., Cui et al., 2023).

The following inventory replenishment policies can be applied reactively in an RH fashion, with vehicle routes subsequently determined by solving a VRP instance (Roldán et al., 2016):

i. Fixed quantity policy: Customer i is replenished with a specified amount of product every period t. This fixed quantity may be defined as a fraction of each customer's capacity $\bar{U}_i$.

ii. OU policy: Customer i is refilled to its inventory capacity $\bar{U}_i$ whenever visited.

iii. Zero-step lookahead policy: The decision-maker relies on a single-point forecast of the current period's demand, without assessing a lookahead model for the following periods. The replenished quantity might correspond to the predicted demand.

iv. (s_i, S_i) policy: Every time the inventory level of customer i falls below a reorder point s_i, it is refilled to a predetermined inventory level S_i. Typically, S_i is set as the inventory capacity $\bar{U}_i$, or a fraction thereof. Several strategies may define the reorder point s_i as either a fraction of S_i, or as the average demand over the last H historical periods plus a predefined safety stock.

The first studies on the SDIRP relied on RH reoptimization approaches, although some recent works have also adopted them. Dror and Ball (1987) minimize the annual delivery costs by solving twelve dependent successive weeks in RH. Customers are first assigned to delivery days at the beginning of each week, resorting to a linear programming-based generalized assignment algorithm. A VRP is then solved for each day using a modified Clarke and Wright algorithm. The original solution is subsequently improved through inter-route and inter-day customer exchanges. Trudeau and Dror (1992) adopted a similar OU policy, accounting for the possibility of route failures when the actual demand exceeds the available vehicle capacity.

Alternatively, Reiman et al. (1999) model a dynamic stochastic distribution system as a queuing control problem and solve it using heavy traffic approximations to anticipate future states and optimize decisions. The control concerns two decisions: whether the vehicle should be busy or idle, and how to distribute the load among retailers during a tour. Three strategies are evaluated: the first based on direct shipping, the second based on a pre-specified tour, and the third based on the dynamic choice between direct shipping and the traveling salesman problem (TSP).

In Li et al. (2016), replenishment lead-times are introduced, and a myopic policy aims to minimize the average costs between two consecutive replenishments. A decision stage may contain multiple periods, and it works as a replenishment cycle so that decisions are made in the first period with no replenishment in subsequent periods. In this reactive RH

approach, the authors solve two problems: determining the optimal order-up-to-level (S) for a given stage length and optimizing the number of periods in a stage.

Based on Giesen et al. (2005), Giesen et al. (2010) suggest two different strategies: optimization-based and fixed-tour-based. Concerning the optimization-based strategies, the inventory control component is resolved beforehand, and its results are used as target levels when updating plans. The authors model the problem as an offline deterministic optimization one and solve it statically (i.e., without foresight into future conditions) in RH using a commercial solver. Regarding the fixed-tour strategies, the routing component is determined in advance, and the resulting routes are implemented with recourse actions depending on real-time information.

Achamrah et al. (2022) develop a matheuristic based on the hybridization of mathematical modeling, the global search capacity of genetic algorithms and the self-adaptability of deep reinforcement learning. Deep Q-learning helps extract valuable knowledge to improve the search performance and speed of the proposed metaheuristic. Initially, they propose a routing model to be solved during each period using the genetic algorithm. The solution obtained from this model serves as input for a second one called transshipment and substitution model, which is solved using a commercial solver with default parameters.

The following works handle inventory and routing decisions sequentially, both employing (s,S) inventory policies. Jarugumilli and Grasman (2007) provide a methodology for dynamic control of a radio-frequency identification-enabled IRP, comprising two phases. The first involves determining inventory control parameters for each wholesaler location using an (R,s,S) policy. The second focuses on dynamically calculating the delivery quantities and planning the routing for a single vehicle. Furthermore, Roldán et al. (2016) extend the reactive policies of Coelho et al. (2014a), proposing three different customer selection methods and two inventory replenishment policies: fixed quantity and (s,S) for different values of s and S. Finally, vehicle routes are computed by solving a TSP.

Additionally, we identify methodologies in which vehicle routes are generated in advance, without considering uncertainty, reducing the solution space of admissible routes and improving the solution time for large-scale instances (e.g., Li et al., 2023). For example, Cui et al. (2023) adopt a distributional robust optimization approach from the service-level perspective, where transportation costs in a given period might not exceed a pre-specified budget. In the first stage, the visiting and routing decisions must be made for all periods at the horizon's beginning. Secondly, replenishment decisions are made at the start of each period after the previous period's demands have been revealed using the Benders decomposition method.

2.5.2 Policy search

Policy search consists of searching across a family of functions to identify the one that works better over time according to a given objective function (2.19) by simulating their performance. The searching process can be performed over several classes of policies, i.e., PFA, or a set of parameters that define a given policy, i.e., CFA (Powell, 2022b). Both tend to follow straightforward structures that effectively account for uncertainty, typically without the need to solve demanding optimization problems, although requiring tuning.

2.5.2.1 Policy function approximation

Although stochastic dynamic environments might lead to complex model representations, pre-analytical insights may be evident and pre-experienced. Consequently, we might be able to design a policy that solves our problem based on these considerations. PFA is a method that directly links a state variable to a decision without necessarily considering a direct lookahead model or exclusively focusing on immediate rewards. It can be expressed by (1) lookup tables: given the current state, the policy returns a discrete action directly from a predefined action-state matrix; (2) parametric representations: an analytical function is designed upon prior knowledge. Tunable parameters are obtained through a simulation-based optimization process; (3) non-parametric representations: deep neural networks might be designed to reveal the action we should take when in a given state.

Formally, a PFA policy can be expressed as:

$$A^{PFA}(S_t) = f(\phi(S_t)), \tag{2.21}$$

where S_t defines the current state, ϕ represents a set of problem features, and f is a parametric or non-parametric function. We illustrate two examples of PFAs in this section, with further hybrid policies discussed in Section 2.5.5 (e.g., Raba et al., 2020).

Inventory control rules. PFAs are particularly common in decoupled approaches, where intuitive inventory replenishment rules are defined first, and routing is subsequently solved using exact or approximate methods. The inherent complexity of the VRP often constrains the applicability of PFA methods for directly computing routes. Regarding inventory control, let us consider the following order-up-to-level (s, S) rule, where a decision corresponds to the quantity delivered to customer i at period t (q_{it}):

$$A^{PFA}(S_t \mid \theta^{low-high}) = \begin{cases} \theta^{high} - \phi_i(S_t), & \text{if } \phi_i(S_t) \leq \theta^{low}, \\ 0, & \text{otherwise}, \end{cases} \tag{2.22}$$

where $\phi_i(S_t)$ represents the current inventory level of a product at customer i (I_{it}), and θ^{low} and θ^{high} stand for a reorder point (s) and order-up-to level (S), respectively. Parameters θ^{low} and θ^{high} can be customer-dependent and tuned via simulation-optimization or defined a priori. Once delivery quantities are set, vehicle routes are obtained by solving a VRP instance.

Delivery priority rules. PFAs can also be used to determine which customers to replenish in a given period. For instance, the following parametric rule computes a visiting decision for each customer i (y_{it}):

$$A^{PFA}(S_t \mid \theta) = \begin{cases} 1, & \text{if } \theta_0 + \theta_1\phi_1(S_t) + \cdots + \theta_F\phi_F(S_t) > 0, \\ 0, & \text{otherwise}, \end{cases} \tag{2.23}$$

where $\phi = (\phi_1, ..., \phi_F)$ represents a set of problem features (e.g., current inventory, expected demand, holding and stockout costs, see Section 2.3.2.1) and $\theta = (\theta_0, ..., \theta_F)$ a set

of tunable weights. This priority rule can be subsequently integrated into a three-step policy. First, consider the visiting decisions. Second, determine the replenishment quantities, which may be guided by an OU inventory rule. Finally, obtain the vehicle routes by solving a VRP instance.

The following studies implement this policy class. Vonolfen et al. (2013) propose a simulation-based optimization approach to learn a delivery policy that includes two rules: selecting the customers to visit and the number of products to deliver. The learning strategy implies optimizing eleven parameters for several scenarios using a genetic algorithm through a continuous evolutionary process. Subsequently, the routing problem is solved using a push-forward insertion heuristic. Moreover, Mes et al. (2014) present a fast and parameterized heuristic responsible for scheduling the emptiness of underground waste containers and creating daily vehicle routes. This heuristic comprises a set of tunable parameters optimized through two offline learning techniques to face uncertainty, depending on the day of the week. The cheapest insertion method is then employed to establish the routes.

2.5.2.2 Cost function approximation

This is a class of policies that result from a parametric modification of a cost function, contributing to the flexibility and robustness of myopic policies. Conceptually, CFAs are parameterized optimization models in which a set of parameters θ is introduced to adjust the objective function or constraints, either avoiding the need to solve a lookahead model (e.g., Greif et al., 2024) or enhancing a lookahead model to better balance immediate and future costs (e.g., Cuellar-Usaquén et al., 2024). Despite its potential, and similarly to PFA, CFA remains largely underexplored in the context of the SDIRP.

A CFA policy can be defined as follows:

$$A^{CFA}(S_t \mid \theta) = \underset{a_t \in \mathcal{A}(S_t,\theta)}{\arg\min}\ \bar{C}(S_t, a_t \mid \theta), \tag{2.24}$$

where a_t defines a decision at S_t, $\bar{C}(S_t, a_t \mid \theta)$ is a parametrically modified objective function, and $\mathcal{A}(S_t, \theta)$ represents the feasible decision space defined by parametrically modified constraints. Two examples are provided in the following, the first addressing a parametrically modified objective function, and the second a set of parametrically modified constraints. For the sake of simplicity, both examples refer to a single-item, single-vehicle problem variant. Additional CFA-based works incorporating other policy classes are discussed in Section 2.5.5.

Parametrically modified objective function. A multi-period formulation, comprising holding, stockout, and routing costs, can be decomposed into single-period subproblems, becoming less complex to solve. Following an OU inventory replenishment strategy, a prize vector $\theta = (\theta_i)_{\ i \in \mathcal{N}'}$ can be introduced to reflect the priority of visiting each customer i on a given period t. The original objective function (2.17) can then be modified to balance reward collection with transportation costs. Accordingly, inventory holding and stockout costs are implicitly captured through the parametric vector θ, while the optimization model

explicitly accounts for routing costs and prioritized customer visits. The resulting parametric CFA policy can be expressed as:

$$A^{CFA}(S_t \mid \theta) = \underset{a_t \in \mathcal{A}(S_t)}{\arg\max} \sum_{i \in \mathcal{N}'} \theta_i \, y_{it} - \sum_{(i,j) \in \mathcal{E}} c_{ij} \, x_{ijt} \tag{2.25}$$

Once the visiting and routing decisions are determined, the resulting inventory holding, stockout, and routing costs are calculated to assess the overall policy performance. For instance, Greif et al. (2024) propose a learning-based policy for the SDIRP, which combines a machine learning layer and a combinatorial optimization layer. The machine learning layer predicts the prize vector θ using a physics-informed neural network with current inventories and historical observations as input. The combinatorial optimization layer then translates into a single-period capacitated prize-collecting TSP, embedding the predicted prizes into the objective function. The resulting policy achieves lower inference time and benefits from contextual information.

Parametrically modified constraints. Let us consider a direct delivery problem where a vector of calibrated parameters $\theta = (\theta^{demand}, \theta^{min}, \theta^{capacity})$ is introduced to account for demand variability, minimum delivery requirements, and capacity restrictions. Moreover, $\hat{D}_{it}$ denotes the expected demand of customer i in period t, and c_{0i} the distance supplier-customer. The resulting single-period optimization problem, intended to be resolved in RH, is:

$$\min \sum_{i \in \mathcal{N}'} c_{0i} \, q_{it}, \tag{2.26}$$

subject to:

$$q_{it} \geq \hat{D}_{it} + \theta^{demand}, \quad \forall i \in \mathcal{N}', t \in \mathcal{T} \tag{2.27}$$

$$q_{it} \leq \bar{U}_i \, \theta^{capacity}, \quad \forall i \in \mathcal{N}', t \in \mathcal{T} \tag{2.28}$$

$$q_{it} \geq \theta^{min}, \quad \forall i \in \mathcal{N}', t \in \mathcal{T} \tag{2.29}$$

Accordingly, the parametric CFA policy is as follows:

$$A^{CFA}(S_t \mid \theta) = \underset{a_t \in \mathcal{A}(S_t, \theta)}{\arg\min} \sum_{i \in \mathcal{N}'} c_{0i} \, q_{it} \tag{2.30}$$

2.5.3 Direct lookahead approximation

Direct lookahead policies are formulated as finite-horizon mathematical programming models, commonly employed in an RH fashion, where a lookahead model is reoptimized whenever new information realizations become available. DLAs can be computationally demanding, as they rely on solving simplified representations of future states and decisions (Powell, 2022b). Consequently, they are typically used as a fallback option when other policy classes fail to provide satisfactory results. These policies promote integrated approaches, where inventory and routing decisions are optimized together over a truncated

horizon of l periods, providing a suitable structure to represent both operational constraints and future uncertainty. Given a state S_t, we aim to determine a decision $a_t = A(S_t)$ that minimizes not only the immediate inventory and routing costs $C(S_t, a_t)$, but also the expected cost of future decisions over a lookahead horizon $L = \min(t+l, T)$, so that:

$$A^{DLA}(S_t) = \underset{a_t \in \mathcal{A}(S_t)}{\arg\min} \left(C(S_t, a_t) + \mathbb{E}\left\{ \min_{\pi} \mathbb{E}\left\{ \sum_{t'=t+1}^{L} C(S_{t'}, A^{\pi}(S_{t'})) \mid S_{t+1} \right\} \mid (S_t, a_t) \right\} \right) \quad (2.31)$$

The primary challenge lies in designing and computing an approximation of future states and decisions using a lookahead model. This lookahead model is typically formulated as a mixed-integer optimization problem, incorporating all relevant constraints within the feasible decision space $\mathcal{A}(S_t)$, as defined in Section 2.3.2.1. The choice between deterministic and stochastic lookaheads depends on the information nature and availability: when accurate forecasts, i.e., single expected values, are available, deterministic lookaheads are preferred (refer to Section 2.5.3.1), whereas full distributional knowledge suggests stochastic lookaheads (refer to Section 2.5.3.2), leveraging stochastic programming techniques. Furthermore, it is important to remember that longer planning periods do not necessarily lead to better solutions, as the state and decision spaces grow substantially, turning the problem computationally more demanding (Coelho et al., 2014a).

2.5.3.1 Rolling horizon with deterministic lookaheads

Deterministic lookahead models remain a common strategy to approximate the future states of a stochastic problem. However, by relying on deterministic approximations, we are neglecting the uncertainty in each state transition. Therefore, the current decision may not adequately prepare us for random events in the future. Forecasting techniques are often used to predict demand in future states, converting the stochastic optimization problem into a deterministic one, in which:

$$A^{DLA_a}(S_t) = \underset{a_t, a_{t+1}, \ldots, a_L}{\arg\min} \left(C(S_t, a_t) + \sum_{t'=t+1}^{L} C(S_{t'}, a_{t'}) \right) \quad (2.32)$$

The structure of the deterministic lookahead model is similar to the decision space defined in Section 2.3.2.2. Accordingly, for each customer, we must consider an expected demand for the current period t, and for each lookahead period $t' \in \{t+1, \ldots, L\}$. For instance, Coelho et al. (2014a) present a proactive approach that employs a forecasting model based on the exponential smoothing technique and resorts to an ALNS matheuristic to solve the problem. Moreover, after demands are revealed, the authors assess the impact of considering lateral emergency transshipment as a recourse action for stockouts. Neves-Moreira et al. (2022) develop an exact branch-and-cut algorithm with patching heuristics to tackle the multi-product IRP with simultaneous P&D. At the beginning of each period, the exact demands of the current period are known. However, the authors consider a forecast model to include the demands of future periods in the current decision.

Referring to Markov et al. (2015) and Markov et al. (2018), Markov et al. (2020) model a recyclable waste collection problem as a non-linear mixed-integer program and solve it by an ALNS algorithm incorporating a forecasting model tested and validated on real data. Demand forecasts are updated at each rollover, with revealed demands reducing uncertainty. Moreover, Elbek and Wøhlk (2016) devise a variable neighborhood search algorithm to solve daily planning problems with uncertain accretion rates. Each daily planning considers the current day and a short planning period of eleven days, benefiting from including a future cost function that estimates the long-term effects of short-term decisions. The solution is then reoptimized every subsequent day by improving the collection in the planning period.

Cubillos et al. (2022) propose a variable neighborhood search algorithm responsible for planning what customers should be visited and how to combine them into a single-vehicle route for the following planning periods according to an OU policy. Demand forecasts are computed using the double smoothing method, based on demand data from the previous 100 periods. The experimental setup includes a 200-period planning horizon and rolling planning intervals of seven periods.

More recently, Chiu et al. (2024) first propose three modifications to the traditional IRP, such as defining minimum inventory levels for the last period, applying an artificial discount rate in the objective function and introducing safety stock constraints. Following this, a new policy consisting of an optimization model that approximates the cost of future periods using linear variables instead of integer variables (IRP with relaxed periods) is introduced. A simpler heuristic and an RH are evaluated using a two-stage stochastic model. A standard branch-and-cut is used to solve the models.

2.5.3.2 Rolling horizon with stochastic lookaheads

A two-stage stochastic programming model might be employed to approximate a fully sequential (i.e., multi-stage) problem in an RH environment, resulting in the following lookahead policy (Powell, 2022b):

$$A^{DLA_b}(S_t) = \underset{a_t, a_{t+1}(\omega), \ldots, a_L(\omega)}{\arg\min} C(S_t, a_t) + \sum_{\omega \in \Omega_t} P(\omega) \sum_{t'=t+1}^{L} C(S_{t'}(\omega), a_{t'}(\omega)) \tag{2.33}$$

This formulation distinguishes three elements: i) current ("here-and-now") decision a_t when in state S_t; ii) scenario-dependent future realizations $\omega \in \Omega_t$, where ω represents a sequence of potential realizations for the current and future periods $t' \in \{t+1, \ldots, L\}$. The finite set of scenarios Ω_t is typically generated via Monte Carlo simulation, with each scenario assigned probability $P(\omega) = 1/|\Omega_t|$; iii) future (recourse) decisions $a_{t+1}(\omega), \ldots, a_L(\omega)$. The stochastic lookahead model's structure resembles the decision space introduced in Section 2.3.2.2. However, it models several demand scenarios for each period of the lookahead horizon. The proposed lookahead policy can then be solved using deterministic methods, including exact algorithms, metaheuristics, or even matheuristics.

Relevant literature has explored the use of this policy class. For instance, Jaillet et al. (2002) adopt the RH framework proposed by Bard et al. (1998). The authors reduce an annual horizon to a biweekly RH period using incremental cost approximations, in which only the first week's decisions are implemented. The assignment of customers to days of the planning period depends on the optimal delivery frequency and the incremental costs incurred by not following the optimal replenishment day. After assigning customers to days, a set of daily VRPs results. Moreover, special deliveries are made in case of a stockout, and penalty costs are incurred.

Hvattum et al. (2009) exploit finite scenario trees as lookahead models to reduce the infinite MDP and approximate the expected future costs. A finite scenario tree is composed of a set of possible demand realizations. The authors devise a greedy randomized adaptive search procedure to solve a scenario tree at each state (e.g., day), representing customers' inventory levels. Subsequently, Hvattum and Løkketangen (2009) recur to the progressive hedging algorithm to solve each scenario tree generated at each epoch based on the current inventory levels of each customer and demand distributions.

Nolz et al. (2014) model an infectious medical waste collection problem as a two-stage stochastic optimization problem. The authors propose a sampling method and an ALNS algorithm, and compare both policies. Results show that the ALNS performs very well and thus might be applied to generate good approximations of the optimal solution for real-world instances. After selecting the ALNS, they evaluate three different collection policies under five real-world instances and investigate a radio-frequency identification scenario within an RH framework.

Furthermore, Mirzapour Al-e hashem et al. (2019) propose a bi-objective two-stage stochastic programming model with transshipments between suppliers, considering a two-period planning horizon. Second-stage decisions are part of the optimization process but are not immediately applied. Instead, they inform the current optimization by estimating future costs under uncertainty. Given that a standard linear programming technique is impractical, the authors propose a hybrid L-shaped method that combines the L-shaped method (i.e., a decomposition approach for stochastic optimization) and compromise programming, a well-known approach for multi-objective optimization.

In a different light, Cavagnini et al. (2022) solve a multi-stage stochastic fixed-charge transportation problem with transshipments as a recourse action, resorting to a commercial solver. The first-stage decision comprises the delivery plan using direct shipping and an OU policy, while transshipment and backorders are the recourse decisions. The authors use scenario trees to model the uncertain demands. Moreover, Violi et al. (2020) adopt a multi-stage stochastic linear program, where the uncertain future demand is represented by a scenario tree that is updated daily. The model is formulated and solved using a commercial solver for a planning period comprising the upcoming day to the end of the planning horizon. Although there is a decision variable in the mathematical model that determines whether a given route is assigned to a given node of the scenario tree, these routes are pre-generated.

Finally, Brinkmann et al. (2020) address a bike-sharing system, extending Brinkmann et al. (2019)'s dynamic lookahead policy to include a fleet of multiple vehicles. Demand follows a regular pattern of bike usage throughout the day, caused by commuters, shopping,

and leisure activities, but displays considerable short-term variability. Bikes are dynamically reallocated between stations, given the newly revealed information during travels, to minimize the occurrence of failed demands. The proposed dynamic lookahead policy anticipates future demands based on historical observations, shifting the anticipation process of Brinkmann et al. (2019) to an analytical offline procedure to reduce computational burden.

2.5.3.3 Rollout algorithm

Rollout policies rely on approximating the stochastic lookahead model with a straightforward, easily computable function, also known as base policy π_b (Powell, 2022b). Such policies are typically employed when enumerating the complete set of possible decisions $a_t \in \mathcal{A}(S_t)$ in state S_t is impractical. From a previously generated set of candidate decisions and information realization sequences, the rollout policy allows us to individually apply each candidate decision a_t to the current state S_t, designing new post-decision states S_t^a and later computing an estimate of their values $\hat{V}(S_t^a)$, i.e., $J^{\pi_b}(S_t^a)$. To balance computational feasibility with decision quality, each simulation (i.e., sequence of realizations) includes a reasonable but limited number of future periods.

A rollout policy can be expressed as follows, where $J^{\pi}(S_t^a)$ is the cost-to-go of base policy π_b:

$$A^{DLA_c}(S_t) = \arg\min_{a_t} \left[C(S_t, a_t) + J^{\pi_b}(S_t^a)\right] \tag{2.34}$$

From the initial post-decision state S_t^a, the base policy maps a state into an action, originating new transitions and consecutive post-decision states. Thus, $J^{\pi_b}(S_t^a)$ is updated according to the rewards, i.e., (-) costs, obtained in each transition. After a predefined number of simulations, we get the estimated value of the original post-decision state S_t^a, and the process is repeated for the next candidate decision. After the whole set of decisions has been evaluated, the Bellman equation is applied to select the final decision. We can also include a discount factor applied to the state-value function to prioritize future periods to the detriment of immediate rewards.

The following works illustrate the application of this policy. Bertazzi et al. (2013) introduce a hybrid rollout algorithm to perform periodic replannings. The authors use a sampling approach to generate stationary demand scenarios for the current period, considering the average demands for future periods. A base policy approximates the cost-to-go function obtained by exactly solving a mixed-integer linear programming model using a branch-and-cut algorithm. Subsequently, Bertazzi et al. (2015) employ a matheuristic combining the rollout algorithm with an optimal solution of mixed-integer linear programming models for realistic-size instances, as in Bertazzi et al. (2013). This one-step lookahead policy approximates the value of the optimal cost-to-go and the approximate set of controls by solving a mixed-integer linear programming model.

2.5.4 Value function approximation

VFA is a powerful technique often used in reinforcement learning to approximate value functions that solve complex stochastic dynamic programming (SDP) problems, usually modeled MDPs. As a value-based method, these policies aim to approximate the value of being in a given state and tend to perform well when the downstream value of that state is straightforward to estimate (Powell, 2007). We replace the lookahead model of direct lookahead policies in Equation (2.31) with a value function $V(S_{t+1})$, following Bellman's optimality principle. Since computing its exact value is often intractable, we tend to approximate $V(S_{t+1})$ as in Equation (2.35), where $S_{t+1} = \Gamma(S_t^a, \omega_{t+1})$. Alternatively, the value function may be approximated around the post-decision state $S_t^a = (S_t, a_t)$, as shown in Equation (2.36) (Powell, 2019).

$$V(S_t) = \min_{a_t} \Big(C(S_t, a_t) + \mathbb{E}\{V(S_{t+1}) | (S_t, a_t)\}) \Big) \tag{2.35}$$

$$V(S_t) = \min_{a_t} \Big(C(S_t, a_t) + V(S_t^a) \Big) \rightarrow A^{VFA}(S_t) = \underset{a_t \in \mathcal{A}(S_t)}{\arg\min} \Big(C(S_t, a_t) + V(S_t^a) \Big) \tag{2.36}$$

VFA policies are well-suited for infinite-horizon problems, where decisions must account for long-term impacts. The learning process is based on a set of simulations in which the resulting observations (i.e., state, action, post-state, reward) are stored to approximate the value of each state at the end of each simulation. Thus, each decision a_t directly depends on the current state S_t and observations collected during the learning stage. As soon as the simulations are performed and the state-value functions are learned, we can dynamically apply the obtained policy at each decision state without requiring additional optimization.

We proceed by highlighting several studies that adopt this policy class. Minkoff (1993) propose a decomposition heuristic to reduce the original problem's complexity to a set of subproblems that are easier to solve and computationally less expensive. Firstly, a transportation cost allocation problem is solved as a linear programming problem. Secondly, a set of delivery quantity subproblems is solved, one for each customer. Thirdly, an itinerary selection problem is constructed and solved at each decision point to obtain the itineraries to dispatch, devising an optimal branch-and-bound algorithm. The value functions of the individual subproblems are added to approximate the value function of the combined problem.

Berman and Larson (2001) aim to minimize the expected costs regarding early and late deliveries, shortfalls, and non-empty returns to the depot over an infinite horizon. The authors propose four dynamic programming formulations that differ in state variables and the possibility of dumping excess product when returning to the depot. Both programs are solved without using approximations. The routing problem is not considered here, but instead a prespecified sequence of customers is provided. Therefore, every time the vehicle arrives at a customer, the delivery quantities of a single product are computed.

Kleywegt et al. (2004) extend the formulation and solution approach of Kleywegt et al. (2002) to include multiple routes limited to at most three customers per route. The approximate value function is obtained by decomposing the overall problem into two-customer subproblems. The problem of maximizing the expected discounted value (i.e., sales revenues, production costs, transportation costs, inventory holding costs, and shortage penal-

ties) over an infinite planning horizon is addressed to determine the approximate optimal decision for each state. A three-step local search heuristic is developed for instances with more than four customers and two vehicles.

Instead of using a simulation-based method as Kleywegt et al. (2004), Adelman (2004) propose a price-directed approach that uses optimal dual prices from two linear programming relaxations to find a VFA for each customer that approximates the future costs of the current decisions over an infinite planning horizon. The price-directed policy is compared to three benchmark policies: the direct shipment policy by Kleywegt et al. (2002), the myopic policy described in Kleywegt et al. (2004), and the procedure outlined by Minkoff (1993).

Crama et al. (2018) develop four solution methods, where the last two rely on an SDP model. The authors start with an expected value method to reduce the complexity of the problem. Secondly, they replace the first step of the previous algorithm with an OU replenishment policy by introducing a predefined target service level. Thirdly, they propose a decomposition method based on an independent SDP model for each store. Routing costs are considered as a fixed cost-to-serve. Finally, the decomposition-integration method is introduced as an optimization problem to simultaneously determine the delivery quantities and routes. A matheuristic is proposed to solve the optimization model and approximate the optimal expected total profit.

McKenna et al. (2020) propose an approximate policy iteration algorithm based on VFA that uses least-squares temporal difference learning for policy evaluation. The inventory status of each combat outpost, along with the number of cargo unmanned aerial vehicles and the threat index, defines the state space. Every six hours, they must decide how many fully loaded vehicles to dispatch to each combat outpost. Experiments show that the performance of the resulting policy significantly decreases as the number of combat outposts increases, although it remains robust against changes in other problem features.

Recently, Liang et al. (2024) introduce a dual policy reinforcement learning algorithm that decouples inventory and routing decisions to solve a dynamic bike repositioning problem. Each decision point occurs when a vehicle arrives at or departs from a station after completing rebalancing operations. Two deep Q-networks, i.e., neural networks that approximate the action-value function $Q(s,a)$, are conceived to train the inventory and routing decision networks jointly. The output layer is designed to align with the action space of each network, enabling the network to compute Q-values, i.e., expected cumulative rewards, for all potential actions relative to a given state.

2.5.5 Hybrid policies

Hybrid policies emerge to enhance and reinforce the different policy classes applied individually. For instance, DLA policies could benefit from CFA to leverage their predictive power and the computational efficiency of cost approximations, or alternatively, be parameterized via VFA. On the other hand, PFA can take advantage of lookahead considerations to improve the adaptability of existing parameterized functions. The following studies reveal how different policy classes can be hybridized.

Brinkmann et al. (2019) present a dynamic lookahead policy that anticipates potential future demands in current inventory decisions. Due to the variation in daily demand patterns, lookahead horizons become time-dependent and challenging to define. Accordingly, the authors use a non-parametric policy search method to find proper lookahead horizons using VFA. From a set of horizons, this method allows a state-dependent policy selection according to a set of state parameters. Three potential percentages leading to three different target fill levels are considered during policy search. After that, the lookahead determines the next station to visit using online simulation.

Moreover, Raba et al. (2020) propose a reactive simheuristic combining a biased-randomized algorithm with Monte Carlo simulation using online data. A static and two dynamic approaches are introduced. The non-reactive (static) simheuristic approach optimizes the whole set of periods at the same time, at the beginning of the planning horizon, following Onggo et al. (2019) and Gruler et al. (2018). Additionally, the authors devise a reactive (dynamic) simheuristic approach that reoptimizes the initial previous solution in each rolled period as real demands are revealed. However, the same refill policy is applied to all silos. Lastly, the authors introduce the same reactive simheuristic approach, but based on the expected demands, so that a different policy can be selected for each silo.

Cuellar-Usaquén et al. (2024) unveil a dynamic policy incorporating an adaptive CFA within a two-stage stochastic program to obtain the inventory, purchasing and routing decisions for the current period. Besides customer demands, suppliers' purchasing prices and supply volumes are also uncertain and periodically revealed. CFA reflects the current and future routing costs within the stochastic lookahead, maintaining the solution quality within reasonable computation times. Subsequently, a conventional routing heuristic is employed to compute the routing decisions every period. The authors compare their policy with eleven benchmarks, including myopic, PFA, and expected value policies.

In this light, Jia et al. (2025) propose a scenario predict-then-optimize strategy, a fully data-driven online IRP approach involving two consecutive steps at each real-time decision point. This approach exploits multi-horizon quantile recurrent neural networks to predict future demand quantiles and further design a scenario sampling approach. The next scenario-optimize step solves a scenario-based two-stage stochastic programming approximation model.

Ortega et al. (2024) address the stochastic IRP with dynamic demands and intra-day depletion. The authors formulate the problem as a finite-horizon SDP and devise an iterative lookahead algorithm with an ALNS for the routing and policy learning for the replenishment decisions. Initially, the authors plan the current period and the following ones that comprise the lookahead horizon using a pre-defined replenishment policy. Vehicle routes are calculated via an ALNS-based metaheuristic or cheapest insertion. Secondly, the previous routes are used as input for an iterative improvement of the replenishment policies, i.e., policy learning. Thirdly, new routes are computed.

Finally, Hasturk et al. (2024) devise a parameterized CFA-based mixed-integer program to handle static, tactical-level and routing decisions and an MDP for dynamic, operational-level purchasing decisions. The producer can buy hydrogen to meet customer demands and sell hydrogen for an extra profit. At each period, the producer must ensure enough hydrogen to satisfy customer demands as imposed by the transportation delivery

schedules. First, the current inventory level and customer demands for that period are observed. Based on this information, hydrogen is either purchased from or sold to the market to address under or overproduction. Afterwards, deliveries to customers are performed. Both static and dynamic decisions are optimized together to achieve long-term global optimal solutions. The outcome of the tactical decisions determines the properties of the MDP underlying the operational-level dynamic purchasing decisions.

2.5.6 Learning and simulation in policy design

The interest in dynamic problems is rapidly growing, with machine learning algorithms, including supervised and reinforcement learning, revealing promising outcomes when dealing with complex combinatorial optimization problems with large state and action spaces. This section explores the integration of learning mechanisms within policy design. This includes inventory and routing decisions, problem parameters **(P)** that support decision-making, and hyperparameters **(H)** intrinsic to the optimization method (refer to Table 2.7). Regarding problem decisions, we identify whether any of them, i.e., select the customers to visit **(S)**, determine the delivery or purchase quantities **(Q)**, and combine customers into vehicle routes **(R)**, are learned directly or indirectly (e.g., through a set of tuned parameters).

Table 2.7: Learning domains and subdomains taxonomy.

Learning domain	Learning subdomain	Code
Problem decisions	Select the customers to visit	S
	Determine the delivery (or purchase) quantities	Q
	Combine customers into vehicle routes	R
Problem parameters	e.g., Prize values, customer demand	P
Methods' hyperparameters	e.g., Operators weights, exploration level	H

Tabular methods (e.g., Q-learning) offer a straightforward and exact representation of state-value or action-value functions. Nevertheless, they become impractical in large-scale and continuous scenarios, as the number of states and actions increases exponentially, making it infeasible to store or update a table. Approximate solution methods are generally more suitable for solving SDIRPs due to their efficiency and ability to generalize across a wide range of scenarios. However, these methods also present challenges like complexity, stability issues, and approximation errors. In this light, machine learning algorithms can be employed to enhance: i) VFA methods, by estimating the value functions; ii) PFA methods, by refining analytical functions that map states to actions with tunable parameters; iii) CFA methods, by improving the estimation of cost functions.

Policy and value iteration are two reinforcement learning methods for computing an optimal or near-optimal policy by minimizing the expected long-term discounted costs. The learning procedure can be conducted offline or online (Brinkmann et al., 2019). However, due to environmental unpredictability, the best input parameters may only apply to specific periods within the planning horizon. Therefore, reoptimization may be necessary whenever significant environmental changes occur, e.g., the start of a holiday or changing weather

conditions (Mes et al., 2014). For additional insights, we refer to Mazyavkina et al. (2021), who present a survey on reinforcement learning for a set of combinatorial optimization problems, including the TSP and bin packing problems, highlighting a promising research direction. Within the scope of stochastic dynamic vehicle routing problems (SDVRPs), we highlight the contributions of Hildebrandt et al. (2023).

On the other hand, supervised learning focuses on training a model that accurately predicts target values based on input features (Bengio et al., 2021). For example, Greif et al. (2024) take advantage of CFA properties by adopting a combinatorial optimization-machine learning pipeline. The machine learning layer consists of a physics-informed neural network that outputs a set of delivery prizes based on the current customer inventories and other contextual information. Jia et al. (2025) introduce a fully data-driven approach based on a scenario predict-then-optimize strategy. This method employs multi-horizon recurrent neural networks to predict future customer demand quantiles.

2.5.6.1 Problem decisions

Kleywegt et al. (2002) suggest introducing a set of parameters β to improve the approximation of the proposed value function. The best combination of these parameters is achieved through stochastic approximation and Bellman error methods, resulting in parametric VFAs: $\hat{V}(S_t,\beta)$. A feasible decision involves dispatching up to a given number of vehicle trips to a customer, along with the corresponding delivery quantities (**S, Q**). As an extension of the previous one, the approximate value function of Kleywegt et al. (2004) is obtained by decomposing the original problem into subproblems. Therefore, $\hat{V}(S_t)$ is set as the sum of the optimal value functions of each state subproblem. Each decision contains information about **(S, Q, R)**.

Furthermore, McKenna et al. (2020) employ approximate dynamic programming (ADP) methods via policy iteration, referring to the least squares temporal difference learning for policy evaluation. This approach helps determine the number of fully loaded vehicles dispatched to each combat outpost **(S, Q)**. In Liang et al. (2024), two deep Q-networks (i.e., an off-policy temporal-difference reinforcement learning technique) are simultaneously trained for inventory and routing decisions, aiming to approximate the Q-value function **(S, Q, R)**. A heuristic policy concerning the routing subproblem is initially employed to accelerate the training process.

Vonolfen et al. (2013) introduce a delivery rule composed of two priority rules **(S, Q)** consisting of eleven parameters tuned for different demand scenarios using a genetic algorithm through a simulation-based optimization approach. Further regarding PFA, the parameterized heuristic of Mes et al. (2014), responsible for scheduling the emptying of underground waste containers **(S)**, is composed of a set of tunable parameters optimized via two different offline learning techniques: the Sequential Kriging Optimization from Huang et al. (2006) and the Hierarchical Knowledge Gradient from Mes et al. (2011).

In Brinkmann et al. (2019), the algorithm starts from three potential fill levels, i.e., low (25%), medium (50%), and high (75%). It iteratively simulates future realizations over a lookahead horizon (offline learning), approximating the impact of the inventory decisions **(Q)**. Moreover, Raba et al. (2020) present two reactive strategies combining a

biased-randomized algorithm with Monte Carlo simulation to find the best refill policy for each retail centre in a given period **(S, Q)**. Finally, Ortega et al. (2024) introduce an iterative procedure built on the concept of an adaptive lookahead algorithm. The authors propose a customer decomposition-PFA method to derive a replenishment policy per customer **(S, Q)**, considering an estimate of the expected routing costs.

2.5.6.2 Problem parameters

Our attention is now redirected to works that, rather than directly learning a specific decision, seek to find optimal problem parameters **(P)** that enhance decision-making by refining either the decision policy or the lookahead model (e.g., Jia et al., 2025; Greif et al., 2024). For instance, Cuellar-Usaquén et al. (2024) reduce the routing decisions in the two-stage stochastic programming model using an approximate routing cost achieved by repeated training simulations (i.e., CFA). This strategy helps maintain the quality of the solution in reasonable computation times. Inventory and purchasing decisions are then guided by the lookahead policy, with the final routing decision thereafter determined by a conventional heuristic. Instead, Hasturk et al. (2024)'s approach involves iteratively solving a parameterized mixed-integer program to obtain a transportation delivery schedule, followed by solving an MDP for the dynamic purchasing decision. To conclude, Brinkmann et al. (2019) resort to a non-parametric policy search method to find suitable and dynamic lookahead horizons. The DLA horizons per hour are autonomously parameterized through VFA.

2.5.6.3 Methods' hyperparameters

We proceed to investigate the tuning procedures behind the optimization methods, focusing on hyperparameters **(H)**. Hvattum et al. (2009) develop a greedy randomized adaptive search procedure to solve a scenario tree at every decision state. At each iteration, a single decision is randomly selected either from a dynamically updated fraction $\gamma \in [0, 1]$ of the top-ranked decisions (rank-based), or from a set of decisions whose values lie close to that of the best decision (value-based). In Hvattum and Løkketangen (2009), the calibration of the progressive hedging algorithm involves determining a fixed value of a single penalty parameter, ρ. Initial tests show that no single value of ρ works well for every scenario tree problem instance, proving that different values for ρ are needed for various problems, defending a dynamic update.

Nolz et al. (2014) propose an ALNS algorithm wherein each destroy and repair mechanism is assigned a dynamic weight representing its performance during the search, which controls the preference of a particular mechanism. Likewise, the ALNS metaheuristic of Coelho et al. (2014a) implies a previous tuning phase to obtain the best settings for the operators and parameters. Moreover, Markov et al. (2020) combines ALNS with simulated annealing to improve the metaheuristic's ability to avoid local optima. The authors first tune the simulated annealing-related parameters, followed by the ALNS-related and the operator-related parameters. Finally, Achamrah et al. (2022)'s reactive policy relies on the hybridization of mathematical modeling, genetic algorithms and deep reinforcement learning. The genetic algorithm parameters, such as crossover and mutation probability,

population size, and maximum number of iterations, are carefully tuned to enhance solution quality.

Table 2.8: Decision policies classification.

Authors (year)	Modeling	Decision epoch	Policy class	Optimization approach (Inventory Management + Routing)	Learning domain
1. Dror and Ball (1987)	Math	P	RO	Generalized assignment algorithm + C&W algorithm	-
2. Trudeau and Dror (1992)	Math	P	RO	Generalized assignment algorithm + C&W algorithm	-
3. Minkoff (1993)	MDP	P	VFA	Future value decomposition heuristic (B&B)	-
4. Reiman et al. (1999)	QCP	RT	RO	Heavy traffic approximation	-
5. Berman and Larson (2001)	SDP	RT	VFA	SDP algorithms + Predefined customer sequence	-
6. Jaillet et al. (2002)	Math	P	DLA_b	Incremental cost approximation + C&W, GRASP[a]	-
7. Kleywegt et al. (2002)	MDP	P	VFA	ADP (iterative heuristic) + Direct shipping	S, Q
8. Kleywegt et al. (2004)	MDP	P	VFA	ADP (iterative heuristic)	S, Q, R
9. Adelman (2004)	MDP	P	VFA	ADP (linear programming optimal dual prices)	-
10. Giesen et al. (2005)	Math	RT	RO	Commercial solver	-
11. Jarugumilli and Grasman (2007)	Math	RT	RO	Inventory policy + TSP	-
12. Hvattum et al. (2009)	MDP	P	DLA_b	Scenario-tree based heuristic: GRASP, Commercial solver	H
13. Hvattum and Løkketangen (2009)	MDP	P	DLA_b	Progressive hedging algorithm	H
14. Giesen et al. (2010)	Math	RT	RO	Commercial solver	-
15. Bertazzi et al. (2013)	DP	P	DLA_c	B&C algorithm	-
16. Vonolfen et al. (2013)	Math	P	PFA	Simheuristic: GA + Push forward insertion heuristic	S, Q
17. Nolz et al. (2014)	Math	P	DLA_b	ALNS algorithm, Commercial solver	H
18. Coelho et al. (2014a)	Math	P	DLA_a	ALNS matheuristic	H
19. Mes et al. (2014)	Math	P	PFA	Parameterized heuristic (HKG and SKO) + Cheapest insertion	S
20. Bertazzi et al. (2015)	SDP	P	DLA_c	Rollout-based matheuristic	-
21. Roldán et al. (2016)	Math	P	RO	(Priority + Inventory) policies + Commercial solver[b]	-
22. Elbek and Wøhlk (2016)	Math	P	DLA_a	VNS algorithm	-
23. Li et al. (2016)	Math	P	RO	Exact DP vs. Order-up policies + Exact DP vs. GA	-
24. Morales et al. (2018)	Math	P	SDP	SDP with finite stages approximation	-
25. Crama et al. (2018)	SDP	P	VFA	Matheuristic (Commercial solver)	-
26. Mirzapour Al-e hashem et al. (2019)	Math	P	DLA_b	Hybrid L-shaped method	-
27. Brinkmann et al. (2019)	MDP	RT	DLA_b, PFA, VFA	Policy search (offline sim.) + Static algorithm (online sim.)	Q, H
28. Markov et al. (2020)	Math	P	DLA_a	ALNS algorithm	H
29. Brinkmann et al. (2020)	MDP	RT	DLA_b	Heuristic-based inventory policies + Assignment problem	H
30. Violi et al. (2020)	Math	P	DLA_b	Commercial solver	-
31. McKenna et al. (2020)	MDP	P	VFA	ADP (Least squares temporal difference) + Direct shipping	S, Q
32. Raba et al. (2020)	Math	P	DLA_b, PFA	Simheuristic + Savings heuristic	S, Q
33. Cubillos et al. (2022)	Math	P	DLA_a	VNS algorithm	-
34. Achamrah et al. (2022)	Math	P	RO	Matheuristic (Commercial solver, GA and deep Q-learning)	H
35. Neves-Moreira et al. (2022)	Math	P	DLA_a	Exact B&C algorithm with patching heuristics	-
36. Cavagnini et al. (2022)	Math	P	DLA_b	Commercial solver	-
37. Cui et al. (2023)	Math	P	RO	Benders decomposition (Commercial solver)	-
38. Li et al. (2023)	Math	P	RO	Tabu search with column generation	-
39. Violi et al. (2023)	Math	P	DLA_b	Commercial solver	-
40. Cuellar-Usaquén et al. (2024)	MDP	P	DLA_b, CFA	Two-stage SP (Commercial solver) + Conventional routing heuristic	P
41. Ortega et al. (2024)	SDP	P	DLA_a, PFA	Policy learning + ALNS-based metaheuristic	S, Q
42. Chiu et al. (2024)	Math	P	DLA_a	Standard B&C algorithm	-
43. Jia et al. (2025)	MDP	P	CFA, DLA_b	MQRNN and Matheuristic algorithm	P
44. Greif et al. (2024)	MDP	P	CFA	Physics-Informed NN and Commercial solver	P
45. Hasturk et al. (2024)	Math + MDP	P	CFA, VF	Commercial solver (IRP) + Value iteration (purchasing)	Q, P
46. Liang et al. (2024)	MDP	RT	VFA	Dual Policy RL algorithm	S, Q, R

Legend: [a]Bard et al. (1998) | [b]Applegate et al. (2011) | DP - Dynamic programming; **Modeling:** Math - Mathematical programming | MDP - Markov decision process | QCP - Queueing control problem | SDP - Stochastic DP; **Decision epoch:** P - Periodic | RT - Real-time; **Decision policy:** Table 2.6; **Learning domain:** Table 2.7; **Optimization approach:** ADP - Approximate DP | ALNS - Adaptive large neighborhood search | B&B - Branch & bound | B&C - Branch & cut | C&W - Clarke and Wright | GA - Genetic algorithm | GRASP - Greedy randomized adaptive search procedure | HKG - Hierarchical knowledge gradient | MQRNN - Multi-horizon quantile recurrent NN | NN - Neural network | RL - Reinforcement learning | SKO - Sequential kriging optimization | SP - Stochastic programming | TSP - Traveling salesman problem | VNS - Variable neighborhood search

2.6. Theoretical and empirical instances

While several well-established, open-access IRP instances support comparability across studies (e.g., Coelho, 2012), they may not fully capture the range of complexities found in real-world contexts, such as demand uncertainty, inventory and service constraints, multi-echelon structures, or sufficiently long planning horizons. Therefore, we recognize the need to develop new open-access instance sets so that researchers can train, validate and benchmark their decision policies under realistic conditions. This section outlines the historical path of the most significant theoretical problem instances available in the literature, from

the deterministic IRP to the SDIRP. Additionally, Table 2.9 reports the main characteristics of the instances used across all reviewed papers.

The initial set of instances designed for the deterministic and static variant of the IRP was proposed by Archetti et al. (2007) and comprises 160 instances from five to 50 customers and three to six periods. Building on this, Coelho and Laporte (2013) introduced a new set of 810 instances for the multiple-product-vehicle IRP. Additionally, Coelho et al. (2014a) proposed a novel collection of 450 benchmark instances for the single-product-vehicle SDIRP. According to the authors, these follow some of the standards adopted by more than a dozen papers in IRP, such as the mean demand, initial inventory levels, vehicle capacity, and geographical coordinates.

Subsequent studies on the SDIRP have modified these benchmarks to address broader problem settings. For instance, Bertazzi et al. (2013) developed two test sets based on Archetti et al. (2007)'s instances, while Roldán et al. (2016) adopted the single-vehicle-product instances of Coelho et al. (2014a), with five to 200 customers and five to 20 periods. Later, Achamrah et al. (2022) extended the previous set to approach the multi-vehicle-product SDIRP. Additional extensions derive from changes to classic VRP instances. For example, Crama et al. (2018) considered the original VRP instances of Solomon (1987) with some modifications. Moreover, the testbed of Raba et al. (2020) was obtained from the 27 VRP instances of Augerat et al. (1995) and adapted for the IRP by Juan et al. (2014).

Furthermore, Neves-Moreira et al. (2022) addressed the Coelho and Laporte (2013)'s instances for the multi-vehicle-product IRP variant. Chiu et al. (2024) utilized a collection of instances similar to those presented by Coelho et al. (2014a). Moreover, Greif et al. (2024) incorporated additional demand scenarios and aligned the remaining parameters with those in Archetti et al. (2007), Archetti et al. (2011) and Coelho et al. (2014a). Recently, Bertazzi et al. (2025) published their online IRP instances and computational results at Coelho (2025).

Some other studies address real-world cases. For example, Trudeau and Dror (1992) collected a large and representative real-world database from one district in Pennsylvania, containing more than 2 000 customers. Vonolfen et al. (2013) considered a large-scale scenario based on an Austrian retailer, including 84 supermarkets and a central depot. Cuellar-Usaquén et al. (2024) combined data shared by a Colombian agri-food e-commerce platform and publicly available data to capture market dynamics. Typically, these studies do not share their instances due to confidentiality concerns. Nevertheless, they provide some statistics of their instances, which allow some level of reproducibility. There is also an opportunity to extend the more established instances with features of these real-world cases.

2.7. Conclusions and future directions

Over the past decade, significant advancements have been made in understanding and representing the inherent complexities of the SDIRP. This review presents a thorough analysis of the literature, categorizing the existing studies according to problem settings and adopted decision policies. This section starts by pointing out research gaps in the SDIRP literature, which can serve as important directions for future extensions, thereby enhancing

Table 2.9: Summary of instances across reviewed papers.

Authors (year)	Suppliers	Customers	Planning horizon	Vehicles	Products	Source
1. Dror and Ball (1987)	1	10^3 (thousands)	12 weeks (5 weekdays)	2	1	RW
2. Trudeau and Dror (1992)	1	> 2000	12 weeks (5 weekdays)	2	1	RW
3. Minkoff (1993)	1	[3, 10]	100 periods	n.d.	1	RG
4. Reiman et al. (1999)	1	5	{36k, 240k} time units (RT)	1	1	RG
5. Berman and Larson (2001)	1	n.d. (e.g., 3)	1 year (RT)	1	1	RG
6. Jaillet et al. (2002)	n.d.	n.d.	{0.5, 1, 2, 5, 10} years	n.d.	1	RG
7. Kleywegt et al. (2002)	1	[2, 60]	800 periods	[1, 16]	1	RG (ref. to [b])
8. Kleywegt et al. (2004)	1	[4, 20]	800 periods	[1, 9]	1	RG (ref. to [c]), RW
9. Adelman (2004)	1	[10, 40]	600 periods	n.d.	1	RG (ref. to [d])
10. Giesen et al. (2005)	1	5	100 hours (RT)	1	1	RG
11. Jarugumilli and Grasman (2007)	1	10	5 time units (RT)	1	1	RG
12. Hvattum et al. (2009)	1	[2, 60]	600 periods	[1, 30]	1	RG (ref. to [b,c,d])
13. Hvattum and Løkketangen (2009)	1	[2, 60]	600 periods	[1, 30]	1	RG (ref. to [b,c,d])
14. Giesen et al. (2010)	1	7	100 hours (RT)	1	1	RG
15. Bertazzi et al. (2013)	1	[5, 50]	{3, 6, 9} periods	1	1	RG (ref. to [e])
16. Vonolfen et al. (2013)	1	84	3 months (7 weekdays)	n.d.	5113	RW
17. Nolz et al. (2014)	1	{5, 30}	{10, 20} periods	1	1	RG, RW
18. Coelho et al. (2014a)	1	[5, 200]	{5, 10, 20} periods	1	1	RG (ref. to [e,g])
19. Mes et al. (2014)	1	{100, 378, 500}	24 weeks (5 weekdays)	{1, 2}	1	RW
20. Bertazzi et al. (2015)	1	{2, 5, 10, 20}	{3, 6} periods	1	1	RG (ref. to [e])
21. Roldán et al. (2016)	1	[5, 200]	{5, 10, 20} periods	1	1	RG (ref. to [g])
22. Elbek and Wøhlk (2016)	2	{1764, 2052}	365 days	1	2	RW
23. Li et al. (2016)	1	{2, 14}	{8, 100} periods	1	{1, 3}	RG
24. Morales et al. (2018)	1	[5, 35]	3 periods	2	1	RG (ref. to [f])
25. Crama et al. (2018)	1	40	30 periods	n.d.	1	RG (ref. to [a])
26. Mirzapour Al-e hashem et al. (2019)	8	1	{2, 3, 5} periods	12	5	RG
27. Brinkmann et al. (2019)	1	169	24 hours (RT)	1	1	RW
28. Markov et al. (2020)	2	(avg, max) ∈ {(41, 86)}	{1, 2} weeks (5 weekdays)	{1, 2}	1	RW
29. Brinkmann et al. (2020)	1	{35, 169}	24 hours (RT)	[1, 4]	1	RW
30. Violi et al. (2020)	[1, 6]	[5, 30]	1 week (7 weekdays)	n.d.	1	RW
31. McKenna et al. (2020)	1	{9, 12, 15}	{1, 2, 3} months (6h-periods)	{4, 6, 8}	1	RW
32. Raba et al. (2020)	1	[27, 80]	[3, 10] periods	[5, 10]	1	RG (ref. to [h])
33. Cubillos et al. (2022)	1	100	200 periods	1	1	RG
34. Achamrah et al. (2022)	1	[5, 200]	{5, 10, 20} periods	{1, 10}	{1, 20, 40}	RG (ref. to [g])
35. Neves-Moreira et al. (2022)	1	[10, 100]	[3, 10] periods	{1, 3, 5}	{1, 3, 5}	RG (ref. to [f])
36. Cavagnini et al. (2022)	1	{2, 3, 4}	[1, 5] periods	n.d.	1	RG
37. Cui et al. (2023)	1	{10, 109}	{4, 5, 30} periods	1	1	RG, RW
38. Li et al. (2023)	1	[21, 101]	[5, 8] periods	1	1	RW
39. Violi et al. (2023)	2	8	1 week (7 weekdays)	3	1	RW
40. Cuellar-Usaquén et al. (2024)	20	1	20 periods	n.d.	5	RW
41. Ortega et al. (2024)	1	[5, 40]	6 periods (5 subperiods)	[2, 5]	1	RW
42. Chiu et al. (2024)	1	{10, 20, 30}	20 periods	{1, 2, 3}	1	RG (ref. to [f])
43. Jia et al. (2025)	1	[5, 18]	30 periods	n.d.	1	RW
44. Greif et al. (2024)	1	10	10 periods	1	1	RG (ref. to [e,g])
45. Hasturk et al. (2024)	1	{10, 15, 20}	{7, 10} cycle length	n.d.	1	RW
46. Liang et al. (2024)	21	60	50 periods	4	1	RG, RW

Legend: Solomon (1987)[a] | Kleywegt et al. (2002)[b] | Kleywegt et al. (2004)[c] | Adelman (2004)[d] | Archetti et al. (2007)[e] | Coelho and Laporte (2013)[f] | Coelho et al. (2014a)[g] | RG - Randomly generated instances | RW - Real-world instances | n.d. - not defined

the realism of these problems. In addition, we shed some light on how different problem characteristics shape the design of decision policies, while identifying specific points that need further research.

2.7.1 Bridging the gap to realistic problems

Although all problem settings discussed in Section 2.4 have been well explored in static IRP variants, the same does not apply to the stochastic dynamic one. This limitation arises from the increased scale and complexity of the problem, characterized by large state and decision spaces, making the introduction of additional features a non-trivial task. Accordingly, we highlight the overlooked dimensions that can lead to more realistic models.

Extended network configurations. An overlooked dimension within the SDIRP is the evaluation of multiple-supplier networks (e.g., Markov et al., 2020, discussed in Section

2.4.1.1). Considering multiple suppliers enhances resilience and provides more flexibility in decision-making when responding to supply disruptions and customer stockouts. However, it also introduces new coordination challenges, such as supplier selection, fleet allocation, and delivery schedule synchronization. In short, exploring these extended network configurations represents a promising path to developing more realistic and resilient distribution systems.

Multiple items and perishability. There is a predominance of single-item SDIRP models in the current literature (outlined in Section 2.4.2.2). Consequently, extending the current models to handle multiple products represents an important avenue for future research. There is also a need to explore perishable items, where different shelf lives are often modeled as multiple items. Perishability requires careful study in the context of the SDIRP (refer to Section 2.4.2.3), because it directly impacts both inventory management and vehicle routing (e.g., Violi et al., 2023). For instance, any minor delay in delivery can significantly impact inventory waste and lost revenues, depending on the product's shelf life and storage conditions. Therefore, delivery time windows and expiration constraints must be considered during the modeling stage. These challenges are particularly relevant in applications and business sectors such as the distribution of medicines, groceries or emergency services, in which coordination and resource sharing among suppliers is essential to reduce costs.

Pickup and delivery operations. Traditional SDIRP formulations often simplify operational complexities found in real-world settings, such as flexible delivery structures, which are essential for managing real-time demand fluctuations and network disruptions. In particular, P&D operations play a critical role in supply chain management by offering operational flexibility, such as in retail and medical supply (e.g., Mirzapour Al-e hashem et al., 2019), where demand variability and inventory shortages are expected (addressed in Section 2.4.3.5). However, for the sake of simplicity, simultaneous P&D operations are often omitted from the existing SDIRP models. Consequently, most studies in the literature focus exclusively on delivery, overlooking the potential benefits of integrating both P&D. As an exception, bike-sharing systems stand out (e.g., Brinkmann et al., 2020). Furthermore, split delivery represents another important but underexplored dimension within the SDIRP, despite being extensively studied within the VRP community (as mentioned in Section 2.4.3.7).

Multiple uncertainty sources. While most studies consider customer demand the only source of stochasticity, we see potential in expanding this scope to further sources of uncertainty. For example, supply fluctuations (e.g., Cuellar-Usaquén et al., 2024) directly influence inventory availability and the robustness of replenishment strategies. Moreover, traffic and weather conditions introduce variability in delivery times, causing delays and missed time windows, which in turn lead to increased operational costs. Despite their relevance, the existing road-based SDIRP literature often overlooks travel times (e.g., Morales et al., 2018) and driver availability as further sources of uncertainty (discussed in Section 2.4.1.2). To mitigate such disruptions, emergency lateral transshipments (e.g., Achamrah et al., 2022) and direct deliveries from suppliers appear as recourse mechanisms to face stockouts and balance inventory levels across customer locations (refer to Section 2.4.3.6).

For instance, their significance is reflected in electronics and fashion delivery networks, where communication between stores is essential for a prompt redistribution of products.

Multiple objectives. Expanding the current SDIRP models to include additional objectives could substantially enhance their practical relevance, despite increasing the problem's complexity. Our survey reveals a single study addressing multi-objective optimization (Mirzapour Al-e hashem et al., 2019, outlined in Section 2.4.1.4). For instance, sustainability always stands out when discussing supply chain management and logistics operations as one of the biggest concerns. Global environmental awareness should contribute to incorporating sustainable goals into emerging problems, in addition to the classical inventory holding, stockout and transportation costs. These objectives might include minimizing the carbon footprint, with a particular focus on greenhouse gas emissions. Beyond environmental considerations, further operational objectives could include minimizing costs linked to drivers' labor (e.g., per hour or per day) and expenses related to warehouse handling activities.

2.7.2 Designing sequential decision policies

The plethora of problems and solution approaches reviewed in this paper exhibits important differences. Clearly, no single approach is universally superior, and its choice will highly depend on the problem's characteristics. The following discussion explores how the scale, complexity, and real-time considerations of a given problem can influence the choice of a policy class, decomposition approach and optimization method.

2.7.2.1 Small-scale problems

Small-scale problems, involving a limited number of suppliers, customers, products, and vehicles, are well-suited for exact methods and integrated decision policies that seek to balance anticipatory decision-making and computational efficiency. In this context, and looking at the current literature, DLA policies are the most requested, being particularly effective when solving finite planning horizons, building upon classical mathematical programming formulations to jointly optimize inventory and routing decisions, capturing their interdependencies to achieve optimal or near-optimal solutions (refer to Section 2.5.3). However, as the problem scale increases (for instance, with a larger lookahead period), these policies become more computationally demanding.

Lookahead models have evolved over time, incorporating machine learning algorithms to obtain accurate demand scenarios in stochastic environments, based on historical data and further contextual information. However, most studies model demands as being stochastic and stationary, with decisions typically based on probabilistic assumptions or one-point predictions via regression algorithms. Along with Jia et al. (2025), we see an opportunity to develop new policies that move beyond static distributional assumptions and single-scenario forecasts, by leveraging machine learning to create data-driven future demand scenarios for each customer (refer to Section 2.5.5). These advances would improve anticipatory decision-making and facilitate the integration of learning and simulation-based optimization, contributing to robust and adaptive policies.

To further enhance DLA policies, CFA can be a valuable tool. By leveraging parametric approximations, they can enhance existing lookahead models, preserving solution quality while reducing the computational burden of complex stochastic programming formulations. For example, Cuellar-Usaquén et al. (2024) leverage CFAs to account for routing decisions without using explicit variables. The method evaluates the potential cost of decisions within the stochastic lookahead, maintaining solution integrity within reasonable computational times. Although challenges remain in terms of parameter design and calibration, CFA serves as a natural bridge between machine learning and stochastic programming (Powell, 2022b).

Building on this connection between learning and optimization, applying VFAs to complex configurations can be quite challenging. For instance, as the number of customers or products increases, the state space can expand significantly, making it difficult to estimate state-value and action-value functions. As a result, their application has been most feasible in simplified SDIRP settings, such as single-item environments (e.g., Liang et al., 2024). In scenarios where routing decisions do not require continuous optimization, such as regular tours or direct deliveries, VFAs can be valuable for inventory decisions. Thus, VFAs should also be considered in small-scale problems with simplified settings, as they can effectively capture the long-term effects of sequential decisions.

2.7.2.2 Large-scale problems

As the scale and complexity of the SDIRP increase, fully integrated optimization becomes computationally challenging. As outlined for small-scale problems, CFA appears as a promising underexplored approach when hybridized with DLA policies. Their strength lies in adjusting the original objective function and problem constraints by introducing calibrated parameters that simplify the overall problem (see Section 2.5.2.2), contributing to reducing the dimensionality of state and decision spaces. Alternatively, CFA might contribute to developing richer single-period models solved in an RH framework by introducing tunable parameters, such as delivery scores, that implicitly account for future uncertainties and costs (e.g., Greif et al., 2024).

Despite advances in DLA-CFA hybridization, large-scale problems tend to favor approximate methods, such as metaheuristics and matheuristics, and decoupled policies, where inventory and routing decisions are determined separately. This decomposition reduces the global computational complexity, although sacrificing optimality. These policies are typically handled sequentially, with inventory decisions made first and routing afterwards (e.g., Cuellar-Usaquén et al., 2024). Alternatively, vehicle routes might be static and computed at the beginning of the planning horizon for all subsequent periods (i.e., regular tours), whereas inventory decisions are periodically optimized (e.g., Cui et al., 2023).

PFAs play an important role in these decoupled decision frameworks, contributing with non-parametric and parametric decision rules, e.g., delivery priority rules to prioritize customer visits and inventory replenishment rules to compute delivery quantities (discussed in Section 2.5.2.1). Despite its potential, its application within the field remains limited. On the other hand, VFAs are suitable for infinite-horizon problems, where both short-term and long-term effects must be jointly considered (refer to Section 2.5.4). For example, Liang

et al. (2024) address a bike-sharing logistics system that utilizes VFAs in a decoupled fashion, with individual neural networks for inventory and routing decisions. Despite emerging as a promising approach for dynamic problems with large or continuous state spaces, the training process can be computationally demanding, and scalability remains a challenge for high-dimensional SDIRPs. Thus, further research on this topic is needed.

2.7.2.3 Real-time problems

A notable trend in VMI replenishment systems lies in the improvement, i.e., quantity and quality, of information shared among stakeholders and collected by IoT devices and sensors. This data, which may contain real-time inventory levels, historical sales, seasonal trends, and vehicle tracking, is an opportunity to design enhanced decision support systems. By leveraging this data, adaptive and machine learning-based policies have gained significant attention within the operations research community, especially in uncertain and dynamic environments. Such policies enable continuous adjustment to real-time information realizations and evolving system states, offering a practical alternative to DLAs, usually adopted when other policy classes fail to perform as expected (Powell, 2022b).

In practice, fully integrated inventory and routing decisions often sacrifice feasibility, as DLAs become computationally impractical, particularly when decisions must be made at tight intervals such as minutes, or even seconds. PFAs are particularly useful in situations that require straightforward and transparent decision-making, becoming suitable for real-time applications. Despite the limited exploration of policy search methods within the SDIRP literature, and given the promising outcomes in similar sequential decision problems, such as the SDVRP, we reinforce the importance of continuing to explore PFA and CFA policies, as well as hybrid approaches, which may also bring DLAs into the discussion (see Section 2.5.5). With such hybridization, we could take advantage of each approach's strengths by leveraging DLA's ability to handle stochasticity and exploiting the flexibility and efficiency of policy-based methods to explore large state and decision spaces.

Alternatively, VFAs may also offer a good trade-off between accuracy and responsiveness, though the training effort can still be substantial. In this regard, reinforcement and supervised learning are two machine learning strategies that provide promising avenues for approximating these functions, contributing to adaptive and efficient policy learning (refer to Section 2.5.6). Once the learning stage is completed, the resulting policies can be easily applied to adjust inventory and routing decisions without further demanding optimization. Finally, we find it interesting to bring explainability and transparency to the policies, resorting to explainable artificial intelligence (AI) methods instead of the traditional black boxes. This would help stakeholders to understand and trust complex decisions more easily.

Acknowledgements

This work was supported by FCT - Fundação para a Ciência e Tecnologia [reference: 2022.13466.BD and DOI identifier https://doi.org/10.54499/2022.13466.BD]; the European Union's Horizon 2020 Research and Innovation Programme [TRUST-AI research

project, grant number 952060]; the European Union's Horizon Europe Research and Innovation Programme [PEER research project, grant number 101120406].

Bibliography

Achamrah, F.E., Riane, F., Limbourg, S., 2022. Solving inventory routing with transshipment and substitution under dynamic and stochastic demands using genetic algorithm and deep reinforcement learning. International Journal of Production Research 60, 6187–6204. https://doi.org/10.1080/00207543.2021.1987549.

Adelman, D., 2004. A price-directed approach to stochastic inventory/routing. Operations Research 52, 499–514. https://doi.org/10.1287/opre.1040.0114.

Aghezzaf, E.H., 2008. Robust distribution planning for supplier-managed inventory agreements when demand rates and travel times are stationary. Journal of the Operational Research Society 59, 1055–1065. https://doi.org/10.1057/palgrave.jors.2602444.

Agra, A., Christiansen, M., Hvattum, L.M., Rodrigues, F., 2018. Robust optimization for a maritime inventory routing problem. Transportation Science 52, 509–525. https://doi.org/10.1287/trsc.2017.0814.

Alinaghian, M., Tirkolaee, E.B., Dezaki, Z.K., Hejazi, S.R., Ding, W., 2021. An augmented tabu search algorithm for the green inventory-routing problem with time windows. Swarm and Evolutionary Computation 60, 100802. https://doi.org/10.1016/j.swevo.2020.100802.

Andersson, H., Hoff, A., Christiansen, M., Hasle, G., Løkketangen, A., 2010. Industrial aspects and literature survey: Combined inventory management and routing. Computers & Operations Research 37, 1515–1536. https://doi.org/10.1016/j.cor.2009.11.009.

Applegate, D.L., Bixby, R.E., Chvátal, V., Cook, W.J., 2011. The traveling salesman problem: a computational study, Princeton University Press. https://doi.org/10.1515/9781400841103.

Archetti, C., Bertazzi, L., 2021. Recent challenges in routing and inventory routing: E-commerce and last-mile delivery. Networks 77, 255–268. https://doi.org/10.1002/net.21995.

Archetti, C., Bertazzi, L., Laporte, G., Speranza, M.G., 2007. A branch-and-cut algorithm for a vendor-managed inventory-routing problem. Transportation Science 41, 382–391. https://doi.org/10.1287/trsc.1060.0188.

Archetti, C., Bertazzi, L., Paletta, G., Speranza, M.G., 2011. Analysis of the maximum level policy in a production-distribution system. Computers & Operations Research 38, 1731–1746. https://doi.org/10.1016/j.cor.2011.03.002.

Augerat, P., Naddef, D., Belenguer, J., Benavent, E., Corberan, A., Rinaldi, G., 1995. Computational results with a branch and cut code for the capacitated vehicle routing problem .

Baita, F., Ukovich, W., Pesenti, R., Favaretto, D., 1998. Dynamic routing-and-inventory problems: a review. Transportation Research Part A: Policy and Practice 32, 585–598. https://doi.org/10.1016/S0965-8564(98)00014-7.

Bard, J.F., Huang, L., Jaillet, P., Dror, M., 1998. A decomposition approach to the inventory routing problem with satellite facilities. Transportation science 32, 189–203. https://doi.org/10.1287/trsc.32.2.189.

Bell, W.J., Dalberto, L.M., Fisher, M.L., Greenfield, A.J., Jaikumar, R., Kedia, P., Mack, R.G., Prutzman, P.J., 1983. Improving the distribution of industrial gases with an on-line computerized routing and scheduling optimizer. Interfaces 13, 4–23. https://doi.org/10.1287/inte.13.6.4.

Bengio, Y., Lodi, A., Prouvost, A., 2021. Machine learning for combinatorial optimization: a methodological tour d'horizon. European Journal of Operational Research 290, 405–421. https://doi.org/10.1016/j.ejor.2020.07.063.

Berman, O., Larson, R.C., 2001. Deliveries in an inventory/routing problem using stochastic dynamic programming. Transportation Science 35, 192–213. https://doi.org/10.1287/trsc.35.2.192.10134.

Bertazzi, L., Bosco, A., Guerriero, F., Lagana, D., 2013. A stochastic inventory routing problem with stock-out. Transportation Research Part C: Emerging Technologies 27, 89–107. https://doi.org/10.1016/j.trc.2011.06.003.

Bertazzi, L., Bosco, A., Laganà, D., 2015. Managing stochastic demand in an inventory routing problem with transportation procurement. Omega 56, 112–121. https://doi.org/10.1016/j.omega.2014.09.010.

Bertazzi, L., Chagas, G.O., Coelho, L.C., Laganà, D., Vocaturo, F., 2025. Online algorithms for the multi-vehicle inventory-routing problem with real-time demands. Transportation Research Part C: Emerging Technologies 170, 104892. https://doi.org/10.1016/j.trc.2024.104892.

Bertazzi, L., Paletta, G., Speranza, M.G., 2002. Deterministic order-up-to level policies in an inventory routing problem. Transportation Science 36, 119–132. https://doi.org/10.1287/trsc.36.1.119.573.

Blom, M., Shekh, S., Gossink, D., Miller, T., Pearce, A.R., 2020. Inventory routing for defense: Moving supplies in adversarial and partially observable environments. The Journal of Defense Modeling and Simulation 17, 55–81. https://doi.org/10.1177/1548512918798056.

Van den Bogaert, J., Van Jaarsveld, W., 2022. Vendor-managed inventory in practice: understanding and mitigating the impact of supplier heterogeneity. International Journal of Production Research 60, 6087–6103. https://doi.org/10.1080/00207543.2021.1983222.

Brinkmann, J., Ulmer, M.W., Mattfeld, D.C., 2019. Dynamic lookahead policies for stochastic-dynamic inventory routing in bike sharing systems. Computers & Operations Research 106, 260–279. https://doi.org/10.1016/j.cor.2018.06.004.

Brinkmann, J., Ulmer, M.W., Mattfeld, D.C., 2020. The multi-vehicle stochastic-dynamic inventory routing problem for bike sharing systems. Business Research 13, 69–92. https://doi.org/10.1007/s40685-019-0100-z.

Campbell, A., Clarke, L., Kleywegt, A., Savelsbergh, M., 1998. The inventory routing problem, in: Fleet management and logistics. Springer, pp. 95–113. https://doi.org/10.1007/978-1-4615-5755-5_4.

Cavagnini, R., Bertazzi, L., Maggioni, F., 2022. A rolling horizon approach for a multi-stage stochastic fixed-charge transportation problem with transshipment. European Journal of Operational Research 301, 912–922. https://doi.org/10.1016/j.ejor.2021.11.037.

Chiu, A., Angulo, G., Larrain, H., 2024. Optimizing the long-term costs of an inventory routing problem using linear relaxation. Transportation Research Part E: Logistics and Transportation Review 183, 103447. https://doi.org/10.1016/j.tre.2024.103447.

Coelho, L., 2012. Inventory routing problem instances. https://www.leandro-coelho.com/instances/inventory-routing/ (Accessed: 28 April 2025).

Coelho, L., 2025. Online irp algorithms. https://www.leandro-coelho.com/online-irp-algorithms/ (Accessed: 28 April 2025).

Coelho, L.C., Cordeau, J.F., Laporte, G., 2012. The inventory-routing problem with transshipment. Computers & Operations Research 39, 2537–2548. https://doi.org/10.1016/j.cor.2011.12.020.

Coelho, L.C., Cordeau, J.F., Laporte, G., 2014a. Heuristics for dynamic and stochastic inventory-routing. Computers & Operations Research 52, 55–67. https://doi.org/10.1016/j.cor.2014.07.001.

Coelho, L.C., Cordeau, J.F., Laporte, G., 2014b. Thirty years of inventory routing. Transportation Science 48, 1–19. https://doi.org/10.1287/trsc.2013.0472.

Coelho, L.C., Laporte, G., 2013. A branch-and-cut algorithm for the multi-product multi-vehicle inventory-routing problem. International Journal of Production Research 51, 7156–7169. https://doi.org/10.1080/00207543.2012.757668.

Crama, Y., Rezaei, M., Savelsbergh, M., Woensel, T.V., 2018. Stochastic inventory routing for perishable products. Transportation Science 52, 526–546. https://doi.org/10.1287/trsc.2017.0799.

Cubillos, M., Spliet, R., Wøhlk, S., 2022. On the effect of using sensors and dynamic forecasts in inventory-routing problems. INFOR: Information Systems and Operational Research 60, 473–490. https://doi.org/10.1080/03155986.2022.2073110.

Cuellar-Usaquén, D., Ulmer, M.W., Gomez, C., Álvarez-Martínez, D., 2024. Adaptive stochastic lookahead policies for dynamic multi-period purchasing and inventory routing. European Journal of Operational Research 318, 1028–1041. https://doi.org/10.1016/j.ejor.2024.06.011.

Cui, Z., Long, D.Z., Qi, J., Zhang, L., 2023. The inventory routing problem under uncertainty. Operations Research 71, 378–395. https://doi.org/10.1287/opre.2022.2407.

Dror, M., Ball, M., 1987. Inventory/routing: Reduction from an annual to a short-period problem. Naval Research Logistics (NRL) 34, 891–905. https://doi.org/10.1002/1520-6750(198712)34:6<891::AID-NAV3220340613>3.0.CO;2-J.

Elbek, M., Wøhlk, S., 2016. A variable neighborhood search for the multi-period collection of recyclable materials. European Journal of Operational Research 249, 540–550. https://doi.org/10.1016/j.ejor.2015.08.035.

Federgruen, A., Tzur, M., 1996. Detection of minimal forecast horizons in dynamic programs with multiple indicators of the future. Naval Research Logistics (NRL) 43, 169–189. https://doi.org/10.1002/(SICI)1520-6750(199603)43:2%3C169::AID-NAV2%3E3.0.CO;2-8.

Federgruen, A., Zipkin, P., 1984. A combined vehicle routing and inventory allocation problem. Operations research 32, 1019–1037. https://doi.org/10.1287/opre.32.5.1019.

Giesen, R., Mahmassani, H.S., Jaillet, P., 2005. Strategies for online inventory routing problem under real-time information. Transportation research record 1923, 164–179. https://doi.org/10.1177/0361198105192300118.

Giesen, R., Mahmassani, H.S., Jaillet, P., 2010. Coping with disruptions: Performance comparison of strategies for online inventory routing systems under demand surges. Transportation research record 2168, 63–70. https://doi.org/10.3141/2168-08.

Greif, T., Bouvier, L., Flath, C.M., Parmentier, A., Rohmer, S.U., Vidal, T., 2024. Combinatorial optimization and machine learning for dynamic inventory routing. arXiv preprint arXiv:2402.04463 https://doi.org/10.48550/arXiv.2402.04463.

Gruler, A., Panadero, J., de Armas, J., Pérez, J.A.M., Juan, A.A., 2018. Combining variable neighborhood search with simulation for the inventory routing problem with stochastic demands and stock-outs. Computers & Industrial Engineering 123, 278–288. https://doi.org/10.1016/j.cie.2018.06.036.

Mirzapour Al-e hashem, S.M., Rekik, Y., Hoseinhajlou, E.M., 2019. A hybrid l-shaped method to solve a bi-objective stochastic transshipment-enabled inventory routing problem. International Journal of Production Economics 209, 381–398. https://doi.org/10.1016/j.ijpe.2017.06.020.

Hasturk, U., Schrotenboer, A.H., Ursavas, E., Roodbergen, K.J., 2024. Stochastic cyclic inventory routing with supply uncertainty: A case in green-hydrogen logistics. Transportation Science 58, 315–339. https://doi.org/10.1287/trsc.2022.0435.

Hildebrandt, F.D., Thomas, B.W., Ulmer, M.W., 2023. Opportunities for reinforcement learning in stochastic dynamic vehicle routing. Computers & Operations Research 150, 106071. https://doi.org/10.1016/j.cor.2022.106071.

Huang, D., Allen, T.T., Notz, W.I., Zeng, N., et al., 2006. Global optimization of stochastic black-box systems via sequential kriging meta-models. Journal of global optimization 34, 441–466. https://doi.org/10.1007/s10898-005-2454-3.

Hvattum, L.M., Løkketangen, A., 2009. Using scenario trees and progressive hedging for stochastic inventory routing problems. Journal of Heuristics 15, 527–557. https://doi.org/10.1007/s10732-008-9076-0.

Hvattum, L.M., Løkketangen, A., Laporte, G., 2009. Scenario tree-based heuristics for stochastic inventory-routing problems. INFORMS Journal on Computing 21, 268–285. https://doi.org/10.1287/ijoc.1080.0291.

Jaillet, P., Bard, J.F., Huang, L., Dror, M., 2002. Delivery cost approximations for inventory routing problems in a rolling horizon framework. Transportation science 36, 292–300. https://doi.org/10.1287/trsc.36.3.292.7829.

Jarugumilli, S., Grasman, S.E., 2007. RFID-enabled inventory routing problems. International Journal of Manufacturing Technology and Management 10, 92–105. https://doi.org/10.1504/IJMTM.2007.011403.

Jia, M., Schrotenboer, A.H., Chen, F., 2025. Scenario predict-then-optimize for data-driven online inventory routing. Transportation Science 59, 1032–1056. https://doi.org/10.1287/trsc.2024.0613.

Juan, A.A., Grasman, S.E., Caceres-Cruz, J., Bektaş, T., 2014. A simheuristic algorithm for the single-period stochastic inventory-routing problem with stock-outs. Simulation Modelling Practice and Theory 46, 40–52. https://doi.org/10.1016/j.simpat.2013.11.008.

Kleywegt, A.J., Nori, V.S., Savelsbergh, M.W., 2002. The stochastic inventory routing problem with direct deliveries. Transportation science 36, 94–118. https://doi.org/10.1287/trsc.36.1.94.574.

Kleywegt, A.J., Nori, V.S., Savelsbergh, M.W., 2004. Dynamic programming approximations for a stochastic inventory routing problem. Transportation Science 38, 42–70. https://doi.org/10.1287/trsc.1030.0041.

Li, M., Wang, Z., Chan, F.T., 2016. An inventory routing policy under replenishment lead time. IEEE Transactions on Systems, Man, and Cybernetics: Systems 47, 3150–3164. https://doi.org/10.1109/TSMC.2016.2562561.

Li, R., Cui, Z., Kuo, Y.H., Zhang, L., 2023. Scenario-based distributionally robust optimization for the stochastic inventory routing problem. Transportation Research Part E: Logistics and Transportation Review 176, 103193. https://doi.org/10.1016/j.tre.2023.103193.

Liang, J., Liu, D., Jena, S.D., Lodi, A., Vidal, T., 2024. Dual policy reinforcement learning for real-time rebalancing in bike-sharing systems. arXiv preprint arXiv:2406.00868 https://doi.org/10.48550/arXiv.2406.00868.

Liu, S.C., Lee, W.T., 2011. A heuristic method for the inventory routing problem with time windows. Expert Systems with Applications 38, 13223–13231. https://doi.org/10.1016/j.eswa.2011.04.138.

Markov, I., Bierlaire, M., Cordeau, J.F., Maknoon, Y., Varone, S., 2018. A unified framework for rich routing problems with stochastic demands. Transportation Research Part B: Methodological 114, 213–240. https://doi.org/10.1016/j.trb.2018.05.015.

Markov, I., Bierlaire, M., Cordeau, J.F., Maknoon, Y., Varone, S., 2020. Waste collection inventory routing with non-stationary stochastic demands. Computers & Operations Research 113, 104798. https://doi.org/10.1016/j.cor.2019.104798.

Markov, I., Lapparent, M.d., Bierlaire, M., Varone, S., 2015. Modeling a waste disposal process via a discrete mixture of count data models, in: In: Proceedings of the 15th Swiss Transport Research Conference, Monte Verità-Ascona, April 15-17, 2015, 16 p., STRC.

Mayring, P., et al., 2004. Qualitative content analysis. A companion to qualitative research 1, 159–176.

Mazyavkina, N., Sviridov, S., Ivanov, S., Burnaev, E., 2021. Reinforcement learning for combinatorial optimization: A survey. Computers & Operations Research 134, 105400. https://doi.org/10.1016/j.cor.2021.105400.

McKenna, R.S., Robbins, M.J., Lunday, B.J., McCormack, I.M., 2020. Approximate dynamic programming for the military inventory routing problem. Annals of Operations Research 288, 391–416. https://doi.org/10.1007/s10479-019-03469-8.

Mes, M., Schutten, M., Rivera, A.P., 2014. Inventory routing for dynamic waste collection. Waste Management 34, 1564–1576. https://doi.org/10.1016/j.wasman.2014.05.011.

Mes, M.R., Powell, W.B., Frazier, P.I., 2011. Hierarchical knowledge gradient for sequential sampling. Journal of Machine Learning Research 12.

Minkoff, A.S., 1993. A Markov decision model and decomposition heuristic for dynamic vehicle dispatching. Operations Research 41, 77–90. https://doi.org/10.1287/opre.41.1.77.

Moin, N.H., Salhi, S., 2007. Inventory routing problems: a logistical overview. Journal of the Operational Research Society 58, 1185–1194. https://doi.org/10.1057/palgrave.jors.2602264.

Morales, F., Franco, C., Mendez-Giraldo, G., 2018. Dynamic inventory routing problem: Policies considering network disruptions. International Journal of Industrial Engineering Computations 9, 523–534. https://doi.org/10.5267/j.ijiec.2017.11.001.

Neves-Moreira, F., Almada-Lobo, B., Guimarães, L., Amorim, P., 2022. The multi-product inventory-routing problem with pickups and deliveries: Mitigating fluctuating demand via rolling horizon heuristics. Transportation Research Part E: Logistics and Transportation Review 164, 102791. https://doi.org/10.1016/j.tre.2022.102791.

Niakan, F., Rahimi, M., 2015. A multi-objective healthcare inventory routing problem; a fuzzy possibilistic approach. Transportation Research Part E: Logistics and Transportation Review 80, 74–94. https://doi.org/10.1016/j.tre.2015.04.010.

Nolz, P.C., Absi, N., Feillet, D., 2014. A stochastic inventory routing problem for infectious medical waste collection. Networks 63, 82–95. https://doi.org/10.1002/net.21523.

Onggo, B.S., Panadero, J., Corlu, C.G., Juan, A.A., 2019. Agri-food supply chains with stochastic demands: A multi-period inventory routing problem with perishable products. Simulation Modelling Practice and Theory 97, 101970. https://doi.org/10.1016/j.simpat.2019.101970.

Ortega, E.J.A., Malicki, S., Doerner, K.F., Minner, S., 2024. Stochastic inventory routing with dynamic demands and intra-day depletion. Computers & Operations Research 163, 106503. https://doi.org/10.1016/j.cor.2023.106503.

Pérez Kaligari, E., Guerrero Rueda, W.J., 2015. Métodos de optimización para el problema de ruteo de vehículos con inventarios y ventanas de tiempo duras. Revista Ingeniería Industria 14, 31–49.

Pillac, V., Gendreau, M., Guéret, C., Medaglia, A.L., 2013. A review of dynamic vehicle routing problems. European Journal of Operational Research 225, 1–11. https://doi.org/10.1016/j.ejor.2012.08.015.

Powell, W.B., 2007. Approximate Dynamic Programming: Solving the curses of dimensionality. volume 703. John Wiley & Sons.

Powell, W.B., 2019. A unified framework for stochastic optimization. European Journal of Operational Research 275, 795–821. https://doi.org/10.1016/j.ejor.2018.07.014.

Powell, W.B., 2022a. Designing lookahead policies for sequential decision problems in transportation and logistics. IEEE Open Journal of Intelligent Transportation Systems 3, 313–327. https://doi.org/10.1109/OJITS.2022.3148574.

Powell, W.B., 2022b. Reinforcement Learning and Stochastic Optimization: A Unified Framework for Sequential Decisions. John Wiley & Sons, Hoboken, NJ. https://doi.org/10.1002/9781119815068.

Puterman, M.L., 2014. Markov decision processes: discrete stochastic dynamic programming. John Wiley & Sons.

Raba, D., Estrada-Moreno, A., Panadero, J., Juan, A.A., 2020. A reactive simheuristic using online data for a real-life inventory routing problem with stochastic demands. International Transactions in Operational Research 27, 2785–2816. https://doi.org/10.1111/itor.12776.

Reiman, M.I., Rubio, R., Wein, L.M., 1999. Heavy traffic analysis of the dynamic stochastic inventory-routing problem. Transportation Science 33, 361–380. https://doi.org/10.1287/trsc.33.4.361.

Rodrigues, F., Agra, A., Christiansen, M., Hvattum, L.M., Requejo, C., 2019. Comparing techniques for modelling uncertainty in a maritime inventory routing problem. European Journal of Operational Research 277, 831–845. https://doi.org/10.1016/j.ejor.2019.03.015.

Roldán, R.F., Basagoiti, R., Coelho, L.C., 2016. Robustness of inventory replenishment and customer selection policies for the dynamic and stochastic inventory-routing problem. Computers & Operations Research 74, 14–20. https://doi.org/10.1016/j.cor.2016.04.004.

Roldán, R.F., Basagoiti, R., Coelho, L.C., 2017. A survey on the inventory-routing problem with stochastic lead times and demands. Journal of Applied Logic 24, 15–24. https://doi.org/10.1016/j.jal.2016.11.010.

Sethi, S., Sorger, G., 1991. A theory of rolling horizon decision making. Annals of operations research 29, 387–415. https://doi.org/10.1007/BF02283607.

Shaabani, H., 2022. A literature review of the perishable inventory routing problem. The Asian Journal of Shipping and Logistics 38, 143–161. https://doi.org/10.1016/j.ajsl.2022.05.002.

Soeffker, N., Ulmer, M.W., Mattfeld, D.C., 2022. Stochastic dynamic vehicle routing in the light of prescriptive analytics: A review. European Journal of Operational Research 298, 801–820. https://doi.org/10.1016/j.ejor.2021.07.014.

Solomon, M.M., 1987. Algorithms for the vehicle routing and scheduling problems with time window constraints. Operations research 35, 254–265. https://doi.org/10.1287/opre.35.2.254.

Solyalı, O., Cordeau, J.F., Laporte, G., 2012. Robust inventory routing under demand uncertainty. Transportation Science 46, 327–340. https://doi.org/10.1287/trsc.1110.0387.

Soysal, M., Çimen, M., Belbağ, S., Toğrul, E., 2019. A review on sustainable inventory routing. Computers & Industrial Engineering 132, 395–411. https://doi.org/10.1016/j.cie.2019.04.026.

Trudeau, P., Dror, M., 1992. Stochastic inventory routing: Route design with stockouts and route failures. Transportation Science 26, 171–184. https://doi.org/10.1287/trsc.26.3.171.

Violi, A., De Maio, A., Fattoruso, G., Olivieri, M.G., 2023. An age-based dynamic approach for distribution of perishable commodities with stochastic demands. Soft Computing 27, 7039–7050. https://doi.org/10.1007/s00500-023-07917-3.

Violi, A., Laganá, D., Paradiso, R., 2020. The inventory routing problem under uncertainty with perishable products: an application in the agri-food supply chain. Soft Computing 24, 13725–13740. https://doi.org/10.1007/s00500-019-04497-z.

Vonolfen, S., Affenzeller, M., Beham, A., Lengauer, E., Wagner, S., 2013. Simulation-based evolution of resupply and routing policies in rich vendor-managed inventory scenarios. Central European Journal of Operations Research 21, 379–400. https://doi.org/10.1007/s10100-011-0232-5.

CHAPTER 3

Policy design for the stochastic dynamic inventory-routing problem

Learning policies for the stochastic dynamic inventory-routing problem via evolutionary algorithms

Francisco Maia* · **Gonçalo Figueira*** · **Fábio Neves-Moreira†**

Submitted to *Transportation Research Part E: Logistics and Transportation Review*, 2026

Highlights

- Models the stochastic dynamic problem as a Markov decision process;
- Proposes anticipatory and efficient policies based on a three-stage decision process;
- Builds on evolutionary algorithms to learn policy components;
- Evaluates the proposed policies through comprehensive experiments.

Abstract The stochastic dynamic inventory-routing problem (SDIRP) is a cornerstone of supply chain operations that combines inventory management and vehicle routing while addressing the stochastic and dynamic nature of customer demand. This research aims to provide new insights into solving the SDIRP, focusing on a central warehouse that replenishes a set of geographically dispersed customers over a finite multi-period planning horizon. The objective is to minimize the total inventory holding, stockout and routing costs. We propose a three-stage decision policy applied in a rolling horizon fashion that: (i) computes a priority score for each customer using a rule learned via genetic programming, (ii) determines the corresponding delivery quantities based on an (s,S) inventory replenishment rule whose control parameters are optimized using a genetic algorithm, and (iii) solves a routing problem using a mathematical solver. The resulting policies are evaluated on instances with up to 50 customers and 20 periods, across multiple demand uncertainty levels, unit holding and shortage costs, and vehicle capacity settings. In our computational experiments, the proposed policy outperforms existing reactive policies by at least 14%, while being orders of magnitude faster than direct lookahead approximations.

Keywords Stochastic dynamic inventory-routing · Routing · Inventory management · Sequential decision-making · Markov decision process · Evolutionary algorithms

*INESC TEC, Faculdade de Engenharia, Universidade do Porto, Rua Dr. Roberto Frias, Porto, 4200-465, Portugal
†INESC TEC, Faculdade de Economia, Universidade do Porto, Rua Dr. Roberto Frias, Porto, 4200-464, Portugal

3.1. Introduction

In response to the growing need for coordinated supply chain strategies, integrating inventory management and transportation planning becomes critical to improving responsiveness and operational performance. Rooted in the principles of vendor-managed inventory, where the supplier takes responsibility for monitoring and replenishing the inventory held by one or multiple customers, the inventory-routing problem (IRP) is a complex combinatorial optimization problem that involves the joint optimization of inventory and routing decisions over time. The increasing significance of this problem is reflected across multiple application sectors such as healthcare (e.g., Niakan and Rahimi, 2015), retail (e.g., Aghezzaf, 2008), military (e.g., McKenna et al., 2020), bike-sharing (e.g., Brinkmann et al., 2020) and waste collection (e.g., Elbek and Wøhlk, 2016).

Over the past decades, much of the literature on the IRP has focused on static and deterministic variants, leading to significant advances in modeling complex problem settings and developing effective solution methods. Nevertheless, these studies often fail to capture the uncertain and dynamic evolution of information in real-world environments. In practice, inventory-routing systems operate under continuously changing conditions, such as fluctuations in customer demand (e.g., Coelho et al., 2014) and inventory supply (e.g., Cuellar-Usaquén et al., 2024). Continuous replanning is therefore essential to a more informed, adaptive and effective decision-making. The joint coordination of inventory replenishment and vehicle routing under such stochastic dynamic conditions defines the stochastic dynamic inventory-routing problem (SDIRP).

In this paper, we consider an SDIRP where the supplier, operating through a central warehouse, is responsible for replenishing a set of geographically dispersed customers over a multi-period planning horizon. Customer demand is stochastic and gradually revealed at the end of each period. Accordingly, inventory and routing decisions must be repeatedly determined at the beginning of each period based on the current system state. Deliveries are subject to routing constraints, including limited vehicle capacity and a single route per period. The objective is to minimize the total expected inventory holding, stockout and routing costs.

The problem can be modeled as a sequential decision process characterized by a sequence of states, decisions, information realizations representing stochastic demand, and resulting costs. In this context, a practical solution is no longer a single and static plan for the entire planning horizon but rather a policy that guides inventory and routing decisions over time based on the observed state.

Most of the literature on SDIRPs refers to (integrated) policies based on direct lookahead approximations (DLAs) that anticipate future states and decisions by solving a deterministic or stochastic lookahead model in rolling horizon (RH). However, these can become computationally demanding for online application, particularly when solving large-scale instances. Therefore, we propose a three-stage policy that, in each period, (i) computes a delivery priority score for each customer based on the current state; (ii) determines the corresponding delivery quantities using an (s, S) inventory replenishment rule; (iii) combines the selected customers into a feasible vehicle route.

The priority rule is learned via genetic programming (GP), an evolutionary algorithm that generates symbolic expressions. The (s,S) inventory control parameters are computed using an adaptation of the revised power approximation (RPA) of Ehrhardt and Mosier (1984), whose parameters are further calibrated to the SDIRP setting using a genetic algorithm (GA). Finally, the sequence of customer visits is determined by solving a routing problem to optimality using a mathematical solver.

We conduct a comprehensive computational study to evaluate the performance of our proposed policy across a significant range of experimental settings. This includes problem instances with up to 50 customers (with different demand variability) and 20 periods, as well as different unit inventory holding and shortage cost configurations, and vehicle capacity settings. Our contributions are threefold:

1. We develop an anticipatory and computationally efficient policy, whose components (i.e., priority rule and (s,S) inventory control parameters) are learned with evolutionary algorithms; to the best of our knowledge, this is the first application of GP to dynamic inventory-routing.
2. We are the first to transfer and adapt the RPA concept to dynamic inventory-routing, linking stochastic inventory control with dynamic multi-period routing decisions.
3. We extend the benchmark instance design of Coelho et al. (2014) to represent a broader range of SDIRP settings and inventory-routing trade-offs, including different inventory storage and vehicle capacities, as well as unit holding and stockout costs.

The remainder of this paper is as follows. Section 3.2 reviews the related works on SDIRPs and sequential decision-making. The problem is defined in Section 3.3, including a general description, an illustrative example, and a Markov decision process (MDP) formulation. Section 3.4 details our solution method, including the learning strategies. The experimental setting is described in Section 3.5, including the instance design, performance metrics, and benchmark policies. Section 3.6 details our computational results, showing the value of each policy component and providing a scenario-level analysis. Finally, Section 3.7 presents the conclusions and discusses future research directions.

3.2. Related works

This section aims to contextualize our research within the following streams: SDIRP models (Section 3.2.1) and decision policies (Section 3.2.2) to cover the range of problem settings and existing solution approaches considered in the literature; and evolutionary algorithms as learning strategies (Section 3.2.3).

3.2.1 Stochastic dynamic inventory-routing models

Building on the vehicle routing problem (VRP), the IRP was first introduced by Bell et al. (1983) to integrate inventory management with transportation logistics, with the objective of minimizing the total expected inventory and transportation costs over a finite (e.g.,

Neves-Moreira et al., 2022) or infinite (e.g., Jia et al., 2025) time horizon. To better capture real-world complexities, in which information models are uncertain and gradually revealed over time, research has been shifting from static and deterministic variants toward the SDIRP. Most of the SDIRP literature assumes that a single supplier replenishes a limited set of customers over a multi-period horizon (e.g., Achamrah et al., 2022). Customer demand remains the predominant source of uncertainty (e.g., Li et al., 2023), being modeled as a random variable following a given probability distribution (e.g., Achamrah et al., 2022), which aligns with our problem.

Two main inventory replenishment policies are considered: order-up-to-level (OU) and maximum-level (ML). An OU policy assumes that whenever a customer is visited, its inventory is replenished up to its storage capacity (e.g., Cubillos et al., 2022). An ML policy allows the supplier to freely decide the quantity to deliver to each customer in each period of the planning horizon (e.g., Cui et al., 2023). When demand exceeds the available stock, stockouts can be considered as lost sales (e.g., Raba et al., 2020), or, less often, as backlogged orders to be fulfilled in subsequent periods (e.g., Jia et al., 2025). Regarding vehicle routing, fleets may consist of a single vehicle (e.g., Elbek and Wøhlk, 2016) or multiple vehicles (e.g., Chiu et al., 2024), which can be either homogeneous (e.g., Violi et al., 2023) or heterogeneous (e.g., Brinkmann et al., 2020).

The literature has been gradually shifting from static mathematical programming formulations toward sequential decision frameworks. Initial studies primarily focused on deterministic models solved reactively in an RH fashion, without accounting for future uncertainty (e.g., Dror and Ball, 1987). Proactive RH strategies subsequently emerged to account for future uncertainty, either through deterministic approximations based on single-point demand forecasts (e.g., Coelho et al., 2014) or through scenario-based formulations, leveraging Monte Carlo simulation and stochastic programming methods (e.g., Nolz et al., 2014). Nevertheless, these approaches may fail to fully capture the long-term effects of current inventory-routing decisions, leading to suboptimal outcomes. In this context, representing the SDIRP as an MDP provides a structured framework for dynamic decision-making under uncertainty by defining the concepts of state, decision, stochastic transition, and cost function, thereby supporting the development of effective policies that account for the long-term consequences of current decisions (e.g., Liang et al., 2024).

3.2.2 Sequential decision policies

Given the stochastic dynamic nature of the problem, a static solution for the entire planning horizon is no longer realistic, highlighting the need for a decision policy that evolves as new information becomes available. For a comprehensive review on policy design for the SDIRP, we refer to Maia et al. (2026).

Lookahead approximations provide a means to estimate the impact of a right-now decision on future states and decisions. In particular, DLA policies have been widely explored in the literature, in which current-period inventory-routing decisions are obtained by optimizing over a limited deterministic (e.g., Chiu et al., 2024) or stochastic (e.g., Cavagnini et al., 2022) lookahead horizon, combining an approximation of future information with future decisions. However, several challenges arise when designing a lookahead model,

including determining the number of future periods to consider, representing uncertainty, and balancing the trade-off between solution quality and computational effort.

The computational challenges faced by current DLA policies have increased the interest in alternative policy classes, including policy search methods that often avoid the need to solve computationally demanding lookahead models over time. Policy search involves searching over a set of policies or policy parameters to identify the one that performs best over time by minimizing the total expected costs. Examples include (s,S) inventory replenishment rules in which customer-dependent reorder points (s) and order-up-to-levels (S) are learned via simulation-optimization, and customer prioritization rules responsible for assigning delivery priorities or even deciding which customers to visit in a given period (e.g., Vonolfen et al., 2013; Mes et al., 2014).

Among the existing studies, the work of Roldán et al. (2016) is the one that closely aligns with our proposed policy, as it is grounded on a three-step decision structure. The authors decompose the inventory-routing decision process into selecting the customers to visit and determining the delivery quantities based on classic inventory replenishment rules, guided by a customer prioritization strategy given the limited vehicle capacity (e.g., lowest storage first and big orders first). The vehicle route is then computed by solving a traveling salesman problem (TSP) using a commercial solver.

Recent advances in machine learning and simulation-optimization have demonstrated significant potential to improve existing policies. These developments create an opportunity to enhance the decision process outlined by Roldán et al. (2016) by learning effective priority rules and (s,S) inventory control parameters. Consequently, inventory-routing interdependencies can be better captured while preserving the ability to anticipate future demand, leading to more adaptive and effective decisions.

3.2.3 Evolutionary algorithms

Evolutionary algorithms are population-based optimization methods inspired by the concepts of natural selection and genetic operations. In particular, GAs have been widely investigated in computer science and operations research to find near-optimal and robust solutions to complex problems (Alhijawi and Awajan, 2024). Given their usefulness in exploring large search spaces, we find several applications of GAs for the deterministic (e.g., Mahjoob et al., 2022) and stochastic (e.g., Shukla et al., 2013) inventory-routing variants.

Following the emergence of the first GAs, the concept of GP was introduced by Koza (1994). In GP, individuals are represented as trees composed of a set of terminals (e.g., expected demand, unit holding and stockout costs) and mathematical operators (e.g., addition, subtraction, square root). The evolutionary process evolves symbolic expressions or rules, rather than fixed-length representations (i.e., chromosomes) as in GAs. Genetic programming has proven effective in both inventory management (e.g., Lopes et al., 2020) and dynamic VRPs (e.g., Jakobović et al., 2023). Motivated by these findings, we see potential in using GP to develop effective and compact delivery priority rules, capturing non-obvious inventory-routing relationships. Additionally, we recognize the potential of using a GA to calibrate the parameters of the original RPA to the SDIRP setting.

This paper constitutes the first application of GP to the SDIRP. The closest conceptual and methodological parallel to our research is provided by Vonolfen et al. (2013). The authors propose a simulation-based optimization approach to develop resupply and routing policies to minimize total costs and maximize resource usage while maintaining a given service level. The delivery rule includes two rules: selecting the customers to visit and the number of products to deliver. The learning strategy implies optimizing eleven parameters for several scenarios using a GA. The routing subproblem is then solved using a push-forward insertion heuristic.

3.3. Problem definition

In this section, we formally define the problem. First, we provide a general problem description (Section 3.3.1) and an illustrative example (Section 3.3.2). Thereafter, we present an MDP formulation (Section 3.3.3).

3.3.1 Problem description

We approach the SDIRP, following the same setting of Coelho et al. (2014), Roldán et al. (2016) and Greif et al. (2024). The problem is defined on a graph $\mathcal{G} = (\mathcal{N}, \mathcal{E})$, where a single supplier denoted by node 0 is responsible for delivering an homogeneous product to a set of N geographically dispersed customers: $\mathcal{N}' = \mathcal{N} \setminus \{0\} = \{1, ..., N\}$. The nodes are connected by a set of edges $\mathcal{E} = \{(i, j) : i, j \in \mathcal{N}, i \neq j\}$. For modeling purposes, the planning horizon is split into T consecutive discrete periods: $t \in \mathcal{T} = \{0, 1, ..., T-1\}$ and solved accordingly.

The demand faced by customer i in period t is represented by the random variable D_{it}, which becomes known at the end of period t. This demand is stochastic and characterized by an expected value μ_i and variance σ_i^2. At the beginning of the planning horizon ($t = 0$), the supplier is informed about the inventory levels of each customer i (I_{it}). We further assume that the supplier has sufficient inventory to meet all planned deliveries in each period. Each customer i has a maximum storage capacity $\bar{U}_i$ and incurs a unit holding cost h_i per period. The continuous variables q_{it} denote the planned delivery quantity for customer i in period t.

Given the stochastic nature of demand, inventory shortages may occur at the end of each period, after demand realization. Let the continuous variable l_{it} be the unmet demand at customer i in period t, resulting in lost sales, and b_i be the unit shortage cost at customer i. The continuous variables I_{it+1} define the remaining stock of product at customer i at the end of period t.

A vehicle with capacity Q is responsible for completing the planned deliveries, incurring a distance-dependent cost c_{ij} for traversing edge (i, j). Every route starts and ends at the supplier, with the vehicle returning empty. The binary decision variables x_{ijt} indicate whether edge (i, j) is traversed in period t, and y_{it} equals one if customer i is visited in period t. The objective is to minimize the total inventory holding, stockout, and routing costs. Table 3.1 summarizes the notation introduced.

Table 3.1: Notation summary.

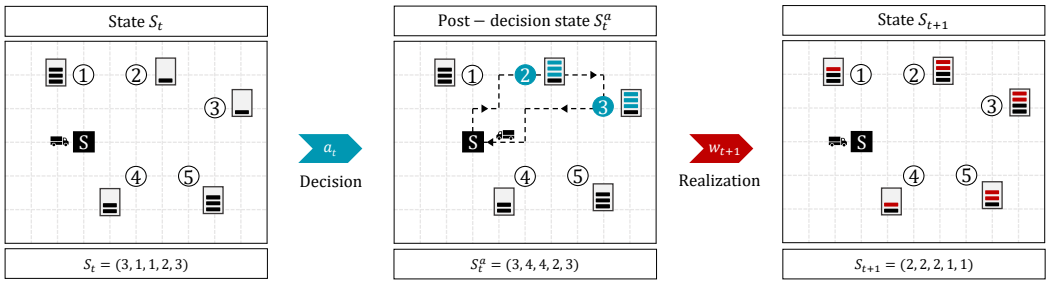
	Sets
$\mathcal{N}$	Set of customers and supplier
$\mathcal{N}'$	Set of customers indexed by $i \in \mathcal{N} \setminus \{0\}$
$\mathcal{E}$	Set of edges (i, j)
$\mathcal{T}$	Planning horizon indexed by t
	Parameters
I_{i0}	Inventory level at customer i at the beginning of the planning horizon
$\bar{U}_i$	Inventory storage capacity at customer i
Q	Vehicle capacity
h_i	Unit holding cost at customer i
b_i	Unit shortage cost at customer i
D_{it}	Demand of customer i in period t
μ_i	Expected demand of customer i
σ_i^2	Variance of demand of customer i
c_{ij}	Cost for traversing edge (i, j)
	Decision variables
x_{ijt}	Equal to 1, if edge $(i, j) \in \mathcal{E}$ is traversed by the vehicle in period t
y_{it}	Equal to 1, if customer i is visited by the vehicle in period t
q_{it}	Delivery quantity for customer i in period t
l_{it}	Lost sales at customer i in period t after the demand is revealed
I_{it+1}	Inventory level at customer i at the end of period t

3.3.2 Illustrative example

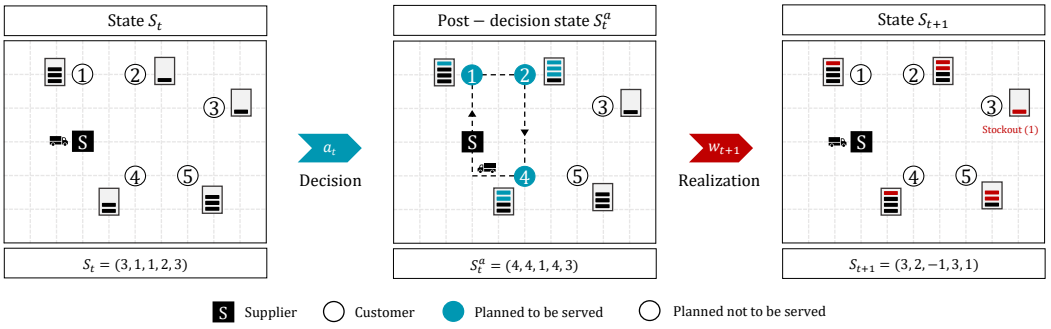
Figure 3.1 presents an illustrative example that supports the MDP formulation described in Section 3.3.3. The example depicts a potential problem state (S_t), two alternative decisions (a_t), the corresponding post-decision states (S_t^a), the stochastic information realization (ω_{t+1}), and the resulting states (S_{t+1}). In both scenarios, we consider a central supplier, indicated by a black square labeled 'S', and a set of five customers. The current inventory levels are shown in the inventory deposits next to each customer. The fleet consists of a single vehicle that can transport six units of product simultaneously. We assume a unit holding cost of one cost unit (CU) for all customers and a unit stockout cost of two CUs.

The first block, on the left, represents the system's initial state, comprising the inventory levels at the beginning of period t, so that $S_t = (3, 1, 1, 2, 3)$. In this case, customers *1* and *5* hold three product units each, customers *2* and *3* have one unit each, and customer *4* holds two units. According to our problem setting, the supplier is always able to meet the planned deliveries in each period.

The central block illustrates two distinct decisions, referred to as scenarios (a) and (b). Each decision a_t involves selecting the customers to visit, determining the corresponding delivery quantities, and planning the vehicle route. The customers to be visited in each scenario are influenced by a delivery priority. In scenario (a), customers are prioritized based on their initial inventory levels. Following an OU inventory replenishment rule and considering the vehicle capacity of six units, only customers *2* and *3* are visited in period t,



(a) Lowest-inventory-first priority.



(b) Distance-to-supplier priority.

Figure 3.1: Example of a potential state, two alternative decisions (based on two different priority rules), corresponding post-decision states, demand realization and next states.

each receiving three units, in that delivery sequence. In scenario (b), the customers to be visited are selected based on their proximity to the supplier. Following the same OU strategy, customers *1*, *2* and *4* are replenished with one, three and two units, respectively. Each alternative decision leads to a different post-decision state S_t^a, illustrated in the second block. These states represent the updated inventory levels after deliveries and before demand is revealed.

The third and final block addresses demand realization, denoted as ω_{t+1}. In both scenarios, the demand vector is $\omega_{t+1} = (-1, -2, -2, -1, -2)$. Scenario (b) results in a stockout of one unit at customer *3*. The updated inventories after demand is revealed at the end of period t define the next state S_{t+1}. Subsequently, the inventory holding, stockout and routing costs are calculated. In scenario (a), the resulting holding costs amount to 8 CUs, with no stockout costs. In contrast, scenario (b) incurs holding costs of 9 CUs and stockout costs of 2 CUs. Assuming each edge (i, j) represents a cost of 1 CU, the routing costs total 14 CUs for scenario (a) and 10 CUs for scenario (b).

3.3.3 Markov decision process

In this section, we formulate the SDIRP as an MDP, a discrete-time stochastic control process and a valuable tool for modeling sequential decision problems in complex and uncertain contexts (Puterman, 1994). A sequence of states S_t, decisions a_t, and demand

realizations ω_{t+1} defines the sequential decision process. The following sections provide formal definitions for each of these components.

3.3.3.1 State (S_t)

A state S_t represents all the information required for decision-making at the beginning of period t. It contains each customer's inventory level, denoted by I_{it}, along with additional contextual information, Φ_t. This information may include both static and dynamic features, such as storage capacities ($\bar{U}_i$); inter-location distances (c_{ij}); unit holding and stockout costs (h_i and b_i); and the H historical demand observations ($\hat{D}_{it} = \{D_{it-H}, ..., D_{it-1}\}$). Then, $\Phi_t = (\bar{U}_i, c_{ij}, h_i, b_i, \hat{D}_{it})_{\ i \in \mathcal{N}', (i,j) \in \mathcal{E}}$. The initial state is therefore defined as $S_0 = (I_0, \Phi_0)$, with $I_0 = (I_{i0})_{\ i \in \mathcal{N}'}$.

3.3.3.2 Decision (a_t)

The set of all possible decisions at state S_t is represented by the decision space $\mathcal{A}(S_t)$. A decision $a_t \in \mathcal{A}(S_t)$ is selected at the beginning of period t, answering: (1) which customers to serve; (2) how much to deliver to each customer; (3) how to combine them into a vehicle route. Therefore, a_t consists of four components linked to inventory and routing decisions. For each customer i, the inventory component is defined by two decision variables: the delivery quantity q_{it}, and the post-delivery inventory I^a_{it}. This relationship is expressed as $I^a_{it} = I_{it} + q_{it}$. The routing component is expressed by y_{it} and x_{ijt}. For simplification purposes, let us consider $y_t = (y_{it})_{\ i \in \mathcal{N}}$, $q_t = (q_{it})_{\ i \in \mathcal{N}'}$, $I^a_t = (I^a_{it})_{\ i \in \mathcal{N}'}$, and $x_t = (x_{ijt})_{\ (i,j) \in \mathcal{E}}$. Accordingly, a decision $a_t = (y_t, q_t, I^a_t, x_t)$ is valid if the following constraints are satisfied:

$$I^a_{it} = I_{it} + q_{it}, \quad \forall\, i \in \mathcal{N}' \tag{3.1}$$

$$I^a_{it} \leq \bar{U}_i, \quad \forall\, i \in \mathcal{N}' \tag{3.2}$$

$$q_{it} \leq Q\, y_{it}, \quad \forall\, i \in \mathcal{N}' \tag{3.3}$$

$$\sum_{i \in \mathcal{N}'} q_{it} \leq Q\, y_{0t} \tag{3.4}$$

$$\sum_{j \in \mathcal{N}: j \neq i} x_{jit} = y_{it}, \quad \forall\, i \in \mathcal{N} \tag{3.5}$$

$$\sum_{j \in \mathcal{N}: j \neq i} x_{ijt} = y_{it}, \quad \forall\, i \in \mathcal{N} \tag{3.6}$$

$$\sum_{j \in \mathcal{N}'} x_{0jt} = \sum_{i \in \mathcal{N}'} x_{i0t} \tag{3.7}$$

$$\sum_{i \in \mathcal{B}} \sum_{j \in \mathcal{B}: j \neq i} x_{ijt} \leq \sum_{i \in \mathcal{B}} y_{it} - y_{\ell t}, \quad \forall \mathcal{B} \subseteq \mathcal{N}', |\mathcal{B}| \geq 2, \ell \in \mathcal{B} \tag{3.8}$$

$$q_{it}, I^a_{it} \geq 0, \quad \forall\, i \in \mathcal{N}' \tag{3.9}$$

$$y_{it} \in \{0,1\}, \quad \forall\, i \in \mathcal{N} \tag{3.10}$$

$$x_{ijt} \in \{0,1\}, \quad \forall\, (i,j) \in \mathcal{E} \tag{3.11}$$

Constraints (3.1) balance the inventory levels at each customer i. Constraints (3.2) ensure that the post-delivery inventory level does not exceed the storage capacity $\bar{U}_i$. Constraints (3.3) link the delivery and visit decision variables. Constraints (3.4) guarantee that the total delivery quantities assigned to the vehicle do not exceed its capacity Q. Constraints (3.5) and (3.6) ensure the flow conservation at each location. Constraints (3.7) ensure that the vehicle begins and ends its route at the supplier. Constraints (3.8) refer to the subtour elimination constraints. Constraints (3.9) - (3.11) ensure the domain of each variable. The combination of S_t and a_t originates the post-decision state $S_t^a = (I_t^a, \Phi_t^a)$.

3.3.3.3 Stochastic information (ω_{t+1}) and transition function (Γ)

In the considered SDIRP, customer demand represents the stochastic information revealed at the end of period t, denoted by ω_{t+1}. Accordingly, we define $D_t^\omega = (D_{it}^\omega)_{\, i \in \mathcal{N}'}$ as an $|N|$-dimensional vector. The transition function Γ governs the evolution of the system from the post-decision state S_t^a to the next state S_{t+1}, based on the demand realization ω_{t+1}, such that $S_{t+1} = \Gamma(S_t^a, \omega_{t+1})$. Both the inventories and the state variables are updated accordingly:

$$I_{it+1} = I_{it}^a - D_{it}^\omega + l_{it}, \quad \forall i \in \mathcal{N}' \tag{3.12}$$

$$l_{it} = -\min(0, I_{it}^a - D_{it}^\omega), \qquad \forall i \in \mathcal{N}' \tag{3.13}$$

After demand is revealed, the following inventory holding, stockout and routing costs result:

$$C(S_t^a, \omega_{t+1}) = \sum_{i \in \mathcal{N}'} h_i \, I_{it+1} + \sum_{i \in \mathcal{N}'} b_i \, l_{it} + \sum_{(i,j) \in \mathcal{E}} c_{ij} \, x_{ijt} \tag{3.14}$$

3.3.3.4 Policy (π)

The solution to the SDIRP is a policy $\pi \in \Pi$ that assigns a decision a_t to any state S_t, in order to minimize the expected inventory holding, stockout and routing costs over a given planning horizon $\mathcal{T}$, given an initial state S_0:

$$\arg\min_{\pi \in \Pi} \mathbb{E}\left[\sum_{t \in \mathcal{T}} C(S_t^a, \omega_{t+1}) \mid S_0 \right] \tag{3.15}$$

3.4. Solution method

In the following, we first introduce the proposed decision policy (Section 3.4.1). We then present the learning strategy behind each policy stage (Section 3.4.2). Finally, we define the fitness function used to evaluate candidate policies during the evolutionary search (Section 3.4.3).

3.4.1 Three-stage decision policy

Integrating inventory and routing decisions in a stochastic dynamic setting yields a challenging optimization problem that becomes computationally intractable as the number of customers, vehicles, products, and planning periods increases. Although integrated direct lookahead approaches can explicitly account for future demand uncertainty, selecting an appropriate lookahead horizon involves a difficult trade-off: longer horizons may better capture future demand but significantly increase computational effort. This motivates the development of a policy that remains computationally tractable while still adapting effectively to the current and future states.

To address this challenge, we propose an adaptive and anticipatory decision policy applied in an RH fashion, leveraging the current state information. Our policy decomposes the decision process into three consecutive stages: (1) computing a delivery priority score for each customer, (2) determining the customers to visit, based on the previous priorities, and computing the corresponding delivery quantities, and (3) sequencing the selected customers into a feasible vehicle route. This decomposition makes the policy suitable for real-time application (see Algorithm 1).

Algorithm 1 Three-stage decision policy.

1: **input:** state $S_t = (I_t, \Phi_t)$, $I_t = (I_{it})_{\,i \in \mathcal{N}'}$, $\Phi_t = (\bar{U}_i, c_{ij}, h_i, b_i, \hat{D}_{it}, Q)_{\,i \in \mathcal{N}', (i,j) \in \mathcal{E}}$, inventory control parameters $(s_i, S_i)_{\,i \in \mathcal{N}'}$
2: **output:** decision $a_t = (y_t, q_t, I_t^a, x_t)$, $y_t = (y_{it})_{\,i \in \mathcal{N}}$, $q_t = (q_{it})$ and $I_t^a = (I_{it}^a)_{\,i \in \mathcal{N}'}$, $x_t = (x_{ijt})_{\,(i,j) \in \mathcal{E}}$
3: $(s_i, S_i) \leftarrow$ RPA (see Algorithm 2) for each customer $i \in \mathcal{N}'$
4: $\mathcal{N}_d \leftarrow \emptyset$, $\mathcal{N}_a \leftarrow \mathcal{N}'$, $q_{it} \leftarrow 0$ and $I_{it}^a \leftarrow I_{it}$ $\forall i \in \mathcal{N}'$, and $y_{it} \leftarrow 0$ $\forall i \in \mathcal{N}$
5: $p_i \leftarrow$ PriorityScore(S_t) for each customer $i \in \mathcal{N}'$
6: **while** $\mathcal{N}_a \neq \emptyset$ and $Q > 0$ **do**
7: $\quad i \leftarrow \arg\max_{i' \in \mathcal{N}_a} p_{i'}$
8: $\quad \mathcal{N}_a \leftarrow \mathcal{N}_a \setminus \{i\}$
9: $\quad$ **if** $I_{it} \leq s_i$ **then**
10: $\quad\quad y_{it} \leftarrow 1$
11: $\quad\quad \mathcal{N}_d \leftarrow \mathcal{N}_d \cup i$
12: $\quad\quad q_{it} \leftarrow \min(S_i - I_{it}, Q)$
13: $\quad\quad Q \leftarrow Q - q_{it}$
14: $\quad\quad I_{it}^a \leftarrow I_{it} + q_{it}$
15: $\quad$ **end if**
16: **end while**
17: $x_t \leftarrow$ SolveTSP$(\mathcal{N}_d)$

The first stage establishes an ordering of customers based on their delivery priorities, given the current state (cf. line 5). This ordering is critical, as routing constraints (i.e., vehicle capacity) affect the feasibility of both visit and inventory replenishment decisions in each period. Formally, a delivery priority rule is a function that maps a customer's current state to a priority score (e.g., the score could be equal to the customer's inventory level). These priority scores do not directly define the final set of customers to be visited; rather, they define the order in which customers are inspected for potential replenishment. The delivery priority rule, therefore, serves as the link between inventory needs and routing efficiency.

The second stage selects the customers to be visited, based on the previously computed priority scores, and the corresponding replenishment quantities. Under quite general conditions, a policy that aims to minimize review, replenishment, holding, and stockout costs under stochastic demand belongs to the (R, s, S) family (Silver et al., 2016). The inventory position of a given customer is reviewed every R periods. If it falls below the reorder point s, the customer is replenished to the target level S. In our problem setting, inventory positions are observed at the beginning of each period, and replenishments take place within the same period. We therefore set the review period as $R = 1$, and assume zero lead time (as in Jemai et al., 2013).

To determine the (s, S) control parameters for each customer, we adopt the RPA of Ehrhardt and Mosier (1984), a computationally efficient algorithm for obtaining approximately optimal values (see Algorithm 2) that proved very accurate across a wide range of demand distributions and parameter settings (e.g., mean demand, lead time, unit stockout and setup cost values).

Algorithm 2 Revised power approximation (RPA).

1: **input:** customer i, setup cost (SC_i), μ_i, σ_i^2, h_i, b_i, $\bar{U}_i$, and $\theta = (\alpha, \beta, \gamma, \alpha_0, \alpha_1, \alpha_2, \alpha_3, \lambda)$
2: **output:** reorder point (s_i), order-up-to-level (S_i)
3: $G_{i,p} \leftarrow \alpha \mu_i^{1-\beta} \left(\frac{SC_i}{h_i}\right)^{\beta} \left[1 + \frac{\sigma_i^2}{\mu_i^2}\right]^{\gamma}$
4: $s_{i,p} \leftarrow \alpha_0 \mu_i + \sigma_i \left(\frac{\alpha_1}{z_i} + \alpha_2 + \alpha_3 z_i\right)$, where $z_i = \sqrt{\frac{G_{i,p} h_i}{\sigma_i b_i}}$
5: **if** $\frac{G_{i,p}}{\mu_i} > \lambda$ **then**
6: $\quad s_i \leftarrow s_{i,p}$ and $S_i \leftarrow s_{i,p} + G_{i,p}$
7: **else**
8: $\quad S_{i,0} \leftarrow \mu_i + k_i \sigma_i$, where $k_i = \Phi^{-1}\left(\frac{b_i}{b_i + h_i}\right)$ and Φ^{-1} is the standard normal quantile function
9: $\quad s_i \leftarrow \min(s_{i,p}, S_{i,0})$ and $S_i \leftarrow \min(s_{i,p} + G_{i,p}, S_{i,0})$
10: **end if**
11: $S_i \leftarrow \min(S_i, \bar{U}_i)$ and $s_i \leftarrow \min(s_i, S_i)$

The approximation first computes an approximate lot-size $G_{i,p}$ (line 3) and a preliminary reorder point $s_{i,p}$ (line 4), from which the final (s_i, S_i) parameters are derived (lines 6-9). The algorithm is governed by a set of parameters $\theta = (\alpha, \beta, \gamma, \alpha_0, \alpha_1, \alpha_2, \alpha_3, \lambda)$, originally obtained by solving a wide range of representative problems to optimality and subsequently fitting the approximation to the resulting optimal solutions, yielding $\theta = (1.30, 0.506, 0.116, 0.973, 0.183, 1.063, -2.192, 1.50)$. Furthermore, a setup (or ordering) cost input is required (line 3), representing a fixed cost for visiting a customer within the network. Since this cost is not directly available, several routing-informed setup cost estimations are considered, including the distance from customer i to the supplier and the average distance from customer i to the δ closest customers (cf. Appendix 3.B).

After determining the customers to be visited and the corresponding delivery quantities, the final stage computes the vehicle route that minimizes the total travel distance, i.e., routing cost. Since our problem setting involves a single vehicle, routing decisions translate into solving a TSP instance. After the inventory and routing decisions are determined, the resulting holding, stockout, and routing costs for the current period are calculated, and the process repeats in an RH fashion until the end of the time horizon.

3.4.2 Learning strategies

Customer prioritization plays a crucial role in the proposed methodology. Standard heuristics often prioritize customers based on current inventory levels (e.g., lowest inventory first, big orders first), mitigating immediate stockout risks but overlooking important routing and geographic factors, such as the proximity between customers and the supplier. To bridge this gap, we propose learning new delivery priority rules that respond effectively to the current state, capturing inventory-routing interdependencies and anticipating future demand realizations. In this first learning step, the RPA is used in its original form. Subsequently, we propose calibrating the original RPA parameters for the best-performing learned priority rule, aiming to better align visit and inventory replenishment decisions with the SDIRP setting.

To support the learning and evaluation of such rules, we develop a simulation-optimization framework that mimics the stochastic dynamic evolution of demand over a multi-period horizon. The simulator receives a set of problem instances as input, each containing a given number of customers and periods, demand realizations, and further instance and customer-dependent information. Each candidate rule is evaluated only when embedded within the full three-stage policy.

During training, routing decisions are approximated using a constructive nearest-neighbor heuristic enhanced with 2-opt local search, yielding high-quality solutions at low computational cost and allowing candidate policies to be evaluated efficiently (cf. Appendix 3.A). Each candidate policy is applied at the beginning of each period, after which demand is realized and the resulting inventory holding, stockout, and routing costs are computed. Figure 3.2 illustrates the offline training procedure used to evaluate and evolve candidate three-stage policies.

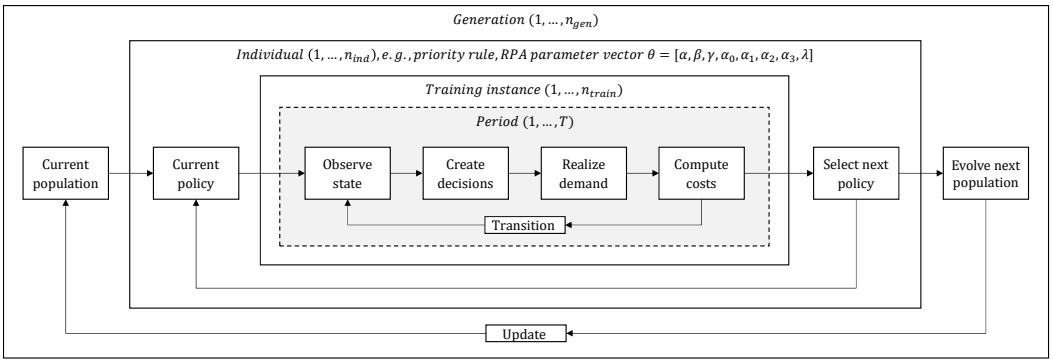


Figure 3.2: Training procedure of the proposed three-stage policy.

3.4.2.1 Learning priority rules

Genetic Programming is particularly well-suited to discover rules that capture complex, non-obvious inventory-routing relationships. In GP, candidate rules are represented as symbolic expressions, typically structured as trees, formed by combining a predefined set of terminals (features) with mathematical operators (see Table 3.2). The evolution-

ary process starts from an initial population of n_{ind} candidate rules. High-performing rules are selected and recombined over successive generations using genetic operations such as crossover and mutation, generating progressively improved rules until a predefined number of generations n_{gen} is reached. Each candidate rule is evaluated using a fitness function (Section 3.4.3). The terminal set integrates both inventory and routing-related information, allowing the learned rules to capture both inventory urgency and routing efficiency.

Table 3.2: Terminal and operator sets.

Category	Terminal	Description	Nature
Inventory terminals	pLS_{it}	Predicted lost sales	Product units
	$nPTS_{it}$	Expected periods to stockout	Time periods
	$nPSLD_{it}$	Periods since last delivery	Time periods
	UR_{it}	Inventory utilization rate	Ratio
	CR_{it}	Inventory coverage rate	Ratio
	h_i	Unit holding cost	Cost
	b_i	Unit shortage cost	Cost
	μ_i	Expected demand	Product units
	σ_i	Demand standard deviation	Product units
Routing terminals	VC	Vehicle capacity	Product units
	dTd_i	Distance to depot	Cost
	$Isol_i$	Isolation from other customers	Cost
	$IsolPC_{it}$	Isolation from priority customers	Cost
	SC_i	Setup cost	Cost
	nC_i	Number of customers for which i is the nearest neighbor	Count
	$minD_i$	Minimum distance	Cost
	$maxD_i$	Maximum distance	Cost
Operators	$+$	Addition	
	$-$	Subtraction	
	$\times$	Multiplication	
	$\div$	Protected division	
	$-x$	Negation	
	x^{-1}	Inverse	
	x^2	Square	
	$\sqrt{}$	Square root	

Inventory-related terminals include the predicted lost sales (pLS_{it}), which estimate the future unmet demand if customer i is not visited in period t, as well as the expected number of periods to stockout ($nPTS_{it}$) and the number of periods since the last delivery ($nPSLD_{it}$). The current inventory level is assessed through a utilization rate (UR_{it}) and a coverage rate (CR_{it}), representing the ratio of stock to storage capacity and to expected demand, respectively. The unit holding cost (h_i) and unit shortage cost (b_i) are preserved in their original form, and the stochastic demand is represented by its expected value (μ_i) and standard deviation (σ_i).

Routing-related terminals include the vehicle capacity (VC), along with distance-based metrics such as the distance from each customer i to the depot (dTd_i), isolation (i.e., average distance) from the remaining customers ($Isol_i$), and isolation from customers whose

current inventory levels are below their respective expected demands ($IsolPC_i$), i.e., inventory-priority customers. Additional terminals include a setup cost for each customer i (SC_i), the number of customers for which customer i is the nearest neighbor (nC_i), as well as the distance to its closest customer ($minD_i$) and to its farthest customer ($maxD_i$).

3.4.2.2 Calibrating the RPA parameters

The original (s,S) inventory replenishment rule is obtained using the RPA of Ehrhardt and Mosier (1984). After identifying the best-performing non-dominated delivery priority rule, we tune the original RPA parameters for that priority rule to better align (s,S) values with the SDIRP setting. In particular, we use a GA to calibrate the following parameter vector (used in Algorithm 2):

$$\theta = (\alpha,\beta,\gamma,\alpha_0,\alpha_1,\alpha_2,\alpha_3,\lambda) \tag{3.16}$$

3.4.3 Fitness function

During the learning process, the performance of each candidate policy is evaluated by applying it to a set of training instances, denoted as n_{train}. Each instance is characterized by a given number of customers and periods, unit holding and stockout costs, and vehicle capacity, resulting in a set of instance scenarios $\mathcal{M} = \{1,...,M\}$. The resulting cost components, used as negative rewards during policy learning, must be normalized to ensure comparability across training instances, as their magnitudes can vary substantially between scenarios.

For each scenario $m \in \mathcal{M}$, a benchmark policy is simulated on the training set to compute the reference cost $C_{\text{bench},m}$. Each candidate policy is then evaluated on the same training set, yielding a total cost $C_{\text{policy},m}$. The reference policy follows the same three-stage decision framework, using the lowest inventory first as the priority rule, the original RPA for computing (s,S) values, and the proposed nearest-neighbor routing heuristic. To obtain the total cost that aggregates the performance of a given candidate policy across all M scenarios, we compute the following geometric mean:

$$C_{mean} = \left(\prod_{m=1}^{M} \frac{C_{\text{policy},m}}{C_{\text{bench},m}} \right)^{1/M} \tag{3.17}$$

The cost function (3.17) defines the performance measure used in both learning experiments (i.e., priority rule learning and RPA calibration).

Furthermore, our GP algorithm controls the complexity of each priority rule by limiting tree depth and penalizing the size of symbolic expressions, i.e., the total number of terminals and operators, via a penalty term κ to discourage too complex rules and improve generalization. Accordingly, for GP-evolved priority rules, the fitness function f is defined as:

$$f = C_{mean} + |ind| \cdot (g+1) \cdot \kappa \cdot \left(\frac{1000}{n_{gen}} \right) \tag{3.18}$$

The term $|ind|$ represents the size of the priority rule under evaluation. The variable g denotes the current generation, and n_{gen} the total number of generations. Accordingly, the term $(g+1)$ increases the penalty's significance as the evolutionary process progresses, encouraging the exploration of complex solutions in earlier generations while promoting convergence in later stages. The scaling factor $1000/n_{gen}$ keeps the penalty magnitude consistent across experiments with different numbers of generations (i.e., $n_{gen} \in \{500, 1000\}$).

3.5. Experimental setting

This section offers an overview of our experimental setup. We first describe the instance design, including the layout, demand and supply, vehicle capacity and costs (Section 3.5.1). Second, we present the performance metrics used to evaluate a given policy (Section 3.5.2). Third, we outline the training configurations for priority rule learning and RPA calibration (Section 3.5.3). Fourth, we introduce the benchmark policies (Section 3.5.4). Finally, we describe implementation details (Section 3.5.5).

3.5.1 Instance setting

Our instance design follows Coelho et al. (2014), originally adapted from the deterministic and static setting introduced by Archetti et al. (2007). We further extend it to different problem settings, thereby covering a broader range of inventory-routing trade-offs inherent in the SDIRP. In the following, we provide a description of our instance setting, including the layout, demand and supply, vehicle capacity, and related costs. All details are provided in Table 3.3.

Layout. Each problem instance involves a single supplier that periodically replenishes a set of geographically distributed customers, with $N \in \{5, 10, 25, 50\}$, over a planning horizon of $T \in \{10, 20\}$ periods. The location of each node is defined by its respective geographic coordinates (x, y). These coordinates are generated following a uniform distribution within the range $[0, 500]$, aligned with Coelho et al. (2014).

Demand and supply. Each customer i experiences periodic, stochastic demand for a single product. Demand realizations are generated as integer random numbers following a normal distribution, $D_{it} \sim \mathcal{N}(\mu_i, \sigma_i^2)$, where $\mu_i \sim \mathcal{U}(10, 100)$ and $\sigma_i \in \{0.1, 0.2, 0.3\} \cdot \mu_i$. Accordingly, different customers within the same problem instance may be assigned different uncertainty levels. Initial inventory levels are generated as integer random numbers following a uniform distribution, $I_{i0} \sim \mathcal{U}(\mu_i, \bar{U}_i - \mu_i)$, ensuring at least one period of expected demand coverage. Each customer's storage capacity is set proportional to the mean demand: $\bar{U}_i = f \cdot \mu_i$. While Coelho et al. (2014) focus on inventory capacity factors $f \in \{2, 3, 4\}$, which tend to favor an OU replenishment strategy, we extend this range to $f \in \{3, 4, 5, 6\}$ to investigate a wider set of operating conditions. This allows us to assess the robustness and adaptability of our policies while capturing additional trade-offs among holding, stockout, and routing costs.

Vehicle. Vehicle capacity is defined based on the expected demand volumes: $Q = vf \cdot \sum_{i \in \mathcal{N}'} \mu_i$, where vf is a scaling factor. In this study, we extend the setting proposed by Coelho et al. (2014), i.e., $vf = 1.50$, by introducing two additional factors, $vf \in \{1.25, 1.75\}$.

Costs. Unit holding costs are drawn from a continuous uniform distribution: $h_i \sim hf \cdot \mathcal{U}(0.02, 0.10)$, and unit stockout costs are defined as $s_i = sf \cdot h_i$. The initial configuration, $(hf, sf) = (1, 200)$, is aligned with Coelho et al. (2014) and implies a service level of 99.5%. We further extend this configuration to $(hf, sf) = (10, 20)$, corresponding to a service level of 95.2%, so that $(hf, sf) \in \{(1, 200), (10, 20)\}$. Routing costs are determined using the Euclidean distances between nodes.

To obtain a balanced experimental design, we generate multiple independent instances for each scenario using different random seeds. For the test set, ten instances are generated for each of the 48 scenarios (obtained by combining the different parameter values in a full factorial design), yielding $n_{test} = 480$ instances. The validation set follows the same 48-scenario structure, using different random seeds, yielding $n_{val} = 480$. The training set follows the same scenario structure but uses different random seeds and five instances per scenario, yielding $n_{train} = 240$.

Table 3.3: Training, validation and test instance settings.

Hierarchy	Parameter	Value
Instance-level	Customers	$N \in \{5, 10, 25, 50\}$
	Planning horizon	$T \in \{10, 20\}$
	Unit (holding, stockout) cost factor	$(hf, sf) \in \{(1, 200), (10, 20)\}$
	Vehicle capacity factor	$vf \in \{1.25, 1.50, 1.75\}$
	Vehicle capacity	$Q = vf \cdot \sum_{i \in \mathcal{N}'} \mu_i$
Customer-level $\left\{ \begin{smallmatrix} i \in \mathcal{N}' \\ (i,j) \in \mathcal{E} \end{smallmatrix} \right\}$	Coordinates	$(x_i, y_i) \sim \mathcal{U}(0, 500)$
	Demand $\sim \mathcal{N}(\mu_i, \sigma_i^2)$	$\mu_i \sim \mathcal{U}(10, 100)$
		$\sigma_i \sim \text{DiscreteUniform}(\{0.1, 0.2, 0.3\}) \cdot \mu_i$
	Initial inventory	$I_{i0} \sim \mathcal{U}(\mu_i, \bar{U}_i - \mu_i)$
	Inventory capacity	$\bar{U}_i \sim \text{DiscreteUniform}(\{3, 4, 5, 6\}) \cdot \mu_i$
	Unit holding cost	$h_i \sim hf \cdot \mathcal{U}(0.02, 0.10)$
	Unit stockout cost	$b_i = sf \cdot h_i$
	Routing cost	$c_{ij} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} + 0.5$

3.5.2 Performance metrics

In line with the fitness function introduced in Section 3.4.3, we use C_{mean} to evaluate policy performance across all scenarios ($m \in \mathcal{M}$) in terms of total cost. Nevertheless, it is important to note that some policies may present strong average performance yet underperform in particular scenarios. To address this, we also report the worst-case performance of each policy across all scenarios, represented as:

$$C_{max} = \max_{m \in \mathcal{M}} \left(\frac{C_{\text{policy},m}}{C_{\text{bench},m}} \right) \tag{3.19}$$

Additionally, we introduce a metric that tells the number of test instances (W) in which a given policy outperforms the reference one. Consequently, the win rate WR is defined as:

$$WR = \frac{W}{n_{test}} \tag{3.20}$$

3.5.3 Training configurations

In the following, we outline the parameterization of the evolutionary algorithms used to learn effective priority rules (GP) and to calibrate the RPA (GA). To account for the random nature of both algorithms, each training configuration is replicated over 10 independent runs.

Learning priority rules. The GP configuration used to learn our priority rules is summarized in Table 3.4. This setup results in 72 training configurations, corresponding to 720 runs. The GP algorithm uses the terminal set specified in Table 3.2, while considering four different operator sets to assess their impact on rule complexity and policy performance.

Table 3.4: Parameterization of the GP algorithm.

Parameter	Value
Number of runs	720
Population, Generations	(300,1000), (600,500)
Initialization	Ramped Half-and-Half
Mutation, Crossover rates (%)	(20,80), (50,50), (80,20)
Selection method	Tournament (Size = 7)
Max depth	5, 17
Penalty values	0, 3E-06, 1E-05
Breed elite	4
Operator set (1)	+, -, neg
Operator set (2)	x, /, inv
Operator set (3)	+, -, neg, x, /, inv
Operator set (4)	x, /, inv, quad, rquad

Calibrating the RPA parameters. Following the previous experiment, the best-performing priority rule is selected, and the original parameters of RPA are calibrated. The parameter vector $\theta = (\alpha, \beta, \gamma, \alpha_0, \alpha_1, \alpha_2, \alpha_3, \lambda)$ is tuned over the search space $(\alpha, \beta, \alpha_0, \lambda) \in [0, 5]$ and $(\gamma, \alpha_1, \alpha_2, \alpha_3) \in [-5, 5]$, providing sufficient flexibility around the original parameter values. In total, 120 independent runs are performed to ensure statistical robustness. The GA configuration is reported in Table 3.5.

3.5.4 Benchmarks

To assess the performance and robustness of the proposed policy, we compare it against several benchmark approaches. The first group follows the same three-stage structure as the proposed policy but uses alternative priority and inventory replenishment rules adapted

Table 3.5: Parameterization of the GA algorithm.

Parameter	Value
Number of runs	120
Population, Generations	(300,1000), (600,500)
Initialization	Random initialization
Mutation rate (%)	20, 50, 80
Mutation operator	Reset
Crossover operator	One-point, Two-point
Selection op.	Tournament (Size = 7)
Breed elite	4

from the literature (Roldán et al., 2016). This allows us to isolate the value of learning the priority rule and calibrating the RPA to the SDIRP setting.

LSF-OU, BOF-OU & LIF-OU. These three-stage benchmarks combine an OU inventory replenishment strategy with three standard priority rules, the first two from Roldán et al. (2016): lowest storage first **(LSF)**, big orders first **(BOF)** and lowest inventory first **(LIF)**. **LSF** prioritizes customers with low storage capacities, **BOF** prioritizes customers according to the volume of their planned deliveries, and **LIF** prioritizes customers with low inventory levels. In both cases, vehicle routes are solved to optimality using a commercial solver.

LSF-($\alpha\bar{U}_i, \bar{U}_i$), BOF-($\alpha\bar{U}_i, \bar{U}_i$) & LIF-($\alpha\bar{U}_i, \bar{U}_i$). These three-stage benchmarks combine an (s, S) inventory rule with the **LSF**, **BOF** and **LIF** priority rules. In this setting, the order-up-to-level (S) corresponds to the customer's storage capacity $\bar{U}_i$. We investigate three values for the reorder point (s), defined as percentages of the customer's storage capacity: $\alpha \in \{0.25, 0.50, 0.75\}$. Vehicle routes are solved to optimality using a commercial solver.

A second group consists of integrated inventory-routing optimization benchmarks with different anticipatory capabilities, namely a myopic and a DLA policy.

MYOPIC. This benchmark solves an integrated IRP at the beginning of each period without explicitly accounting for future demand realizations. Given the current state, inventory-routing decisions are jointly optimized using a commercial solver, minimizing the current-period costs. Demand is approximated as $\hat{D}_{it} = \mu_i + k_i\sigma_i$, where k_i is the safety factor. This benchmark allows us to compare our proposed three-stage policy against an exact but non-anticipatory optimization approach.

DLA. This benchmark represents a direct lookahead approximation **(DLA)** policy. We consider two variants, $\mathbf{DLA}_{l=1}$ and $\mathbf{DLA}_{l=2}$, with one and two lookahead periods (plus the current period), respectively. At the beginning of each period, inventory-routing decisions are jointly optimized over the current and future lookahead periods in an RH fashion using a commercial solver. The objective is to minimize the total expected cost over the optimization horizon. Demand is approximated as $\hat{D}_{it} = \mu_i + k_i\sigma_i$.

Finally, we include a perfect-information benchmark for instances with 5 customers and 10 periods, given the problem's NP-hard nature.

PH. The perfect-hindsight **(PH)** benchmark solves the entire planning horizon to optimality for instances with 5 customers and 10 periods. All future demand realizations are known in advance, and inventory-routing decisions are jointly optimized over the full planning horizon. To mitigate end-of-horizon effects, we impose that, on average, the inventory levels at the end of the horizon should at least match the initial ones.

3.5.5 Implementation details

The simulation environment was developed in Java 17.0.1 to replicate the stochastic dynamic problem setting, and the ECJ library was used to support the GP and GA algorithms. During the training and validation stages, vehicle routes are computed using a constructive nearest-neighbor heuristic enhanced with 2-opt local search (details are provided in Appendix 3.A). During testing, vehicle routes are solved to optimality using Gurobi Optimizer 10.0.2 for all three-stage policies. For **MYOPIC**, **DLA** and **PH** benchmarks, the integrated inventory-routing optimization problems are also solved using Gurobi, with a 2-hour time limit per planning period (optimality gaps are reported in Appendix 3.E). Each run was executed on a single core with 2 GB of RAM. The nodes are equipped with Intel E5-2450 CPUs and run a Linux-based operating system. Across all 720 GP runs, the average training time was 12.6 hours, with a standard deviation of 3.3 hours. Across all 120 GA runs, the average training time was 14.0 hours, with a standard deviation of 2.2 hours.

3.6. Computational results

In this section, we evaluate the performance of the proposed policy and the corresponding benchmarks on the test set. First, we show the value of the RPA for computing the control parameters of our (s,S) inventory replenishment rule, compared with standard inventory replenishment strategies (Section 3.6.1). Second, we investigate the benefits of learning the priority rule and further calibrating the original RPA parameters (Section 3.6.2). Third, we compare each policy performance against the perfect hindsight (PH) benchmark (Section 3.6.3). Finally, we present a scenario-level analysis to understand the relative performance of the policies in different settings (Section 3.6.4).

3.6.1 Improving inventory replenishment

Table 3.6 reports three groups of metrics: total cost performance, operational behavior, and individual cost components. All metrics use **MYOPIC** as the reference baseline. The first group summarizes total cost performance using C_{mean}, C_{max} and WR (in %). The operational behavior and cost component metrics directly compare each policy $\boldsymbol{\pi}$ with **MYOPIC** in terms of the total number of customer visits, delivered quantities, lost sales, holding, stockout, and routing costs, with each metric X computed by aggregating over all scenarios $m \in \mathcal{M}$ as $\sum_m X_{\pi,m} / \sum_m X_{MYOPIC,m}$. Given their superior overall performance, only the

LIF-based variants are reported in the table. The same applies to the $(\mathbf{0.25\bar{U}_i, \bar{U}_i})$ replenishment rule relative to the alternatives. The overall performance of all variants is presented in Appendix 3.C.

Table 3.6: Overall performance per policy (**MYOPIC** as the baseline).

Policy (π)	Total cost performance			Operational behavior			Cost component		
	C_{mean}	C_{max}	WR	Number of visits	Delivery qty.	Stockout qty.	Holding	Stockout	Routing
LIF-OU	1.52	2.48	0%	1.09	1.25	0.00	7.79	**0.00**	1.11
LIF-$(0.25\bar{U}_i, \bar{U}_i)$	0.90	1.56	67%	0.30	1.13	1.35	4.57	1.73	**0.62**
LIF-RPA	0.81	1.05	85%	0.40	1.08	1.08	2.36	1.33	0.67
DLA	**0.80**	**0.96**	**99%**	0.66	1.06	0.35	**2.13**	0.36	0.69

Order-up-to-level (OU) policies are primarily designed to mitigate the risk of stockouts. This conservative replenishment strategy results in higher delivery quantities and, consequently, higher inventory levels at customer locations, thereby increasing holding costs. Since no reorder point is considered, the frequency of customer visits is the highest among all benchmarks, leading to a significant increase in routing costs. Although inventory shortages are null in **LIF-OU**, the combined high holding and routing costs make OU policies the worst-performing ones, with $C_{mean} = 1.52$.

The integration of a reorder point within the previous OU replenishment strategy results in a better balance between inventory and routing costs, with the effectiveness of **LIF-$(\boldsymbol{\alpha\bar{U}_i, \bar{U}_i})$** depending on the reorder point value. As the reorder point decreases, the overall performance of these three-stage benchmarks tends to improve, with the priority rule having a greater impact (cf. Appendix 3.C). For example, **LIF-$(\mathbf{0.25\bar{U}_i, \bar{U}_i})$** results in $C_{mean} = 0.90$, outperforming the **MYOPIC** (integrated) baseline. Therefore, lower reorder points lead to reduced holding and routing costs (i.e., fewer customer visits) but to more stockouts, with $\alpha = 0.25$ providing a good compromise.

The value of the RPA in calculating (s, S) control parameters is established by **LIF-RPA** with $C_{mean} = 0.81$ and $WR = 85\%$. The results show that the RPA provides a significant improvement over both OU and $(0.25\bar{U}_i, \bar{U}_i)$ inventory replenishment rules, reducing C_{mean} by 46.7% and 10.0%, respectively (using LIF as the priority rule).

Finally, **DLA** shows a significant improvement over **MYOPIC**, with $C_{mean} = 0.80$ and $WR = 99\%$, highlighting the value of the lookahead model in anticipating future demand, with stockout and routing costs reduced by 64% and 31%, respectively, although holding costs were 2.13 times higher. Among the two lookahead horizons considered ($l = 1$ and $l = 2$ periods), the one-period configuration yields a slightly lower C_{mean} (0.796 vs. 0.799) while requiring substantially less computation time and yielding smaller optimality gaps, particularly for large-scale instances (cf. Appendix 3.E). The one-period configuration is therefore referred to as **DLA** thereafter. Although it outperforms all the remaining three-stage benchmarks, **DLA** yields only a small improvement over **LIF-RPA**, reducing C_{mean} by 1.2% but requiring approximately four orders of magnitude more computation time (more details in Appendix 3.D).

3.6.2 Learning priority rules and calibrating the RPA parameters

This section investigates the impact of learning new delivery priority rules via GP. The analysis isolates the effect of each priority rule (GP_i) by comparing several **GP_i-RPA** policies with **LIF-RPA**, while keeping the inventory replenishment and routing stages unchanged. Table 3.7 reports the non-dominated GP rules selected on the validation set after the training stage. The table also presents the size of each rule, together with the total cost performance metrics, i.e., C_{mean}, C_{max} and WR, using **LIF-RPA** as the reference baseline. We focus on the non-dominated rules with a size up to 15, since larger rules tend to be too complex and may overfit the training set, without providing significant performance benefits.

Table 3.7: Trade-off between rule's complexity and total cost performance (**LIF-RPA** as the baseline).

Policy	Priority rule (GP_i)	Size	C_{mean}	C_{max}	WR
GP_1-RPA	$pLS \cdot b$	3	0.997	1.03	33%
GP_2-RPA	$h^2 \cdot pLS$	4	0.995	1.03	34%
GP_3-RPA	$h - h \cdot CR$	5	0.994	1.02	48%
GP_4-RPA	$\frac{\frac{h}{CR}}{IsolPC - nPTS}$	7	0.991	1.02	45%
GP_5-RPA	$\frac{h}{IsolPC \cdot CR - \sigma + pLS}$	9	0.986	**1.01**	51%
GP_6-RPA	$\frac{\left(\frac{1}{IsolPC - h} + pLS\right) \cdot b}{\mu}$	10	0.985	**1.01**	50%
GP_7-RPA	$\left(h \cdot pLS - \frac{1}{nC - IsolPC}\right) \cdot \frac{1}{\mu}$	11	0.986	1.02	49%
GP_8-RPA	$\frac{1}{IsolPC - \frac{\mu}{dTd}} + b \cdot \frac{pLS}{\mu}$	12	0.985	1.02	51%
GP_9-RPA	$\left((nPTS + pLS) \cdot b - dTd \cdot \frac{1}{IsolPC}\right) \cdot \frac{1}{\mu}$	13	0.985	**1.01**	51%
GP_{10}-RPA	$\frac{1}{IsolPC - \frac{\mu}{dTd}} + CR + b \cdot \frac{pLS}{\mu}$	14	0.983	1.02	51%
GP_{10}-tRPA	$\frac{1}{IsolPC - \frac{\mu}{dTd}} + CR + b \cdot \frac{pLS}{\mu}$	14	**0.974**	1.02	**71%**

The performance metrics reported in the table suggest a trade-off between interpretability and solution quality. Larger, more complex rules tend to better capture inventory-routing interdependencies; however, the resulting performance gains are limited, with simpler rules still achieving competitive results. For example, GP_1 prioritizes customers with higher expected stockout costs, with **GP_1-RPA** obtaining a similar overall performance to **LIF-RPA** (C_{mean} = 0.997). In contrast, the best-performing learned rules, namely GP_{10}, capture richer inventory-routing relationships and therefore provide a more effective ordering

of customers for the subsequent inventory replenishment stage. For instance, **GP_{10}-RPA** achieves $C_{mean} = 0.983$ and $WR = 51\%$.

After selecting GP_{10} as the best-performing non-dominated priority rule, we evaluate the benefits of calibrating the RPA parameters via GA, yielding **GP_{10}-tRPA**. Accordingly, **GP_{10}-tRPA** provides a further improvement, though limited, achieving $C_{mean} = 0.974$. In short, the RPA calibration complements the priority rule learning step, though the larger improvement comes from replacing LIF with GP_{10} rather than RPA with tRPA. Comparing to the best policy reported in the literature (**LSF-($0.25\bar{U}_i, \bar{U}_i$)**), our **GP_{10}-tRPA** policy achieves an average improvement of approximately 14% (cf. Appendix 3.C).

3.6.3 Perfect-hindsight comparison

For small-scale instances, i.e., $N = 5$ and $T = 10$, we also compare each policy's performance with a PH benchmark that assumes full knowledge of future demand realizations for the entire planning horizon ($C_{mean} = 0.60$ over **MYOPIC**). According to Table 3.8, **GP_{10}-RPA** and **GP_{10}-tRPA** are the closest policies to **PH**. Of the potential improvement of 0.40, **GP_{10}-tRPA** captures 0.28, i.e., 70% of the potential improvement. The remaining gap highlights the value of complete demand information and joint inventory-routing optimization.

Table 3.8: Total cost performance for $N = 5$ and $T = 10$ (**MYOPIC** as the baseline).

Policy	C_{mean}	C_{max}	WR
DLA	0.79	0.87	**100%**
LIF-RPA	0.75	0.85	98%
GP_{10}-RPA	0.72	0.84	98%
GP_{10}-tRPA	0.72	0.84	98%
PH	**0.60**	**0.73**	**100%**

3.6.4 Scenario-level analysis

This section assesses how each policy performs across different scenario configurations, including the number of customers, the length of the planning horizon, unit holding and stockout costs, and vehicle capacity. Table 3.9 reports the performance of **LIF-RPA**, **GP_{10}-RPA**, **GP_{10}-tRPA** and **DLA** for each scenario group with C_{mean} as the primary metric, using **MYOPIC** as the reference baseline.

Overall, **GP_{10}-tRPA** achieves the best performance among all benchmarks, with $C_{mean} = 0.78$, improving upon **LIF-RPA** ($C_{mean} = 0.81$), **GP_{10}-RPA** ($C_{mean} = 0.79$), and **DLA** ($C_{mean} = 0.80$). Although these improvements may seem subtle, their significance varies across experimental scenario configurations. The main findings are summarized as follows:

Number of customers. The advantage of **GP_{10}-RPA** over **LIF-RPA** is more pronounced for instances with fewer customers, i.e., $N \in \{5, 10\}$. Since the average distance between customers is greater in those scenarios, the decision to visit or not visit a given customer becomes even more important to the final incurred costs. Therefore, looking exclusively

Table 3.9: Total cost performance (C_{mean}) across scenario groups (**MYOPIC** as the baseline).

Scenario group	Configuration	Policy			
		LIF-RPA	GP_{10}-RPA	**GP_{10}-tRPA**	DLA
Customers	$N = 5$	0.74	**0.71**	**0.71**	0.78
	$N = 10$	0.79	**0.77**	**0.77**	0.78
	$N = 25$	0.84	0.83	0.81	**0.80**
	$N = 50$	0.87	0.86	0.84	**0.83**
Planning horizon	$T = 10$	0.81	**0.79**	**0.79**	0.81
	$T = 20$	0.80	0.79	**0.78**	**0.78**
Holding, stockout costs	$(hf, sf) = (1, 200)$	0.70	**0.68**	**0.68**	0.73
	$(hf, sf) = (10, 20)$	0.93	0.92	0.91	**0.87**
Vehicle capacity	$vf = 1.25$	0.83	0.82	**0.81**	**0.81**
	$vf = 1.50$	0.80	0.79	**0.78**	0.80
	$vf = 1.75$	0.79	0.77	**0.76**	0.78
Overall performance		0.81	0.79	**0.78**	0.80

to the inventory component when prioritizing customers is not ideal. In contrast, as the number of customers for the same service area increases, customer density also increases, and the performance of **GP_{10}-RPA** converges towards **LIF-RPA**.
Regarding **DLA**, the table shows that it outperforms **GP_{10}-tRPA** for $N \in \{25, 50\}$. This suggests that the more flexible and integrated structure of **DLA** (which overall leads to more visits than any of the three-stage policies in the table) becomes particularly relevant in those scenarios, unlike the three-stage policies, where the condition for a customer to be visited is that its current inventory is below a reorder point.
Finally, the gaps for **MYOPIC** tend to decrease as the number of customers increases, reflecting the same integrated-decision effect observed for **DLA**.

Planning horizon. The impact of the number of periods in the overall policy performance is limited, being more evident in **DLA**, whose performance improves from $C_{mean} = 0.81$ for $T = 10$ to $C_{mean} = 0.78$ for $T = 20$, reaching **GP_{10}-tRPA** in the longer-horizon setting. This is consistent with the nature of direct lookahead policies, since the effect of inventory-routing decisions can propagate over a larger number of future states when the planning horizon is longer.

Holding and stockout costs. The comparison of the two cost settings reveals a difference in the performance of each policy. For $(hf, sf) = (1, 200)$, the considered three-stage policies in the table clearly outperform **DLA**, with both **GP_{10}-RPA** and **GP_{10}-tRPA** achieving $C_{mean} = 0.68$, compared with $C_{mean} = 0.73$ for **DLA**.
In contrast, for $(hf, sf) = (10, 20)$, the unit holding cost increases tenfold while the resulting unit stockout cost remains unchanged. Accordingly, the advantage of the three-stage policies over **DLA** decreases, and **DLA** gets the best performance, with $C_{mean} = 0.87$, while **GP_{10}-tRPA** achieves $C_{mean} = 0.91$. Finally, the gap to **MYOPIC** also becomes smaller for $(hf, sf) = (10, 20)$.

Vehicle capacity. Although the differences in policy performance are not as evident as those observed for the other parameters, a consistent trend emerges: the gap to **MYOPIC** increases with vehicle capacity. This reveals that both three-stage and **DLA** policies exploit the extra vehicle capacity more effectively than **MYOPIC** when determining which customers to visit and how much to deliver.

3.7. Conclusion

This paper addresses the SDIRP, where a single supplier replenishes a set of geographically dispersed customers with a homogeneous product over a multi-period horizon, using a vehicle with limited capacity. We propose a three-stage policy, applied on an RH basis, that decomposes the inventory-routing decision process into customer delivery prioritization, inventory replenishment, and routing optimization.

First, a delivery priority rule learned via GP computes a priority score for each customer, providing better guidance for the subsequent inventory replenishment stage than standard heuristics in the literature, while being compact and interpretable. Second, an (s, S) inventory replenishment rule, whose control parameters are computed using an adaptation of the RPA of Ehrhardt and Mosier (1984) and further calibrated to the SDIRP setting using a GA, determines which customers to visit and corresponding delivery quantities. To the best of our knowledge, we are the first to transfer the RPA and GP concepts to dynamic inventory-routing. Third, a TSP is solved to optimality using a mathematical solver to compute a feasible route.

Our computational study shows that the proposed policy (**GP_{10}-tRPA**) achieves an effective balance between cost performance and computational efficiency, outperforming the best three-stage benchmark in the literature (**LSF-($0.25\bar{U}_i, \bar{U}_i$)**) and the direct lookahead approximation (**DLA**) by 14% and 2.5%, respectively, while requiring approximately four orders of magnitude less computation time than **DLA**. The scenario-level analysis further shows that our policy performs particularly well in settings where routing decisions have a greater impact on the final incurred costs, in particular for instances with up to ten customers. In contrast, **DLA** becomes more competitive in larger instances ($N \in \{25, 50\}$) and when the inventory cost component becomes more significant.

To conclude, we point out several promising avenues for future research. First, there is an opportunity to extend the proposed policy to broader operational settings, including multi-vehicle fleets, multi-item scenarios, and multi-supplier networks. Second, we see potential in exploring more integrated learning strategies to jointly learn the priority rule and calibrate the RPA parameters, rather than sequentially. Finally, we recognize the potential to investigate additional policy classes to develop effective hybrid policies that combine both the predictive power of DLAs and the computational efficiency of policy and cost function approximations, offering a promising direction to improve solution quality while preserving low running times.

Acknowledgements

This work was supported by FCT - Fundação para a Ciência e Tecnologia [reference: 2022.13466.BD and DOI identifier https://doi.org/10.54499/2022.13466.BD]; the European Union's Horizon 2020 Research and Innovation Programme [TRUST-AI research project, grant number 952060]; the European Union's Horizon Europe Research and Innovation Programme [PEER research project, grant number 101120406].

Bibliography

Achamrah, F.E., Riane, F., Limbourg, S., 2022. Solving inventory routing with transshipment and substitution under dynamic and stochastic demands using genetic algorithm and deep reinforcement learning. International Journal of Production Research 60, 6187–6204. https://doi.org/10.1080/00207543.2021.1987549.

Aghezzaf, E.H., 2008. Robust distribution planning for supplier-managed inventory agreements when demand rates and travel times are stationary. Journal of the Operational Research Society 59, 1055–1065. https://doi.org/10.1057/palgrave.jors.2602444.

Alhijawi, B., Awajan, A., 2024. Genetic algorithms: Theory, genetic operators, solutions, and applications. Evolutionary Intelligence 17, 1245–1256. https://doi.org/10.1007/s12065-023-00822-6.

Archetti, C., Bertazzi, L., Laporte, G., Speranza, M.G., 2007. A branch-and-cut algorithm for a vendor-managed inventory-routing problem. Transportation Science 41, 382–391. https://doi.org/10.1287/trsc.1060.0188.

Bell, W.J., Dalberto, L.M., Fisher, M.L., Greenfield, A.J., Jaikumar, R., Kedia, P., Mack, R.G., Prutzman, P.J., 1983. Improving the distribution of industrial gases with an on-line computerized routing and scheduling optimizer. Interfaces 13, 4–23. https://doi.org/10.1287/inte.13.6.4.

Brinkmann, J., Ulmer, M.W., Mattfeld, D.C., 2020. The multi-vehicle stochastic-dynamic inventory routing problem for bike sharing systems. Business Research 13, 69–92. https://doi.org/10.1007/s40685-019-0100-z.

Cavagnini, R., Bertazzi, L., Maggioni, F., 2022. A rolling horizon approach for a multi-stage stochastic fixed-charge transportation problem with transshipment. European Journal of Operational Research 301, 912–922. https://doi.org/10.1016/j.ejor.2021.11.037.

Chiu, A., Angulo, G., Larrain, H., 2024. Optimizing the long-term costs of an inventory routing problem using linear relaxation. Transportation Research Part E: Logistics and Transportation Review 183, 103447. https://doi.org/10.1016/j.tre.2024.103447.

Coelho, L.C., Cordeau, J.F., Laporte, G., 2014. Heuristics for dynamic and stochastic inventory-routing. Computers & Operations Research 52, 55–67. https://doi.org/10.1016/j.cor.2014.07.001.

Cubillos, M., Spliet, R., Wøhlk, S., 2022. On the effect of using sensors and dynamic forecasts in inventory-routing problems. INFOR: Information Systems and Operational Research 60, 473–490. https://doi.org/10.1080/03155986.2022.2073110.

Cuellar-Usaquén, D., Ulmer, M.W., Gomez, C., Álvarez-Martínez, D., 2024. Adaptive stochastic lookahead policies for dynamic multi-period purchasing and inventory routing. European Journal of Operational Research 318, 1028–1041. https://doi.org/10.1016/j.ejor.2024.06.011.

Cui, Z., Long, D.Z., Qi, J., Zhang, L., 2023. The inventory routing problem under uncertainty. Operations Research 71, 378–395. https://doi.org/10.1287/opre.2022.2407.

Dror, M., Ball, M., 1987. Inventory/routing: Reduction from an annual to a short-period problem. Naval Research Logistics (NRL) 34, 891–905. https://doi.org/10.1002/1520-6750(198712)34:6<891::AID-NAV3220340613>3.0.CO;2-J.

Ehrhardt, R., Mosier, C., 1984. A revision of the power approximation for computing (s, s) policies. Management Science 30, 618–622. https://doi.org/10.1287/mnsc.30.5.618.

Elbek, M., Wøhlk, S., 2016. A variable neighborhood search for the multi-period collection of recyclable materials. European Journal of Operational Research 249, 540–550. https://doi.org/10.1016/j.ejor.2015.08.035.

Greif, T., Bouvier, L., Flath, C.M., Parmentier, A., Rohmer, S.U., Vidal, T., 2024. Combinatorial optimization and machine learning for dynamic inventory routing. arXiv preprint arXiv:2402.04463 https://doi.org/10.48550/arXiv.2402.04463.

Jakobović, D., Đurasević, M., Brkić, K., Fosin, J., Carić, T., Davidović, D., 2023. Evolving dispatching rules for dynamic vehicle routing with genetic programming. Algorithms 16, 285. https://doi.org/10.3390/a16060285.

Jemai, Z., Rekik, Y., Kalaï, R., 2013. Inventory routing problems in a context of vendor-managed inventory system with consignment stock and transshipment. Production Planning & Control 24, 671–683. https://doi.org/10.1080/09537287.2012.666844.

Jia, M., Schrotenboer, A.H., Chen, F., 2025. Scenario predict-then-optimize for data-driven online inventory routing. Transportation Science 59, 1032–1056. https://doi.org/10.1287/trsc.2024.0613.

Koza, J.R., 1994. Genetic programming as a means for programming computers by natural selection. Statistics and Computing 4, 87–112. https://doi.org/10.1007/BF00175355.

Li, R., Cui, Z., Kuo, Y.H., Zhang, L., 2023. Scenario-based distributionally robust optimization for the stochastic inventory routing problem. Transportation Research Part E: Logistics and Transportation Review 176, 103193. https://doi.org/10.1016/j.tre.2023.103193.

Liang, J., Liu, D., Jena, S.D., Lodi, A., Vidal, T., 2024. Dual policy reinforcement learning for real-time rebalancing in bike-sharing systems. arXiv preprint arXiv:2406.00868 https://doi.org/10.48550/arXiv.2406.00868.

Lopes, R.L., Figueira, G., Amorim, P., Almada-Lobo, B., 2020. Cooperative coevolution of expressions for (r, q) inventory management policies using genetic programming. International Journal of Production Research 58, 509–525. https://doi.org/10.1080/00207543.2019.1597293.

Mahjoob, M., Fazeli, S.S., Milanlouei, S., Tavassoli, L.S., Mirmozaffari, M., 2022. A modified adaptive genetic algorithm for multi-product multi-period inventory routing problem. Sustainable Operations and Computers 3, 1–9. https://doi.org/10.1016/j.susoc.2021.08.002.

Maia, F., Figueira, G., Neves-Moreira, F., 2026. Stochastic dynamic inventory-routing: A comprehensive review. Computers & Operations Research , 107383https://doi.org/10.1016/j.cor.2026.107383.

McKenna, R.S., Robbins, M.J., Lunday, B.J., McCormack, I.M., 2020. Approximate dynamic programming for the military inventory routing problem. Annals of Operations Research 288, 391–416. https://doi.org/10.1007/s10479-019-03469-8.

Mes, M., Schutten, M., Rivera, A.P., 2014. Inventory routing for dynamic waste collection. Waste Management 34, 1564–1576. https://doi.org/10.1016/j.wasman.2014.05.011.

Neves-Moreira, F., Almada-Lobo, B., Guimarães, L., Amorim, P., 2022. The multi-product inventory-routing problem with pickups and deliveries: Mitigating fluctuating demand via rolling horizon heuristics. Transportation Research Part E: Logistics and Transportation Review 164, 102791. https://doi.org/10.1016/j.tre.2022.102791.

Niakan, F., Rahimi, M., 2015. A multi-objective healthcare inventory routing problem; a fuzzy possibilistic approach. Transportation Research Part E: Logistics and Transportation Review 80, 74–94. https://doi.org/10.1016/j.tre.2015.04.010.

Nolz, P.C., Absi, N., Feillet, D., 2014. A stochastic inventory routing problem for infectious medical waste collection. Networks 63, 82–95. https://doi.org/10.1002/net.21523.

Puterman, M.L., 1994. Markov decision processes: discrete stochastic dynamic programming. John Wiley & Sons. https://doi.org/10.1002/9780470316887.

Raba, D., Estrada-Moreno, A., Panadero, J., Juan, A.A., 2020. A reactive simheuristic using online data for a real-life inventory routing problem with stochastic demands. International Transactions in Operational Research 27, 2785–2816. https://doi.org/10.1111/itor.12776.

Roldán, R.F., Basagoiti, R., Coelho, L.C., 2016. Robustness of inventory replenishment and customer selection policies for the dynamic and stochastic inventory-routing problem. Computers & Operations Research 74, 14–20. https://doi.org/10.1016/j.cor.2016.04.004.

Shukla, N., Tiwari, M., Ceglarek, D., 2013. Genetic-algorithms-based algorithm portfolio for inventory routing problem with stochastic demand. International Journal of Production Research 51, 118–137. https://doi.org/10.1080/00207543.2011.653010.

Silver, E.A., Pyke, D.F., Thomas, D.J., 2016. Inventory and Production Management in Supply Chains (4th ed.). CRC Press. https://doi.org/10.1201/9781315374406.

Violi, A., De Maio, A., Fattoruso, G., Olivieri, M.G., 2023. An age-based dynamic approach for distribution of perishable commodities with stochastic demands. Soft Computing 27, 7039–7050. https://doi.org/10.1007/s00500-023-07917-3.

Vonolfen, S., Affenzeller, M., Beham, A., Lengauer, E., Wagner, S., 2013. Simulation-based evolution of resupply and routing policies in rich vendor-managed inventory scenarios. Central European Journal of Operations Research 21, 379–400. https://doi.org/10.1007/s10100-011-0232-5.

Appendix 3.A Constructive routing heuristic

Algorithm 3 introduces the constructive nearest-neighbor heuristic with a 2-opt local search improvement to obtain vehicle routes, i.e., the third and final stage of our three-stage policy, during the training and validation phases. In the final evaluation (i.e., test set), vehicle routes are instead obtained using Gurobi.

Algorithm 3 Constructive nearest-neighbor heuristic with 2-opt improvement.

input: supplier (0), set of customers to be visited (C), distance matrix ($dist(i, j)$)
output: best route (R^*) and corresponding distance (d_{R^*})
initialize $d_{R^*} \leftarrow \infty$, $R^* \leftarrow \emptyset$, $C' \leftarrow \emptyset$, $u \leftarrow 0$, $v \leftarrow 0$
for each customer i in C **do**
    initialize route $r \leftarrow (0, i)$ and corresponding distance $d_r \leftarrow dist(0, i)$
    $C' \leftarrow C \setminus \{i\}$
    $u \leftarrow i$
    while $C' \neq \emptyset$ **do**
        find $v \in C'$ such that $dist(u, v)$ is minimized
        $r \leftarrow r \oplus (v)$
        $C' \leftarrow C' \setminus \{v\}$
        $d_r \leftarrow d_r + dist(u, v)$
        $u \leftarrow v$
    end while
    $r \leftarrow r \oplus (0)$
    $d_r \leftarrow d_r + dist(u, 0)$
    $r \leftarrow$ 2-opt(r) and update d_r
    if $d_r < d_{R^*}$ **then**
        $d_{R^*} \leftarrow d_r$
        $R^* \leftarrow r$
    end if
end for
return R^* and d_{R^*}

Appendix 3.B Setup cost estimates comparison

Table 3.10 compares several setup cost configurations using C_{mean} as the total cost performance metric, with the customer-supplier distance configuration as the reference baseline. Each setup cost configuration is evaluated on the same training set used to learn the priority rules and to calibrate the RPA parameters, using a fixed three-stage policy structure: (i) a standard priority rule (lowest inventory first), (ii) the original RPA parameters, and (iii) the constructive nearest-neighbor heuristic for the routing optimization stage. The setup cost configuration based on the average distance to the closest $\delta = 10\%$ of customers yields the lowest aggregate cost, with $C_{mean} = 0.97$, being the one selected for the following learning experiments. In contrast, the configuration based on the average distance to the closest $\delta = 5\%$ of customers is the one performing worst, with $C_{mean} = 1.34$.

Table 3.10: Setup cost configurations comparison (**customer-supplier distance** as the baseline).

Policy (π)	C_{mean}
LIF-RPA($\delta = 5\%$)	1.34
LIF-RPA($\delta = 10\%$)	**0.97**
LIF-RPA($\delta = 20\%$)	0.98
LIF-RPA($\delta = 30\%$)	0.98
LIF-RPA($\delta = 40\%$)	0.99
LIF-RPA($\delta = 50\%$)	0.99
LIF-RPA($\delta = 60\%$)	0.99
LIF-RPA($\delta = 70\%$)	0.99
LIF-RPA($\delta = 80\%$)	1.00
LIF-RPA($\delta = 90\%$)	1.00
LIF-RPA($\delta = 100\%$)	1.00

Appendix 3.C Solution quality

Table 3.11 presents the overall performance of all policies considered in this work, as a complement to Section 3.6. The value of RPA in calculating the (s,S) inventory control parameters of our inventory replenishment rule over the standard $(\alpha\bar{U}_i, \bar{U}_i)$ and OU rules is reported in Section 3.6.1. The value of learning the priority rule via GP and then calibrating the RPA parameters to the SDIRP setting using a GA is described in Section 3.6.2.

Table 3.11: Overall performance per policy (**MYOPIC** as the baseline).

Policy (π)	Total cost performance			Operational behavior			Cost component		
	C_{mean}	C_{max}	WR	Number of visits	Delivery qty.	Stockout qty.	Holding	Stockout	Routing
BOF-OU	1.51	2.48	0%	1.05	1.24	0.30	7.80	0.37	1.08
LSF-OU	1.54	2.57	0%	1.11	1.24	0.27	7.80	0.35	1.12
LIF-OU	1.52	2.48	0%	1.09	1.25	0.00	7.79	**0.00**	1.11
BOF-$(0.25\bar{U}_i, \bar{U}_i)$	0.95	1.68	63%	0.29	1.12	2.60	4.60	3.32	**0.61**
BOF-$(0.50\bar{U}_i, \bar{U}_i)$	1.06	1.89	52%	0.43	1.18	0.14	5.97	0.17	0.74
BOF-$(0.75\bar{U}_i, \bar{U}_i)$	1.32	2.26	18%	0.71	1.22	0.22	7.15	0.27	0.92
LSF-$(0.25\bar{U}_i, \bar{U}_i)$	0.91	1.59	67%	0.30	1.12	1.55	4.58	2.01	0.62
LSF-$(0.50\bar{U}_i, \bar{U}_i)$	1.06	1.90	53%	0.43	1.18	0.02	5.97	0.03	0.75
LSF-$(0.75\bar{U}_i, \bar{U}_i)$	1.33	2.28	14%	0.74	1.22	0.11	7.15	0.14	0.94
LIF-$(0.25\bar{U}_i, \bar{U}_i)$	0.90	1.56	67%	0.30	1.13	1.35	4.57	1.73	0.62
LIF-$(0.50\bar{U}_i, \bar{U}_i)$	1.06	1.90	52%	0.43	1.18	0.01	5.97	0.01	0.75
LIF-$(0.75\bar{U}_i, \bar{U}_i)$	1.33	2.24	15%	0.73	1.22	0.00	7.15	**0.00**	0.94
BOF-RPA	0.83	1.07	81%	0.40	1.08	1.62	2.36	2.05	0.67
LSF-RPA	0.81	1.07	83%	0.40	1.08	1.20	2.36	1.50	0.67
LIF-RPA	0.81	1.05	85%	0.40	1.08	1.08	2.36	1.33	0.67
$DLA_{l=1}$	0.80	**0.96**	**99%**	0.66	1.06	0.35	2.13	0.36	0.69
$DLA_{l=2}$	0.80	**0.96**	**99%**	0.64	1.07	0.36	2.29	0.40	0.69
GP_{10}-RPA	0.79	1.05	86%	0.40	1.08	1.25	2.36	1.36	0.66
GP_{10}-tRPA	**0.78**	1.00	93%	0.44	1.08	1.23	**2.04**	1.30	0.68

Appendix 3.D Trade-off between performance and runtime

Figure 3.3 compares our best-performing three-stage policy (**GP$_{10}$-tRPA**) with the **MYOPIC** and **DLA** (one and two-period lookahead) benchmarks in terms of total cost performance (C_{mean}) and average runtime (RT) per scenario configuration $m \in \{1, ..., M\}$. Our policy achieves better overall total cost performance, taking, on average, only a fraction of a second to a few seconds to solve large instances. In contrast, **DLA$_{l=1}$** and **DLA$_{l=2}$** require from a few seconds (for small instances) to several hours (for large instances). In particular, for some instances with 25 and 50 customers, both **DLAs** reach the two-hour time limit imposed at each decision period using Gurobi, with running times extending to tens of hours. Overall, **GP$_{10}$-tRPA** requires approximately three orders of magnitude less computation time than **MYOPIC**, and four orders of magnitude less than **DLA$_{l=1}$** and **DLA$_{l=2}$**.

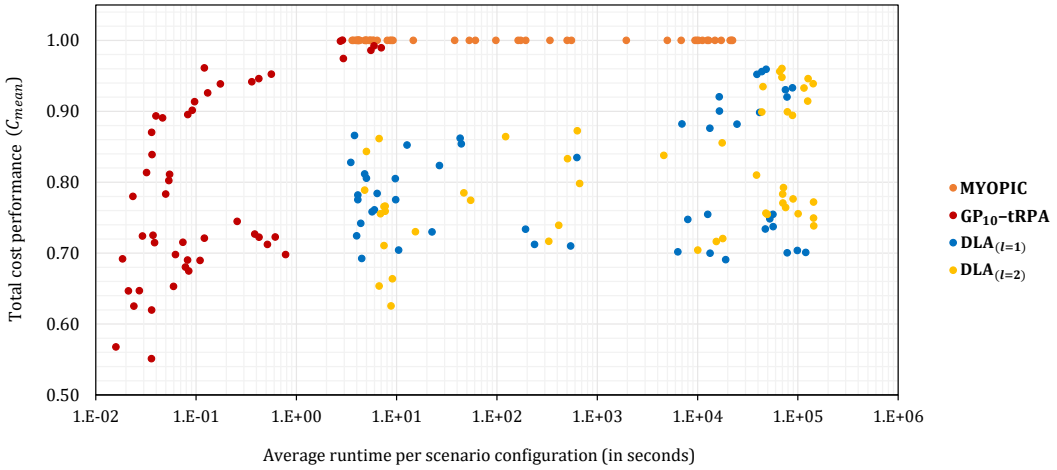


Figure 3.3: Total cost performance (**MYOPIC** as the baseline) vs. average runtime per scenario configuration.

Appendix 3.E Gurobi optimality gaps

As described in Section 3.5.5, at each decision period of the planning horizon, both **MYOPIC** and **DLA** benchmarks solve an integrated inventory-routing optimization subproblem using Gurobi Optimizer 10.0.2. A 2-hour time limit is imposed to ensure that the computational experiments remain tractable. Table 3.12 presents the average optimality gaps aggregated by scenario group, highlighting the challenge of solving large-scale instances within a feasible timeframe, even for a one-period lookahead horizon.

Table 3.12: Gurobi optimality gaps (on average) for **MYOPIC** and **DLA**.

Scenario group	Gap (in %)		
	MYOPIC	$DLA_{l=1}$	$DLA_{l=2}$
$N = 5$	0.00	0.00	0.00
$N = 10$	0.00	0.00	0.15
$N = 25$	0.04	0.75	3.21
$N = 50$	0.24	2.39	6.24
$T = 10$	0.10	0.95	2.40
$T = 20$	0.05	0.70	2.40
$(hf, sf) = (1, 200)$	0.08	0.99	3.14
$(hf, sf) = (10, 20)$	0.06	0.58	1.66
$vf = 1.25$	0.07	0.89	2.77
$vf = 1.50$	0.07	0.70	2.13
$vf = 1.75$	0.08	0.76	2.30

CHAPTER 4

Policy design for a real-world pharmaceutical distribution problem

Stochastic dynamic inventory-routing via cost function approximation: An application to pharmaceutical distribution

Francisco Maia[*] · **Marlin Ulmer**[‡] · **Florentin Hildebrandt**[‡] · **Gonçalo Figueira**[*] · **Fábio Neves-Moreira**[†]

Submitted to *European Journal of Operational Research*, 2026

Highlights

- Considers inventory-routing operations of a Portuguese pharmaceutical distributor;
- Models the stochastic dynamic problem as a sequential decision process;
- Proposes novel inventory-routing policies based on cost function approximation;
- Evaluates the proposed policies through comprehensive experiments.

Abstract This research addresses a stochastic dynamic inventory-routing problem motivated by the operations of a pharmaceutical distributor in Portugal that replenishes a set of pharmacies from a central warehouse over a multi-period horizon. In this setting, the dominant trade-off lies between inventory holding and stockout costs, while routing acts primarily as a resource constraint. To address the inherent complexities of uncertain demand and multi-period routing, we propose a cost function approximation-based policy within the framework of approximate dynamic programming. Our policy relies on a parameterized mathematical programming model that yields current-period inventory-routing decisions in a rolling horizon fashion. Its modified objective function: (1) minimizes inventory deviations from an optimized target level at each pharmacy, anticipating future demand; and (2) promotes inventory synchronization across geographically close pharmacies through a routing-aware penalty term, thereby accounting for future routing limitations. Both policy parameters, namely the individual inventory targets and a penalty weight controlling the trade-off between the two objectives, are jointly learned via Bayesian optimization. We evaluate our policy's performance across six experimental scenarios defined by three levels

[*]INESC TEC, Faculdade de Engenharia, Universidade do Porto, Rua Dr. Roberto Frias, Porto, 4200-465, Portugal
[‡]Otto-von-Guericke-Universität, Universitätsplatz 2, Magdeburg, 39106, Germany
[†]INESC TEC, Faculdade de Economia, Universidade do Porto, Rua Dr. Roberto Frias, Porto, 4200-464, Portugal

of demand uncertainty and two vehicle capacity settings. We consider five benchmarks: three practice-oriented that reflect real-world operations, and two method-oriented. Results show that anticipating future demand via proposed inventory target levels, balancing inventories across geographically proximate pharmacies, and learning scenario-dependent policy parameters are already effective individually. However, their joint integration yields superior performance, proving the strategic advantage of treating pharmacies as an interconnected network rather than isolated entities.

Keywords Pharmaceutical distribution · Routing · Stochastic dynamic inventory-routing · Sequential decision-making · Cost function approximation

4.1. Introduction

Pharmaceutical distribution has experienced significant advancements in recent decades, evolving into a globally integrated and regulated sector. This sector is characterized by complex operational dynamics, including strict service-level requirements, regulatory constraints, and high levels of uncertainty stemming from fluctuations in demand. From a pharmacy's perspective, ensuring a quick response to customer needs while controlling operational costs is a challenging task that requires close coordination with suppliers. However, these operational, logistical, and coordination complexities often prevent effective alignment among stakeholders, leading each party to prioritize its own objectives. For example, pharmacies and healthcare facilities may focus primarily on product availability, while suppliers concentrate on maximizing distribution efficiency within their operational constraints. To bridge this gap, a centralized decision-making framework, such as vendor-managed inventory (VMI), is essential to synchronize the network and balance product availability with costs (Van den Bogaert and Van Jaarsveld, 2022). In practice, although most pharmacies still rely on independent replenishment policies, a growing number are moving toward VMI. This paper focuses on those pharmacies. In addition, while standard distribution operations often rely on consistent, periodic routes, dynamic multi-period routing becomes increasingly important for reducing logistical waste and optimizing vehicle utilization.

The unavailability of medicines, vaccines, and other pharmaceutical products remains a critical concern, affecting healthcare provision and the continuity of treatments. For instance, shortages have been reported for essential drugs such as antidiabetics, antihypertensives, and medications for attention-deficit disorders (Infarmed, 2024). The reasons for such unavailability are threefold. First, transportation operations, which are often costly and capacity-constrained, may fail to replenish inventory in time. Second, pharmacies tend to limit on-hand stock in order to avoid significant holding costs. Third, demand for certain medicines is highly volatile and uncertain, which complicates replenishment decisions (Cornelissen et al., 2026). The central challenge, therefore, lies in managing the trade-off between having the right amount of inventory and ensuring cost-efficient routing over a multi-period horizon. These complexities align with the inventory-routing problem (IRP), which combines inventory management and vehicle routing, providing a practical optimization framework for integrating replenishment and distribution decisions.

The inventory-routing literature has been widely explored over the past decades, with most contributions focusing on static variants, whether deterministic or stochastic. Although dynamic variants have attracted growing attention in recent years, they remain comparatively less explored given the increased complexity and difficulty of jointly accounting for demand uncertainty and sequential decision-making (Maia et al., 2026). This gap is particularly important in practical applications such as pharmaceutical distribution, where service reliability is critical. In these settings, routing often acts as a limiting resource shaping feasible replenishment decisions, while the dominant trade-offs arise from inventory holding and stockout costs.

In this paper, we consider a stochastic dynamic inventory-routing problem (SDIRP) motivated by the operations of a Portuguese pharmaceutical distributor. The distributor manages the inventories of multiple pharmacies in southern Portugal, supplying them with a single medicine from a regional warehouse over a finite set of periods (e.g., days or intra-day shifts). Demand is stochastic and realized for each pharmacy at the end of each period. At the beginning of each period, the distributor must determine the replenishment quantities for each pharmacy as well as the corresponding vehicle routes, using a limited fleet of specialized vehicles subject to operational restrictions, including vehicle capacity, route duration and service times. Accordingly, the problem can be modeled as a sequential decision process, where inventory and routing decisions are repeatedly made over time to react to and anticipate future uncertain demand realizations, aiming to minimize the total inventory holding and stockout costs.

Finding effective decisions can be quite challenging, as our problem combines inventory management for many pharmacies with dynamic multi-period routing. On the one hand, it requires addressing immediate pharmacy demand, keeping holding costs limited, and ensuring long-term inventory stability. On the other hand, inventory decisions are constrained by daily routing resources, whose feasibility depends on available delivery time and capacity. To address both dimensions in an integrated fashion, we combine classical concepts of inventory management with approximate dynamic programming (ADP) to guide inventory-routing decisions.

In particular, we introduce an inventory target level for each pharmacy based on well-established inventory management strategies to hedge against demand uncertainty and balance inventory holding and stockout costs (Silver et al., 2016). Naturally, meeting these inventory targets may be difficult to achieve in every period due to constrained delivery resources, which limit inventory flexibility in response to future developments. Thus, we penalize post-delivery inventory deviations from these target levels in our current-period decisions. Since routing constraints are embedded in the decision space, the resulting decisions already yield feasible replenishment plans and effective routes. However, they do not anticipate the potential of replenishing pharmacies in future periods. Indeed, preliminary tests reveal that pharmacies located at the outskirts of the service area may receive limited stock today and might not be fully replenished tomorrow. As a consequence, those pharmacies may experience poor service quality. Anticipating future operational limitations in today's decision-making is therefore essential for effective operations.

The crucial question then becomes how to integrate future routing considerations into the proposed inventory target strategy. In our stochastic dynamic setting, routing limi-

tations and their consequences extend beyond current decisions, propagating into future periods and depending on uncertain demand realizations. Since an exact calculation of this impact is computationally infeasible, we propose approximating it instead. The idea of the approximation is to measure local inventory imbalance and encourage similar stock levels among geographically close pharmacies. Ideally, this leads to synchronized replenishment needs at similar periods and, as a result, effective inventory-routing decisions. In the framework of ADP, this procedure is known as cost function approximation (CFA) (Powell, 2022b). The cost function combines the individual inventory target strategy and a routing-aware inventory synchronization strategy via a penalty weight, guiding current-period decisions while preserving computational tractability.

We evaluate our **CFA** policy using the network of a Portuguese pharmaceutical distributor across six experimental scenarios, considering three levels of demand uncertainty (low, medium, high) and two vehicle capacity settings (small and large). In this context, both inventory target levels and the penalty weight are adaptive to each scenario and learned via Bayesian optimization (BO) to ensure our policy remains robust across different situations. Our study provides the following key insights:

- Moving from traditional, consistent periodic routes to dynamic multi-period routing significantly reduces total inventory costs, contributing to more flexible and effective replenishments;
- Our **CFA** policy effectively adapts to varying levels of demand volatility and vehicle capacity settings, demonstrating that our integrated approach is essential to capture the critical interdependencies between inventory and routing decisions;
- Anticipating demand via proposed inventory target levels, balancing inventories across geographically proximate pharmacies, and learning scenario-dependent policy parameters are effective individually but achieve superior performance when combined;
- Overall, the improvements of the proposed **CFA** policy over the benchmarks are most pronounced under tighter vehicle capacity constraints and lower demand uncertainty.

Our contributions are as follows. We address a real-world distribution problem involving the daily replenishment of multiple pharmacies from a central warehouse using a fleet of vehicles operating under realistic conditions. To handle the general trade-off between inventory and routing inherent in many multi-period IRPs, we propose an integrated **CFA** policy by defining a parametrically modified objective function that anticipates future demand, aligns inventories across geographically close pharmacies, and thus enables more efficient pharmacy visits and replenishments, substantially reducing holding and stockout costs. The proposed approach is also applicable to other application domains in which inventory replenishment and routing decisions must be coordinated dynamically under uncertainty.

The remainder of the paper is organized as follows. We review the relevant literature in Section 4.2, both on pharmaceutical distribution and on stochastic dynamic inventory-routing. The problem is defined in Section 4.3, starting with a general description, followed

by an illustrative example, and a formulation of the problem as a sequential decision process. In Section 4.4, we present our solution method. The experimental setting is described in Section 4.5, including instance design, benchmarks and implementation details. The computational study and a thorough analysis of the results are presented in Section 4.6. Section 4.7 presents the main conclusions of our research and identifies multiple directions for further research.

4.2. Related works

This section provides an overview of the existing research on pharmaceutical distribution (Section 4.2.1), SDIRP models and methods (Section 4.2.2). For a detailed and comprehensive review on SDIRPs, we refer to Maia et al. (2026).

4.2.1 Pharmaceutical distribution

The literature on pharmaceutical distribution can be organized into multiple research streams, from strategic problems, such as network design, to more operational problems, such as vehicle routing, and inventory-routing. In the context of network design problems, Martins et al. (2017) propose a hybrid optimization-simulation framework, validated on a pharmaceutical wholesaler case study, that accounts for both operational costs and customer service levels. Redesign decisions are first optimized using a mixed-integer programming model, and the resulting solution is then simulated to evaluate its impact on the wholesaler's daily operations. Janatyan et al. (2025) propose a robust optimization model for the design of a sustainable pharmaceutical distribution network under uncertain demand and transportation cost. The authors formulate a multi-objective mixed-integer linear programming model to determine the main and local distribution centers as the strategic decision, and then the medicine flows in each route as the operational decision.

Research on vehicle routing focuses on improving transportation efficiency and consistency in pharmaceutical logistics. Soysal and Çimen (2017) address a green capacitated vehicle routing problem (VRP) that considers time-dependent speeds and transportation emissions in a real-life urban distribution setting. The authors propose a simulation-based restricted dynamic programming approach. Kramer et al. (2019) formulate a rich multi-period VRP with flexible time windows for a third-party logistics provider, considering multi-day planning, early deliveries to hospitals, periodic deterministic demands and a maximum number of customers per route. The authors solve the problem using a multi-start iterated local search algorithm. Campelo et al. (2019) investigate a consistent VRP for a Portuguese pharmaceutical distributor, considering multiple daily deliveries and various service level agreements, such as time windows, release dates and multiple shipping periods per customer. The authors propose an instance size reduction algorithm and a matheuristic approach to solve the problem. Chen et al. (2024) formulate a multi-depot VRP with time windows for refrigerated pharmaceuticals, developing mixed-integer linear programming models and collaborative strategies that incorporate time-dependent travel speeds and vehicle flow equilibrium. Three hybrid heuristics based on simulated annealing and variable neighborhood search are developed and evaluated. Lee and Kim (2024)

investigate a one-to-many delivery problem in a pharmaceutical cold chain, where each customer requests multiple products with specific due dates. Each product has distinct characteristics, such as storage temperature range and shelf life. The authors formulate the refrigerated routing and scheduling problem with order acceptance as a mixed-integer linear programming model, aiming to maximize total profit. A hybrid genetic algorithm that embeds an insertion heuristic with modification operation is proposed.

Vehicle routing is often integrated with inventory replenishment, in the so-called IRP. Niakan and Rahimi (2015) address a healthcare IRP involving drug distribution from a supplier to multiple healthcare facilities under uncertain demand. The authors propose a multi-objective mathematical model in which the first objective minimizes total inventory and routing costs, while the second maximizes customer satisfaction. A hybrid possibilistic method is employed to handle uncertainty, and an interactive fuzzy approach is used to solve an auxiliary crisp multi-objective model. Timajchi et al. (2019) explore an IRP for hazardous and deteriorating pharmaceutical items in a healthcare network with deterministic, time-varying demand. Hospitals receive supplies from a central pharmacy or through lateral transshipments from other hospitals' drug stores. The problem is formulated as a bi-objective mixed-integer mathematical programming model. The first objective is to minimize logistical costs, including ordering, transportation and inventory. The second focuses on minimizing the maximum accident loss during distribution. The model is solved using a commercial solver, and the results are compared with solutions from an efficient hybrid genetic algorithm.

Furthermore, Zandkarimkhani et al. (2020) propose a bi-objective mixed-integer linear programming model to design a perishable pharmaceutical three-echelon supply chain network with uncertain demands. The model defines an inventory-location-routing problem, and considers procurement discounts, product lifetime, and time window constraints. The objective is to minimize total costs, including fuel consumption, transportation, inventory holding and penalties for exceeding time windows, as well as the total shortage amount. The authors suggest a hybrid approach that combines fuzzy theory with chance-constrained programming. Peng and Liao (2025) address a two-level supply chain network IRP in pharmaceutical cold chain logistics, considering economic, social and environmental criteria. The authors propose a mixed-integer nonlinear programming model with an objective function that accounts for three distinct cost components: inventory, transportation, and carbon emissions. A comparative analysis is performed between sequential and integrated decision-making approaches.

Despite the critical role of pharmaceutical distribution in healthcare, current research overlooks the stochastic and dynamic nature of demand faced by hospitals, pharmacies, and other healthcare providers. Therefore, there is a notable scarcity of studies on the SDIRP in the context of pharmaceutical distribution. This limited literature underscores the need for models that explicitly capture the temporal evolution and uncertainty of demand, as well as for dynamic decision policies that adapt inventory replenishment and routing decisions over time.

4.2.2 Stochastic dynamic inventory-routing

The SDIRP comprises several challenges, including inventory management, multi-period routing and stochastic dynamic decision-making. Given its practical relevance, the problem has been addressed in many application domains, from grocery (e.g., Vonolfen et al., 2013) and energy distribution (e.g., Li et al., 2023) to bike-sharing (e.g., Brinkmann et al., 2020) and waste collection (e.g., Markov et al., 2020). We provide a brief overview of the existing literature.

Uncertainty primarily stems from customer demand. Additional sources, such as travel times, supply and purchase prices, are occasionally acknowledged (e.g., Morales et al., 2018; Hasturk et al., 2024; Cuellar-Usaquén et al., 2024). While most studies model uncertainty using random variables with known probability distributions, alternative approaches include fuzzy (e.g., Niakan and Rahimi, 2015) and interval-based formulations (e.g., Agra et al., 2018). In contrast, when no future information is provided, decisions are made in response to real-time service requests (e.g., Bertazzi et al., 2025). In our problem setting, uncertainty arises from the stochastic demand faced by a set of pharmacies, driven by several factors, including random patient arrivals, seasonal outbreaks, and geographical issues.

From a high-level perspective on inventory management, the IRP literature distinguishes between two main inventory policies: order-up-to-level (OU) and maximum-level (ML). Under an ML policy, the supplier decides how much to deliver to each customer based on the available resources, expected demand, storage and vehicle capacities (e.g., Brinkmann et al., 2020). In contrast, an OU policy implies that a customer's inventory is fully replenished at each visit (e.g., Greif et al., 2024). Stockouts can be modeled either as lost sales or backlogged demand for future periods. Given the urgent nature of pharmaceutical operations and the high service levels required, modeling stockouts as lost sales better reflects operational reality. In extreme scenarios, pharmacies can count on emergency shipments from other suppliers or lateral transshipments from other pharmacies.

In periodic settings where decisions are made daily, the current models often neglect time-related routing constraints, such as route duration limits and service times, treating deliveries as instantaneous. We address this gap by explicitly incorporating these real-world concerns. Additionally, most literature considers homogeneous fleets of multiple vehicles (e.g., Achamrah et al., 2022). Such configurations enhance delivery capacity and routing flexibility, ensuring that service-level requirements are met while accommodating fluctuating demand. In contrast, some studies adopt single-vehicle architectures, reducing the routing challenge to a traveling salesman problem (TSP) (e.g., Cui et al., 2023), while others assume direct deliveries from suppliers to customers (e.g., Cavagnini et al., 2022).

Another critical aspect is the problem's optimization objective. While most studies focus on inventory holding, stockout, and routing costs, some incorporate additional cost factors. For instance, within the context of industrial gas distribution, Berman and Larson (2001) minimize the total expected costs associated with earliness, lateness, product shortfall, and returning to the depot non-empty. Mes et al. (2014) minimize the total transportation and handling costs, including a penalty for overflow waste. Liang et al. (2024) minimize the unmet demand in bike-sharing logistics. In pharmaceutical distribution, the primary concern is ensuring product availability while minimizing inventory holding and

stockout costs. Routing decisions are managed by explicit travel time constraints, including a maximum allowable route duration that enforces operational limits on driver working hours, as well as service times that capture the handling and unloading of products at pharmacy locations. These constraints ensure routing feasibility and support on-time deliveries, allowing transportation costs to be omitted from the objective function.

Interest in dynamic problems has rapidly escalated, with the combination of operations research and machine learning revealing promising outcomes when dealing with complex combinatorial optimization problems with large state and decision spaces, where it fits our SDIRP. Following the unified framework of Powell (2022b), we highlight the following four classes of policies: policy function approximation (PFA), cost function approximation (CFA), direct lookahead approximation (DLA) and value function approximation (VFA). Policy search suggests searching across a set of policies, i.e., PFA (e.g., Mes et al., 2014), or a set of parameters of a given policy, i.e., CFA (e.g., Greif et al., 2024), to identify the one that works better over time by simulating their behavior. On the other hand, lookahead approximations contribute to policies that approximate the impact of a right-now decision on future states, through a direct lookahead model, i.e., DLA (e.g., Chiu et al., 2024) or a value function, i.e., VFA (e.g., Crama et al., 2018).

Direct lookahead policies are considered the most brute-force approach within the four policy classes. The lookahead model can be deterministic, relying on direct demand predictions (e.g., Neves-Moreira et al., 2022), or stochastic, involving multiple scenarios as in multi-stage stochastic programming (e.g., Cavagnini et al., 2022). Value function approximation is a common tool to solve complex stochastic dynamic programming problems by approximating the value of being in a given state (e.g., Lagos, 2026). Reinforcement learning (RL) algorithms are typically used to compute the value functions, performing well when approximating the downstream value of a state is straightforward. For instance, Liang et al. (2024) introduce a dual policy RL algorithm that disintegrates inventory and routing decisions to solve a dynamic bike repositioning problem. Two deep Q-networks are trained jointly for each decision.

Despite the numerous contributions based on DLA, these approaches often face scalability limitations as the lookahead horizon increases, given the significant growth of the state and decision spaces. This has motivated growing interest in policies based on PFA and CFA, which tend to be less computationally demanding, although requiring tuning. In particular, PFA is a method that directly maps a state into a decision without necessarily relying on a lookahead model. Inventory replenishment rules such as (s,S) are a good example of PFA when tuned via simulation-optimization (e.g., Vonolfen et al., 2013). Cost function approximation is a class of policies that arises from a parametric modification of a cost function (i.e., objective function or model's constraints), thereby improving the flexibility and robustness of myopic policies (e.g., Hasturk et al., 2024).

Although the literature on policy search methods remains limited, CFA holds significant promise as a bridge linking combinatorial optimization and machine learning. For instance, Greif et al. (2024) combine a physics-informed neural network that outputs state-dependent delivery prizes with a single-period capacitated prize-collecting TSP, achieving state-of-the-art performance with reduced inference time. Cuellar-Usaquén et al. (2024) incorporate an adaptive CFA within a two-stage stochastic programming method to ob-

tain inventory, purchasing and routing decisions for the current period, preserving solution quality within reasonable computation times. Motivated by this stream, we leverage CFA and combinatorial optimization to tackle the complex challenges of our SDIRP. We propose a **CFA** policy based on a parameterized mathematical programming model solved in a rolling horizon (RH) fashion, which decomposes the overall multi-period problem into a set of tractable single-period subproblems. The resulting policy captures future demand uncertainty and anticipates future routing considerations, contributing to more informed and proactive inventory-routing decisions.

4.3. Problem definition

In this section, we formally define the problem at hand (Section 4.3.1). We begin with a general description, supported by an illustrative example (Section 4.3.2), and then model the problem as a sequential decision process (Section 4.3.3).

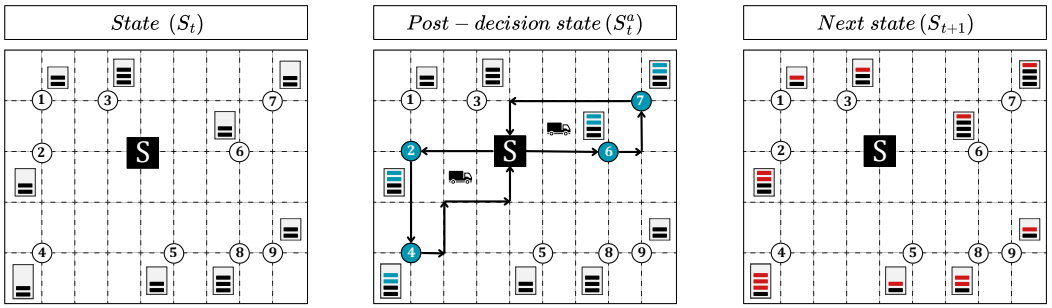
4.3.1 Problem description

We consider a problem motivated by a Portuguese pharmaceutical distributor that supplies medicines from a central warehouse (also referred to as the supplier) to a set of geographically dispersed pharmacies (also referred to as customers) along the southern coast of Portugal. The planning horizon comprises a finite number of periods (e.g., days). Each pharmacy has a maximum storage capacity, and the initial inventory levels are provided at the beginning of the planning horizon. Deliveries occur at the beginning of each period and are used to meet the demand (in discrete box units) of that period and subsequent ones. Demand is stochastic and dynamic, being revealed at the end of each period. Each pharmacy incurs a unit holding cost at the end of each period, as well as a unit shortage cost that captures the expense of a potential emergency replenishment in case of a stockout. The supplier is assumed to have sufficient stock to meet all planned deliveries, which are carried out using a fixed fleet of vehicles with limited capacity and maximum route duration. Each vehicle is assigned at most one route per period. Each route starts and ends at the warehouse, with the vehicle returning empty. These routing constraints significantly affect the daily deliveries and, therefore, inventory availability at pharmacies across the network.

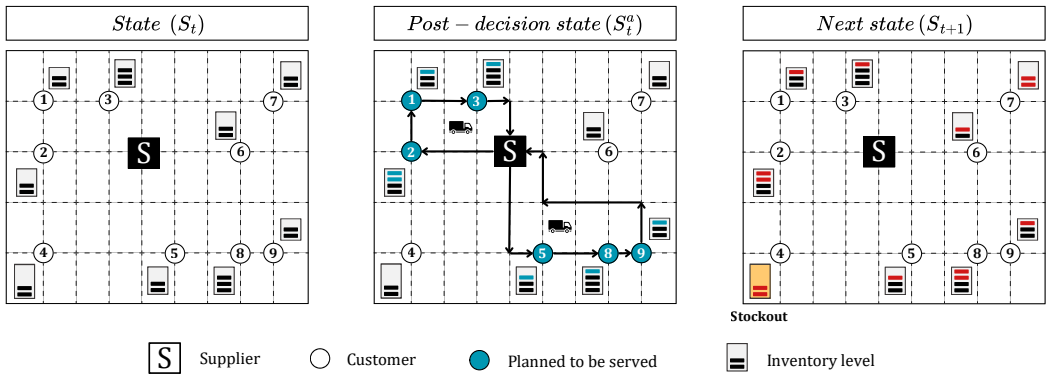
4.3.2 Illustrative example

In the following, we present an illustrative example that precedes the formal sequential decision process modeling proposed in Section 4.3.3. Figure 4.1 shows a potential state of the problem (later denoted as S_t), as well as two different decisions (a_t), resulting post-decision states (S_t^a), stochastic information realization (ω_{t+1}), and next states (S_{t+1}).

We consider a central warehouse, represented by the black square, labeled 'S', and a set of nine pharmacies, represented by white circles. The inventory levels next to each pharmacy show their current stock. Based on discussions with our practice partner, and for illustrative purposes, we consider a unit holding cost of 0.15 monetary units (MU) for all pharmacies and a stockout cost of 30 MU per shortage unit. Both costs are calculated at



(a) Prioritizing pharmacies with lower inventory utilization rates.



(b) Prioritizing geographically proximate pharmacies.

Figure 4.1: Example of a state, two possible decisions, corresponding post-decision states, demand realization, and next states.

the end of each period. In case of a stockout, the missing units are considered lost sales. A fleet of two vehicles, each with a capacity of four product units, is available for deliveries. For the sake of simplicity, travel times are not represented in the example. The first block, on the left, identifies the current state of the system (S_t). This state includes the inventory levels available at the beginning of period t. For instance, Pharmacy *1* currently holds two product units and has a storage capacity of three units.

Two potential decisions are outlined in Scenarios (a) and (b), each specifying the pharmacies to be visited, the quantities to be delivered, and the corresponding vehicle routes. Scenario (a) illustrates a decision that prioritizes pharmacies with the lowest utilization rates, defined as the ratio between current inventory and storage capacity. The first route visits Pharmacies *2* and *4* in that order, delivering two units to each. The second route visits Pharmacies *6* and *7* and delivers two units to each. Scenario (b) prioritizes geographically close pharmacies, promoting route consolidation. The first route visits Pharmacies *2*, *1*, and *3*, providing two units to the first and one unit to each of the remaining two. The second route visits Pharmacies *5*, *8*, and *9*, delivering one unit to each. Each scenario leads to a different post-decision state (S_t^a), depicted in the second block. This reflects the state of the system after each decision and before demand realization.

The third block illustrates the next state S_{t+1} for each scenario, with the realized demand visually represented as red bars in each pharmacy's depot. At this stage, the inventory holding and stockout costs for the current period can be computed. In both scenarios, Pharmacies *1*, *3*, *5*, *6*, and *9* face a demand of one unit, whereas Pharmacies *2*, *7*, and *8* experience a demand of two units. Pharmacy *4* faces a demand of three units and consequently incurs one unit of stockout. The difference between the two scenarios lies in the final inventory levels and the corresponding holding and stockout costs. In Scenario (a), there are no stockouts, and the total cost consists only of holding costs, amounting to $15\,(\text{units}) \times 0.15\,(\text{MU}) = 2.25\,\text{MU}$. In Scenario (b), a stockout of one unit occurs at Pharmacy *4*, leading to a stockout cost of 30 MU. The holding costs in this scenario remain at $14\,(\text{units}) \times 0.15\,(\text{MU}) = 2.10\,\text{MU}$.

4.3.3 Sequential decision process

The problem at hand is both stochastic and dynamic. The stochastic nature stems from demand, which is revealed at the end of each period. The dynamic aspect arises from the need to make decisions sequentially at the beginning of each period, where each decision affects not only the current inventory levels but also future states and decisions. This SDIRP can then be modeled as a sequential decision process, characterized by a sequence of states S_t, decisions a_t and demand realizations ω_{t+1}. We now proceed to define each of these components, beginning with the global notation.

4.3.3.1 Global notation

The problem is defined on a graph $\mathcal{G} = (\mathcal{N}, \mathcal{E})$, with $\mathcal{N} = \{0\} \cup \mathcal{N}'$, where a central warehouse, denoted by index 0, is responsible for delivering a single product to a set of N geographically distributed pharmacies, denoted by $\mathcal{N}' = \{1, 2, ..., N\}$. The connection between the locations is defined by a set of arcs $\mathcal{E} = \{(i, j) : i, j \in \mathcal{N}, i \neq j\}$. Each arc is characterized by a travel distance, denoted by d_{ij}, and a corresponding travel time, denoted by τ_{ij}, while each pharmacy visit requires a service time, denoted by δ. The planning horizon $\mathcal{T}$ splits into a finite set of T consecutive periods, each representing a day $t \in \{0, 1, ..., T-1\}$. The initial inventory level of pharmacy i is defined as I_{i0}. The demand faced by pharmacy i in period t is represented by the random variable D_{it}, which is realized at the end of period t. This demand is stochastic, with expected value μ_{it}.

The central warehouse is assumed to have sufficient inventory to meet all planned deliveries, and each pharmacy $i \in \mathcal{N}'$ has a maximum storage capacity $\bar{U}_i$. The parameters h_i and z_i define the unit holding and stockout costs of pharmacy i. A fleet of vehicles, denoted by $\mathcal{K}$, is responsible for carrying out the planned deliveries. Each vehicle $k \in \mathcal{K}$ has a given capacity, Q, and is subject to a maximum route duration, denoted by ε. Vehicles are allowed to do up to one route per period, returning empty to the warehouse. The objective is to minimize the total inventory holding and stockout costs for the planning horizon.

4.3.3.2 State (S_t)

A state comprises all information required to make a decision. We denote the state at the beginning of period $t \in \mathcal{T}$ as S_t, which includes the current inventory levels of each pharmacy i (I_{it}). Accordingly, the state at the beginning of period t can be defined as: $S_t = (I_{it}) \ \forall\, i \in \mathcal{N}'$. The initial state is given by: $S_0 = (I_{i0}) \ \forall\, i \in \mathcal{N}'$.

4.3.3.3 Decision (a_t)

A decision at period t is defined by a_t and comprises four components that disclose inventory and routing decisions. The inventory component of each pharmacy i is modeled by two decision variables: the replenishment quantity depicted by q_{ikt}, and the post-delivery inventory I^a_{it}, so that $I^a_{it} = I_{it} + \sum_k q_{ikt}$. The routing component is defined by the binary variables y_{ikt} and x_{ijkt}, where x_{ijkt} indicates whether the arc (i, j) is traversed during the planning route of vehicle k at period t, and y_{ikt} equals one if pharmacy i is visited in period t by vehicle k. Let us consider $y_t = (y_{ikt}) \ \forall\, i \in \mathcal{N}, k \in \mathcal{K}$, $q_t = (q_{ikt}) \ \forall\, i \in \mathcal{N}', k \in \mathcal{K}$, $I^a_t = (I^a_{it}) \ \forall\, i \in \mathcal{N}'$ and $x_t = (x_{ijkt}) \ \forall\, (i, j) \in \mathcal{N}, k \in \mathcal{K}$. Accordingly, a decision $a_t = (q_t, I^a_t, y_t, x_t)$ at period t is valid if the following constraints are satisfied:

$$I^a_{it} = I_{it} + \sum_{k \in \mathcal{K}} q_{ikt}, \ \forall\, i \in \mathcal{N}' \tag{4.1}$$

$$I_{it} + \sum_{k \in \mathcal{K}} q_{ikt} \leq \bar{U}_i, \ \forall\, i \in \mathcal{N}' \tag{4.2}$$

$$q_{ikt} \leq Q\, y_{ikt}, \ \forall\, i \in \mathcal{N}', k \in \mathcal{K} \tag{4.3}$$

$$\sum_{i \in \mathcal{N}'} q_{ikt} \leq Q\, y_{0kt}, \ \forall\, k \in \mathcal{K} \tag{4.4}$$

$$\sum_{j \in \mathcal{N}: j \neq i} x_{jikt} = y_{ikt}, \ \forall\, i \in \mathcal{N}, k \in \mathcal{K} \tag{4.5}$$

$$\sum_{j \in \mathcal{N}: j \neq i} x_{ijkt} = y_{ikt}, \ \forall\, i \in \mathcal{N}, k \in \mathcal{K} \tag{4.6}$$

$$\sum_{j \in \mathcal{N}'} x_{0jkt} = \sum_{i \in \mathcal{N}'} x_{i0kt}, \ \forall\, k \in \mathcal{K} \tag{4.7}$$

$$\sum_{i \in \mathcal{B}} \sum_{j \in \mathcal{B}: j \neq i} x_{ijkt} \leq \sum_{i \in \mathcal{B}} y_{ikt} - y_{\ell kt}, \ \forall\, \mathcal{B} \subseteq \mathcal{N}', |\mathcal{B}| \geq 2, k \in \mathcal{K}, \ell \in \mathcal{B} \tag{4.8}$$

$$\sum_{(i,j) \in \mathcal{E}} \tau_{ij}\, x_{ijkt} + \sum_{i \in \mathcal{N}'} \delta\, y_{ikt} \leq \varepsilon, \ \forall\, k \in \mathcal{K} \tag{4.9}$$

$$\sum_{k \in \mathcal{K}} y_{ikt} \leq 1, \ \forall\, i \in \mathcal{N}' \tag{4.10}$$

$$I^a_{it}, q_{ikt} \in \mathbb{Z}^+_0, \ \forall\, i \in \mathcal{N}', k \in \mathcal{K} \tag{4.11}$$

$$y_{ikt} \in \{0, 1\}, \ \forall\, i \in \mathcal{N}, k \in \mathcal{K} \tag{4.12}$$

$$x_{ijkt} \in \{0, 1\}, \ \forall\, (i, j) \in \mathcal{E}, k \in \mathcal{K} \tag{4.13}$$

Constraints (4.1) balance the inventory levels at each pharmacy $i \in \mathcal{N}'$. Constraints (4.2) ensure that the inventory level of each pharmacy $i \in \mathcal{N}'$ does not exceed its storage capacity $\bar{U}_i$. Constraints (4.3) link the delivery quantity and visit decision variables. Constraints (4.4) guarantee that the total delivery quantities assigned to vehicle k do not exceed its capacity Q. Constraints (4.5) - (4.6) ensure the vehicle's flow conservation. The requirement that a vehicle start and complete its route at the warehouse is ensured by Constraints (4.7). Constraints (4.8) refer to the subtour elimination constraints. The maximum route duration is ensured by Constraints (4.9). Constraints (4.10) limit the number of visits to each pharmacy in each period. Constraints (4.11) - (4.13) define the domain of the decision variables.

The combination of S_t and a_t leads to the post-decision state S_t^a, where inventories are updated according to the initial inventories and delivery quantities, so that $S_t^a = (I_t^a)$.

4.3.3.4 Stochastic information (ω_{t+1}) and transition function (Γ)

The stochastic information $\omega_{t+1} = (D_t^\omega)$ discloses the demand at each pharmacy, with $D_t^\omega = (D_{it}^\omega)\ \forall\, i \in \mathcal{N}'$ as an $|N|$-dimensional vector. A transition function Γ determines how the system evolves from the post-decision state S_t^a to the next state S_{t+1}. This function captures the impact of stochastic demand realizations. Therefore, the combination of S_t^a and information realization ω_{t+1} results in a new state $S_{t+1} = \Gamma(S_t^a, \omega_{t+1})$. The state variables are updated as follows:

$$I_{it+1} = I_{it}^a - D_{it}^\omega + l_{it},\ \forall\, i \in \mathcal{N}' \tag{4.14}$$

$$l_{it} = \max(0, D_{it}^\omega - I_{it}^a),\ \forall\, i \in \mathcal{N}' \tag{4.15}$$

The resulting cost function has two components: holding and stockout (lost sales) costs:

$$C(S_t^a, \omega_{t+1}) = \sum_{i \in \mathcal{N}'} h_i\, I_{it+1} + z_i\, l_{it} \tag{4.16}$$

4.3.3.5 Policy (π)

A solution for our SDIRP is a policy $\pi \in \Pi$ that assigns every state S_t to a decision $a_t = A^\pi(S_t)$ at a particular period t of the planning horizon $\mathcal{T}$, with $A^\pi(S_t)$ being the feasible decision space. For a given state S_t, each decision a_t is associated with a (negative) reward representing the holding and stockout costs. We aim to find a policy that minimizes the expected total costs, given an initial state S_0:

$$\pi = \arg\min_{\pi \in \Pi} \mathbb{E}\left[\sum_{t \in \mathcal{T}} C(S_t^a, \omega_{t+1}) \mid S_0\right] \tag{4.17}$$

4.4. Solution method

In the following, we detail our problem-solving methodology. We first motivate our decision framework and provide a general overview (Section 4.4.1). Second, we present our **CFA** policy, focusing on the proposed parametrically modified objective function, and then discuss its components (Section 4.4.2). Third, we describe our BO algorithm for learning the policy parameters (Section 4.4.3).

4.4.1 Motivation and overview

The SDIRP brings together challenges from two problem domains, inventory management and vehicle routing, both in a stochastic dynamic environment. Inventory management aims to balance holding and stockout costs, while vehicle routing seeks efficient routes to satisfy today's and future demand. To successfully tackle the SDIRP, both domains have to be considered thoroughly. However, they are inherently interdependent. Even the small example in Figure 4.1 illustrates the challenges and their trade-offs. When today's inventory levels are optimized, tomorrow's routing resources may not be able to handle the resulting consequences. If the focus lies on efficient routing today and consolidation potential tomorrow, some inventory levels may not be satisfactory. So, how to find effective, anticipatory decisions that consider both parts, inventory and routing, in an integrated manner? In particular, how to embed today's and future routing limitations in effective inventory management strategies?

For inventory management, we propose employing a conventional policy that seeks to satisfy an individual target inventory level for each pharmacy (I_i^*) in every period. To search the routing space, we use a mathematical optimization solver. Since routing constraints may prevent all target levels from being attained, our policy penalizes stock deviations in a non-linear (quadratic) fashion $(I_i^* - I_{it}^a)^2$. This policy may lead to the decision shown in Figure 4.1(a), where pharmacies all over the service area may require delivery the next day, which may be difficult to accommodate. Therefore, we propose integrating future routing considerations in the evaluation of today's decisions. The idea is to introduce a routing-aware inventory synchronization strategy into current decision-making, encouraging pharmacies located in the same region to reach similar post-delivery inventory positions, thereby increasing the opportunity to serve them jointly on the same route. Given the curse of dimensionality, we use an approximation based on the pairwise travel distances between pharmacies (d_{ij}) and their respective post-delivery inventory levels (I_{it}^a and I_{jt}^a), as expressed in Equation (4.18), cf. Section 4.4.2. In the framework of ADP, our method is classified as a parametric CFA that accounts for the problem's future dynamics.

4.4.2 Cost function approximation

Building on Powell (2022a)'s unified framework for sequential decision-making, we propose a policy that focuses on making informed current-period decisions, while capturing the effects of future demand uncertainty and geographical interdependencies through a parameterized mathematical programming model with a modified objective function. Our

CFA must capture inventory and routing dynamics simultaneously. First, we anticipate future demand uncertainty by setting an inventory target level for each pharmacy, which represents the desired inventory position in each period. Second, future routing limitations are likewise anticipated, acknowledging that their impacts propagate over time and depend on uncertain demand realizations (Bellman, 1956).

Our policy minimizes the sum of two inventory-related components for each period $t \in \mathcal{T}$ in an RH fashion, as shown in Equation (4.18). The first component tracks the squared difference between each pharmacy's learned inventory target level (I_i^*) and its post-decision inventory (I_{it}^a), ensuring that larger deviations result in more significant penalties compared to minor fluctuations. The second component addresses inventory discrepancies between pharmacies with greater emphasis on geographically close pairs (i.e., low d_{ij}) by considering the expected days of stock (I_{it}^a/μ_{it}) scaled by the corresponding pairwise distance (d_{ij}). Therefore, this component encourages geographically proximate pharmacies to have similar inventory coverage, thereby promoting the synchronization of their deliveries. A penalty weight (λ) controls the trade-off between achieving individual target levels and maintaining local inventory stability. The decision space is defined in Section 4.3.3.3.

$$\min \sum_{i \in \mathcal{N}'} \left| I_i^* - I_{it}^a \right|^2 + \lambda \sum_{\substack{(i,j) \in \mathcal{E}: \\ i,j \in \mathcal{N}', i<j}} \frac{1}{d_{ij}} \left| \frac{I_{it}^a}{\mu_{it}} - \frac{I_{jt}^a}{\mu_{jt}} \right| \tag{4.18}$$

$$\text{s.t. Constraints (4.1) - (4.13)} \tag{4.19}$$

The proposed **CFA** provides us with integrated inventory and routing decisions that satisfy the operational requirements of the pharmaceutical distribution network, including pharmacy storage capacities, vehicle capacities, service times, and route operating time limits. Our policy is guided by scenario-dependent parameters (I_i^*, λ), dynamically adapting decisions to varying levels of uncertainty and vehicle capacity while avoiding the increased computational cost of solving a multi-period model. Accordingly, decisions are simulated, demand is revealed, and the resulting current-period holding and stockout costs are computed. This process operates in an RH fashion, allowing the policy to adjust inventory and routing decisions as time evolves.

4.4.3 Bayesian optimization algorithm

To determine effective policy parameter values (I_i^*, λ) and efficiently explore the solution space, we resort to BO (Wang et al., 2023). This method is particularly well-suited to large and continuous combinatorial search spaces, as in our case, facilitating the efficient optimization of computationally expensive objective functions in stochastic environments (Frazier, 2018). The search space for the two policy parameters, and the numerical settings of the training procedure are specified in Section 4.5.3. For each candidate pair (I_i^*, λ), the inventory target levels (I_i^*) are defined as a common percentage of each pharmacy's storage capacity. The BO algorithm is implemented using a Gaussian Process surrogate model

and an Expected Improvement acquisition function. This approach effectively guides the selection of new parameter values by balancing exploration and exploitation.

Our training set comprises n_{train} problem instances across six scenario configurations, defined by three levels of demand uncertainty and two vehicle capacity settings. Each instance contains T demand realizations. The training stage is conducted offline. We perform n_{iter} training iterations over the n_{train} instances, yielding n_{iter} successive updates of (I_i^*, λ). For example, higher uncertainty may justify higher inventory target levels to reduce stockout risk. Similarly, greater vehicle capacity increases delivery flexibility, potentially making higher target levels beneficial due to stockout costs. Following training, the best-performing candidate pair is retained for each scenario $s \in \mathcal{S}$, thereby parameterizing the proposed **CFA** policy.

Figure 4.2 summarizes the training procedure for our policy (more details on Appendix 4.A). Whenever a state is observed, the optimization model is solved using a mathematical heuristic solver (Step I). After computing the inventory and routing decisions, they are simulated for the current period. Once demand is realized, the system transitions to a new state, where inventories are updated, and holding and stockout costs are calculated. An instance is considered fully simulated once its T corresponding demand realizations, representing T periods, have been completed. A training iteration is completed after simulating all n_{train} training instances and evaluating the current candidate pair (I_i^*, λ) using the aggregated inventory holding and stockout costs. At that point, the BO algorithm updates the Gaussian Process surrogate model and selects a new candidate pair (I_i^*, λ) to be evaluated in the next training iteration (Step II).

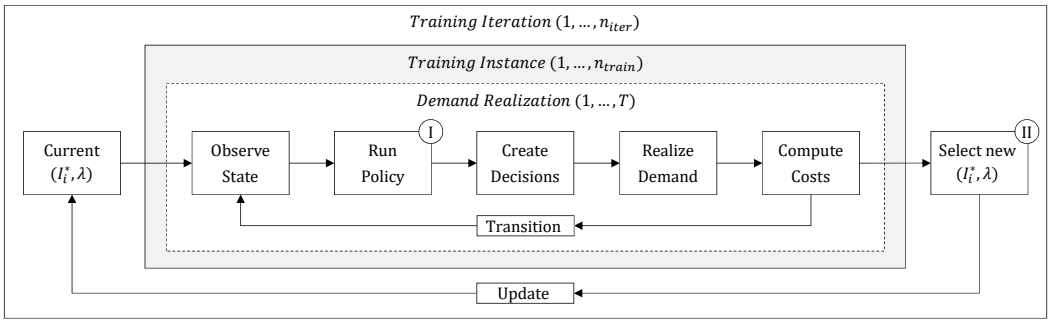


Figure 4.2: Training procedure of our **CFA** policy.

4.5. Experimental setting

In this section, we discuss the instance setting (Section 4.5.1), benchmark policies (Section 4.5.2), and implementation details (Section 4.5.3).

4.5.1 Instance setting

The instance setting is based on empirical data from a pharmaceutical distributor operating in the Algarve region of southern Portugal, where a central warehouse supplies several ge-

ographically dispersed pharmacies. Within the provided dataset, we identified a particular medicine with complete and consistent records from January 2023 to March 2023. During this period, we collected data on delivery quantities, sales figures, expected demand, and inventory-related costs. To complement the distributor's operational data, we incorporate additional inputs, including vehicle fleet details, initial inventory levels, pharmacy storage capacities, and stockout costs, all sourced from the well-established inventory-routing literature (Coelho et al., 2012).

Layout. Our instances comprise 25 nodes (one central warehouse and 24 pharmacies), a single product, and a planning horizon of 20 periods ($N = 24$, $T = 20$). The coordinates of the central warehouse and pharmacies are provided in the format of (latitude, longitude). Figure 4.3 illustrates the corresponding distribution network.

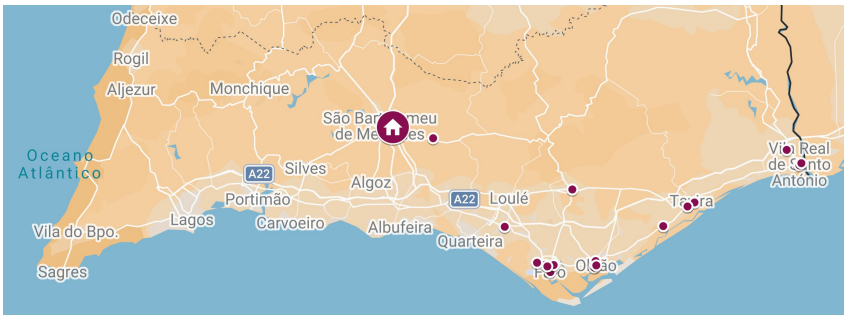


Figure 4.3: Distribution network including the central warehouse and 24 pharmacies in southern Portugal.

Vehicles. We consider two vehicle capacity scenarios, small and large, defined based on aggregate expected demand volumes, following standard assumptions from the inventory-routing literature. Specifically, the capacity of each vehicle is set to $Q = \alpha \sum_i \bar{\mu}_i$ product units, where $\alpha = 0.5$ corresponds to the small-capacity scenario and $\alpha = 1.0$ to the large-capacity scenario, and $\bar{\mu}_i = \sum_{t \in \mathcal{T}} \mu_{it}/T$ denotes the average expected demand of pharmacy i. The fleet consists of two homogeneous vehicles, each restricted to a single route per period, with a five-hour operating time limit and a 15-minute service time at each pharmacy, reflecting realistic operational conditions. Travel times are calculated based on Euclidean distances, considering the limited impact of traffic congestion in the current problem, assuming an average travel speed of 50 kilometers per hour.

Demand. Pharmacies experience periodic demand, which we model using a lognormal probability distribution based on real-world data. The expected demand of pharmacy i in period t, denoted by μ_{it}, is time-dependent and varies by weekday. Figure 4.4 illustrates the evolution of the average expected demand across the pharmacy network over one week (Monday to Sunday). This weekly pattern repeats every seven periods throughout the 20-period planning horizon (y-axis values are masked for confidentiality reasons). Although each pharmacy has a distinct expected demand per period, differing in magnitude, all pharmacies follow a synchronized weekly pattern. For example, demand gradually increases during the week and declines significantly on Sunday. The planning horizon starts

on a Monday (period 1) and ends on a Saturday (period 20). We investigate three levels of demand uncertainty: low, medium, and high, controlled via the standard deviation $\sigma_{it} = \{0.25, 0.50, 0.75\}\mu_{it}$. Each pharmacy begins the planning horizon with an initial inventory proportional to its storage capacity, defined as $I_{i0} \sim \mathcal{U}(0.3, 0.5)\,\bar{U}_i$, with $\bar{U}_i = 5\,\bar{\mu}_i$ corresponding to approximately five days of demand coverage.

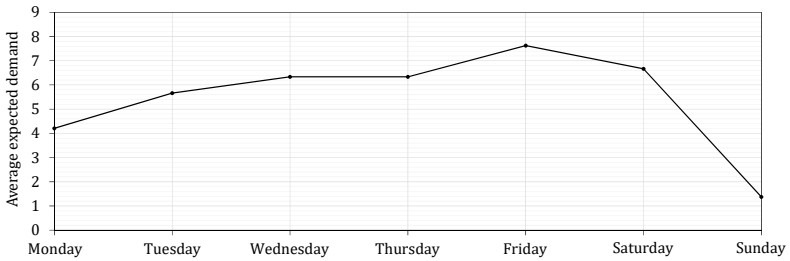


Figure 4.4: Average expected demand across the pharmacy network for each weekday (Monday-Sunday).

Costs. For each pharmacy, we consider a unit holding cost h_i (MU) applied to each unit of inventory that carries over from one period to the next, based on values observed in practice. The unit stockout cost z_i (MU) represents the cost incurred for each unit of unmet demand and is defined proportionally to the holding cost, so that $z_i = 200\,h_i$, following Coelho et al. (2012) and reflecting the critical importance of service levels in pharmaceutical distribution.

4.5.2 Benchmarks

We evaluate the performance of our proposed **CFA** policy by comparing it to five benchmark policies. These are categorized into practice-oriented (**CR-OU**, **OU** and **SB**) and method-oriented ($\mathbf{CFA_{LD}}$ and $\mathbf{CFA_{QD}}$). The same mathematical solver is used for all policies.

CR-OU. This policy reflects a standard industry practice that prioritizes operational stability through **consistent routes (CR)** paired with an **order-up-to-level (OU)** inventory rule. By grouping pharmacies into fixed clusters and visiting them on a regular schedule, the distributor reduces operational complexity at the expense of daily routing flexibility. This benchmark is conceptually aligned with Cui et al. (2023), in the sense that vehicle routes are fixed *a priori* for the entire planning horizon, and replenishment quantities are then computed periodically. For each vehicle capacity scenario (small and large), we define a distinct set of four routes ($r \in \mathcal{R}$) that jointly cover the 24 pharmacies (with each pharmacy i assigned to exactly one route), aligned with our two-vehicle fleet and an average replenishment frequency of two days per location, while ensuring that expected demand and routing constraints are satisfied. In each period t, a priority score is computed for each route r as: $\sum_{i \in \mathcal{N}_r}(-I_{it}/\mu_{it})\ \forall\, r \in \mathcal{R}$, where $\mathcal{N}_r$ denotes the set of pharmacies assigned to route r. The two highest-priority routes are performed, and the pharmacies assigned to each route are visited in the predefined order. The corresponding delivery quantities are determined by solving:

$\min \sum_{i \in \mathcal{N}'} (\bar{U}_i - I^a_{it})$, s.t. Constraints (4.1) - (4.13).

OU. This policy replaces fixed, periodic routes with dynamic multi-period routing. It serves as a critical reference point to determine how much of the system's performance is driven by the ability to change visit decisions and routing sequences daily. The policy jointly optimizes the current period's routing and inventory decisions in an RH fashion, according to: $\min \sum_{i \in \mathcal{N}'} (\bar{U}_i - I^a_{it})$, s.t. Constraints (4.1) - (4.13).

SB. This policy directly incorporates economic trade-offs, namely inventory holding and stockout costs, into the optimization process. Unlike the rule-based **OU** policies, the **safety-buffer (SB)** approach serves as a stochastic, information-augmented baseline. It evaluates the performance gains from incorporating demand forecasting into the joint inventory-routing framework, prioritizing pharmacies based on the expected holding and stockout costs. This approach relies on a demand estimate that includes an individual safety buffer for each pharmacy to hedge against uncertainty, defined as: $\hat{D}_{it} = \exp(\mu^{log}_{it} + k_i \sigma^{log}_{it})$, where $\mu^{log}_{it} = \ln(\mu_{it}) - \frac{\left(\sigma^{log}_{it}\right)^2}{2}$, $\sigma^{log}_{it} = \sqrt{\ln\left(1 + \frac{\sigma^2_{it}}{\mu^2_{it}}\right)}$ and $k_i = \Phi^{-1}\left(\frac{z_i}{z_i + h_i}\right)$ captures the stockout-holding cost trade-off. Through a reactive RH procedure, **SB** minimizes the expected current-period inventory costs: $\sum_{i \in \mathcal{N}'} h_i \hat{I}_{it+1} + z_i \hat{l}_{it}$, s.t. Constraints (4.1) - (4.13), where $\hat{I}_{it+1} = I^a_{it} - \hat{D}_{it} + \hat{l}_{it}$, and $\hat{l}_{it} = \max(\hat{D}_{it} - I^a_{it}, 0) \; \forall \, i \in \mathcal{N}'$.

CFA$_{\mathbf{QD}}$. This policy isolates the impact of the local inventory synchronization strategy of our proposed **CFA** policy and thereby serves as an ablation baseline. In each period of the planning horizon, we solve an integrated optimization problem that quadratically penalizes deviations between each pharmacy's post-delivery inventory and its corresponding optimized target level: $\min \sum_{i \in \mathcal{N}'} |I^*_i - I^a_{it}|^2$, s.t. Constraints (4.1) - (4.13). The quadratic structure ensures that larger deviations are penalized more severely. The target levels (I^*_i) for the entire network are learned offline via BO.

CFA$_{\mathbf{LD}}$. This policy serves as a second ablation baseline by replacing the quadratic penalty with a linear one: $\min \sum_{i \in \mathcal{N}'} |I^*_i - I^a_{it}|$, s.t. Constraints (4.1) - (4.13). Similar to **CFA$_{\mathbf{QD}}$**, the inventory targets (I^*_i) are learned offline via BO. This ensures that any performance difference between **CFA$_{\mathbf{LD}}$** and **CFA$_{\mathbf{QD}}$** can be attributed exclusively to the penalty structure (linear vs. quadratic) rather than the target values themselves.

4.5.3 Implementation details

A simulation environment was developed in Java 17.0.3 to replicate the stochastic dynamic problem setting. The decision spaces were searched using Hexaly 13.5, a mathematical solver that combines several exact and heuristic optimization methods, well-suited to challenging combinatorial and non-linear problems that are often difficult for classical approaches. To ensure computational feasibility while preserving solution quality, we imposed time limits of 5 seconds per period during training and 10 seconds during testing. The BO algorithm was implemented in Python 3.9. All experiments were conducted on a workstation running Windows 10, equipped with an AMD Ryzen Threadripper PRO 3995WX 64-core CPU @ 2.70 GHz (up to 16 cores used) and 48 GB of RAM.

Independent BO training procedures were conducted for each of the three policies, namely **CFA**, $\mathbf{CFA_{LD}}$, and $\mathbf{CFA_{QD}}$. Each procedure was performed offline for $n_{iter} = 100$ iterations, using $n_{train} = 30$ instances (i.e., five per scenario, generated with different random seeds), each with a planning horizon of $T = 20$ periods. Based on preliminary experiments, the search space for the proposed policy parameters was defined as follows: inventory target levels (I_i^*) range from 50% to 100% of each pharmacy's storage capacity, with the same percentage applied across all pharmacies, i.e., $I_i^* \in [0.5\,\bar{U}_i, 1.0\,\bar{U}_i]$, while the penalty weight (λ) ranges within $[0.0, 14.0]$. The test set comprises $n_{test} = 60$ instances (i.e., 10 per scenario, generated with different random seeds).

4.6. Computational results

In this section, we first compare the performance of all policies, with particular emphasis on inventory holding and stockout costs (Section 4.6.1). Second, we provide a detailed analysis of our **CFA** policy, focusing on how the local inventory synchronization strategy leads to more informed and effective replenishment decisions, ultimately reducing stockouts (Section 4.6.2). Third, we evaluate the impact of different levels of demand uncertainty and vehicle capacity settings on policy performance (Section 4.6.3). Furthermore, we conduct a sensitivity analysis to assess how these problem parameters affect the learned values of (I_i^*, λ) (cf. Appendix 4.B).

4.6.1 Solution quality

All policies are assessed over the full test set. Let C^{CFA} denote the total cost incurred by **CFA**, and C^{π} the total cost incurred by the baseline policy $\pi \in \{\mathbf{CR\text{-}OU}, \mathbf{OU}, \mathbf{SB}, \mathbf{CFA_{LD}}, \mathbf{CFA_{QD}}\}$. The cost reduction (in %) achieved by **CFA** relative to baseline π is computed as:

$$\frac{C^{\pi} - C^{CFA}}{C^{\pi}} \times 100\% \tag{4.20}$$

Additional performance metrics consider the number of delivery stops, delivery quantities, and stockout quantities. Equation (4.20) is equally applied to these metrics.

Table 4.1 summarizes the operational performance of each baseline compared to **CFA**. All metrics are aggregated over all six scenario configurations to provide a global assessment of policy behavior (disaggregated results are analyzed in Section 4.6.3). Positive values indicate reductions achieved by **CFA** relative to the baseline.

Focusing first on the baselines that most closely align with standard pharmaceutical replenishment practices, **CR-OU**, characterized by limited routing flexibility, results in both a high frequency and significant magnitude of stockouts. Consequently, it incurs the highest total inventory costs among all policies, making it the least cost-efficient benchmark. In contrast, the dynamic multi-period routing in **OU** contributes to more flexible visiting decisions, significantly reducing stockout costs by 23 percentage points (pp) compared to **CR-OU**, despite a 23 pp increase in holding costs due to more frequent delivery stops and larger delivery quantities.

Table 4.1: Overall comparison with the proposed **CFA** policy.

Baseline policy	Holding cost (in %)	Stockout cost (in %)	Total cost (in %)	Delivery		Stockout quantity (in %)
				stops (in %)	quantity (in %)	
CR-OU	-2.6	70.3	59.9	-60.7	-4.3	67.6
OU	20.4	47.3	39.9	-19.3	2.3	47.9
SB	-19.8	49.8	36.2	-15.6	-8.6	77.1
CFA_{LD}	7.9	33.5	25.9	-6.3	0.5	34.5
CFA_{QD}	4.5	7.9	6.7	-0.6	0.8	7.0

The **SB** policy is characterized by more predictive, controlled deliveries, minimizing immediate holding and stockout costs while explicitly considering expected demand and a safety stock buffer during the joint inventory-routing optimization. Although **SB** results in slightly more delivery stops than **OU**, it yields lower delivery quantities, demonstrating the impact of the current inventory replenishment strategy. As a result, **SB** achieves the lowest holding costs among all considered policies, which is sufficient to reduce total costs by 3.7 pp compared to **OU**, even with a marginal increase in stockout costs.

Both method-oriented baselines (**CFA_{LD}** and **CFA_{QD}**) are rooted in an inventory target level for each pharmacy (I_i^*) that acts as a proxy for "optimal" inventory. This approach encourages post-delivery inventories to align with each pharmacy's target inventory position in each period, rather than its storage capacity ($\bar{U}_i$), as in **OU**. Consequently, **CFA_{LD}** results in a significant decrease in stockout quantities and costs relative to the previous three baselines. By anticipating uncertainty more effectively, it lowers total costs by 10.3 pp compared to **SB** and by 14.0 pp compared to **OU**. These benefits become even more prominent when inventory target deviations are penalized quadratically rather than linearly. Therefore, **CFA_{QD}** achieves a noteworthy 19.2 pp reduction in total cost relative to **CFA_{LD}**.

Finally, the proposed **CFA** policy incorporates a geographically-aware mechanism that encourages proximate pharmacies to maintain synchronized (I_{it}^a/μ_{it}) ratios. This coordination results in a 6.7% improvement over **CFA_{QD}**, promoting more informed and proactive delivery decisions. A further discussion on the importance of local inventory synchronization is presented in the following section.

4.6.2 The value of local inventory synchronization

As illustrated in Figure 4.4, there is a clear pattern in daily demand, increasing along the week and significantly declining on Sunday. The impact of these fluctuations is captured in Figures 4.5 and 4.6, which represent the two objective function components of the proposed **CFA** policy: inventory target deviations and proximity-weighted inventory imbalance, respectively. All figures in this subsection present aggregated results over the six scenario configurations. According to Figure 4.5, both **CFA_{QD}** and **CFA** experience increased difficulty in reaching the optimized target levels following high-demand periods. Conversely, after Sunday and, to a lesser extent, after Monday, it becomes easier to reach the targets, although not completely due to the routing constraints.

The cost function structure of each policy shows that the resulting inventory target deviations are consistently larger in **CFA** than in $\mathbf{CFA_{QD}}$ (cf. Figure 4.5), as the former must simultaneously balance two objectives rather than focusing exclusively on aligning inventories with optimized individual target levels. This difference becomes more evident on Sundays, when the low expected demand (μ_{it}) increases the significance of the inventory synchronization component $\left(\left|\frac{I^a_{it}}{\mu_{it}} - \frac{I^a_{jt}}{\mu_{jt}}\right|\right)$. During these periods, **CFA** promotes geographically coordinated inventory positions across the network, as shown in Figure 4.6, thereby better preparing pharmacies for subsequent periods.

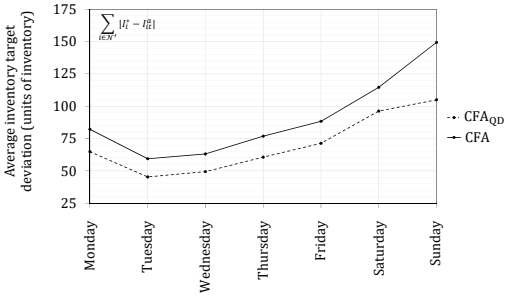


Figure 4.5: Average inventory target deviation by $\mathbf{CFA_{QD}}$ and **CFA**.

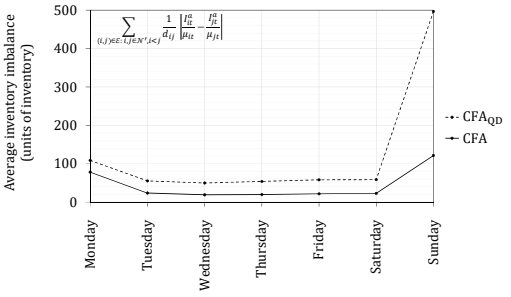


Figure 4.6: Average proximity-weighted inventory imbalance by $\mathbf{CFA_{QD}}$ and **CFA**.

The cost performance of both policies is observed in Figures 4.7 and 4.8, where **CFA** consistently achieves lower holding and stockout costs than the $\mathbf{CFA_{QD}}$ baseline. More than the total number of visits, delivered quantities, or vehicle utilization (93.3% vs. 93.2% for small and 54.1% vs. 54.9% for large vehicle-capacity settings, for **CFA** and $\mathbf{CFA_{QD}}$ respectively), the main advantage of **CFA** over $\mathbf{CFA_{QD}}$ lies in more informed and coordinated replenishment quantities and inventory positioning, reflected in lower stockout costs. The penalty weight (λ) controls the trade-off between the two objectives, ensuring global network stability and preventing the inventory imbalances observed in $\mathbf{CFA_{QD}}$, where pharmacies develop inconsistent stock levels more often over the planning horizon.

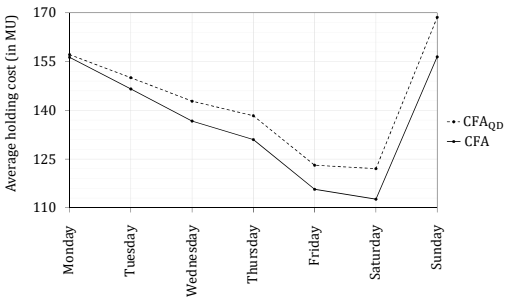


Figure 4.7: Average holding costs by $\mathbf{CFA_{QD}}$ and **CFA**.

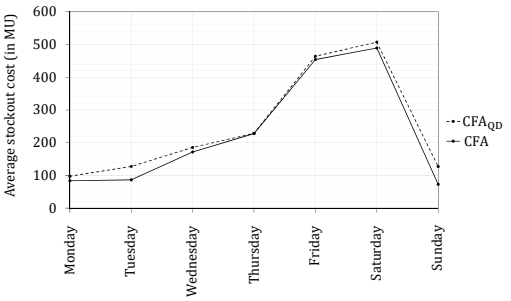


Figure 4.8: Average stockout costs by $\mathbf{CFA_{QD}}$ and **CFA**.

4.6.3 Instance data analysis

To better understand the impact of demand uncertainty and vehicle capacity on policy performance, we disaggregate the results across the six scenario configurations. Table 4.2 reports the improvements in total inventory costs achieved by the proposed **CFA** policy relative to each baseline, using the same metric as in Section 4.6.1. The results reveal that the cost improvements are highly sensitive to the interplay between available delivery capacity and demand volatility.

Table 4.2: Scenario-wise comparison with the proposed **CFA** policy.

Metric Gap to **CFA** (in %)	Instance scenario		Baseline policy				
	Uncertainty	Vehicle cap.	CR-OU	OU	SB	CFA_{LD}	CFA_{QD}
Holding cost	Low	Small	19.4	30.2	-110.1	9.2	-0.1
		Large	-11.1	24.4	-199.8	10.2	0.8
	Medium	Small	16.1	23.6	1.0	4.5	2.2
		Large	-8.6	24.2	-21.0	19.0	16.2
	High	Small	9.9	20.2	28.7	5.7	-1.2
		Large	-36.7	4.1	-5.0	-2.5	3.7
	Average		-1.8	21.1	-51.0	7.7	3.6
Stockout cost	Low	Small	59.7	61.3	65.6	42.7	18.2
		Large	96.9	-15.0	94.9	67.7	55.8
	Medium	Small	56.6	57.5	47.2	39.8	6.4
		Large	89.2	-73.8	37.8	-6.9	-67.1
	High	Small	40.6	39.0	18.0	25.4	5.6
		Large	86.2	16.9	12.7	24.4	2.8
	Average		71.5	14.3	46.0	32.2	3.6
Total cost	Low	Small	54.0	56.5	56.7	37.0	14.4
		Large	77.4	21.3	60.2	25.3	13.1
	Medium	Small	50.9	52.4	40.5	33.9	5.4
		Large	72.2	7.3	7.2	12.2	0.2
	High	Small	36.6	36.3	20.2	22.4	4.4
		Large	71.9	10.1	3.5	11.2	3.3
	Average		60.5	30.7	31.4	23.7	6.8

The **CR-OU** policy tends to perform worst, particularly in large vehicle capacity scenarios, with cost gaps ranging from 36.6% to 77.4%. Because it relies on fixed periodic delivery plans, it cannot exploit the additional flexibility provided by larger vehicle capacities to mitigate stockouts; indeed, the total cost gap increases with delivery capacity. For the remaining practice-oriented baselines, the improvements offered by **CFA** tend to narrow as demand uncertainty or vehicle capacity increases. In large-capacity scenarios, **OU** encourages replenishing a large number of pharmacies up to their respective storage capacities, thereby reducing stockouts, despite increasing holding costs, leading to significantly smaller total cost gaps (as low as 7.3% in medium-uncertainty conditions). The same does not occur in small-capacity scenarios.

In highly uncertain environments, **SB** reduces its performance gap with **CFA** (to 3.5% in large-capacity scenarios) more effectively than $\mathbf{CFA_{LD}}$. This suggests that incorporating holding and stockout costs into the optimization objective, while anticipating uncertainty through demand forecasting, is more effective at mitigating stockouts than the linear inventory target deviation strategy.

The differences between **CFA** and the method-oriented baselines ($\mathbf{CFA_{LD}}$ and $\mathbf{CFA_{QD}}$) are generally smaller in large-capacity scenarios. This suggests that greater routing flexibility reduces the value of synchronized deliveries, although **CFA** remains beneficial across all scenarios. Moreover, the local inventory synchronization mechanism proves beneficial, with its value being particularly evident in environments with lower demand uncertainty. The consistently positive gaps, which average 6.8% relative to $\mathbf{CFA_{QD}}$, indicate that coordinating inventory ratios (I^a_{it}/μ_{it}) across geographically close pharmacies provides an additional layer of protection against future routing limitations and demand fluctuations.

4.7. Conclusion

This paper addresses an SDIRP inspired by the operations of a Portuguese pharmaceutical distributor, considering a central warehouse, 24 geographically dispersed pharmacies, and a fleet of two vehicles. We propose a **CFA**-based policy that combines an inventory target level strategy accounting for future demand uncertainty with a local inventory synchronization strategy that incorporates future routing considerations. Our computational study shows that our **CFA** policy consistently outperforms both practice-oriented and method-oriented benchmarks across different experimental scenarios of demand uncertainty and delivery capacity. Both the inventory target levels for each pharmacy and a penalty weight that balances the two objectives are scenario-dependent and learned via BO. The proposed approach can also be extended to other application domains where inventory and routing decisions must be jointly optimized over time under uncertainty.

In our problem, we have assumed that a particular medicine must be delivered from a central warehouse to a set of pharmacies. However, there might be other products (e.g., with smaller demand but higher holding cost) that must be delivered as well. Thus, future work may consider extending our model and method to multi-product settings. Furthermore, while the medicine considered in this study was assumed to be available without any restrictions, others might have limited (and uncertain) supply. This may affect decision-making, as transporting large quantities to a given pharmacy in a given period may reduce flexibility in future periods. Finally, real-world distribution operations might benefit from greater operational flexibility than we assumed in our computational study. In particular, future research could quantify the benefits of considering split deliveries or transshipments between pharmacies. Such strategies might help mitigate local shortages and enhance overall service levels.

From a methodological standpoint, we introduced a routing-aware mechanism that bridges inventory control and future routing considerations. This is a general contribution that might be applied to other dynamic settings with limited routing resources. Furthermore, while we set the penalty weight globally, future work may investigate more ad-

vanced mechanisms. For example, penalty weights may be learned state-dependently by means of RL, shifting the focus between efficient routing and inventory coordination based on current state information.

Acknowledgements

This work was supported by FCT - Fundação para a Ciência e Tecnologia [reference: 2022.13466.BD and DOI identifier https://doi.org/10.54499/2022.13466.BD]; the European Union's Horizon 2020 Research and Innovation Programme [TRUST-AI research project, grant number 952060]; and the European Union's Horizon Europe Research and Innovation Programme [PEER research project, grant number 101120406].

Bibliography

Achamrah, F.E., Riane, F., Limbourg, S., 2022. Solving inventory routing with transshipment and substitution under dynamic and stochastic demands using genetic algorithm and deep reinforcement learning. International Journal of Production Research 60, 6187–6204. https://doi.org/10.1080/00207543.2021.1987549.

Agra, A., Christiansen, M., Hvattum, L.M., Rodrigues, F., 2018. Robust optimization for a maritime inventory routing problem. Transportation Science 52, 509–525. https://doi.org/10.1287/trsc.2017.0814.

Bellman, R., 1956. Dynamic programming and lagrange multipliers. Proceedings of the National Academy of Sciences 42, 767–769. https://doi.org/10.1073/pnas.42.10.767.

Berman, O., Larson, R.C., 2001. Deliveries in an inventory/routing problem using stochastic dynamic programming. Transportation Science 35, 192–213. https://doi.org/10.1287/trsc.35.2.192.10134.

Bertazzi, L., Chagas, G.O., Coelho, L.C., Laganà, D., Vocaturo, F., 2025. Online algorithms for the multi-vehicle inventory-routing problem with real-time demands. Transportation Research Part C: Emerging Technologies 170, 104892. https://doi.org/10.1016/j.trc.2024.104892.

Van den Bogaert, J., Van Jaarsveld, W., 2022. Vendor-managed inventory in practice: understanding and mitigating the impact of supplier heterogeneity. International Journal of Production Research 60, 6087–6103. https://doi.org/10.1080/00207543.2021.1983222.

Brinkmann, J., Ulmer, M.W., Mattfeld, D.C., 2020. The multi-vehicle stochastic-dynamic inventory routing problem for bike sharing systems. Business Research 13, 69–92. https://doi.org/10.1007/s40685-019-0100-z.

Campelo, P., Neves-Moreira, F., Amorim, P., Almada-Lobo, B., 2019. Consistent vehicle routing problem with service level agreements: A case study in the pharmaceutical distribution sector. European Journal of Operational Research 273, 131–145. https://doi.org/10.1016/j.ejor.2018.07.030.

Cavagnini, R., Bertazzi, L., Maggioni, F., 2022. A rolling horizon approach for a multi-stage stochastic fixed-charge transportation problem with transshipment. European Journal of Operational Research 301, 912–922. https://doi.org/10.1016/j.ejor.2021.11.037.

Chen, T., Chu, F., Zhang, J., Sun, J., 2024. Sustainable collaborative strategy in pharmaceutical refrigerated logistics routing problem. International Journal of Production Research 62, 5036–5060. https://doi.org/10.1080/00207543.2023.2283566.

Chiu, A., Angulo, G., Larrain, H., 2024. Optimizing the long-term costs of an inventory routing problem using linear relaxation. Transportation Research Part E: Logistics and Transportation Review 183, 103447. https://doi.org/10.1016/j.tre.2024.103447.

Coelho, L.C., Laporte, G., Cordeau, J.F., 2012. Dynamic and stochastic inventory-routing. CIRRELT Montreal.

Cornelissen, N., Zielhuis, S., van den Bemt, P., van den Bemt, B., 2026. Causes and management of drug shortages: a scoping review. Research in Social and Administrative Pharmacy 22, 232–253. https://doi.org/10.1016/j.sapharm.2025.10.008.

Crama, Y., Rezaei, M., Savelsbergh, M., Woensel, T.V., 2018. Stochastic inventory routing for perishable products. Transportation Science 52, 526–546. https://doi.org/10.1287/trsc.2017.0799.

Cuellar-Usaquén, D., Ulmer, M.W., Gomez, C., Álvarez-Martínez, D., 2024. Adaptive stochastic lookahead policies for dynamic multi-period purchasing and inventory routing. European Journal of Operational Research 318, 1028–1041. https://doi.org/10.1016/j.ejor.2024.06.011.

Cui, Z., Long, D.Z., Qi, J., Zhang, L., 2023. The inventory routing problem under uncertainty. Operations Research 71, 378–395. https://doi.org/10.1287/opre.2022.2407.

Frazier, P.I., 2018. A tutorial on bayesian optimization. arXiv preprint arXiv:1807.02811. https://doi.org/10.48550/arXiv.1807.02811.

Greif, T., Bouvier, L., Flath, C.M., Parmentier, A., Rohmer, S.U., Vidal, T., 2024. Combinatorial optimization and machine learning for dynamic inventory routing. arXiv preprint arXiv:2402.04463. https://doi.org/10.48550/arXiv.2402.04463.

Hasturk, U., Schrotenboer, A.H., Ursavas, E., Roodbergen, K.J., 2024. Stochastic cyclic inventory routing with supply uncertainty: A case in green-hydrogen logistics. Transportation Science 58, 315–339. https://doi.org/10.1287/trsc.2022.0435.

Infarmed, Autoridade Nacional do Medicamento e Produtos de Saúde, I., 2024. Relatório anual 2024: Gestão da disponibilidade de medicamentos. URL: https://www.infarmed.pt/documents/15786/10901776/Relat%C3%B3rio+2024/2f02f60e-ef04-a5b1-7962-be7d5ef353a2. (Accessed December 2, 2025).

Janatyan, N., Zandieh, M., Alem-Tabriz, A., Rabieh, M., 2025. A robust optimization model for sustainable pharmaceutical distribution network design: a case study. Annals of Operations Research 349, 911–930. https://doi.org/10.1007/s10479-020-03900-5.

Kramer, R., Cordeau, J.F., Iori, M., 2019. Rich vehicle routing with auxiliary depots and anticipated deliveries: An application to pharmaceutical distribution. Transportation Research Part E: Logistics and Transportation Review 129, 162–174. https://doi.org/10.1016/j.tre.2019.07.012.

Lagos, F., 2026. The stochastic and dynamic inventory routing problem with fine time granularity: A reinforcement learning approach. Computers & Operations Research 185, 107298. https://doi.org/10.1016/j.cor.2025.107298.

Lee, S.J., Kim, B.S., 2024. Vehicle routing and scheduling problem with order acceptance for pharmaceutical refrigerated logistics. Applied Soft Computing 164, 111983. https://doi.org/10.1016/j.asoc.2024.111983.

Li, R., Cui, Z., Kuo, Y.H., Zhang, L., 2023. Scenario-based distributionally robust optimization for the stochastic inventory routing problem. Transportation Research Part E: Logistics and Transportation Review 176, 103193. https://doi.org/10.1016/j.tre.2023.103193.

Liang, J., Liu, D., Jena, S.D., Lodi, A., Vidal, T., 2024. Dual policy reinforcement learning for real-time rebalancing in bike-sharing systems. arXiv preprint arXiv:2406.00868. https://doi.org/10.48550/arXiv.2406.00868.

Maia, F., Figueira, G., Neves-Moreira, F., 2026. Stochastic dynamic inventory-routing: A comprehensive review. Computers & Operations Research 189, 107383. https://doi.org/10.1016/j.cor.2026.107383.

Markov, I., Bierlaire, M., Cordeau, J.F., Maknoon, Y., Varone, S., 2020. Waste collection inventory routing with non-stationary stochastic demands. Computers & Operations Research 113, 104798. https://doi.org/10.1016/j.cor.2019.104798.

Martins, S., Amorim, P., Figueira, G., Almada-Lobo, B., 2017. An optimization-simulation approach to the network redesign problem of pharmaceutical wholesalers. Computers & Industrial Engineering 106, 315–328. https://doi.org/10.1016/j.cie.2017.01.026.

Mes, M., Schutten, M., Rivera, A.P., 2014. Inventory routing for dynamic waste collection. Waste Management 34, 1564–1576. https://doi.org/10.1016/j.wasman.2014.05.011.

Morales, F., Franco, C., Mendez-Giraldo, G., 2018. Dynamic inventory routing problem: Policies considering network disruptions. International Journal of Industrial Engineering Computations 9, 523–534. https://doi.org/10.5267/j.ijiec.2017.11.001.

Neves-Moreira, F., Almada-Lobo, B., Guimarães, L., Amorim, P., 2022. The multi-product inventory-routing problem with pickups and deliveries: Mitigating fluctuating demand via rolling horizon heuristics. Transportation Research Part E: Logistics and Transportation Review 164, 102791. https://doi.org/10.1016/j.tre.2022.102791.

Niakan, F., Rahimi, M., 2015. A multi-objective healthcare inventory routing problem; a fuzzy possibilistic approach. Transportation Research Part E: Logistics and Transportation Review 80, 74–94. https://doi.org/10.1016/j.tre.2015.04.010.

Peng, Y., Liao, W., 2025. Inventory routing optimization for pharmaceutical cold chain logistics from economic and environmental perspective. International Journal of Systems Science: Operations & Logistics 12, 2474690. https://doi.org/10.1080/23302674.2025.2474690.

Powell, W.B., 2022a. Designing lookahead policies for sequential decision problems in transportation and logistics. IEEE Open Journal of Intelligent Transportation Systems 3, 313–327. https://doi.org/10.1109/OJITS.2022.3148574.

Powell, W.B., 2022b. Reinforcement Learning and Stochastic Optimization: A Unified Framework for Sequential Decisions. John Wiley & Sons, Hoboken, NJ. https://doi.org/10.1002/9781119815068.

Silver, E.A., Pyke, D.F., Thomas, D.J., 2016. Inventory and Production Management in Supply Chains (4th ed.). CRC press. https://doi.org/10.1201/9781315374406.

Soysal, M., Çimen, M., 2017. A simulation based restricted dynamic programming approach for the green time dependent vehicle routing problem. Computers & Operations Research 88, 297–305. https://doi.org/10.1016/j.cor.2017.06.023.

Timajchi, A., Al-e Hashem, S.M.M., Rekik, Y., 2019. Inventory routing problem for hazardous and deteriorating items in the presence of accident risk with transshipment option. International Journal of Production Economics 209, 302–315. https://doi.org/10.1016/j.ijpe.2018.01.018.

Vonolfen, S., Affenzeller, M., Beham, A., Lengauer, E., Wagner, S., 2013. Simulation-based evolution of resupply and routing policies in rich vendor-managed inventory scenarios. Central European Journal of Operations Research 21, 379–400. https://doi.org/10.1007/s10100-011-0232-5.

Wang, X., Jin, Y., Schmitt, S., Olhofer, M., 2023. Recent advances in Bayesian optimization. ACM Computing Surveys 55, 1–36. https://doi.org/10.1145/3582078.

Zandkarimkhani, S., Mina, H., Biuki, M., Govindan, K., 2020. A chance constrained fuzzy goal programming approach for perishable pharmaceutical supply chain network design. Annals of Operations Research 295, 425–452. https://doi.org/10.1007/s10479-020-03677-7.

Appendix 4.A Bayesian optimization algorithm

Algorithm 4 introduces the logic behind the Bayesian optimizer used to learn the **CFA** policy parameters: (I_i^*, λ). For the method-oriented benchmark policies, $\mathbf{CFA_{LD}}$ and $\mathbf{CFA_{QD}}$, the Bayesian optimizer is used to learn the inventory target levels: (I_i^*). The algorithm begins by defining the search space for the policy parameters (line 1) and fitness function (line 2), as well as the number of training iterations (line 3), the surrogate model (line 4), the acquisition function (line 5), and the lists that will contain the evaluated parameters and corresponding fitness value (line 6). The learning process is described in lines 7-27. Finally, although training is performed jointly across the six experimental scenarios (30 training instances, with five instances per scenario), the algorithm returns both the best-performing parameterization for the full training set and the best-performing policy parameters for each individual scenario (line 28).

Algorithm 4 Bayesian optimization algorithm for (I_i^*, λ) tuning.

1: Initialize the search space for I_i^* and λ as continuous variables: $I_i^* \in [0.5, 1.0]\,\bar{U}_i$, and $\lambda \in [0.0, 14.0]$
2: Define the fitness (cost) function: $f(I_i^*, \lambda) \leftarrow$ Equation (4.16)
3: Define the maximum number of training iterations: $n_{iter} \leftarrow 100$
4: Initialize the Bayesian optimizer with a Gaussian Process surrogate model
5: Configure the Expected Improvement acquisition function
6: Define $X_{\text{evaluated}} \leftarrow []$, $Y_{\text{evaluated}} \leftarrow []$
7: **for** each training iteration $k \in \{1, ..., n_{iter}\}$ **do**
8: Request the next candidate $(I_i^*, \lambda)_k$ from the Bayesian optimizer
9: Initialize $\text{totalCost}_k \leftarrow 0$
10: **for** each training instance $n \in \{1, \ldots, n_{train}\}$ **do**
11: Initialize $\text{totalCost}_n \leftarrow 0$
12: **for** each period $t \in \mathcal{T}$ **do**
13: Observe state S_t
14: Apply the **CFA** policy with parameters $(I_i^*, \lambda)_k$
15: Simulate decision $a_t \leftarrow (q_t, I_t^a, y_t, x_t)$
16: Update inventories and transit to post-decision state S_t^a
17: Realize demand $\omega_{t+1} \leftarrow (D_{it}^{\omega})$, $\forall i \in \mathcal{N}'$
18: Update inventories and transit to state S_{t+1}
19: Calculate inventory holding and stockout costs for the current period t: $C(S_t^a, \omega_{t+1})$
20: Update total cost for training instance n: $\text{totalCost}_n \leftarrow \text{totalCost}_n + C(S_t^a, \omega_{t+1})$
21: **end for**
22: Update total cost for $(I_i^*, \lambda)_k$: $\text{totalCost}_k \leftarrow \text{totalCost}_k + \text{totalCost}_n$
23: **end for**
24: Compute $y_k \leftarrow \text{totalCost}_k$
25: Append $(I_i^*, \lambda)_k$ to $X_{\text{evaluated}}$ and y_k to $Y_{\text{evaluated}}$
26: Update the Gaussian Process surrogate model
27: **end for**
28: **return** the best (I_i^*, λ) policy parameters

Appendix 4.B The value of (adaptive) scenario-dependent policy parameters

Table 4.3 reports the scenario-specific policy parameters learned for **CFA$_{\mathbf{LD}}$**, **CFA$_{\mathbf{QD}}$**, and **CFA** across the six experimental scenarios considered in the computational study (refer to Section 4.5.1 for details). To ensure a fair comparison across scenarios, all three policies were trained independently, as discussed in Sections 4.4.3 and 4.5.3. For **CFA$_{\mathbf{LD}}$** and **CFA$_{\mathbf{QD}}$**, this results in six scenario-specific target levels (I_i^*), while for **CFA** it results in six scenario-specific (I_i^*, λ) pairs. For readability, the learned inventory target levels (I_i^*) are expressed as percentages of each pharmacy's storage capacity ($\bar{U}_i$).

Table 4.3: Learned policy parameters for each experimental scenario.

Demand uncertainty	Vehicle capacity	Policy parameters		
		CFA$_{\text{LD}}$ (I_i^*)	CFA$_{\text{QD}}$ (I_i^*)	CFA (I_i^*, λ)
Low	Small	74% $\bar{U}_i$	75% $\bar{U}_i$	(77% $\bar{U}_i$, 8.1)
	Large	89% $\bar{U}_i$	83% $\bar{U}_i$	(83% $\bar{U}_i$, 9.4)
Medium	Small	77% $\bar{U}_i$	82% $\bar{U}_i$	(80% $\bar{U}_i$, 5.9)
	Large	97% $\bar{U}_i$	94% $\bar{U}_i$	(83% $\bar{U}_i$, 9.4)
High	Small	80% $\bar{U}_i$	82% $\bar{U}_i$	(84% $\bar{U}_i$, 3.6)
	Large	97% $\bar{U}_i$	100% $\bar{U}_i$	(100% $\bar{U}_i$, 8.6)

The first trend we observe is a correlation between the inventory target levels and demand uncertainty. As the degree of uncertainty increases, each policy tends to be more conservative, establishing higher inventory buffers to protect pharmacies against potential stockouts. Furthermore, as vehicle capacity increases, all policies leverage this additional capacity to raise target levels, ensuring inventories remain adequate to mitigate stockouts, which are substantially more costly than holding inventory. Finally, by jointly learning both parameters, **CFA** balances each pharmacy's individual needs with the local inventory synchronization strategy, thereby adapting to each experimental scenario.

Chapter 5

Conclusions and future work

This PhD project focuses on investigating the stochastic dynamic inventory-routing problem (SDIRP), an increasingly relevant problem in supply chain management, and proposing novel decision policies to obtain effective inventory-routing decisions within reduced computation times, drawing on insights from inventory management and vehicle routing while exploring the potential of machine learning. Three major contributions stand out: (i) understanding the current literature on dynamic inventory-routing problems under uncertainty, identifying research gaps in terms of problem settings and decision policies, and pointing out promising future directions (Chapter 2); and proposing novel anticipatory and efficient policies for (ii) the classical SDIRP (Chapter 3) and (iii) a real-world pharmaceutical distribution problem (Chapter 4). These contributions are interconnected and complementary: the first provides the foundation for the following two, and together all three advance the field of SDIRPs.

5.1. Fulfillment of the research objectives and answers to the research questions

This section is organized around the thesis's research objectives, with each subsection summarizing the contributions and addressing the corresponding research questions.

RO 1: Literature review on stochastic dynamic inventory-routing problems

Over the past decade, we have observed a significant increase in publications on dynamic inventory-routing under uncertainty, reflecting its practical relevance across multiple real-world applications. This thesis starts with a literature review on SDIRPs, providing the first comprehensive review in the field since Baita et al. (1998). This work is presented in Chapter 2 and serves as the basis for the subsequent stages of the project. The main contributions are:

- It proposes a classification framework that categorizes the existing studies according to two main domains: problem settings (general, inventory and routing features) and decision policies (policy class, inventory-routing decomposition approach, optimization method and potential learning domains);
- It investigates how different problem aspects, including scale, complexity and real-time considerations, shape the design of decision policies;
- It points out current trends, emerging challenges, and future directions.

The work was disseminated during the development phase at the XXII Congresso da APDIO (November 6-8, 2022, Évora, Portugal), with the title *Dynamic Inventory Routing Problem - A Literature Review*. The literature review has resulted in the following paper:

- Maia, F., Figueira, G., & Neves-Moreira, F. (2026). Stochastic dynamic inventory routing: A comprehensive review. Computers & Operations Research, 107383. https://doi.org/10.1016/j.cor.2026.107383.

Research question 1.1

Which problem settings are most explored in the SDIRP literature, and which remain underexplored despite their practical relevance?

Although static deterministic and stochastic inventory-routing problem (IRP) variants are well explored in terms of problem dimensions, the same is not true for the stochastic dynamic variant. The following dimensions stand out as overlooked dimensions in the SDIRP literature despite their potential to yield more realistic models.

Extended network configurations. Many-to-many endpoint structures, involving multiple suppliers and customers, remain underexplored in the literature. Although these extended networks provide greater flexibility in responding to potential supply disruptions and stockouts, the problem's complexity and scale also increase, and new coordination challenges emerge, such as supplier selection and delivery synchronization. This dimension constitutes an interesting research path for bridging the gap toward more realistic models.

Multiple items and perishability. Most of the literature relies on single-item formulations, indicating the opportunity to extend the current models to handle multiple items. Furthermore, perishability requires further attention, impacting both inventory management and vehicle routing decisions, particularly in applications such as grocery distribution and emergency services.

Pickup and delivery operations. Current formulations tend to overlook several operational complexities found in real-world settings, such as pickup & delivery (P&D) operations, which are useful for managing demand fluctuations and supply disruptions. Bike-sharing systems are an exception, where the literature has devoted some attention. Split delivery is another dimension that deserves further investigation, despite being extensively explored in the vehicle routing problem (VRP) literature.

Multiple uncertainty sources. The primary source of uncertainty comes from customer demand, which gradually unfolds over time and is often stochastic. Nevertheless, future research could focus on additional sources of uncertainty, such as supply disruptions. This is a critical issue faced by real-world supply chains, affecting both inventory availability and replenishment decisions.

Multiple objectives. Most of the literature focuses on a single objective function that minimizes the total expected inventory holding, stockout, and routing costs. Nonetheless, sustainability emerges as a major concern when discussing supply chain operations, highlighting the opportunity to incorporate global environmental awareness into current models.

Research question 1.2

Which policy classes have been explored in the SDIRP literature, and how do different problem aspects, including scale, complexity and real-time considerations, shape the design and suitability of decision policies?

The proposed classification framework categorizes the decision policies adopted in each study using the nomenclature of Powell (2022), which distinguishes four main policy classes: policy function approximation (PFA), cost function approximation (CFA), direct lookahead approximation (DLA), and value function approximation (VFA). In fact, there is no single policy class universally superior to the others; rather, its suitability depends on the problem setting, with each policy class offering advantages and limitations.

Small-scale problems. Integrated inventory-routing policies and exact methods are particularly suitable for small-scale settings. Looking at the literature, DLA is the most common policy class, anticipating future uncertainty through a deterministic or stochastic lookahead model solved in rolling horizon (RH). The nature and length of this lookahead model are two critical design choices that determine how future uncertainty is represented and whether inventory-routing decisions are more or less myopic (with longer horizons requiring higher computational effort). In this context, parametric CFA emerges as a promising strategy to improve DLA policies, aiming to reduce the computational burden of solving complex stochastic programming models while preserving solution quality (e.g., Cuellar-Usaquén et al., 2024).

Large-scale problems. The increase in problem scale and complexity tends to favor approximate methods and policies that decompose inventory and routing decisions into different stages. Such decomposition reduces the complexity of an integrated inventory-routing optimization problem, at the expense of optimality. Within this context, PFA represents a promising but underexplored policy class, contributing to parametric or non-parametric decision rules that directly map problem states to decisions, e.g., delivery priority rules for customer selection and inventory replenishment rules to determine delivery quantities. Alternatively, CFA can support the design of parameterized single-period models that capture inventory-routing interdependencies while accounting for future uncertainty through a modified objective function or problem constraints.

Real-time problems. In settings where decisions must be made within short time windows, policies that require solving large lookahead models may become less suitable. Therefore, PFA and CFA can be quite useful in this context, and deserve to be better explored in the SDIRP literature. Alternatively, VFA offers a good trade-off between generalization and computational tractability by approximating the value of future states resulting from current decisions. However, learning accurate value functions in high-dimensional problems can be challenging, with reinforcement learning (RL) providing a robust and promising framework to estimate these value functions (e.g., Lagos, 2026).

Research question 1.3

How can learning mechanisms support decision policies for the SDIRP?

Interest in dynamic problems continues to grow, with learning algorithms playing an important role in supporting policy design for complex combinatorial optimization problems. The proposed review explores how learning mechanisms can be integrated into policy design for the SDIRP across three main dimensions: inventory and routing decisions, policy parameters that support decision-making, and optimization method's hyperparameters. In particular, these mechanisms can enhance different policy classes: PFA by refining parametric or non-parametric functions that directly map states to decisions (e.g., customer delivery priority rules, inventory replenishment rules); CFA by improving the estimation of cost functions (e.g., approximating future inventory and routing costs); and VFA by estimating the value functions.

RO 2: Policy design for the stochastic dynamic inventory-routing problem

Building on the promising directions identified in Chapter 2, the second research objective focused on developing anticipatory and efficient policies for the standard SDIRP by combining insights from both the inventory management and vehicle routing literatures. This work is presented in Chapter 3, with the following main contributions:

- It proposes anticipatory and efficient inventory-routing policies within a three-stage decision structure, comprising a customer delivery priority rule learned via genetic programming (GP), an (s, S) inventory replenishment rule tuned by a genetic algorithm (GA), and a routing optimization stage;
- It stands as the first work to transfer and adapt the revised power approximation (RPA) concept to dynamic inventory-routing, linking stochastic inventory control with dynamic multi-period routing;
- It extends existing literature benchmark instances to represent a broader range of SDIRP settings and inventory-routing trade-offs.

The work was disseminated in several national and international conferences, notably the 33rd European Conference on Operational Research (June 30 - July 3, 2024, Copenhagen, Denmark) with the title *Evolutionary Approaches to Solve the Dynamic and Stochastic Inventory-Routing Problem*; and the 34th European Conference on Operational Research (June 22-25, 2025, Leeds, United Kingdom) with the title *Adaptive Policies for the Stochastic Dynamic Multi-Period Inventory-Routing Problem*. The following paper resulted:

- Maia, F., Figueira, G., & Neves-Moreira, F. (2026). Learning policies for the stochastic dynamic inventory-routing problem via evolutionary algorithms. Submitted to Transportation Research Part E: Logistics and Transportation Review.

Research question 2.1

How can insights from both inventory management and vehicle routing be combined to support the design of anticipatory and efficient policies?

The proposed policies build on an established three-stage decision framework, applied in an RH fashion, in which the inventory-routing decision process decomposes into customer prioritization, inventory replenishment, and routing optimization. First, a delivery priority rule that computes a priority score for each customer in each period is learned using GP. This evolutionary algorithm generates compact symbolic expressions intended to capture inventory-routing interdependencies. Second, inventory decisions are guided by an (s,S) inventory replenishment rule, whose control parameters are determined using the RPA, an efficient algorithm for single-item inventory systems with stochastic demand. The RPA parameters are further calibrated to the SDIRP setting using a GA. Third, routing decisions are obtained by solving a VRP using a mathematical solver. In summary, the proposed policies show how insights from both the inventory management and vehicle routing literatures can inform the structure of anticipatory and efficient policies, with evolutionary algorithms supporting the learning of policy components.

Research question 2.2

How do the proposed policies perform in terms of solution quality and computational effort?

The computational study compares the proposed policies against two main groups of benchmarks: three-stage policies based on standard priority and inventory replenishment rules, and integrated optimization benchmarks, including reactive and DLA approaches. Regarding the three-stage benchmarks, the GP-evolved priority rules proved more effective at capturing inventory-routing interdependencies than standard priority rules found in the literature that focus exclusively on inventory urgency, thereby better guiding the following inventory replenishment stage. Furthermore, the RPA used to compute the (s,S) values proved beneficial compared with classical order-up-to-level (OU) and (s,S) inventory replenishment rules, with the further calibration yielding additional, although limited, gains. Together, these components deliver the best aggregate cost performance among all evaluated approaches. The anticipatory and efficient structure of the proposed policies provides a competitive alternative to DLA in terms of solution quality while offering significant computation time savings (around four orders of magnitude).

Research question 2.3

Under which problem conditions are the proposed policies most advantageous?

The scenario-level analysis shows that the performance of the proposed policies is consistently better than all three-stage benchmark policies, performing quite effectively in settings with low customer density, with routing decisions (i.e., visiting or postponing a given customer) having a greater impact on total cost. In contrast, the integrated DLA benchmark

proves more effective for larger instances, particularly when customer density is higher within the same service area, having greater flexibility in jointly balancing inventory and routing decisions. A similar pattern is observed when inventory holding costs have more impact on the total cost structure. Ultimately, vehicle capacity and time horizon have less impact on policy performance than the number of customers and inventory cost structure.

RO 3: Policy design for a real-world pharmaceutical distribution problem

Addressing a real-world problem reveals multiple operational challenges that are often overlooked in well-established IRP instances found in the literature. This research objective focused on developing an anticipatory and efficient inventory-routing policy for a pharmaceutical distribution problem. The work is presented in Chapter 4, and the following main contributions stand out:

- It addresses an SDIRP motivated by the operations of a Portuguese pharmaceutical distributor;
- It proposes an integrated inventory-routing policy based on CFA that combines an inventory target level strategy that accounts for future demand uncertainty with a local inventory synchronization strategy that anticipates future routing limitations;
- It learns (adaptive) policy parameters for different experimental scenarios of demand uncertainty and vehicle capacity, using Bayesian optimization (BO).

The work was disseminated at the Optimization 2026 (July 20-22, 2026, Lisbon, Portugal), with the title *Balancing inventory and routing in stochastic dynamic inventory-routing: An application to pharmaceutical distribution*. The following paper resulted:

- Maia, F., Ulmer, M., Hildebrandt, F., Figueira, G., & Neves-Moreira, F. (2026). Stochastic dynamic inventory-routing via cost function approximation: An application to pharmaceutical distribution. Submitted to European Journal of Operational Research.

Research question 3.1

What operational challenges arise when addressing a real-world pharmaceutical distribution problem?

The problem concerns a one-to-many endpoint structure in which a central warehouse replenishes a set of 24 geographically dispersed pharmacies in southern Portugal, using a fleet of two vehicles with limited capacity. Pharmacies face daily stochastic demand, so inventory and routing decisions must be made at the beginning of each day. The main operational complexities arise from a maximum route duration constraint per vehicle, service times at pharmacy locations, and a multi-vehicle configuration. Inventory management aims to balance the trade-off between inventory holding and stockout costs, while routing acts primarily as a resource constraint seeking efficient routes to satisfy today's and future demand. The central question lies in how to find effective decisions that consider both components.

Research question 3.2

How to leverage the potential of CFA to design an anticipatory and efficient policy?

This work proposes an integrated inventory-routing policy grounded in a parameterized mathematical programming model responsible for current-period decisions, solved in an RH fashion. Through a modified objective function (CFA), the proposed policy manages to i) anticipate future demand uncertainty through the introduction of a desired inventory position at each pharmacy in each period, and ii) anticipate future routing limitations through a routing-aware inventory synchronization mechanism. The trade-off between these two objectives is controlled via a penalty weight. Both the inventory target levels for each pharmacy and the penalty weight are adaptive to each experimental scenario and learned using a BO algorithm.

Research question 3.3

How does the proposed policy perform compared with standard industry practices in terms of solution quality and robustness?

The proposed policy outperforms all practice and method-oriented benchmarks across different experimental scenarios of demand uncertainty and vehicle capacity. Nevertheless, our policy improvements compared to the benchmarks tend to decrease as the degree of uncertainty and vehicle capacity increase. The computational study shows that anticipating future demand through the proposed inventory target deviation strategy, synchronizing inventories across geographically close pharmacies, and learning adaptive policy parameters are already effective when used separately. Furthermore, their integration allows the proposed CFA policy to achieve superior performance, demonstrating the advantage of treating pharmacies as an interconnected network, according to a vendor-managed inventory (VMI) strategy, rather than as isolated and independent entities.

5.2. Future work

This thesis has met the research objectives and corresponding questions established in Chapter 1. However, there are still avenues for future research discussed throughout Chapters 2, 3, and 4.

The first avenue arises from the opportunities highlighted in the literature review presented in Chapter 2 regarding the problem dimensions overlooked in the SDIRP literature and not addressed in this thesis, including multiple supplier networks, multi-objective formulations, P&D operations, perishability, and further sources of uncertainty. Therefore, there is an opportunity to extend the proposed policies to these richer problem settings and bridge the gap to more realistic models.

The second avenue concerns the integration of learning mechanisms into policy design. This thesis illustrates their application at different levels, including evolving delivery priority rules via GP to guide the subsequent inventory replenishment decision stage (Chapter 3), and tuning CFA-based policy parameters via BO (Chapter 4). Future work could explore

the potential of alternative machine learning techniques, including supervised learning (SL) and RL, to develop even more effective policy structures. For instance, SL could be used to train a neural network responsible for determining which customers to visit at a given point in time, and RL could be used to enhance VFA-based policies.

The third avenue refers to the further investigation of novel hybrid policies, as discussed in the literature review of Chapter 2, leveraging the strengths of each policy class (PFA, CFA, DLA, and VFA). For instance, DLA could benefit from CFA through a parametrically modified objective function or constraints, thereby preserving solution quality within reasonable computation times. Furthermore, a VFA can reduce the computational burden of DLA by approximating the downstream value of future states.

The fourth avenue focuses on further enriching the problem formulation introduced in Chapter 4. This may include a broader supplier-pharmacy network and the replenishment of multiple medicines. Finally, future research should also address the potential operational challenges encountered in the real-world environment when implementing the proposed policies, e.g., integrating a given policy with existing information systems.

Bibliography

Baita, F., Ukovich, W., Pesenti, R., Favaretto, D., 1998. Dynamic routing-and-inventory problems: a review. Transportation Research Part A: Policy and Practice 32, 585–598. https://doi.org/10.1016/S0965-8564(98)00014-7.

Cuellar-Usaquén, D., Ulmer, M.W., Gomez, C., Álvarez-Martínez, D., 2024. Adaptive stochastic lookahead policies for dynamic multi-period purchasing and inventory routing. European Journal of Operational Research 318, 1028–1041. https://doi.org/10.1016/j.ejor.2024.06.011.

Lagos, F., 2026. The stochastic and dynamic inventory routing problem with fine time granularity: A reinforcement learning approach. Computers & Operations Research 185, 107298. https://doi.org/10.1016/j.cor.2025.107298.

Powell, W.B., 2022. Reinforcement Learning and Stochastic Optimization: A Unified Framework for Sequential Decisions. John Wiley & Sons, Hoboken, NJ. https://doi.org/10.1002/9781119815068.