HEC MONTRÉAL
École affiliée à l'Université de Montréal

Stochastic Production Routing Problem With Adaptive Routing

par
Ali Kermani

Thèse présentée en vue de l'obtention du grade de Ph. D. en administration
(spécialisation Gestion des opérations et de la logistique)

Août 2025

HEC MONTRÉAL

École affiliée à l'Université de Montréal

Cette thèse intitulée :

Stochastic Production Routing Problem With Adaptive Routing

Présentée par :

Ali Kermani

a été évaluée par un jury composé des personnes suivantes :

Yossiri Adulyasak
HEC Montréal
Président-rapporteur

Jean-François Cordeau
HEC Montréal
Codirecteur de recherche

Raf Jans
HEC Montréal
Codirecteur de recherche

Walter Rei
Université du Québec à Montréal
Membre du jury

Stéphane Dauzère-Pérès
École des Mines de Saint-Étienne
Examinateur externe

Okan Arslan
HEC Montréal
Représentant du directeur de HEC Montréal

Résumé

Le Problème de Tournées de Production (PRP) intègre des décisions essentielles de planification de la chaîne d'approvisionnement, notamment la production, la gestion des stocks et la livraison des marchandises aux clients à l'aide d'une flotte de véhicules. Son objectif principal est d'améliorer l'efficacité globale de la chaîne d'approvisionnement et de réduire les coûts en renforçant la coordination et en optimisant les décisions dans ces domaines interdépendants. L'ignorance de l'incertitude de la demande peut entraîner une augmentation significative des coûts de production et de distribution. Traditionnellement, la majorité des études sur le PRP reposent sur des modèles déterministes ou des approches stochastiques dans lesquelles les décisions de tournées sont fixées avant la réalisation de la demande. Cette limitation conduit souvent à des visites client inutiles et à des coûts de transport élevés. Cette recherche contribue à la littérature en introduisant une stratégie de tournées adaptatives, dans laquelle les itinéraires de livraison sont ajustés en fonction de la demande réalisée. Cette flexibilité est particulièrement bénéfique lorsque les coûts de livraison sont élevés ou que les décisions doivent être adaptées à la demande.

Le premier chapitre propose une formulation de programmation stochastique à deux étapes pour le PRP avec un seul produit et une demande incertaine. La contribution principale réside dans la possibilité de prendre les décisions de tournées en deuxième étape, après la réalisation de la demande. Cette formulation, appelée SPRP-AR, se distingue des modèles antérieurs où les tournées sont fixées dès la première étape. Pour résoudre ce problème complexe, un algorithme de Progressive Hedging (PH) est combiné à une matheuristique en trois phases. L'algorithme PH décompose le problème en

sous-problèmes spécifiques à chaque scénario et guide de manière itérative les décisions de première étape en ajustant des multiplicateurs lagrangiens. Les résultats montrent que le modèle proposé permet de réaliser des économies de coûts significatives par rapport à l'approche traditionnelle à tournées fixes.

Le deuxième chapitre étend le SPRP-AR en intégrant des contraintes de niveau de service. L'étude évalue quatre mesures de niveau de service, chacune imposant des restrictions différentes sur les ruptures de stock et les retards, et examine leur application selon une granularité spécifique au client ou au niveau de l'usine, et ce, soit sur une période unique, soit sur l'ensemble de l'horizon de planification. Un algorithme matheuristique itératif est développé, composé de trois phases alternant diversification et intensification. L'algorithme génère successivement les décisions de lancement, détermine les quantités de production et de livraison sous contraintes de niveau de service, puis améliore les tournées. Les résultats mettent en évidence l'importance managériale du choix des types de niveaux de service et de la granularité adoptée, qui influencent fortement les coûts.

Le troisième chapitre élargit le cadre de modélisation à un PRP stochastique à deux échelons, incluant une usine, des entrepôts et des clients. Les produits sont transportés de l'usine vers les entrepôts, puis des entrepôts vers les clients à l'aide de véhicules distincts. L'affectation des clients aux entrepôts est flexible selon les périodes et les scénarios. Alors que les tournées du premier échelon sont fixes, celles du second varient selon les scénarios, conformément à l'approche des tournées adaptatives. Une formulation stochastique en deux étapes est proposée et résolue à l'aide d'un Algorithme Heuristique Hybride combinant un programme en nombres entiers pour les décisions de première étape et une métaheuristique de Recherche Locale Itérée pour les décisions de deuxième étape spécifiques à chaque scénario. Les résultats confirment la capacité de l'algorithme à résoudre efficacement des instances déterministes de grande taille et des cas stochastiques de taille moyenne, et soulignent les avantages de la prise en compte de l'incertitude dans des contextes à plusieurs échelons.

Mots-clés

Problème de tournées de production, Logistique, Problème de tournées de production à deux échelons, Programmation stochastique, Méthode Progressive Hedging, Contraintes de niveau de service, Matheuristiques, Métaheuristiques, Heuristiques, Programmation en nombres entiers mixtes, Algorithmes de décomposition

Méthodes de recherche

Recherche opérationnelle, Programmation mathématique, Programmation stochastique

Abstract

The Production Routing Problem (PRP) integrates critical supply chain planning decisions, including production, inventory management, and the delivery of goods to customers via a fleet of vehicles. Its primary objective is to enhance overall supply chain efficiency and reduce costs by improving coordination and optimizing decisions across these interconnected areas. When demand uncertainty is ignored, it can lead to significant increases in production and distribution costs. Traditionally, most PRP studies have relied on deterministic models or stochastic approaches in which routing decisions are fixed prior to demand realization. This limitation often results in unnecessary customer visits and higher transportation costs. This research advances the literature by introducing the adaptive routing strategy, in which delivery routes are adjusted in response to realized demand. This flexibility is particularly beneficial when delivery costs are high or when demand-driven decisions are required.

The first chapter presents a two-stage stochastic programming formulation for the PRP with a single product and uncertain demand. The main contribution lies in enabling routing decisions to be made in the second stage, after demand realization. This formulation, referred to as SPRP-AR, contrasts with prior models that fix routing in the first stage. To solve this challenging problem, a Progressive Hedging (PH) algorithm is combined with a three-phase matheuristic. The PH algorithm decomposes the problem into scenario-specific subproblems and iteratively guides first-stage decisions by adjusting Lagrangean multipliers. Results show that the proposed model achieves significant cost savings compared to the traditional fixed-route approach.

The second chapter extends the SPRP-AR by incorporating service level constraints. The study evaluates four service level measures, each imposing different restrictions on stockouts and backlogs, and examines their implementation under customer-specific versus plant-level granularity and over single periods or the entire planning horizon. An Iterative Matheuristic algorithm is developed, consisting of three phases that alternate between diversification and intensification. The algorithm successively generates setup decisions, determines production and delivery quantities under service level constraints, and refines routing. Results highlight the managerial value of choosing appropriate service level types and levels of granularity, both of which significantly affect cost.

The third chapter extends the modeling framework to a stochastic two-echelon PRP, which includes a plant, warehouses, and customers. Products are transported from the plant to warehouses, and then from warehouses to customers using separate vehicles. Customer-to-warehouse assignments are flexible across periods and scenarios. While first-echelon routes are fixed, second-echelon routes vary by scenario, following the adaptive routing approach. A two-stage stochastic formulation is proposed and solved using a Hybrid Heuristic Algorithm that combines a Mixed-Integer Program for first-stage decisions with an Iterated Local Search metaheuristic for scenario-based second-stage decisions. The results confirm the algorithm's ability to handle large-scale deterministic instances and medium-scale stochastic cases, and underscore the benefits of modeling uncertainty in multi-echelon settings.

Keywords

Production Routing Problem, Logistics, Two-Echelon Production Routing Problem, Stochastic Programming, Progressive Hedging, Service Level Constraints, Matheuristics, Metaheuristics, Heuristics, Mixed-Integer Programming, Decomposition Algorithms

Research Methods

Operations Research, Mathematical Programming, Stochastic Programming

Contents

List of Tables

List of Figures

List of Acronyms

ALNS Adaptive Large Neighborhood Search

ANR Average Number of Routes

ANVC Average Per-Route Number of Visited Customers

ARL Average Route Length

ARP Assembly Routing Problem

BC Branch-and-Cut

CVRP Capacitated Vehicle Routing Problem

EEV Expected Value of the EV Solution

EV Expected Value

EVPI Expected Value of Perfect Information

FIFO First-In-First-Out

GHG Greenhouse Gases

GRASP Greedy Randomized Adaptive Search Procedure

ILS Iterated Local Search

IMH Iterative Matheuristic

HHA Hybrid Heuristic Algorithm

IRP Inventory Routing Problem

LB Lower Bound

LS Local Search

LSP Lot Sizing Problem

ML Maximum Level

MTZ Miller–Tucker–Zemlin

MIP Mixed-Integer Programming

OU Order-Up-To-Level

PDF Probability Distribution Functions

PH Progressive Hedging

PRP Production Routing Problem

RP Recourse Problem

RVND Randomized Variable Neighborhood Descent

SAA Sample Average Approximation

SCM Supply Chain Management

SEC Subtour Elimination Constraints

SPRP-AR Stochastic PRP with Adaptive Routing

SPRP-FSR Stochastic PRP with first-stage Routing

S2EPRP-AR Stochastic Two-Echelon Production Routing Problem with Adaptive Routing

TSP Traveling Salesman Problem

TS Tabu Search

UB Upper Bound

VMI Vendor-Managed Inventory

VRP Vehicle Routing Problem

VSS Value of the Stochastic Solution

WS Wait-and-See

2E-PRPCS Two-Echelon PRP with Cross-docking Satellites

2E-VRP Two-Echelon Vehicle Routing Problem

3LSP Three-Level Lot Sizing Problem

To my wife, Tina.

Your love, patience, and constant support gave me the strength to continue through every challenge. This accomplishment would not have been possible without you.

To the memory of my friend Sina.

We started this journey together. Although you are no longer here, your passion for learning and your friendship have remained a source of inspiration throughout this work.

Acknowledgements

I would like to express my deepest gratitude to my supervisors, Jean-François Cordeau and Raf Jans, for their constant support, guidance, and commitment throughout the course of this research. Their scientific expertise, mentorship, and encouragement were invaluable at every stage of this journey, and this thesis would not have been possible without them.

I am also sincerely grateful to Walter Rei (UQAM) for his insightful feedback, which helped shape and strengthen this work from its early stages. I would also like to thank Ivan Contreras (Concordia University), Julie Paquette, Okan Arslan, Laurent Charlin (HEC Montréal), and Janosch Ortmann (UQAM) from whom I learned a lot. I extend my thanks to Serge Bisaillon (CIRRELT) for his continued technical support and to Aldair Alvarez for his assistance. I am equally thankful to Yossiri Adulyasak (HEC Montréal) and Masoud Chitsaz for their constructive discussions and feedback. I also wish to acknowledge the thoughtful comments and support of my colleagues at HEC Montréal, including Narges Sereshti and Matthieu Gruson.

I feel fortunate to have shared this experience with exceptional colleagues and friends at HEC Montréal and Polytechnique Montréal, among them Ali, Kamyar, Birce, Nicolas, and Maedeh. I also extend my appreciation to Guillaume Michaud and Pierre Girard for their technical assistance.

This research was made possible through the financial support of the HEC Montréal Research Chair in Supply Chain Operations Planning and the Natural Sciences and Engineering Research Council of Canada (NSERC). I also gratefully acknowledge the Digital

Research Alliance of Canada for providing the computational resources essential to this work.

Last but not least, I am deeply grateful to my parents, whose guidance and enduring belief in me have shaped my path throughout life. I am also thankful to my wife for her constant support and encouragement.

Finally, I wish to honor the memory of my dear friend Sina Shahnejat-Bushehri.

General Introduction

Supply chains represent complex networks that typically involve a focal firm responsible for assembling or manufacturing one or more products, a group of suppliers providing necessary raw materials or components, intermediate warehouses for storage and distribution, and a network of retailers delivering the final products to end customers. These networks are connected by transportation fleets that enable the flow of materials between facilities. Transportation is a critical component, not only because it accounts for a significant portion of the overall costs, but also due to its substantial impact on environmental sustainability. This multi-tiered structure highlights the complexity of modern supply chains, where each level plays a crucial role in ensuring the efficient and effective delivery of products from origin to destination.

Traditionally, supply chain management (SCM) systems have been characterized by a segmented approach, where each entity within the supply chain operates independently, focusing solely on its own objectives. This often involves solving separate optimization problems such as the lot sizing problem (LSP) to plan production and inventory management at the production plant, the economic order quantity model and various inventory policies at the retailer level to minimize inventory costs, and the vehicle routing problem (VRP) to determine the most efficient distribution routes. While these efforts can lead to improvements in local efficiency, they often fail to account for the interdependencies between different supply chain components, potentially leading to suboptimal performance across the entire network.

The lack of coordination in traditional SCM can result in inefficiencies, as decisions made by one party may inadvertently increase costs or create challenges for others. For instance, a production plant might optimize its operations to minimize inventory costs, but this could lead to delays in deliveries to retailers, causing stockouts and lost sales. Similarly, optimizing inventory levels without considering transportation capacity constraints can lead to higher transportation costs or disruptions within the supply chain.

Recognizing these challenges, organizations have increasingly sought to adopt more integrated and collaborative approaches to SCM. A common approach is the Vendor-Managed Inventory (VMI) system, which addresses the lack of coordination by shifting the inventory management responsibility from retailers to the manufacturer. Under a VMI system, the focal firm not only manages its own production and inventory but also oversees inventory levels at the retailer level. This holistic view of the supply chain enables better synchronization between production, inventory management, and distribution activities.

Adopting VMI can lead to several benefits, including reduced inventory levels, improved service levels, and lower overall supply chain costs. The focal firm can optimize production schedules based on real-time demand information from retailers, reducing the risk of overproduction or stockouts. Additionally, by coordinating transportation and distribution activities more effectively, companies can reduce transportation costs and minimize their environmental impact, contributing to a more sustainable supply chain. As supply chains continue to evolve, the importance of coordination and integration will only grow, making approaches like VMI essential for achieving long-term success in a competitive global market.

Supply chain integration (SCI) seeks to unify various components of the supply chain, balancing costs and service levels by improving the flow of materials and information across the network. According to Stevens (1989), SCI focuses on minimizing inventories, enhancing customer service, and facilitating communication between different parts of an organization or between different entities in the supply chain. SCI encourages companies to operate as a unified entity, allowing for better demand forecasting and improved alignment between supply chain partners (Flynn et al., 2010). SCI can be broadly categorized into internal integration (II) and external integration (EI). Internal integration enables information exchange within an organization, fostering collaboration between different departments, while external integration extends this collaboration to external partners such as suppliers and customers. Lu et al. (2018) argue that successful internal integration is a prerequisite for effective external integration, as it ensures that a company can respond quickly and accurately to market demands by aligning supply and demand. The benefits of SCI include improved operational efficiency, enhanced customer satisfaction, and reduced overall costs.

While empirical studies have demonstrated the strategic benefits of SCI, there is also a need for mathematical modeling and analytical approaches to better understand how to achieve optimal results given existing resources. These studies often introduce mathematical formulations that treat the supply chain as a single entity, analyzing the cost savings and efficiency gains from intra- and inter-organizational integration. For instance, Bell et al. (1983) and Chandra (1993) laid the groundwork for integrated supply chain models, emphasizing the importance of managing material flow and planning across the entire supply chain network. As supply chain networks have grown more complex, these integrated models have evolved to include various aspects of supply chain planning, such as the three-level lot sizing problem (3LSP), two-echelon vehicle routing problem (2E-VRP), inventory routing problem (IRP), and production routing problem (PRP) (Absi et al., 2015; Adulyasak et al., 2015b; Coelho and Laporte, 2014; Gruson et al., 2019; Gruson et al., 2023; Gu et al., 2022; Hrabec et al., 2022; Perboli et al., 2011; Sluijk et al., 2023). However, handling these integrated problems presents significant challenges, requiring sophisticated techniques that go beyond the traditional approach of addressing subproblems in isolation (Absi et al., 2018).

Production routing, in particular, represents a significant advance in supply chain planning by integrating production, inventory, and transportation decisions into a single optimization problem. This approach enhances coordination among different supply chain components, leading to increased efficiency and cost savings. Since its introduction by Chandra (1993) and Chandra and Fisher (1994), the PRP has been the subject of extensive research, with various formulations and solution algorithms being proposed to tackle its inherent complexity. Adulyasak et al. (2015b) provide an overview of the PRP, including both formulations and solution approaches. Moreover, a systematic review by Hrabec et al. (2022) provides valuable insights into how employing the PRP could help companies increase their efficiencies. Researchers have explored both deterministic and stochastic versions of the PRP, with the latter accounting for demand uncertainty, which is a critical factor in real-world supply chains. The shift from deterministic to stochastic models in PRP research reflects a broader trend in supply chain optimization, where the focus has expanded to include the uncertainties and variabilities that characterize modern supply chains.

Addressing demand uncertainty represents a significant advancement in the study of the PRP. In this context, Adulyasak et al. (2015a) develop formulations for the PRP that account for demand uncertainty by incorporating a penalty cost for unmet demand. Their approach relies on fixed routing across all possible scenarios and employs a Benders decomposition technique to solve the problem effectively. Agra et al. (2018) study the PRP in the context of demand uncertainty, including backlogs. Zhang et al. (2018) introduce a two-stage stochastic model for the PRP that considers demand uncertainty as well as integrates remanufacturing processes and simultaneous pickup and delivery operations. Additionally, their model incorporates a carbon cap-and-trade policy, demonstrating the growing emphasis on environmental considerations within the PRP framework. Wang et al. (2021) explore the PRP within a setting where both demand and cost uncertainties are present. More recently, Mousavi et al. (2022) proposed a formulation that addresses both demand uncertainty and product perishability. Despite these advances, there remains a gap in the literature concerning the integration of adaptive routing within the PRP, which would allow supply chains to dynamically respond to demand fluctuations by determining the routing decisions after the demand becomes known, and enhance overall operational performance.

In this thesis, we explore three key and interconnected topics within supply chain management: the PRP, the challenges posed by demand uncertainty, and the complexities of multi-echelon supply chain networks. Our research contributes to the field by introducing novel models and solution algorithms designed to improve the efficiency and adaptability of supply chain operations in the face of various practical constraints and uncertainties.

First, we address the PRP under demand uncertainty with adaptive routing, developing a two-stage stochastic programming formulation that enhances flexibility and responsiveness by allowing routing decisions to be made in the second stage based on realized demand. Next, we expand this framework by integrating service level constraints, offering an approach to balance cost control and service reliability under uncertainty. Finally, we extend the scope of the PRP to a two-echelon PRP, introducing adaptive routing and dynamic customer-to-warehouse assignments to improve overall network efficiency. Below, we provide detailed summaries of the contributions made in each of these three papers.

Chapter 1

Kermani, A., Cordeau, J.-F., Jans, R., 2024. A progressive hedging-based matheuristic for the stochastic production routing problem with adaptive routing. Computers & Operations Research, 169, 106745.

In the first paper, we address the PRP under demand uncertainty with adaptive routing (SPRP-AR) by introducing a two-stage stochastic programming formulation. This approach deviates from traditional models where routing decisions are predetermined in the first stage, thus remaining fixed regardless of actual demand (Adulyasak et al., 2015a). Instead, our model allows these routing decisions to be made in the second stage, offering increased flexibility and responsiveness to real-time demand fluctuations. This adaptability is crucial in reducing unnecessary customer visits and decreasing costs associated with demand uncertainty.

Our research makes several key contributions. First, we develop a progressive hedging (PH) algorithm that decomposes the complex SPRP-AR into more manageable subproblems. The PH algorithm iteratively adjusts the first-stage decisions by modifying the cost parameters in the objective function to drive convergence. We further enhance this approach by integrating a three-phase matheuristic algorithm. The first phase involves solving a Traveling Salesman Problem (TSP) to determine an a priori tour, which serves as the foundation for the subsequent stages. In the second phase, we refine the production setup, quantity, and visit decisions by solving subproblems restricted to the initial tour until consensus or a stopping criterion is achieved. The final phase involves solving a capacitated vehicle routing problem (CVRP) for each period and scenario, optimizing the routing decisions based on the refined first-stage variables.

Additionally, we explore a static-static case inspired by the lot-sizing literature, where production quantities are also considered in the first stage. Our computational experiments validate the effectiveness of the proposed algorithm, demonstrating substantial improvements in cost efficiency and operational flexibility when routing decisions are made adaptively in the second stage. This research not only advances the methodology for solving the PRP under uncertainty but also sets a precedent for future studies on adaptive decision-making in supply chains.

Chapter 2

Kermani, A., Cordeau, J.-F., Jans, R., 2025. The stochastic production routing problem with adaptive routing and service level constraints. Omega, under review.

The second article expands on the SPRP-AR framework by incorporating service level constraints, an essential consideration in ensuring that supply chain operations meet predetermined performance standards. We recognize that managing uncertainty in supply chains is not just about minimizing costs but also about maintaining service reliability. To address this, we incorporate four distinct service level metrics, namely α, β, γ, and δ, each representing a different aspect of service level assurance.

The α service level is event-based, focusing on the probability that demand can be fully met from available inventory. The β service level, commonly referred to as the fill rate, measures the proportion of demand satisfied directly from inventory, emphasizing the importance of minimizing backorders. The γ service level targets the ratio of expected backlogs to average demand, providing a balanced view of inventory management. Finally, the δ service level imposes a constraint on the expected backlog relative to the maximum expected backlog. Our study is novel in that it applies these service level constraints within the context of the SPRP-AR, offering a more nuanced approach to managing uncertainty. We propose an iterative matheuristic (IMH) algorithm that systematically addresses the complex interplay between production, inventory, and routing decisions while respecting the service level constraints. The IMH algorithm is designed to solve the problem in stages: first, by generating setup decisions through a two-level lot sizing problem with direct shipments; second, by addressing the PRP with fixed setups and an aggregated vehicle; and finally, by refining routing decisions while ensuring that service level requirements are met.

In addition, we explore various levels of service level granularity, from individual customer constraints to aggregate service levels across all customers and periods. Our computational results reveal the trade-offs between different service level strategies and their impact on cost and efficiency, providing valuable insights for supply chain managers aiming to balance cost control with service reliability.

Chapter 3

Kermani, A., Cordeau, J.-F., Jans, R., 2025. The Stochastic Two-Echelon Production Routing Problem with Adaptive Routing. Networks, under review.

The third paper extends the scope of the PRP by addressing a more complex stochastic two-echelon production routing problem (S2EPRP-AR) that considers both production and distribution decisions across multiple levels of the supply chain. In this model, we introduce two key innovations: the incorporation of adaptive routing in the second stage of the problem under demand uncertainty, and the flexibility to dynamically adjust customer-to-warehouse assignments.

The S2EPRP-AR model is particularly relevant in today's intricate supply chains, which often involve multiple intermediary stages, such as warehouses, between production plants and end customers. Traditional approaches, which rely on fixed routing and customer assignments, can lead to inefficiencies, especially in the face of fluctuating demand. Our model allows for adaptive routing, where transportation routes from warehouses to customers are determined based on realized demand, thereby improving responsiveness and reducing unnecessary visits and transportation costs.

Furthermore, our model introduces dynamic customer-to-warehouse assignments, allowing for adjustments in the assignment of customers to warehouses depending on the scenario and period. This added flexibility enhances the supply chain ability to cope with demand variability and improves overall network efficiency.

To solve the S2EPRP-AR, we decompose the problem into two subproblems and develop a hybrid heuristic algorithm (HHA). The first subproblem addresses the first stage of the problem, where the setup plan, production quantities, and initial transportation routes from the production plant to warehouses are decided. As the number of warehouses is assumed to be limited in the presented problem, all possible combinations of warehouses are considered. The optimal route for visiting the warehouses is explicitly defined in the model and solved as an MIP using a general solver. When solving this problem, the integer variables of the second stage are fixed while the continuous variables are allowed to be optimized.

The second stage of the problem includes the inventory management of warehouses and customer demands, as well as routing from warehouses to customers for each period and scenario. This part is solved using an Iterated Local Search (ILS) algorithm, while the values of the first-stage decision variables remain fixed. This makes the second subproblem scenario-decomposable, which helps improve the algorithm's performance. The HHA iterates over the two subproblems and, in each iteration, attempts to reassign customers to other warehouses either to improve the solution or to diversify the search. The first objective is achieved by reassigning customers facing shortages to other warehouses if it reduces the total cost. To achieve the second objective, the algorithm considers a perturbation by randomly removing a warehouse from a period and scenario and allowing its customers to be reassigned to other warehouses.

The developed algorithm is also adapted to solve larger instances of the deterministic problem than those considered in previous studies. Both the 2EPRP and the 2EPRP with cross-docking satellites are addressed using the proposed algorithm.

References

Absi, N., Archetti, C., Dauzère-Pérès, S., and Feillet, D. (2015). "A Two-Phase Iterative Heuristic Approach for the Production Routing Problem". *Transportation Science* 49.4, pp. 784–795. DOI: 10.1287/trsc.2014.0523.

Absi, N., Archetti, C., Dauzère-Pérès, S., Feillet, D., and Speranza, M. G. (2018). "Comparing sequential and integrated approaches for the production routing problem". *European Journal of Operational Research* 269.2, pp. 633–646. DOI: 10.1016/j.ejor.2018.01.052.

Adulyasak, Y., Cordeau, J.-F., and Jans, R. (2015a). "Benders Decomposition for Production Routing Under Demand Uncertainty". *Operations Research* 63.4, pp. 851–867. DOI: 10.1287/opre.2015.1401.

Adulyasak, Y., Cordeau, J.-F., and Jans, R. (2015b). "The production routing problem: A review of formulations and solution algorithms". *Computers & Operations Research* 55, pp. 141–152. DOI: 10.1016/j.cor.2014.01.011.

Agra, A., Requejo, C., and Rodrigues, F. (2018). "An adjustable sample average approximation algorithm for the stochastic production-inventory-routing problem". *Networks* 72.1, pp. 5–24. DOI: 10.1002/net.21796.

Bell, W. J., Dalberto, L. M., Fisher, M. L., Greenfield, A. J., Jaikumar, R., Kedia, P., Mack, R. G., and Prutzman, P. J. (1983). "Improving the Distribution of Industrial Gases with an On-Line Computerized Routing and Scheduling Optimizer". *Interfaces* 13.6, pp. 4–23. DOI: 10.1287/inte.13.6.4.

Chandra, P. (1993). "A dynamic distribution model with warehouse and customer replenishment requirements". *Journal of the Operational Research Society* 44.7. DOI: 10.1057/jors.1993.117.

Chandra, P. and Fisher, M. L. (1994). "Coordination of production and distribution planning". *European Journal of Operational Research* 72.3, pp. 503–517. DOI: 10.1016/0377-2217(94)90419-7.

Coelho, L. C. and Laporte, G. (2014). "Improved solutions for inventory-routing problems through valid inequalities and input ordering". *International Journal of Production Economics* 155, pp. 391–397. DOI: 10.1016/j.ijpe.2013.11.019.

Flynn, B. B., Huo, B., and Zhao, X. (2010). "The impact of supply chain integration on performance: A contingency and configuration approach". *Journal of Operations Management* 28.1, pp. 58–71. DOI: 10.1016/j.jom.2009.06.001.

Gruson, M., Bazrafshan, M., Cordeau, J.-F., and Jans, R. (2019). "A comparison of formulations for a three-level lot sizing and replenishment problem with a distribution structure". *Computers & Operations Research* 111, pp. 297–310. DOI: 10.1016/j.cor.2019.07.005.

Gruson, M., Cordeau, J.-F., and Jans, R. (2023). "Split demand and deliveries in an integrated three-level lot sizing and replenishment problem". *Computers & Operations Research*, p. 106434. DOI: 10.1016/j.cor.2023.106434.

Gu, W., Archetti, C., Cattaruzza, D., Ogier, M., Semet, F., and Grazia Speranza, M. (2022). "A sequential approach for a multi-commodity two-echelon distribution problem". *Computers & Industrial Engineering* 163, p. 107793. DOI: 10.1016/j.cie.2021.107793.

Hrabec, D., Hvattum, L. M., and Hoff, A. (2022). "The value of integrated planning for production, inventory, and routing decisions: A systematic review and meta-analysis". *International Journal of Production Economics* 248, p. 108468. DOI: https://doi.org/10.1016/j.ijpe.2022.108468.

Lu, D., Ding, Y., Asian, S., and Paul, S. K. (2018). "From supply chain integration to operational performance: The moderating effect of market uncertainty". *Global Journal of Flexible Systems Management* 19.1, pp. 3–20. DOI: 10.1007/s40171-017-0161-9.

Mousavi, R., Bashiri, M., and Nikzad, E. (2022). "Stochastic production routing problem for perishable products: Modeling and a solution algorithm". *Computers & Operations Research* 142, p. 105725. DOI: 10.1016/j.cor.2022.105725.

Perboli, G., Tadei, R., and Vigo, D. (2011). "The two-echelon capacitated vehicle routing problem: Models and math-based heuristics". *Transportation Science* 45.3, pp. 364–380. DOI: 10.1287/trsc.1110.0368.

Sluijk, N., Florio, A. M., Kinable, J., Dellaert, N., and Van Woensel, T. (2023). "Two-echelon vehicle routing problems: A literature review". *European Journal of Operational Research* 304.3, pp. 865–886. DOI: doi.org/10.1016/j.ejor.2022.02.022.

Stevens, G. C. (1989). "Integrating the Supply Chain". *International Journal of Physical Distribution & Materials Management* 19.8, pp. 3–8. DOI: 10.1108/eum0000000000329.

Wang, C., Ni, Y., and Yang, X. (2021). "The production routing problem under uncertain environment". *IEEE Access* 9, pp. 15375–15387. DOI: 10.1109/ACCESS.2021.3053046.

Zhang, Y., Alshraideh, H., and Diabat, A. (2018). "A stochastic reverse logistics production routing model with environmental considerations". *Annals of Operations Research* 271.2, pp. 1023–1044. DOI: 10.1007/s10479-018-3045-2.

Chapter 1

A Progressive Hedging-based Matheuristic for the Stochastic Production Routing Problem with Adaptive Routing

Abstract

The production routing problem (PRP) arises in the context where a manufacturing facility manages its production schedule, the delivery of goods to customers by a fleet of vehicles, and the inventory levels both at the plant and at the customers. The presence of uncertainty often complicates the planning process. In particular, production and distribution costs may significantly increase if demand uncertainty is ignored in the planning phase. Nevertheless, only a few studies have considered demand uncertainty in the PRP. In this article, we propose a two-stage stochastic programming approach for a one-to-many PRP with a single product and demand uncertainty. Unlike previous studies in the literature, we consider the case where routing decisions are made in the second stage after customer demands become known. This offers more flexibility, which can decrease transportation

costs by preventing unnecessary customer visits. In addition to the static-dynamic case, in which setup decisions are made first and production quantities are decided in the second stage, we also consider the static-static setting where both sets of decisions must be made prior to the demand realization. A progressive hedging algorithm combined with a matheuristic is developed to solve the problem. The role of the progressive hedging algorithm is to decompose the stochastic problem into more tractable scenario-specific subproblems and lead the first-stage variables toward convergence by modifying their Lagrangean multipliers. However, solving the subproblems remains challenging since they include routing decisions, and we thus propose a matheuristic to exploit the structural characteristics of the subproblems. First, a Traveling Salesman Problem (TSP) is solved to find the optimal tour for all customers regardless of demand and capacity. Utilizing the sequence obtained from the TSP, we then solve a restricted PRP while taking into account the other constraints of the original problem. Finally, for each period and scenario, a vehicle routing problem is solved to improve the quality of the solutions. Computational experiments are conducted to analyze the algorithm's efficiency and to assess the benefits that can be achieved by handling routing in the second stage.

1.1 Introduction

The production routing problem (PRP) aims to make production, inventory, and routing decisions simultaneously by integrating the lot-sizing and vehicle routing problems. Adulyasak et al. (2015b) provide an overview of the PRP literature, including both formulations and solution methods. The majority of the formulations discussed in this survey focus on the deterministic PRP, but in recent years, the stochastic version of the problem has attracted more attention. In particular, demand uncertainty is known to have a great impact on both production planning and vehicle routing. However, one should keep in mind that the PRP is an NP-hard problem and that considering stochasticity further increases the difficulty of the problem. Adulyasak et al. (2015a) introduced a two-stage and a multi-stage model for the stochastic production routing problem (SPRP) with demand uncertainty. In both problems, setup and routing decisions are made in the first stage, while the remaining decisions, including production quantities, delivery quantities and inventory levels, belong to the second stage. When solving the two-stage formulation, the demand for the entire planning horizon is assumed to become known at the beginning of the second stage, whereas in the multi-stage problem, the demand of each period is realized at the beginning of that period. Although the multi-stage setting offers more flexibility, the two-stage model can be solved more efficiently. It can also be used in a rollout algorithm to provide a heuristic solution to the multi-stage problem.

Considering the routing decisions among the first-stage decisions may lead to unnecessary customer visits and higher costs. In this paper, we introduce a two-stage stochastic programming formulation for the PRP with adaptive routing, i.e., the routing becomes a second-stage decision, which provides more flexibility in response to the observed demand. We assume a plant that produces and distributes a single product to multiple customers in a finite horizon using a fleet of homogeneous vehicles. If the demand of a customer cannot be fully satisfied from the available inventory, a cost per unit of unmet demand is imposed to represent the outsourcing costs.

1.1.1 An illustrative example

We provide below a simple example with two periods, three customers, and two scenarios to illustrate the possible benefits of considering routing decisions in the second stage. In this example, we assume that there are no holding or production costs, that the production and inventory capacities are large, and that there are two vehicles with a capacity of 50. The two scenarios have the same probability $p_{s_1} = p_{s_2} = 0.5$. The rest of the parameters is provided in Table 1.1, where the first two rows are the x and y coordinates of each node and the next four rows are the demands of the customers. The transportation costs are calculated using Euclidean distances. Note that the scenario and period pairs are denoted by (s,t). The production plant is denoted by 0 and customers are indexed from 1 to 4. Figure 1.1 displays the optimal solution to this example under both routing strategies. The

Table 1.1: Parameters of the example

Node No.		0	1	2	3	4
x coordinate		143	89	76	285	401
y coordinate		99	159	314	63	325
demand	(1,1)	-	4	14	21	0
	(1,2)	-	9	17	31	6
	(2,1)	-	24	12	3	13
	(2,2)	-	29	15	10	18

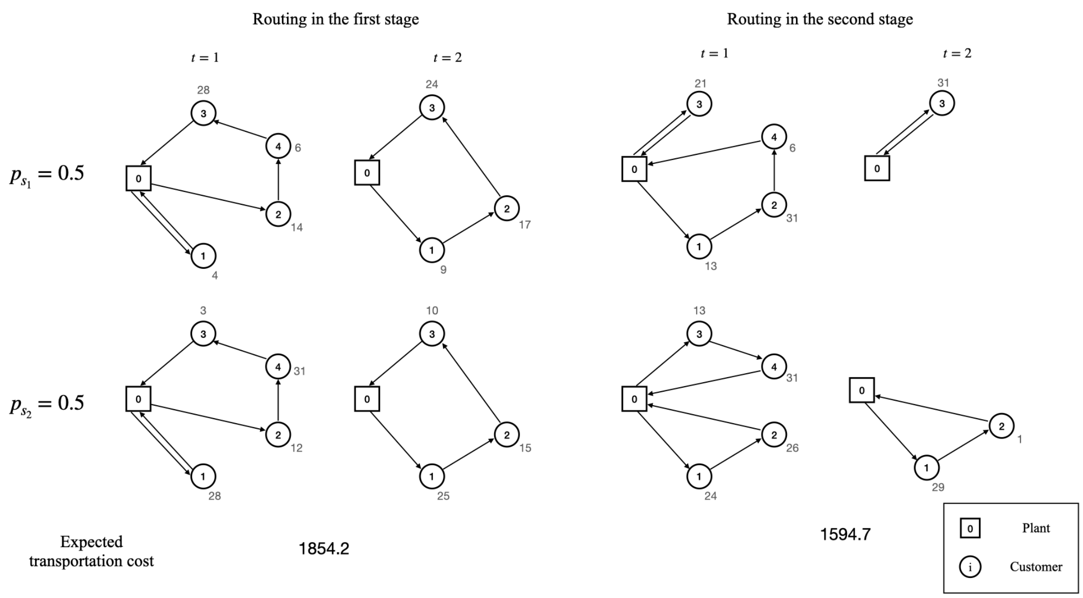


Figure 1.1: An example of considering routing in the first stage vs. the second stage

numbers written next to the nodes are the delivered quantities. If we solve this problem with first-stage routing, the optimal expected cost is 1854.2. However, by considering adaptive routing, the optimal expected cost is 1594.7, demonstrating how this flexibility can lower the cost.

1.1.2 Contributions

To the best of our knowledge, this is the first study to consider adaptive routing in the SPRP, which is the first contribution of our study. Our second contribution is to develop a progressive hedging (PH) approach combined with a three-phase matheuristic algorithm. Through the PH algorithm, the problem is decomposed into subproblems, and the first-stage variables are guided toward convergence by changing their corresponding costs in the objective function. The first phase of the process involves solving a TSP to find an a priori tour. Production setup, quantities, and visit decisions are determined by restricting subproblems to the route defined in the first phase of the algorithm and by solving them until a consensus is reached in the first-stage variables or a stopping criterion is met. An aggregation procedure is applied at this point to construct the solution, and in the final stage of the solution process, a capacitated vehicle routing problem (CVRP) is solved for each period and every scenario to further improve the routing decisions. The third contribution of this study is the consideration of the static-static case that has been adopted from the lot sizing literature, where the production quantities are also considered in the first stage of the problem. Finally, computational experiments are conducted in order to demonstrate the effectiveness of the proposed algorithm and the improvements that can be obtained by determining the routing decisions in the second stage.

The remainder of the paper is organized as follows. A literature review of the PRP, SPRP, and PH heuristic is provided in Section 1.2 to show how the current work relates to previous studies. The mathematical formulation and notations are provided in Section 1.3 while Section 1.4 describes the PH-based three-phase matheuristic algorithm. Computational experiments are presented in Section 1.5. Finally, we discuss our findings and draw conclusions in Section 1.6.

1.2 Literature Review

In this section, we summarize the relevant literature related to our study. Section 1.2.1 explores the deterministic PRP literature and the various algorithms employed to tackle this problem. Additionally, we offer a concise review of the different variations of the problem. Then, Section 1.2.2 focuses on studies addressing the stochastic PRP. Finally, we provide a brief overview of the literature concerning the PH algorithm in Section 1.2.3.

1.2.1 The deterministic PRP

First introduced by Chandra (1993), the PRP aims to improve supply chain efficiency and reduce costs by increasing collaboration between partners and incorporating production and distribution decisions into a single problem. The classic PRP considers a production plant, multiple customers, and a fleet of vehicles to transport the goods to customers. However, several variants of the problem have also been investigated to address specific real-world situations. In order to find high-quality solutions for the PRP in a reasonable amount of time, many heuristic and metaheuristic algorithms have been developed over the years. One heuristic approach that is commonly employed is to take advantage of the problem's structure and decompose it into less complex subproblems.

Boudia et al. (2008) introduce the "uncoupled" and the "coupled" heuristics to solve the problem in two phases. Bard and Nananukul (2009) provide a two-phase reactive tabu search (TS) algorithm with path-relinking. Archetti et al. (2011) compare the order-up-to-level (OU) and maximum level (ML) replenishment policies and propose a hybrid heuristic to solve the production and routing subproblems sequentially. Armentano et al. (2011) deal with a multi-product, multi-vehicle PRP and present two variants of TS as the solution algorithm. Adulyasak et al. (2014b) decompose the main problem into less complex subproblems and employ an ALNS-based heuristic. Absi et al. (2015) propose a two-phase heuristic that solves a two-level lot sizing problem with approximate routing costs and uses a back-and-forth iteration between the two phases to improve the solutions. Solyalı and Süral (2017) propose a multi-phase matheuristic algorithm that first solves a restricted PRP using a determined set of routes and improves the production and routing decisions in the following phases. Russell (2017) develops two matheuristic algorithms. The first one utilizes the set partitioning reformulation of the problem followed by a TS algorithm to improve the solutions. In the second algorithm, by assigning artificial demands to customers, they are divided into relatively similar-sized clusters, and approximate routes are constructed using the idea of seed routes. Vadseth et al. (2023) propose a path-flow formulation for the PRP and develop a new multi-start matheuristic to solve the problem. In each start of the algorithm, they generate an initial solution by employing a decomposition matheuristic. Then, they iterate over an improvement problem, which operates on a small set of routes to enhance the quality of the solutions. Ben Ahmed et al. (2023) introduce a multi-phase matheuristic and conduct an investigation into the performance of various mathematical formulations of the PRP solved using this algorithm.

Miranda et al. (2018) extend the decomposition heuristic presented by Absi et al. (2015) to solve a rich PRP derived from a real-world problem at a Brazilian furniture manufacturer. A heterogeneous fleet, multiple products, and routes that can extend over multiple days are all taken into account in their formulation. Considering the multi-product PRP with the possibility of outsourcing, Li et al. (2019) solve the problem using a three-level heuristic algorithm. Avci and Topaloglu Yildiz (2020) investigate the PRP with transshipment between retailers or between a retailer and the production plant. They propose a multi-phase matheuristic algorithm to solve the problem for small and medium size instances. A two-echelon PRP for the petrochemical industry is introduced by Schenekemberg et al. (2021) such that both pickups and deliveries may occur at suppliers, plants, and customers. A local search is combined with a branch-and-cut algorithm to solve the problem. Gruson et al. (2023) also discuss a two-echelon extension of the PRP, in which there are direct shipments from the plant to several warehouses, and routes have to be established for the distribution from the warehouses to the customers. They consider the case of split deliveries and split demand.

Manousakis et al. (2022) propose a two-commodity flow formulation for the PRP under the ML policy. A matheuristic algorithm is proposed that first solves the relaxed PRP considering approximate routing costs, and the routing problem is then enhanced using a Greedy Randomized Adaptive Search Procedure (GRASP). Finally, a local search algorithm that explores both feasible and infeasible search spaces is utilized to improve the quality of the solutions. In their study, Neves-Moreira et al. (2019) present a three-phase matheuristic algorithm to solve a multi-product PRP taking into account the perishability of the products and time windows. Rodrigues et al. (2023) introduce a memetic algorithm along with novel operators specifically tailored for tackling the multi-product PRP with a homogeneous fleet. A study by **Alvarez2022** addresses PRPs with perishable products and transshipment in which the profit decreases gradually over time as the products reach their shelf life. In order to solve the problem, they propose a hybrid heuristic algorithm based on iterative local search (ILS). Chitsaz et al. (2019) study the assembly routing problem (ARP), where there is an assembly unit that requires components from different suppliers to produce a final product. A unified decomposition matheuristic is developed that produces solutions to the ARP and related problems in three phases.

A few studies have introduced exact algorithms to solve small to medium size PRPs to optimality. Bard and Nananukul (2010) propose a branch-and-price algorithm to solve the PRP with a homogeneous fleet. Adulyasak et al. (2014a) propose four formulations for the PRP which differ with respect to the replenishment policy considered and whether or not they include a vehicle index. They develop a branch-and-cut algorithm to solve the problems utilizing several valid inequalities to strengthen the problem formulation. Qiu et al. (2019) study the PRP with perishable products and propose a mixed integer programming (MIP) formulation based on the Miller–Tucker–Zemlin (MTZ) subtour elimination constraints. A branch-and-cut algorithm using valid inequalities is applied to solve this problem. Chitsaz et al. (2020) extend the ARP to consider products that can be made by multiple suppliers, and they solve the problem using a branch-and-cut algorithm. In a study conducted by Schenekemberg et al. (2023), they introduce a novel three-front parallel algorithm designed to exploit the two-index and the three-index formulations using the Branch-and-Cut (BC) algorithm and a local search matheuristic approach, independently. The core concept of this algorithm involves the simultaneous solution of these three distinct fronts, all within an integrated framework that facilitates information sharing among these fronts. The experimental results demonstrate the remarkable performance of their algorithm. Specifically, it has exhibited the ability to identify previously undiscovered optimal solutions, as well as achieving the best-known solutions for a significant subset of benchmark instances.

Kumar et al. (2016) present a bi-objective formulation for the PRP where the second objective function evaluates the carbon emissions by considering the fuel consumption based on the vehicles' total travel time. Qiu et al. (2017) propose a PRP formulation for considering Greenhouse Gases (GHG) and specifically carbon emission costs under the carbon cap-and-trade policy and solve the problem using a branch-and-price algorithm. Qiu et al. (2018) propose a model with simultaneous pickup and delivery to incorporate remanufacturing into the PRP and solve it using a branch-and-price algorithm. This problem involves products being returned to remanufacturing facilities different from the production plant.

1.2.2 The stochastic PRP

As mentioned earlier, despite the significance of the SPRP, only a few studies have addressed this problem due to its complexity. Adulyasak et al. (2015a) introduce the stochastic PRP that considers the routing and production setup decision as the first-stage and the production and delivery quantities as the second-stage decisions. In their study, a Benders decomposition is combined with a branch-and-cut algorithm to solve the problem. In the two-stage case, the Benders algorithm is strengthened by lower bound lifting inequalities, scenario group cuts, and Pareto-optimal cuts. Mousavi et al. (2022) propose a two-stage stochastic programming formulation for the PRP with a single perishable product and demand uncertainty. In this study, a five-stage matheuristic solution approach based on the algorithm proposed by Solyalı and Süral (2017) is developed to solve the problem. Agra et al. (2018) propose a two-stage stochastic production-inventory routing problem in which production periods, quantities and routing decisions are the first-stage variables. The second-stage variables are the delivery quantities, inventory levels, and backlogging quantities. Several sets of valid inequalities are introduced and customized to the problem, and two solution algorithms based on the sample average approximation (SAA) approach are presented.

Zhang et al. (2018) study the PRP problem with recycling and remanufacturing in which both pickup and delivery can occur at a retailer. This research proposes a deterministic and a two-stage stochastic optimization problem considering demand uncertainty. In the stochastic problem, production quantities in both manufacturing and remanufacturing firms, as well as the routing decisions, are considered in the first stage, while the inventory and the number of worn-out items that are carried back to the remanufacturing firms are the second-stage variables. Shuang et al. (2019) study the PRP problem with recycling and remanufacturing with carbon taxes as well as the carbon cap-and-trade policy in a similar context.

In all prior SPRP studies, routing decisions are made in the first stage (SPRP-FSR) and unnecessary visits may occur in the second stage. In order to address this issue, we propose an adaptive routing strategy to reduce routing costs. Adaptive routing is especially relevant when delivery costs are high and more flexible decisions are required based on the actual demand. In each realized scenario, routing decisions are part of the solution that is influenced by both inventory and demand. Moreover, the static-static strategy has been neglected in the SPRP studies while it constitutes a significant part of the lot sizing problem (LSP) literature.

1.2.3 The progressive hedging algorithm

Routing decisions and customer visits that are scenario-specific increase the complexity of the problem. Thus, we develop an algorithm to obtain high-quality solutions to the problem. More specifically, we introduce a progressive hedging algorithm that is combined with a matheuristic to solve the proposed stochastic problem. The PH algorithm was first proposed by Rockafellar and Wets (1991) as a scenario decomposition approach. A number of studies have investigated the use of the PH approach in combination with other heuristics in order to obtain high-quality solutions to integer stochastic problems. Løkketangen and Woodruff (1996) present a PH-based TS for stochastic binary MIP problems. A study by Haugen et al. (2001) implements a PH algorithm for solving the stochastic LSP.

Crainic et al. (2011) propose a heuristic adjustment strategy and compare the heuristic and the Lagrangean strategies combined with the TS algorithm for a time-dependent stochastic network design problem. Alvarez et al. (2021) solve the two-stage stochastic IRP with stochastic demand and supply using a heuristic PH algorithm and an ILS heuristic. They consider visit and routing decisions as the first-stage variables and propose an aggregation strategy for each of the three possible cases, i.e., consensus on both sets of variables, consensus on visit variables but not on routing variables or consensus on neither. In this study, we utilize the problem's structure and employ a PH algorithm combined with an efficient decomposition heuristic to solve the problem.

1.3 Problem Formulation

This section presents a formulation for the two-stage stochastic PRP with demand uncertainty and adaptive routing for a single product that must be produced and delivered to N customers, where $\mathscr{N} = \{0, ..., N\}$ is the set of all nodes including $\{0\}$, which denotes the production plant, and $\mathscr{N}_c$, which represents the set of customers. Let $G = (\mathscr{N}, \mathscr{E})$ be a complete and undirected graph where $\mathscr{E} = \{(i, j) : i, j \in \mathscr{N}, i < j\}$ represents the set of edges that connect each pair of nodes in $\mathscr{N}$. Homogeneous vehicles with capacity $\mathscr{Q}$ deliver the product to the customers, and they must start and end their routes at the production plant in each period. The set of all vehicles is defined by $\mathscr{K} = \{1, ..., K\}$. We consider a discrete and finite planning horizon of T periods, and the set of periods is denoted by $\mathscr{T} = \{1, ..., T\}$. In order to model demand uncertainty, we define a finite set of scenarios $\phi = \{1, ..., S\}$, each of which may occur with probability $\pi_\varphi > 0$, $\forall \varphi \in \phi$ and $\sum_{\varphi \in \phi} \pi_\varphi = 1$. The demand of each customer is assumed to be a random variable with a known distribution, and the realized demand of customer $i \in \mathscr{N}_c$ in period $\tau \in \mathscr{T}$ under scenario $\varphi \in \phi$ is denoted by $d_{i\tau}^{\varphi}$.

In this study, we adopt two of the strategies proposed by Bookbinder and Tan (1988) for the stochastic LSP: The static-static strategy in which both setup and production quantity decisions are made prior to the demand realization and the more flexible static-dynamic strategy where production quantities are determined in the second stage while setup decisions are made in the first stage. The visiting, delivery quantity, outsourcing, and routing decisions are made in the second stage in both variants of the PRP.

1.3.1 The static-dynamic strategy

In this section, we present the formulation of the problem under the static-dynamic strategy. The binary variable y_τ takes value 1 if production takes place in period $\tau \in \mathscr{T}$, with a fixed setup cost of F, and 0 otherwise. In addition, a per unit production cost u applies to the production quantity, which is represented by variable p_τ^φ . This production quantity is bounded by the production capacity C. The integer variable $x_{ij\kappa\tau}^\varphi$ is a recourse variable that denotes how many times edge $(i,j) \in \mathscr{E}$ is visited by vehicle $\kappa \in \mathscr{K}$ in period $\tau \in \mathscr{T}$ under scenario $\varphi \in \phi$. Traversing edge (i,j) by a vehicle induces a transportation cost of c_{ij}. A unit holding cost of h_i applies to the inventory at the end of each period for all nodes $i \in \mathscr{N}$. Accordingly, we define variable $I_{i\tau}^\varphi$ as the inventory of node $i \in \mathscr{N}$ at the end of each period τ and for each scenario φ. An initial inventory of I_{i0} exists at each node $i \in \mathscr{N}$ at the beginning of the planning horizon. If customer $i \in \mathscr{N}_c$ faces a stockout, outsourcing to a third-party supplier can take place with cost β_i. The recourse variable $o_{i\tau}^\varphi$ represents the amount of demand at this customer which is satisfied using the outsourcing option under scenario φ. We define binary variable $z_{i\kappa\tau}^\varphi$ to take value 1 if node $i \in \mathscr{N}$ is visited by vehicle $\kappa \in \mathscr{K}$ in period $\tau \in \mathscr{T}$ under scenario $\varphi \in \phi$, and 0 otherwise. In addition, variable $q_{i\kappa\tau}^\varphi$ denotes the amount delivered to customer $i \in \mathscr{N}_c$ by vehicle $\kappa \in \mathscr{K}$ in period τ under scenario φ. Table 1.2 provides a summary of all sets, parameters, and decision variables used in the mathematical formulation. The SPRP with adaptive routing (SPRP-AR) and assuming the static-dynamic strategy ($SPRP-AR_{SD}$) can now be formulated as the following MIP:

$$(SPRP-AR_{SD}) \quad \min \sum_{\tau \in \mathscr{T}} \Bigg(F y_\tau + \sum_{\varphi \in \phi} \pi_\varphi \Big(u p_\tau^\varphi + \sum_{(i,j) \in \mathscr{E}} \sum_{\kappa \in \mathscr{K}} c_{ij} x_{ij\kappa\tau}^\varphi + \sum_{i \in \mathscr{N}} h_i I_{i\tau}^\varphi + \sum_{i \in \mathscr{N}_c} \beta_i o_{i\tau}^\varphi \Big) \Bigg) \quad (1.1)$$

s.t.

$$p_\tau^\varphi \le \mathscr{M}_\tau^\varphi y_\tau \qquad \forall \tau \in \mathscr{T}, \varphi \in \phi \quad (1.2)$$

$$I_{0\tau}^\varphi = I_{0,\tau-1}^\varphi + p_\tau^\varphi - \sum_{i \in \mathscr{N}_c} \sum_{\kappa \in \mathscr{K}} q_{i\kappa\tau}^\varphi \qquad \forall \tau \in \mathscr{T}, \varphi \in \phi \quad (1.3)$$

$$I_{i\tau}^\varphi = I_{i,\tau-1}^\varphi + \sum_{\kappa \in \mathscr{K}} q_{i\kappa\tau}^\varphi + o_{i\tau}^\varphi - d_{i\tau}^\varphi \qquad \forall i \in \mathscr{N}_c, \tau \in \mathscr{T}, \varphi \in \phi \quad (1.4)$$

$$I_{0\tau}^\varphi \le L_0 \qquad \forall \tau \in \mathscr{T}, \varphi \in \phi \quad (1.5)$$

$$I_{i\tau}^\varphi + d_{i\tau}^\varphi \le L_i \qquad \forall i \in \mathscr{N}_c, \tau \in \mathscr{T}, \varphi \in \phi \quad (1.6)$$

$$\sum_{i \in \mathscr{N}_c} q_{i\kappa\tau}^\varphi \le \mathscr{Q} z_{0\kappa\tau}^\varphi \qquad \forall \kappa \in \mathscr{K}, \tau \in \mathscr{T}, \varphi \in \phi \quad (1.7)$$

$$\sum_{\kappa \in \mathscr{K}} z_{i\kappa\tau}^\varphi \le 1 \qquad \forall i \in \mathscr{N}_c, \tau \in \mathscr{T}, \varphi \in \phi \quad (1.8)$$

$$q_{i\kappa\tau}^\varphi \le \mathscr{W}_{i\tau}^\varphi z_{i\kappa\tau}^\varphi \qquad \forall i \in \mathscr{N}_c, \kappa \in \mathscr{K}, \tau \in \mathscr{T}, \varphi \in \phi \quad (1.9)$$

$$\sum_{(j,j') \in \varepsilon(\{i\})} x_{jj'\kappa\tau}^\varphi = 2 z_{i\kappa\tau}^\varphi \qquad \forall i \in \mathscr{N}, \kappa \in \mathscr{K}, \tau \in \mathscr{T}, \varphi \in \phi \quad (1.10)$$

$$\sum_{(i,j) \in \mathscr{E}(\eta)} x_{ij\kappa\tau}^\varphi \le \sum_{i \in \eta} z_{i\kappa\tau}^\varphi - z_{e\kappa\tau}^\varphi$$

$$\forall \eta \subseteq \mathscr{N}_c, |\eta| \ge 2, e \in \eta, \kappa \in \mathscr{K}, \tau \in \mathscr{T}, \varphi \in \phi \quad (1.11)$$

$$y_\tau \in \{0,1\} \qquad \forall \tau \in \mathscr{T} \quad (1.12)$$

$$p_\tau^\varphi \ge 0 \qquad \forall \tau \in \mathscr{T}, \varphi \in \phi \quad (1.13)$$

$$q_{i\kappa\tau}^\varphi \ge 0 \qquad \forall i \in \mathscr{N}, \kappa \in \mathscr{K}, \tau \in \mathscr{T}, \varphi \in \phi \quad (1.14)$$

$$I_{i\tau}^\varphi \ge 0 \qquad \forall i \in \mathscr{N}, \tau \in \mathscr{T}, \varphi \in \phi \quad (1.15)$$

$$o_{i\tau}^\varphi \ge 0 \qquad \forall i \in \mathscr{N}, \tau \in \mathscr{T}, \varphi \in \phi \quad (1.16)$$

$$z_{i\kappa\tau}^\varphi \in \{0,1\} \qquad \forall i \in \mathscr{N}, \kappa \in \mathscr{K}, \tau \in \mathscr{T}, \varphi \in \phi \quad (1.17)$$

$$x_{ij\kappa\tau}^\varphi \in \{0,1\} \qquad \forall (i,j) \in \mathscr{E}, i \neq 0, \kappa \in \mathscr{K}, \tau \in \mathscr{T}, \varphi \in \phi \quad (1.18)$$

$$x_{0j\kappa\tau}^\varphi \in \{0,1,2\} \qquad \forall j \in \mathscr{N}_c, \kappa \in \mathscr{K}, \tau \in \mathscr{T}, \varphi \in \phi. \quad (1.19)$$

The objective function (1.1) minimizes the setup costs and the expected cost of the second-stage variables including the production, routing, holding, and outsourcing costs. Constraints (1.2) impose that the setup must take place if production is required in a period. The production quantity in each period is bounded by the minimum of either the

Table 1.2: Sets, parameters, and decision variables.

Sets:	
$\mathscr{N}$	Set of nodes, $i \in \{0,...,N\}$, $\{0\}$ is the plant.
$\mathscr{N}_c$	Set of customers, $i \in \{1,...,N\}$.
$\mathscr{E}$	Set of edges, $\mathscr{E} = \{(i,j) : i,j \in \mathscr{N}, i < j\}$.
$\mathscr{T}$	Set of time periods, $\tau \in \{0,...,T\}$.
$\mathscr{K}$	Set of vehicles, $\kappa \in \{1,...,K\}$.
ϕ	Set of scenarios, $\varphi \in \{1,...,S\}$.
$\mathscr{E}(\eta)$	Subset of edges $(i,j) \in \mathscr{E}$ such that $i,j \in \eta$ and $\eta \subseteq \mathscr{N}$ is a given set of nodes.
$\varepsilon(\{i\})$	Subset of edges incident to node $i \in \mathscr{N}$.
Parameters:	
F	Setup cost.
u	Unit production cost.
C	Production capacity.
c_{ij}	Cost of visiting edge $(i,j) \in \mathscr{E}$.
h_i	Per unit holding cost at node $i \in \mathscr{N}$.
$\mathscr{Q}$	Vehicle capacity.
β_i	Unit outsourcing cost at node $i \in \mathscr{N}_c$.
π_φ	Probability of scenario $\varphi \in \phi$.
$d_{i\tau}^{\varphi}$	Demand of node $i \in \mathscr{N}_c$, in period $\tau \in \mathscr{T}$, under scenario $\varphi \in \phi$.
I_{i0}	Initial inventory at node $i \in \mathscr{N}$.
L_i	Storage capacity at node $i \in \mathscr{N}$.
Decision variables:	
y_τ	Setup decision, equal to 1 if setup takes place at period $\tau \in \mathscr{T}$, and 0 otherwise.
p_τ	Production quantity at period $\tau \in \mathscr{T}$ (for $SPRP-AR_{SS}$).
p_τ^{φ}	Production quantity at period $\tau \in \mathscr{T}$ under scenario $\varphi \in \phi$ (for $SPRP-AR_{SD}$).
$x_{ij\kappa\tau}^{\varphi}$	Number of times edge $(i,j) \in \mathscr{E}$ is traversed by vehicle $\kappa \in \mathscr{K}$ in period $\tau \in \mathscr{T}$ under scenario $\varphi \in \phi$.
$I_{i\tau}^{\varphi}$	Inventory of node $i \in \mathscr{N}$ at the end of each period $\tau \in \mathscr{T}$ under scenario $\varphi \in \phi$.
$z_{i\kappa\tau}^{\varphi}$	Visiting decision, equal to 1 if node $i \in \mathscr{N}$ is visited by vehicle $\kappa \in \mathscr{K}$ in period $\tau \in \mathscr{T}$ under scenario $\varphi \in \phi$, and 0 otherwise.
$o_{i\tau}^{\varphi}$	Quantity of outsourced products at node $i \in \mathscr{N}_c$ in period $\tau \in \mathscr{T}$ under scenario $\varphi \in \phi$.
$q_{i\kappa\tau}^{\varphi}$	Amount of product delivered to customer $i \in \mathscr{N}_c$ by vehicle $\kappa \in \mathscr{K}$ in period $\tau \in \mathscr{T}$ under scenario $\varphi \in \phi$.

production capacity or the sum of all customer demands for the rest of the periods, denoted by $\mathscr{M}_\tau^{\varphi} = \min\left\{\mathscr{C}, \sum_{l=\tau}^{T} \sum_{i\in\mathscr{N}_c} d_{il}^{\varphi}\right\}$. Constraints (1.3) and (1.4) are the inventory balance constraints for the plant and customers, respectively. Each node has a maximum storage capacity of L_i. Constraints (1.5) control the inventory capacity at the plant and constraints (1.6) impose that the inventory of each customer at the end of the period plus the demand of that period cannot exceed the capacity of that customer. Constraints (1.7) force the vehicle to leave the plant if it is assigned to deliver products to customers and impose the vehicle capacity. Constraints (1.8) and (1.9) indicate that each customer can only be visited by one vehicle in a period (split deliveries are not allowed) and the maximum delivery to a customer must not exceed the smallest value of the vehicle capacity, customer's storage, or the customer's demand for the remaining periods of the planning horizon, which is indicated by $\mathscr{W}_{i\tau}^{\varphi} = \min\left\{\mathscr{Q}, L_i, \sum_{l=\tau}^{T} d_{il}^{\varphi}\right\}$.

We define $\mathscr{E}(\eta)$ as a set of edges $(i,j) \in \mathscr{E}$ such that $i,j \in \eta$ and $\eta \subseteq \mathscr{N}$ is a given set of nodes. Moreover, $\varepsilon(\{i\})$ is the set of edges incident to node $i \in \mathscr{N}$. Constraints (1.10) indicate that two edges must be incident to a visited node and constraints (1.11) are the subtour elimination constraints (SEC). As mentioned earlier, the routing problem depends on the realized scenario, which means that the routing is decided after the demand realization. Therefore, these constraints must be considered for every possible demand realization and all periods. Finally, constraints (1.12)−(1.19) define the domain of the decision variables.

1.3.2 The static-static strategy

The only distinction between the static-static and static-dynamic strategies lies in the timing of production decisions within the problem framework. Specifically, the static-static strategy entails making production decisions in the first stage, while the static-dynamic strategy involves making such decisions in the second stage. Consequently, p_τ is defined as the first-stage variable for the production quantity in the static-static case, while the remaining decision variables remain consistent between the two models. We introduce a formulation, called $SPRP-AR_{SS}$ for the static-static strategy, assuming that production quantities are also determined in the first stage:

$$(SPRP-AR_{SS}) \quad \min \sum_{\tau\in\mathscr{T}} \Bigg(Fy_\tau + up_\tau + \sum_{\varphi\in\phi} \pi_\varphi \Big(\sum_{(i,j)\in\mathscr{E}} \sum_{\kappa\in\mathscr{K}} c_{ij} x^{\varphi}_{ij\kappa\tau} + \sum_{i\in\mathscr{N}} h_i I^{\varphi}_{i\tau} + \sum_{i\in\mathscr{N}_c} \beta_i o^{\varphi}_{i\tau} \Big) \Bigg) \tag{1.20}$$

s.t. (1.4) - (1.12) and (1.14) - (1.19),

$$p_\tau \leq \mathscr{M}^{(max)}_\tau y_\tau \qquad \forall \tau \in \mathscr{T} \tag{1.21}$$

$$I^{\varphi}_{0\tau} = I^{\varphi}_{0,\tau-1} + p_\tau - \sum_{i\in\mathscr{N}_c} \sum_{\kappa\in\mathscr{K}} q^{\varphi}_{i\kappa\tau} \qquad \forall \tau \in \mathscr{T}, \varphi \in \phi \tag{1.22}$$

$$p_\tau \geq 0 \qquad \forall \tau \in \mathscr{T}. \tag{1.23}$$

In the above formulation, the objective function (1.20) minimizes the production costs instead of their expected value since production is no longer a scenario-dependent variable.

In addition, in constraints (1.21) the production quantity must be decided regardless of the scenario and is bounded by $\mathscr{M}^{max}_\tau = \min\left\{\mathscr{C}, \max_{\varphi\in\phi}\{\sum_{l=\tau}^{T} \sum_{i\in\mathscr{N}_c} d^{\varphi}_{il}\}\right\}$. Constraints (1.22) are the adapted inventory balance constraints at the plant to take into account the scenario-independent production quantity.

1.4 Progressive Hedging-Based Matheuristic Algorithm

The goal of this section is to introduce a PH algorithm that is combined with a three-phase matheuristic in order to solve the SPRP-AR. This method works by replacing the first-stage variables with scenario-specific variables, leading to a set of independent subproblems representing each scenario, thereby reducing the complexity of the problem. It should be noted that the solution to each subproblem is only optimal in the context of that specific scenario. Accordingly, in order to reach a consensus over all the variables involved in the first stage of the stochastic process and achieve a feasible solution, an augmented Lagrangean strategy is adopted.

The primary motivation for choosing the PH algorithm lies in its efficiency in addressing stochastic problems featuring binary variables, particularly in the first phase. By having the routing decisions in the second stage, only a small subset of binary variables serve as the first-stage variables, making the PH a good choice for solving the SPRP-AR. Additionally, the structure of the SPRP-AR allows for decomposition, leading to deterministic PRP subproblems. Although the deterministic PRP is challenging to solve, recent studies have developed efficient heuristics for it. This led us to employ the PH algorithm, integrated with a modified version of one of the most efficient matheuristic algorithms for solving deterministic PRPs (Solyalı and Süral, 2017).

In order to implement the PH algorithm, we must duplicate the first-stage variables and divide the problem into scenario-specific subproblems, which we call SSPRP$^{(\varphi)}$ for the static-static strategy and SDPRP$^{(\varphi)}$ for the static-dynamic strategy. The decomposition procedure and the formulations are described in Section 1.4.1 for both strategies. After decomposing the original problem into scenario-specific subproblems, we essentially encounter S deterministic PRPs. While solving these problems is comparatively easier than tackling the stochastic problem, they are still challenging to solve. Therefore, an effective heuristic algorithm is necessary to solve this problem efficiently. The heuristic solution algorithm for the scenario-specific subproblem consists of three phases: the first phase involves finding an a priori tour over all nodes, the second phase focuses on solving restricted PRPs (RPRPs), and the third phase is dedicated to solving CVRPs for each period. When integrating this heuristic with the PH algorithm, we only need to solve the TSP once at the beginning. Furthermore, the third phase (solving CVRPs) is only executed for the best solution found by the PH algorithm. The three-phase algorithm and its integration with the PH approach are explained in Section 1.4.3.

In this study, the heuristic adjustment approach proposed by Crainic et al. (2011) is used to accomplish the goal of getting consensus on the first-stage variables. Throughout the algorithm, costs associated with each of these variables can be adjusted at every iteration to achieve consensus. After the subproblems have been solved sequentially in each iteration, the stopping criteria are checked to determine whether to stop the algorithm or proceed to the next iteration with the updated parameters. At the end of each iteration, the following four criteria must be evaluated:

- If a consensus over all variables has been achieved;
- If the maximum number of iterations has been reached;
- If the maximum run-time has been exceeded;
- If the number of non-improving iterations has reached a specified number.

If the first criterion is met (consensus on all first-stage variables), the solution is feasible for the stochastic problem. If, after several iterations, the inconsistency persists, an aggregation strategy is applied to obtain a feasible solution. The adjustment and the aggregation strategy are discussed in detail in Section 1.4.2. The pseudo-code of the overall algorithm is given in Algorithm 1.

In the context of the PH algorithm, exploration and exploitation play crucial roles in guiding the search process towards high-quality solutions. The algorithm starts with the exploration phase, where it systematically decomposes the original problem into independent subproblems. During this phase, the algorithm searches through a wide range of possible solutions to gain insights into the problem structure and identify potential regions of interest in the solution space. As the algorithm progresses, it transitions towards the exploitation phase, focusing on refining solutions and improving convergence. In this phase, the algorithm modifies the costs of the first-stage variables in the objective function to encourage convergence towards optimal values. By adjusting these costs, the algorithm directs its search towards regions of the solution space likely to yield high-quality solutions. Through a balanced combination of exploration and exploitation, the PH algorithm effectively navigates complex optimization problems, iteratively refining solutions to converge towards optimal or near-optimal outcomes.

Algorithm 1 PH-based matheuristic algorithm

```
Solve TSP:
return α_i and σ_i for all i ∈ 𝒩
Initialize PH:
    v ← 0
    for φ ∈ ϕ do
        F̄_τ^{φ(v)} ← F
        Solve RPRP^(φ)
        Set ReferenceSolution (Eq. (1.38))
        Set BestSolution (Eq. (1.42))
    end for
End Initialization
repeat
    v ← v + 1
    Perform global adjustment (Eq. (1.39))
    for φ ∈ ϕ do
        Perform local adjustment (Eq. (1.40))
        Fix eligible variables (Eq. (1.41))
        Solve RPRP^(φ)
        Update ReferenceSolution (Eq. (1.38))
        Update BestSolution (Eq. (1.42))
    end for
until Stopping criterion is met
return Setup, production, and inventory decisions
for φ ∈ ϕ do
    for τ ∈ 𝒯 do
        Solve CVRP
    end for
end for
return Routing and delivery decisions
```

1.4.1 Scenario decomposition

One can observe that the SPRP-AR formulations presented in Section 1.3 have a block-diagonal structure, where each block represents a deterministic PRP for a given scenario $\varphi \in \phi$. Constraints (1.2) and constraints (1.21) and (1.22) are the linking constraints that connect the first-stage and the second-stage variables. Thus, by duplicating the setup variables (y_τ), we introduce a specific setup decision variable y_τ^φ for each scenario $\varphi \in \phi$

and the static-dynamic problem can be reformulated as follows:

$$(PRP_{SD}) \quad \min \sum_{\varphi\in\phi} \pi_\varphi \Bigg(\sum_{\tau\in\mathscr{T}} \Big(Fy^\varphi_\tau + up^\varphi_\tau + \sum_{(i,j)\in\mathscr{E}} \sum_{\kappa\in\mathscr{K}} c_{ij} x^\varphi_{ij\kappa\tau} + \sum_{i\in\mathscr{N}} h_i I^\varphi_{i\tau} + \sum_{i\in\mathscr{N}_c} \beta_i o^\varphi_{i\tau} \Big) \Bigg) \tag{1.24}$$

s.t. (1.3) - (1.11) and (1.13) - (1.19),

$$p^\varphi_\tau \le \mathscr{M}^\varphi_\tau y^\varphi_\tau \qquad \forall\tau\in\mathscr{T}, \varphi\in\phi \tag{1.25}$$

$$y^\varphi_\tau = \bar{y}_\tau \qquad \forall\tau\in\mathscr{T}, \varphi\in\phi \tag{1.26}$$

$$y^\varphi_\tau \in \{0,1\} \qquad \forall\tau\in\mathscr{T}, \varphi\in\phi \tag{1.27}$$

$$\bar{y}_\tau \in \{0,1\} \qquad \forall\tau\in\mathscr{T}. \tag{1.28}$$

Scenario-decomposable constraints (1.25) replace the original linking constraints in the above formulation, while the non-anticipativity constraints (1.26) ensure that the setup decisions do not depend on the scenario and that the final solution of the first-stage variables is consistent across all scenarios. In this regard $\bar{y}_\tau \in \{0,1\}$, $\forall\tau\in\mathscr{T}$ is defined and is known as the "overall design vector".

In a similar manner, by duplicating the setup and production variables (y_τ and p_τ), and defining y^φ_τ and p^φ_τ for each scenario $\varphi\in\phi$ for the static-static case, we can reformulate the second problem as follows:

$$(PRP_{SS}) \quad \min \sum_{\varphi\in\phi} \pi_\varphi \Bigg(\sum_{\tau\in\mathscr{T}} \Big(Fy^\varphi_\tau + up^\varphi_\tau + \sum_{(i,j)\in\mathscr{E}} \sum_{\kappa\in\mathscr{K}} c_{ij} x^\varphi_{ij\kappa\tau} + \sum_{i\in\mathscr{N}} h_i I^\varphi_{i\tau} + \sum_{i\in\mathscr{N}_c} \beta_i o^\varphi_{i\tau} \Big) \Bigg) \tag{1.29}$$

s.t. (1.4) - (1.11) and (1.14) - (1.19),

$$p_{\tau}^{\varphi} \leq \mathscr{M}_{\tau}^{max} y_{\tau}^{\varphi} \qquad \forall \tau \in \mathscr{T}, \varphi \in \phi \quad (1.30)$$

$$I_{0\tau}^{\varphi} = I_{0,\tau-1}^{\varphi} + p_{\tau}^{\varphi} - \sum_{i \in \mathscr{N}_c} \sum_{\kappa \in \mathscr{K}} q_{i\kappa\tau}^{\varphi} \qquad \forall \tau \in \mathscr{T}, \varphi \in \phi \quad (1.31)$$

$$y_{\tau}^{\varphi} = \bar{y}_{\tau} \qquad \forall \tau \in \mathscr{T}, \varphi \in \phi \quad (1.32)$$

$$p_{\tau}^{\varphi} = \bar{p}_{\tau} \qquad \forall \tau \in \mathscr{T}, \varphi \in \phi \quad (1.33)$$

$$y_{\tau}^{\varphi} \in \{0,1\} \qquad \forall \tau \in \mathscr{T}, \varphi \in \phi \quad (1.34)$$

$$\bar{y}_{\tau} \in \{0,1\} \qquad \forall \tau \in \mathscr{T} \quad (1.35)$$

$$p_{\tau}^{\varphi} \geq 0 \qquad \forall \tau \in \mathscr{T}, \varphi \in \phi \quad (1.36)$$

$$\bar{p}_{\tau} \geq 0 \qquad \forall \tau \in \mathscr{T}. \quad (1.37)$$

Again, the original linking constraints are replaced by constraints (1.30) and (1.31), the non-anticipativity constraints (1.32) and (1.33) are added, and $\bar{y}_{\tau} \in \{0,1\}$, $\forall \tau \in \mathscr{T}$ and $\bar{p}_{\tau} \geq 0$, $\forall \tau \in \mathscr{T}$ are defined as the overall design vector.

Rockafellar and Wets (1991) propose an augmented Lagrangean approach to relax non-anticipativity constraints (1.26), (1.32), and (1.33) and make the problem scenario-decomposable. Using Lagrangean multipliers, the relaxed constraints are incorporated into the objective function. These multipliers are updated at each iteration in order to maintain convergence. It is shown that this strategy leads to a global optimum in continuous problems, whereas this is not necessarily the case for integer problems (Rockafellar and Wets, 1991). In this study, we employ the heuristic adjustment strategy proposed by Crainic et al. (2011) to lead the first-stage variables to a consensus as it is shown to be more efficient, especially for binary first-stage variables.

1.4.2 Adjustment Strategy

The heuristic adjustment strategy relies on adjusting the cost of the first-stage variables based on their frequency in the solutions of the scenario subproblems instead of updating the Lagrangean multipliers. We use a similar approach to that introduced in Crainic et al.

(2011) to modify the setup costs during each iteration. As noted earlier, the solution to the SDPRP$^{(\varphi)}$(SSPRP$^{(\varphi)}$) is only optimal for a given scenario, while through an aggregation operator, we can benefit from these local solutions to lead our search toward the consensus of the first-stage variables. Thus, we define the reference solution $\bar{y}_\tau^{(v)}$ for each iteration v, which is the weighted sum of scenario-determined setup decisions, i.e., $\hat{y}_\tau^{\varphi(v)}$, where the probability of the occurrence of a scenario is considered as the weight of that scenario. The reference solution for the setup variables that can be used in both strategies is calculated for a given iteration v as follows:

$$\bar{y}_\tau^{(v)} = \sum_{\varphi \in \phi} \pi_\varphi \hat{y}_\tau^{\varphi(v)} \qquad \forall \tau \in \mathscr{T}. \quad (1.38)$$

For a given period $\tau \in \mathscr{T}$, $\bar{y}_\tau^{(v)} \in \{0,1\}$ implies that the consensus is achieved for this period over all scenarios in this iteration. If this situation applies to all first-stage variables, it means that the algorithm reached a consensus and the solution is feasible. However, the value of setup decisions in the reference solution is usually such that $0 < \bar{y}_\tau^{(v)} < 1$, which results in an infeasible solution due to the integrality conditions on the setup variables. Nevertheless, this solution contains valuable information about the likelihood of production occurring in a given period. Global adjustment strategies can use this information to guide the algorithm toward consensus. Lower values of $\bar{y}_\tau^{(v)}$ indicate that it is less probable that production takes place at that period in the final solution, while higher values state the opposite. In this regard, we define two thresholds θ_L and θ_H on $\bar{y}_\tau^{(v)}$ values of a given period. If $\bar{y}_\tau^{(v)}$ is greater than θ_H the corresponding cost is decreased while for values less than θ_L the setup cost is increased. Thus, the global adjustment strategy (1.39) aims to modify the setup costs according to the average value of variables through all scenarios:

$$\bar{F}_\tau^{(v)} = \begin{cases} \lambda \bar{F}_\tau^{(v-1)} & \text{if } \bar{y}_\tau^{(v-1)} < \theta_L \\ \frac{1}{\lambda} \bar{F}_\tau^{(v-1)} & \text{if } \bar{y}_\tau^{(v-1)} > \theta_H \\ \bar{F}_\tau^{(v-1)} & \text{otherwise,} \end{cases} \quad (1.39)$$

where $\lambda > 1$ is the modification rate and $\bar{F}_\tau^{(v)}$ is the modified global setup cost for period τ in iteration v in all scenarios to replace F in (1.43). Note that $0 < \theta_L < 0.5$ and $0.5 <$

$\theta_H < 1$ in equation (1.39). Moreover, another adjustment can be applied based on the solution of a specific scenario. In this regard, when the absolute difference of a scenario's solution from its reference point is larger than γ_F, the corresponding cost must be adjusted as a factor of λ. Thus, we define the local adjustment strategy (1.40) to be only applied locally for a specific scenario φ to the corresponding costs of variables that are far from their reference point:

$$\bar{F}_\tau^{\varphi(v)} = \begin{cases} \lambda \bar{F}_\tau^{(v)} & \text{if } |y_\tau^{\varphi(v-1)} - \bar{y}_\tau^{(v-1)}| \geq \gamma_F \text{ and } y_\tau^{\varphi(v-1)} = 1 \\ \frac{1}{\lambda} \bar{F}_\tau^{(v)} & \text{if } |y_\tau^{\varphi(v-1)} - \bar{y}_\tau^{(v-1)}| \geq \gamma_F \text{ and } y_\tau^{\varphi(v-1)} = 0 \\ \bar{F}_\tau^{(v)} & \text{otherwise,} \end{cases} \tag{1.40}$$

where the distance threshold is $0.5 < \gamma_F < 1$ and $\bar{F}_\tau^{\varphi(v)}$ is the modified local setup cost for scenario $\varphi \in \phi$. Finally, we define another threshold $0 < \gamma_N < 0.5$ to fix the solutions that are close to the reference point for the next iteration, using the following equation:

$$y_\tau^{\varphi(v)} = y_\tau^{\varphi(v-1)} \qquad \text{if } |y_\tau^{\varphi(v-1)} - \bar{y}_\tau^{(v-1)}| \leq \gamma_N. \tag{1.41}$$

The above equation can improve the speed of the algorithm by reducing the size of the problem at each iteration. At the end of each iteration, we need to define the best solution that aggregates all variables to provide a feasible solution to the master problem. This function fixes the setup decisions to one, even if production is required for a single scenario in that period. Thus, the best solution is calculated using the following equation:

$$y_\tau^{best(v)} = \max_{\varphi \in \phi} \left(y_\tau^{\varphi(v)} \right) \qquad \forall \tau \in \mathscr{T}. \tag{1.42}$$

If the PH algorithm stops due to the convergence of the first-stage variables, the solution is feasible for the original problem, allowing the algorithm to proceed to the next phase. However, if the algorithm stops because of a stopping criterion other than convergence, it implies that a consensus among all scenarios concerning the first-stage variables has not been reached. In such cases, we designate the values of $y_\tau^{best(v)}$ and the corresponding second-stage solutions as the best feasible solution for the problem. If the setup decision of a period has the value 1 in at least one of the subproblems, its value is fixed to 1 in the final solution. As a result, all subproblems become feasible, guaranteeing the feasibility of the solution for the original problem.

1.4.3 Solving Subproblems

The purpose of this section is to present a three-phase matheuristic algorithm for solving the scenario-specific subproblems sequentially. We must solve S deterministic PRPs per iteration in order to solve the SPRP-AR. The performance of the algorithm is improved by applying some modifications, since this is a demanding task even for a heuristic algorithm. The first step in the process consists of solving a TSP disregarding the capacity and demands of each node to determine an a priori tour over all nodes. This problem only needs to be solved once at the beginning, as the optimal TSP tour is the same for all subproblems. A restricted problem can then be solved using the sequence obtained in the first phase. The restricted PRPs (RPRP) must be solved for each scenario at each iteration until one of the stopping criteria of the PH algorithm is met. During the third phase of the algorithm, once the production and delivery variables have been fixed, a CVRP is solved for each pair of period and scenario, in order to enhance the quality of the routing decisions. Note that this third phase is not executed for every subproblem in the PH, but only at the end of the PH for the best solution.

Our algorithm draws inspiration from the five-phase heuristic developed by Solyalı and Süral (2017), a methodology that has demonstrated promising outcomes for both the PRP and IRP (Solyalı and Süral, 2022). Nonetheless, our proposed algorithm exhibits two key distinctions from the referenced approach. Firstly, in the second stage of our algorithm, we address the routing for each vehicle individually, contrasting with the aggregate constraint considered in the prior algorithm. Consequently, our algorithm yields a feasible solution to the original problem by the end of the second stage, whereas the referenced algorithm necessitates an additional step to modify the solution for feasibility. This adaptation proved advantageous in tackling our stochastic problem, which is notably more complex than the deterministic one. Our results showcase that we achieved high-quality solutions by employing a more streamlined process. Secondly, we strategically integrated the algorithmic structure with the progressive hedging algorithm, enabling us to attain high-quality solutions within a reasonable timeframe for the stochastic problem. It is worth noting that utilizing the original algorithm would have introduced an extra feasibility step, making the problem complex and nearly intractable for larger instances.

The First Phase: A priori tour

In this phase, the Concorde solver is used to solve a TSP by considering all nodes while ignoring their demands (Applegate et al., 2020). This yields an a priori tour that starts from the plant, visits all customers and eventually returns to the starting point. Using the solution of this problem we define two sets α_i and σ_i for each node $i \in \mathscr{N}$ which will be used to impose the sequence of the a priori tour in the second-phase problem. The set α_i denotes the nodes that can be visited after node i, which includes the plant for all $i \in \mathscr{N}_c$ since the vehicle can return to the plant after making a delivery to a customer. The set σ_i defines the nodes that can be reached prior to node i, which also always includes $\{0\}$ since the vehicle can go to any customer after it departs from the plant. Finally, α_0 and σ_0 contain all the customers, meaning that they can be visited both before and after the plant since routes start and end at the plant.

The Second Phase: The restricted PRP

In the second stage of the algorithm, we present an MIP formulation that uses the visiting sequence obtained from phase one to provide feasible solutions to the main problem. The sequence of the a priori tour is imposed, but it is possible to skip nodes. Solving this restricted stochastic problem is easier since SECs are no longer required. In order to solve this problem, we temporarily change the structure of the problem into a directed graph, and we assume the same transportation costs for both directions between any two vertices. In the current formulation, imposing an integrality condition on $x^{\varphi}_{ij\kappa\tau}$ is no longer necessary since integer $z^{\varphi}_{i\kappa\tau}$ variables in combination with sequence sets (α_i and σ_i) instead of

SECs result in the integrality property on continuous $x^{\varphi}_{ij\kappa\tau}$. Thus, we replace the routing variables with the continuous variables $0 \leq \delta^{\varphi}_{ij\kappa\tau} \leq 1$. The formulation for the $RPRP^{(\varphi)}$ of a given scenario $\varphi \in \phi$ is presented below.

$$(RPRP^{(\varphi)}) \quad \min \sum_{\tau\in\mathscr{T}} \Big(\bar{F}^{\varphi(v)}_{\tau} y_\tau + up_\tau + \sum_{i\in\mathscr{N}} \sum_{j\in\alpha_i} \sum_{\kappa\in\mathscr{K}} c_{ij}\delta_{ij\kappa\tau} + \sum_{i\in\mathscr{N}} h_i I_{i\tau} + \sum_{i\in\mathscr{N}_c} \beta_i o_{i\tau}\Big) \tag{1.43}$$

s.t.

$$p_\tau \leq \mathscr{M}_\tau y_\tau \qquad \forall\tau\in\mathscr{T} \tag{1.44}$$

$$I_{0\tau} = I_{0,\tau-1} + p_\tau - \sum_{i\in\mathscr{N}_c}\sum_{\kappa\in\mathscr{K}} q_{i\kappa\tau} \qquad \forall\tau\in\mathscr{T} \tag{1.45}$$

$$I_{i\tau} = I_{i,\tau-1} + \sum_{\kappa\in\mathscr{K}} q_{i\kappa\tau} + o_{i\tau} - d^{\varphi}_{i\tau} \qquad \forall i\in\mathscr{N}_c, \tau\in\mathscr{T} \tag{1.46}$$

$$I_{0\tau} \leq L_0 \qquad \forall\tau\in\mathscr{T} \tag{1.47}$$

$$I_{i\tau} + d_{i\tau} \leq L_i \qquad \forall i\in\mathscr{N}_c, \tau\in\mathscr{T} \tag{1.48}$$

$$\sum_{i\in\mathscr{N}_c} q_{i\kappa\tau} \leq \mathscr{Q} z_{0\kappa\tau} \qquad \forall\kappa\in\mathscr{K}, \tau\in\mathscr{T} \tag{1.49}$$

$$\sum_{\kappa\in\mathscr{K}} z_{i\kappa\tau} \leq 1 \qquad \forall i\in\mathscr{N}_c, \tau\in\mathscr{T} \tag{1.50}$$

$$q_{i\kappa\tau} \leq \mathscr{W}^{\varphi}_{i\tau} z_{i\kappa\tau} \qquad \forall i\in\mathscr{N}_c, \kappa\in\mathscr{K}, \tau\in\mathscr{T} \tag{1.51}$$

$$\sum_{j\in\alpha_i} \delta_{ij\kappa\tau} = z_{i\kappa\tau} \qquad \forall i\in\mathscr{N}, \kappa\in\mathscr{K}, \tau\in\mathscr{T} \tag{1.52}$$

$$\sum_{j\in\sigma_i} \delta_{ji\kappa\tau} = z_{i\kappa\tau} \qquad \forall i\in\mathscr{N}, \kappa\in\mathscr{K}, \tau\in\mathscr{T} \tag{1.53}$$

$$y_\tau \in \{0,1\} \qquad \forall\tau\in\mathscr{T} \tag{1.54}$$

$$p_\tau \geq 0 \qquad \forall\tau\in\mathscr{T} \tag{1.55}$$

$$q_{i\kappa\tau} \geq 0 \qquad \forall i\in\mathscr{N}, \kappa\in\mathscr{K}, \tau\in\mathscr{T} \tag{1.56}$$

$$I_{i\tau} \geq 0 \qquad \forall i\in\mathscr{N}, \tau\in\mathscr{T} \tag{1.57}$$

$$o_{i\tau} \geq 0 \qquad \forall i\in\mathscr{N}, \tau\in\mathscr{T} \tag{1.58}$$

$$z_{i\kappa\tau} \in \{0,1\} \qquad \forall i\in\mathscr{N}, \kappa\in\mathscr{K}, \tau\in\mathscr{T} \tag{1.59}$$

$$0 \leq \delta_{ij\kappa\tau} \leq 1 \qquad \forall i\in\mathscr{N}, j\in\alpha_i, \kappa\in\mathscr{K}, \tau\in\mathscr{T}. \tag{1.60}$$

The objective function (1.43) is similar to the deterministic PRP except that if a vehicle is in node i the next visited node must be chosen from the set α_i. In addition, we remove the SECs (1.11) since they are no longer required. To impose the sequence sets on routing decisions, we replace constraints (1.10) with sets of constraints (1.52) and (1.53).

The same RPRP$^{(\varphi)}$ can be used for the static-static case where the only difference is that constraints (1.44) are replaced with the following constraints:

$$p_\tau \leq \mathscr{M}_\tau^{(max)} y_\tau \qquad \forall \tau \in \mathscr{T}. \quad (1.61)$$

Within the PH-based matheuristic, after solving the subproblems for all scenarios, the algorithm moves to the next iteration and the corresponding costs of first-stage variables are updated in order to lead the PH algorithm to a consensus over all the first-stage variables. If a stopping criterion is met, we go to the third phase of the algorithm. The goal of this phase is to improve routing decisions for a specific period within a scenario.

In the static-static strategy, the production quantities are also scenario-independent while the result of the RPRPs provides scenario-specific values for these variables. Thus, another step is required to obtain a consensus for the production quantities as well. Since the p_τ variables are aligned with the setup decisions, the setup decisions obtained from the PH algorithm are used to find the production quantities. In addition, we fix the routing and visit decisions as they will be improved further in the next phase. Thus, the resulting model is a linear programming (LP) problem that can be solved to find the production quantities.

The Third Phase: Route improvement

The second-phase problem yields feasible solutions to the subproblems, ensuring that the final iteration of the PH algorithm provides a feasible solution to the SPRP-AR. However, relying on predefined sets in the RPRPs may lead to non-optimal routes during certain periods. In order to enhance routing decisions, we address the CVRP for all periods within each scenario, aiming to minimize routing costs. In the CVRP formulation, we consider $q^{\varphi}_{i\kappa\tau}$ from the preceding phase as the demand of node $i \in \mathscr{N}_c$, during period $\tau \in \mathscr{T}$, and under scenario $\varphi \in \phi$. It is important to note that the CVRP is solved for a specific (τ, φ) pair only if there exists at least one $z^{\varphi}_{i\kappa\tau}$ equal to one. We mathematically model this problem as an MIP and employ a BC algorithm strengthened by valid inequalities for its solution. Further details on this algorithm can be found in Section 1.4.4.

1.4.4 Branch-and-Cut algorithm

To solve the SPRP-AR, we also propose a BC algorithm to compare its results with the ones from the PH-based matheuristic. Several sets of valid inequalities are employed to strengthen the problem's formulation and enhance the LP bounds.

Symmetry breaking inequalities

When there is a homogeneous fleet of vehicles, considering the vehicle index leads to redundant enumerations. Assuming a fleet of K vehicles, only a subset of n^{φ}_{τ} ($n^{\varphi}_{\tau} < K$) may be dispatched in period $\tau \in \mathscr{T}$ under scenario $\varphi \in \phi$ out of the $\binom{K}{n^{\varphi}_{\tau}}$ options with the same cost for selecting the required vehicles (Adulyasak et al., 2014b). Thus, by adding constraints (1.62), this symmetry issue can be prevented:

$$z^{\varphi}_{0\kappa\tau} \leq z^{\varphi}_{0,\kappa-1,\tau} \qquad 2 \leq \kappa \leq K, \forall \tau \in \mathscr{T}, \varphi \in \phi. \quad (1.62)$$

Other symmetry-breaking inequalities are the lexicographic constraints (1.63) that impose an order to assign customers to vehicles in each period (Adulyasak et al., 2014b; Jans, 2009):

$$\sum_{i=1}^{j} 2^{(j-i)} z^{\varphi}_{i\kappa\tau} \leq \sum_{i=1}^{j} 2^{(j-i)} z^{\varphi}_{i,\kappa-1,\tau} \qquad \forall j \in \mathscr{N}_c, 2 \leq \kappa \leq K, \tau \in \mathscr{T}, \varphi \in \phi. \quad (1.63)$$

Logical inequalities

Logical inequalities can also be applied to reduce the solution space of the problem (Archetti et al., 2007; Coelho and Laporte, 2014). We add constraints (1.64) to the formulation to ensure that if edge $(i, j) \in \mathscr{E}$ is travelled by vehicle $\kappa \in \mathscr{K}$, then node $j \in \mathscr{N}_c$

is visited by the same vehicle:

$$x^{\varphi}_{ij\kappa\tau} \leq z^{\varphi}_{j\nu\tau} \qquad \forall i \in \mathcal{N}_c, j \in \mathcal{N}_c, \kappa \in \mathcal{K}, \tau \in \mathcal{T}, \varphi \in \phi. \quad (1.64)$$

Note that variable $x^{\varphi}_{ij\kappa\tau}$ can take 0, 1, or 2 when one of the incident nodes is the depot while it can only take values 0 and 1 otherwise. The last set of valid inequalities imposes that the vehicle $\kappa \in \mathcal{K}$ is able to visit customer $i \in \mathcal{N}_c$ only if it is dispatched from the depot:

$$z^{\varphi}_{i\kappa\tau} \leq z^{\varphi}_{0\kappa\tau} \qquad \forall i \in \mathcal{N}_c, \kappa \in \mathcal{K}, \tau \in \mathcal{T}, \varphi \in \phi. \quad (1.65)$$

The proposed valid inequalities are also used in solving the SPRP-FSR with the BC algorithm by dropping the scenario index from the variables. In addition, we utilize all valid inequalities except for constraints (1.64) in the RPRP problems to increase the efficiency of the matheuristic algorithm.

Upon the integration of valid inequalities into the formulation, we relax the SECs (1.11). Subsequently, we employ a minimum s-t cut problem as the exact separation algorithm to identify and incorporate any violated constraints. Particularly, for the execution of this separation algorithm, we use the Concorde library (Applegate et al., 2020). We should highlight that this identical separation algorithm is applied in both phases 1 and 3 of our algorithm, effectively addressing the SECs associated with the TSP and the CVRP.

1.5 Computational Experiments

We performed experiments on two different datasets derived from the deterministic instances presented in Archetti et al. (2011). Monte Carlo simulation is employed as a sampling approach in order to generate scenarios. The demand of a customer in each period is an independent random variable that follows a discrete uniform distribution with a range of $[\bar{d}_{it}(1-\varepsilon), \bar{d}_{it}(1+\varepsilon)]$, where $\bar{d}_{it}$ represents the nominal demand from the deterministic instances and ε is the uncertainty level, where $\varepsilon \in [0,1]$. Each scenario may occur with a probability of $\pi_\varphi = 1/S, \forall \varphi \in \phi$, so that all scenarios have the same probability. The outsourcing cost is calculated as $\beta_i = \lceil \hat{\alpha}(u + f/C + 2c_{0i}/Q) \rceil$ as in Adulyasak et al. (2015a), where $\hat{\alpha}$ is a predefined penalty coefficient.

On the first dataset, the experiments were only carried out with the static-dynamic strategy in order to compare the results of the SPRP-AR proposed in this paper and the SPRP-FSR presented in Adulyasak et al. (2015a). Specifically, this dataset contains two test sets: a small set called $\mathscr{P}2$ and a large set called $\mathscr{G}2$. Set $\mathscr{P}2$ consists of 5, 10, 15, and 20 customers with $T = 3$ and $K = 1$ and four instance types with distinct characteristics namely the standard setting, high unit production costs, high transportation costs, and no customer inventory costs. The larger set $\mathscr{G}2$ comprises only the standard setting and instances with up to 3 vehicles, $T = 3$ and 5 to 30 customers or $T = 6$ and up to 20 customers. Our analyses only contained cases with 100 scenarios since increasing the number of scenarios to 500 or 1000 did not provide significant improvements in the results, which is consistent with the findings of Adulyasak et al. (2015a), while they can drastically increase the computation time. All experiments in the first set have a penalty coefficient of $\hat{\alpha} = 5$ and an uncertainty level of $\varepsilon = 0.2$.

To evaluate the stochastic solutions, we report the expected value of perfect information (EVPI) and the value of the stochastic solution (VSS) relative to the objective function value of the stochastic problem. The EVPI (1.66) is the difference between the weighted sum of the optimal objective values of the wait-and-see (WS*) problems and the optimal objective value of the recourse problem (RP*). The VSS, on the other hand, is calculated by first solving the expected value (EV) problem based on the average demand. Once the first-stage variables are found, the corresponding second-stage problems must be solved for all scenarios considering the fixed first stage decisions obtained from the EV problem. The expected value of the EV solution (EEV*) can be calculated as the weighted sum of the optimal objective function value of the EEV* subproblems. The VSS is the difference between the EEV* and the RP*. One of the challenges of this study is that the second stage of the SPRP-AR is not a linear problem that can be solved quickly. Specifically, it is an MIP formulation that includes SECs in each scenario, thus significantly increasing the complexity of the problem. Therefore, we require new measurements to assess the quality of solutions.

Since we cannot solve the SPRP-AR to optimality, we obtain an upper bound (UB) to the stochastic problem (RP^{UB}). As a consequence, when using RP^{UB} instead of RP*, we obtain UBs to the actual EVPIs and lower bounds (LBs) to the actual VSSs. We applied a BC algorithm to solve the WS and EV subproblems with a computation time limit of four hours. However, in instances where these problems cannot be optimally solved within the allocated time, we utilize the average LB of WS subproblems (WS^{LB}) solved with the BC to establish a UB for the actual EVPI, denoted as EVPI^{UB}.

$$EVPI = RP^* - WS^* \leq RP^{UB} - WS^* \leq RP^{UB} - WS^{LB} = EVPI^{UB}. \tag{1.66}$$

In addition, we use the LB and the UB on the EEV* (EEV^{LB} and EEV^{UB}) to establish both an LB and a UB to VSS, which are denoted as VSS^{LB} (1.67) and VSS^{UB} (1.68), respectively. Given that our proposed algorithm does not directly yield an LB for the RP, we can use the LB obtained by solving the RP with the BC algorithm as the LB to the problem. Furthermore, the weighted sum of the LB of the WS subproblems also provides a valid LB to the RP. Thus, we determine RP^{LB} by selecting the maximum value between these two alternatives.

$$VSS^{LB} = EEV^{LB} - RP^{UB} \leq EEV^* - RP^{UB} \leq EEV^* - RP^* = VSS \tag{1.67}$$

$$VSS \leq EEV^{UB} - RP^* \leq EEV^{UB} - RP^{LB} = VSS^{UB}. \tag{1.68}$$

We introduce a new dataset called $\mathscr{M}$ in order to conduct experiments on both static-dynamic and static-static strategies. The motivation behind introducing this second dataset lies in the observation that setup decisions in the $\mathscr{P}2$ and $\mathscr{G}2$ sets often exhibit a degree of triviality. In such cases, optimal setup decisions can be readily obtained without factoring in stochastic elements. This is evident in the setup where solving the EV problem yields optimal values for first-stage variables, exhibiting that solving the more complex stochastic problem is redundant, as illustrated in Table 1.3.

The new dataset is created based on the same Archetti et al. (2011) benchmark set while it differs in some parameters to yield non-trivial setup decisions. The penalty coefficient $\hat{\alpha}$ is set to 50 in $\mathscr{M}$ as the penalty costs appear to be insufficient in the first dataset, resulting in a high level of unmet demand. The number of periods dramatically increases the complexity of the problem, but we do not generate instances with $T = 3$ because in practice setup decisions are often required to be made over a longer period of time. The demands in the deterministic dataset are constant over periods, which allows us to generate sets with longer planning horizons. In this regard, we constructed new sets considering the standard setting and with $T = 6$, $K \leq 2$, $N \leq 30$; $T = 6$, $K = 3$, $N \leq 20$; $T = 9$; $K = 1$, $N \leq 30$; and $T = 9$, $K = 2$, $N \leq 20$. The production capacity is set to $C = 3\sum_{i \in \mathscr{N}_c} \bar{d}_{it}$. In order to calculate the transportation costs, Euclidean distances are used for all instances, while in set $\mathscr{M}$ we multiply them by a transportation coefficient (tc), where $tc = 4$ in all experiments.

Computational experiments were performed on an Intel Xeon Gold 6148 2.4 GHz processor and 32 GB of RAM. The algorithms were coded in C and Python programming languages while CPLEX 22.1.0 was used to solve the mathematical formulations. In CPLEX, eight threads were assigned for the parallel processing and the optimality gap was set to 10^{-6} in all experiments.

1.5.1 Static-Dynamic Strategy

In this section, we report the experiments on the Static-Dynamic strategy. In Table 1.3 the average performance of the heuristic and the BC algorithms on the $\mathscr{P}2$ and $\mathscr{G}2$ sets is provided. The solution of the heuristic algorithm was given to the BC algorithm as a warm start, and the computation time was set to 4 hours. The reported CPU time for BC represents the combined computation time of the BC algorithm and the heuristic. Besides the limit on the computation time, in several cases the limit on the memory caused the BC to stop while the algorithm still had not reached its time limit. Hence, there are cases where the optimality gap is positive while the reported CPU time is less than four hours.

Table 1.3: The results of SPRP-AR on $\mathscr{P}2$ and $\mathscr{G}2$ sets under the Static-Dynamic Strategy

T	K	S	#Ins	PH-M					BC	
				#Best	CPU (secs)	VSS^{LB} (%)	VSS^{UB} (%)	EVPI^{UB} (%)	CPU (secs)	Gap (%)
3	1	100	16	16	7.7	1.3	1.5	0.5	4,084.0	0.3
3	1	100	6	6	11.0	1.3	1.5	0.4	4,639.7	0.2
3	2	100	6	6	67.0	1.4	1.8	0.4	7,962.1	0.4
3	3	100	6	6	492.4	1.7	2.0	0.5	10,096.2	0.5
6	1	100	4	4	42.5	0.0	0.0	0.0	9,264.6	0.0
6	2	100	4	4	291.9	0.0	0.2	0.2	13,675.8	0.2
6	3	100	4	4	1,362.2	0.0	1.0	0.7	15,490.3	0.7
Total			46	46	224.6	1.0	1.3	0.4	9,316.1	0.3

Each row of the table reports the average results of a specific setting. The first three columns display the number of periods, the number of vehicles, and the number of scenarios, respectively. Column **#Ins** is the number of instances in each set. Under the PH-based matheuristic (PH-M), column **#Best** denotes the number of instances for which the BC algorithm could not find a better solution than the one obtained by the heuristic algorithm. The time limit for the PH-M was set to 2 hours and the actual running time is stated in seconds in column **CPU** under the PH-M. The details of the computation time for each phase are provided in Appendix A.1. As stated in the previous section, we provide both an LB and a UB to VSS and a UB to EVPI which are in the VSS^{LB}, VSS^{UB} and EVPI^{UB} columns, respectively. Column **Gap(%)** reports the relative optimality gap. The maximum number of iterations for the PH algorithm was set to 50 and the maximum number of non-improving iterations was set to 10. Moreover, θ_L, θ_H, γ_N, and γ_F were set to 0.4, 0.6, 0.2, and 0.8, respectively.

To showcase the advantages of adaptive routing over fixed routes and demonstrate the effectiveness of our proposed heuristic algorithm, we employed the BC algorithm to solve the SPRP-FSR. The optimality gap and solution time for this algorithm are detailed in Table 1.4, with a time limit set to 4 hours. Additionally, the inequalities presented in Section 1.4.4 were incorporated to enhance the LP bounds. To further optimize the solutions obtained for the SPRP-FSR, a post-optimization approach was implemented. This approach involved skipping customers that were visited in scenarios with no delivery quantity, thereby reducing transportation costs. Specifically, after solving the SPRP-FSR using the BC algorithm, any customer i with $z_{i\kappa\tau} = 1$ and $q^{\varphi}_{i\kappa\tau} = 0$ was removed from the corresponding route in that scenario. Subsequently, the visited nodes for each scenario were updated, and a TSP was solved for the revised route. The routing costs of the SPRP-FSR were then replaced with the updated results. In the **PostOpt** column, we show the improvement obtained from this procedure as a percentage compared to the value of the objective function of the SPRP-FSR.

Column **ImpLB(%)** demonstrates the LB of the cost improvement in percentage by using adaptive routing instead of first-stage routing. This is obtained by comparing the LB of the FSR problem and the result of the PH-M. We also report the UB of the possible cost improvement in column **ImpUB(%)** using the LB of the SPRP-AR and the UB of the FSR problem. In other words, **ImpLB(%)** and **ImpUB(%)** indicate the LB and the UB of the possible cost improvement by employing adaptive routing rather than fixed routes. It should be noted that value of zero was assigned to VSS^{LB} and ImpLB in all cases where these values were negative due to the fact that zero is always a valid LB for both. We should highlight that the average values for the VSS that are shown in Table 1.3 usually result from a few high VSS values while most of the instances have a VSS equal to 0. Thus, if the optimal setup decisions can be achieved by solving the EV problem, the result of the EEV problems can lead us to the optimal solution for the stochastic problem. It should be noted that, despite the fact that having VSS = 0 when solving a stochastic problem is not desirable, we can still achieve valuable improvements with adaptive routing over FSR even after applying the post-optimization technique (Table 1.4). We observe that for these data sets, the LB on the improvement is 0.7%. In addition, in this dataset, we still have cases where the VSS is large enough so that it is reasonable to consider the stochastic problem rather than the mean value problem.

As mentioned earlier, we created the set $\mathcal{M}$ to compare the results where solving the stochastic problem has a more tangible impact on the outcome. Table 1.5 shows the result of SPRP-AR under the static-dynamic strategy. As expected the CPU time of both the PH-M and the BC algorithm significantly increases when more scenarios are considered.

Table 1.4: Comparing the results of the SPRP-AR and the SPRP-FSR on $\mathscr{P}2$ and $\mathscr{G}2$ sets under the Static-Dynamic Strategy

T	K	S	#Ins	SPRP-AR (PH-M) Gap (%)	CPU (secs)	SPRP-FSR (BC) Gap (%)	CPU (secs)	PostOpt (%)	ImpLB (%)	ImpUB (%)
3	1	100	16	0.3	7.7	0.0	7.1	0.1	0.6	0.8
3	1	100	6	0.2	11.0	0.0	49.6	0.1	0.7	0.9
3	2	100	6	0.4	67.0	0.0	623.1	0.2	0.8	1.2
3	3	100	6	0.5	492.2	0.0	2,095.8	0.4	1.4	1.9
6	1	100	4	0.0	42.5	0.0	322.8	0.0	0.7	0.7
6	2	100	4	0.2	291.9	0.0	2,353.2	0.1	0.8	1.0
6	3	100	4	0.7	1,362.2	3.9	11,246.5	0.8	0.5	4.3
Total			46	0.3	224.6	0.3	1,574.3	0.2	0.7	1.3

In addition, for 200 scenarios, the BC algorithm was able to find a better solution for one of the instances. This results from the second stage of the RPRPs being unable to reach their optimality within the algorithm's time limit. In these experiments, VSS^{LB}, VSS^{UB}, and EVPI^{UB} remained almost the same regardless of the number of scenarios. When $S = 50$, the BC algorithm was able to find better LBs than the WS subproblems, which resulted in a lower optimality gap compared to cases with more scenarios. Table 1.6 provides the comparison of the results with the SPRP-FSR. ImpLB decreases with an increase in scenarios, as uncertainty has a greater impact on the FSR problem since routing decisions must be made in the first stage and reducing the number of scenarios may have a significant impact on the FSR. However, the AR problem does not seem to be affected as much by the number of scenarios. The uncertainty level ε was set to 0.2 and the problem was solved with 50, 100, and 200 scenarios. It is worth mentioning that in the heuristic algorithm, the 2-hour time limit was split into one hour for solving the phase two subproblems and one hour for the phase three subproblems, as phase 1 only takes a few seconds to solve.

Figure 1.2a presents the average value of the objective function of the SPRP-AR model under different numbers of scenarios. One can observe that the objective function values for the AR problem remain relatively consistent across all scenarios. However, the objective function values for the FSR problem and FSR(PO) (FSR after post-

Table 1.5: The results of the SPRP-AR for different number of scenarios on set $\mathscr{M}$ under the Static-Dynamic Strategy

T	K	S	#Ins	PH-M #Best	PH-M CPU (secs)	PH-M VSS^{LB} (%)	PH-M VSS^{UB} (%)	PH-M $EVPI^{UB}$ (%)	BC CPU (secs)	Gap (%)
6	1	50	6	6	59.2	3.4	4.2	0.8	7,581.2	0.8
6	2	50	6	6	891.5	3.9	6.0	1.6	13,088.6	1.6
6	3	50	4	4	904.8	4.0	9.3	3.2	13,530.1	3.2
9	1	50	6	6	283.6	2.2	3.1	0.9	10,618.7	0.9
9	2	50	4	4	1,269.5	2.0	7.7	3.6	13,606.3	3.6
Total			26	26	619.4	3.1	5.7	1.8	11,395.3	1.8
6	1	100	6	6	114.7	3.3	4.1	0.8	12,405.5	0.8
6	2	100	6	6	973.5	3.2	5.9	2.5	13,234.9	2.5
6	3	100	4	4	1,212.7	3.5	9.0	3.5	12,997.5	3.5
9	1	100	6	6	644.0	2.0	3.1	1.1	11,665.6	1.1
9	2	100	4	4	1,694.2	2.1	7.8	4.0	15,211.7	4.0
Total			26	26	847.0	2.8	5.6	2.2	12,948.9	2.2
6	1	200	6	6	289.1	3.3	4.1	0.8	12,701.7	0.8
6	2	200	6	6	2,505.8	3.2	5.8	2.2	16,145.1	2.2
6	3	200	4	4	1,576.9	3.7	9.1	4.2	13,416.7	4.2
9	1	200	6	5[1]	2,036.6	1.8	2.9	1.3	16,439.3	1.3
9	2	200	4	4	1,903.9	1.5	7.5	5.3	16,307.6	5.3
Total			26	25	1,650.5	2.7	5.5	2.4	15,023.6	2.4

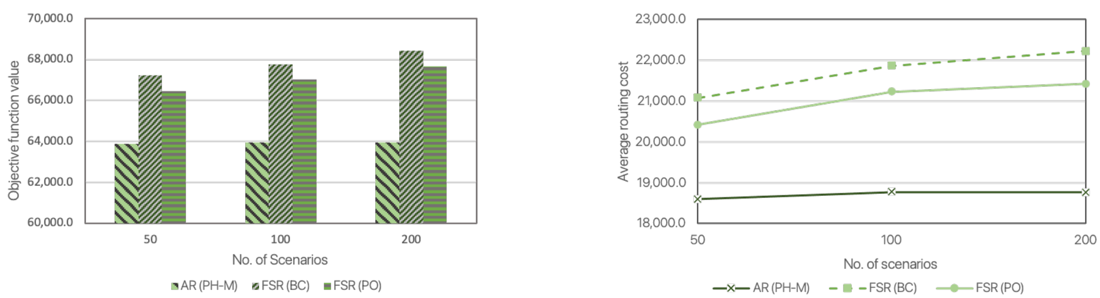
[1] The Branch-and-Cut UB has improved the optimality gap by 0.1%

optimization) are consistently higher than those of the AR. Notably, for larger problem instances, the BC algorithm often failed to find high-quality feasible solutions for SPRP-FSR, resulting in instances where the only viable solution was to outsource all demands. Consequently, such cases exhibited significantly larger objective function values and optimality gaps. To ensure a more realistic comparison, instances where the optimality gap for the FSR exceeded 20% in all cases were disregarded in cost comparisons. Otherwise, the cost of FSR would be considerably higher than that of AR. In Figure 1.2b, the routing costs of AR, FSR, and FSR(PO) are depicted.

The results of the SPRP-AR with different uncertainty levels including $\varepsilon = 0.4, 0.6, 0.8$ with 100 scenarios are shown in Table 1.7. The BC algorithm was not able to find a better solution than the one obtained by PH-M in 151 of the 156 cases, keeping in mind that the BC algorithm was started with the PH-M solution as a starting point. This denotes the de-

Table 1.6: Comparing the results of the SPRP-AR and the SPRP-FSR for different number of scenarios on set $\mathscr{M}$ under the Static-Dynamic Strategy

T	K	S	#Ins	SPRP-AR (PH-M) Gap (%)	SPRP-AR (PH-M) CPU (secs)	SPRP-FSR (BC) Gap (%)	SPRP-FSR (BC) CPU (secs)	SPRP-FSR (BC) PostOpt (%)	ImpLB (%)	ImpUB (%)
6	1	50	6	0.8	59.2	0.0	303.5	0.6	3.9	4.7
6	2	50	6	1.6	891.5	4.3	10,049.5	2.0	2.5	8.3
6	3	50	4	3.2	904.8	8.0	12,132.2	4.1	2.0	11.6
9	1	50	6	0.9	283.6	1.4	9,556.4	0.9	2.5	4.8
9	2	50	4	3.6	1,269.5	13.7	14,403.5	2.8	0.7	15.1
Total			26	1.8	619.4	4.7	8,676.9	1.9	2.7	8.2
6	1	100	6	0.8	114.7	0.0	1,413.7	0.7	4.0	4.8
6	2	100	6	2.5	973.5	9.5	11,699.5	2.2	2.2	12.7
6	3	100	4	3.5	1,212.7	12.0	14,416.2	2.6	1.8	14.4
9	1	100	6	1.1	644.0	3.9	10,349.7	1.0	2.0	7.0
9	2	100	4	4.0	1,694.2	20.2	14,402.3	1.3	0.5	20.4
Total			26	2.2	847.0	8.1	9,848.1	1.5	2.4	11.0
6	1	200	6	0.8	289.1	0.2	6,153.1	0.9	4.0	5.0
6	2	200	6	2.2	2,505.8	39.3	12,893.1	2.4	1.7	40.5
6	3	200	4	4.2	1,576.9	51.7	14,401.8	4.9	1.8	53.4
9	1	200	6	1.3	2,036.6	18.5	12,389.3	0.9	1.8	20.8
9	2	200	4	5.3	1,903.9	55.3	14,401.0	0.8	0.2	55.0
Total			26	2.4	1,650.5	29.8	11,685.6	1.4	2.1	32.0



(a) Objective function value

(b) Average routing costs

Figure 1.2: Comparison of costs for different number of scenarios on set $\mathscr{M}$

ficiency of the exact algorithms in solving such a complicated problem. For the instances in which the BC was able to obtain a better solution, the average gap improved by 0.3%, which is indicative of the efficiency of the PH-M algorithm. The average optimality gap of PH-M for all instances of set $\mathscr{M}$ under the static-dynamic strategy with different levels of uncertainty was 3%. Additionally, we conducted experiments on larger instances

Table 1.7: Summary of the SPRP-AR results for different uncertainty levels on set $\mathscr{M}$ under the Static-Dynamic Strategy

T	K	ε	#Ins	PH-M					BC	
				#Best	CPU (secs)	VSS^{LB} (%)	VSS^{UB} (%)	EVPI^{UB} (%)	CPU (secs)	Gap (%)
6	1	0.4	6	6	119.2	10.3	11.0	0.8	12,000.3	0.8
6	2	0.4	6	6	1,139.7	11.3	13.2	1.7	12,426.7	1.7
6	3	0.4	4	4	1,241.8	12.0	16.7	3.0	13,015.7	3.0
9	1	0.4	6	6	796.5	6.7	7.7	1.0	14,703.7	1.0
9	2	0.4	4	4	1,971.6	5.5	12.5	4.0	16,041.0	4.0
Total			26	26	968.7	9.2	11.9	1.9	13,500.4	1.9
6	1	0.6	6	5[1]	121.3	16.8	18.4	2.0	13,084.8	1.9
6	2	0.6	6	6	1,013.7	18.4	21.3	3.1	12,476.5	3.1
6	3	0.6	4	4	1,257.5	19.3	25.1	5.2	10,725.5	5.2
9	1	0.6	6	5[2]	753.9	11.6	13.2	2.2	14,600.0	1.6
9	2	0.6	4	4	1,985.5	10.1	17.6	5.4	16,388.7	5.4
Total			26	24	934.8	15.3	18.8	3.3	13,439.4	3.3
6	1	0.8	6	5[3]	162.9	21.6	24.0	3.5	11,813.1	3.4
6	2	0.8	6	6	1,102.1	23.5	27.5	4.6	12,084.9	4.6
6	3	0.8	4	4	1,403.4	24.8	31.5	7.1	13,345.0	7.1
9	1	0.8	6	5[4]	841.2	15.3	17.7	2.9	12,105.3	2.9
9	2	0.8	4	4	2,359.0	14.0	21.8	6.4	16,762.1	6.4
Total			26	24	1,064.9	19.9	24.2	4.6	12,940.3	4.6

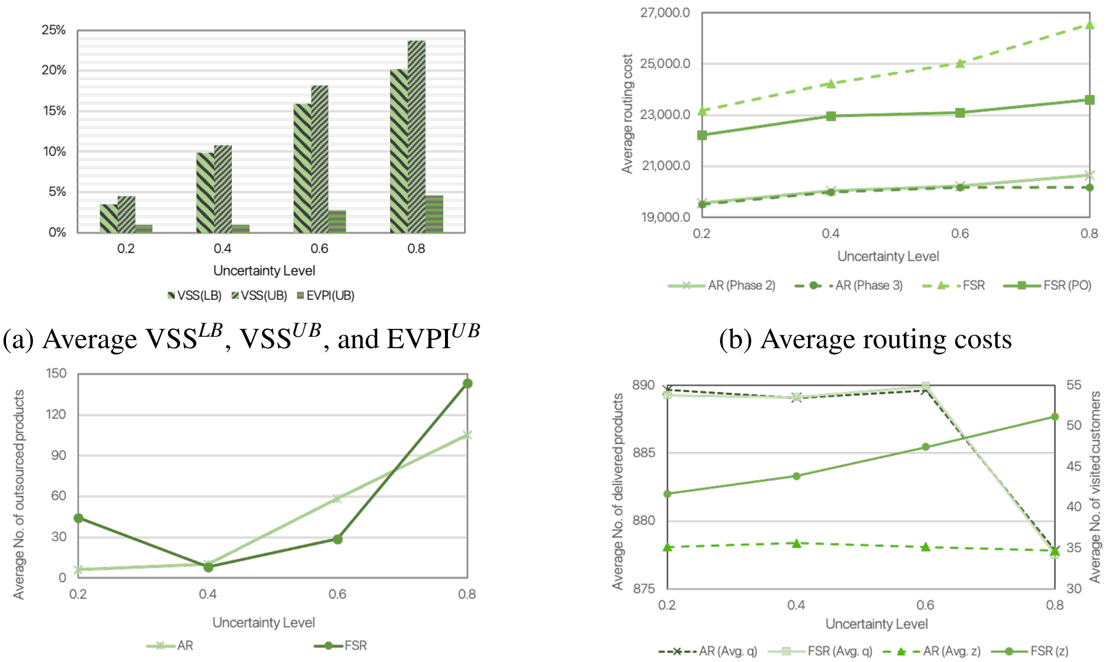
[1] The Branch-and-Cut UB has improved the optimality gap by 0.05%; [2] The Branch-and-Cut UB has improved the optimality gap by 0.6%; [3] The Branch-and-Cut UB has improved the optimality gap by 0.5%; [4] The Branch-and-Cut UB has improved the optimality gap by 0.3%.

involving up to 50 customers for six periods with one or two vehicles, as well as nine periods with one vehicle. The purpose of these experiments was to illustrate the performance of the algorithms. However, due to computational limitations, we only solved the problem using the PH-M and BC algorithms. The results are provided in Appendix A.2.

We summarize the results of the AR and the FSR problems with different uncertainty levels, solved with the PH-M and BC algorithms, in Table 1.8. As can be seen from the table, the optimality gap for the BC algorithm remained relatively constant for various levels of uncertainty. In contrast, the PH-M gap increased for higher uncertainty levels. There are two reasons for this. First, since LBs are derived from WS subproblems, increasing uncertainty naturally increases the EVPI, resulting in worse LBs; second, using the PH algorithm we decompose the stochastic problem into subproblems each representing a specific scenario. The presence of greater stochasticity makes it more challenging to find consensus and aggregate the first-stage variables. However, it is important to keep in mind that PH-M still has a lower optimality gap while taking less computational resources. When observing the improvement of the objective function value from the **ImpLB** column, it is important to remember that this is the minimum possible saving of using the AR, whereas the actual savings could be greater, as in some cases the optimality gap for the BC algorithm was so high that for those cases we had to consider a 0% improvement on the LB. Even so, with 100 scenarios the LB of the average improvement is as low as 2.4% with $\varepsilon = 0.2$ and goes up to 6.4% when the $\varepsilon = 0.8$. There is a substantial improvement in the objective function which is primarily attributable to routing costs. Based on a comparison between the objective function of the AR and FSR(PO), we find that considering flexible routes results in better solutions, while at the same time providing a more reasonable solution time for the PH-M. The AR becomes even more relevant when routing costs are high or when the environmental impact of transportation is taken into consideration.

One may observe that in several instances, the improvement obtained by applying post-optimization on the FSR problem is more than ImpLB. We should note that these two values are not directly comparable unless the optimality gap of the FSR is zero. It is due to the fact that the former is obtained by improving the UB of the FSR while the latter represents the relative difference between the AR UB and the FSR LB. Thus, a weak LB for the FSR can result in a low value of ImpLB which is the LB of the potential improvements. However, by looking at ImpUB we can see in all such cases the UB of the potential improvement is also so high which means that we can expect to have higher improvements by employing adaptive routing in these cases, as well.

We provide Figure 1.3 to better illustrate the behavior of the solutions while having different values of uncertainty level. Figure 1.3a indicates the average values of VSS^{LB}, VSS^{UB}, and EVPI^{UB} for different uncertainty levels. We can observe the rapid increase



(a) Average VSS^{LB}, VSS^{UB}, and EVPI^{UB}

(b) Average routing costs

(c) Average outsourced products

(d) Average visits and delivered products

Figure 1.3: An analysis of the behavior of solutions under the static-dynamic strategy with different uncertainty levels

of VSS for higher levels of uncertainty. The VSS^{LB} goes from 3.5% when $\varepsilon = 0.2$ to 20.2% when $\varepsilon = 0.8$. It is evident, therefore, that when the level of uncertainty is high, it is crucial to consider the stochastic problem instead of the mean value problem. As shown by the low values of EVPI^{UB}, more accurate information would not be of significant value, demonstrating the robustness of the solutions.

Figure 1.3b presents the routing cost of both the AR and the FSR problems. Since the heuristic PH-M algorithm includes two stages of routing decisions, we present the average routing cost after phases two and three. To assess the cost improvement that is achieved by utilizing flexible routes, we report the routing costs of the FSR before and after the post-optimization process. The third phase of the PH-M usually leads to a slight improvement in all cases, which is beneficial as this phase takes only a small portion of the solution time and often enhances the result. Here, one of the most significant observations is the difference between the AR and the FSR routing costs, which emphasizes the importance of considering adaptive routing. As can be seen from Table 1.7, this im-

Table 1.8: Comparing the results of the SPRP-AR and the SPRP-FSR for different uncertainty levels on set $\mathscr{M}$ under the Static-Dynamic Strategy

T	K	ε	#Ins	SPRP-AR (PH-M)		SPRP-FSR (BC)				
				Gap (%)	CPU (secs)	Gap (%)	CPU (secs)	PostOpt (%)	ImpLB (%)	ImpUB (%)
6	1	0.4	6	0.8	119.2	0.2	3,826.7	1.0	4.3	5.2
6	2	0.4	6	1.7	1,139.7	6.0	10,555.7	1.7	2.5	9.4
6	3	0.4	4	3.0	1,241.8	12.5	14,404.4	3.6	2.2	15.9
9	1	0.4	6	1.0	796.5	2.8	9,957.5	1.3	2.7	6.4
9	2	0.4	4	4.0	1,971.6	13.3	14,402.3	1.7	1.0	14.4
Total			26	1.9	968.7	6.1	10,048.7	1.8	2.9	9.5
6	1	0.6	6	1.9	121.3	0.0	2,107.2	2.2	4.5	5.9
6	2	0.6	6	3.1	1,013.7	6.6	10,801.6	4.2	3.0	12.3
6	3	0.6	4	5.2	1,257.5	11.0	14,403.6	2.6	2.5	16.9
9	1	0.6	6	2.2	753.9	1.9	9,794.6	1.7	3.6	7.1
9	2	0.6	4	5.4	1,985.5	13.5	14,402.1	2.1	1.6	16.7
Total			26	3.2	934.8	5.7	9,670.9	2.5	3.5	11.0
6	1	0.8	6	3.4	162.9	0.1	3,205.9	2.5	7.6	9.9
6	2	0.8	6	4.6	1,102.1	7.5	11,446.0	4.7	7.1	17.8
6	3	0.8	4	7.1	1,403.4	11.6	14,402.1	3.9	4.1	22.3
9	1	0.8	6	2.9	841.2	1.9	9,766.1	2.6	6.6	10.9
9	2	0.8	4	6.4	2,359.0	13.0	14,403.9	5.3	2.2	20.2
Total			26	4.4	1,064.9	6.0	10,066.6	3.6	6.4	15.5

provement is associated with a lower CPU time, as the average CPU time for SPRP-AR is approximately 15 minutes, whereas the average CPU time for SPRP-FSR is slightly less than three hours, with the heuristic algorithm yielding a reasonable gap.

In both AR and the FSR, the trend line for outsourced products indicate that the average number of outsourced products increases as the uncertainty level rises with one exception for FSR when $\varepsilon = 0.2$, as depicted in Figure 1.3c. This discrepancy is likely attributed to the higher optimality gap observed in FSR under this specific setting. Additionally, the analysis presented in Figure 1.3d indicates that the number of delivered products remains approximately equal for both problems, while the number of visits required by the AR problem is significantly lower. Consequently, an equivalent quantity of products can be delivered to customers with a reduced number of visits, showcasing the advantages of employing the AR problem. These observations were consistent across all

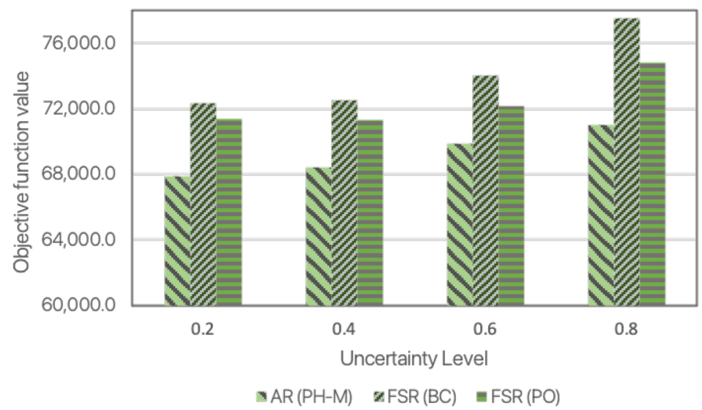


Figure 1.4: Objective function values for the AR (PH-M), FSR (BC), and FSR after post-optimization

cases considered, with a focus on instances where $S = 100$.

Figure 1.4 presents a comparison of the average value of the objective function for these two problems. The objective function values for AR solved with the PH-M, the UB of the FSR solved with BC, and the value of FSR after applying the post-optimization on the UB. This figure provides a comprehensive understanding of why ImpLB is not directly comparable to FSR(PO), as the objective function values of AR are consistently lower than those of FSR(PO) in all cases.

1.5.2 Static-static Strategy

The following section presents the findings of the SPRP-AR considering the “static-static” setting. This method limits the flexibility of the problem by requiring the determination of lot sizes at the beginning of the planning horizon. However, it can reduce the nervousness of the system due to the fact that both setups and production quantities are planned in advance (Tunc et al., 2013). It is particularly useful when respecting production capacity is crucial since deterministic variables for production quantities can ensure that these limitations are addressed throughout the planning horizon (Tempelmeier and Herpers, 2010).

In Table 1.9, we present the results of the SPRP-AR with the static-static case for dif-

Table 1.9: The results of the SPRP-AR for different number of scenarios on set $\mathscr{M}$ under the Static-Static Strategy

T	K	S	#Ins	PH-M					BC	
				Gap (%)	CPU (secs)	VSS^{LB} (%)	VSS^{UB} (%)	EVPI^{UB} (%)	CPU (secs)	Gap (%)
6	1	50	6	3.2	68.4	5.1	8.1	4.7	12,083.6	3.2
6	2	50	6	5.5	980.8	5.4	11.0	5.5	13,004.4	5.5
6	3	50	4	6.8	1,026.2	4.4	15.4	6.8	15,429.2	6.8
9	1	50	6	3.7	310.2	3.1	6.8	3.9	14,714.4	3.7
9	2	50	4	7.0	1,588.9	2.1	12.0	7.0	14,956.5	6.9
Total			26	5.0	716.0	4.6	10.2	5.4	13,859.9	5.0
6	1	100	6	3.9	140.0	4.4	8.2	5.0	14,543.7	3.9
6	2	100	6	6.4	1,234.4	4.4	10.5	6.5	15,638.1	6.4
6	3	100	4	7.4	1,554.3	3.7	14.6	7.4	15,960.6	7.4
9	1	100	6	4.7	873.1	2.5	7.0	4.9	15,276.1	4.6
9	2	100	4	7.9	1,717.0	2.1	12.3	7.9	14,228.5	7.9
Total			26	5.8	1,021.9	4.0	10.1	6.1	15,134.8	5.8
6	1	200	6	4.6	268.5	4.1	8.5	4.9	14,548.3	4.5
6	2	200	6	6.5	1,979.2	3.7	10.3	6.5	16,384.0	6.5
6	3	200	4	8.5	1,919.3	3.6	14.4	8.5	16,323.3	8.5
9	1	200	6	4.8	1,189.3	1.9	6.7	4.8	15,591.8	4.7
9	2	200	4	11.0	2,858.7	1.5	11.9	11.0	17,262.0	11.0
Total			26	6.7	1,528.2	3.4	9.9	6.8	15,903.3	6.6

ferent scenarios using similar settings as in the previous section. Note that the instances from the previous section have been used for comparative purposes. As expected, optimality gaps are higher for this strategy. However, this does not necessarily imply that algorithm's performance is inferior since finding a better LB becomes more difficult in this case. The reason is that considering production quantities in the first stage also increases the EVPI. In order to demonstrate the efficacy of the presented algorithm for the static-static strategy, we provide the optimality gaps for both PH-M and BC algorithms. It can be observed that, in some cases, the BC algorithm provided better solutions than the PH-M algorithm. Even so, these improvements are usually so small that the average total improvement is 0.1%. It should be emphasized once again that the BC algorithm was given the PH-M solution as a warm start and was still unable to improve it within the limits of the computation in most instances.

A significant observation in the static-static case is the higher VSS^{LB} values, which is again due to the presence of more first-stage variables. As a result, stochastic programming becomes even more important when the company is required to employ this strategy. Table 1.10 presents a comparison between the results of the first-stage and second-stage routing problems under the static-static strategy. Based on the results, the LB of the cost improvement is slightly worse than in the previous case. The BC algorithm for the FSR also has a larger optimality gap, which may be one of the factors contributing to the difference. The ImpUB is also significantly higher, which makes it difficult to draw any conclusive comparison between the two cases in terms of possible improvements. Even so, we can observe that adaptive routing reduces routing costs and the improvements are better than those obtained by post-optimization. A comparison of the results of the static-static strategy with different uncertainty levels can be found in Tables 1.11 and 1.12. As can be seen, the VSS values are significantly higher than the static-dynamic strategy.

Figure 1.5 displays the comparison of the static-static and the static-dynamic strategies. We can observe the value of the objective function for different numbers of scenarios in Figure 1.5a and for different uncertainty levels in Figure 1.5b. In both figures, the trend line of the static-static strategy is above the static-dynamic strategy, indicating higher objective function values for the former. A significant portion of the higer values can be attributed to higher holding costs since the model under the static-static strategy tends to produce more products in earlier periods to avoid outsourcing in later periods. This can be seen in Figure 1.5c. However, the number of outsourced products is also higher in the static-static strategy (Figure 1.5d), particularly at higher levels of uncertainty.

1.5.3 Other Probability Distributions

To further demonstrate the applicability of the proposed algorithm, we extended the experiments conducted in the previous section to include stochastic demand generated from the Normal and Gamma distribution functions. The same algorithm was applied, and a similar Monte Carlo Sampling technique was employed to generate scenarios. It is worth

Table 1.10: Comparing the results of the SPRP-AR and the SPRP-FSR for different number of scenarios on set $\mathscr{M}$ under the Static-Static Strategy

T	K	S	#Ins	SPRP-AR (PH-M) Gap (%)	SPRP-AR (PH-M) CPU (secs)	SPRP-FSR (BC) Gap (%)	SPRP-FSR (BC) CPU (secs)	SPRP-FSR (BC) PostOpt (%)	ImpLB (%)	ImpUB (%)
6	1	50	6	3.2	68.4	0.0	331.6	0.6	3.2	6.6
6	2	50	6	5.5	980.8	6.0	9,990.3	2.9	2.0	12.7
6	3	50	4	6.8	1,026.2	8.5	12,456.4	3.0	1.9	15.1
9	1	50	6	3.7	310.2	1.4	9,708.0	1.1	2.1	7.1
9	2	50	4	7.0	1,588.9	12.5	14,403.5	3.3	0.7	16.4
Total			26	5.0	716.0	4.9	8,754.6	2.0	2.3	11.0
6	1	100	6	3.9	140.0	0.0	2,059.7	0.8	3.3	7.3
6	2	100	6	6.4	1,234.4	8.1	10,784.2	2.6	1.9	14.0
6	3	100	4	7.4	1,554.3	10.6	14,402.2	2.2	1.4	15.8
9	1	100	6	4.7	873.1	2.3	10,139.7	1.2	1.1	8.1
9	2	100	4	7.9	1,717.0	35.1	14,403.4	2.5	0.3	37.5
Total			26	5.8	1,021.9	9.4	9,735.5	0.9	1.8	15.0
6	1	200	6	4.6	268.5	0.2	6,916.3	0.8	3.4	8.0
6	2	200	6	6.5	1,979.2	37.5	12,699.3	1.2	1.8	41.4
6	3	200	4	8.5	1,919.3	33.1	14,401.2	1.9	0.9	36.2
9	1	200	6	4.8	1,189.3	3.7	12,170.5	0.9	1.4	9.0
9	2	200	4	11.0	2,858.7	39.3	14,401.4	1.1	0.1	41.5
Total			26	6.7	1,528.2	19.9	11,766.4	1.8	1.8	24.8

noting that for consistency in the results, we utilized the same mean and standard deviation for all probability distribution functions (PDF). Specifically, we used the nominal demand as the mean value, and the standard deviation obtained from the Uniform distribution as the given standard deviation for both the Normal and Gamma distributions.

Furthermore, to handle the potential occurrence of negative values in the Normal distribution, we set the demand to zero in such cases. Moreover, to maintain integer demand values, we truncate the obtained numbers from the probability function. This approach ensures that the demand values remained integer throughout the analysis.

For these experiments, we employed identical instances featuring 100 scenarios and four distinct levels of demand uncertainty. The time limit for the PH-M algorithm remained consistent at two hours, and for all the BC algorithms, we set a four-hour time limit. The parameters of the heuristic algorithm remain unchanged, as previously detailed

Table 1.11: Summary of the SPRP-AR results for different uncertainty levels on set $\mathcal{M}$ under the Static-Static Strategy

T	K	ε	#Ins	PH-M					BC	
				Gap (%)	CPU (secs)	VSS^{LB} (%)	VSS^{UB} (%)	EVPI^{UB} (%)	CPU (secs)	Gap (%)
6	1	0.4	6	4.0	177.6	15.6	19.1	7.5	12,606.4	4.0
6	2	0.4	6	8.5	1,201.5	15.3	23.0	9.0	14,579.8	8.5
6	3	0.4	4	11.1	1,394.4	11.7	28.9	11.1	12,717.4	11.1
9	1	0.4	6	5.8	773.5	9.0	14.3	6.7	13,988.7	5.6
9	2	0.4	4	10.2	1,940.4	5.9	21.5	10.2	16,342.3	10.1
Total			26	7.5	1009.8	13.3	20.8	8.6	13,972.6	7.4
6	1	0.6	6	5.1	178.1	26.5	30.2	10.9	12,405.0	5.1
6	2	0.6	6	9.1	1,284.6	26.7	33.7	12.0	14,118.2	9.1
6	3	0.6	4	12.2	1,263.9	20.3	40.2	13.6	12,571.6	12.2
9	1	0.6	6	6.4	890.5	16.5	22.1	9.6	15,293.4	6.3
9	2	0.6	4	12.0	2,622.5	11.6	30.2	13.3	17,025.6	12.0
Total			26	8.5	1,140.9	23.5	30.7	11.6	14,203.4	8.4
6	1	0.8	6	6.3	191.5	31.6	36.0	14.6	11,763.9	6.3
6	2	0.8	6	10.9	1,246.5	31.0	38.9	16.6	13,456.8	10.9
6	3	0.8	4	12.6	1,328.5	23.5	44.3	17.4	12,520.0	12.6
9	1	0.8	6	6.6	854.0	20.3	25.8	12.3	15,257.1	6.5
9	2	0.8	4	11.0	2,565.6	14.2	32.3	15.3	16,327.3	10.9
Total			26	9.1	1,128.0	27.8	35.0	15.1	13,779.1	9.1

in Section 1.5.1. In Table 1.13, we present a summary of the results for the Normal and Gamma distributions. Notably, the average optimality gap for different uncertainty levels remains below 4% across all studied PDFs. However, this gap is higher for the Normal and Gamma distributions, primarily due to their higher EVPI^{UB}. The increased variability in demand values within the Normal and Gamma distributions explains this higher EVPI^{UB}.

Moreover, the average values of the VSS^{LB} underscore the importance of considering stochastic programming over mean-value problem formulations, particularly when demand uncertainty is high. Additionally, the total average value of ImpLB falls between 3% and 4% across all PDFs. It is worth noting that the proposed heuristic has an average CPU time ranging from 16 to 20 minutes, whereas solving the SPRP-FSR using the BC algorithm requires over 2.5 hours on average.

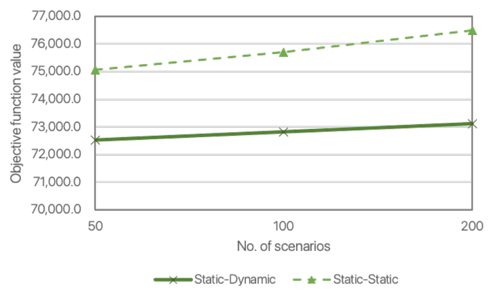
Table 1.12: Comparing the results of the SPRP-AR and the SPRP-FSR for different uncertainty levels on set $\mathscr{M}$ under the Static-Static Strategy

T	K	ε	#Ins	SPRP-AR (PH-M) Gap (%)	SPRP-AR (PH-M) CPU (secs)	SPRP-FSR (BC) Gap (%)	SPRP-FSR (BC) CPU (secs)	SPRP-FSR (BC) PostOpt (%)	ImpLB (%)	ImpUB (%)
6	1	0.4	6	4.0	177.6	0.4	5,125.9	1.0	3.8	8.0
6	2	0.4	6	8.5	1,201.5	6.2	11,030.9	1.6	2.3	14.5
6	3	0.4	4	11.1	1,394.4	13.3	14,401.9	3.3	1.9	21.6
9	1	0.4	6	5.8	773.5	2.6	9,946.0	0.6	1.7	10.1
9	2	0.4	4	10.2	1,940.4	13.1	14,402.6	2.2	0.8	18.9
Total			26	7.5	1,009.8	6.2	10,455.2	1.8	2.5	13.8
6	1	0.6	6	5.1	178.1	0.0	1,931.3	1.9	3.6	8.5
6	2	0.6	6	9.1	1,284.6	9.1	10,816.9	3.3	2.3	19.0
6	3	0.6	4	12.2	1,263.9	10.8	14,402.5	3.6	2.2	22.4
9	1	0.6	6	6.4	890.5	1.4	9,714.7	0.9	1.5	10.1
9	2	0.6	4	12.0	2,622.5	33.3	14,403.8	2.7	1.3	40.0
Total			26	8.5	1,140.9	9.2	9,615.5	2.5	2.8	18.3
6	1	0.8	6	6.3	191.5	0.0	4,484.9	2.5	6.3	11.8
6	2	0.8	6	10.9	1,246.5	7.2	11,526.7	4.6	4.5	20.8
6	3	0.8	4	12.6	1,328.5	12.0	14,403.2	6.6	3.2	26.4
9	1	0.8	6	6.6	854.0	2.3	9,324.2	2.4	5.7	13.9
9	2	0.8	4	11.0	2,565.6	13.7	14,402.7	3.5	2.0	24.8
Total			26	9.1	1,128.0	6.1	10,278.4	3.8	5.0	18.6

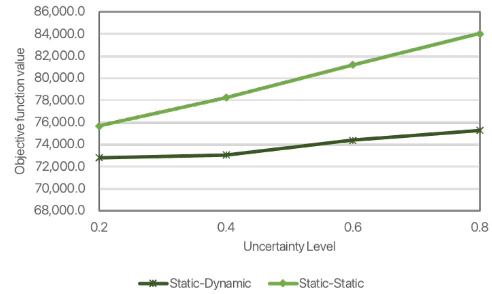
We present a comparative analysis of the average routing costs associated with three different routing algorithms: AR, FSR, and FSR(PO). The results are depicted in Figure 1.6, where instances exhibiting a gap of more than 20% for FSR across all cases have been excluded. Across all probability functions examined, we observe a clear trend of increased routing costs as the level of uncertainty rises. Notably, the costs are comparatively lower for scenarios following the Normal and Gamma distributions, particularly at lower uncertainty levels. This phenomenon can be attributed to the fact that the probability distribution in the Normal and Gamma cases is more concentrated around the mean value. Consequently, a larger proportion of scenarios fall within a narrow range around the mean, whereas in the Uniform distribution, the probability remains constant across the entire range. However, it is worth noting that the effect of this characteristic diminishes as ε increases, as evidenced in Figure 1.6a. Despite this observation, the routing cost of FSR(PO) consistently exceeds that of AR, providing evidence of the algorithm's

Table 1.13: Summary of the results of the Normal and Gamma distribution for different uncertainty levels on set $\mathscr{M}$ under the Static-Dynamic Strategy

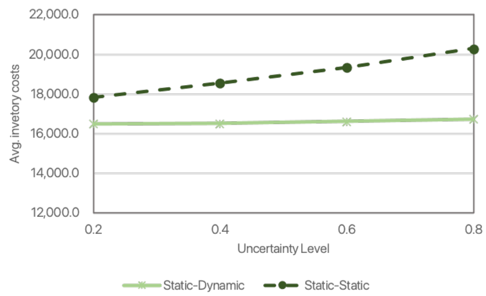
	ε	#Ins	SPRP-AR (PH-M) Gap (%)	CPU (secs)	VSS^{LB} (%)	VSS^{UB} (%)	$EVPI^{UB}$ (%)	SPRP-FSR (BC) Gap (%)	CPU (secs)	PostOpt (%)	ImpLB (%)	ImpUB (%)
Normal	0.2	26	3.2	1,143.2	0.4	3.7	3.2	9.9	9,800.2	1.5	1.3	12.4
	0.4	26	3.3	1,063.4	6.3	9.3	3.3	13.7	10,078.6	1.6	2.0	17.3
	0.6	26	3.0	1,149.4	11.6	15.0	3.0	14.9	9,776.4	2.5	3.7	20.1
	0.8	26	5.3	1,008.3	16.6	21.7	5.3	6.7	8,793.8	3.2	5.7	16.0
Total		104	3.7	1,091.1	8.8	12.4	3.7	11.3	9,612.3	2.2	3.2	16.4
Gamma	0.2	26	3.1	1,146.2	0.5	3.6	3.1	10.5	9,994.5	1.4	1.3	12.8
	0.4	26	4.6	1,174.3	5.9	9.3	4.6	9.9	10,065.0	2.1	1.5	13.7
	0.6	26	3.4	1,233.1	11.8	15.6	3.4	9.3	9,202.8	3.2	4.6	16.1
	0.8	26	4.7	1,073.8	17.9	22.4	4.7	5.9	9,398.8	4.2	8.5	18.3
Total		104	3.9	1,156.9	9.0	12.7	3.9	8.9	9,665.2	2.7	4.0	15.2



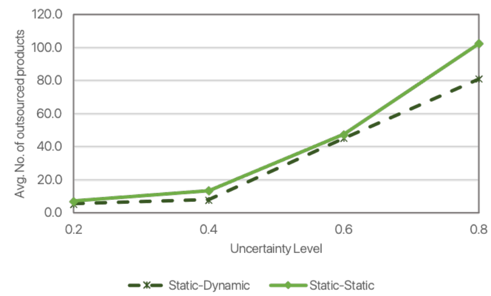
(a) OF value for different number of scenarios



(b) OF value for different Uncertainty Levels



(c) Average inventory costs



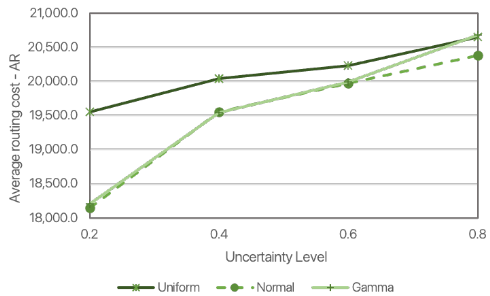
(d) Average outsourced products

Figure 1.5: The comparison of Static-Static and Static-Dynamic strategies

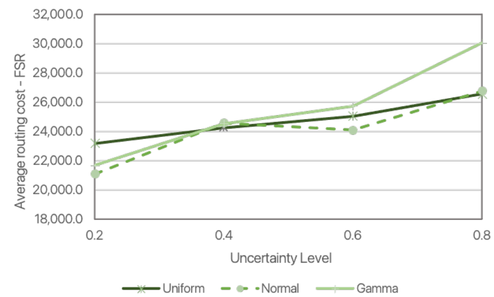
effectiveness across all instances.

Figure 1.7 presents a comparison of three performance metrics: VSS^{LB}, VSS^{UB}, and EVPI^{UB} for different probability functions. As anticipated, when $\varepsilon = 0.2$, scenarios generated by the Normal and Gamma distributions tend to cluster closely around the mean value. Consequently, the EV problem yields a high-quality first-stage solution, leading to lower VSS^{LB} values. However, as ε increases, the VSS^{LB} also increases. Figure 1.8 illustrates the objective function values for various probability distribution functions across different uncertainty levels. Regardless of the distribution, we consistently observe that the average objective function value is lower for AR, even after applying post-optimization to FSR. On average, the improvement achieved by utilizing AR instead of FSR amounts to 6.5%, 6.2%, and 7.8% for the Uniform, Normal, and Gamma distributions, respectively. If we consider the post-optimization on FSR, the improvement percentages become 4.3%, 4.3%, and 5.6% for the respective distributions.

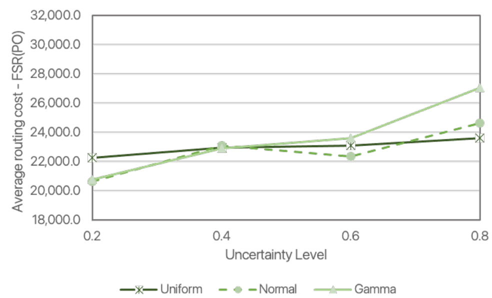
These results highlight the superior performance of AR over FSR in terms of achieving



(a) Avg. routing cost for AR



(b) Avg. routing cost for FSR

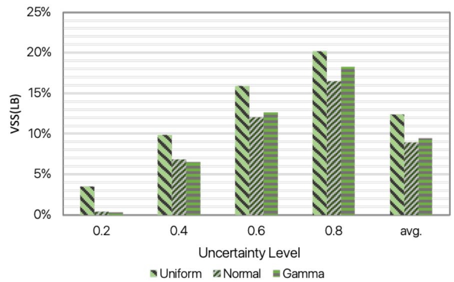


(c) Avg. routing cost for FSR(PO)

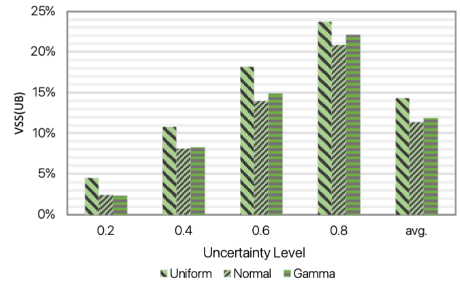
Figure 1.6: The average value of routing costs under the SD strategy with different ε for different PDFs

lower objective function values across all instances and also indicate the effectiveness and efficiency of the PH-M algorithm.

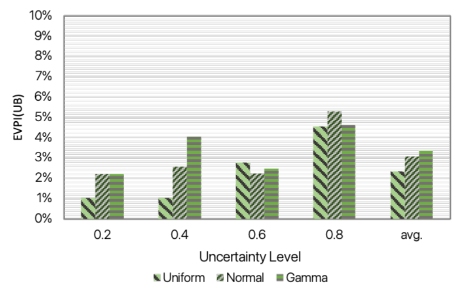
We also investigated three distinct factors to discern their contributions to the variations in routing costs across different levels of uncertainty and distribution functions for demand. These factors encompass the average number of routes (ANR), the average route length (ARL), and the average per-route number of visited customers (ANVC). A detailed table of the values of these factors is provided in Appendix A.3. Notably, our analysis revealed that while the ARL and ANVC show a lower level of fluctuation for different uncertainty levels, the ANR increases as the uncertainty levels go up. However, its value is roughly similar for different probability functions. This suggests that increasing the uncertainty level mostly results in a higher average number of routes while not having a significant impact on the average route length or average number of visited customers.



(a) Avg. VSS^{LB}

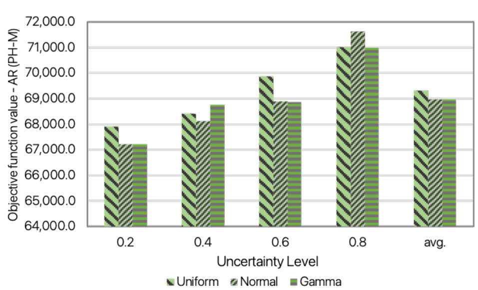


(b) Average VSS^{UB}

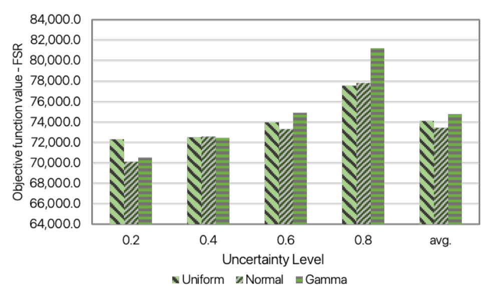


(c) Avg. EVPI^{UB}

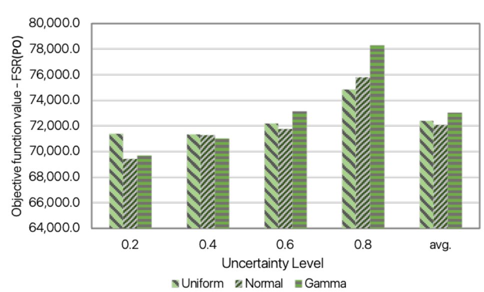
Figure 1.7: Comparison of relative VSS^{LB}, VSS^{UB}, and EVPI^{UB} under the SD strategy with different ε for different PDFs



(a) Avg. OF for AR



(b) Avg. OF for FSR



(c) Avg. OF for FSR(PO)

Figure 1.8: The average objective function values under the SD strategy with different ε for different PDFs

1.6 Conclusion

This study addresses a significant research gap in the field of the SPRP by focusing on adaptive second-stage routing decisions. By introducing this novel concept in the stochastic PRP, the study enhances the flexibility of the problem, resulting in reduced routing costs and improved overall cost efficiency for the system. This increased flexibility provides valuable managerial advantages by enabling quick responses to unexpected demands. To tackle this complex problem, the study proposes a progressive hedging-based matheuristic algorithm. This algorithm is designed to generate high-quality solutions within a reasonable timeframe. The effectiveness of this approach is demonstrated through extensive computational experiments on 566 instances. Additionally, the algorithm is extended to incorporate the static-static strategy, which further stabilizes the production schedule but reduces system flexibility.

Our study highlights the significant advantages practitioners can gain from adopting the stochastic problem over the deterministic one, resulting in average cost reductions ranging from 12.4% to 14.3%. Notably, the deterministic approach not only escalates costs but also introduces potential infeasibilities, jeopardizing customer satisfaction by failing to meet demand. Additionally, our research underscores the effectiveness of adaptive routing, with average savings of 6.5% compared to fixed routes. In environments characterized by high uncertainty, these savings can increase to 12.6%, offering substantial cost reductions for companies. Moreover, adaptive routing leads to a reduction in both the number of visited customers and routing costs, enhancing distribution efficiency and transportation logistics effectiveness. It is important to note that our study focused solely on single-product systems; however, adopting adaptive routing in multi-product scenarios could yield even greater benefits.

Our comparison of the SPRP with adaptive routing against fixed routes or fixed routes with post-optimization reveals a minimum improvement of 6.5%. This indicates that flexible routing not only reduces routing costs but also minimizes inventory and outsourcing expenses within the system. Furthermore, our proposed algorithm achieves these results efficiently, with an average CPU time of less than 20 minutes, contrasting with the lengthy processing time required by the BC algorithm. We also analyze the UB on the average expected value of perfect information (EVPI^{UB}) and the LB on the average minimum value of the stochastic solution (VSS^{LB}). The EVPI^{UB} remains relatively low, barely exceeding 5%, suggesting limited additional value from more information. Conversely, the VSS^{LB} can reach up to 12.4%, highlighting the critical importance of factoring uncertainty into the decision-making process.

References

Absi, N., Archetti, C., Dauzère-Pérès, S., and Feillet, D. (2015). "A Two-Phase Iterative Heuristic Approach for the Production Routing Problem". *Transportation Science* 49.4, pp. 784–795. DOI: 10.1287/trsc.2014.0523.

Adulyasak, Y., Cordeau, J.-F., and Jans, R. (2014a). "Formulations and Branch-and-Cut Algorithms for Multivehicle Production and Inventory Routing Problems". *INFORMS Journal on Computing* 26.1, pp. 103–120. DOI: 10.1287/ijoc.2013.0550.

Adulyasak, Y., Cordeau, J.-F., and Jans, R. (2014b). "Optimization-Based Adaptive Large Neighborhood Search for the Production Routing Problem". *Transportation Science* 48.1, pp. 20–45. DOI: 10.1287/trsc.1120.0443.

Adulyasak, Y., Cordeau, J.-F., and Jans, R. (2015a). "Benders Decomposition for Production Routing Under Demand Uncertainty". *Operations Research* 63.4, pp. 851–867. DOI: 10.1287/opre.2015.1401.

Adulyasak, Y., Cordeau, J.-F., and Jans, R. (2015b). "The production routing problem: A review of formulations and solution algorithms". *Computers & Operations Research* 55, pp. 141–152. DOI: 10.1016/j.cor.2014.01.011.

Agra, A., Requejo, C., and Rodrigues, F. (2018). "An adjustable sample average approximation algorithm for the stochastic production-inventory-routing problem". *Networks* 72.1, pp. 5–24. DOI: 10.1002/net.21796.

Alvarez, A., Cordeau, J.-F., Jans, R., Munari, P., and Morabito, R. (2021). "Inventory routing under stochastic supply and demand". *Omega* 102, p. 102304. DOI: 10.1016/j.omega.2020.102304.

Applegate, D., Bixby, R., Chvátal, V., and Cook, W. (2020). *Concorde TSP Solver*. URL: https://www.math.uwaterloo.ca/tsp/concorde.html.

Archetti, C., Bertazzi, L., Laporte, G., and Speranza, M. G. (2007). "A Branch-and-Cut Algorithm for a Vendor-Managed Inventory-Routing Problem". *Transportation Science* 41.3, pp. 382–391. DOI: 10.1287/trsc.1060.0188.

Archetti, C., Bertazzi, L., Paletta, G., and Speranza, M. G. (2011). "Analysis of the maximum level policy in a production-distribution system". *Computers & Operations Research* 38.12, pp. 1731–1746. DOI: 10.1016/j.cor.2011.03.002.

Armentano, V. A., Shiguemoto, A. L., and Løkketangen, A. (2011). "Tabu search with path relinking for an integrated production–distribution problem". *Computers & Operations Research* 38.8, pp. 1199–1209. DOI: 10.1016/j.cor.2010.10.026.

Avci, M. and Topaloglu Yildiz, S. (2020). "A mathematical programming-based heuristic for the production routing problem with transshipments". *Computers & Operations Research* 123, p. 105042. DOI: 10.1016/j.cor.2020.105042.

Bard, J. F. and Nananukul, N. (2009). "The integrated production–inventory–distribution–routing problem". *Journal of Scheduling* 12.3, pp. 257–280. DOI: 10.1007/s10951-008-0081-9.

Bard, J. F. and Nananukul, N. (2010). "A branch-and-price algorithm for an integrated production and inventory routing problem". *Computers & Operations Research* 37.12, pp. 2202–2217.

Ben Ahmed, M., Hvattum, L. M., and Agra, A. (2023). "The effect of different mathematical formulations on a matheuristic algorithm for the production routing problem". *Computers & Operations Research* 155, p. 106232. DOI: 10.1016/j.cor.2023.106232.

Bookbinder, J. H. and Tan, J.-Y. (1988). “Strategies for the Probabilistic Lot-Sizing Problem with Service-Level Constraints”. *Management Science* 34, pp. 1096–1108. DOI: 10.1287/mnsc.34.9.1096.

Boudia, M., Louly, M. A. O., and Prins, C. (2008). “Fast heuristics for a combined production planning and vehicle routing problem”. *Production Planning & Control* 19.2, pp. 85–96. DOI: 10.1080/09537280801893356.

Chandra, P. (1993). “A dynamic distribution model with warehouse and customer replenishment requirements”. *Journal of the Operational Research Society* 44.7. DOI: 10.1057/jors.1993.117.

Chitsaz, M., Cordeau, J.-F., and Jans, R. (2019). “A Unified Decomposition Matheuristic for Assembly, Production, and Inventory Routing”. *INFORMS Journal on Computing* 31.1, pp. 134–152. DOI: 10.1287/ijoc.2018.0817.

Chitsaz, M., Cordeau, J.-F., and Jans, R. (2020). “A branch-and-cut algorithm for an assembly routing problem”. *European Journal of Operational Research* 282.3, pp. 896–910. DOI: 10.1016/j.ejor.2019.10.007.

Coelho, L. C. and Laporte, G. (2014). “Improved solutions for inventory-routing problems through valid inequalities and input ordering”. *International Journal of Production Economics* 155, pp. 391–397. DOI: 10.1016/j.ijpe.2013.11.019.

Crainic, T. G., Fu, X., Gendreau, M., Rei, W., and Wallace, S. W. (2011). “Progressive hedging-based metaheuristics for stochastic network design”. *Networks* 58.2, pp. 114–124. DOI: 10.1002/net.20456.

Gruson, M., Cordeau, J.-F., and Jans, R. (2023). “Split demand and deliveries in an integrated three-level lot sizing and replenishment problem”. *Computers & Operations Research*, p. 106434. DOI: 10.1016/j.cor.2023.106434.

Haugen, K. K., Løkketangen, A., and Woodruff, D. L. (2001). “Progressive hedging as a meta-heuristic applied to stochastic lot-sizing”. *European Journal of Operational Research* 132.1, pp. 116–122. DOI: 10.1016/S0377-2217(00)00116-8.

Jans, R. (2009). "Solving Lot-Sizing Problems on Parallel Identical Machines Using Symmetry-Breaking Constraints". *INFORMS Journal on Computing* 21.1, pp. 123–136. DOI: 10.1287/ijoc.1080.0283.

Kumar, R. S., Kondapaneni, K., Dixit, V., Goswami, A., Thakur, L. S., and Tiwari, M. K. (2016). "Multi-objective modeling of production and pollution routing problem with time window: A self-learning particle swarm optimization approach". *Computers & Industrial Engineering* 99, pp. 29–40. DOI: 10.1016/j.cie.2015.07.003.

Li, Y., Chu, F., Chu, C., and Zhu, Z. (2019). "An efficient three-level heuristic for the large-scaled multi-product production routing problem with outsourcing". *European Journal of Operational Research* 272.3, pp. 914–927. DOI: 10.1016/j.ejor.2018.07.018.

Løkketangen, A. and Woodruff, D. L. (1996). "Progressive hedging and tabu search applied to mixed integer (0, 1) multistage stochastic programming". *Journal of Heuristics* 2.2, pp. 111–128. DOI: 10.1007/BF00247208.

Manousakis, E. G., Kasapidis, G. A., Kiranoudis, C. T., and Zachariadis, E. E. (2022). "An infeasible space exploring matheuristic for the Production Routing Problem". *European Journal of Operational Research* 298.2, pp. 478–495. DOI: 10.1016/j.ejor.2021.05.037.

Miranda, P. L., Cordeau, J.-F., Ferreira, D., Jans, R., and Morabito, R. (2018). "A decomposition heuristic for a rich production routing problem". *Computers & Operations Research* 98, pp. 211–230. DOI: 10.1016/j.cor.2018.05.004.

Mousavi, R., Bashiri, M., and Nikzad, E. (2022). "Stochastic production routing problem for perishable products: Modeling and a solution algorithm". *Computers & Operations Research* 142, p. 105725. DOI: 10.1016/j.cor.2022.105725.

Neves-Moreira, F., Almada-Lobo, B., Cordeau, J.-F., Guimarães, L., and Jans, R. (2019). "Solving a large multi-product production-routing problem with delivery time windows". *Omega* 86, pp. 154–172. DOI: 10.1016/j.omega.2018.07.006.

Qiu, Y., Ni, M., Wang, L., Li, Q., Fang, X., and Pardalos, P. M. (2018). "Production routing problems with reverse logistics and remanufacturing". *Transportation Research*

Part E: Logistics and Transportation Review 111, pp. 87–100. DOI: 10.1016/j.tre.2018.01.009.

Qiu, Y., Qiao, J., and Pardalos, P. M. (2017). "A branch-and-price algorithm for production routing problems with carbon cap-and-trade". *Omega* 68, pp. 49–61. DOI: 10.1016/j.omega.2016.06.001.

Qiu, Y., Qiao, J., and Pardalos, P. M. (2019). "Optimal production, replenishment, delivery, routing and inventory management policies for products with perishable inventory". *Omega* 82, pp. 193–204. DOI: 10.1016/j.omega.2018.01.006.

Rockafellar, R. T. and Wets, R. J. B. (1991). "Scenarios and Policy Aggregation in Optimization Under Uncertainty". *Mathematics of Operations Research* 16.1, pp. 119–147. DOI: 10.1287/moor.16.1.119.

Rodrigues, L. F., Santos, M. O. D., and Almada-Lobo, B. (2023). "A Memetic Algorithm for the multi-product Production Routing Problem". *Computers & Industrial Engineering* 182, p. 109388. DOI: 10.1016/j.cie.2023.109388.

Russell, R. A. (2017). "Mathematical programming heuristics for the production routing problem". *International Journal of Production Economics* 193, pp. 40–49. DOI: 10.1016/j.ijpe.2017.06.033.

Schenekemberg, C. M., Scarpin, C. T., Pécora, J. E., Guimarães, T. A., and Coelho, L. C. (2021). "The two-echelon production-routing problem". *European Journal of Operational Research* 288.2, pp. 436–449. DOI: 10.1016/j.ejor.2020.05.054.

Schenekemberg, C. M., Guimarães, T. A., Augusto, C. A., and Coelho, L. C. (2023). "A Three-Front Parallel Branch-and-Cut Algorithm for Production and Inventory Routing Problems". *Transportation Science*. DOI: 10.1287/trsc.2022.0261.

Shuang, Y., Diabat, A., and Liao, Y. (2019). "A stochastic reverse logistics production routing model with emissions control policy selection". *International Journal of Production Economics* 213, pp. 201–216. DOI: 10.1016/j.ijpe.2019.03.006.

Solyalı, O. and Süral, H. (2017). "A multi-phase heuristic for the production routing problem". *Computers & Operations Research* 87, pp. 114–124. DOI: 10.1016/j.cor.2017.06.007.

Solyalı, O. and Süral, H. (2022). “An Effective Matheuristic for the Multivehicle Inventory Routing Problem”. *Transportation Science* 56.4, pp. 1044–1057. DOI: 10.1287/trsc.2021.1123.

Tempelmeier, H. and Herpers, S. (2010). “ABC β–a heuristic for dynamic capacitated lot sizing with random demand under a fill rate constraint”. *International Journal of Production Research* 48.17, pp. 5181–5193. DOI: 10.1080/00207540903179782.

Tunc, H., Kilic, O. A., Tarim, S. A., and Eksioglu, B. (2013). “A simple approach for assessing the cost of system nervousness”. *International Journal of Production Economics* 141.2, pp. 619–625. DOI: 10.1016/j.ijpe.2012.09.022.

Vadseth, S. T., Andersson, H., Stålhane, M., and Chitsaz, M. (2023). “A multi-start route improving matheuristic for the production routeing problem”. *International Journal of Production Research* 61.22, pp. 7608–7629. DOI: 10.1080/00207543.2022.2154402.

Zhang, Y., Alshraideh, H., and Diabat, A. (2018). “A stochastic reverse logistics production routing model with environmental considerations”. *Annals of Operations Research* 271.2, pp. 1023–1044. DOI: 10.1007/s10479-018-3045-2.

Chapter 2

The Stochastic Production Routing Problem with Adaptive Routing and Service Level Constraints

Abstract

Demand uncertainty poses a challenge to most companies in manufacturing and services as it can lead to significant profit losses if not addressed properly. To deal with this risk, companies may adopt specific service level targets to satisfy at least a certain proportion of their demand while considering operational constraints and minimizing the total cost. In this study we address the stochastic production routing problem (PRP) with adaptive routing and service level constraints. The PRP unifies the production, inventory and routing decisions into an integrated problem aimed at improving coordination across different parts of the system. We consider four different types of service levels, where each type uses a specific metric based on assumptions aligning with the needs of the company. These metrics encompass aspects such as the occurrence of stockouts or allowed ratios of backlogs or backorders to average demand. A two-stage stochastic formulation is proposed for each type of service level. Setup decisions are made in the first stage, and

production, inventory, and routing decisions are adapted after demand realization. Considering routing decisions in the second stage increases flexibility while lowering overall costs. However, the resulting optimization problem is more challenging to solve than the case where routing decisions are made in the first stage. To address this issue, we introduce an iterative matheuristic algorithm designed to yield high-quality solutions within a reasonable computation time. The effectiveness of the proposed heuristic algorithm is demonstrated through extensive experiments, highlighting its potential to assist companies in managing demand uncertainty and enhancing operational efficiency.

2.1 Introduction

The production routing problem (PRP) combines the optimization of production, inventory, and routing decisions into an integrated problem. In the classical setting of this problem, a single production facility is responsible for both manufacturing and distributing a product to a set of dispersed customers in a discrete and finite time horizon. Integrating the different types of decisions enhances the overall efficiency of the supply chain, resulting in more cost-effective decision-making. The PRP was originally introduced by Chandra (1993) and Chandra and Fisher (1994), with the primary goal of reducing the waste resulting from treating the problems individually.

A comprehensive review of various formulations and solution algorithms for the PRP can be found in the work of Adulyasak et al. (2015b). A systematic review of this subject is also presented by Hrabec et al. (2022), who address two pivotal questions: what is the added value of integration and what are the circumstances under which it proves most advantageous to consider this integrated problem. Their findings indicate cost savings ranging from 6.58% to 15.58%, with a 95% confidence level, based on previous studies. Furthermore, they suggest that the benefits of integration tend to diminish in supply chains with higher production costs.

In the deterministic PRP (DPRP), a fundamental assumption is that all customer demands are known in advance, and the primary objective is to minimize production, inventory holding, and delivery costs. However, this simplified perspective does not adequately capture the complexities of the real world, where demand is often subject to uncertainty. Considering demand uncertainty allows companies to better prepare for demand variations, reducing costs associated with last-minute adjustments and avoiding unplanned stockouts.

There are various approaches to address uncertainty in the PRP. One method involves introducing a penalty cost per unit of unmet demand, which could represent the cost of fulfilling demand through third-party suppliers or the cost incurred due to lost sales. Adulyasak et al. (2015a) introduce two-stage and multi-stage stochastic programming models to tackle the stochastic PRP (SPRP). Their study demonstrates the benefits of incorporating demand uncertainty and provides valuable insights for problems where routing decisions must be fixed in advance. While this approach is well-suited for problems with operational constraints that require fixed routes, it may be less effective in situations where routing flexibility is available to adapt to fluctuating demand.

To address this limitation, Kermani et al. (2024) propose a two-stage stochastic programming model addressing the stochastic PRP with adaptive routing (SPRP-AR). Adaptive routing refers to the ability to determine delivery routes in the second stage after the realization of demand scenarios. Unlike traditional approaches where routes are predetermined and must be adhered to regardless of actual demand, adaptive routing enables the optimization of routes based on scenario-specific information. This flexibility allows the model to better respond to varying demand patterns and to reduce transportation costs by adapting routes to the specific needs of each scenario. While incorporating adaptive routing increases the complexity of the problem, the findings of Kermani et al. (2024) underscore its significance. The results indicate an average improvement of 6.5% in cost efficiency compared to traditional approaches that fix routing decisions in the first stage. This improvement is attributed to the ability to dynamically adjust routes based on realized demand, enabling better alignment between operational decisions and actual demand conditions.

Another strategy to handle uncertainty is adopting a specific service level strategy, which aims to balance cost reduction and demand satisfaction. This approach is particularly important in situations where outsourcing is not a viable option, as it ensures demand fulfillment while managing risks. Various service level measures have been proposed and studied in the literature, especially in the context of the lot-sizing problem (LSP) (Helber et al., 2013; Tempelmeier, 2013). These studies typically focus on production and inventory decisions, proposing different service level metrics to manage uncertainty. However, the PRP, as an extension of the LSP, introduces the additional challenge of routing decisions alongside production and inventory planning. Despite its practical importance, the integration of service level constraints into the PRP framework remains largely unexplored.

While service level constraints have been extensively studied in the LSP literature, these studies do not address the complexities introduced by routing decisions in the PRP, where demand uncertainty significantly impacts both production and transportation planning. To the best of our knowledge, our research is the first to incorporate service level constraints into the SPRP-AR. By addressing this gap, our work offers a novel framework for integrating service level constraints, enabling more practical solutions to real-world problems. This contribution bridges the gap between the well-established LSP literature and the PRP, providing valuable insights for managing uncertainty in scenarios where both production and routing decisions must be optimized simultaneously.

The first service level measure considered in our study is the so-called α service level, which can be characterized as an event-based service level. Specifically, α constrains the probability that available inventory will fail to meet the entire demand. By enforcing a predefined probability threshold for fully satisfying demand, this measure is particularly relevant in scenarios where a fixed cost is incurred for any stockout, irrespective of the shortage duration or quantity (Silver et al., 1998). The second service level, denoted by β, is more commonly known as the fill rate. This service level limits the ratio of average backorders to average demand, or equivalently, the proportion of demand met directly from on-hand inventory. It is widely used in practice and is especially relevant when each unit of unsatisfied demand incurs a specific cost, such as overtime production needed to fulfill backorders (Silver et al., 1998). However, it has the disadvantage that if demand is backlogged, then the waiting time for the customer is not considered (Helber et al., 2013). To remedy this, the next two service levels are specifically developed for the case where backlogging is allowed.

The third service level, γ, specifies that the ratio of expected backlogs to average demand must not exceed a predefined threshold. This measure is particularly relevant when not only the amount of shortage but also the time-related cost of those shortages is important (Silver et al., 1998). However, the gamma service level does not have a clear managerial interpretation since it can become negative or undefined (Helber et al., 2013). Finally, the δ service level limits the expected backlog to the maximum expected backlog. It has been shown by Helber et al. (2013) that this service level is a linear function of the average expected waiting time. By incorporating these service levels into the SPRP-AR framework, our study enables decision-makers to tailor their strategies to a wide range of operational contexts and industry-specific requirements.

In addition to these four types of service levels, we also explore various levels of granularity for each type of service levels. Unlike the LSP, where service levels are typically applied globally from the plant's standpoint, the PRP allows for the consideration of inventory levels and demand for each specific customer. As a result, we investigate service levels from both the customer's perspective, with distinct constraints for each customer, and from the plant's perspective, where the service level is aggregated across all customers. Furthermore, following the approach presented in Tempelmeier (2013), we investigate service levels for a single period or over the entire planning horizon. Considering the different combinations of both types of aggregation level (per customer or aggregated, and per period or over the whole horizon) results in a total of four distinct granularity levels for each of the four available service levels.

The first contribution of our study is the development of a novel two-stage stochastic formulation for the PRP with demand uncertainty and adaptive routing under four distinct service level metrics. While Kermani et al. (2024) introduced the SPRP-AR to address the limitations of fixed routing decisions in traditional stochastic PRP models, it does not incorporate service level constraints. This study extends the existing framework by formulating service levels as chance constraints, providing a practical approach for balancing cost efficiency with customer satisfaction.

In our formulation, setup decisions are made in the first stage, while production quantities, customer visits, routing decisions, and delivery quantities are determined in the second stage based on realized demand. This structure follows the static-dynamic strategy proposed by Bookbinder and Tan (1988), which has been widely applied in practice for its ability to balance cost efficiency and system stability. A key challenge in production planning is "nervousness", which arises from the frequent need to revise production decisions, leading to instability and costly disruptions (Kilic and Tarim, 2011; Koca et al., 2018). Tunc et al. (2013) argue that setup-oriented nervousness can be avoided with a minor cost, whereas quantity-oriented nervousness is significantly more expensive to address. Fixing setup decisions in advance prevents nervousness related to production periods while preserving the flexibility to adjust operational decisions as demand is realized. Thus, adopting the static-dynamic strategy can improve system performance compared to the static-static strategy.

The second contribution is the introduction of individual service level constraints for each customer, which extends prior work by providing a more granular approach to production, inventory, and routing planning. In contrast to global service level formulations often used in the LSP literature, our customer-specific constraints enable decision-makers to manage service levels at a more detailed level, ensuring that diverse customer requirements are met effectively. Our third contribution is the development of an iterative matheuristic (IMH) algorithm designed to address the complexity of the SPRP-AR, particularly with respect to its adaptive routing component in the second stage. The IMH algorithm integrates service level constraints and dynamically refines solutions across three distinct phases. In the first phase, we solve a relaxation of a two-level LSP with a single aggregated customer and direct shipments to generate production setup decisions. In the second phase, we solve a restricted PRP with fixed setup plans from the first phase and a single aggregated vehicle, incorporating service level constraints to ensure feasibility. Finally, in the third phase, we refine routing decisions while maintaining flexibility in production and inventory decisions.

In addition, we apply the proposed formulation and algorithm to a range of benchmark instances, featuring various service levels, uncertainty thresholds, and levels of granularity. Our results highlight the distinct benefits and trade-offs among different types of service level measures, emphasize the value of adaptive routing for added flexibility, and demonstrate how different levels of granularity influence overall performance. Furthermore, we provide managerial insights based on these findings, offering practical guidance for decision-makers confronting similar challenges.

The remainder of the paper is structured as follows. A comprehensive review of the related literature is presented in Section 2.2. The proposed formulations for different service levels are detailed in Section 2.3. The solution algorithm is thoroughly discussed in Section 2.4. In Section 2.5, we apply the algorithm to benchmark instances, reporting results and findings from these experiments. Finally, in Section 2.6, we draw conclusions based on our research.

2.2 Literature Review

Although the PRP yields enhanced efficiency compared to the isolated planning of production, inventory, and routing, obtaining a high-quality solution remains challenging due to the inherent difficulty of the problem. Several studies focus on refining exact algorithms to address the DPRP, yet their applicability is confined to small or medium-scale instances. An early endeavor to devise an exact solution algorithm for the PRP is the branch-and-price algorithm proposed by Bard and Nananukul (2010). Archetti et al. (2011) introduce a branch-and-cut (BC) algorithm, developed for addressing a PRP featuring a single vehicle. This study investigates two replenishment policies: Order-Up-To-Level (OU) and Maximum Level (ML) policies. In the OU policy, deliveries are forced to fill the inventory to its maximum level, whereas the more flexible ML policy allows for any delivery quantity within the inventory capacity limit.

Adulyasak et al. (2014a) present a BC algorithm for the PRP, considering two formulations based on whether routing variables incorporate a vehicle index or not. They also explore the implications of the ML and OU replenishment policies in their analysis. Schenekemberg et al. (2021) combine a Local Search (LS) heuristic with a BC algorithm within a parallel framework to address a two-echelon PRP. This algorithm proves effective for the classical PRP by yielding several new best-known solutions for benchmark instances. Building upon this algorithm, Schenekemberg et al. (2023) introduce an improved algorithm by incorporating both the two-index and three-index PRP formulations as well as an LS heuristic within a framework that addresses each problem independently, while information is shared across these problems to enhance overall algorithmic efficiency.

Due to the limitations of exact algorithms, a predominant trend in research involves the development of heuristic and matheuristic algorithms aimed at obtaining high-quality solutions for larger instances that are closer to the real-world problems. Archetti et al. (2011) present a hybrid heuristic for the PRP that decomposes the problem into two sequential subproblems, employing an iterative remove-insert procedure to refine the solution. Adulyasak et al. (2014b) propose an Optimization-based Adaptive Large Neighborhood Search (Op-ALNS) algorithm, specifically designed to solve large PRP instances. Absi et al. (2015) introduce a two-phase matheuristic algorithm, decomposing the PRP into lot-sizing and routing phases that are iteratively solved. An approximate visit cost is employed in the first phase that is updated in each iteration to enhance solution quality.

Solyalı and Süral (2017) put forth a matheuristic approach based on the sequential solution of five mixed-integer programming (MIP) problems, achieving new best known solutions for several benchmark instances. The algorithm includes solving a giant Traveling Salesman Problem (TSP), a restricted PRP using the initial TSP tour, a Capacitated Vehicle Routing Problems (CVRP) for each period, an MIP permitting customer removal and insertion in each period for further improvement, and a final TSP for each period and vehicle. Russell (2017) propose two heuristic algorithms for the PRP. The first employs a set partitioning approach with predefined routes, while the second utilizes seed routes, proving more effective for larger instances. Manousakis et al. (2022) propose a novel approach, formulating the PRP as a two-commodity flow problem and developing a matheuristic algorithm that explores both the feasible and infeasible spaces of the problem.

Studies have investigated variants of the PRP other than its basic formulation to address specific real-world challenges. Qiu et al. (2017) propose a model that incorporates carbon emissions in the PRP under a carbon cap-and-trade system. Miranda et al. (2018) tackle the complexities of a multi-product PRP with a heterogeneous fleet of vehicles, incorporating the possibility of routes spanning over multiple periods. Li et al. (2019) present a three-level heuristic algorithm designed for solving a multi-product PRP with outsourcing. Avci and Topaloglu Yildiz (2020) introduce a matheuristic algorithm for solving the PRP with transshipment. Neves-Moreira et al. (2019) investigate a PRP variant within the real-world context of the meat industry, accounting for perishability and time windows. Their three-phase matheuristic algorithm addresses the complications of a PRP with multiple product families. **Alvarez2022** consider perishability in a PRP with transshipment, incorporating the assumption that the product value decreases over time. Their hybrid heuristic combines iterative local search (ILS) and MIP methods to enhance solutions.

Drawing inspiration from the petrochemical industry, Schenekemberg et al. (2021) introduce a two-echelon PRP, encompassing both material pickup from suppliers and final product delivery to customers. Gruson et al. (2023) propose a PRP formulation involving direct shipments from the plant to warehouses, with subsequent customer deliveries originating from the warehouse. Chitsaz et al. (2019) contribute to the literature by introducing the Assembly Routing Problem (ARP), essentially the inbound version of the PRP. This problem focuses on the delivery of components from suppliers to the plant and they propose a decomposition matheuristic to solve the problem. In another paper, Chitsaz et al. (2020) introduce valid inequalities to strengthen the ARP formulation and propose an exact BC algorithm.

Addressing demand uncertainty represents a valuable extension of the PRP. Adulyasak et al. (2015a) propose formulations for the PRP with demand uncertainty, incorporating a penalty cost for unmet demand within a given period. Their approach assumes fixed routing among all possible scenarios and employs a Benders decomposition technique for solving the problem. Zhang et al. (2018) present a two-stage stochastic formulation for the PRP, accounting for demand uncertainty along with remanufacturing and simultaneous pickups and deliveries. Their model also incorporates considerations related to a carbon cap-and-trade emissions policy, highlighting the integration of environmental concerns into the PRP framework. Mousavi et al. (2022) introduce a formulation for the PRP considering both demand uncertainty and perishability. This extension adds another layer of complexity, incorporating time-sensitive considerations to the decision-making process. Kermani et al. (2024) investigate the advantages of considering adaptive routing in the SPRP with demand uncertainty. Their approach involves a matheuristic algorithm embedded within a progressive hedging framework to obtain routing decisions that are optimized for the realized scenario.

Service level constraints are widely used to address demand uncertainty, providing a structured way to maintain a desired level of demand satisfaction throughout the planning horizon. Such service levels are widely used in inventory management (Silver et al., 1998). The literature on LSP has also explored various service level measures, each with unique definitions and applications across different problem variants (Helber et al., 2013; Tempelmeier, 2007, 2013; Tempelmeier and Herpers, 2010). These constraints enable decision-makers to balance inventory costs with service performance, aligning with diverse operational objectives. Several studies underscore the advantages of incorporating service level constraints into LSP models. For example, Bookbinder and Tan (1988) classify strategies for managing uncertainty in LSP into three categories: static, dynamic, and static-dynamic, demonstrating their relevance in addressing service level requirements. Their study focuses on the α service level, which minimizes the probability of stockouts. The α service level is particularly valued in risk-averse settings, such as industries where high reliability is critical, and stockouts impose significant costs on the company. The α service level effectively minimizes supply disruptions by ensuring a minimum probability of meeting the entire demand. Tarim and Kingsman (2004) investigate the stochastic dynamic production/inventory LSP with α service constraints, showcasing its utility in ensuring a predefined stockout probability threshold in probabilistic scenarios. Tunc et al. (2014) address a stochastic LSP with α service level constraints, proposing a reformulation of the problem that strengthens its linear relaxation.

Among the various service levels, the β service level, defined as the fill rate or the proportion of demand met directly from available inventory, has been extensively studied due to its operational relevance. The β service level is well-suited for industries where, in addition to the possibility of stockouts, the amount of backorders is also a critical factor, making it essential to maintain a high level of customer satisfaction (Schneider, 1981). Tempelmeier (2007) analyzes the β service level in the stochastic uncapacitated LSP, emphasizing its significance in maintaining high customer satisfaction. This emphasis continues in later works such as Tempelmeier and Herpers (2010), where a heuristic approach for dynamic capacitated LSP incorporates the β service level, and Tempelmeier and Herpers (2011), which explores column generation heuristics under fill rate constraints. In addition, Tunc et al. (2018) introduce a MIP formulation for stochastic LSP under the static-dynamic strategy, accommodating both α and β service levels.

The γ service level is relevant in cases where backlogging is allowed. This service level limits the expected backlog relative to average demand, and accounts for the duration of stockouts in addition to the amount of stockouts (Schneider, 1981). Gade and Küçükyavuz (2013) examine a single-item LSP under γ service level constraints, while Stadtler and Meistering (2019) analyze the roles of α, β, and γ service levels in deterministic capacitated LSPs.

Another measure is the δ service level, which limits the ratio of expected backlog to the maximum possible backlog. This service level takes the waiting time of customers into account and is a linear function of the average expected waiting time (Helber et al., 2013). Gruson et al. (2018) study the effects of α, β, and δ service levels in capacitated LSPs with deterministic demands. Further, Sereshti et al. (2024) apply the δ service level in a two-stage stochastic programming framework for multi-level LSP, demonstrating its suitability for volatile supply chain environments. Sereshti et al. (2021) analyze the α, β, γ, and δ service levels within a multi-product lot-sizing framework. Similarly, Tomazella et al. (2023) investigate both aggregate and per-customer β service levels in an integrated procurement and lot-sizing model, illustrating the practicality of β service levels in operational decision-making.

While service levels such as α, β, γ, and δ have been extensively studied in the LSP literature, they have not been considered for the PRP. This gap is notable given the potential of service level constraints to enhance decision-making by balancing cost efficiency and demand satisfaction. Additionally, the importance of adaptive routing in dynamic environments cannot be overstated, as it allows for flexible and scenario-specific route adjustments based on realized demand. This flexibility is crucial in addressing the complexities of modern supply chains, where demand fluctuations and uncertainty significantly impact operational efficiency. By integrating service level constraints with adaptive routing, our study provides a novel framework to improve operational flexibility in supply chains.

2.3 Problem Formulation

We address the SPRP-AR in the presence of uncertain demand. Our study encompasses four distinct service levels, namely α, β, γ, and δ. The central focus is on a single product that is manufactured at a production plant and subsequently distributed to a total of N customers over a finite planning horizon where the set of periods is denoted by $\mathscr{T} = \{1,...,T\}$. To formally define the problem, we introduce a complete and undirected graph $\mathscr{G} = (\mathscr{N}, E)$, where $\mathscr{N} = \{0,...,N\}$ denotes the set of nodes, and $E = \big\{(i,j) : i,j \in \mathscr{N}, i < j\big\}$ represents the set of edges connecting each node pair in $\mathscr{N}$. The production plant is denoted by $\{0\}$, while the set $\mathscr{N}_c = \{1,...,N\}$ represents the customer nodes. The notation $\varepsilon(\{i\})$ represents the set of nodes that are incident to node $i \in \mathscr{N}$ (i.e., all nodes $j \in \mathscr{N}$ for which an edge $(i,j) \in E$ exists). Moreover, for any subset of nodes $\eta \subseteq \mathscr{N}$, $E(\eta)$ is defined as the set of edges $(i,j) \in E$ where both nodes i and j belong to η. The product is delivered to customers by a fleet of homogeneous vehicles, which are identified by the set $\mathscr{K} = \{1, ..., K\}$ with a specified capacity Q.

The demand of each customer in a given period is modeled as a continuous random variable following a probability distribution. By analyzing historical data, it is possible to derive the probability distribution function for the demand of each customer. To approximate demand based on the learned distribution, we consider a finite set of scenarios denoted by $\phi = \{1, ..., S\}$. Each scenario occurs with a probability $\xi_s > 0$, $\forall s \in \phi$, and $\sum_{s \in \phi} \xi_s = 1$. The stochastic variable representing customer demands is denoted by d_{it}^s, where i corresponds to a specific customer, t denotes the time period, and s refers to a particular scenario.

In this study, we adopt the static-dynamic strategy in a two-stage setting, wherein the setup decisions are made in the first stage, and the production quantities are determined after the demand realization (Bookbinder and Tan, 1988). The setup decisions are represented by binary decision variables y_t, with a value of 1 implying a setup with a fixed production cost of F when production occurs in period t, and 0 otherwise. The production quantities in each period t and scenario s are denoted by the recourse variables p_t^s, incurring a cost of u per produced unit. We consider a production capacity, represented by C. We define $\mathscr{M}_t^s = \min \left\{ \mathscr{C}, \sum_{l=t}^{T} \sum_{i \in \mathscr{N}_c} d_{il}^s \right\}$ as an upper bound on the amount produced in each period t and scenario s. Moreover, products can be stored either at the production plant or sent to customers for future consumption. The inventory level at each node i at the end of period t under scenario s is denoted by I_{it}^s, and an associated unit holding cost h_i is charged, with an inventory limit of L_i, $\forall i \in \mathscr{N}$. At the beginning of the planning horizon, an initial inventory of I_{i0} may exist at node i. In the presence of uncertain demand, stockouts may occur in each period, and backlogs are used to satisfy unmet demands in a future period in a First-In-First-Out (FIFO) order. The cumulative unmet demand of customer i in period t and scenario s is represented as the backlog variable b_{it}^s.

The transportation plan is a crucial component of the overall strategy, entailing adaptive routing decisions. To this end, we introduce the binary variable x_{ijkt}^s, which indicates whether edge $(i, j) \in E$ is traversed by vehicle k in period t under scenario s. The variable takes the value 1 if the edge is traversed, and 0 otherwise. Additionally, it may take the value 2 if only one node is visited by vehicle k in period t under scenario s, resulting in back-and-forth travel by the vehicle. The transportation cost c_{ij} is incurred whenever an edge $(i,j) \in E$ is traveled. The transportation plan is represented using the binary variable z^s_{ikt}, indicating whether node i is visited by vehicle k in period t under scenario s (1 if visited, 0 otherwise). Lastly, the continuous variable q^s_{ikt} defines the number of products delivered to node i by vehicle k in period t under scenario s, where split deliveries are not allowed. The maximum number of items that can be delivered to customer i is $\mathscr{W}^s_{it} = \min\left\{Q, L_i, \sum_{l=t}^{T} d^s_{il}\right\}$.

2.3.1 Formulation for the SPRP-AR with α Sevice Level

In this section, we present mathematical formulations for the SPRP-AR with α service level constraints, denoted as SPRP-AR$_\alpha$. The α service level aims to ensure a minimum level of service by controlling the probability of stockouts occurring in each period (α_c) or the average of this probability over the entire planning horizon (α_p). The former implies the minimal service level in each period, while the latter can be considered as the mean service level Tempelmeier (2013). To formulate this service level, we introduce the binary variable o^s_{it}, which takes a value of 1 when a stockout occurs at customer i in period t under scenario s. We also define $D^s_{it} = \sum_{l=1}^{t} d^s_{il}$, representing the cumulative demand of customer i up to period t under scenario s.

As mentioned earlier, we consider the service levels from both the plant and customer perspectives. We define the parameter $\alpha_c^{customer}$ as the service level that must be met for each customer i in period t. It ensures that the proportion of scenarios with a stockout does not exceed $1-\alpha_c^{customer}$. The formulation for the SPRP-AR$_{\alpha_c^{customer}}$ is as follows:

(SPRP-AR$_{\alpha_c^{customer}}$)

$$\min \sum_{t\in\mathscr{T}} \left(Fy_t + \sum_{s\in\phi} \xi_s \left(up^s_t + \sum_{(i,j)\in E} \sum_{k\in\mathscr{K}} c_{ij} x^s_{ijkt} + \sum_{i\in\mathscr{N}} h_i I^s_{it} \right) \right) \tag{2.1}$$

s.t.

$$p_t^s \leq \mathscr{M}_t^s y_t \qquad \forall t \in \mathscr{T}, s \in \phi \quad (2.2)$$

$$I_{0t}^s = I_{0,t-1}^s + p_t^s - \sum_{i \in \mathscr{N}_c} \sum_{k \in \mathscr{K}} q_{ikt}^s \qquad \forall t \in \mathscr{T}, s \in \phi \quad (2.3)$$

$$I_{it}^s = b_{it}^s + I_{i,t-1}^s + \sum_{k \in \mathscr{K}} q_{ikt}^s - b_{i,t-1}^s - d_{it}^s \qquad \forall i \in \mathscr{N}_c, t \in \mathscr{T}, s \in \phi \quad (2.4)$$

$$I_{0t}^s \leq L_0 \qquad \forall t \in \mathscr{T}, s \in \phi \quad (2.5)$$

$$I_{it}^s + d_{it}^s \leq L_i \qquad \forall i \in \mathscr{N}_c, t \in \mathscr{T}, s \in \phi \quad (2.6)$$

$$\sum_{i \in \mathscr{N}_c} q_{ikt}^s \leq Q z_{0kt}^s \qquad \forall k \in \mathscr{K}, t \in \mathscr{T}, s \in \phi \quad (2.7)$$

$$\sum_{k \in \mathscr{K}} z_{ikt}^s \leq 1 \qquad \forall i \in \mathscr{N}_c, t \in \mathscr{T}, s \in \phi \quad (2.8)$$

$$q_{ikt}^s \leq \mathscr{W}_{it}^s z_{ikt}^s \qquad \forall i \in \mathscr{N}_c, k \in \mathscr{K}, t \in \mathscr{T}, s \in \phi \quad (2.9)$$

$$D_{it}^s - \sum_{l=1}^{t} \sum_{k \in \mathscr{K}} q_{ikl}^s - I_{i0} \leq D_{it}^s o_{it}^s \qquad \forall i \in \mathscr{N}_c, t \in \mathscr{T}, s \in \phi \quad (2.10)$$

$$\sum_{s \in \phi} \xi_s o_{it}^s \leq 1 - \alpha_c^{customer} \qquad \forall i \in \mathscr{N}_c, t \in \mathscr{T} \quad (2.11)$$

$$\sum_{(j,j') \in \varepsilon(\{i\})} x_{jj'kt}^s = 2 z_{ikt}^s \qquad \forall i \in \mathscr{N}, k \in \mathscr{K}, t \in \mathscr{T}, s \in \phi \quad (2.12)$$

$$\sum_{(i,j) \in E(\eta)} x_{ijkt}^s \leq \sum_{i \in \eta} z_{ikt}^s - z_{ekt}^s \qquad \forall \eta \subseteq \mathscr{N}_c, |\eta| \geq 2, e \in \eta, k \in \mathscr{K}, t \in \mathscr{T}, s \in \phi \quad (2.13)$$

$$y_t \in \{0,1\} \qquad \forall t \in \mathscr{T} \quad (2.14)$$

$$p_t^s \geq 0 \qquad \forall t \in \mathscr{T}, s \in \phi \quad (2.15)$$

$$q_{ikt}^s \geq 0 \qquad \forall i \in \mathscr{N}_c, k \in \mathscr{K}, t \in \mathscr{T}, s \in \phi \quad (2.16)$$

$$I_{it}^s \geq 0 \qquad \forall i \in \mathscr{N}, t \in \mathscr{T}, s \in \phi \quad (2.17)$$

$$b_{it}^s \geq 0 \qquad \forall i \in \mathscr{N}_c, t \in \mathscr{T}, s \in \phi \quad (2.18)$$

$$z_{ikt}^s \in \{0,1\} \qquad \forall i \in \mathscr{N}, k \in \mathscr{K}, t \in \mathscr{T}, s \in \phi \quad (2.19)$$

$$x_{ijkt}^s \in \{0,1\} \qquad \forall (i,j) \in E, i \neq 0, k \in \mathscr{K}, t \in \mathscr{T}, s \in \phi \quad (2.20)$$

$$x_{0jkt}^s \in \{0,1,2\} \qquad \forall j \in \mathscr{N}_c, k \in \mathscr{K}, t \in \mathscr{T}, s \in \phi \quad (2.21)$$

$$o_{it}^s \in \{0,1\} \qquad \forall i \in \mathscr{N}_c, t \in \mathscr{T}, s \in \phi. \quad (2.22)$$

The objective function (2.1) of the proposed model aims to minimize the overall cost, which includes setup costs and the expected cost of the second-stage variables encompassing production, routing, and inventory holding. To ensure proper production planning, we impose constraints (2.2), which dictate that setups must occur when any production is done in a specific period. We set inventory balance constraints for both the production plant (2.3) and customers (2.4). Constraints (2.5) control the inventory capacity at the plant, ensuring that it does not exceed the defined limit. Constraints (2.6) impose an upper bound on customer inventory that restricts the sum of the inventory at the end of the period and the demand during that period to be smaller than or equal to the customer's storage capacity.

To manage the vehicle fleet, constraints (2.7) dictate that a vehicle must depart from the plant if it is assigned to deliver products to customers while adhering to the vehicle's capacity constraint. For the delivery process, constraint set (2.8) ensures that each customer can only be visited by one vehicle within a given period, preventing split deliveries. Additionally, the constraint set (2.9) imposes a maximum limit on the delivery quantity to each customer. To guarantee a certain service level denoted by $\alpha_c^{customer}$, two sets of constraints are introduced. Constraint set (2.10) activates binary variable o_{it}^s in case of stockouts occurring for a specific customer in a particular period and scenario. Constraint set (2.11) limits the probability of a stockout for each customer i in period t. We assume the service level agreement for each customer is set individually, hence the service level is modeled for each client separately.

To ensure proper connections, constraint set (2.12) mandates that each visited node should be connected by two traveled edges. To accommodate an adaptive routing strategy, subtour elimination constraints (SECs) are applied to each period and scenario, as specified by constraint set (2.13). This form of SECs is shown to be efficient when used in the BC algorithm (Adulyasak et al., 2014a, 2015b). Finally, the decision variables are appropriately constrained in (2.14) to (2.22), defining their scope and feasible ranges in the problem.

For $\alpha_p^{customer}$, which enforces the mean service level for each customer i, the corresponding constraints (2.11) should be replaced with constraints (2.24). The updated formulation is as follows:

$(\text{SPRP-AR}_{\alpha_p^{customer}})$

$$\min \sum_{t\in\mathscr{T}} \left(Fy_t + \sum_{s\in\phi} \xi_s \Big(up_t^s + \sum_{(i,j)\in E} \sum_{k\in\mathscr{K}} c_{ij} x_{ijkt}^s + \sum_{i\in\mathscr{N}} h_i I_{it}^s \Big) \right) \tag{2.23}$$

s.t. (2.2) - (2.10) and (2.12) - (2.22),

$$\frac{\sum_{t\in\mathscr{T}} \sum_{s\in\phi} \xi_s o_{it}^s}{T} \leq 1 - \alpha_p^{customer} \qquad \forall i \in \mathscr{N}_c. \tag{2.24}$$

From the plant's perspective, considering α_c^{plant} and α_p^{plant} service levels involves accounting for stockouts across all customers collectively. Therefore, the binary variable o_t^s is defined, taking a value of 1 when a stockout occurs for any customer in period t under scenario s. To formulate α_c^{plant}, constraints (2.10) and (2.11) must be replaced by (2.26) and (2.27), respectively:

$(\text{SPRP-AR}_{\alpha_c^{plant}})$

$$\min \sum_{t\in\mathscr{T}} \left(Fy_t + \sum_{s\in\phi} \xi_s \Big(up_t^s + \sum_{(i,j)\in E} \sum_{k\in\mathscr{K}} c_{ij} x_{ijkt}^s + \sum_{i\in\mathscr{N}} h_i I_{it}^s \Big) \right) \tag{2.25}$$

s.t. (2.2) - (2.9) and (2.12) - (2.21),

$$D_{it}^s - \sum_{l=1}^{t} \sum_{k\in\mathscr{K}} q_{ikl}^s - I_{i0} \leq D_{it}^s\, o_t^s \qquad \forall i \in \mathscr{N}_c, t \in \mathscr{T}, s \in \phi \tag{2.26}$$

$$\sum_{s\in\phi} \xi_s o_t^s \leq 1 - \alpha_c^{plant} \qquad \forall t \in \mathscr{T} \tag{2.27}$$

$$o_t^s \in \{0,1\} \qquad \forall t \in \mathscr{T}, s \in \phi. \tag{2.28}$$

Finally, for the α_p^{plant} service level, constraints (2.30) are used instead of constraints (2.27):

$(\text{SPRP-AR}_{\alpha_p^{plant}})$

$$\min \sum_{t\in\mathscr{T}} \left(Fy_t + \sum_{s\in\phi} \xi_s \Big(up_t^s + \sum_{(i,j)\in E} \sum_{k\in\mathscr{K}} c_{ij} x_{ijkt}^s + \sum_{i\in\mathscr{N}} h_i I_{it}^s \Big) \right) \tag{2.29}$$

s.t. (2.2) - (2.9), (2.12) - (2.21), (2.26), and (2.28),

$$\frac{\sum_{t\in\mathscr{T}} \sum_{s\in\phi} \xi_s o_t^s}{T} \leq 1 - \alpha_p^{plant}. \tag{2.30}$$

2.3.2 Formulation for the SPRP-AR with β Sevice Level

The β service level operates based on the backorders, which represents the amount of demand that has not been fulfilled in the period when it occurred (Tempelmeier, 2013). To calculate backorders, we introduce a non-negative continuous variable bo_{it}^{s} that can be computed using the following non-linear constraint:

$$bo_{it}^{s} = \min\left\{b_{it}^{s}, d_{it}^{s}\right\} \qquad \forall i \in \mathcal{N}_c, t \in \mathcal{T}, s \in \phi. \quad (2.31)$$

It is important to note that this constraint holds true only when the First-In-First-Out (FIFO) strategy is employed. To linearize the above constraints, we adopt the linearization approach proposed by Tomazella et al., 2023 for the deterministic case. This involves introducing auxiliary binary variables v_{it}^{s}, which take the value 1 if $b_{it}^{s} < d_{it}^{s}$, and 0 otherwise. Similar to the α service level, we first define $\beta_c^{customer}$ which enforces the service level at the customer level and for any period t. The formulation for the SPRP-AR$_{\beta_c^{customer}}$ is as follows:

(SPRP-AR$_{\beta_c^{customer}}$)

$$\min \sum_{t\in\mathcal{T}} \left(Fy_t + \sum_{s\in\phi} \xi_s \left(up_t^s + \sum_{(i,j)\in E} \sum_{k\in\mathcal{K}} c_{ij} x_{ijkt}^{s} + \sum_{i\in\mathcal{N}} h_i I_{it}^{s} \right) \right) \quad (2.32)$$

s.t. (2.2) - (2.9) and (2.12) - (2.21),

$$bo_{it}^{s} \leq b_{it}^{s} \qquad \forall i \in \mathcal{N}_c, t \in \mathcal{T}, s \in \phi \quad (2.33)$$

$$bo_{it}^{s} \leq d_{it}^{s} \qquad \forall i \in \mathcal{N}_c, t \in \mathcal{T}, s \in \phi \quad (2.34)$$

$$d_{it}^{s} - b_{it}^{s} \leq d_{it}^{s}\, v_{it}^{s} \qquad \forall i \in \mathcal{N}_c, t \in \mathcal{T}, s \in \phi \quad (2.35)$$

$$b_{it}^{s} - bo_{it}^{s} \leq D_{it}^{s}\,(1 - v_{it}^{s}) \qquad \forall i \in \mathcal{N}_c, t \in \mathcal{T}, s \in \phi \quad (2.36)$$

$$b_{it}^{s} - d_{it}^{s} \leq D_{it}^{s}\,(1 - v_{it}^{s}) \qquad \forall i \in \mathcal{N}_c, t \in \mathcal{T}, s \in \phi \quad (2.37)$$

$$d_{it}^{s}\,(1 - v_{it}^{s}) \leq bo_{it}^{s} \qquad \forall i \in \mathcal{N}_c, t \in \mathcal{T}, s \in \phi \quad (2.38)$$

$$\frac{\sum_{s\in\phi} \xi_s\, bo_{it}^{s}}{\bar{d}_{it}} \leq 1 - \beta_c^{customer} \qquad \forall i \in \mathcal{N}_c, t \in \mathcal{T} \quad (2.39)$$

$$bo_{it}^{s} \geq 0 \qquad \forall i \in \mathcal{N}, t \in \mathcal{T}, s \in \phi \quad (2.40)$$

$$v_{it}^{s} \in \{0,1\} \qquad \forall i \in \mathcal{N}, t \in \mathcal{T}, s \in \phi. \quad (2.41)$$

The objective function (2.32) is similar to (2.1). Constraints (2.31) have been substituted with their linear counterparts, specifically constraints (2.33)-(2.38). Parameter $\beta_c^{customer}$ in the above formulation specifies the minimum expected service level. Consequently, the ratio of the expected backorder to the expected demand of customer i in period t ($\overline{d}_{it}$) should not exceed $1-\beta_c^{customer}$, as imposed by constraints (2.39).

The global β service level across customers, denoted as $\beta^{customer}$, is based on the cumulative expected backorder divided by the cumulative expected demand of the entire planning horizon. Thus, to formulate the SPRP-AR$_{\beta^{customer}}$, we need to impose constraints (2.43) instead of constraints (2.39):

(SPRP-AR$_{\beta^{customer}}$)

$$\min \sum_{t\in\mathscr{T}} \left(Fy_t + \sum_{s\in\phi} \xi_s \Big(up_t^s + \sum_{(i,j)\in E} \sum_{k\in\mathscr{K}} c_{ij} x_{ijkt}^s + \sum_{i\in\mathscr{N}} h_i I_{it}^s \Big) \right) \tag{2.42}$$

s.t. (2.2) - (2.9), (2.12) - (2.21), (2.33) - (2.38), and (2.40) - (2.41)

$$\frac{\sum_{t\in\mathscr{T}} \sum_{s\in\phi} \xi_s bo_{it}^s}{\sum_{t\in\mathscr{T}} \overline{d}_{it}} \le 1-\beta^{customer} \qquad \forall i\in\mathscr{N}_c. \tag{2.43}$$

To formulate SPRP-AR$_{\beta_c^{plant}}$, we replace constraints (2.39) with constraints (2.45):

(SPRP-AR$_{\beta_c^{plant}}$)

$$\min \sum_{t\in\mathscr{T}} \left(Fy_t + \sum_{s\in\phi} \xi_s \Big(up_t^s + \sum_{(i,j)\in E} \sum_{k\in\mathscr{K}} c_{ij} x_{ijkt}^s + \sum_{i\in\mathscr{N}} h_i I_{it}^s \Big) \right) \tag{2.44}$$

s.t. (2.2) - (2.9), (2.12) - (2.21), (2.33) - (2.38), and (2.40) - (2.41)

$$\frac{\sum_{s\in\phi} \xi_s \sum_{i\in\mathscr{N}_c} bo_{it}^s}{\overline{DP}_t} \le 1-\beta_c^{plant} \qquad \forall t\in\mathscr{T}, \tag{2.45}$$

where $\overline{DP}_t$ is the total average demand of all customers in period t ($\overline{DP}_t = \sum_{i\in\mathscr{N}_c} \overline{d}_{it}$). Lastly, to formulate SPRP-AR$_{\beta^{plant}}$, constraints (2.45) must be substituted with constraints (2.47):

(SPRP-AR$_{\beta^{plant}}$)

$$\min \sum_{t\in\mathscr{T}} \left(Fy_t + \sum_{s\in\phi} \xi_s \Big(up_t^s + \sum_{(i,j)\in E} \sum_{k\in\mathscr{K}} c_{ij} x_{ijkt}^s + \sum_{i\in\mathscr{N}} h_i I_{it}^s \Big) \right) \tag{2.46}$$

s.t. (2.2) - (2.9), (2.12) - (2.21), (2.33) - (2.38), and (2.40) - (2.41)

$$\frac{\sum_{t\in\mathscr{T}}\sum_{s\in\phi}\xi_s\sum_{i\in\mathscr{N}_c}bo^s_{it}}{\sum_{t\in\mathscr{T}}\overline{DP}_t}\leq 1-\beta^{plant}. \tag{2.47}$$

2.3.3 Formulation for the SPRP-AR with γ Service Level

The next service level that we investigate is the γ service level (Helber et al., 2013). To ensure a minimum service level of $\gamma_c^{customer}$, we enforce that the proportion of the expected backlog to the expected demand of customer i in period t cannot exceed $1-\gamma_c^{customer}$. The formulation for the SPRP-AR$_{\gamma_c^{customer}}$ is as follows:

(SPRP-AR$_{\gamma_c^{customer}}$)

$$\min\sum_{t\in\mathscr{T}}\left(Fy_t+\sum_{s\in\phi}\xi_s\Big(up^s_t+\sum_{(i,j)\in E}\sum_{k\in\mathscr{K}}c_{ij}x^s_{ijkt}+\sum_{i\in\mathscr{N}}h_iI^s_{it}\Big)\right) \tag{2.48}$$

s.t. (2.2) - (2.9) and (2.12) - (2.21),

$$\frac{\sum_{s\in\phi}\xi_s b^s_{it}}{\overline{d}_{it}}\leq 1-\gamma_c^{customer} \qquad \forall i\in\mathscr{N}_c, t\in\mathscr{T}. \tag{2.49}$$

The formulation for the SPRP-AR$_{\gamma^{customer}}$ constraints (2.49) are replaced by constraints (2.51). The modified formulation is as follows:

(SPRP-AR$_{\gamma^{customer}}$)

$$\min\sum_{t\in\mathscr{T}}\left(Fy_t+\sum_{s\in\phi}\xi_s\Big(up^s_t+\sum_{(i,j)\in E}\sum_{k\in\mathscr{K}}c_{ij}x^s_{ijkt}+\sum_{i\in\mathscr{N}}h_iI^s_{it}\Big)\right) \tag{2.50}$$

s.t. (2.2) - (2.9) and (2.12) - (2.21),

$$\frac{\sum_{t\in\mathscr{T}}\sum_{s\in\phi}\xi_s b^s_{it}}{\sum_{t\in\mathscr{T}}\overline{d}_{it}}\leq 1-\gamma^{customer} \qquad \forall i\in\mathscr{N}_c. \tag{2.51}$$

For the SPRP-AR$_{\gamma_c^{plant}}$ constraints (2.49) need to be substituted with (2.53):

(SPRP-AR$_{\gamma_c^{plant}}$)

$$\min\sum_{t\in\mathscr{T}}\left(Fy_t+\sum_{s\in\phi}\xi_s\Big(up^s_t+\sum_{(i,j)\in E}\sum_{k\in\mathscr{K}}c_{ij}x^s_{ijkt}+\sum_{i\in\mathscr{N}}h_iI^s_{it}\Big)\right) \tag{2.52}$$

s.t. (2.2) - (2.9) and (2.12) - (2.21),

$$\frac{\sum_{s\in\phi}\xi_s\sum_{i\in\mathscr{N}_c}b^s_{it}}{\overline{DP}_t} \leq 1-\gamma^{plant}_c \qquad \forall t\in\mathscr{T}. \quad (2.53)$$

Finally, in SPRP-AR$_{\gamma^{plant}}$, constraints (2.53) are replaced with (2.55):

(SPRP-AR$_{\gamma^{plant}}$)

$$\min\sum_{t\in\mathscr{T}}\left(Fy_t+\sum_{s\in\phi}\xi_s\left(up^s_t+\sum_{(i,j)\in E}\sum_{k\in\mathscr{K}}c_{ij}x^s_{ijkt}+\sum_{i\in\mathscr{N}}h_iI^s_{it}\right)\right) \quad (2.54)$$

s.t. (2.2) - (2.9) and (2.12) - (2.21),

$$\frac{\sum_{t\in\mathscr{T}}\sum_{s\in\phi}\xi_s\sum_{i\in\mathscr{N}_c}b^s_{it}}{\sum_{t\in\mathscr{T}}\overline{DP}_t} \leq 1-\gamma^{plant}. \quad (2.55)$$

2.3.4 Formulation for the SPRP-AR with δ Sevice Level

The last service level that we are going to discuss is the δ service level (Helber et al., 2013). The proportion of expected backlog to the maximum expected backlog is restricted by $1-\delta$, where δ denotes the minimum required service level. The formulation for the SPRP-AR$_{\delta^{customer}_c}$ is as follows:

(SPRP-AR$_{\delta^{customer}_c}$)

$$\min\sum_{t\in\mathscr{T}}\left(Fy_t+\sum_{s\in\phi}\xi_s\left(up^s_t+\sum_{(i,j)\in E}\sum_{k\in\mathscr{K}}c_{ij}x^s_{ijkt}+\sum_{i\in\mathscr{N}}h_iI^s_{it}\right)\right) \quad (2.56)$$

s.t. (2.2) - (2.9) and (2.12) - (2.21),

$$\frac{\sum_{s\in\phi}\xi_s b^s_{it}}{\sum_{l=1}^{t}\bar{d}_{il}} \leq 1-\delta^{customer}_c \qquad \forall i\in\mathscr{N}_c, t\in\mathscr{T}. \quad (2.57)$$

For the SPRP-AR$_{\delta^{customer}}$ we need to replace constraints (2.57) with constraints (2.59):

(SPRP-AR$_{\delta^{customer}}$)

$$\min\sum_{t\in\mathscr{T}}\left(Fy_t+\sum_{s\in\phi}\xi_s\left(up^s_t+\sum_{(i,j)\in E}\sum_{k\in\mathscr{K}}c_{ij}x^s_{ijkt}+\sum_{i\in\mathscr{N}}h_iI^s_{it}\right)\right) \quad (2.58)$$

s.t. (2.2) - (2.9) and (2.12) - (2.21),

$$\frac{\sum_{t\in\mathscr{T}}\sum_{s\in\phi}\xi_s b_{it}^s}{\sum_{t\in\mathscr{T}}(T-t+1)\overline{d}_{it}} \leq 1-\delta^{customer} \qquad \forall i\in\mathscr{N}_c. \quad (2.59)$$

The formulation for the SPRP-AR$_{\delta_c^{plant}}$ is as follows:

(SPRP-AR$_{\delta_c^{plant}}$)

$$\min \sum_{t\in\mathscr{T}}\left(Fy_t+\sum_{s\in\phi}\xi_s\Big(up_t^s+\sum_{(i,j)\in E}\sum_{k\in\mathscr{K}}c_{ij}x_{ijkt}^s+\sum_{i\in\mathscr{N}}h_iI_{it}^s\Big)\right) \quad (2.60)$$

s.t. (2.2) - (2.9) and (2.12) - (2.21),

$$\frac{\sum_{s\in\phi}\xi_s\sum_{i\in\mathscr{N}_c}b_{il}^s}{\sum_{l=1}^{t}\overline{DP}_l} \leq 1-\delta_c^{plant} \qquad \forall t\in\mathscr{T}. \quad (2.61)$$

Finally, to formulate SPRP-AR$_{\delta^{plant}}$, constraints (2.61) are replaced by (2.63):

(SPRP-AR$_{\delta^{plant}}$)

$$\min \sum_{t\in\mathscr{T}}\left(Fy_t+\sum_{s\in\phi}\xi_s\Big(up_t^s+\sum_{(i,j)\in E}\sum_{k\in\mathscr{K}}c_{ij}x_{ijkt}^s+\sum_{i\in\mathscr{N}}h_iI_{it}^s\Big)\right) \quad (2.62)$$

s.t. (2.2) - (2.9) and (2.12) - (2.21),

$$\frac{\sum_{t\in\mathscr{T}}\sum_{s\in\phi}\xi_s\sum_{i\in\mathscr{N}_c}b_{it}^s}{\sum_{t\in\mathscr{T}}(T-t+1)\overline{DP}_t} \leq 1-\delta^{plant}. \quad (2.63)$$

It is worth highlighting that the objective function remains consistent across all problems, with service levels enforced through constraints based on their respective definitions.

2.4 Solution Algorithm

To address the problems presented in Section 2.3, we develop an iterative matheuristic algorithm (IMH). This approach involves breaking down the original problem into three distinct subproblems. The first subproblem is a stochastic two-level LSP with a single customer, which we refer to as the SLSP-SC. In this problem we introduce a dummy node to represent all customers in the problem. The primary objective of this subproblem is to rapidly generate setup plans for subsequent phases (Section 2.4.1).

Once a setup plan is established, we proceed to the second subproblem, which is a restricted stochastic PRP with a single vehicle and aggregate capacity (RSPRP-SV). This problem closely resembles the original problem with the exception of the routing aspect and having a fixed setup plan from the first phase. In this stage, our primary goal is to derive a high-quality solution for production planning and delivery quantities given the setup decisions made earlier. The routing decisions are not yet taken into account at this stage, as we reserve the optimization of routing for the subsequent phase of our algorithm. Instead, we assume a single vehicle with a modified aggregate capacity for delivering the products to customers (Section 2.4.2).

In the final phase, we use the fixed setup plan and also impose an upper limit on backlogs, where we tackle a restricted multi-vehicle deterministic PRP (RDPRP-MV) for each scenario to refine our routing decisions (Section 2.4.3). The purpose of setting a backlog upper bound is to ensure that the service levels achieved in the second phase are satisfied while giving the production and distribution decisions some level of flexibility.

Our algorithm involves both an outer iteration, which focuses on generating new setup plans that are fixed in the second and third phases (Diversification), and an inner iteration, where we update the estimated visit costs after the third phase and iterate over the second and third phases (Intensification). The algorithm iterates over either phases until a stopping criterion is met (Section 2.4.4). The intensification phase aims to enhance production and delivery solutions by considering the approximate visit cost updates after the third phase while taking service level constraints into account. It is important to note that, in the first iteration of the intensification phase, we employ RSPRP-SV to quickly verify the feasibility of the current setup plan. However, after obtaining a feasible solution from the second and third phases, we solve a modified version of RSPRP-SV in the second phase, which includes the approximate cost of visiting customers (RSPRP-AC). This modification contributes to obtaining improved solutions by incorporating an approximate delivery cost (Section 2.4.2).

To simplify the notations, we only provide the mathematical formulations of the algorithm for the per customer α service level. The same algorithm is applicable for all other cases by modifying the related service level constraints in the second phase. The overall structure of the algorithm is outlined in Algorithm 2.

Algorithm 2 Iterative Matheuristic Algorithm

```
initialize:
        Solve TSP.
        Set σ̂
        outerIter ← 0
        setupPool ← { }
        bestObjVal ← +∞
repeat
    outerIter ← outerIter + 1
    Cut setupPool from solution space of SLSP-SC.                ▷ Phase One
    Solve SLSP-SC                                                ▷ Phase One
    Fix y                                                        ▷ Phase One
    Add y to setupPool                                           ▷ Phase One
    innerIter ← 0
    repeat
        innerIter ← innerIter + 1
        if innerIter = 1 or RDPRP-MV(s) is infeasible for at least one s ∈ φ , then:
            Solve the RSPRP-SV                                   ▷ Phase Two
        else:
            Solve RSPRP-AC                                       ▷ Phase Two
        Set backlog upper bounds                                 ▷ Phase Three
        Solve RDPRP-MV(s) for all s ∈ φ                          ▷ Phase Three
        if Solution of RDPRP-MV(s) is feasible for all s ∈ φ
            Update σ_it^s                                        ▷ Phase Three
        else:
            Update λ                                             ▷ Phase Three
    until Stopping criterion is met (for inner iteration)
    Update objVal
    if (objVal < bestObjVal), then:
        Update bestObjVal
until Stopping criterion is met (for outer iteration)
return Incumbent Solution
```

2.4.1 Phase One: SLSP-SC

To formulate the SLSP-SC, we introduce a dummy node which forces us to modify certain parameters and variables. The demand of this dummy node is assumed to be the aggregated demand of all customers for a specific scenario, denoted by $\hat{d}_t^s$. The cost of visiting this single customer is set to the cost of an optimal Traveling Salesman Problem (TSP) tour over all nodes ($\hat{\sigma}$). It is important to note that the delivery to this node is also constrained by the aggregated capacity of all available vehicles. The binary variable e_t^s indicates the visit of the dummy node in period $t \in \mathscr{T}$ and scenario $s \in \phi$, and g_t^s represents the delivery quantity to this node.

Additionally, an initial inventory is considered equal to the sum of the initial inventory of all customers. The unit holding cost of this node is set as the average of the unit holding cost of all customers ($\hat{h}$). We define IP_t^s as the inventory level of the plant, ID_t^s as the inventory level of the customer, and bD_t^s as the backlog of the customer. We relax the integrality constraints on all binary variables, except for the y_t variables, to better focus on the purpose of the first phase. The formulation for the SLSP-SC$_\alpha$ is as follows:

$$(\text{SLSP-SC}_\alpha) \quad \min \sum_{t \in \mathscr{T}} \left(F y_t + \sum_{s \in \phi} \xi_s \left(u p_t^s + \hat{\sigma} e_t^s + h_0\, IP_t^S + \hat{h}\, ID_t^S \right) \right) \tag{2.64}$$

s.t. (2.2), (2.14), and (2.15)

$$IP_t^s = IP_{t-1}^s + p_t^s - g_t^s \qquad \forall t \in \mathscr{T}, s \in \phi \tag{2.65}$$

$$ID_t^s = bD_t^s + ID_{t-1}^s + g_t^s - bD_{t-1}^s - \hat{d}_t^s \qquad \forall t \in \mathscr{T}, s \in \phi \tag{2.66}$$

$$IP_t^s \leq L_0 \qquad \forall t \in \mathscr{T}, s \in \phi \tag{2.67}$$

$$ID_t^s + \hat{d}_t^s \leq \sum_{i \in \mathscr{N}_c} L_i \qquad \forall t \in \mathscr{T}, s \in \phi \tag{2.68}$$

$$g_t^s \leq \hat{\mathscr{W}}_t^s\, e_t^s \qquad \forall t \in \mathscr{T}, s \in \phi \tag{2.69}$$

$$\hat{D}_t^s - \sum_{l=1}^{t} g_l^s - ID_0 \leq \hat{D}_t^s\, \hat{o}_t^s \qquad \forall i \in \mathscr{N}_c, t \in \mathscr{T}, s \in \phi \tag{2.70}$$

$$\sum_{s \in \phi} \xi_s \hat{o}_t^s \leq 1 - \alpha \qquad \forall t \in \mathscr{T} \tag{2.71}$$

$$g_t^s \geq 0 \qquad \forall t \in \mathscr{T}, s \in \phi \tag{2.72}$$

$$IP_t^s \geq 0 \qquad \forall t \in \mathscr{T}, s \in \phi \tag{2.73}$$

$$ID_t^s \geq 0 \qquad \forall t \in \mathscr{T}, s \in \phi \tag{2.74}$$

$$bD_t^s \geq 0 \qquad \forall t \in \mathscr{T}, s \in \phi \tag{2.75}$$

$$0 \leq e_t^s \leq 1 \qquad \forall t \in \mathscr{T}, s \in \phi \quad (2.76)$$

$$0 \leq \hat{o}_t^s \leq 1 \qquad \forall t \in \mathscr{T}, s \in \phi. \quad (2.77)$$

The objective function (2.64) minimizes the setup cost and the expected production and holding cost of the plant, along with the expected approximate delivery and holding cost in the dummy node. Constraints (2.65) ensure the plant's inventory balance, while constraints (2.66) represent the customer's inventory balance. Constraints (2.67) and (2.68) set the inventory limits for the plant and the dummy customer, respectively. In constraints (2.69), we enforce the delivery limit for each period t and scenario s. To achieve this, we modify the delivery limit to the dummy customer by introducing $\hat{\mathscr{W}}_t^s = \min\left\{KQ, \sum_{i\in\mathscr{N}_c} L_i, \sum_{l=t}^{T} \hat{d}_l^s\right\}$. Constraints (2.70) and (2.71) are the modified constraints for the α service level. The variable $\hat{o}_t^s$ is assigned the value 1 if a stockout occurs in the dummy customer, and 0 otherwise. Additionally, we define $\hat{D}_t^s = \sum_{l=1}^{t} \hat{d}_l^s$ as the cumulative demand for the dummy node until period t.

Similarly, we can formulate the subproblem for the other three service levels. At each iteration of the algorithm we cut the previously explored setup plans from the solution space (line 9 of Algorithm 2) using the local branching constraints introduced in Fischetti and Lodi, 2003:

$$\sum_{t|\hat{y}_t=0} y_t + \sum_{t|\hat{y}_t=1} (1-y_t) \geq 1, \qquad (2.78)$$

where $\hat{y}_t$ is the value of the setup decisions in the previous iterations for period t. By adding these constraints for all the solutions that exist in the *setupPool*, we can make sure that the problem searches through new setup plans.

2.4.2 Second Phase

RSPRP-SV

In the second phase of the algorithm, the objective is to verify if the generated setup decisions from the first phase ($\hat{y}$) can result in a feasible solution when considering customers separately and enforcing the integrality of variables. However, to keep the problem tractable, we still maintain a single aggregate vehicle per period and scenario. To facilitate this, we introduce the variable $\check{q}_{it}^{s}$, representing the delivery quantity to customer i in period t under scenario s. The formulation for the RSPRP-SV with the α service level is presented below:

$$(\text{RSPRP-SV}_{\alpha}) \quad \min \sum_{t \in \mathscr{T}} \left(\sum_{s \in \phi} \xi_s \Big(u p_t^s + \sum_{i \in \mathscr{N}} h_i I_{it}^s \Big) \right) \tag{2.79}$$

s.t. (2.5), (2.6), (2.11), (2.15), (2.17), (2.18), and (2.22)

$$p_t^s \leq \mathscr{M}_t^s \hat{y}_t \qquad \forall t \in \mathscr{T}, s \in \phi \tag{2.80}$$

$$I_{0t}^s = I_{0,t-1}^s + p_t^s - \sum_{i \in \mathscr{N}_c} \check{q}_{it}^s \qquad \forall t \in \mathscr{T}, s \in \phi \tag{2.81}$$

$$I_{it}^s = b_{it}^s + I_{i,t-1}^s + \check{q}_{it}^s - b_{i,t-1}^s - d_{it}^s \qquad \forall i \in \mathscr{N}_c, t \in \mathscr{T}, s \in \phi \tag{2.82}$$

$$\sum_{i \in \mathscr{N}_c} \check{q}_{it}^s \leq \lambda K Q \qquad \forall t \in \mathscr{T}, s \in \phi \tag{2.83}$$

$$\check{q}_{it}^s \leq \mathscr{W}_{it}^s \qquad \forall i \in \mathscr{N}_c, t \in \mathscr{T}, s \in \phi \tag{2.84}$$

$$D_{it}^s - \sum_{l=1}^{t} \check{q}_{il}^s - I_{i0} \leq D_{it}^s o_{it}^s \qquad \forall i \in \mathscr{N}_c, t \in \mathscr{T}, s \in \phi \tag{2.85}$$

$$\check{q}_{it}^s \geq 0 \qquad \forall i \in \mathscr{N}_c, t \in \mathscr{T}, s \in \phi. \tag{2.86}$$

In the above formulation, the objective function (2.79) focuses solely on the expected production and holding costs. While incorporating visit variables and their associated costs can enhance solutions, it introduces extra binary variables, potentially increasing problem complexity. Therefore, in the first iteration, we solve the above problem to assess feasibility (line 16 of Algorithm 2). In the subsequent iterations of the intensification phase, where we solve the RSPRP-AC, we incorporate visit variables (line 18 of Algorithm 2).

It is important to note that when multiple vehicles are present in the problem, we introduce the modification coefficient $\lambda \in (0,1]$ to account for the consideration of a single vehicle with aggregate capacity. The initial value for λ is set at the beginning of each inner iteration, and we have the flexibility to update its value if the algorithm fails to obtain feasible routing decisions in the third phase (line 25 of Algorithm 2). It is possible to modify this coefficient a few times until it either leads to a feasible solution, where we can update the approximate visit costs (line 23 of Algorithm 2) and start the intensification phase, or we can decide to go back to the diversification phase to generate a new setup plan if the current setup plan is not promising.

RSPRP-AC

As mentioned earlier, if the third phase results in a feasible solution, we update the approximate visit costs and initiate the intensification phase, where we iterate over RSPRP-AC and RDPRP-MV to refine the recourse decisions for the current setup plan. The RSPRP-AC mirrors the RSPRP-SV, with the addition of the visit variable $\check{z}_{it}^{s}$ for visiting customer i in period t under scenario s, along with an associated approximate visit cost denoted as σ_{it}^{s}.

$$(\text{RSPRP-AC}_\alpha) \quad \min \sum_{t\in\mathscr{T}} \left(\sum_{s\in\phi} \xi_s \Big(up_t^s + \sum_{i\in\mathscr{N}_c} \sigma_{it}^{s} \check{z}_{it}^{s} + \sum_{i\in\mathscr{N}} h_i I_{it}^{s} \Big) \right) \tag{2.87}$$

s.t. (2.5), (2.6), (2.11), (2.15), (2.17), (2.18), (2.22), (2.80) - (2.83), (2.85), and (2.86)

$$\check{q}_{it}^{s} \leq \mathscr{W}_{it}^{s} \check{z}_{it}^{s} \qquad \forall i \in \mathscr{N}_c, t \in \mathscr{T}, s \in \phi \tag{2.88}$$

$$\check{z}_{it}^{s} \in \{0,1\} \qquad \forall i \in \mathscr{N}_c, t \in \mathscr{T}, s \in \phi. \tag{2.89}$$

2.4.3 Phase Three: RDPRP-MV

In the third phase of the algorithm, we impose an upper bound on backlogs. This strategy ensures the satisfaction of service level constraints while permitting minor adjustments in delivery schedules. However, for the β service level, we also need to fix the v_{it}^{s} variables to specify backorders explicitly and ensure service level satisfaction. By ensuring service level satisfaction, we can eliminate the linking constraints (2.11), (2.39), (2.49), and (2.57) for the α, β, γ, and δ service levels, respectively. Removing these linking constraints allows us to decompose the original problem into a deterministic PRP for each scenario s. Consequently, we must solve this problem for all scenarios (line 21 of Algorithm 2). For a given scenario s, the RDPRP-MV is formulated as follows:

$$(\text{RDPRP-MV}_{\alpha}(s)) \quad \min \quad \sum_{t\in\mathcal{T}} \Big(up_t + \sum_{(i,j)\in E} \sum_{k\in\mathcal{K}} c_{ij} x_{ijkt} + \sum_{i\in\mathcal{N}} h_i I_{it} \Big) \tag{2.90}$$

s.t.

$$p_t \le \mathscr{M}_t^{(s)} \hat{y}_t \qquad \forall t \in \mathcal{T} \tag{2.91}$$

$$I_{0t} = I_{0,t-1} + p_t - \sum_{i\in\mathcal{N}_c} \sum_{k\in\mathcal{K}} q_{ikt} \qquad \forall t \in \mathcal{T} \tag{2.92}$$

$$I_{it} = b_{it} + I_{i,t-1} + \sum_{k\in\mathcal{K}} q_{ikt} - b_{i,t-1} - d_{it}^{(s)} \qquad \forall i \in \mathcal{N}_c, t \in \mathcal{T} \tag{2.93}$$

$$I_{0t} \le L_0 \qquad \forall t \in \mathcal{T} \tag{2.94}$$

$$I_{it} + d_{it}^{(s)} \le L_i \qquad \forall i \in \mathcal{N}_c, t \in \mathcal{T} \tag{2.95}$$

$$\sum_{i\in\mathcal{N}_c} q_{ikt} \le Q z_{0kt} \qquad \forall k \in \mathcal{K}, t \in \mathcal{T} \tag{2.96}$$

$$\sum_{k\in\mathcal{K}} z_{ikt} \le 1 \qquad \forall i \in \mathcal{N}_c, t \in \mathcal{T} \tag{2.97}$$

$$q_{ikt} \le \mathscr{W}_{it}^{(s)} z_{ikt} \qquad \forall i \in \mathcal{N}_c, k \in \mathcal{K}, t \in \mathcal{T} \tag{2.98}$$

$$b_{it} \le \hat{b}_{it}^{(s)} \qquad \forall i \in \mathcal{N}_c, t \in \mathcal{T} \tag{2.99}$$

$$\sum_{(j,j')\in\varepsilon(\{i\})} x_{jj'kt} = 2 z_{ikt} \qquad \forall i \in \mathcal{N}, k \in \mathcal{K}, t \in \mathcal{T} \tag{2.100}$$

$$\sum_{(i,j)\in E(\eta)} x_{ijkt} \le \sum_{i\in\eta} z_{ikt} - z_{ekt} \qquad \forall \eta \subseteq \mathcal{N}_c, |\eta| \ge 2, e \in \eta, k \in \mathcal{K}, t \in \mathcal{T} \tag{2.101}$$

$$y_t \in \{0,1\} \qquad \forall t \in \mathcal{T} \tag{2.102}$$

$$p_t \ge 0 \qquad \forall t \in \mathcal{T} \tag{2.103}$$

$$q_{ikt} \ge 0 \qquad \forall i \in \mathcal{N}_c, k \in \mathcal{K}, t \in \mathcal{T} \tag{2.104}$$

$$I_{it} \ge 0 \qquad \forall i \in \mathcal{N}, t \in \mathcal{T} \tag{2.105}$$

$$b_{it} \ge 0 \qquad \forall i \in \mathcal{N}_c, t \in \mathcal{T} \tag{2.106}$$

$$z_{ikt} \in \{0,1\} \qquad \forall i \in \mathcal{N}, k \in \mathcal{K}, t \in \mathcal{T} \tag{2.107}$$

$$x_{ijkt} \in \{0,1\} \qquad \forall (i,j) \in E, i \ne 0, k \in \mathcal{K}, t \in \mathcal{T} \tag{2.108}$$

$$x_{0jkt} \in \{0,1,2\} \qquad \forall j \in \mathcal{N}_c, k \in \mathcal{K}, t \in \mathcal{T}. \tag{2.109}$$

The above formulation represents a deterministic PRP where the setup decisions ($\hat{y}$) are fixed from the first phase, and $\hat{b}_{it}^{(s)}$ denotes the obtained backlog from the second phase for customer i in period t under scenario s. It is crucial to emphasize that in addressing the RDPRP-MV, we initially relax the SECs. To effectively manage these constraints, we utilize a BC algorithm that strategically adds necessary constraints through a separation process. This critical step involves employing the minimum s-t cut algorithm from the Concorde solver as our separation algorithm (Applegate et al., 2020). The primary function of this algorithm is to identify and incorporate the violated constraints into the model.

Constraints (2.99) ensure the satisfaction of service levels. However, for the β service level, the following constraints are crucial to ensure the feasibility of the problem:

$$bo_{it} \leq b_{it} \qquad \forall i \in \mathcal{N}_c, t \in \mathcal{T} \quad (2.110)$$

$$bo_{it} \leq d_{it} \qquad \forall i \in \mathcal{N}_c, t \in \mathcal{T} \quad (2.111)$$

$$d_{it} - b_{it} \leq d_{it}\, \hat{v}_{it}^{(s)} \qquad \forall i \in \mathcal{N}_c, t \in \mathcal{T} \quad (2.112)$$

$$b_{it} - bo_{it} \leq D_{it}\, (1 - \hat{v}_{it}^{(s)}) \qquad \forall i \in \mathcal{N}_c, t \in \mathcal{T} \quad (2.113)$$

$$b_{it} - d_{it} \leq D_{it}\, (1 - \hat{v}_{it}^{(s)}) \qquad \forall i \in \mathcal{N}_c, t \in \mathcal{T} \quad (2.114)$$

$$d_{it}\, (1 - \hat{v}_{it}^{(s)}) \leq bo_{it} \qquad \forall i \in \mathcal{N}_c, t \in \mathcal{T} \quad (2.115)$$

$$bo_{it} \geq 0 \qquad \forall i \in \mathcal{N}, t \in \mathcal{T}, \quad (2.116)$$

where the $\hat{v}_{it}^{(s)}$ are obtained from the second phase of the IMH algorithm. Additionally, we incorporate the following valid inequalities to the third phase problem to enhance the algorithm's performance by eliminating symmetries from the solution space (Adulyasak et al., 2014b; Jans, 2009):

$$z_{ikt}^{s} \leq z_{0kt}^{s} \qquad \forall i \in \mathcal{N}_c, k \in \mathcal{K}, t \in \mathcal{T}, s \in \phi \quad (2.117)$$

$$z_{0kt}^{s} \leq z_{0,k-1,t}^{s} \qquad 2 \leq k \leq K, \forall t \in \mathcal{T}, s \in \phi \quad (2.118)$$

$$\sum_{i=1}^{j} 2^{(j-i)} z_{ikt}^{s} \leq \sum_{i=1}^{j} 2^{(j-i)} z_{i,k-1,t}^{s} \qquad \forall j \in \mathcal{N}_c, 2 \leq k \leq K, t \in \mathcal{T}, s \in \phi. \quad (2.119)$$

Upon completion of this phase, if the RDPRP-MV is feasible for all scenarios, the routing decisions obtained are guaranteed to be feasible for the main problem. Subsequently, we update the routing costs and return to solve RSPRP-AC. To modify the approximate visit costs σ_{it}^s, we adopt an approach similar to the one used by Absi et al., 2015 and Chitsaz et al., 2019. If customer i is visited in period t and scenario s, the visit cost is given by $\sigma_{it}^s = c_{j'i} + c_{ij''} - c_{j'j''}$, where j' and j'' are the nodes visited right before and after node i. If a node is not visited in period t and scenario s, we set σ_{it}^s to the minimum insertion cost of adding this node to the existing route.

2.4.4 Stopping Criteria

In the inner iteration, we set a maximum number of iterations or a time limit as the stopping criteria (line 26 of Algorithm 2). Following each inner iteration, if a feasible solution is obtained, it is compared to the previous best solution for the current setup plan, and the current $objVal$ is updated accordingly (line 27 of Algorithm 2). Conversely, if no improvement is achieved during the intensification phase, the process is stopped, and a new iteration is initiated to find a new setup plan. After the completion of each iteration, a comparison is made with the previous best solution, and if a better objective value is obtained, a new incumbent solution is found and the $bestObjVal$ is updated (lines 28-29 of Algorithm 2). Additionally, the process is terminated if a feasible solution cannot be found after updating λ for a specific number of iterations. The algorithm continues to iterate until either a maximum number of iterations or a maximum time limit is reached (line 30 of Algorithm 2).

2.5 Computational Experiments

In this section, we present the results of our experiments on the SPRP-AR with different service level constraints. Algorithms are programmed in C++ and run on a machine with Intel Xeon Gold 6148 2.4 GHz processors and 32GB of memory. We use IBM ILOG

CPLEX 22.1.0 as the solver to solve mathematical formulations.

2.5.1 Test Instances

To carry out our experiments, we generate instances using the benchmark examples outlined in Archetti et al., 2011 for the deterministic PRP. We adopt a methodology similar to that of Adulyasak et al., 2015a and Kermani et al., 2024 for generating scenarios through Monte Carlo Simulation. We have two sets in total. The first set, denoted as $\mathscr{S}$, comprises three periods $(T = 3)$ and up to three vehicles $(K \leq 3)$. Within this dataset, we consider $N = 5$ to 30 customers at intervals of 5 and we utilize the instance with high transportation costs from the original dataset as it better demonstrates the advantage of using adaptive routing. We solve each instance for all four service levels, setting the service level between 70% and 95% with 5% intervals, resulting in six different service level values for each type of service level. In total, we solve 4×108 instances for the first dataset.

The second dataset, labeled $\mathscr{L}$, contains larger instances with six periods and up to three vehicles, and nine periods with up to two vehicles. We consider $N \leq 30$ for $T = 6$ with either $K = 1$ or $K = 2$, and $T = 9$ with $K = 1$. For $T = 6$ with $K = 3$ and $T = 9$ with $K = 2$, we assume instances with $N \leq 20$. In this set, we again focus on the high transportation cost instance group, resulting in 156 instances for each type of service level. Overall, we solve 264 instances for each of the 4 combinations for the granularity level (per customer or aggregated, and per period or over the whole horizon) and service level type (α, β, γ and δ), amounting to a total of 4,224 experiments. We solve the problems with $S = 100$ unless stated otherwise. Table 2.1 summarizes the different instance configurations in our datasets.

Table 2.1: Datasets configuration

Dataset	No. of Instances	No. of Customers	No. of Periods	No. of Vehicles	SL values
$\mathscr{S}$	108	$N \leq 30$	3	1,2,3	$[70\%, 95\%]$
$\mathscr{L}$	108	$N \leq 30$	6/9	1,2/1	$[70\%, 95\%]$
	48	$N \leq 20$	6/9	3/2	$[70\%, 95\%]$

It is worth noting that we consider the demand of the deterministic dataset as the expected demand ($\overline{d}_{it}$) and utilize a discrete uniform distribution for generating stochastic demands in the range $[\overline{d}_{it}(1-\varepsilon), \overline{d}_{it}(1+\varepsilon)]$, where $0 \leq \varepsilon \leq 1$ represents the uncertainty level. The value of ε is set to 0.2 throughout all experiments unless specified otherwise. This baseline uncertainty level is chosen to analyze the model's response in a low-uncertainty environment and examine how even a small amount of uncertainty affects the solution. Additionally, higher uncertainty levels are explored in Section 2.5.7 to assess the model's performance under more uncertain conditions.

2.5.2 Algorithm Implementation

In our algorithm, we encounter various optimization problems at different stages, and the specifics of each problem's settings are discussed in this section. The maximum CPU time for the entire algorithm is set to 7200 seconds, with a maximum iteration limit of 50. For inner iterations, there is a time limit of 1800 seconds and a maximum iteration count of 10. As discussed in Section 2.4.2 the value of λ can be updated. Initially, λ is set to 1 for all configurations. However, if a feasible solution is found in the second phase but proves infeasible in the third phase (routing), λ is reduced by 0.1. The algorithm repeats this process three times until λ reaches 0.7 before concluding the inner iteration and returning to the diversification phase. It is important to note that the intensification phase only begins when feasible solutions for both the second and third phases are obtained using the latest λ.

In the first phase, 10 threads are allocated to CPLEX, with the optimality gap set to 10^{-6} and a time limit of 300 seconds. In the second phase, for RSPRP-SV, 10 threads are allocated and a time limit of 600 seconds is set. As the primary goal is to find a feasible solution, the MIP emphasis parameter of CPLEX is set to feasibility, and an optimality gap of 5% is employed. However, for solving RSPRP-AC, the CPLEX MIP emphasis parameter is set to optimality, and the gap is set to 1%. It is worth mentioning that, for enhanced efficiency, the solution from the previous iteration serves as a warm start for the next iteration within the intensification phase. In the third phase, where we need to solve

S deterministic subproblems, we use parallel computing by assigning one thread to each problem, using a total of 40 threads. For each subproblem a time limit of 600 seconds and a 1% gap is considered. We also add the SECs of a particular scenario from previous iterations to the root node of the branch-and-bound tree of future iterations.

We also solve the problem using a BC algorithm implemented in CPLEX to compare its results with those of the IMH. For details on the BC algorithm, readers can refer to Appendix B.1. However, the BC is unable to find feasible solutions even for the smallest instances. Consequently, the IMH solution serves as a warm start for the BC, and we report the improvements achieved through the BC algorithm. The BC is configured with 10 threads, 7,200 CPU seconds, and a 10^{-6} gap. For dataset $\mathscr{S}$, we also perform a comparative analysis between SPRP-AR and SPRP-FR (SPRP with first-stage routing) to illustrate how adaptive routing enhances flexibility and reduces overall costs. We solve SPRP-FR with the BC algorithm using the same configuration as for SPRP-AR. After obtaining the results of SPRP-FR, we apply a post-optimization approach to the best feasible solution from SPRP-FR to adjust routes to scenarios. This allows us to make a comparison between adaptive and fixed routing, as well as to determine whether it is better to solve SPRP-AR directly or to solve the SPRP-FR and adjust the routes.

In the post-optimization approach, we search each scenario for nodes with no deliveries that exist in a route and remove these nodes from that particular scenario. We then update the solutions by solving a TSP for each of these routes.

2.5.3 Customer Level-Single Period

In this section, we present the results for the service levels at the customer level over a single period. Our analysis utilizes separate tables for each of the four service levels and there are also two separate tables corresponding to datasets $\mathscr{S}$ and $\mathscr{L}$ for each service level. The columns labeled T, K, S, and SL detail the configuration for each row, indicating the number of periods, vehicles, scenarios, and target service level (TSL), respectively. Column #INS shows the number of instances in each configuration where the

IMH successfully identified a feasible solution.

For both the $\mathscr{S}$ and $\mathscr{L}$ datasets, we compare the performance of the IMH and BC algorithms. In the IMH section, the column CPU indicates the computational time required by the IMH algorithm. In the BC section, the columns CPU, Gap, and IMH∇ represent, respectively, the total computational time of the BC algorithm (which includes the IMH computational time, as the IMH solution is provided as a warm start), the optimality gap, and the percentage improvement of the BC UB relative to the IMH solution. The IMH∇ value is calculated using the formula $\frac{\text{UB}_{\text{SPRP-AR}} - bestObjVal_{\text{IMH}}}{bestObjVal_{\text{IMH}}}$, where $\text{UB}_{\text{SPRP-AR}}$ is the UB obtained from the BC algorithm, and $\text{bestObjVal}_{\text{IMH}}$ is the UB obtained from the IMH algorithm. This comparison highlights the extent to which the BC algorithm improves the solution when initialized with the IMH solution as a warm start.

Moreover, for dataset $\mathscr{S}$, we also address the SPRP-FR problem as previously mentioned. This introduces the FR-BC section, under which we report the computational time and optimality gap when solving this problem using the BC algorithm. Column PostOpt∇ for this dataset displays the relative difference between the UB of SPRP-AR and the UB of the SPRP-FR improved by the post-optimization, which is calculated by formula $\frac{\text{UB}_{\text{SPRP-AR}} - \text{UB}_{\text{SPRP-FR(PostOpt)}}}{\text{UB}_{\text{SPRP-FR(PostOpt)}}}$. Columns LB(FR-BC)$\nabla$ and UB(FR-BC)∇ represent the LB and UB of the average relative differences in the objective function of SPRP-AR and SPRP-FR, expressed as a percentage. The value of LB(FR-BC)∇ is computed using $\frac{\text{UB}_{\text{SPRP-AR}} - \text{LB}_{\text{SPRP-FR}}}{\text{LB}_{\text{SPRP-FR}}}$, and UB(FR-BC)$\nabla$ employs $\frac{\text{LB}_{\text{SPRP-AR}} - \text{UB}_{\text{SPRP-FR}}}{\text{UB}_{\text{SPRP-FR}}}$. These two values represent a lower bound and upper bound on the potential improvement obtained by doing adaptive routing instead of fixed routing. It is important to note that if the value of LB(FR-BC)∇ is positive, we set it to zero, as it can occur due to a weak LB of the SPRP-FR or a weak UB for the SPRP-AR.

Table 2.2 presents the summarized results for dataset $\mathscr{S}$ considering $\alpha_c^{customer}$, where the service level is applied individually to each customer, and the TSL is consistently enforced on every period. The IMH successfully identified feasible solutions for all cases within this dataset. The average optimality gap of the BC solution was 8.9%, and the BC algorithm improved the UB by 0.4%. This gap is typically larger for the α and β

Table 2.2: Summary of the results for the $\alpha_c^{customer}$ service level (on dataset $\mathscr{S}$)

					IMH	BC			FR-BC				
T	K	S	SL	#INS	CPU (secs)	CPU (secs)	Gap (%)	IMH∇ (%)	CPU (secs)	Gap (%)	PostOpt∇ (%)	LB(FR-BC)∇ (%)	UB(FR-BC)∇ (%)
3	1	100	70%	6	798.4	7,753.3	4.7	-0.2	4,179.3	0.8	-4.6	-14.5	-19.3
			75%	6	748.4	7,953.1	4.9	-0.2	1,832.7	0.0	-4.1	-12.4	-16.8
			80%	6	928.1	8,130.1	5.1	-0.4	4,907.5	1.9	-10.7	-16.1	-21.9
			85%	6	1,557.3	8,759.9	5.6	-0.2	7,204.5	7.3	-12.2	-10.3	-21.7
			90%	6	1,120.2	8,321.9	8.1	-0.1	3,729.9	0.0	-9.3	-12.9	-19.9
			95%	6	476.5	7,681.1	5.0	-0.1	2,587.3	0.0	-9.0	-11.0	-15.4
3	2	100	70%	6	2,477.1	9,679.3	7.2	-0.3	5,202.3	5.9	-6.1	-10.4	-21.8
			75%	6	2,372.2	9,574.5	7.0	-0.5	4,994.6	4.8	-5.7	-9.2	-19.6
			80%	6	2,803.4	10,005.9	7.9	-0.5	5,336.2	11.0	-12.3	-9.9	-26.5
			85%	6	3,648.3	10,851.1	8.4	-0.9	6,875.4	18.7	-19.7	-6.4	-30.5
			90%	6	2,825.7	10,028.0	10.0	-0.2	5,642.8	6.5	-10.2	-8.1	-22.8
			95%	6	2,775.7	9,978.4	7.6	0.0	6,050.0	5.1	-9.5	-6.9	-18.5
3	3	100	70%	6	3,386.8	10,589.4	12.3	-0.8	7,201.4	11.3[2]	-9.4	-7.3	-26.8
			75%	6	3,404.4	10,607.0	12.3	-0.8	7,202.2	15.0[1]	-10.2	-4.0	-27.8
			80%	6	3,657.0	10,859.4	12.6	-1.1	7,202.1	17.8[1]	-13.1	-3.3	-30.2
			85%	6	3,887.6	11,090.0	14.3	-0.1	7,202.3	24.1[2]	-17.9	-2.1	-34.4
			90%	6	3,800.2	11,002.5	14.6	-0.3	7,200.8	14.1[1]	-12.6	-2.5	-28.0
			95%	6	3,907.9	11,110.5	12.2	-0.1	7,202.8	13.3	-13.0	-2.8	-25.0
Total				108	2,476.4	9,665.3	8.9	-0.4	5,545.7	8.2	-10.4	-8.6	-23.3

The number inside [-] indicates the number of instances where the BC could not find a feasible solution for the SPRP-FR

service levels due to the higher number of binary variables, making it more challenging for the BC to close the gap. In contrast, for the other two service levels, the BC achieved a narrower gap.

The BC algorithm for the SPRP-FR yielded an average optimality gap of 8.2%. The total average of LB(FR-BC)∇ and UB(FR-BC)∇ indicate a potential improvement range of 8.6% to 23.3% upon implementation of adaptive routing, which highlights the benefits of tackling this more flexible problem. After applying the post-optimization we can still observe a significant difference, averaging 10.4% between the two problems, i.e., the solution found by BC for the AR problem is 10.4% better than the solution obtained by solving the FR problem and then applying post-optimization. Table 2.3 shows the results for dataset $\mathscr{L}$, where the capability of the IMH becomes particularly apparent as the BC often fails to enhance the UB established by the IMH or improvements are only marginal.

Tables 2.4 and 2.5 detail the outcomes for $\beta_c^{customer}$, which poses the greatest challenge due to its complex constraints on binary variables. Consequently, the average CPU time for the IMH is the longest among all service level instances. Despite this, the IMH successfully identified feasible solutions for every instance in both the $\mathscr{S}$ and $\mathscr{L}$ datasets. Ad-

Table 2.3: Summary of the results for the $\alpha_c^{customer}$ service level (on dataset $\mathscr{L}$)

T	K	S	SL	#INS	IMH CPU (secs)	BC CPU (secs)	BC Gap (%)	BC IMH∇ (%)
6	1	100	70%	6	5,939.0	13,140.5	16.3	-0.1
			75%	6	5,969.1	13,170.7	13.5	-0.1
			80%	6	5,667.6	12,869.1	13.5	-0.1
			85%	6	6,185.7	13,387.2	14.4	0.0
			90%	6	5,880.7	13,082.6	11.9	0.0
			95%	6	5,140.3	12,341.8	10.4	0.0
6	2	100	70%	6	6,361.9	13,564.5	24.3	-0.2
			75%	6	6,193.7	13,396.5	23.0	-0.1
			80%	6	6,276.5	13,540.8	21.8	-0.2
			85%	6	5,852.9	13,055.6	23.6	-0.1
			90%	6	6,198.9	13,401.5	21.0	0.0
			95%	6	6,129.1	13,332.0	19.9	0.0
6	3	100	70%	4	6,003.6	13,205.9	26.9	-0.2
			75%	4	6,135.2	13,337.5	23.0	-0.1
			80%	4	6,750.4	13,952.8	23.0	-0.2
			85%	4	5,749.0	12,951.3	24.2	-0.2
			90%	4	6,086.9	13,289.1	22.2	-0.1
			95%	4	5,950.0	13,242.6	18.5	0.0
9	1	100	70%	6	6,298.8	13,503.1	27.8	-0.1
			75%	6	6,798.1	14,026.1	26.7	-0.1
			80%	6	6,700.7	13,903.0	27.0	-0.1
			85%	6	6,340.7	13,543.7	29.4	-0.1
			90%	6	6,185.4	13,387.6	21.8	-0.1
			95%	6	5,871.9	13,093.7	19.0	-0.1
9	2	100	70%	4	6,651.2	13,853.5	30.4	-0.1
			75%	4	6,685.0	13,887.4	30.2	-0.2
			80%	4	6,759.3	13,961.7	31.0	-0.2
			85%	3	6,697.7	13,899.4	26.0	-0.1
			90%	4	6,104.5	13,306.8	27.8	-0.2
			95%	3	6,248.3	13,450.0	23.5	-0.1
Total				154	6,170.7	13,380.4	21.9	-0.1

Table 2.4: Summary of the results for the $\beta_c^{customer}$ service level (on dataset $\mathscr{S}$)

T	K	S	SL	#INS	IMH CPU (secs)	BC CPU (secs)	BC Gap (%)	BC IMH∇ (%)	FR-BC CPU (secs)	FR-BC Gap (%)	PostOpt∇ (%)	LB(FR-BC)∇ (%)	UB(FR-BC)∇ (%)
3	1	100	70%	6	6,011.4	13,212.8	16.5	0.0	6,182.2	12.1	-5.1	-2.7	-27.4
			75%	6	5,851.2	13,053.8	14.3	0.0	5,587.7	10.0	-4.4	-1.7	-23.6
			80%	6	5,377.9	12,581.0	14.3	0.0	6,628.9	6.4	-1.6	-1.4	-18.6
			85%	6	5,674.6	12,386.5	10.8	-0.1	6,350.6	7.3	-3.0	-0.6	-15.1
			90%	6	5,839.3	13,042.5	7.3	0.0	7,202.6	11.2	-8.4	-0.2	-16.7
			95%	6	5,388.0	12,589.0	7.4	0.0	7,201.1	4.3	-4.0	-1.3	-12.0
3	2	100	70%	6	6,202.6	13,404.7	20.6	-0.2	7,202.8	22.9[2]	-12.1	-1.8	-36.9
			75%	6	6,336.0	13,543.1	18.1	-0.2	7,205.7	24.6[1]	-14.6	-1.5	-36.0
			80%	6	6,198.2	13,403.0	16.2	-0.2	7,207.3	15.0[1]	-5.9	-1.1	-25.6
			85%	6	6,466.6	13,672.8	14.1	-0.2	7,206.9	18.0[1]	-10.7	-0.6	-26.4
			90%	6	5,909.1	13,116.3	10.9	-0.1	7,203.9	21.8[1]	-15.6	0.0	-27.0
			95%	6	5,445.8	12,651.4	10.6	-0.1	7,200.8	10.3[1]	-6.3	-0.4	-17.4
3	3	100	70%	6	6,523.6	13,726.6	28.4	-0.4	7,202.3	32.6[4]	-13.4	0.0	-41.4
			75%	6	6,372.0	13,729.4	25.6	0.0	7,204.1	28.1[3]	-13.8	-0.1	-39.6
			80%	6	6,459.0	13,662.4	22.1	-0.2	7,201.7	33.8[4]	-27.0	0.0	-44.0
			85%	6	6,256.4	13,458.6	19.7	-0.1	7,203.2	22.3[3]	-11.6	0.0	-29.3
			90%	6	5,904.4	13,106.5	17.8	0.0	7,201.3	29.4[4]	-23.1	0.0	-36.2
			95%	6	4,940.5	12,143.8	17.0	-0.1	7,208.1	19.7[3]	-11.0	-0.3	-25.8
Total				108	5,953.2	13,138.0	16.2	-0.1	6,898.9	15.6	-8.8	-0.9	-25.2

The number inside [-] indicates the number of instances where the BC could not find a feasible solution for the SPRP-FR

ditionally, a larger gap is noticeable in the SPRP-FR, accompanied by a higher incidence of instances where the BC algorithm failed to find feasible solutions. The expected improvement from implementing the SPRP-AR (compared to SPRP-FR) ranges from 0.9% to 25.2%. Even after post-optimization, there remains an 8.8% difference, underscoring the value of dealing with the complexity and challenge of the adaptive routing

The observed patterns for the $\gamma_c^{customer}$ and $\delta_c^{customer}$ service levels, as shown in Tables 2.6, 2.7, 2.8, and 2.9, align with previous trends. These service levels, characterized by a smaller number of binary variables, exhibit modest improvements obtained by BC (compared to IMH) with 0.3% for $\gamma_c^{customer}$ and 0.5% for $\delta_c^{customer}$ in the smaller dataset. This difference further narrows when using the $\mathscr{L}$ dataset. The range of potential enhancement through adaptive routing (compared to fixed routing) stands at 2.2% to 12.8% for $\gamma_c^{customer}$ and 6.3% to 15.9% for $\delta_c^{customer}$. There still exists a difference of 4.0% and 6.9% between the SPRP-AR and the SPRP-FR after post-optimization, for $\gamma_c^{customer}$ and $\delta_c^{customer}$, respectively.

In Figure 2.1a, we depict the objective function (OF) values for each of the four service levels at different TSLs. As expected, setting more stringent TSL increases the objective

Table 2.5: Summary of the results for the $\beta_c^{customer}$ service level (on dataset $\mathscr{L}$)

T	K	S	SL	#INS	IMH CPU (secs)	BC CPU (secs)	BC Gap (%)	BC IMH∇ (%)
6	1	100	70%	6	6,434.5	13,636.3	32.7	-0.1
			75%	6	6,340.5	13,542.2	29.1	-0.1
			80%	6	6,561.4	13,763.2	27.8	-0.2
			85%	6	6,795.8	13,997.7	24.4	-0.1
			90%	6	6,624.8	13,826.6	20.3	0.0
			95%	6	6,272.2	13,474.2	15.9	0.0
6	2	100	70%	6	6,859.7	14,062.7	38.6	-0.1
			75%	6	6,872.7	14,075.5	35.5	-0.1
			80%	6	6,758.8	13,961.6	34.5	-0.1
			85%	6	7,137.3	14,340.7	30.9	-0.1
			90%	6	6,905.5	14,108.5	28.2	0.0
			95%	6	7,116.9	14,319.8	23.7	0.0
6	3	100	70%	4	6,578.3	13,780.8	39.9	-0.1
			75%	4	6,703.7	13,906.1	37.4	-0.1
			80%	4	6,475.8	13,678.2	34.8	0.0
			85%	4	6,594.3	13,796.7	31.6	-0.1
			90%	4	6,865.4	14,067.7	30.1	0.0
			95%	4	6,690.8	13,893.3	26.6	-0.1
9	1	100	70%	6	6,921.3	14,124.1	42.3	-0.4
			75%	6	6,835.1	14,037.9	39.4	-0.2
			80%	6	7,128.9	14,331.6	36.1	-0.4
			85%	6	6,973.4	14,176.0	30.4	-0.3
			90%	6	7,040.9	14,243.5	25.1	-0.1
			95%	6	6,870.4	14,087.9	20.3	-0.1
9	2	100	70%	4	7,006.1	14,208.8	45.6	-0.3
			75%	4	7,162.5	14,365.4	41.9	-0.2
			80%	4	7,154.8	14,357.6	40.0	-0.2
			85%	4	7,255.1	14,458.3	35.9	-0.1
			90%	4	7,156.5	14,361.8	31.3	-0.1
			95%	4	6,870.0	14,072.7	27.9	0.0
Total				156	6,825.3	14,028.5	31.4	-0.1

function values. Interestingly, the difference between various service levels decreases as the TSL increases. It is important to note that $\gamma_c^{customer}$ represents the most strict service level, followed by $\beta_c^{customer}$ and $\delta_c^{customer}$. $\alpha_c^{customer}$, being event-based, differs fundamentally from these categories. The objective function values for $\alpha_c^{customer}$ and $\beta_c^{customer}$ are remarkably similar. Figure 2.1b displays the total average production costs, where we can observe that it is highest for $\gamma_c^{customer}$, showing the highest demand satisfaction for this service level, and lowest for $\delta_c^{customer}$.

Figures 2.1c and 2.1d display the total inventory and transportation costs, respectively. The α service level incurs higher average inventory holding costs for most TSLs compared to the other types. These higher inventory costs are attributed to the fact that $\alpha_c^{customer}$ operates on the occurrence of stockouts, which necessitates maintaining the entire demand in stock in most of the scenarios, whereas for the other service levels, demands can be met partially, allowing for inventory reduction.

In Figure 2.2, we examine further the routing costs for dataset $\mathscr{S}$, considering both the fixed routing problem and the post-optimization, along with the adaptive routing case. Each figure illustrates the OF value and average transportation costs (ATC) for a distinct TSL, encompassing the results of all three problems for various service level types. This figure highlights the substantial cost savings achieved through the implementation of adaptive routing.

Evaluating the Value of the Stochastic Solution (VSS) is a standard approach to assessing how considering uncertainty affects the objective function value. However, computing the VSS is impractical in our problem setting since using the average demand to determine first-stage variables (i.e., setup decisions) in a deterministic problem may lead to an inability to meet the required service levels in the second stage. To address this, we solve a deterministic problem using the maximum demand for each customer, then fix the first-stage variables to solve the SPRP-AR with service level constraints. Further details and analyses are provided in Appendix B.2, where we discuss the results and our proposed framework in more depth.

Table 2.6: Summary of the results for the $\gamma_c^{customer}$ service level (on dataset $\mathscr{S}$)

T	K	S	SL	#INS	IMH	BC			FR-BC				
					CPU (secs)	CPU (secs)	Gap (%)	IMH∇ (%)	CPU (secs)	Gap (%)	PostOpt∇ (%)	LB(FR-BC)∇ (%)	UB(FR-BC)∇ (%)
3	1	100	70%	6	1,211.1	8,415.0	5.4	-0.2	90.8	0.0	-3.5	-5.4	-10.5
			75%	6	1,120.2	8,324.1	5.4	-0.1	74.8	0.0	-2.5	-4.1	-9.3
			80%	6	1,002.2	8,215.7	5.0	-0.1	79.5	0.0	-2.1	-3.2	-8.0
			85%	6	1,110.7	8,318.8	4.7	-0.1	65.1	0.0	-1.1	-2.0	-6.6
			90%	6	1,477.0	8,684.4	4.0	-0.1	64.4	0.0	-0.6	-1.1	-5.1
			95%	6	1,766.2	8,971.9	3.4	-0.1	179.5	0.0	-3.0	-3.1	-6.4
3	2	100	70%	6	2,893.0	10,096.7	8.3	-0.5	2,757.7	1.4	-3.5	-3.7	-13.0
			75%	6	2,992.6	10,195.8	8.3	-0.6	3,267.8	2.2	-3.9	-2.9	-12.9
			80%	6	3,019.7	10,222.4	8.0	-0.4	3,424.5	2.2	-2.6	-1.6	-11.5
			85%	6	3,298.8	10,502.7	8.0	-0.3	4,034.0	1.1	-2.1	-1.4	-10.3
			90%	6	3,372.7	10,575.8	7.0	-0.3	4,033.1	1.6	-1.9	-0.8	-9.0
			95%	6	3,338.0	10,540.7	6.6	-0.2	3,642.3	0.7	-2.6	-2.2	-9.3
3	3	100	70%	6	3,857.0	11,059.4	11.7	-0.6	4,893.7	12.4	-8.7	-2.0	-21.9
			75%	6	3,777.3	10,982.0	11.9	-0.5	4,883.5	12.7	-9.5	-2.0	-22.1
			80%	6	3,681.7	10,888.8	12.5	-0.4	4,888.6	12.7	-7.9	-1.3	-20.7
			85%	6	3,711.5	10,341.1	12.7	-0.2	4,867.9	11.4	-5.8	-0.9	-19.0
			90%	6	3,685.1	10,889.3	12.0	-0.3	4,907.8	11.4	-6.1	-1.7	-18.8
			95%	6	3,680.5	10,886.1	11.9	-0.2	4,856.4	9.1	-4.4	-1.0	-16.3
Total				108	2,722.0	9,895.0	8.2	-0.3	2,834.0	4.4	-4.0	-2.2	-12.8

The number inside [-] indicates the number of instances where the BC could not find a feasible solution for the SPRP-FR

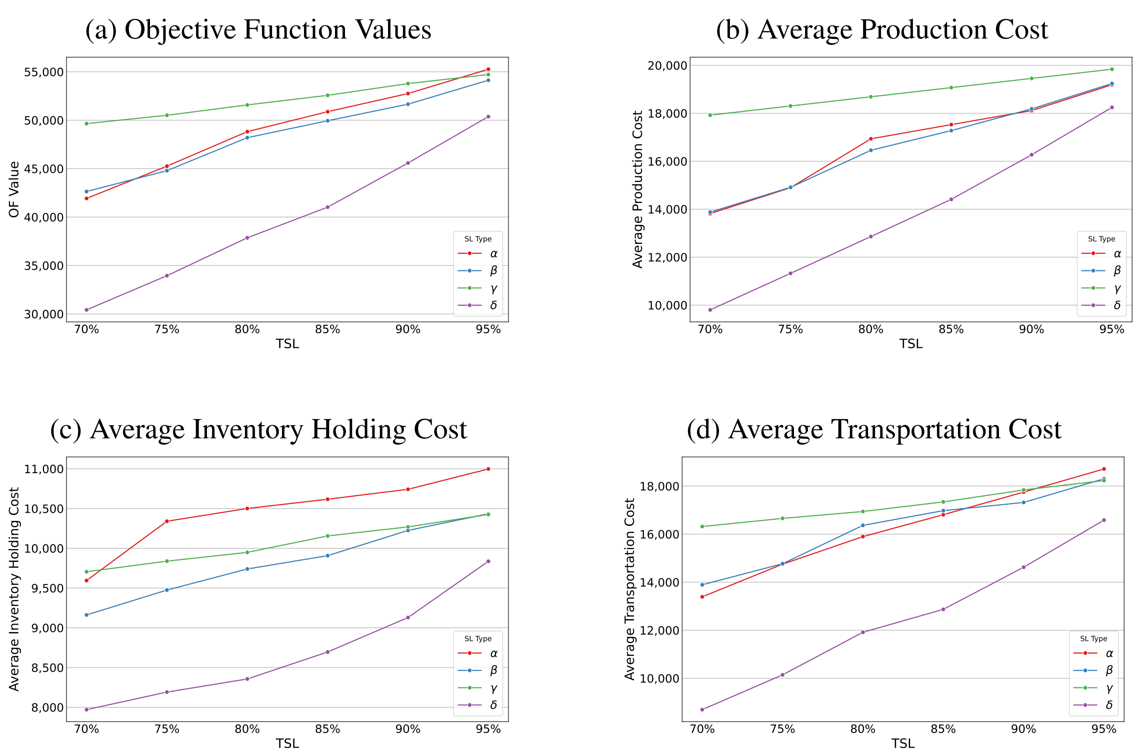


Figure 2.1: Cost comparison for different service level measures - customer level-single period

Table 2.7: Summary of the results for the $\gamma_c^{customer}$ service level (on dataset $\mathscr{L}$)

T	K	S	SL	#INS	IMH CPU (secs)	BC CPU (secs)	BC Gap (%)	BC IMH∇ (%)
6	1	100	70%	6	5,653.5	12,855.2	11.0	-0.1
			75%	6	5,443.2	12,645.1	10.4	0.0
			80%	6	5,273.1	12,476.2	10.9	0.0
			85%	6	5,443.3	12,645.2	10.2	0.0
			90%	6	5,316.4	12,517.8	9.3	0.0
			95%	6	5,267.9	12,469.4	8.0	0.0
6	2	100	70%	6	5,584.5	12,787.3	18.3	-0.1
			75%	6	5,719.7	12,922.4	17.6	-0.1
			80%	6	5,800.5	13,003.2	18.1	-0.1
			85%	6	5,593.7	12,796.3	17.4	-0.1
			90%	6	5,555.2	12,758.6	18.0	-0.1
			95%	6	5,604.1	12,809.4	16.4	-0.1
6	3	100	70%	4	5,881.5	13,089.7	19.1	-0.2
			75%	4	5,602.5	12,804.6	19.0	-0.1
			80%	4	6,064.9	13,268.4	21.3	-0.1
			85%	4	5,721.3	12,923.5	20.9	0.0
			90%	4	6,028.2	13,230.3	20.4	-0.1
			95%	4	5,797.4	13,006.4	19.9	-0.1
9	1	100	70%	6	5,830.2	13,032.2	14.4	-0.1
			75%	6	5,702.1	12,904.2	14.4	-0.1
			80%	6	5,568.7	12,784.0	14.2	-0.1
			85%	6	5,689.3	12,891.4	13.4	-0.1
			90%	6	5,739.6	12,941.7	12.8	-0.1
			95%	6	5,491.3	12,693.4	11.6	-0.1
9	2	100	70%	4	6,290.6	13,492.9	23.1	-0.2
			75%	4	5,865.7	13,068.0	22.6	-0.2
			80%	4	5,976.5	13,178.9	22.6	-0.2
			85%	4	5,614.6	12,816.9	22.7	-0.1
			90%	4	5,689.2	12,891.4	21.6	-0.2
			95%	3	5,996.7	13,198.2	18.1	-0.2
Total				155	5,663.1	12,866.3	15.9	-0.1

Table 2.8: Summary of the results for the $\delta_c^{customer}$ service level (on dataset $\mathscr{S}$)

T	K	S	SL	#INS	IMH	BC			FR-BC				
					CPU (secs)	CPU (secs)	Gap (%)	IMH∇ (%)	CPU (secs)	Gap (%)	PostOpt∇ (%)	LB(FR-BC)∇ (%)	UB(FR-BC)∇ (%)
3	1	100	70%	6	577.3	7,471.8	6.0	-0.9	19.3	0.0	-13.3	-13.6	-18.8
			75%	6	761.6	7,676.8	6.2	-0.9	30.9	0.0	-10.0	-11.5	-17.0
			80%	6	641.8	7,701.3	6.1	-0.3	35.1	0.0	-7.6	-9.3	-14.8
			85%	6	986.9	8,132.8	5.0	-0.3	34.6	0.0	-5.8	-6.9	-11.5
			90%	6	1,119.8	8,329.7	4.9	-0.2	87.1	0.0	-3.5	-5.4	-10.0
			95%	6	1,259.7	8,194.4	4.3	-0.2	69.3	0.0	-1.0	-1.8	-6.1
3	2	100	70%	6	1,634.7	8,841.6	7.7	-0.9	98.9	0.0	-11.8	-12.1	-18.9
			75%	6	2,178.1	9,384.4	8.7	-0.9	244.3	0.0	-7.5	-8.9	-16.9
			80%	6	2,458.1	9,663.4	8.5	-0.6	308.3	0.0	-5.8	-7.2	-15.1
			85%	6	2,614.1	9,823.0	8.1	-0.5	751.7	0.0	-4.3	-5.4	-13.1
			90%	6	2,849.7	10,053.5	8.0	-0.5	3,053.8	1.4	-3.4	-3.4	-12.4
			95%	6	3,165.1	10,368.2	7.6	-0.2	4,477.4	1.4	-2.0	-1.4	-10.1
3	3	100	70%	6	3,501.5	10,706.8	11.6	-0.8	486.7	0.0	-8.8	-9.3	-19.9
			75%	6	3,371.7	10,575.9	11.6	-0.5	2,631.4	2.4	-7.7	-7.1	-19.9
			80%	6	3,784.5	10,989.1	11.4	-0.4	4,056.5	4.3	-7.4	-4.6	-19.2
			85%	6	3,675.5	10,882.1	11.3	-0.4	4,297.4	6.0	-6.6	-3.5	-19.2
			90%	6	3,766.9	10,973.9	11.6	-0.5	4,872.0	14.4	-11.4	-1.9	-23.8
			95%	6	3,686.6	10,891.9	12.2	-0.2	4,855.8	12.4	-6.8	-0.9	-19.6
Total				108	2,335.2	9,481.1	8.4	-0.5	1,689.5	2.4	-6.9	-6.3	-15.9

The number inside [-] indicates the number of instances where the BC could not find a feasible solution for the SPRP-FR

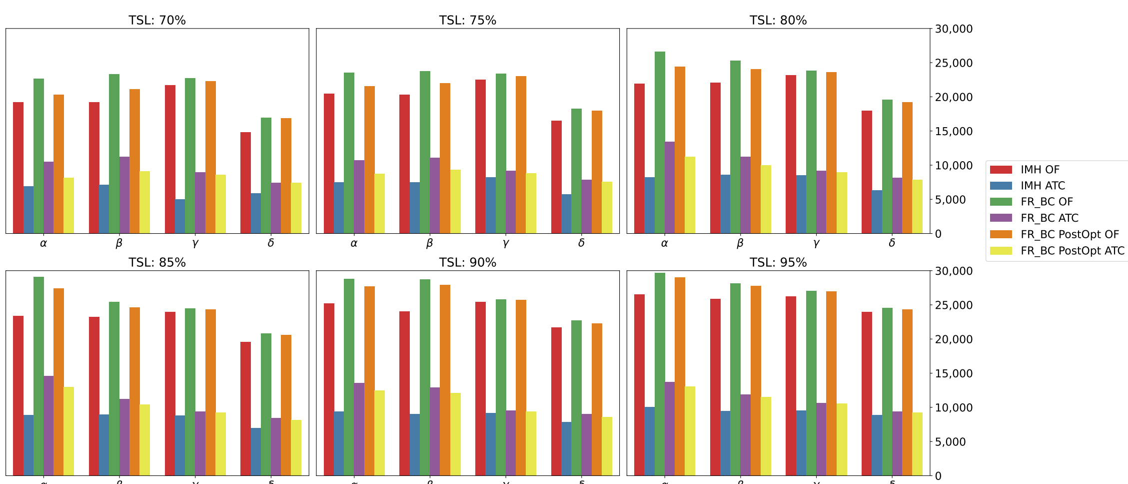


Figure 2.2: Routing costs comparison for different service level measures and algorithms (on dataset $\mathscr{S}$) - customer level-single period

Table 2.9: Summary of the results for the $\delta_c^{customer}$ service level (on dataset $\mathscr{L}$)

T	K	S	SL	#INS	IMH CPU (secs)	BC CPU (secs)	BC Gap (%)	BC IMH∇ (%)
6	1	100	70%	6	4,207.9	11,410.2	10.9	-0.1
			75%	6	4,527.1	11,728.9	12.9	0.0
			80%	6	4,698.8	11,900.5	11.2	-0.1
			85%	6	5,283.2	12,484.9	11.4	-0.1
			90%	6	5,090.6	12,291.9	11.4	0.0
			95%	6	5,303.8	12,505.2	9.8	0.0
6	2	100	70%	6	4,875.2	12,078.0	16.0	-0.7
			75%	6	5,301.0	12,503.8	17.3	-0.5
			80%	6	5,251.2	12,453.7	16.8	-0.4
			85%	6	5,354.6	12,561.3	17.5	-0.4
			90%	6	5,594.8	12,797.6	18.1	-0.1
			95%	6	5,518.1	12,720.5	16.6	-0.1
6	3	100	70%	4	5,452.2	12,656.7	16.7	-0.4
			75%	4	5,713.9	12,916.1	19.4	-0.2
			80%	4	5,649.4	12,861.6	28.1	0.0
			85%	4	5,659.6	12,867.4	18.9	-0.2
			90%	4	5,790.1	12,992.3	21.0	-0.2
			95%	4	5,573.3	12,775.5	19.3	-0.1
9	1	100	70%	6	5,212.8	12,414.9	15.4	0.0
			75%	6	5,136.0	12,338.1	15.2	-0.1
			80%	6	5,470.2	12,672.3	14.4	-0.1
			85%	6	5,526.1	13,059.5	15.3	-0.1
			90%	6	5,395.1	12,597.2	15.2	-0.1
			95%	6	5,593.2	12,795.2	14.2	-0.1
9	2	100	70%	4	5,899.7	13,101.9	19.9	-0.1
			75%	4	5,947.0	13,149.1	20.8	-0.2
			80%	4	6,055.1	13,257.6	21.2	-0.2
			85%	4	5,727.8	12,930.0	21.6	-0.3
			90%	4	6,455.9	13,658.2	22.1	-0.3
			95%	4	5,886.9	13,089.1	23.0	-0.3
Total				156	5,380.0	12,592.5	16.5	-0.2

2.5.4 Customer Level-Global

In this section, we summarize the results for the case where service levels are considered at the customer level over the entire planning horizon rather than for every period. Table 2.10 presents the summary for dataset $\mathscr{S}$, while Table 2.11 displays a summary of the results for dataset $\mathscr{L}$. Detailed tables are provided in the online Appendix, Section 1.

For dataset $\mathscr{S}$, the average CPU time for the IMH ranges between 40 to 110 minutes, with most instances completing in less than 60 minutes except for $\beta^{customer}$. While the IMH found a feasible solution for all instances, the BC algorithm failed to find a feasible solution for the SPRP-FR in 8 instances for $\alpha_p^{customer}$, 15 instances for $\beta^{customer}$, and 1 instance for $\gamma^{customer}$. The BC algorithm improved the objective function of the $\mathscr{S}$ instances obtained by IMH by 0.8% for $\alpha_p^{customer}$, 0.4% for $\beta^{customer}$, 0.6% for $\gamma^{customer}$, and 1.1% for $\delta^{customer}$, respectively. The lowest minimum expected improvement obtained from solving the SPRP-AR instead of SPRP-FR is for $\beta^{customer}$ with an average of 1.4%, while the highest minimum expected improvement is for $\delta^{customer}$ with an average of 13.7%. For dataset $\mathscr{L}$, the IMH was also able to solve the problem for all instances, with the average improvement obtained by the BC algorithm not exceeding 0.8%.

Figure 2.3 displays the OF value for different TSLs. It can be observed that, in comparison to Figure 2.1a, the OF values for $\gamma^{customer}$ and $\beta^{customer}$ exhibit less difference. However, for $\delta^{customer}$, the values do not converge to those of other service levels even at higher TSLs. Additional details on the CPU time for different phases of the IMH algorithm are provided in Appendix B.3.

Table 2.10: Summary of the results for different service levels for customer level-global on the $\mathscr{S}$ dataset

SL Type	T	K	S	#INS	IMH CPU (secs)	BC CPU (secs)	BC Gap (%)	BC IMH∇ (%)	FR-BC CPU (secs)	FR-BC Gap (%)	PostOpt∇ (%)	LB(FR-BC)∇ (%)	UB(FR-BC)∇ (%)
$\alpha_p^{customer}$	3	1	100	36	1,167.9	8,327.4	5.8	-0.4	6,587.5	1.4	-5.4	-6.8	-13.5
		2	100	36	2,465.3	9,669.4	7.7	-1.1	7,195.7	9.7	-9.5	-3.5	-19.5
		3	100	36	3,647.6	10,851.9	10.9	-1.0	6,854.1	15.2[8]	-11.2	-1.9	-24.4
Total				108	2,426.9	9,616.3	8.1	-0.8	6,881.1	8.3	-8.5	-4.2	-18.7
$\beta^{customer}$	3	1	100	36	5,955.4	13,052.9	10.6	-0.1	6,359.1	6.9	-6.9	-2.0	-17.8
		2	100	36	6,187.1	13,353.8	14.0	-0.6	6,959.3	16.3[2]	-10.9	-1.3	-25.8
		3	100	36	6,465.6	13,672.9	18.9	-0.6	7,135.0	22.5[13]	-15.2	-0.6	-33.1
Total				108	6,202.7	13,359.9	14.5	-0.4	6,770.4	14.2	-10.4	-1.4	-24.5
$\gamma^{customer}$	3	1	100	36	2,330.7	9,330.8	3.6	-0.1	138.9	0.0	-1.5	-3.0	-6.5
		2	100	36	2,994.5	10,136.6	7.2	-0.9	1,706.6	0.9	-2.1	-2.7	-10.4
		3	100	36	3,760.2	10,962.0	10.3	-0.8	4,038.3	6.3[1]	-5.2	-2.3	-16.7
Total				108	3,028.5	10,143.1	7.0	-0.6	1,958.9	2.4	-2.9	-2.7	-11.2
$\delta^{customer}$	3	1	100	36	2,796.1	9,864.2	6.7	-0.9	156.0	0.0	-11.5	-15.1	-20.6
		2	100	36	3,078.1	10,212.6	12.0	-1.2	1,375.4	0.1	-10.6	-13.9	-23.8
		3	100	36	3,302.0	10,473.2	14.9	-1.1	3,466.1	5.7	-12.1	-12.0	-28.4
Total				108	3,058.7	10,183.3	11.2	-1.1	1,665.8	2.0	-11.4	-13.7	-24.3

The number inside [-] indicates the number of instances where the BC could not find a feasible solution for the SPRP-FR

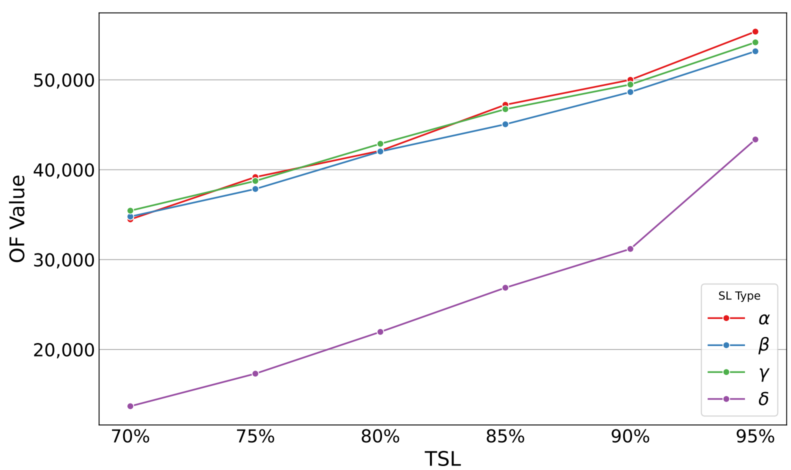


Figure 2.3: Objective function values for different service level measures - customer level-global

Table 2.11: Summary of the results for different service levels for customer level-global on the $\mathscr{L}$ dataset

SL Type	T	K	S	#INS	IMH CPU (secs)	BC CPU (secs)	BC Gap (%)	BC IMH∇ (%)
$\alpha_p^{customer}$	6	1	100	36	6,241.2	13,445.5	19.2	-0.1
		2	100	36	6,295.3	13,498.1	32.5	-0.1
		3	100	24	6,208.6	13,410.9	34.7	-0.2
	9	1	100	36	6,009.6	13,260.9	38.8	-0.5
		2	100	23	6,280.0	13,522.2	47.2	-0.2
Total				155	6,200.7	13,421.3	33.3	-0.2
$\beta^{customer}$	6	1	100	36	6,117.3	13,312.3	25.0	-0.1
		2	100	36	6,316.0	13,519.0	33.9	-0.3
		3	100	24	6,683.7	13,886.5	37.8	-0.5
	9	1	100	36	6,337.2	13,539.9	41.0	-0.5
		2	100	24	6,849.4	14,052.3	49.1	-0.4
Total				156	6,413.7	13,614.7	36.4	-0.3
$\gamma^{customer}$	6	1	100	36	6,038.5	13,240.9	8.1	0.0
		2	100	36	6,369.1	13,572.0	14.5	-0.4
		3	100	24	6,509.0	13,711.8	18.7	-0.5
	9	1	100	36	5,990.8	13,193.8	9.9	-0.1
		2	100	24	6,176.2	13,378.4	19.3	-0.3
Total				156	6,197.4	13,400.0	13.4	-0.3
$\delta^{customer}$	6	1	100	36	5,957.8	13,159.7	6.3	-0.3
		2	100	36	6,207.6	13,409.6	12.8	-0.8
		3	100	24	6,035.3	13,238.2	16.2	-0.7
	9	1	100	36	5,708.7	12,910.6	12.9	-0.1
		2	100	24	5,703.4	12,906.5	19.2	-0.4
Total				156	5,930.7	13,133.0	13.1	-0.4

2.5.5 Plant Level-Single Period

We now shift our focus to studying service levels at the plant level, where relevant constraints are applied to all customers in an aggregate way. Specifically, in terms of demand satisfaction, each customer is no longer considered separately; thus, all customers are treated as a single entity. This section presents the results for the plant level, where service levels are imposed for each period separately. Table 2.12 summarizes the results for all service levels for this problem using dataset $\mathscr{S}$, while Table 2.13 illustrates the results for dataset $\mathscr{L}$. Detailed results are provided in the online Appendix, Section 2.

From Tables 2.12 and 2.13, we can observe that the IMH was able to solve the problem for all instances, with the highest average improvement obtained from the BC algorithm being 1.0%, while for most cases, it was below 0.5%. However, the BC algorithm was not able to find a feasible solution for the SPRP-AR for 42 instances out of 432 instances of dataset $\mathscr{S}$.

In Figure 2.4, it is evident that, similar to the results from Section 2.5.3, the discrepancy among different service levels decreases as the TSLs rise. However, in this case, the objective function value for α_c^{plant} is notably higher than that for β_c^{plant}. This can be attributed mainly to the fact that experiencing a stockout in even one of the customers affects the satisfaction of the α_c^{plant} service level at the plant level, whereas for other types of service levels, it is still possible to partially meet the demand without compromising the service levels.

Table 2.12: Summary of the results for different service levels for plant level-single period on the $\mathscr{S}$ dataset

SL Type	T	K	S	#INS	IMH CPU (secs)	BC CPU (secs)	BC Gap (%)	BC IMH∇ (%)	FR-BC CPU (secs)	FR-BC Gap (%)	PostOpt∇ (%)	LB(FR-BC)∇ (%)	UB(FR-BC)∇ (%)
α_c^{plant}	3	1	100	36	177.4	6,643.2	2.7	-0.1	1,600.9	0.2	-7.8	-15.1	-17.6
		2	100	36	2,320.9	9,322.3	6.0	-0.1	5,124.7	5.8	-9.6	-11.8	-22.1
		3	100	36	3,694.6	10,900.4	9.3	-0.1	5,305.1	13.1[4]	-13.5	-8.4	-28.2
Total				108	2,064.3	8,955.3	6.0	-0.1	3,960.4	6.1	-10.2	-11.9	-22.4
β_c^{plant}	3	1	100	36	4,916.0	11,868.2	10.7	0.0	7,202.5	13.8[3]	-8.8	-0.5	-20.0
		2	100	36	4,737.9	11,940.0	15.4	-0.4	7,039.4	15.2[15]	-9.7	-0.7	-24.0
		3	100	36	5,000.5	12,136.9	22.0	-0.6	7,068.5	24.4[19]	-14.8	-0.2	-32.8
Total				108	4,884.8	11,981.7	16.0	-0.3	7,122.2	16.8	-10.5	-0.4	-24.2
γ_c^{plant}	3	1	100	36	1,326.3	8,419.2	3.6	-0.3	185.4	0.0	-1.4	-1.4	-4.5
		2	100	36	3,108.8	10,166.8	9.3	-0.3	3,800.8	2.9	-2.9	-1.5	-11.9
		3	100	36	3,797.2	10,825.9	14.6	-0.5	4,862.3	11.1[1]	-5.8	-1.1	-19.6
Total				108	2,744.1	9,803.9	9.2	-0.4	2,931.6	4.6	-3.4	-1.3	-11.9
δ_c^{plant}	3	1	100	36	723.7	7,445.3	7.9	-0.8	627.4	0.0	-1.8	-1.8	-8.9
		2	100	36	2,403.1	9,496.7	12.6	-1.5	3,277.5	1.6	-2.2	-1.6	-13.9
		3	100	36	3,934.6	10,901.7	16.8	-0.6	4,309.8	5.8	-2.4	-1.0	-17.5
Total				108	2,353.8	9,298.4	12.5	-1.0	2,738.3	2.5	-2.1	-1.5	-13.4

The number inside [-] indicates the number of instances where the BC could not find a feasible solution for the SPRP-FR

Table 2.13: Summary of the results for different service levels for plant level-single period on the $\mathscr{L}$ dataset

SL Type	T	K	S	#INS	IMH CPU (secs)	BC CPU (secs)	BC Gap (%)	BC IMH∇ (%)
α_c^{plant}	6	1	100	36	5,181.7	12,386.5	10.4	0.0
		2	100	36	5,471.1	12,674.5	21.1	-0.2
		3	100	24	5,860.4	13,064.0	23.6	-0.2
	9	1	100	36	5,747.8	13,110.9	21.5	-0.1
		2	100	24	5,703.4	13,154.1	28.4	-0.2
Total				156	5,563.8	12,836.6	20.1	-0.1
$\beta_c^{customer}$	6	1	100	36	6,502.2	13,704.3	24.3	-0.2
		2	100	36	6,399.8	13,602.7	30.6	-0.1
		3	100	24	6,633.0	13,835.4	31.8	-0.1
	9	1	100	36	6,669.1	13,872.0	27.8	-0.4
		2	100	24	6,660.0	13,885.2	32.7	-0.1
Total				156	6,561.5	13,766.0	29.0	-0.2
$\gamma_c^{customer}$	6	1	100	36	5,351.3	12,511.8	9.5	0.0
		2	100	36	5,578.7	12,781.5	15.7	-0.1
		3	100	24	5,887.8	13,062.6	19.2	-0.3
	9	1	100	36	6,027.6	13,230.8	11.0	-0.1
		2	100	24	5,926.1	13,146.7	19.8	-0.2
Total				156	5,730.8	12,920.9	14.3	-0.1
$\delta_c^{customer}$	6	1	100	36	4,579.2	11,613.2	9.1	-0.6
		2	100	36	5,497.6	12,700.8	13.5	-0.8
		3	100	24	5,750.5	12,860.6	14.8	-0.9
	9	1	100	36	5,270.1	12,473.1	11.6	-0.3
		2	100	24	5,870.2	13,072.7	18.1	-0.4
Total				156	5,329.4	12,479.1	13.0	-0.6

2.5.6 Plant Level-Global

Following the previous sections, our last analysis focuses on the plant granularity level while considering the service levels across the entire planning horizon. We once again present two summary tables, Table 2.14 and Table 2.15, to display our results on the $\mathscr{S}$ and $\mathscr{L}$ datasets, respectively. For detailed results, we refer the reader to the online Appendix, Section 3.

The average computation time mostly ranges between 30 to 50 minutes for α_p^{plant}, γ^{plant}, and δ^{plant}, while it is notably higher for β^{plant}, averaging around 90 minutes, as

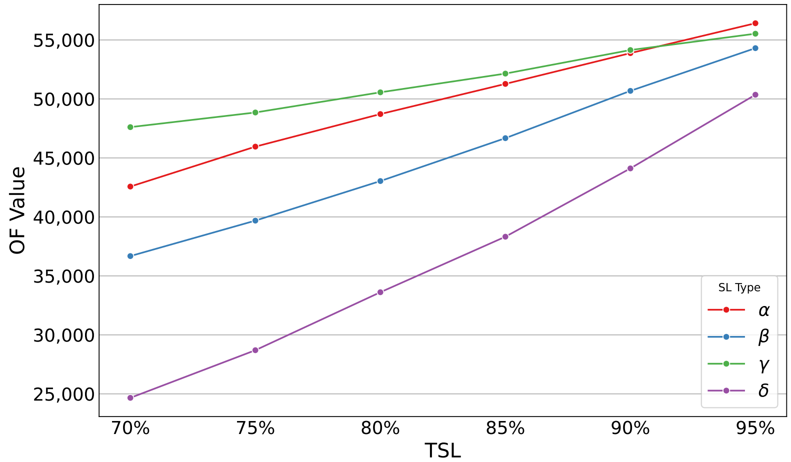


Figure 2.4: Objective function values for different service levels - plant level-single period

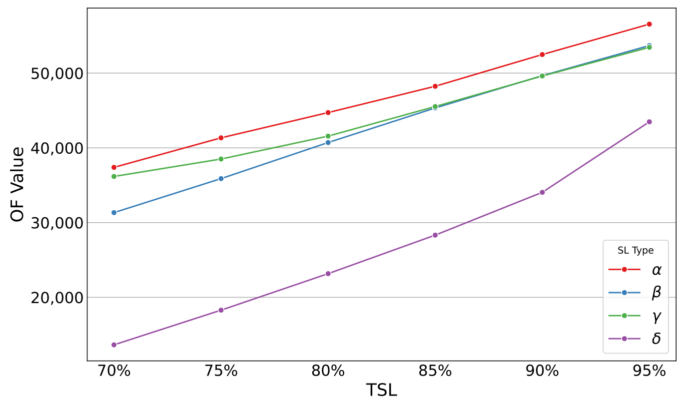


Figure 2.5: Objective function values for different service level measures - plant level-global

it represents the most challenging service level type in our comparison. Additionally, it is observed that the BC algorithm was unable to solve the SPRP-FR for 22 instances of β^{plant}. In dataset $\mathscr{S}$, the average improvement obtained by BC ranges from 0.3% to 1.6%, while the average optimality gap ranges between 5.4% to 17.2%.

The upper bound obtained by AR is still 1.4% to 7.9% better than the upper bound obtained by FR and post-optimization. Additionally, it is noted that the minimum expected improvements by solving the adaptive routing problem could range from 0.5% to 4.7% for different service levels. We also provide the OF value of different service levels for various TSLs in Figure 2.5.

Table 2.14: Summary of the results for different service levels for plant level-global on the $\mathscr{S}$ dataset

SL Type	T	K	S	#INS	IMH CPU (secs)	BC CPU (secs)	BC Gap (%)	BC IMH∇ (%)	FR-BC CPU (secs)	FR-BC Gap (%)	PostOpt∇ (%)	LB(FR-BC)∇ (%)	UB(FR-BC)∇ (%)
α_p^{plant}	3	1	100	36	411.6	7,466.1	2.4	-0.7	404.7	0.0	-3.0	-5.9	-8.2
		2	100	36	2,016.3	8,766.7	5.2	-0.2	4,888.1	4.7	-6.8	-5.0	-12.8
		3	100	36	3,683.9	10,890.1	8.5	-0.1	5,634.2	14.7	-13.8	-3.1	-24.1
Total				108	2,037.3	9,051.5	5.4	-0.3	3,642.3	6.5	-7.9	-4.7	-15.0
β^{plant}	3	1	100	36	5,287.3	12,129.9	12.8	-0.3	7,064.0	6.6	-3.1	-0.9	-15.5
		2	100	36	5,348.6	12,102.7	16.8	-0.9	6,962.7	14.2[6]	-6.9	-0.3	-23.2
		3	100	36	5,584.7	12,727.0	22.0	-0.7	6,984.3	15.7[16]	-8.0	-0.2	-27.1
Total				108	5,406.9	12,319.9	17.2	-0.7	7,010.2	11.4	-5.6	-0.5	-20.9
γ^{plant}	3	1	100	36	1,740.0	8,672.0	3.9	-0.8	172.9	0.0	-0.8	-0.9	-4.6
		2	100	36	2,499.1	9,454.8	9.0	-1.2	1,867.8	0.3	-0.9	-0.7	-9.5
		3	100	36	3,517.5	10,438.5	11.8	-0.9	4,078.2	3.7	-2.5	-0.9	-13.9
Total				108	2,585.5	9,529.7	8.3	-1.0	2,039.7	1.3	-1.4	-0.8	-9.3
δ^{plant}	3	1	100	36	2,405.6	9,348.4	5.4	-1.3	208.3	0.0	-2.2	-2.2	-7.5
		2	100	36	2,815.2	10,021.3	12.5	-2.3	1,560.4	0.2	-2.1	-2.0	-14.3
		3	100	36	3,416.9	10,396.4	16.2	-1.3	3,317.9	1.6	-2.6	-2.0	-18.3
Total				108	2,879.2	9,922.0	11.4	-1.6	1,695.5	0.6	-2.3	-2.1	-13.4

The number inside [-] indicates the number of instances where the BC could not find a feasible solution for the SPRP-FR

2.5.7 Insights

In this section, we provide additional insights into our experiments by offering a further comparison of the results among different service levels. Moreover, we conduct two more sensitivity analyses in this section, where we present the results for different numbers of scenarios and provide an analysis of how uncertainty levels affect the solutions.

Different Granularity Level

In this section, we compare the results in terms of different granularity levels. Figure 2.6 illustrates the comparison of the average OF value for each service level across various granularity levels. It is evident that, across all cases, the plant level-global scenario exhibits significantly lower costs compared to all other cases. This difference can be attributed to the increased flexibility at this granularity level, where the service level is imposed from the plant perspective over the entire planning horizon. This aggregation allows for less strict demand satisfaction at each customer, resulting in lower OF values.

The customer level-global case follows as the second lowest in cost, offering less flexibility compared to the plant level-global. In this case, each customer must meet a specific

Table 2.15: Summary of the results for different service levels for plant level-global period on the $\mathscr{L}$ dataset

SL Type	T	K	S	#INS	IMH CPU (secs)	BC CPU (secs)	BC Gap (%)	BC IMH∇ (%)
α_p^{plant}	6	1	100	36	6,057.8	13,260.8	13.5	-0.1
		2	100	36	6,318.0	13,520.9	26.6	0.0
		3	100	24	6,510.7	13,713.8	30.4	-0.1
	9	1	100	36	6,125.3	13,484.2	33.1	-0.1
		2	100	24	6,308.7	13,511.0	42.3	-0.2
Total				156	6,241.7	13,480.6	28.1	-0.1
$\beta^{customer}$	6	1	100	36	6,194.0	13,398.1	27.0	-1.1
		2	100	36	6,467.3	13,670.1	35.4	-0.6
		3	100	24	6,606.1	13,808.3	38.7	-0.3
	9	1	100	36	6,498.4	13,701.1	40.0	-0.4
		2	100	24	6,875.3	14,077.9	49.3	-0.5
Total				156	6,495.5	13,698.5	37.2	-0.6
$\gamma^{customer}$	6	1	100	36	5,875.3	12,970.8	5.5	0.0
		2	100	36	6,438.4	13,641.7	11.8	-0.5
		3	100	24	6,177.1	13,379.5	16.1	-0.9
	9	1	100	36	6,076.0	13,279.0	8.9	-0.1
		2	100	24	6,141.7	13,343.9	18.5	-0.4
Total				156	6,139.0	13,317.0	11.4	-0.4
$\delta^{customer}$	6	1	100	36	5,319.9	11,874.1	6.6	-0.6
		2	100	36	5,488.2	12,093.3	11.3	-0.9
		3	100	24	5,455.2	12,059.3	14.1	-0.7
	9	1	100	36	6,073.6	13,276.0	7.5	-0.3
		2	100	24	6,317.8	13,528.7	14.3	-0.7
Total				156	5,707.0	12,524.8	10.2	-0.6

service level throughout its entire horizon, resulting in higher OF values. However, it is noteworthy that the OF value of the customer level-global scenario remains lower than the plant level-single period case, where aggregation is solely on the customers while imposing the SL in each time period.

Furthermore, setting service levels for each period instead of on the whole horizon imposes a higher satisfaction rate and reduces flexibility. Consequently, the per-period cases exhibit higher costs compared to the global scenarios. When considering the granularity level per period between customer and plant levels, it is important to note that imposing service levels for every customer results in higher costs. However, the difference in average OF values between the per-period plant and customer granularity levels tends to decrease for higher TSLs.

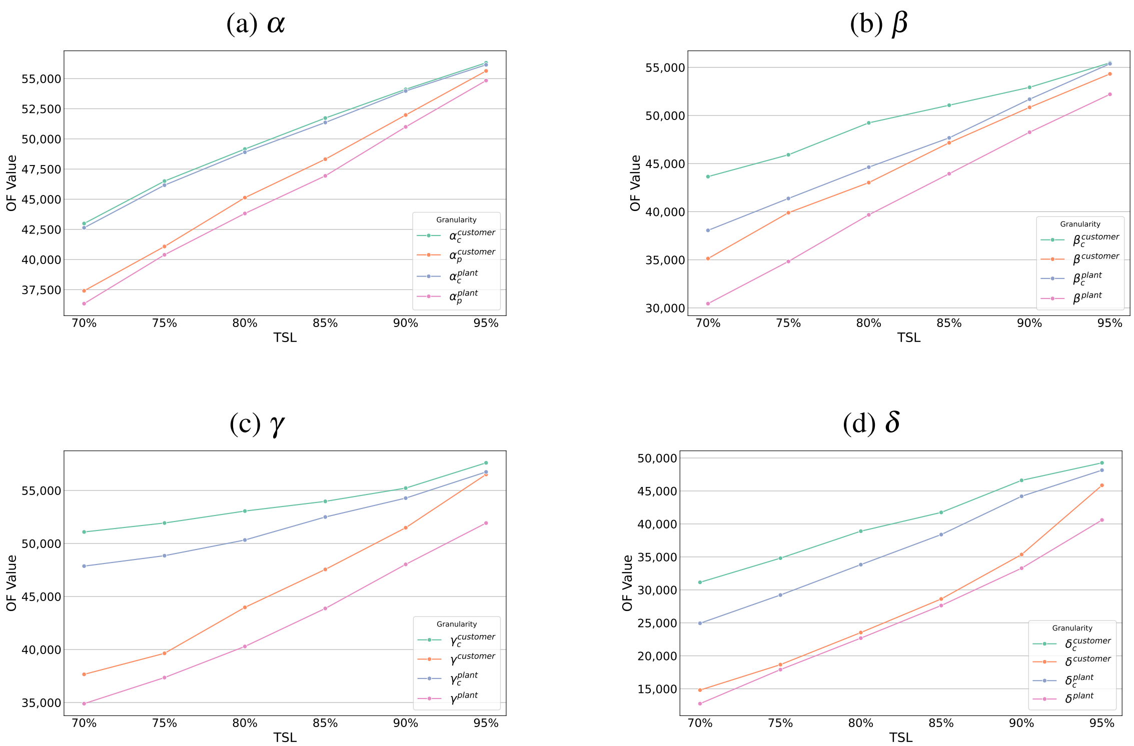


Figure 2.6: Objective function values for different service levels and granularity levels

Different Numbers of Scenarios

We analyze how considering different numbers of scenarios can affect the problem complexity and the computation time. In Table 2.16, we present the results for the customer level-single period case under different numbers of scenarios, including 20, 40, 60, 80, 100, 200, and 500 scenarios. We solve instances from both the $\mathscr{S}$ and $\mathscr{L}$ datasets. This allows us to observe how increasing the number of scenarios affects the solutions. We performed experiments on 264 instances for each number of scenarios, totalling 1848 instances per each type of service level.

As expected the BC gap increased when we solved the problem with a higher number of scenarios. The IMH ability to solve instances also decreases for higher number of scenarios, however, all instances have been solved for instances with up to 200 scenarios and also more than 80% of instances were solved when $S = 500$. It is also evident that the improvement obtained by the BC remains relatively low throughout the entire experiment, where the highest is for $\delta_c^{customer}$ with an average of 0.4%. These results shows the ability of the IMH algorithm even for large instances with 500 scenarios and even for the complex β service level.

Different Uncertainty Level

In this section, we explore how increased uncertainty impacts the solutions to our problem. To do this, we adjust the value of ε, which influences the range of stochastic demands. We set ε at 0.2, 0.4, 0.6, and 0.8 and solved the problem for the customer level-single period case. The experiments were conducted using both the $\mathscr{S}$ and $\mathscr{L}$ datasets. Figure 2.7 displays the average values of the OF for these settings.

Interestingly, the average OF values for α service level decrease for higher uncertainty levels, as shown in Figure 2.7a. Although this may seem counter-intuitive at first glance, a closer examination of this metric's nature reveals the reason. As α service level is an event-based service level, and the problem goal is to minimize total costs, the model primarily satisfies scenarios with lower demand. Consequently, the average demand satisfaction decreases as uncertainty increases. Specifically, with a broader range of demand, the model tends to prioritize scenarios with lower demand, often neglecting scenarios with higher demands. Therefore, it satisfies a smaller portion of the demand, leading to a decrease in the OF value. This could be viewed as a drawback of the α service level in practical applications, as it overlooks higher demand instances. In contrast, for the other three service levels, the OF value behaves as anticipated, where a more uncertain environment results in increased costs.

Table 2.16: Summary of the results for different service levels for customer level-single period under different number of scenarios

SL Type	S	#INS	IMH CPU (secs)	BC CPU (secs)	BC Gap (%)	BC IMH∇ (%)
$\alpha_c^{customer}$	20	264	3737.1	7039.6	10.2	-0.4
	40	264	4088.7	7178.8	12.5	-0.3
	60	264	4426.1	7197.6	14.5	-0.3
	80	264	4521.2	7189.7	16.2	-0.3
	100	264	4656.8	7199.2	16.5	-0.2
	200	264	5024.3	7205.1	20.8	-0.1
	500	224	5798.9	7229.4	28.0	-0.1
Total		1808	4581.2	7174.1	16.4	-0.3
$\beta_c^{customer}$	20	264	5215.7	7097.4	16.1	-0.2
	40	264	5712.9	7127.5	19.4	-0.2
	60	264	5995.6	7197.9	21.4	-0.2
	80	264	6312.9	7170.1	23.7	-0.2
	100	264	6468.5	7195.7	25.2	-0.1
	200	264	6607.0	7206.4	29.7	-0.1
	500	221	6493.7	7207.3	37.2	-0.1
Total		1804	6106.1	7169.4	23.9	-0.1
$\gamma_c^{customer}$	20	264	3903.6	7161.5	8.4	-0.4
	40	264	4134.9	7195.2	10.0	-0.3
	60	264	4326.5	7203.2	11.1	-0.3
	80	264	4387.9	7188.8	11.9	-0.2
	100	264	4459.2	7190.8	12.7	-0.2
	200	264	4720.8	7186.9	13.9	-0.1
	500	218	5118.3	7206.1	16.4	0.0
Total		1802	4418.5	7189.6	11.8	-0.2
$\delta_c^{customer}$	20	264	3338.8	7168.2	8.8	-0.8
	40	264	3701.4	7195.9	10.4	-0.5
	60	264	3949.8	7174.7	11.5	-0.5
	80	264	4085.3	7146.3	12.3	-0.4
	100	264	4134.4	7179.4	13.1	-0.3
	200	264	4559.4	7195.8	14.9	-0.2
	500	238	5002.1	7226.3	18.7	0.0
Total		1822	4097.4	7182.8	12.7	-0.4

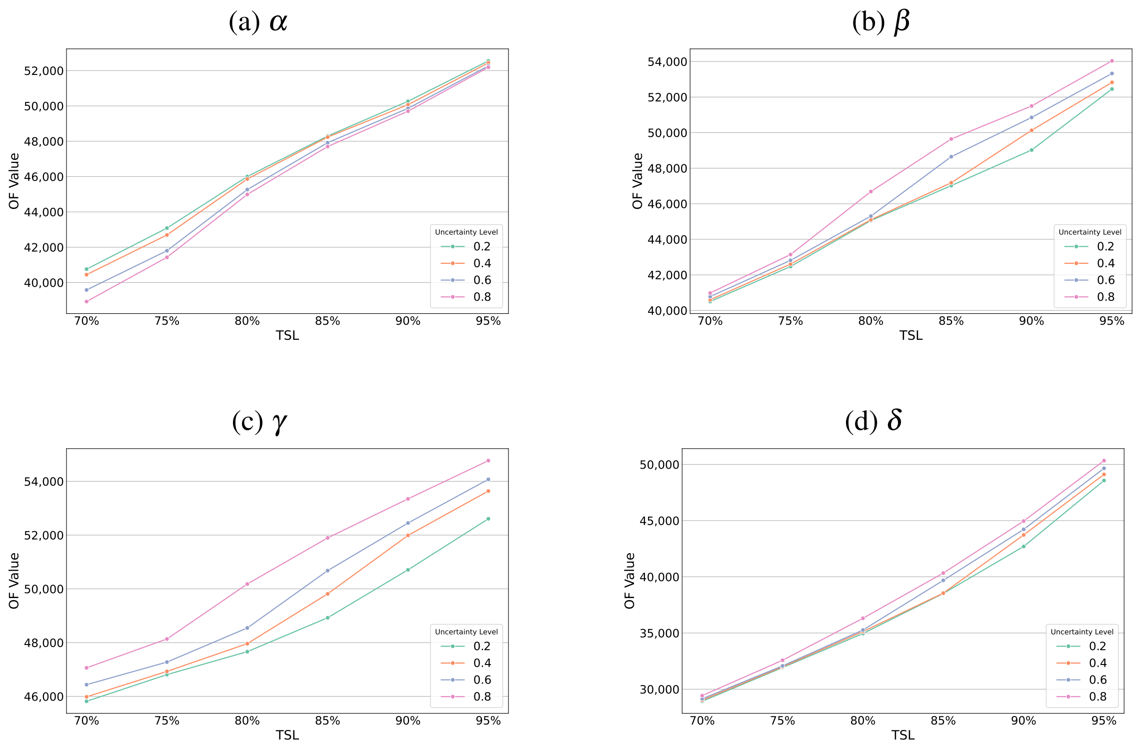


Figure 2.7: Objective function values for different service level measures and uncertainty levels

2.6 Conclusion

In this study, we address the stochastic production routing problem with adaptive routing under demand uncertainty. To tackle this uncertainty, we propose a two-stage stochastic formulation and introduce various service levels from the literature. These service levels offer different approaches to handling uncertainty, particularly in cases where assigning a penalty cost to lost sales is not straightforward, and the primary objective is to maintain a predefined level of demand satisfaction. The adaptive routing component enables flexible routing in the second stage of the problem, resulting in cost reduction.

We also consider four granularity levels, formulating the service levels from either the perspective of customers or the plant, and either per period or over the entire planning horizon. To solve this complex problem, we develop an iterative heuristic approach. In the first phase, we generate setup plans, followed by building feasible solutions through solving production and delivery decisions in the second and third stages, respectively. By iterating over these stages, we aim to improve solutions towards optimality, while also exploring a wider solution space and diversifying solutions by iterating over the first stage.

Finally, we conduct experiments on all the mentioned service levels and different granularity levels, providing insightful discussions. We find that considering service levels for each customer tends to increase costs due to higher demand satisfaction requirements, whereas enforcing service levels from the plant perspective allows for greater flexibility and lower costs. Additionally, we observe that global service levels offer more flexibility compared to individual period service levels. Moreover, we perform experiments under various analyses, including different numbers of scenarios and uncertainty levels, to gain further insight into how the model responds to different situations.

For future studies, extending the problem to a multi-product setting offers practical applicability. Considering multiple products introduces significant complexity due to shared resources, product-specific demands, and constraints. Addressing this extension would likely require the development and refinement of the proposed solution algorithm to effectively handle the increased problem size and complexity. In the presence of multiple products, another promising direction is the consideration of aggregate service levels, as addressed in the LSP literature. Incorporating this concept into the SPRP framework would present new modeling challenges.

Finally, the problem could be examined in a bi-objective setting, where service levels are balanced against another critical objective, such as minimizing carbon emissions. This approach would provide valuable insights into the trade-offs between service level satisfaction and other operational goals, enabling decision-makers to adopt strategies that best align with their priorities.

Supplementary materials

Supplementary files, which include more detailed tables corresponding to Sections 2.5.4, 2.5.5, and 2.5.6, can be found at the following link: https://github.com/AliK094/online_supplements_sprpar_sl. Additional files have also been provided to offer detailed results of all experiments conducted in this study.

Acknowledgment

The authors gratefully acknowledge the support of the Natural Sciences and Engineering Research Council of Canada (RGPIN-2019-05036, RGPIN-2019-05759 and RGPAS-2019-00100), and the HEC Montreal Research Chair in Supply Chain Operations Planning. The authors also acknowledge the support of Digital Research Alliance of Canada for providing computational resources for performing this research.

References

Absi, N., Archetti, C., Dauzère-Pérès, S., and Feillet, D. (2015). "A Two-Phase Iterative Heuristic Approach for the Production Routing Problem". *Transportation Science* 49.4, pp. 784–795. DOI: 10.1287/trsc.2014.0523.

Adulyasak, Y., Cordeau, J.-F., and Jans, R. (2014a). "Formulations and Branch-and-Cut Algorithms for Multivehicle Production and Inventory Routing Problems". *INFORMS Journal on Computing* 26.1, pp. 103–120. DOI: 10.1287/ijoc.2013.0550.

Adulyasak, Y., Cordeau, J.-F., and Jans, R. (2014b). "Optimization-Based Adaptive Large Neighborhood Search for the Production Routing Problem". *Transportation Science* 48.1, pp. 20–45. DOI: 10.1287/trsc.1120.0443.

Adulyasak, Y., Cordeau, J.-F., and Jans, R. (2015a). "Benders Decomposition for Production Routing Under Demand Uncertainty". *Operations Research* 63.4, pp. 851–867. DOI: 10.1287/opre.2015.1401.

Adulyasak, Y., Cordeau, J.-F., and Jans, R. (2015b). “The production routing problem: A review of formulations and solution algorithms”. *Computers & Operations Research* 55, pp. 141–152. DOI: 10.1016/j.cor.2014.01.011.

Applegate, D., Bixby, R., Chvátal, V., and Cook, W. (2020). *Concorde TSP Solver*. URL: https://www.math.uwaterloo.ca/tsp/concorde.html.

Archetti, C., Bertazzi, L., Paletta, G., and Speranza, M. G. (2011). “Analysis of the maximum level policy in a production-distribution system”. *Computers & Operations Research* 38.12, pp. 1731–1746. DOI: 10.1016/j.cor.2011.03.002.

Avci, M. and Topaloglu Yildiz, S. (2020). “A mathematical programming-based heuristic for the production routing problem with transshipments”. *Computers & Operations Research* 123, p. 105042. DOI: 10.1016/j.cor.2020.105042.

Bard, J. F. and Nananukul, N. (2010). “A branch-and-price algorithm for an integrated production and inventory routing problem”. *Computers & Operations Research* 37.12, pp. 2202–2217.

Chandra, P. (1993). “A dynamic distribution model with warehouse and customer replenishment requirements”. *Journal of the Operational Research Society* 44.7. DOI: 10.1057/jors.1993.117.

Chandra, P. and Fisher, M. L. (1994). “Coordination of production and distribution planning”. *European Journal of Operational Research* 72.3, pp. 503–517. DOI: 10.1016/0377-2217(94)90419-7.

Chitsaz, M., Cordeau, J.-F., and Jans, R. (2019). “A Unified Decomposition Matheuristic for Assembly, Production, and Inventory Routing”. *INFORMS Journal on Computing* 31.1, pp. 134–152. DOI: 10.1287/ijoc.2018.0817.

Chitsaz, M., Cordeau, J.-F., and Jans, R. (2020). “A branch-and-cut algorithm for an assembly routing problem”. *European Journal of Operational Research* 282.3, pp. 896–910. DOI: 10.1016/j.ejor.2019.10.007.

Fischetti, M. and Lodi, A. (2003). “Local branching”. *Mathematical Programming* 98, pp. 23–47. DOI: 10.1007/s10107-003-0395-5.

Gade, D. and Küçükyavuz, S. (2013). "Formulations for dynamic lot sizing with service levels". *Naval Research Logistics (NRL)* 60.2, pp. 87–101. DOI: 10.1002/nav.21519.

Gruson, M., Cordeau, J.-F., and Jans, R. (2018). "The impact of service level constraints in deterministic lot sizing with backlogging". *Omega* 79, pp. 91–103. DOI: 10.1016/j.omega.2017.08.003.

Gruson, M., Cordeau, J.-F., and Jans, R. (2023). "Split demand and deliveries in an integrated three-level lot sizing and replenishment problem". *Computers & Operations Research*, p. 106434. DOI: 10.1016/j.cor.2023.106434.

Helber, S., Sahling, F., and Schimmelpfeng, K. (2013). "Dynamic capacitated lot sizing with random demand and dynamic safety stocks". *OR Spectrum* 35.1, pp. 75–105. DOI: 10.1007/s00291-012-0283-6.

Hrabec, D., Hvattum, L. M., and Hoff, A. (2022). "The value of integrated planning for production, inventory, and routing decisions: A systematic review and meta-analysis". *International Journal of Production Economics* 248, p. 108468. DOI: https://doi.org/10.1016/j.ijpe.2022.108468.

Jans, R. (2009). "Solving Lot-Sizing Problems on Parallel Identical Machines Using Symmetry-Breaking Constraints". *INFORMS Journal on Computing* 21.1, pp. 123–136. DOI: 10.1287/ijoc.1080.0283.

Kermani, A., Cordeau, J.-F., and Jans, R. (2024). "A progressive hedging-based matheuristic for the stochastic production routing problem with adaptive routing". *Computers & Operations Research* 169, p. 106745. DOI: https://doi.org/10.1016/j.cor.2024.106745.

Kilic, O. A. and Tarim, S. A. (2011). "An investigation of setup instability in non-stationary stochastic inventory systems". *International Journal of Production Economics* 133.1, pp. 286–292. DOI: 10.1016/j.ijpe.2010.04.021.

Koca, E., Yaman, H., and Aktürk, M. S. (2018). "Stochastic lot sizing problem with nervousness considerations". *Computers & Operations Research* 94, pp. 23–37. DOI: 10.1016/j.cor.2018.01.021.

Li, Y., Chu, F., Chu, C., and Zhu, Z. (2019). “An efficient three-level heuristic for the large-scaled multi-product production routing problem with outsourcing”. *European Journal of Operational Research* 272.3, pp. 914–927. DOI: 10.1016/j.ejor.2018.07.018.

Manousakis, E. G., Kasapidis, G. A., Kiranoudis, C. T., and Zachariadis, E. E. (2022). “An infeasible space exploring matheuristic for the Production Routing Problem”. *European Journal of Operational Research* 298.2, pp. 478–495. DOI: 10.1016/j.ejor.2021.05.037.

Miranda, P. L., Cordeau, J.-F., Ferreira, D., Jans, R., and Morabito, R. (2018). “A decomposition heuristic for a rich production routing problem”. *Computers & Operations Research* 98, pp. 211–230. DOI: 10.1016/j.cor.2018.05.004.

Mousavi, R., Bashiri, M., and Nikzad, E. (2022). “Stochastic production routing problem for perishable products: Modeling and a solution algorithm”. *Computers & Operations Research* 142, p. 105725. DOI: 10.1016/j.cor.2022.105725.

Neves-Moreira, F., Almada-Lobo, B., Cordeau, J.-F., Guimarães, L., and Jans, R. (2019). “Solving a large multi-product production-routing problem with delivery time windows”. *Omega* 86, pp. 154–172. DOI: 10.1016/j.omega.2018.07.006.

Qiu, Y., Qiao, J., and Pardalos, P. M. (2017). “A branch-and-price algorithm for production routing problems with carbon cap-and-trade”. *Omega* 68, pp. 49–61. DOI: 10.1016/j.omega.2016.06.001.

Schenekemberg, C. M., Scarpin, C. T., Pécora, J. E., Guimarães, T. A., and Coelho, L. C. (2021). “The two-echelon production-routing problem”. *European Journal of Operational Research* 288.2, pp. 436–449. DOI: 10.1016/j.ejor.2020.05.054.

Schenekemberg, C. M., Guimarães, T. A., Augusto, C. A., and Coelho, L. C. (2023). “A Three-Front Parallel Branch-and-Cut Algorithm for Production and Inventory Routing Problems”. *Transportation Science*. DOI: 10.1287/trsc.2022.0261.

Schneider, H. (1981). “Effect of service-levels on order-points or order-levels in inventory models”. *International Journal of Production Research* 19.6, pp. 615–631. DOI: 10.1080/00207548108956694.

Sereshti, N., Adulyasak, Y., and Jans, R. (2021). "The value of aggregate service levels in stochastic lot sizing problems". *Omega* 102, p. 102335. DOI: 10.1016/j.omega.2020.102335.

Sereshti, N., Adulyasak, Y., and Jans, R. (2024). "Managing flexibility in stochastic multi-level lot sizing problem with service level constraints". *Omega* 122, p. 102957. DOI: https://doi.org/10.1016/j.omega.2023.102957.

Silver, E. A., Pyke, D. F., Peterson, R., et al. (1998). *Inventory management and production planning and scheduling*. Vol. 3. Wiley New York.

Solyalı, O. and Süral, H. (2017). "A multi-phase heuristic for the production routing problem". *Computers & Operations Research* 87, pp. 114–124. DOI: 10.1016/j.cor.2017.06.007.

Stadtler, H. and Meistering, M. (2019). "Model formulations for the capacitated lot-sizing problem with service-level constraints". *OR Spectrum* 41.4, pp. 1025–1056. DOI: 10.1007/s00291-019-00552-1.

Tarim, S. A. and Kingsman, B. G. (2004). "The stochastic dynamic production/inventory lot-sizing problem with service-level constraints". *International Journal of Production Economics* 88.1, pp. 105–119. DOI: 10.1016/S0925-5273(03)00182-8.

Tempelmeier, H. (2007). "On the stochastic uncapacitated dynamic single-item lotsizing problem with service level constraints". *European Journal of Operational Research* 181.1, pp. 184–194. DOI: 10.1016/j.ejor.2006.06.009.

Tempelmeier, H. (2013). "Stochastic Lot Sizing Problems". *International Series in Operations Research & Management Science*. Vol. 192. Springer, pp. 313–344. DOI: 10.1007/978-1-4614-6777-9_10.

Tempelmeier, H. and Herpers, S. (2010). "ABC β–a heuristic for dynamic capacitated lot sizing with random demand under a fill rate constraint". *International Journal of Production Research* 48.17, pp. 5181–5193. DOI: 10.1080/00207540903179782.

Tempelmeier, H. and Herpers, S. (2011). "Dynamic uncapacitated lot sizing with random demand under a fillrate constraint". *European Journal of Operational Research* 212.3, pp. 497–507. DOI: 10.1016/j.ejor.2011.02.012.

Tomazella, C. P., Santos, M. O., Alem, D., and Jans, R. (2023). "Service-level-driven procurement and production lot-sizing problem with demand fulfilment". *International Journal of Production Research*, pp. 1–22. DOI: 10.1080/00207543.2023.2204958.

Tunc, H., Kilic, O. A., Tarim, S. A., and Eksioglu, B. (2013). "A simple approach for assessing the cost of system nervousness". *International Journal of Production Economics* 141.2, pp. 619–625. DOI: 10.1016/j.ijpe.2012.09.022.

Tunc, H., Kilic, O. A., Tarim, S. A., and Eksioglu, B. (2014). "A reformulation for the stochastic lot sizing problem with service-level constraints". *Operations Research Letters* 42.2, pp. 161–165. DOI: 10.1016/j.orl.2014.01.010.

Tunc, H., Kilic, O. A., Tarim, S. A., and Rossi, R. (2018). "An Extended Mixed-Integer Programming Formulation and Dynamic Cut Generation Approach for the Stochastic Lot-Sizing Problem". *INFORMS Journal on Computing* 30.3, pp. 492–506. DOI: 10.1287/ijoc.2017.0792.

Zhang, Y., Alshraideh, H., and Diabat, A. (2018). "A stochastic reverse logistics production routing model with environmental considerations". *Annals of Operations Research* 271.2, pp. 1023–1044. DOI: 10.1007/s10479-018-3045-2.

Chapter 3

The Stochastic Two-Echelon Production Routing Problem with Adaptive Routing

Abstract

We address a stochastic two-echelon production routing problem with adaptive routing under demand uncertainty. This problem involves a production plant with production and inventory capacities where goods are produced and transported to a set of warehouses using a homogeneous fleet of vehicles. These products are then stored and subsequently delivered to customers. The products distributed from warehouses to customers can satisfy current demand or be stored for future use. We also consider the flexibility to modify customer-to-warehouse assignments for each period and scenario. We introduce a two-stage stochastic formulation to address this problem. In the first stage, the setup plan, production quantities, and routing plan to deliver goods from the production plant to the warehouses are decided. The second stage incorporates the routing decisions to deliver goods from warehouses to customers based on realized demand, with penalty costs applied for each unit of unmet demand. The objective is to minimize the total costs of production, inventory holding, and transportation in the first echelon, as well as the expected inventory holding, transportation, and penalty costs for unsatisfied demand in the second echelon. To reduce the complexity of the problem, we decompose it into two subproblems. The first subproblem addresses the first-stage decisions using an approximation of the demand from the second stage, while the second subproblem focuses on solving the second-stage problem. We propose a hybrid heuristic approach to tackle these subproblems: a mixed-integer programming model is employed for the first stage, and the second stage is handled by an Iterated Local Search metaheuristic. Computational experiments and sensitivity analyses are conducted to demonstrate the benefits of solving this problem and to assess the performance of the algorithm. The proposed algorithm successfully generated feasible solutions for instances with up to five warehouses and two vehicles in the first stage, and three vehicles per warehouse in the second stage. These solutions can typically not be improved, or only to a small extent, by a branch-and-cut algorithm which is run for 2 hours. The results also indicate that solving the stochastic problem yields, on average, a substantial improvement compared to solving a deterministic problem where the random variables are replaced by their expectation.

3.1 Introduction

Integrating different problems of supply chain planning can significantly enhance the efficiency of the entire network. However, the complexity of such integrated problems has traditionally led researchers and practitioners to address these components sequentially to maintain tractability (Hrabec et al., 2022). Recent advances in optimization techniques and computational capabilities have enabled the study of more challenging integrated problems (Absi et al., 2018). One example is the Production Routing Problem (PRP), which integrates production planning, inventory management, and transportation decisions into a single optimization framework. This integration promotes better coordination among supply chain components, offering substantial opportunities for cost reduction and efficiency gains (Adulyasak et al., 2015b).

Following its introduction by Chandra (1993) and Chandra and Fisher (1994), the PRP has been widely studied. In its classical form, the PRP assumes a production facility that manufactures a single product, which must be distributed to customers to meet their demands over a finite planning horizon. The main decisions in this setting involve determining the setup plans of the production facility, production quantities, vehicle routing plans, delivery quantities to customers in each period, and inventory levels at both the plant and customer locations. These decisions are made in an integrated manner to minimize the total production, inventory, and transportation costs while ensuring that customer demands are satisfied throughout the planning horizon.

Despite the complexity of the problem, several studies have attempted to develop and improve exact algorithms to solve small and medium-sized PRP instances. Bard and Nananukul (2010) propose a Branch-and-Price algorithm to solve the problem. Archetti et al. (2011) propose a Branch-and-Cut (BC) algorithm to solve the PRP with a single vehicle. Several studies have improved the BC algorithm and introduced valid inequalities to enhance the efficiency of this algorithm for the PRP (Adulyasak et al., 2014a; Schenekemberg et al., 2021; Schenekemberg et al., 2023).

Most studies, however, focus on developing heuristic and matheuristic methods to leverage the problem structure and obtain good solutions for larger practical instances. Among these, several studies decompose the problem into more tractable subproblems and use iterative algorithms to obtain high-quality solutions for the original problem (Absi et al., 2015; Archetti et al., 2011; Ben Ahmed et al., 2023; Manousakis et al., 2022; Russell, 2017; Solyalı and Süral, 2017; Vadseth et al., 2023). Other studies consider metaheuristic algorithms or hybrid versions of these algorithms within an iterative approach to solve the problem (Adulyasak et al., 2014b; Armentano et al., 2011; Bard and Nananukul, 2009; Rodrigues et al., 2023).

Different variants of the PRP, such as the PRP with perishable products (Alvarez et al., 2022a; Neves-Moreira et al., 2019), multi-product PRP (Li et al., 2019; Qiu et al., 2019; Rodrigues et al., 2023), and PRP with transshipment (Alvarez et al., 2022a; Avci and Topaloglu Yildiz, 2020), have been explored in the literature. Chitsaz et al. (2019,

2020) propose a different variant called the assembly routing problem, where instead of considering multiple customers, they address a setting with multiple suppliers and a manufacturing facility. Alvarez et al. (2022b) study a variant of the PRP with multiple plants and products and incorporate four forms of consistency, namely driver, source, product, and plant consistency. Other factors, such as environmental effects and greenhouse gas emissions, are also addressed in various studies (Kumar et al., 2016; Qiu et al., 2018, 2017).

In real-world supply chains, demand uncertainty often leads to disruptions and unexpected costs. To mitigate these risks, several studies have introduced demand uncertainty into the PRP, enhancing the resiliency of supply chains. Adulyasak et al. (2015a) propose a two-stage and a multi-stage stochastic formulation for the PRP with demand uncertainty, where setup and routing decisions are made in the first stage and production and delivery quantities are decided in the second stage. Agra et al. (2018) investigate the PRP under demand uncertainty while considering backlogs. Zhang et al. (2018) consider the PRP with simultaneous pickup and delivery with remanufacturing possibilities. Mousavi et al. (2022) study the PRP with perishable products under demand uncertainty. Wang et al. (2021) examine the PRP in an uncertain environment with potential uncertainties in demand and costs. Kermani et al. (2024) present a two-stage stochastic problem for the PRP with demand uncertainty and adaptive routing. They highlight that this adaptive structure improves cost efficiency by increasing flexibility in responding to demand uncertainty. Kermani et al. (2025) investigate different types of service levels in the stochastic PRP with adaptive routing where, instead of penalty costs for unmet demands, a minimum demand satisfaction based on different service levels is imposed.

While the PRP represents a critical step toward integrating supply chain components, its traditional form is limited to single-layer distribution networks. However, today's supply chains have evolved into increasingly complex networks, often involving multiple layers that require coordinated decision-making across different echelons. For instance, the inclusion of intermediate layers, specifically, warehouses located between production plants and customers, is increasingly common. This configuration, particularly common in large urban areas, helps reduce congestion and environmental impacts by limiting the use of large trucks within cities (Perboli et al., 2011). In such systems, large vehicles transport goods from manufacturing plants to warehouses, from which smaller, low-emission vehicles handle last-mile deliveries to customers (Qiu et al., 2021).

While both the lot-sizing problem (LSP) and vehicle routing problem (VRP) literature have tackled aspects of multi-echelon supply chains individually, integrated models that consider both production and transportation across multiple layers remain scarce. For example, Gruson et al. (2019) propose a three-level lot-sizing and replenishment problem (3LSPD) involving a manufacturing facility, warehouses, and retailers. However, their model focuses solely on production and replenishment decisions, omitting routing considerations between the different levels. In contrast, the two-echelon VRP (2E-VRP) introduced by Perboli et al. (2011) considers two transportation layers, one from a central depot to distribution centers, and another from these centers to customers. Yet, this model does not account for production decisions.

Only a few studies attempt to bridge this gap by integrating production, inventory, and routing decisions across multiple echelons. Schenekemberg et al. (2021) present a two-echelon PRP inspired by the petrochemical industry, encompassing suppliers, production plants, and customers. Gruson et al. (2023) extend the 3LSPD to include distribution, using direct shipments from production plants to warehouses while considering routing decisions for the warehouse-to-customer level. Qiu et al. (2021) propose a two-echelon PRP with cross-docking satellites (2E-PRPCS), where routing decisions apply to both echelons. Despite these contributions, a significant gap remains in integrating supply chain planning under demand uncertainty within realistic, multi-layered networks. Furthermore, the scale of problems addressed in deterministic models tends to be small compared to real-world applications, underscoring the need for efficient algorithms capable of solving larger instances.

To address these challenges, we propose a two-stage stochastic formulation for a two-echelon production routing problem. In this problem, a single manufacturing facility produces a product and transports it to a set of distribution facilities. The first-stage decisions include the production setup, production quantities, and transportation of goods from the plant to the warehouses. These decisions are made before demand realization while considering expected recourse costs. The second stage, performed after demand scenarios are realized, manages inventory at both the warehouse and customer levels and determines the distribution of goods from warehouses to customers. This flexible routing approach, where routes are selected post-demand realization, aligns with the concept of adaptive routing in stochastic PRP frameworks (Kermani et al., 2024).

To the best of our knowledge, this is the first study to incorporate demand uncertainty in a two-echelon production routing problem with adaptive routing in the second echelon. We refer to this model as the Stochastic Two-Echelon Production Routing Problem with Adaptive Routing (S2EPRP-AR). In the proposed S2EPRP-AR, shortages are allowed, with penalty costs incurred for unmet demand, providing additional flexibility for managing disruptions. Moreover, our model introduces flexible customer-to-warehouse assignments, allowing allocations to vary across periods and scenarios. Nonetheless, each customer remains assigned to one warehouse per period and scenario to maintain operational feasibility.

To solve this complex problem, we adopt a decomposition approach, breaking the model into two subproblems to enhance computational tractability. The first subproblem focuses on the first-stage decisions including production setups, production quantities, and deliveries from the plant to the warehouses, using approximate aggregate demand based on customer-to-warehouse assignments across periods and scenarios. Once these plans are established, the second subproblem handles the distribution of products from warehouses to customers, leveraging the warehouse inventory determined in the first stage. Notably, this second subproblem is scenario-decomposable, allowing for the independent solution of each scenario, thereby improving computational efficiency.

We develop a hybrid heuristic algorithm to solve these subproblems. The first-stage subproblem is formulated as a Mixed-Integer Programming (MIP) model and solved using a general-purpose solver. For the second stage, we employ an Iterated Local Search (ILS) metaheuristic to address the scenario-dependent subproblems. Specifically, we solve a multi-depot Inventory Routing Problem (IRP) for each scenario using ILS, where local search refines the solution while perturbation mechanisms help escape local optima. These subproblems are solved iteratively, with customer-to-warehouse assignments updated in every iteration, progressively improving the overall solution. It is worth noting that we also adapt this algorithm to solve the deterministic two-echelon PRP, enabling it to handle larger instances more effectively.

The remainder of the paper is organized as follows: Section 3.2 presents the definition and mathematical formulation of the S2EPRP-AR, detailing the assumptions and constraints. Section 3.3 describes the solution algorithm developed to address the problem. Computational experiments and results are presented in Section 3.4. Finally, Section 3.5 concludes the paper with a summary of findings and future research directions.

3.2 Problem Formulation

We define the S2EPRP-AR on an undirected graph $\mathscr{G} = (\mathscr{N}, \mathscr{E})$, where $\mathscr{N} = \mathscr{N}_o \cup \mathscr{N}_w \cup \mathscr{N}_c$, with $\mathscr{N}_o = \{0\}$ representing the plant node; $\mathscr{N}_w = \{1, \ldots, N_w\}$ denoting the set of warehouse nodes, where N_w is the number of warehouses; and $\mathscr{N}_c = \{N_w + 1, \ldots, N_w + N_c\}$ the set of customer nodes, where N_c is the number of customers. The set of edges is defined as $\mathscr{E} = \mathscr{E}_{pw} \cup \mathscr{E}_{wc}$, where $\mathscr{E}_{pw} = \{(i, j) : i, j \in \mathscr{N}_o \cup \mathscr{N}_w,\ i < j\}$ represents the edges between the plant and warehouses (first echelon), and $\mathscr{E}_{wc} = \{(i, j) : i \in \mathscr{N}_w, j \in \mathscr{N}_c \vee i, j \in \mathscr{N}_c, i < j\}$ represents the edges related to the second echelon (warehouses to customers and customers to customers). We also define the subset $\tilde{\mathscr{E}}_{wc}(\eta)$ to represent edges $(i, j) \in \mathscr{E}_{wc}$, where i, j belong to η and $\eta \subseteq \mathscr{N}_w \cup \mathscr{N}_c$. For all edges incident to node $i \in \mathscr{N}_w \cup \mathscr{N}_c$, we define the set $\tilde{\varepsilon}_{wc}\{i\}$.

The problem is defined over a finite and discrete planning horizon of T periods, denoted by $\mathscr{T} = \{1, \ldots, T\}$. The set $\mathscr{K}_p = \{1, \ldots, K_p\}$ represents the vehicles available at the plant, where K_p homogeneous vehicles with capacity Q^p are responsible for transporting products from the plant to the warehouses. These vehicles must start and end their routes at the plant in each period. Additionally, each warehouse $w \in \mathscr{N}_w$ is assigned a set of homogeneous vehicles with capacity Q^w, defined as $\mathscr{K}_w = \left\{\sum_{l=1}^{w-1} K_l + 1, \ldots, \sum_{l=1}^{w} K_l\right\}$, where K_l is the number of vehicles at warehouse l.

Figure 3.1 illustrates an example of the S2EPRP-AR structure with three warehouses and ten customers. It demonstrates the flexibility of customer-to-warehouse assignments and highlights that first-echelon routes remain fixed for a given period across all scenarios. In contrast, second-echelon routes can vary from period to period and across different scenarios.

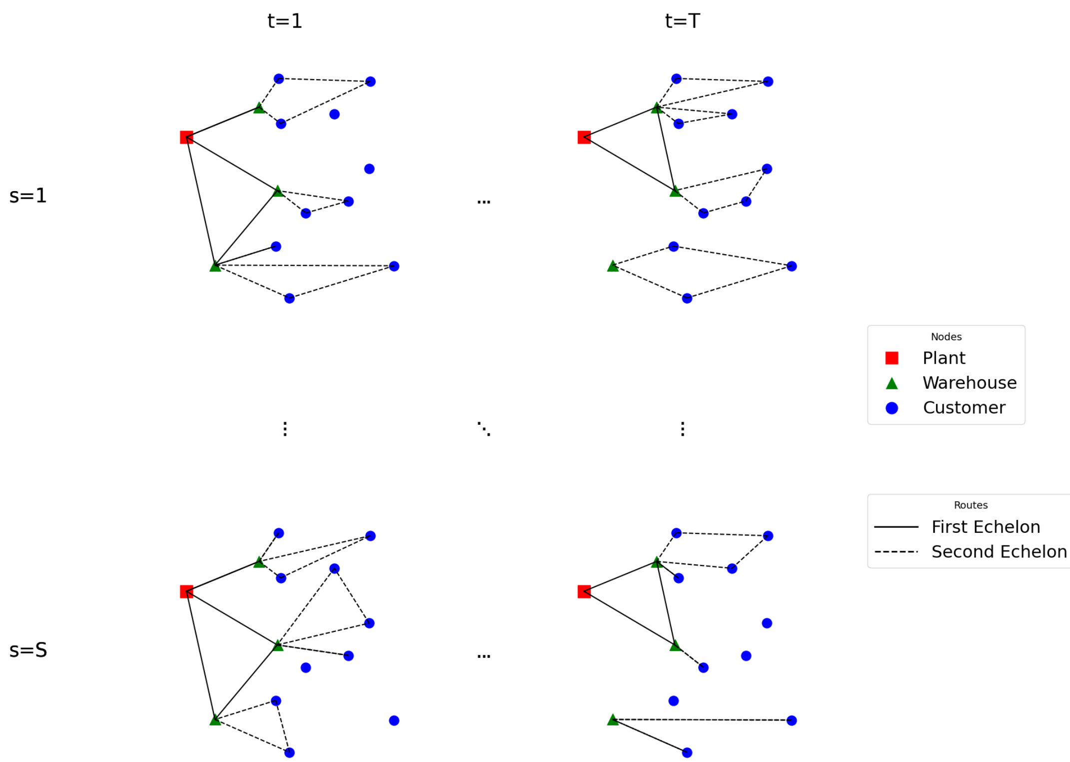


Figure 3.1: Stochastic two-echelon production routing problem with adaptive routing in the second echelon

The production plant has a production capacity of C in each period. A fixed setup cost F is incurred in any period where production takes place, along with an additional cost u per unit produced. Each node $i \in \mathscr{N}$ has an inventory capacity L_i and an initial inventory level I_{i0}. A unit holding cost h_i is applied to every item stored at node $i \in \mathscr{N}$. We assume that shortages are allowed in the model, and each unit of unmet demand for customer i

incurs a penalty cost α_i.

Customer demand is modeled as a random variable with a known probability distribution, estimated from historical data. To account for demand uncertainty, we consider a finite set of scenarios $\mathscr{S} = \{1,\dots,S\}$, where S is the number of scenarios. Each scenario $s \in \mathscr{S}$ has an associated probability $\xi_s > 0$, with $\sum_{s\in\mathscr{S}} \xi_s = 1$. The parameter d_{it}^s represents the demand of customer $i \in \mathscr{N}_c$ in period $t \in \mathscr{T}$ under scenario s.

We define the set $\mathscr{R} = \{1,\dots,R\}$ to represent all possible routes in the first echelon. To construct this set, we assume to have the optimal route with associated cost $\tilde{c}_r$ for every non-empty subset of warehouse nodes, leading to $R = 2^{N_w} - 1$. This definition allows us to model first-echelon transportation constraints based on route selection. We also define a_{wr} as a binary parameter equal to 1 if warehouse w is present in route r, and 0 otherwise. The upper bound on delivery quantities to warehouse w is given by $\mathscr{M}_w = \min\{\mathscr{Q}^p, L_w\}$. In the second echelon, we define a cost c_{ij} for visiting edge $(i,j) \in \mathscr{E}_{wc}$. The delivery quantities to customer i in period t and scenario s are bounded by $\mathscr{M}'^s_{it} = \min\{\mathscr{Q}^w, L_i, \sum_{l=t}^{T} d_{il}^s\}$.

The first-stage decisions of the model are the production setup decisions, represented by the binary variables y_t, which takes value 1 if production occurs in period t, and 0 otherwise. The production quantity in period t is denoted by p_t, and the inventory level at the plant by I_{0t}. The variable q_{wrt} indicates the quantity delivered to warehouse w via route r in period t, and the binary variable o_{rt} is 0 if route r is selected in period t and 0 otherwise.

Second-stage decisions include the inventory levels at warehouses and customer locations, denoted by I_{it}. The amount of unmet demand for customer i in period t under scenario s is captured by l_{it}^s. The delivery quantity to customer i in period t and scenario s via vehicle $k \in \bigcup_{w\in\mathscr{N}_w} \mathscr{K}_w$ is denoted by w_{ikt}^s. The binary variable z_{ikt}^s indicates whether or not node i is visited by vehicle k in period t under scenario s, and the integer variable x_{ijkt}^s represents the number of times edge $(i,j) \in \mathscr{E}_{wc}$ is traversed by vehicle k in period t under scenario s.

A summary of all sets, parameters, and decision variables used in the model is provided in Appendix C.1. Given these notations, the mathematical formulation of the

S2EPRP-AR is presented below.

$$\text{Min} \sum_{t\in\mathscr{T}} \Bigg(F y_t + u p_t + h_0 I_{0t} + \sum_{r\in\mathscr{R}} \tilde{c}_r o_{rt} + \sum_{s\in\mathscr{S}} \xi_s \Big(\sum_{w\in\mathscr{N}_w} I^s_{wt} + \sum_{w\in\mathscr{N}_w} \sum_{k\in\mathscr{K}_w} \sum_{(i,j)\in\mathscr{E}_{wc}} c_{ij} x^s_{ijkt} + \sum_{i\in\mathscr{N}_c} h_i I^s_{it} + \sum_{i\in\mathscr{N}_c} \alpha_i l^s_{it} \Big) \Bigg) \quad (3.1)$$

s.t.

$$p_t \leq C y_t \qquad \forall t \in \mathscr{T} \quad (3.2)$$

$$I_{0t} = I_{0,t-1} + p_t - \sum_{w\in\mathscr{N}_w} \sum_{r\in\mathscr{R}} q_{wrt} \qquad \forall t \in \mathscr{T} \quad (3.3)$$

$$I_{0t} \leq L_0 \qquad \forall t \in \mathscr{T} \quad (3.4)$$

$$I^s_{wt} = I^s_{w,t-1} + \sum_{r\in\mathscr{R}} q_{wrt} - \sum_{i\in\mathscr{N}_c} \sum_{k\in\mathscr{K}_w} w^s_{ikt} \qquad \forall w \in \mathscr{N}_w, t \in \mathscr{T}, s \in \mathscr{S} \quad (3.5)$$

$$I^s_{wt} \leq L_w \qquad \forall w \in \mathscr{N}_w, t \in \mathscr{T}, s \in \mathscr{S} \quad (3.6)$$

$$\sum_{r\in\mathscr{R}} a_{wr} o_{rt} \leq 1 \qquad \forall w \in \mathscr{N}_w, t \in \mathscr{T} \quad (3.7)$$

$$\sum_{r\in\mathscr{R}} o_{rt} \leq K_p \qquad \forall t \in \mathscr{T} \quad (3.8)$$

$$q_{wrt} \leq \mathscr{M}_w a_{wr} \qquad \forall w \in \mathscr{N}_w, r \in \mathscr{R}, t \in \mathscr{T} \quad (3.9)$$

$$\sum_{w\in\mathscr{N}_w} q_{wrt} \leq Q^p o_{rt} \qquad \forall r \in \mathscr{R}, t \in \mathscr{T} \quad (3.10)$$

$$I^s_{it} = I^s_{i,t-1} + \sum_{k\in \bigcup_{w\in\mathscr{N}_w} \mathscr{K}_w} w^s_{ikt} + l^s_{it} - d^s_{it} \qquad \forall i \in \mathscr{N}_c, t \in \mathscr{T}, s \in \mathscr{S} \quad (3.11)$$

$$I^s_{it} + d^s_{it} \leq L_i \qquad \forall i \in \mathscr{N}_c, t \in \mathscr{T}, s \in \mathscr{S} \quad (3.12)$$

$$w^s_{ikt} \leq \mathscr{M}'^s_{it} z^s_{ikt} \qquad \forall i \in \mathscr{N}_c, k \in \bigcup_{w\in\mathscr{N}_w} \mathscr{K}_w, t \in \mathscr{T}, s \in \mathscr{S} \quad (3.13)$$

$$\sum_{i\in\mathscr{N}_c} w^s_{ikt} \leq Q^w z^s_{wkt} \qquad \forall w \in \mathscr{N}_w, k \in \mathscr{K}_w, t \in \mathscr{T}, s \in \mathscr{S} \quad (3.14)$$

$$\sum_{k\in \bigcup_{w\in\mathscr{N}_w} \mathscr{K}_w} z^s_{ikt} \leq 1 \qquad \forall i \in \mathscr{N}_c, t \in \mathscr{T}, s \in \mathscr{S} \quad (3.15)$$

$$\sum_{(j,j')\in\tilde{\varepsilon}(\{i\})} x^s_{jj'kt} = 2 z^s_{ikt} \qquad \forall i \in \mathscr{N}_w \cup \mathscr{N}_c, k \in \bigcup_{w\in\mathscr{N}_w} \mathscr{K}_w, t \in \mathscr{T}, s \in \mathscr{S} \quad (3.16)$$

$$\sum_{(i,j)\in\tilde{\mathscr{E}}(\eta)} x^s_{ijkt} \leq \sum_{i\in\eta} z^s_{ikt} - z^s_{ekt}$$

$$\forall \eta \subseteq \mathscr{N}_c, |\eta| \geq 2, e \in \eta, k \in \bigcup_{w\in\mathscr{N}_w} \mathscr{K}_w, t \in \mathscr{T}, s \in \mathscr{S} \quad (3.17)$$

$$y_t \in \{0,1\} \qquad \forall t \in \mathscr{T} \quad (3.18)$$

$$p_t \geq 0 \qquad \forall t \in \mathscr{T} \quad (3.19)$$

$$q_{wrt} \geq 0 \qquad \forall w \in \mathscr{N}_w, r \in \mathscr{R}, t \in \mathscr{T} \quad (3.20)$$

$$I_{0t} \geq 0 \qquad \forall t \in \mathscr{T} \quad (3.21)$$

$$o_{rt} \in \{0,1\} \qquad \forall r \in \mathscr{R}, t \in \mathscr{T} \quad (3.22)$$

$$I^s_{it} \geq 0 \qquad \forall i \in \mathscr{N}_w \cup \mathscr{N}_c, t \in \mathscr{T}, s \in \mathscr{S} \quad (3.23)$$

$$l^s_{it} \geq 0 \qquad \forall i \in \mathscr{N}_c, t \in \mathscr{T}, s \in \mathscr{S} \quad (3.24)$$

$$w^s_{ikt} \geq 0 \qquad \forall i \in \mathscr{N}_c, k \in \bigcup_{w\in\mathscr{N}_w} \mathscr{K}_w, t \in \mathscr{T}, s \in \mathscr{S} \quad (3.25)$$

$$z^s_{wkt} \in \{0,1\} \qquad \forall w \in \mathscr{N}_w, k \in \mathscr{K}_w, t \in \mathscr{T}, s \in \mathscr{S} \quad (3.26)$$

$$z^s_{ikt} \in \{0,1\} \qquad \forall i \in \mathscr{N}_c, k \in \bigcup_{w\in\mathscr{N}_w} \mathscr{K}_w, t \in \mathscr{T}, s \in \mathscr{S} \quad (3.27)$$

$$x^s_{ijkt} \in \{0,1\} \qquad \forall (i,j) \in \mathscr{E}_{wc}, i \notin \mathscr{N}_w, k \in \bigcup_{w\in\mathscr{N}_w} \mathscr{K}_w, t \in \mathscr{T}, s \in \mathscr{S} \quad (3.28)$$

$$x^s_{ijkt} \in \{0,1,2\} \qquad \forall (i,j) \in \mathscr{E}_{wc}, i \in \mathscr{N}_w, k \in \mathscr{K}_w, t \in \mathscr{T}, s \in \mathscr{S}. \quad (3.29)$$

The objective function (3.1) minimizes the total setup costs, inventory holding costs at the plant, and transportation costs of delivering products from the plant to warehouses, as well as the expected inventory holding costs at the warehouses and customers, transportation costs of goods from warehouses to customers, and penalty costs incurred on every unit of unmet demand. Constraints (3.2) ensure that production is only possible if a setup takes place in period t. Constraints (3.3), (3.5), and (3.11) are the inventory balance constraints for the plant, warehouses, and customers, respectively. The inventory limit of each of these facilities is enforced by constraints (3.4), (3.6), and (3.12). Note that for constraints (3.12), the original form $I^s_{it} + d^s_{it} - l^s_{it} \leq L_i$ accounts for lost sales. While this makes the constraints less restrictive, in the proposed setting a shortage in one period will never be created to satisfy demand in a later period. This is because inventory holding costs are positive and lost sales incur a cost that is independent of the period. Therefore, we will not have both a strictly positive inventory level and lost sales in the same period, and the term l^s_{it} can be omitted without affecting the model's behavior.

For the first echelon of the problem, we assume that all routes are explicitly defined. Constraints (3.7) guarantee that warehouse w is visited by at most one route in each period. Constraints (3.8) limit the number of selected routes to the available vehicles in each period. Constraints (3.9) ensure that delivery to warehouse w is only possible if it is included in route r, while the sum of deliveries on route r is limited to the vehicle capacity by constraints (3.10). Constraints (3.13) enforce that delivery to customer i is only possible if the customer is visited by vehicle k from warehouse w. The capacity on the vehicles at the warehouses is imposed by constraints (3.14). Constraints (3.15) prevent split deliveries to customers. Constraints (3.16) are the degree constraints, and constraints (3.17) are the subtour elimination constraints (SECs). Finally, constraints (3.18 - 3.29) define the domains of the decision variables.

3.3 Hybrid Heuristic Algorithm

Solving the S2EPRP-AR presents significant challenges due to its NP-hard nature, especially when incorporating uncertainty and adaptive routing. Even for medium-size instances of the deterministic version, exact algorithms often struggle to find feasible solutions. In such cases, Qiu et al. (2021) propose a matheuristic algorithm that first generates an incumbent solution, then applies BC to reduce the optimality gap and improve the upper bound (UB).

In response to these challenges, we propose a Hybrid Heuristic Algorithm (HHA) that leverages the problem's structure by decomposing it into smaller subproblems, which are solved iteratively to enhance the overall solution. The HHA consists of two main components: a MIP model for the first echelon, and an ILS metaheuristic to manage the second echelon.

The algorithm begins by constructing an initial solution, which is iteratively refined by solving a restricted S2EPRP-AR with fixed second-echelon routes to improve the first-echelon solutions. Next, ILS is applied to the second-echelon routes, followed by solving an LP to optimize the continuous variables based on the updated routes. The structure of the HHA is outlined in Algorithm 3. The quality of the initial solution is critical to the algorithm's performance, and the construction heuristic used for this purpose is detailed in Section 3.3.1. After obtaining the initial solution, a local search heuristic, specifically a multi-start Randomized Variable Neighborhood Descent (RVND), is applied to each scenario s to improve the solution (Line 5). This local search algorithm is similar to the one used as the local search component of the ILS, as explained in Section 3.3.2.

Algorithm 3 Hybrid Heuristic Algorithm

1: **Input:** Problem Parameters
2: **Output:** Incumbent solution for the S2EPRP-AR
3: Initialize incumbent solution: $sol \leftarrow \emptyset$;
4: $sol_0 \leftarrow$ ConstructInitialSolution;   (Algorithm 4)
5: $sol \leftarrow$ RVND(sol_0);   (Algorithm 6)
6: $sol' \leftarrow sol$
7: **while** stopping criteria not met **do**
8:     **if** no improvement in the global best OF for NIG iterations **then**
9:         $sol' \leftarrow sol$;
10:     **end if**
11:     **for** each period $t \in \mathscr{T}$ and scenario $s \in \mathscr{S}$ **do**
12:         Reassign customers experiencing stockouts to alternative warehouses;
13:     **end for**
14:     **if** no stockouts are present & no local improvements on the OF for NIL iterations **then**
15:         **for** each scenario $s \in \mathscr{S}$ **do**
16:             Select a random warehouse $w \in \mathscr{N}_w$ in a random period $t \in \mathscr{T}$ to remove;
17:             Reassign all customers of the removed warehouse to other feasible warehouses;
18:         **end for**
19:     **end if**
20:     Solve the restricted S2EPRP-AR using the updated customer assignments;
21:     Update sol' with the new solution;
22:     $sol' \leftarrow$ ILS(sol');   (Algorithm 5)
23:     **if** $objVal(sol') < objVal(sol)$ **then**
24:         Update incumbent solution: $sol \leftarrow sol'$
25:     **end if**
26: **end while**
27: **return** sol

In subsequent iterations, the algorithm adjusts customer-to-warehouse assignments whenever stockouts occur. When a shortage is detected for a customer, it is removed from its current warehouse and delivery route and reassigned to a different warehouse's route to reduce the cost of unmet demand. The algorithm evaluates potential routes based on minimum insertion cost. If the insertion cost is lower than the current stockout penalty, and both warehouse and vehicle capacities allow the reassignment, the customer is integrated into the new route. This process is repeated for all customers facing shortages across every period and scenario (lines 11–13).

When no shortages are present in an iteration, the algorithm diversifies the search by randomly removing a warehouse within a selected period. This removal forces the reassignment of its customers to other warehouses across all scenarios in that period. This strategy helps explore new solution spaces and prevents the search from getting trapped in local optima. The removal of a warehouse only occurs if the local best objective function value (OF) has not improved for *NIL* consecutive iterations. If the absence of improvement continues, a new warehouse is removed after every additional *NIL* iterations (lines 14–19). If no improvement in the global best OF is observed after *NIG* iterations, the algorithm resets the current solution to the incumbent solution. This reset allows for the exploration of new neighborhoods and helps avoid prolonged stagnation (lines 8–10).

At the end of each iteration, a restricted version of the S2EPRP-AR is solved in which the second-echelon routes are fixed, allowing the algorithm to update the first-echelon decisions. This is followed by applying Iterated Local Search (ILS) to refine the second-echelon routes. The ILS procedure improves the routes using local search operators and applies perturbations to escape local optima. If an improved solution is found, the current solution is updated; otherwise, the algorithm proceeds to the next iteration (lines 20–22). Additionally, if the current OF is better than the global best, the incumbent solution and the global best value are also updated. This process is repeated until one of the stopping criteria is met.

The algorithm terminates when no improvement is observed in the best solution for *Max_NIG* consecutive iterations, or the algorithm's runtime exceeds the predefined time limit $HHA\_TimeLimit$.

3.3.1 Initial Solution

To generate an initial solution that is both feasible and provides a strong starting point for the algorithm, we decompose the problem into two subproblems. The pseudocode for the construction heuristic algorithm is provided in Algorithm 4. The first subproblem is a stochastic multi-warehouse PRP (MW-PRP), which focuses exclusively on the plant and warehouse nodes. Since customer demand drives the problem, we introduce a virtual demand for each warehouse. This is achieved by assigning customers to their nearest warehouse and by setting the demand of each warehouse equal to the sum of the demands of its assigned customers. Thus, set $\mathscr{C}^w = \left\{ i \in \mathscr{N}_c \mid w = \arg\min_{w'} \{c_{w',i}\} \right\}$ denotes the set of customers that are assigned to $w \in \mathscr{N}_w$.

To prevent over-delivery, the initial inventory of each customer is subtracted from its demand during the calculation. Additionally, no inventory or delivery costs are considered for customers at this stage; however, an approximate penalty cost on each unit of unmet demand is considered based on the average cost of unmet (aggregate) demand for customers assigned to each warehouse. Consequently, the objective function of this subproblem includes fixed and variable production costs, transportation costs between the plant and warehouses, inventory holding costs at both the plant and warehouses, and approximate costs for unmet demand.

It is worth noting that we explicitly define the routes between the plant and warehouses in this subproblem. Although the number of potential routes increases exponentially with the number of warehouses, we assume that the number of warehouses remains limited, as is typical in real-world scenarios. This assumption enables us to explicitly define all possible routes, thereby simplifying the problem. To generate these routes, we solve a TSP for each non-empty subset of nodes, denoted as $\eta_{FE} \subseteq \mathscr{N}_w \setminus \{\emptyset\}$, based on the transportation costs of the edges in $\mathscr{E}_{pw}$. Hence, the total number of binary variables required for the first-echelon routes is $2^{N_w} - 1$.

Algorithm 4 Construction Heuristic

Input: Problem Parameters, Set of customers initially assigned to each warehouse $\mathscr{C}^w$
Output: Feasible solution for the S2EPRP-AR
Initialize $sol \leftarrow \emptyset$
Initialize $sol_{\text{FE}} \leftarrow \emptyset$
Initialize $\hat{d}^s_{wt}$ and $\hat{\alpha}_w$
Solve MW-PRP to obtain production plan and first-echelon routes
Update sol_{FE}
Initialize $sol_{\text{SE}} \leftarrow \emptyset$
Determine $\tilde{q}^s_{wt}$
Construct $sorted\_customers$ as in (3.34)
for each scenario s **do**
  for each warehouse w **do**
    $look\_ahead \leftarrow 1$
    while $look\_ahead \leq \lceil \frac{T}{2} \rceil$ **do**
      Compute projected inventory $\tilde{I}^s_{it}$ for each customer $i \in \mathscr{C}^w$, period $t \in \mathscr{T}$, and scenario $s \in \mathscr{S}$ as in (3.35)
      for each period t **do**
        Initialize empty sets $\vartheta_1 \leftarrow \emptyset$, $\vartheta_2 \leftarrow \emptyset$
        for each customer $i \in sorted\_customers$ **do**
          if customer i is projected to stockout in the current period **then**
            Add customer i to set ϑ_1
          else if customer i is projected to stockout in the next $look\_ahead$ periods **then**
            Add customer i to set ϑ_2
          end if
        end for
        Assign customers in ϑ_1 to routes, respecting delivery quantities to warehouse ($\tilde{q}^s_{wt}$) and vehicle capacities, using the nearest neighbor insertion heuristic
        Assign customers in ϑ_2 to routes, respecting $\tilde{q}^s_{wt}$ and vehicle capacities, using the minimum insertion cost method
        Update $\tilde{I}^s_{it}$ with the computed delivery quantities for each customer
      end for
      Update sol_{SE} with the second-echelon solution for this $look\_ahead$ value
      Increase $look\_ahead$ value by 1
    end while
  end for
end for
Combine sol_{FE} and sol_{SE} to update the overall solution sol
Fix all integer variables in sol
Solve the LP to improve production and delivery plan
Update sol with the improved solution
Return the best feasible solution sol

The mathematical formulation for the MW-PRP is presented below:

$$\text{Min} \sum_{t\in\mathcal{T}} \left(Fy_t + up_t + h_0 I_{0t} + \sum_{r\in\mathcal{R}} \tilde{c}_r o_{rt} + \sum_{s\in\mathcal{S}} \xi_s \Big(\sum_{w\in\mathcal{N}_w} \big(I^s_{wt} + \hat{\alpha}_w \hat{l}^s_{wt} \big) \Big) \right) \tag{3.30}$$

s.t. (3.2) - (3.4), (3.6) - (3.10), (3.18) - (3.20), and (3.23)

$$I^s_{wt} = I^s_{w,t-1} + \sum_{r\in\mathcal{R}} q_{wrt} - \hat{d}^s_{wt} + \hat{l}^s_{wt} \qquad \forall w \in \mathcal{N}_w, t \in \mathcal{T}, s \in \mathcal{S} \tag{3.31}$$

$$I^s_{it} \geq 0 \qquad \forall i \in \mathcal{N}_o \cup \mathcal{N}_w, t \in \mathcal{T}, s \in \mathcal{S} \tag{3.32}$$

$$\hat{l}^s_{wt} \geq 0 \qquad \forall w \in \mathcal{N}_w, t \in \mathcal{T}, s \in \mathcal{S}, \tag{3.33}$$

where $\hat{d}^s_{wt} = \sum\limits_{i\in\mathcal{C}^w} \big(d^s_{it} - (I_{i0} - \sum\limits_{l=1}^{t-1} d^s_{it})^+ \big)$ for warehouse w, period t and scenario s and $\hat{\alpha}_w = \frac{\sum_{i\in\mathcal{C}^w} \alpha_i}{|\mathcal{C}^w|}$ for each warehouse. The variable $\hat{l}^s_{wt}$ is used to indicate the unmet demand of each warehouse in a period and scenario. The variable $\tilde{q}^s_{wt} = \sum_{r\in\mathcal{R}} q_{wrt}$ represents the delivery amount to warehouse w in period t and scenario s. It is used as a parameter in the second phase of the algorithm to compute the warehouse inventory for each scenario.

After determining the production plan and the routes for the first echelon, we proceed to construct a feasible solution for the second echelon. It is important to emphasize that the second-echelon problem is always feasible, as shortages in customer demand are allowed. To generate a feasible solution for the second echelon, we utilize the available inventory at each warehouse obtained in the previous phase while accounting for the capacity of the vehicles located at these warehouses. The solution is derived in three phases. In the first phase, we identify the customers to be visited and estimate the delivery quantities. The second phase involves determining the delivery routes for the selected customers. These two phases are modeled similarly to the approach in Alvarez et al. (2018) with some changes. We also introduce a third phase in our approach, where the continuous variables are optimized by solving an LP, rather than being determined during the construction heuristic.

We begin by ranking customers according to a ratio that compares the cost of unmet demand with the approximate delivery cost per unit. The latter includes the unit production cost, the approximate fixed production cost per unit based on production capacity, the estimated delivery cost of a single unit from the plant to the nearest warehouse based on Q^p, and the estimated delivery cost from the warehouse to the customer based on Q^w. For a given customer i, assuming that warehouse w' is the closest, this relative shortage penalty ratio is computed as follows:

$$\frac{\alpha_i}{u + \frac{F}{C} + \frac{2c_{0,w'}}{Q^P} + \frac{2c_{w',i}}{Q^w}}. \tag{3.34}$$

Based on this ratio, we sort the customers in descending order of priority and define the set $sorted\_customers$. For each customer i, we then calculate the partial inventory levels for each time period and scenario by using the customer's demand, initial inventory, and the deliveries made to the customer. The partial inventory for each time period can be calculated as follows:

$$\tilde{I}_{it}^s = I_{i0} - \sum_{l=1}^{t} d_{il}^s + \sum_{l=1}^{t} \tilde{w}_{il}^s \qquad \forall i \in \mathcal{N}_c, t \in \mathcal{T}, s \in \mathcal{S}, \tag{3.35}$$

where $\tilde{w}_{it}^s$ is the delivery quantity to customer i, in period t and scenario s. In the first iteration of the algorithm, since no delivery is assigned to customers, we have:

$$\tilde{w}_{it}^s = 0 \qquad \forall i \in \mathcal{N}_c, t \in \mathcal{T}, s \in \mathcal{S}. \tag{3.36}$$

Assuming a warehouse w and scenario s, the heuristic is executed for each period $t \in \mathcal{T}$ to determine which customers are assigned to warehouse w. Two sets, ϑ_1 and ϑ_2, are defined. The first set contains customers with $\tilde{I}_{it}^s < 0$, who will face stockouts immediately in period t if not visited. Thus, these customers have a higher priority of being visited. The approximate delivery quantity to each customer in set ϑ_1 is calculated using:

$$\begin{cases} \tilde{w}_{it}^s = \min\{|(\tilde{I}_{it}^s)^-|, Q^w, L_i - (\tilde{I}_{i,t-1}^s)^+\}, & \forall i \in \vartheta_1,\ t \in \mathcal{T}\setminus\{T\},\ s \in \mathcal{S}, \\ \tilde{w}_{iT}^s = \min\{|(\tilde{I}_{iT}^s)^-|, Q^w\}, & \forall i \in \vartheta_1,\ s \in \mathcal{S} \end{cases} \tag{3.37}$$

After assigning customers to ϑ_1, we proceed to assign customers to the second set, ϑ_2, which consists of those projected to experience stockouts in the upcoming *look_ahead* periods ($\tilde{I}^s_{i,t+look\_ahead} < 0$). The approximate delivery quantity to each customer in set ϑ_2 is given by:

$$\tilde{w}^s_{it} = \min\{|(\tilde{I}^s_{i,t+look\_ahead})^-|, Q^w, L_i - (\tilde{I}^s_{i,t-1})^+\}, \quad \forall i \in \vartheta_2,\ t \in \mathscr{T} \backslash \{T\},\ s \in \mathscr{S}. \tag{3.38}$$

When adding customers to ϑ_1 and ϑ_2, we also take *sorted_customers* into account, since shortages are allowed and we aim to prioritize those with a higher penalty ratio for unmet demand. Customers are inserted into each set in rank order, starting with those having the highest relative penalty ratio, thereby increasing their likelihood of being visited. Customers from ϑ_1 may also appear in ϑ_2, in which case they are removed from ϑ_2 and their approximate delivery is updated according to (3.38).

Following the definition of ϑ_1 and ϑ_2, we apply the standard nearest neighbor insertion heuristic (Bräysy and Gendreau, 2005) to add customers from ϑ_1. For customers in ϑ_2, we use the minimum insertion cost criterion to integrate them into the routes. During the assignment of customers to routes, we also consider the remaining available inventory and vehicle capacities to ensure feasibility. Accordingly, the values of $\tilde{w}^s_{il}$ are updated if the available inventory at the warehouse or vehicle capacities do not allow full delivery of the initially estimated quantities. We start by assigning customers from ϑ_1 and then move to ϑ_2, stopping when capacity or available inventory prevents further deliveries.

At the end of each iteration, after assigning deliveries to customers, customer inventories are updated using (3.35). Once routes are established for all periods within the current scenario and warehouse, the *look_ahead* value is increased by 1 and all deliveries are reset to zero to assess whether accounting for additional periods improves the objective function. This process continues until $look\_ahead = \lceil \frac{T}{2} \rceil$, and the solution with the lowest objective value is selected. The same procedure is applied to all warehouses and scenarios, ultimately yielding a feasible solution for the entire problem.

In the third phase of this heuristic, we solve an LP for the entire S2EPRP-AR, where all integer variables are fixed, allowing for further improvements in continuous variables i.e., production and delivery quantities.

3.3.2 Iterated Local Search

The ILS metaheuristic iteratively improves an initial feasible solution by alternating between local search (intensification) to move toward local optima, and a perturbation algorithm to escape those local optima (diversification). At the end of each iteration, if a better solution is found, the incumbent solution is updated. After generating the initial solution and reassigning customers to warehouses, the solution is passed to the ILS for iterative improvement.

Algorithm 5 Iterated Local Search (ILS)

1: **Input:** sol_0
2: **Output:** Incumbent solution for the S2EPRP-AR
3: $sol \leftarrow$ RVND(sol_0)
4: **for** each scenario s **do**
5:     $sol' \leftarrow sol_{\text{SE}}^{(s)}$
6:     **while** Stopping criteria for ILS is not met **do**
7:         $sol'' \leftarrow perturb(sol')$
8:         $sol'' \leftarrow RVND(sol'')$
9:         **if** $objVal(sol'') < objVal(sol')$ **then**
10:             $sol' \leftarrow sol''$
11:         **end if**
12:     **end while**
13:     Update $sol_{\text{SE}}^{(s)}$ with sol'
14: **end for**
15: **return** sol

Since the ILS focuses only on the second-echelon routes and the first-stage binary variables are fixed, the problem can be decomposed into an independent subproblem for each scenario. Thus, the ILS is applied separately to each scenario (lines 4-5). In each iteration, the solution for a given scenario s undergoes a perturbation, which helps escape the current local optimum (line 7). After the perturbation, the solution is improved using a multi-start RVND local search algorithm (line 8). If a better solution is found, the solution for that scenario is updated, and the next iteration begins (line 9-11). This process continues until a stopping criterion is met. The pseudocode of the ILS is presented in algorithm 5.

Although the ILS in this study shares components with those used for the IRP in Alvarez et al. (2018), it differs in key aspects. Previous algorithms relied solely on ILS to optimize both integer and continuous variables. In contrast, our approach solves an LP after each local search operator is executed, which leads to better overall results, especially when accounting for the uncertainty in the problem. The ILS algorithm terminates if the number of iteration reaches $ILS\_TimeLimit$ or the computation time reaches $ILS\_ItertionLimit$.

Local Search

As mentioned earlier, the local search algorithm used in this study (line 3 of Algorithm 5) is a multi-start RVND heuristic (Mladenović and Hansen, 1997), where local search operators are selected randomly and applied to the current solution. The multi-start framework allows the algorithm to explore multiple neighborhoods, as the local optima may differ depending on the sequence and selection of local search operators. This diversification enhances the ability of the algorithm to escape poor-quality local optima. If the application of an operator improves the objective function value, the set of operators is reset to its initial state. Otherwise, the operator is removed from the set. This process continues until all operators have been exhausted. The pseudocode for the RVND procedure is provided in Algorithm 6.

The local search operators used within RVND primarily focus on modifying the routes from a selected warehouse to its assigned customers. Some operators also allow changes in customer-to-warehouse assignments, depending on the current values of the first-stage variables.

The operators used in the RVND are as follows:

- **Or-Opt**(v), $v \in 1,2,3$: Moves v adjacent customers from their current position to another position within the same route.
- **Route-Shift**(v), $v \in 1,2,3$: Selects v adjacent customers from a route and moves them to another route, either of the same warehouse or a different warehouse in the same period.

Algorithm 6 Randomized Variable Neighborhood Descent (RVND)

Input: Solution for scenario s
Output: Improved solution for scenario s
$sol^* \leftarrow sol_0^{(s)}$
$\mathscr{O} \leftarrow$ initialize the set of local search operators
$sol' \leftarrow sol^*$
while $|\mathscr{O}| > 0$ **do**
    Select a random local search operator o from $\mathscr{O}$
    Perform o on sol'
    if o improves sol' **then**
        $sol^* \leftarrow sol'$
        Reset $\mathscr{O}$ to the initial set of operators
    else
        Remove o from $\mathscr{O}$
    end if
end while
return sol^*

- **Period-Shift**: A customer is removed from its current visited route in one period and randomly inserted into another route in a different period.

- **Swap**(v_1, v_2), $v_1, v_2 \in 1,2$, $v_1 \geq v_2$: Exchanges v_1 customers from one route with v_2 customers from another route, either from the same warehouse or from different warehouses in the same period.

- **Remove**: A randomly selected customer is removed from a route.

- **Insert**: A customer that is not visited in period t is inserted into a randomly selected route from warehouse w at the minimum insertion cost position.

- **Merge**: For a customer visited in more than one period, two of the visited periods are merged into one, selected randomly.

- **Transfer**: A customer visited in multiple periods is removed from its visited periods and inserted into unvisited periods for a randomly selected warehouse, based on minimum insertion cost.

After applying an operator, we solve an LP with a fixed production plan and fixed routes for both echelons across all scenarios. If the resulting objective function value improves, the incumbent solution for that scenario is updated.

Perturbation

The perturbation step in ILS is crucial as it significantly enhances the diversification of the search. After applying a perturbation, the algorithm initiates the local search in the hope of exploring solutions beyond those identified in previous iterations. To achieve this, we define three types of perturbations for the proposed ILS, as follows:

- **Random Shift**: A customer is selected from a route and moved to another route, either from the same warehouse or from a different warehouse in the same period.
- **Random Insertion**: A customer that is not visited in period t is inserted into a randomly selected warehouse w at the minimum insertion cost position.
- **Random Removal**: A randomly selected customer is removed from a route.
- **Random Swap**: A randomly selected customer from one route is exchanged with another randomly selected customer from a route from the same or a different warehouse.

A maximum of *ILS_MaxPerturb* perturbations are allowed for each scenario, and the perturbation operators are selected randomly. After applying the perturbation, we solve the LP to evaluate the current value of the objective function before proceeding with the local search. The new solution is only accepted if it improves upon the previous solution after applying the local search.

3.4 Computational Experiments

The algorithms developed in this study were implemented in C++, and IBM ILOG CPLEX version 22.1.1 was used as solver to handle LP and MIP subproblems. All computational experiments were carried out on machines equipped with Intel Xeon Gold 6148 2.4 GHz processors and 32 GB of memory.

Due to the novelty of the problem addressed in this study, benchmark instances available in the literature are very limited. To the best of our knowledge, this is the first study to tackle this specific variant of the two-echelon production routing problem with demand uncertainty.

As a result, we generated a new set of test instances specifically designed for the S2EPRP-AR. These instances are made publicly available at Instances Repository. To generate the new dataset we used the benchmark instances introduced by Archetti et al. (2011), which are widely used in the classical PRP literature. However, the original instances do not include several key elements required in our setting, such as multiple echelons and stochastic demand. We therefore adapted and extended them to fit the structure and requirements of our problem.

We first generated instances for the deterministic version of the problem. These instances vary in the number of warehouses (2 to 5), number of customers (ranging from 20 to 50, in increments of 10), and vehicle configurations: 1 vehicle at both the plant and each warehouse, 2 vehicles at the plant and 2 vehicles at each warehouse, and 2 vehicles at the plant and 3 vehicles at each warehouse. Additionally, we considered the four classes of instances defined in Archetti et al. (2011), namely: standard cost structure, high production cost, high transportation cost, and no inventory cost at the customer level. Warehouse locations were randomly selected from within the customer set using a clustering-based approach. In total, 192 deterministic instances were generated.

A similar approach was used to generate instances for the stochastic problem. For these instances, we considered 2 to 5 warehouses when the number of customers was in $\{20, 25, 30\}$, and 2 and 3 warehouses when having $\{10, 15, 35, 40, 50\}$ customers. The same vehicle configurations as in the deterministic case were used. In total, 66 stochastic instances were generated. A detailed explanation of the instance generation process is provided in Online Supplements.

To incorporate demand uncertainty, we used Monte Carlo simulation to generate demand scenarios based on the average customer demands from the instances. In the standard setting, each instance includes 100 scenarios unless stated otherwise. For each customer i and period t, the demand in scenario s is sampled from a uniform distribution in the range $[\bar{d}_{it}(1-\delta), \bar{d}_{it}(1+\delta)]$, where $\bar{d}_{it}$ is the average demand and δ is the uncertainty level, ranging between 0 and 1. Each scenario $s \in \mathscr{S}$ is assigned an equal probability of $\xi_s = 1/S$.

Finally, the stochastic model includes a penalty cost α_i for each unit of unmet demand for customer i. This cost is calculated as the sum of the unit production cost, an estimate of the fixed cost per unit of production, and the approximate transportation cost of delivering a unit from the plant to the nearest warehouse to customer i, and then from the warehouse to the customer. The total is then multiplied by a penalty factor σ. The penalty is computed as:

Finally, the stochastic model includes a penalty cost α_i for each unit of unmet demand for customer i. This cost is calculated as the sum of the unit production cost, an estimate of the fixed cost per unit of production, and the approximate transportation cost of delivering a unit from the plant to the nearest warehouse to customer i, and then from the warehouse to the customer, each multiplied by two to reflect round trips. The total is then multiplied by a penalty factor σ. The penalty is computed as:

$$\alpha_i = \left\lceil \sigma\big(u + \frac{F}{C} + 2\frac{c_{w_i^*,i}}{Q^w} + 2\frac{c_{0,w_i^*}}{Q^P}\big) \right\rceil, \tag{3.39}$$

where w_i^* is the closest warehouse to customer i and σ is set to 50 as in Kermani et al. (2024), unless stated otherwise.

3.4.1 Deterministic Problem

As mentioned earlier, we employ the HHA algorithm to solve large instances of the deterministic 2EPRP. In the largest configuration, the model considers 5 warehouses, 50 customers, 2 vehicles at the plant and 3 vehicles at each warehouse, resulting in a total of 15 vehicles in the second echelon. Compared to the stochastic variant, the deterministic problem differs in two important aspects: there is no need to account for multiple scenarios, and shortages are no longer allowed. Therefore, the original formulation of the S2EPRP can be simplified. Specifically, the scenario index is removed from all variables and constraints, and the shortage variable l_{it}^s is eliminated. In addition, constraints 3.11 are replaced by the following deterministic version:

$$I_{it} = I_{i,t-1} + \sum_{k \in \bigcup_{w \in \mathscr{N}w} \mathscr{K}w} w_{ikt} - d_{it} \qquad \forall i \in \mathscr{N}_c, t \in \mathscr{T}. \quad (3.40)$$

The complete formulation of the deterministic problem is provided in Appendix C.2. To solve this problem, modifications are also made to the HHA algorithm to remove all iterations over scenarios.

The time limit for the HHA is fixed at 3600 seconds. The MW-PRP, which is solved only once to construct the initial solution, is assigned a time limit of 300 seconds with an optimality gap of 0.01%. The restricted 2EPRP, which is solved in every iteration of HHA, is given 60 seconds of computation time with a 1% optimality gap. This setting is adopted because reducing the optimality gap significantly increases the computational time, while it typically does not yield a proportional improvement in solution quality. All of these computations are performed using 10 threads.

The number of multi-starts for the ILS is set to 10. It is worth noting that parallel computing is incorporated into the deterministic version of the algorithm to enhance computational efficiency. Specifically, parallelism is applied within the multi-start RVND procedure, where 10 threads are used to execute multiple runs concurrently. The algorithm retains the best solution based on the objective function value. Although parallel computing is also employed in the stochastic version, its implementation is adapted to the scenario-based decomposition structure of that problem.

The ILS-specific parameters are set to $ILS\_MaxPerturb = 10$ and $ILS\_IterationLimit = 30$. A grid search was used to tune the parameters Max_NIG and Max_NIL, which were ultimately set to 25 and 5, respectively. The value of Max_NIG was selected from the candidate set $\{10, 15, 25, 50\}$, while Max_NIL was chosen from $3, 5, 10, 15$.

To assess the quality of the HHA solutions, we use a BC algorithm implemented in CPLEX to solve the 2EPRP, using the HHA solution as a warm start. Details of the BC algorithm are provided in Appendix C.4. The BC is run with a 7200 second time limit using 10 threads, and the optimality gap is set to 10^{-6}.

All 192 deterministic instances are defined over a planning horizon of 6 periods. To present the results, we include three tables, each corresponding to one of the aforementioned vehicle configurations. In each table, the number of warehouses (N_w) and the number of customers (N_c) are reported. Each row shows the average solution to four instances corresponding to the four problem classes of instance. Each table summarizes 64 instances in total.

Under the HHA section of the table, the column labeled **OF** shows the best objective function value found by the algorithm, **Gap** $(\%)$ presents the relative optimality gap with respect to the lower bound (LB) obtained from BC, and **CPU (secs)** indicates the computation time in seconds. Under the BC section, the **UB**, **LB**, and **CPU** time are reported. The **Gap** column shows the final optimality gap from BC. It is important to note that the UB from BC is always at least as good as that of HHA, since the HHA solution is provided as a warm start. The final column, labeled **Diff-UB∇**, shows the improvement in the UB after two hours of solving the problem with the BC algorithm.

The results of the first configuration, specifically one vehicle at the plant and one for each warehouse, are provided in Table 3.1. The average optimality gap is 4.38% with an average computation time of 496 seconds. The BC algorithm was able to improve the given UB by only 0.15% on average. This result highlights the performance of the HHA in finding high-quality solutions within a reasonable time. However, the optimality gap is also affected by the weak LB, showing the inability of the BC to close the gap even for the smallest instances.

Table 3.2 displays the results for the second configuration of instances. As expected, both the gap and computation time are higher, at 6.39% and 757 seconds, respectively. The improvement in the UB by BC is even lower at 0.04%, which shows how increasing the size of the problem has a drastic effect on the BC performance, while the HHA is still able to provide solutions in less than 15 minutes.

Finally, Table 3.3 illustrates the results of the largest instance configuration, showing

Table 3.1: Computational results for the deterministic problem with one vehicle at the plant and one vehicle per warehouse.

N_w	N_c	# Ins	HHA OF	HHA Gap (%)	HHA CPU (secs)	BC UB	BC LB	BC CPU (secs)	BC Gap (%)	Diff-UB∇ (%)
2	20	4	143,098	1.88	279.94	142,372	141,387	7,205.92	1.09	0.80
	30	4	190,141	2.07	293.35	189,758	187,911	6,440.05	1.72	0.35
	40	4	239,975	2.77	507.42	239,674	235,954	5,316.52	2.58	0.21
	50	4	282,133	3.29	564.94	281,961	276,997	6,919.66	3.25	0.04
3	20	4	145,713	5.87	291.38	145,526	140,343	5,374.16	5.70	0.20
	30	4	190,852	4.20	345.36	190,816	186,101	3,491.56	4.17	0.04
	40	4	241,380	5.49	364.53	241,271	233,795	5,033.68	5.41	0.09
	50	4	285,557	4.58	776.82	285,225	277,960	4,263.03	4.43	0.17
4	20	4	144,865	5.39	331.61	144,703	140,000	4,930.37	5.16	0.25
	30	4	192,886	5.05	373.03	192,869	186,922	5,234.71	5.04	0.01
	40	4	241,946	4.32	659.30	241,894	235,925	6,714.82	4.28	0.04
	50	4	286,906	5.14	1,086.24	286,895	278,272	7,209.52	5.14	0.01
5	20	4	143,044	4.32	297.82	142,966	139,186	4,431.63	4.28	0.05
	30	4	191,214	4.93	516.39	191,090	185,390	6,010.52	4.86	0.08
	40	4	243,590	5.32	707.10	243,492	235,723	6,821.45	5.23	0.09
	50	4	287,339	5.48	542.48	287,206	278,270	7,026.67	5.43	0.06
Average		64	215,665	4.38	496.11	215,482	210,008	5,776.52	4.23	0.15

an average gap of 7.48% and an average computation time of 1005 seconds. The Diff-UB∇ remains very low at 0.03%, confirming that while BC struggles to improve the solutions, the HHA continues to deliver good solutions within a reasonable time frame.

Table 3.4 presents the average costs associated with each component of the deterministic problem, broken down by instance class and by the number of vehicles per plant and warehouse. Under **First Echelon**, the average total cost of the first echelon, as well as the average production, plant inventory, and transportation costs, are presented, respectively. Under **Second Echelon**, the total cost of the second echelon is shown, followed by warehouse inventory, customer inventory, and second-echelon transportation costs. The second class of instances, characterized by high production costs, incurs the highest total costs across all configurations. This is largely driven by the high setup costs, which significantly influence the objective function. Note that the production cost (Prod) includes both fixed setup costs and variable production costs, and in this class, both components are substantially larger than in other classes.

Table 3.2: Computational results for the deterministic problem with two vehicles at the plant and two vehicles per warehouse.

N_w	N_c	# Ins	HHA OF	HHA Gap (%)	HHA CPU (secs)	BC UB	BC LB	BC CPU (secs)	BC Gap (%)	Diff-UB∇ (%)
2	20	4	144,930	3.90	395.15	144,928	141,544	5,491.38	3.89	0.00
	30	4	194,312	5.18	384.13	194,289	188,270	4,087.31	5.15	0.03
	40	4	242,878	4.76	503.53	242,861	236,281	4,488.20	4.76	0.01
	50	4	286,078	5.78	667.73	286,067	276,840	3,346.37	5.77	0.01
3	20	4	145,911	7.18	449.93	145,735	139,032	3,988.31	7.00	0.20
	30	4	192,038	6.21	461.72	191,983	184,729	4,631.70	6.16	0.05
	40	4	242,137	6.51	816.61	242,101	233,115	7,069.14	6.47	0.04
	50	4	289,535	8.21	803.95	289,473	276,326	5,132.83	8.17	0.04
4	20	4	144,891	6.09	452.91	144,864	139,534	4,265.45	6.08	0.01
	30	4	193,437	6.59	790.12	193,423	185,842	5,112.01	6.58	0.02
	40	4	243,457	6.02	978.18	243,420	235,040	7,209.37	5.99	0.03
	50	4	290,204	9.12	1,021.84	290,174	275,819	5,829.16	9.09	0.03
5	20	4	144,097	4.76	644.42	144,019	139,808	4,648.82	4.72	0.05
	30	4	192,235	6.54	793.02	192,235	184,796	7,105.51	6.54	0.00
	40	4	242,657	7.42	1,045.21	242,591	232,629	7,111.09	7.36	0.06
	50	4	288,275	7.93	1,910.22	288,275	275,856	7,210.29	7.93	0.00
Average		64	217,317	6.39	757.42	217,277	209,091	5,420.43	6.35	0.04

Table 3.3: Computational results for the deterministic problem with two vehicles at the plant and three vehicles per warehouse.

N_w	N_c	# Ins	HHA OF	HHA Gap (%)	HHA CPU (secs)	BC UB	BC LB	BC CPU (secs)	BC Gap (%)	Diff-UB∇ (%)
2	20	4	146,586	5.71	252.5	146,586	141,324	4,291.7	5.71	0.00
	30	4	194,790	5.78	389.9	194,582	187,979	4,674.9	5.62	0.17
	40	4	242,902	4.68	801.6	242,897	236,257	7,074.1	4.68	0.00
	50	4	290,089	9.12	1,115.8	290,058	275,234	5,395.5	9.11	0.01
3	20	4	146,865	8.40	490.5	146,839	138,923	4,925.1	8.39	0.01
	30	4	192,004	6.51	693.6	191,873	184,499	5,891.8	6.43	0.09
	40	4	243,081	7.88	835.5	243,054	231,981	6,718.3	7.86	0.02
	50	4	289,287	9.41	1,350.3	289,199	274,362	6,653.0	9.37	0.04
4	20	4	145,026	6.37	584.2	144,993	139,465	5,134.0	6.36	0.01
	30	4	193,764	6.88	889.8	193,747	185,627	6,080.4	6.86	0.02
	40	4	244,220	8.35	1,043.0	244,184	232,741	7,209.4	8.33	0.02
	50	4	290,915	10.70	1,922.0	290,915	270,424	7,202.3	10.70	0.00
5	20	4	143,953	4.77	871.6	143,950	139,735	5,829.4	4.77	0.00
	30	4	192,332	7.71	1,262.5	192,327	183,874	7,208.0	7.71	0.01
	40	4	241,414	7.47	1,674.8	241,403	231,678	7,205.9	7.47	0.00
	50	4	288,971	9.95	1,902.8	288,971	269,432	7,212.4	9.95	0.00
Average		64	217,887	7.48	1,005.0	217,849	207,721	6,169.1	7.46	0.03

Table 3.4: Breakdown of costs across different components of the problem

K_p	N_w	Class	Total	First Echelon				Second Echelon			
				FE Total	Prod	Inv Plt	Trn FE	SE Total	Inv Whs	Inv Cus	Trn SE
1	1	Class 1	89,669	58,748	52,425	2,882	3,442	30,921	1,076	19,651	10,194
		Class 2	559,995	529,292	522,375	3,172	3,745	30,703	902	19,610	10,191
		Class 3	143,869	70,821	52,988	1,694	16,140	73,048	2,290	20,518	50,240
		Class 4	68,395	57,533	52,425	1,805	3,303	10,862	864	0	9,998
Average			215,482	179,099	170,053	2,388	6,657	36,384	1,283	14,945	20,156
2	2	Class 1	90,744	59,640	52,425	2,950	4,265	31,104	1,056	19,455	10,592
		Class 2	560,930	529,488	522,375	3,051	4,062	31,443	1,117	19,533	10,792
		Class 3	147,654	71,987	52,800	1,645	17,542	75,668	2,993	20,579	52,095
		Class 4	69,781	58,544	52,425	2,119	4,000	11,237	869	0	10,368
Average			217,277	179,915	170,006	2,441	7,467	37,363	1,509	14,892	20,962
2	3	Class 1	90,899	59,437	52,238	3,048	4,152	31,462	1,131	19,541	10,790
		Class 2	561,230	529,584	522,375	2,993	4,216	31,646	1,328	19,454	10,864
		Class 3	149,386	72,096	52,800	1,685	17,611	77,290	3,373	20,685	53,233
		Class 4	69,879	58,310	52,425	1,910	3,975	11,568	1,071	0	10,497
Average			217,849	179,857	169,959	2,409	7,488	37,992	1,726	14,920	21,346

3.4.2 Deterministic Problem with Cross-docking Satellites

In this section, we solve the deterministic 2EPRP with cross-docking satellites (2EPRP-CS), similar to the problem introduced in the study by Qiu et al. (2021). We attempted to obtain the instances used by Qiu et al., 2021, however, despite our efforts, we were unable to access those instances. The main difference between the two variants is that, instead of warehouses that can store products for future distribution, the cross-docking satellites act as distribution hubs where storage is not possible. For simplicity, we use the same notation for satellites as for warehouses. Instead of a unit holding cost, a unit handling cost $\tilde{h}_w$ is defined for each satellite $w \in \mathscr{N}_w$. We compute the handling cost as in Qiu et al. (2021), where $\tilde{h}_w = \theta h_0$, with $\theta = 0.1$ and h_0 representing the unit holding cost at the plant. The mathematical formulation of the 2EPRP-CS is given in Appendix C.3.

To solve the problem, we adapt the algorithm used for the 2EPRP. The structure of the algorithm, as well as the parameter settings and instance configurations, remain the same, while the mathematical formulations are updated to address the 2EPRP-CS wherever needed. Tables 3.5, 3.6, and 3.7 present the results for the three possible vehicle configurations at the plant and warehouses, consistent with those reported in the previous section.

The HHA was able to solve all instances of the first vehicle configuration for the

Table 3.5: Computational results for the 2EPRP-CS with one vehicle at the plant and one vehicle per warehouse.

N_w	N_c	# Ins	HHA OF	HHA Gap (%)	HHA CPU (secs)	BC UB	BC LB	BC CPU (secs)	BC Gap (%)	Diff-UB∇ (%)
2	20	4	151,118	4.10	269.03	142,806	142,008	7,203.08	0.95	3.18
	30	4	191,005	2.07	315.36	190,749	188,684	5,601.15	1.81	0.27
	40	4	240,107	2.02	515.70	239,850	237,120	5,040.86	1.85	0.17
	50	4	285,942	4.68	553.23	281,181	277,549	5,996.36	2.48	2.30
3	20	4	146,330	6.12	501.42	145,901	140,938	5,589.63	5.49	0.65
	30	4	193,534	5.24	456.44	192,995	187,224	4,712.68	5.04	0.21
	40	4	242,705	5.75	500.37	242,671	234,868	5,441.87	5.74	0.01
	50	4	287,625	4.89	921.99	287,338	279,572	6,076.69	4.71	0.20
4	20	4	156,182	10.87	216.87	147,846	140,624	4,599.00	7.97	2.99
	30	4	195,382	6.07	762.54	195,376	187,905	5,457.74	6.07	0.00
	40	4	243,884	5.42	621.63	243,821	236,284	6,628.21	5.37	0.06
	50	4	290,609	6.60	969.71	290,498	279,484	6,060.21	6.51	0.09
5	20	4	154,501	10.90	159.12	147,324	139,510	5,256.21	8.26	2.70
	30	4	194,297	6.90	428.98	194,287	186,188	6,780.14	6.90	0.01
	40	4	244,455	5.58	530.24	244,387	236,340	7,148.48	5.54	0.05
	50	4	289,341	6.65	802.64	289,268	278,576	7,204.13	6.64	0.02
Average		64	219,189	5.86	532.83	217,269	210,805	5,924.78	5.08	0.81

2EPRP-CS. By comparing the results of the 2EPRP and 2EPRP-CS, one can observe that the OF is higher for the latter. This is mainly due to the inability to store products at the warehouse for future deliveries. With storage not permitted at cross-docking satellites, all deliveries to satellites must be forwarded to customers within the same period. Inventory costs are typically lower at warehouses than customers. Additionally, the limited capacity of second-echelon vehicles may prevent full utilization of the first-echelon vehicles' capacity. As a result, more frequent deliveries are required in both echelons, leading to higher delivery costs and, consequently, a higher OF value.

Moreover, the optimality gap in this setting is slightly higher than in the standard 2EPRP, at 5.86%, indicating the increased complexity of the problem. Nonetheless, the Diff_UB∇ remains at 0.81%, which again demonstrates the strong performance of the HHA.

For the other two vehicle configurations of the 2EPRP-CS, a significant observation is that some instances could not be solved. Specifically, the cases with 3 satellites and 40 or 50 customers, as well as those with 5 satellites and 50 customers, could not be

Table 3.6: Computational results for the 2EPRP-CS with two vehicles at the plant and two vehicles per warehouse.

N_w	N_c	# Ins	HHA OF	HHA Gap (%)	HHA CPU (secs)	BC UB	BC LB	BC CPU (secs)	BC Gap (%)	Diff-UB∇ (%)
2	20	4	148,303	7.42	368.19	147,612	141,916	3,046.19	6.32	1.16
	30	4	195,811	5.67	480.88	195,613	188,876	4,099.09	5.48	0.21
	40	4	244,395	5.05	574.68	244,328	237,132	6,580.58	5.01	0.05
	50	4	288,480	6.72	729.94	288,455	277,487	5,658.91	6.69	0.03
3	20	4	155,115	10.34	877.13	144,836	138,623	4,141.77	6.49	4.24
	30	4	193,445	7.19	683.50	193,351	184,994	5,529.70	7.12	0.09
	40	4	-	-	-	-	-	-	-	-
	50	4	-	-	-	-	-	-	-	-
4	20	4	157,011	13.40	459.85	148,340	138,547	4,029.71	9.99	3.59
	30	4	195,212	8.60	1,113.31	195,212	184,877	6,426.97	8.60	0.00
	40	4	245,815	7.98	1,239.66	245,662	233,994	7,210.12	7.91	0.08
	50	4	293,211	10.62	2,009.55	293,205	275,505	7,211.43	10.62	0.00
5	20	4	155,453	13.31	314.79	148,345	137,978	4,322.77	11.54	1.79
	30	4	194,865	8.90	943.21	194,863	184,181	6,672.78	8.90	0.00
	40	4	244,405	9.15	1,554.20	244,393	231,765	7,204.96	9.15	0.00
	50	4	-	-	-	-	-	-	-	-
Average		52	208,579	8.79	872.99	206,478	196,606	5,548.85	7.98	0.86

Table 3.7: Computational results for the 2EPRP-CS with two vehicles at the plant and three vehicles per warehouse.

N_w	N_c	# Ins	HILS OF	HILS Gap (%)	HILS CPU (secs)	BKS UB	BKS LB	BKS CPU (secs)	BKS Gap (%)	Diff-UB∇ (%)
2	20	4	147,486	6.25	644.04	147,016	141,784	4,391.86	5.53	0.76
	30	4	195,705	5.96	547.66	195,693	188,429	5,357.98	5.94	0.02
	40	4	246,684	6.34	953.32	246,583	237,145	6,353.92	6.28	0.07
	50	4	291,802	8.35	1,503.41	291,801	277,196	6,149.98	8.35	0.00
3	20	4	144,917	7.11	1,700.52	144,872	138,428	5,829.22	7.05	0.07
	30	4	194,120	7.87	1,065.78	194,120	184,725	6,673.11	7.87	0.00
	40	4	-	-	-	-	-	-	-	-
	50	4	-	-	-	-	-	-	-	-
4	20	4	156,589	13.39	700.39	150,192	138,364	4,854.37	11.67	1.84
	30	4	195,930	9.66	1,115.02	195,930	184,422	6,390.98	9.66	0.00
	40	4	245,596	10.06	1,734.18	245,596	231,498	7,209.96	10.06	0.00
	50	4	293,978	13.12	2,073.00	293,953	268,143	7,154.80	13.11	0.00
5	20	4	154,917	12.51	836.33	148,751	137,814	6,354.49	10.81	1.80
	30	4	195,267	9.88	1,487.29	195,267	183,634	7,205.81	9.88	0.00
	40	4	244,940	10.39	2,186.08	244,940	230,318	7,204.38	10.39	0.00
	50	4	-	-	-	-	-	-	-	-
Average		52	208,302	9.30	1,272.85	207,286	195,531	6,240.83	8.97	0.35

handled by the HHA. This failure is attributed to insufficient vehicle capacity to meet customer demand, primarily due to the absence of storage capability at the satellites, which otherwise could have been used to buffer deliveries for future periods.

Interestingly, the BC algorithm also failed to find any feasible solution for these instances and did not terminate with an infeasibility status. For the remaining instances, we again observe a slightly higher OF value and optimality gap compared to the standard 2EPRP. Nevertheless, the Diff_UB∇ remains low at 0.86% and 0.35% for the second and third vehicle configurations, respectively.

It is also worth highlighting that the largest instance solved by Qiu et al. (2021) consisted of 2 satellites and 35 customers over 3 periods, with 2 vehicles in the first and 4 in the second echelon. In contrast, we solved instances as large as 5 warehouses and 40 customers over 6 periods, with 2 vehicles in the first echelon and 15 in the second echelon. The average optimality gap for this configuration was 10.39%, while the BC algorithm was unable to improve the UB provided by the HHA.

3.4.3 Stochastic Problem

In this section, we present the results of the experiments for the stochastic version of the problem. All parameters are set similarly to those in the deterministic case. The key difference is that, since the second echelon of the problem is scenario-decomposable, we leverage this structure by solving scenarios in parallel using 40 CPU threads, enabling up to 40 scenarios to be processed simultaneously. For the experiments in this section, the number of scenarios S is set to 100, and the uncertainty level δ is set to 0.2, unless stated otherwise.

After solving the problem with the HHA, the resulting solution is used as a warm start for the BC algorithm. However, the LBs provided by the BC are generally weak; thus, we use the LB of the Wait-and-See (WS) problem as a valid LB to the recourse problem (RP). While this LB is theoretically weak as well, it consistently outperforms the one obtained from the BC. This highlights one of the main challenges in solving this problem, which is to find a strong LB.

Table 3.8 summarizes the main findings. The first five columns show the number of warehouses, the number of vehicles at the plant, the number of vehicles at each warehouse, the number of scenarios, and the number of instances for each configuration. Column **CPU-HHA** reports the average CPU time for the HHA, and column **CPU-BC** displays the computation time for the BC algorithm. We should highlight that building the initial solution from the given warm start may take anywhere from several minutes to several hours in CPLEX, due to the large scale of the problem. The two-hour computation limit is applied only after the initial solution is built. Therefore, the reported CPU time often exceeds two hours, reflecting the total time required for both building the initial solution and solving the problem with BC. Column **Gap** provides the average relative optimality gap in %, based on the best UB and LB obtained for the problem. Column **Diff-UB**∇ shows the average improvement in the UB of the HHA after solving the problem with BC.

To evaluate the value of the stochastic solution, we report the Value of the Stochastic Solution (VSS) and the Expected Value of Perfect Information (EVPI), both expressed as percentages. However, optimality is not guaranteed for the RP nor for the evaluation problems, i.e., the expected value of the expected value (EEV) problems and the WS problems. Thus, the VSS^{LB}, VSS^{UB}, and EVPI^{UB} are reported as ranges around their true values, which we cannot compute exactly. These values are calculated as follows:

$$\text{VSS}^{\text{LB}} = \frac{\text{EEV}^{\text{LB}} - \text{RP}^{\text{UB}}}{\text{EEV}^{\text{LB}}} \tag{3.41}$$

$$\text{VSS}^{\text{UB}} = \frac{\text{EEV}^{\text{UB}} - \text{RP}^{\text{LB}}}{\text{EEV}^{\text{UB}}} \tag{3.42}$$

$$\text{EVPI}^{\text{UB}} = \frac{\text{RP}^{\text{UB}} - \text{WS}^{\text{LB}}}{\text{RP}^{\text{UB}}}, \tag{3.43}$$

where EEV^{LB} and EEV^{UB} are the LB and UB to the EEV problem, and WS^{LB} is the LB to the WS problem. Note that the value of WS^{LB} is similar to the problem's gap, as WS^{LB} provides the LB to the problem in all cases. The EVPI^{LB} is also not reported, as calculating it using a similar approach would result in negative values, which are lower

Table 3.8: Computational results for the stochastic problem

N_w	K_p	K_w	S	# Ins	CPU-HHA (secs)	CPU-BC (secs)	Gap (%)	Diff-UB∇ (%)	VSS^{LB} (%)	VSS^{UB} (%)	$EVPI^{UB}$ (%)
	1	1	100	8	3,587.8	7,868.5	14.8%	0.0%	17.9%	31.1%	14.8%
2	2	2	100	8	3,638.0	8,993.8	14.7%	0.0%	18.8%	33.1%	14.7%
	2	3	100	8	3,640.1	7,643.7	15.7%	0.0%	16.7%	33.2%	15.7%
	1	1	100	8	3,430.5	8,065.5	19.1%	0.0%	13.6%	32.6%	19.1%
3	2	2	100	8	3,587.3	7,996.4	19.3%	0.0%	11.8%	32.3%	19.3%
	2	3	100	8	3,688.4	7,871.7	21.9%	0.0%	11.5%	34.3%	21.9%
	1	1	100	3	3,656.0	7,963.5	23.0%	0.0%	9.9%	36.2%	23.0%
4	2	2	100	3	3,682.3	9,160.0	26.6%	0.0%	4.0%	34.6%	26.6%
	2	3	100	3	3,678.6	8,471.8	28.6%	0.0%	5.1%	34.5%	28.6%
	1	1	100	3	3,668.2	10,670.4	23.6%	0.0%	10.3%	39.3%	23.6%
5	2	2	100	3	3,688.6	10,670.4	22.8%	0.0%	4.4%	34.4%	22.8%
	2	3	100	3	3,680.0	16,631.7	24.6%	0.0%	4.2%	37.1%	24.6%
Average			100	66	3,617.2	8,563.6	19.4%	0.0%	12.7%	33.6%	19.4%

than its natural LB of zero.

As seen in Table 3.8, the average HHA computation time remains around 1 hour, indicating that the time limit was the primary stopping criterion. The BC algorithm, on the other hand, requires approximately 2.5 hours on average. Despite receiving the HHA solution as a warm start, the BC fails to improve the given solution in any of the instances, resulting in an average improvement of 0.0%. The average optimality gap is around 19.4%, mainly due to the lack of a strong LB.

The reported average VSS ranges from 12.7% to 33.6%, demonstrating the added value of solving the stochastic model instead of relying on expected-value decisions. While the LB of VSS tends to decrease for larger instances due to weaker LBs in the EEV problems, the UB remains above 30% in most cases. The EVPI is also reported at 19.1%, which corresponds to the LB of the problem since we used the WS solution to estimate it. While this number is relatively large, we expect the true EVPI to be smaller, as the optimality gap in the evaluation problems introduces overestimation.

Figure 3.2 displays the average of the best obtained UBs for the stochastic problems, compared to the UBs for the WS and EEV problems, across different numbers of warehouses. Only instances with a similar number of customers are used in this figure; that is, only instances with 20, 25, or 30 customers are shown. Based on the figure, we observe

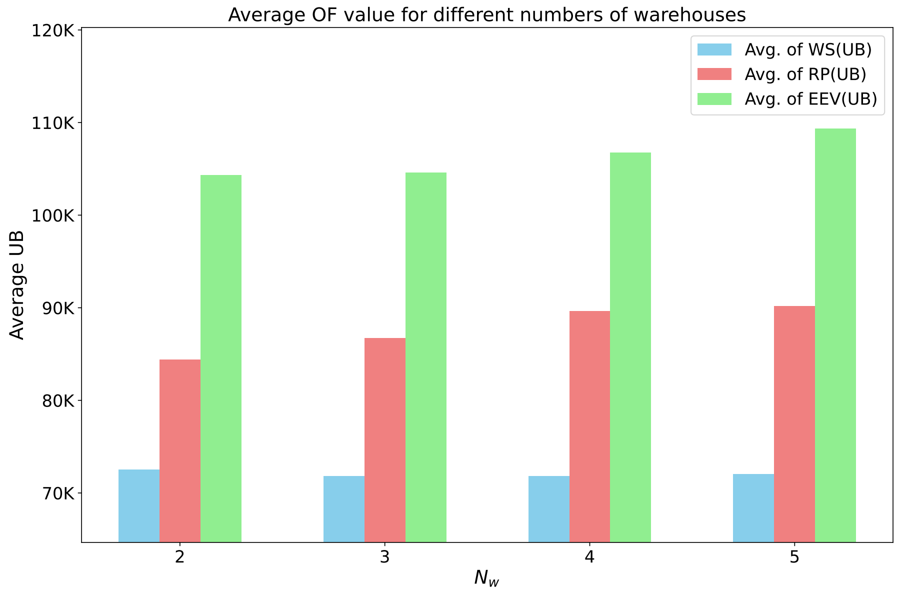


Figure 3.2: Objective function values for different numbers of warehouses

that the objective function value slightly increases as the number of warehouses increases. This is mainly due to higher transportation costs resulting from having more vehicles with lower capacity in the second echelon of the problem.

To provide further insight into the role of customer-to-warehouse assignments, Table 3.9 reports related results. Column **NW** shows the average number of distinct warehouses that serve each customer across all periods and scenarios. A value close to 1 indicates that allowing flexibility in customer-to-warehouse assignments would have little impact on the solution. In contrast, values closer to $\mathcal{N}_w$ suggest that ignoring this flexibility would increase the OF value. The results show that NW is 2 when there are two warehouses, and it increases as the number of warehouses grows. This indicates that, for each customer, there usually is at least one period and scenario in which they are served by a warehouse other than their most frequently assigned warehouse.

We introduce an additional metric to capture the relative importance of warehouses to each customer. The importance metric is computed as the proportion of periods and scenarios ($T \times S$) in which a customer is served by each warehouse. For each customer, we first calculate the importance of all warehouses and rank them in descending order.

Table 3.9: Customer-to-warehouse assignment across periods and scenarios

$\mathcal{N}_w$	#Ins	NW	Most Frequent (%)	Second Most Frequent (%)	Third Most Frequent (%)	Fourth Most Frequent (%)	Fifth Most Frequent (%)	Not Visited (%)
2	9	2.00	55.1%	5.0%	-	-	-	39.9%
3	9	2.93	53.9%	6.0%	1.9%	-	-	38.2%
4	9	3.65	54.1%	4.9%	1.9%	0.6%	-	38.5%
5	9	4.19	55.9%	4.2%	1.2%	0.6%	0.3%	37.8%
Average	36	3.19	54.8%	5.1%	1.7%	0.6%	0.3%	38.6%

Then the average of these values across all customers provide a measure of warehouse importance that is relative to each customer rather than tied to a specific warehouse ID.

The results indicate that, on average, when there are two warehouses, customers are served by their most frequent warehouse 55.1% of the time and by their second most frequent warehouse 5.0% of the time. Note that in some periods, certain customers are not visited by any warehouse. This needs to be accounted for to ensure that the percentages sum to 100%. Interestingly, the importance of the most frequent warehouse, as well as the percentage of periods in which customers are not visited, remains relatively stable as the number of warehouses increases. In contrast, the importance of the second most frequent warehouse declines, with its share redistributed among the other warehouses. This trend occurs because having more warehouses increases the likelihood that some of them are geographically closer to a specific customer, allowing the customer to be served by different warehouses in different periods or scenarios. Even if a warehouse is farther than the most frequent one, serving the customer from that warehouse in certain cases can help reduce overall costs by balancing capacity utilization and improving routing efficiency.

3.4.4 Sensitivity Analysis

In this section, we provide a more detailed analysis of the stochastic problem by discussing how changing parameters such as the uncertainty level and the number of scenarios can affect the solution. To this end, we consider different ranges of uncertainty levels to observe how increasing stochasticity influences the problem. We also solve the problem under various numbers of scenarios to examine how different sample sizes impact the

Table 3.10: Sensitivity analysis for different uncertainty ranges

δ	N_w	S	# Ins	CPU-HHA (secs)	CPU-BC (secs)	Gap (%)	Diff-UB∇ (%)	VSS$^{\text{LB}}$ (%)	VSS$^{\text{UB}}$ (%)	EVPI$^{\text{UB}}$ (%)
0.2	3	100	3	3,704.5	8,522.9	20.7%	0.0%	10.9%	32.9%	20.7%
	4	100	3	3,682.3	12,588.9.0	26.6%	0.0%	4.0%	34.6%	26.6%
0.4	3	100	3	3,671.1	8,623.1	22.1%	0.0%	31.1%	48.3%	22.1%
	4	100	3	3,678.0	8,785.1	26.7%	0.0%	25.6%	49.6%	26.7 %
0.6	3	100	3	3,661.0	8,688.8	25.9%	0.0%	42.9%	58.7%	25.9%
	4	100	3	3,697.7	8,750.9	32.7%	0.0%	35.0%	58.6%	32.7%
0.8	3	100	3	3,726.9	8,630.9	32.1%	0.0%	48.8%	68.3%	32.1%
	4	100	3	3,732.4	9,034.5	41.9%	0.0%	44.8%	66.5%	41.9%
Average		100	24	3,694.2	9252.7	28.6%	0.0%	29.8%	51.6%	28.6%

obtained solution.

For the sensitivity analysis, we use a subset of the introduced instances: those with 3 or 4 warehouses, 20, 25, and 30 customers, 2 vehicles at the plant, and 2 vehicles at each warehouse. The main reason for conducting experiments on a subset of the instances is that solving the problems, especially the evaluation problems, is computationally expensive, which prevents us from running experiments on the entire dataset. Therefore, we selected instances that are representative of the broader set while still allowing us to extract insights regarding various parameter settings. The number of instances used in this part is 6, and the following sections provide further details on the sensitivity analysis.

Uncertainty Level

In this section, we present the results of the problem under various uncertainty levels δ to demonstrate how different levels of uncertainty could affect the results. The uncertainty levels are set to 0.2, 0.4, 0.6, and 0.8, and as mentioned before, instances with 3 and 4 warehouses and 20, 25, and 30 customers were used, which led to a total of 6 instances per each uncertainty level and 24 instances in total. Table 3.10 presents the results for this analysis. The optimality gap is larger for higher levels of uncertainty; however, this is mainly because WS$^{\text{LB}}$ is used as the LB of the problem, and it decreases as uncertainty increases, which results in a larger calculated gap.

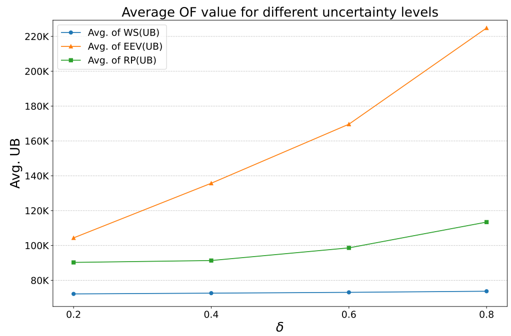
The results demonstrate higher VSS in environments with higher uncertainty, showing an average of 46.8% for the VSS^{LB} when $\delta = 0.8$. This shows the significance of considering uncertainty when demand volatility is high. Even in environments with moderate uncertainty, it is essential to take stochasticity into account, as the average VSS ranges between 28.4% and 49.0% with δ set to 0.4. This section shows the importance of incorporating stochasticity despite the difficulties of solving such a complex problem and the existing limitations in guaranteeing a near-optimal solution. The results show that even with the existing limitations, using HHA for solving the S2EPRP could result in significant cost reductions and improved performance.

Figure 3.3 provides more details on the OF value under different uncertainty levels for the RP as well as the WS and EEV. As shown in Figure 3.3a, the WS stays relatively stable even for higher uncertainty levels, which shows the total cost if perfect information were available and no uncertainty was present. On the other hand, the RP increases in more uncertain environments, while the highest impact is on the EEV, where the first-stage decisions are based on the mean value of demands. This highlights the significance of considering stochasticity in the problem, especially in environments with high volatility.

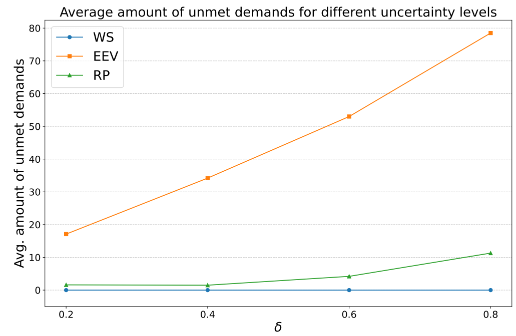
Figure 3.3b illustrates the number of units of unmet demand under different uncertainty levels, following a pattern similar to that of the OF values. One notable observation is that the number of units of unmet demand remains zero in all cases for the WS problem, as expected, since the demand is known in advance. For the RP, the average amount of unmet units stays close to zero in low-uncertainty environments and does not exceed 11.3 units even at the highest uncertainty level. In contrast, solving the deterministic problem using average demand can lead to significantly more unmet demand, ranging from 17.1 to 78.5 units on average, which can dramatically affect the overall costs.

Number of Scenarios

In this section, we provide the sensitivity analysis of the stochastic problem under different numbers of scenarios. The aim is to assess how the solution is affected when the sample size changes. In addition to the standard 100-scenario case used throughout the paper, we also solve the problem with 20, 50, and 200 scenarios. While increasing the



(a) Avg. UB



(b) Avg. amount of unmet demands

Figure 3.3: Average UB and amount of unmet demands for different uncertainty levels

Table 3.11: Sensitivity analysis for different numbers of scenarios

S	N_w	δ	# Ins	CPU-HHA (secs)	CPU-BC (secs)	Gap (%)	Diff-UB∇ (%)	VSS^{LB} (%)	VSS^{UB} (%)	EVPI^{UB} (%)
20	3	0.2	3	2,304.8	7,205.3	17.9%	0.1%	9.6%	27.8%	17.9%
	4	0.2	3	2,900.3	7,206.7	19.7%	0.1%	4.8%	34.9%	19.7%
50	3	0.2	3	3,333.2	7,212.4	18.8%	0.0%	10.0%	30.7%	18.8%
	4	0.2	3	3,415.2	7,217.1	23.1%	0.0%	4.6%	37.7%	23.1%
100	3	0.2	3	3,704.5	8,522.9	20.7%	0.0%	10.9%	32.9%	20.7%
	4	0.2	3	3,682.3	12,588.9	26.6%	0.0%	4.0%	34.6%	26.6%
200	3	0.2	3	3,771.4	11,703.3	21.5%	0.0%	12.8%	27.5%	21.5%
	4	0.2	3	3,699.9	-	23.8%	-	7.7%	46.2%	23.8%
Average		0.2	24	3,351.4	8,503.3	21.5%	0.0%	16.6%	37.3%	21.5%

number of scenarios is generally expected to improve solution quality, it also drastically increases the problem's complexity, creating a computational limitation that may prevent the problem from being solved. Thus, the goal of this analysis is to investigate how the solution behaves under varying sample sizes.

For this part, we use the same data configurations as in the previous section, with a total of 24 instances. A summary of the results is provided in Table 3.11. While the BC algorithm is provided with the HHA solution as a warm start, for instances with 200 scenarios and 4 warehouses, CPLEX was unable to build the initial solution within the time limit. As a result, the BC algorithm was not even started for any instance in this configuration.

For the case with 20 scenarios, we observe a slight improvement in the OF after solving the problem with the BC algorithm. However, this improvement is small and high-

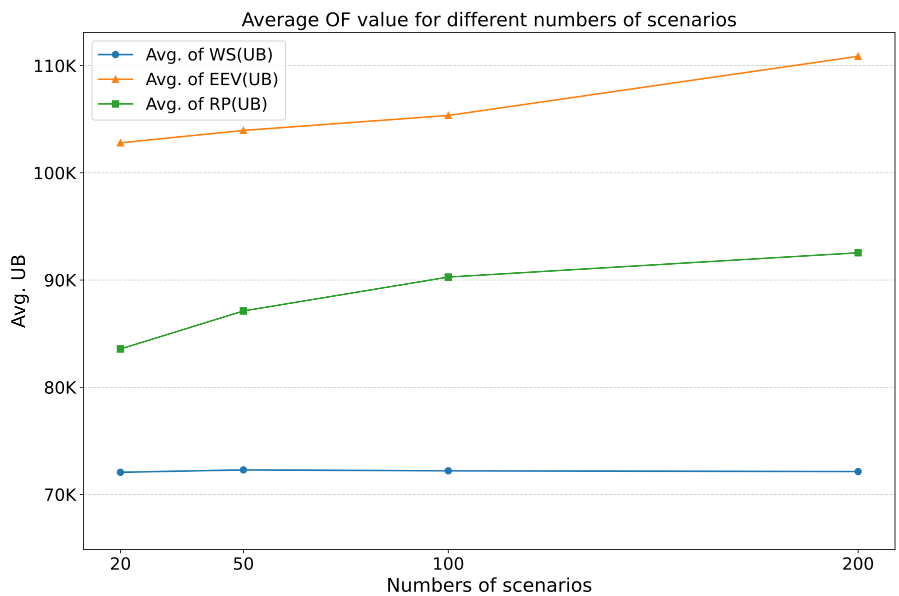


Figure 3.4: Objective function values for different numbers of scenarios

lights the ability of the HHA to obtain high-quality solutions for the problem. In terms of VSS^{LB}, there is an increasing trend in the values, which we believe results from a combination of two factors. First, using a higher number of scenarios yields a better sample set that more closely resembles real-world uncertainty. Second, the increase in VSS^{LB} may also be a consequence of the growing complexity of the problem as more scenarios are considered, which can lead to weaker UBs. Nonetheless, it is important to note that using 100 scenarios, which is the main setting throughout this study, represents a reasonable compromise between problem complexity and capturing existing uncertainty.

Figure 3.4 provides more details on the OF values for different numbers of scenarios. This figure illustrates how increasing the number of scenarios results in higher UBs for the RP, while also showing that the OF appears to converge after 100 scenarios.

3.5 Conclusion

This study addresses the stochastic two-echelon production routing problem. We present a two-stage stochastic formulation in which the first stage models the production decisions at the plant, as well as the deliveries to warehouses. The second stage captures the decisions related to customer inventories and deliveries from warehouses to customers. A fleet of homogeneous vehicles is considered at the plant, along with a separate set of homogeneous vehicles at each warehouse. These warehouse vehicles are assumed to have smaller capacities suitable for intra-city deliveries.

One important assumption of our study is the possibility of assigning customers to different warehouses in different periods or scenarios, which enhances overall efficiency. Additionally, we incorporate adaptive routing into the stochastic model, allowing the delivery routes to vary across scenarios. This flexibility contributes to cost savings but also significantly increases the problem's complexity.

To address this complexity, we developed a hybrid heuristic algorithm that decomposes the problem into two subproblems. The first subproblem handles first-stage decisions, including production planning and deliveries to warehouses, based on estimated demands of customers assigned to each warehouse. A MIP is solved for this subproblem in each iteration. The second subproblem concerns second-echelon routing and deliveries, which are solved using an ILS algorithm followed by an LP to determine the values of continuous variables. These two steps are solved iteratively, and in each iteration, customer-to-warehouse reassignments may occur if potential cost savings are identified. A perturbation phase is also incoporated to explore a wider region of the solution space.

While the main objective of this algorithm is to solve the S2EPRP-AR, it also performs well on large-scale deterministic instances. We first present computational experiments on the deterministic version to evaluate the algorithm's performance. The test instances include cases with up to 5 warehouses and 50 customers, with 2 vehicles in the first echelon and 15 vehicles in the second. The algorithm solved these instances in approximately 30 minutes on average, with an average optimality gap of 7.46%. A total of 192 instances were solved in the deterministic case.

For the stochastic version, we solved 66 instances. The BC algorithm, even with the HHA solution provided as a warm start, was unable to improve the objective function within a 2-hour time limit, demonstrating the difficulty of the problem. Despite this, the developed algorithm was able to solve all instances. An average optimality gap of 19.4% was observed. This relatively large gap is mainly due to the weak LBs obtained from the WS problems. Finding stronger LB remains a key challenge in this study.

For future research, incorporating the supplier level could improve supply chain integration and enable more coordinated decision-making. Additionally, modeling uncertainty on both the supply and demand sides could result in a more reliable system.

Acknowledgment

The authors gratefully acknowledge the support of the Natural Sciences and Engineering Research Council of Canada (RGPIN-2019-05036, RGPIN-2019-05759 and RGPAS-2019-00100), and the HEC Montreal Research Chair in Supply Chain Operations Planning. The authors also acknowledge the support of Digital Research Alliance of Canada for providing computational resources for performing this research.

References

Absi, N., Archetti, C., Dauzère-Pérès, S., and Feillet, D. (2015). "A Two-Phase Iterative Heuristic Approach for the Production Routing Problem". *Transportation Science* 49.4, pp. 784–795. DOI: 10.1287/trsc.2014.0523.

Absi, N., Archetti, C., Dauzère-Pérès, S., Feillet, D., and Speranza, M. G. (2018). "Comparing sequential and integrated approaches for the production routing problem". *European Journal of Operational Research* 269.2, pp. 633–646. DOI: 10.1016/j.ejor.2018.01.052.

Adulyasak, Y., Cordeau, J.-F., and Jans, R. (2014a). "Formulations and Branch-and-Cut Algorithms for Multivehicle Production and Inventory Routing Problems". *INFORMS Journal on Computing* 26.1, pp. 103–120. DOI: 10.1287/ijoc.2013.0550.

Adulyasak, Y., Cordeau, J.-F., and Jans, R. (2014b). "Optimization-Based Adaptive Large Neighborhood Search for the Production Routing Problem". *Transportation Science* 48.1, pp. 20–45. DOI: 10.1287/trsc.1120.0443.

Adulyasak, Y., Cordeau, J.-F., and Jans, R. (2015a). "Benders Decomposition for Production Routing Under Demand Uncertainty". *Operations Research* 63.4, pp. 851–867. DOI: 10.1287/opre.2015.1401.

Adulyasak, Y., Cordeau, J.-F., and Jans, R. (2015b). "The production routing problem: A review of formulations and solution algorithms". *Computers & Operations Research* 55, pp. 141–152. DOI: 10.1016/j.cor.2014.01.011.

Agra, A., Requejo, C., and Rodrigues, F. (2018). "An adjustable sample average approximation algorithm for the stochastic production-inventory-routing problem". *Networks* 72.1, pp. 5–24. DOI: 10.1002/net.21796.

Alvarez, A., Miranda, P., and Rohmer, S. (2022a). "Production routing for perishable products". *Omega* 111, p. 102667. DOI: 10.1016/j.omega.2022.102667.

Alvarez, A., Miranda, P., and Rohmer, S. (2022b). "Production routing for perishable products". *Omega* 111, p. 102667. DOI: 10.1016/j.omega.2022.102667.

Alvarez, A., Munari, P., and Morabito, R. (2018). "Iterated local search and simulated annealing algorithms for the inventory routing problem". *International Transactions in Operational Research* 25.6, pp. 1785–1809. DOI: 10.1111/itor.12547.

Archetti, C., Bertazzi, L., Paletta, G., and Speranza, M. G. (2011). "Analysis of the maximum level policy in a production-distribution system". *Computers & Operations Research* 38.12, pp. 1731–1746. DOI: 10.1016/j.cor.2011.03.002.

Avci, M. and Topaloglu Yildiz, S. (2020). "A mathematical programming-based heuristic for the production routing problem with transshipments". *Computers & Operations Research* 123, p. 105042. DOI: 10.1016/j.cor.2020.105042.

Bard, J. F. and Nananukul, N. (2010). "A branch-and-price algorithm for an integrated production and inventory routing problem". *Computers & Operations Research* 37.12, pp. 2202–2217.

Bräysy, O. and Gendreau, M. (2005). "Vehicle Routing Problem with Time Windows, Part I: Route Construction and Local Search Algorithms". *Transportation Science* 39.1, pp. 104–118. DOI: 10.1287/trsc.1030.0056.

Chandra, P. (1993). "A dynamic distribution model with warehouse and customer replenishment requirements". *Journal of the Operational Research Society* 44.7. DOI: 10.1057/jors.1993.117.

Chitsaz, M., Cordeau, J.-F., and Jans, R. (2019). "A Unified Decomposition Matheuristic for Assembly, Production, and Inventory Routing". *INFORMS Journal on Computing* 31.1, pp. 134–152. DOI: 10.1287/ijoc.2018.0817.

Chitsaz, M., Cordeau, J.-F., and Jans, R. (2020). "A branch-and-cut algorithm for an assembly routing problem". *European Journal of Operational Research* 282.3, pp. 896–910. DOI: 10.1016/j.ejor.2019.10.007.

Gruson, M., Bazrafshan, M., Cordeau, J.-F., and Jans, R. (2019). "A comparison of formulations for a three-level lot sizing and replenishment problem with a distribution structure". *Computers & Operations Research* 111, pp. 297–310. DOI: 10.1016/j.cor.2019.07.005.

Gruson, M., Cordeau, J.-F., and Jans, R. (2023). "Split demand and deliveries in an integrated three-level lot sizing and replenishment problem". *Computers & Operations Research*, p. 106434. DOI: 10.1016/j.cor.2023.106434.

Hrabec, D., Hvattum, L. M., and Hoff, A. (2022). "The value of integrated planning for production, inventory, and routing decisions: A systematic review and meta-analysis". *International Journal of Production Economics* 248, p. 108468. DOI: https://doi.org/10.1016/j.ijpe.2022.108468.

Kermani, A., Cordeau, J.-F., and Jans, R. (2024). "A progressive hedging-based matheuristic for the stochastic production routing problem with adaptive routing". *Computers & Operations Research* 169, p. 106745. DOI: https://doi.org/10.1016/j.cor.2024.106745.

Kermani, A., Cordeau, J.-F., and Jans, R. (2025). "The Stochastic Production Routing Problem with Adaptive Routing and Service Level Constraints".

Kumar, R. S., Kondapaneni, K., Dixit, V., Goswami, A., Thakur, L. S., and Tiwari, M. K. (2016). “Multi-objective modeling of production and pollution routing problem with time window: A self-learning particle swarm optimization approach”. *Computers & Industrial Engineering* 99, pp. 29–40. DOI: 10.1016/j.cie.2015.07.003.

Li, Y., Chu, F., Chu, C., and Zhu, Z. (2019). “An efficient three-level heuristic for the large-scaled multi-product production routing problem with outsourcing”. *European Journal of Operational Research* 272.3, pp. 914–927. DOI: 10.1016/j.ejor.2018.07.018.

Manousakis, E. G., Kasapidis, G. A., Kiranoudis, C. T., and Zachariadis, E. E. (2022). “An infeasible space exploring matheuristic for the Production Routing Problem”. *European Journal of Operational Research* 298.2, pp. 478–495. DOI: 10.1016/j.ejor.2021.05.037.

Mladenović, N. and Hansen, P. (1997). “Variable neighborhood search”. *Computers & Operations Research* 24.11, pp. 1097–1100. DOI: https://doi.org/10.1016/S0305-0548(97)00031-2.

Mousavi, R., Bashiri, M., and Nikzad, E. (2022). “Stochastic production routing problem for perishable products: Modeling and a solution algorithm”. *Computers & Operations Research* 142, p. 105725. DOI: 10.1016/j.cor.2022.105725.

Neves-Moreira, F., Almada-Lobo, B., Cordeau, J.-F., Guimarães, L., and Jans, R. (2019). “Solving a large multi-product production-routing problem with delivery time windows”. *Omega* 86, pp. 154–172. DOI: 10.1016/j.omega.2018.07.006.

Perboli, G., Tadei, R., and Vigo, D. (2011). “The two-echelon capacitated vehicle routing problem: Models and math-based heuristics”. *Transportation Science* 45.3, pp. 364–380. DOI: 10.1287/trsc.1110.0368.

Qiu, Y., Ni, M., Wang, L., Li, Q., Fang, X., and Pardalos, P. M. (2018). “Production routing problems with reverse logistics and remanufacturing”. *Transportation Research Part E: Logistics and Transportation Review* 111, pp. 87–100. DOI: 10.1016/j.tre.2018.01.009.

Qiu, Y., Qiao, J., and Pardalos, P. M. (2019). "Optimal production, replenishment, delivery, routing and inventory management policies for products with perishable inventory". *Omega* 82, pp. 193–204. DOI: 10.1016/j.omega.2018.01.006.

Qiu, Y., Zhou, D., Du, Y., Liu, J., Pardalos, P. M., and Qiao, J. (2021). "The two-echelon production routing problem with cross-docking satellites". *Transportation Research Part E: Logistics and Transportation Review* 147, p. 102210. DOI: doi.org/10.1016/j.tre.2020.102210.

Russell, R. A. (2017). "Mathematical programming heuristics for the production routing problem". *International Journal of Production Economics* 193, pp. 40–49. DOI: 10.1016/j.ijpe.2017.06.033.

Schenekemberg, C. M., Scarpin, C. T., Pécora, J. E., Guimarães, T. A., and Coelho, L. C. (2021). "The two-echelon production-routing problem". *European Journal of Operational Research* 288.2, pp. 436–449. DOI: 10.1016/j.ejor.2020.05.054.

Schenekemberg, C. M., Guimarães, T. A., Augusto, C. A., and Coelho, L. C. (2023). "A Three-Front Parallel Branch-and-Cut Algorithm for Production and Inventory Routing Problems". *Transportation Science*. DOI: 10.1287/trsc.2022.0261.

Solyalı, O. and Süral, H. (2017). "A multi-phase heuristic for the production routing problem". *Computers & Operations Research* 87, pp. 114–124. DOI: 10.1016/j.cor.2017.06.007.

Wang, C., Ni, Y., and Yang, X. (2021). "The production routing problem under uncertain environment". *IEEE Access* 9, pp. 15375–15387. DOI: 10.1109/ACCESS.2021.3053046.

Zhang, Y., Alshraideh, H., and Diabat, A. (2018). "A stochastic reverse logistics production routing model with environmental considerations". *Annals of Operations Research* 271.2, pp. 1023–1044. DOI: 10.1007/s10479-018-3045-2.

General Conclusion

Most existing models for the PRP either ignore uncertainty or assume that routing decisions must be decided before demand is realized. While these simplifications make models easier to solve, they fail to capture key features of real-world operations. In particular, when delivery routes cannot be adjusted based on actual demand, this can result in unnecessary customer visits, higher transportation costs and increased stockouts. As a result, there is a clear need for more flexible and responsive planning models that better reflect operational challenges.

This thesis addresses these limitations by studying SPRP variants that include adaptive routing. Adaptive routing allows delivery routes to be determined after demand is realized. This makes it possible to construct delivery plans that are aligned with actual needs and helps avoid the inefficiencies associated with fixed routing. The study follows a twofold objective. On the one hand, it focuses on modeling extensions of the SPRP, including adaptive routing, service levels, and a two-echelon production-distribution network. On the other hand, it develops heuristic algorithms to solve these complex models, combining heuristics and matheuristics. The three chapters of this thesis are structured to reflect this dual focus.

The first chapter introduces the concept of adaptive routing within a two-stage stochastic programming framework. In this model, routing decisions are deferred to the second stage, allowing them to be adapted based on realized customer demand. This approach provides more flexibility and improves cost-efficiency by avoiding unnecessary visits. Two strategies are explored, following ideas from the lot sizing literature. In the static-

static strategy, both setup decisions and production quantities are decided in the first stage. In the static-dynamic strategy, only setup decisions are fixed in the first stage, while production quantities are decided in the second stage.

To solve the problem, a PH-based matheuristic is developed. The algorithm begins by solving a TSP to generate an a priori customer tour. Then, the problem is decomposed by scenario, and each subproblem is solved independently using the PH algorithm with a heuristic adjustment strategy. This process guides the first-stage variables toward convergence. Once the production plans are identified, the second-stage routing decisions are further refined by solving a CVRP for each scenario and period. This approach provides a structured way to manage scenario-specific routing decisions, and experimental results show that adaptive routing leads to lower costs compared to models with fixed routing.

In the second chapter, the SPRP-AR is extended by incorporating service level constraints. In practice, it is sometimes impossible to assign costs for unmet demand. Instead, businesses operate under minimum service level requirements. The model considers four types of service level constraints. The α service level ensures that the probability of fully meeting demand exceeds a specified threshold. The β service level, also known as the fill rate, measures the proportion of demand that is satisfied directly from inventory. The γ service level limits the ratio of expected backlog to average demand, offering a broader view of service performance over time. The γ service level controls the expected backlog relative to the maximum expected backlog. These service levels are applied at varying levels of granularity. They may be enforced for each period individually or over the entire planning horizon, and they can be defined either per customer or as aggregate requirements across all customers.

To solve the problem, an iterative matheuristic is proposed. The algorithm begins by constructing setup plans based on approximate demands, obtained by aggregating demand across all customers. It then determines production quantities, customer visit decisions, and delivery quantities using approximate cost assumptions in order to generate a feasible solution that satisfies the service level constraints. Once a solution is found, routing decisions are refined to improve delivery efficiency. To diversify the search, the algorithm returns to the initial setup phase and explores new possibilities. This iterative process balances exploration and exploitation and yields high-quality solutions under different service level structures.

The third chapter addresses a more complex problem by extending the SPRP to a two-echelon network. In the S2EPRP-AR, goods are shipped from a production plant to a set of warehouses, and from the warehouses to customers. The second echelon, which includes deliveries from warehouses to customers, features adaptive routing, while first-echelon routes are fixed. In addition, customer-to-warehouse assignments are allowed to vary across scenarios and periods. This problem setting reflects many real-world logistics networks, where central facilities deliver goods to regional warehouses, and local vehicles with limited capacity handle customer deliveries. Although the two-echelon PRP has clear practical importance, only a few studies have considered this model.

The problem is solved using a hybrid heuristic algorithm. In the first stage, setup decisions, production quantities, and plant-to-warehouse deliveries are determined. These decisions are modeled as a mixed integer program. The second-stage decisions, which include warehouse and customer inventories as well as warehouse-to-customer routing, are solved using an ILS algorithm. Because the second stage is decomposable by scenario, the solution process is scalable. The algorithm alternates between cost improvement steps and diversification moves, including the reassignment of customers to different warehouses. The proposed method is also adapted to solve large-scale deterministic instances of the two-echelon PRP, including the 2EPRP with cross-docking satellites variant.

Contributions

This thesis makes several contributions to the literature on stochastic production routing and multi-echelon supply chain optimization. It introduces three novel modeling frameworks that enhance the flexibility of production and distribution planning under demand uncertainty: the Stochastic Production Routing Problem with Adaptive Routing (SPRP-AR), the SPRP-AR extended with four distinct service level constraints implemented at multiple levels of granularity, and the Stochastic Two-Echelon Production Routing Problem with Adaptive Routing (S2EPRP-AR). The thesis also develops advanced heuristic and matheuristic solution methods capable of tackling large and computationally challenging stochastic problems, including progressive hedging-based approaches (PH-M), iterative matheuristics (IMH), and a hybrid heuristic algorithm (HHA). These methods are validated through extensive computational experiments on a wide range of stochastic instances, demonstrating their efficiency, scalability, and ability to deliver high-quality solutions.

I. Stochastic Production Routing Problem with Adaptive Routing (SPRP-AR)

The first contribution is the development of the SPRP-AR, which allows routing decisions to be made after demand is realized. In Chapter 1, we show that shifting routing to the second stage captures the operational benefits of adaptive planning and reduces unnecessary customer visits. This increased flexibility lowers expected transportation costs and better reflects real-world supply chains where routing decisions can respond to demand variability.

II. Progressive Hedging-Based Matheuristic

In Chapter 1, we also present the first algorithmic contribution: a Progressive Hedging (PH)-based matheuristic for the SPRP-AR. The algorithm decomposes the stochastic problem into scenario-specific subproblems and iteratively drives first-stage decisions toward consensus using Lagrangean adjustments. A three-phase matheuristic is embedded to refine production, inventory, and routing decisions, providing high-quality solutions within reasonable computation times.

III. SPRP-AR with Service Level Constraints

In Chapter 2, the SPRP-AR is extended by incorporating four types of service level constraints, each reflecting a different operational priority in managing stockouts and backlogs. These constraints are applied at both customer and plant levels, and for each single period or for the entire planning horizon. This extension provides insights into how service level definitions and granularities affect operational performance and overall cost.

IV. Iterative Matheuristic for Service Level-Constrained SPRP-AR

The second algorithmic contribution is an Iterative Matheuristic (IMH) for the service level-constrained SPRP-AR. This approach alternates between diversification and intensification phases to generate cost-efficient solutions that satisfy service level requirements. Through the IMH, we are able to handle complex service-reliability constraints effectively and obtain high-quality solutions that balance cost and service.

V. Stochastic Two-Echelon Production Routing Problem with Adaptive Routing (S2EPRP-AR)

In Chapter 3, the study extends the adaptive routing concept to a two-echelon supply chain consisting of a plant, warehouses, and customers. The S2EPRP-AR integrates flexible customer-to-warehouse assignment with scenario-dependent second-stage routing, reflecting the complexity of real multi-echelon logistics networks. This model highlights the benefits of combining network flexibility with stochastic optimization in multi-tier supply chains.

VI. Hybrid Heuristic Algorithm for S2EPRP-AR

The third algorithmic contribution is a Hybrid Heuristic Algorithm (HHA) for the two-echelon PRP. It combines a mixed-integer programming (MIP) solver for first-stage decisions with an Iterated Local Search (ILS) for scenario-dependent second-stage routing. This hybrid approach exploits problem decomposition, scales to large deterministic and medium-size stochastic instances, and consistently produces high-quality solutions.

VII. Datasets and Extensive Computational Experiments

To validate the proposed models and algorithms, we generated new instances for the SPRP-AR with service level constraints and for the S2EPRP-AR, building on and extending existing benchmark instances. Extensive computational experiments were conducted across deterministic and stochastic instances, demonstrating the efficiency and scalability of the proposed methods. These experiments also highlight the cost benefits of adaptive routing, the impact of service level definitions, and the applicability of the approaches to real-world logistics systems. Moreover, extensive sensitivity analyses were performed to examine the effects of uncertainty levels, service level choices, and problem size on solution quality and computational effort.

Future Research Directions

The contributions presented in this thesis lead to several potential directions for future work, both in terms of modeling enhancements and algorithmic development. One possible extension involves incorporating multiple products and heterogeneous vehicles. In real-world operations, fleets often consist of vehicles with different capacities, costs, and product compatibilities. Extending the current models to account for this heterogeneity would improve their applicability and reflect a broader range of operational contexts.

Another promising direction is the dynamic and real-time adaptation of routing and inventory decisions. While this thesis uses a two-stage modeling framework, where decisions are adapted once after uncertainty is resolved, many supply chains operate under rolling horizons. Future research could explore multi-stage models or develop rolling horizon algorithms that periodically re-optimize decisions as new demand information becomes available.

The scenario-based stochastic approach adopted in this thesis could also be complemented by alternative uncertainty-handling techniques. In particular, robust optimization provides more conservative solutions that do not rely precise probability distributions. By focusing on worst-case scenarios, these approaches can produce solutions that remain feasible under a wide range of demand realizations. Such methods are especially valuable in environments with limited historical data, highly variable demand patterns, or rapidly changing market conditions, where estimating accurate probability distributions is challenging.

Another important area for future research involves sustainability. While the models developed in this work primarily focus on cost minimization, real-world supply chains increasingly require environmentally conscious planning. Future extensions could incorporate explicit carbon emission constraints, fuel consumption objectives, or broader green logistics policies. Considering the deployment of electric or hybrid vehicles, compliance with low-emission zones, and the impact of carbon taxation in both routing and inventory decisions would make the models more aligned with sustainable operations.

Lastly, the rapid progress in predictive analytics opens further opportunities for enhancing stochastic production routing and multi-echelon decision making. Future research could exploit machine learning techniques to anticipate demand patterns and identify high-probability customer visit combinations before optimization begins. By generating candidate routes or promising partial solutions based on historical data and scenario characteristics, these models could provide the algorithm with a better starting point, improving convergence and overall computational efficiency.

Bibliography

Absi, N., Archetti, C., Dauzère-Pérès, S., and Feillet, D. (2015). "A Two-Phase Iterative Heuristic Approach for the Production Routing Problem". *Transportation Science* 49.4, pp. 784–795. DOI: 10.1287/trsc.2014.0523.

Absi, N., Archetti, C., Dauzère-Pérès, S., Feillet, D., and Speranza, M. G. (2018). "Comparing sequential and integrated approaches for the production routing problem". *European Journal of Operational Research* 269.2, pp. 633–646. DOI: 10.1016/j.ejor.2018.01.052.

Adulyasak, Y., Cordeau, J.-F., and Jans, R. (2014a). "Formulations and Branch-and-Cut Algorithms for Multivehicle Production and Inventory Routing Problems". *INFORMS Journal on Computing* 26.1, pp. 103–120. DOI: 10.1287/ijoc.2013.0550.

Adulyasak, Y., Cordeau, J.-F., and Jans, R. (2014b). "Optimization-Based Adaptive Large Neighborhood Search for the Production Routing Problem". *Transportation Science* 48.1, pp. 20–45. DOI: 10.1287/trsc.1120.0443.

Adulyasak, Y., Cordeau, J.-F., and Jans, R. (2015a). "Benders Decomposition for Production Routing Under Demand Uncertainty". *Operations Research* 63.4, pp. 851–867. DOI: 10.1287/opre.2015.1401.

Adulyasak, Y., Cordeau, J.-F., and Jans, R. (2015b). "The production routing problem: A review of formulations and solution algorithms". *Computers & Operations Research* 55, pp. 141–152. DOI: 10.1016/j.cor.2014.01.011.

Agra, A., Requejo, C., and Rodrigues, F. (2018). “An adjustable sample average approximation algorithm for the stochastic production-inventory-routing problem”. *Networks* 72.1, pp. 5–24. DOI: 10.1002/net.21796.

Alvarez, A., Cordeau, J.-F., Jans, R., Munari, P., and Morabito, R. (2021). “Inventory routing under stochastic supply and demand”. *Omega* 102, p. 102304. DOI: 10.1016/j.omega.2020.102304.

Alvarez, A., Miranda, P., and Rohmer, S. (2022a). “Production routing for perishable products”. *Omega* 111, p. 102667. DOI: 10.1016/j.omega.2022.102667.

Alvarez, A., Miranda, P., and Rohmer, S. (2022b). “Production routing for perishable products”. *Omega* 111, p. 102667. DOI: 10.1016/j.omega.2022.102667.

Alvarez, A., Munari, P., and Morabito, R. (2018). “Iterated local search and simulated annealing algorithms for the inventory routing problem”. *International Transactions in Operational Research* 25.6, pp. 1785–1809. DOI: 10.1111/itor.12547.

Archetti, C., Bertazzi, L., Laporte, G., and Speranza, M. G. (2007). “A Branch-and-Cut Algorithm for a Vendor-Managed Inventory-Routing Problem”. *Transportation Science* 41.3, pp. 382–391. DOI: 10.1287/trsc.1060.0188.

Archetti, C., Bertazzi, L., Paletta, G., and Speranza, M. G. (2011). “Analysis of the maximum level policy in a production-distribution system”. *Computers & Operations Research* 38.12, pp. 1731–1746. DOI: 10.1016/j.cor.2011.03.002.

Avci, M. and Topaloglu Yildiz, S. (2020). “A mathematical programming-based heuristic for the production routing problem with transshipments”. *Computers & Operations Research* 123, p. 105042. DOI: 10.1016/j.cor.2020.105042.

Bard, J. F. and Nananukul, N. (2010). “A branch-and-price algorithm for an integrated production and inventory routing problem”. *Computers & Operations Research* 37.12, pp. 2202–2217.

Bell, W. J., Dalberto, L. M., Fisher, M. L., Greenfield, A. J., Jaikumar, R., Kedia, P., Mack, R. G., and Prutzman, P. J. (1983). “Improving the Distribution of Industrial Gases with an On-Line Computerized Routing and Scheduling Optimizer”. *Interfaces* 13.6, pp. 4–23. DOI: 10.1287/inte.13.6.4.

Ben Ahmed, M., Hvattum, L. M., and Agra, A. (2023). "The effect of different mathematical formulations on a matheuristic algorithm for the production routing problem". *Computers & Operations Research* 155, p. 106232. DOI: 10.1016/j.cor.2023.106232.

Bookbinder, J. H. and Tan, J.-Y. (1988). "Strategies for the Probabilistic Lot-Sizing Problem with Service-Level Constraints". *Management Science* 34, pp. 1096–1108. DOI: 10.1287/mnsc.34.9.1096.

Chandra, P. (1993). "A dynamic distribution model with warehouse and customer replenishment requirements". *Journal of the Operational Research Society* 44.7. DOI: 10.1057/jors.1993.117.

Chandra, P. and Fisher, M. L. (1994). "Coordination of production and distribution planning". *European Journal of Operational Research* 72.3, pp. 503–517. DOI: 10.1016/0377-2217(94)90419-7.

Chitsaz, M., Cordeau, J.-F., and Jans, R. (2019). "A Unified Decomposition Matheuristic for Assembly, Production, and Inventory Routing". *INFORMS Journal on Computing* 31.1, pp. 134–152. DOI: 10.1287/ijoc.2018.0817.

Chitsaz, M., Cordeau, J.-F., and Jans, R. (2020). "A branch-and-cut algorithm for an assembly routing problem". *European Journal of Operational Research* 282.3, pp. 896–910. DOI: 10.1016/j.ejor.2019.10.007.

Coelho, L. C., Cordeau, J.-F., and Laporte, G. (2014). "Thirty Years of Inventory Routing". *Transportation Science* 48.1, pp. 1–19. DOI: 10.1287/trsc.2013.0472.

Coelho, L. C. and Laporte, G. (2014). "Improved solutions for inventory-routing problems through valid inequalities and input ordering". *International Journal of Production Economics* 155, pp. 391–397. DOI: 10.1016/j.ijpe.2013.11.019.

Crainic, T. G., Fu, X., Gendreau, M., Rei, W., and Wallace, S. W. (2011). "Progressive hedging-based metaheuristics for stochastic network design". *Networks* 58.2, pp. 114–124. DOI: 10.1002/net.20456.

Fischetti, M. and Lodi, A. (2003). "Local branching". *Mathematical Programming* 98, pp. 23–47. DOI: 10.1007/s10107-003-0395-5.

Flynn, B. B., Huo, B., and Zhao, X. (2010). "The impact of supply chain integration on performance: A contingency and configuration approach". *Journal of Operations Management* 28.1, pp. 58–71. DOI: 10.1016/j.jom.2009.06.001.

Gade, D. and Küçükyavuz, S. (2013). "Formulations for dynamic lot sizing with service levels". *Naval Research Logistics (NRL)* 60.2, pp. 87–101. DOI: 10.1002/nav.21519.

Gruson, M., Bazrafshan, M., Cordeau, J.-F., and Jans, R. (2019). "A comparison of formulations for a three-level lot sizing and replenishment problem with a distribution structure". *Computers & Operations Research* 111, pp. 297–310. DOI: 10.1016/j.cor.2019.07.005.

Gruson, M., Cordeau, J.-F., and Jans, R. (2018). "The impact of service level constraints in deterministic lot sizing with backlogging". *Omega* 79, pp. 91–103. DOI: 10.1016/j.omega.2017.08.003.

Gruson, M., Cordeau, J.-F., and Jans, R. (2023). "Split demand and deliveries in an integrated three-level lot sizing and replenishment problem". *Computers & Operations Research*, p. 106434. DOI: 10.1016/j.cor.2023.106434.

Gu, W., Archetti, C., Cattaruzza, D., Ogier, M., Semet, F., and Grazia Speranza, M. (2022). "A sequential approach for a multi-commodity two-echelon distribution problem". *Computers & Industrial Engineering* 163, p. 107793. DOI: 10.1016/j.cie.2021.107793.

Haugen, K. K., Løkketangen, A., and Woodruff, D. L. (2001). "Progressive hedging as a meta-heuristic applied to stochastic lot-sizing". *European Journal of Operational Research* 132.1, pp. 116–122. DOI: 10.1016/S0377-2217(00)00116-8.

Helber, S., Sahling, F., and Schimmelpfeng, K. (2013). "Dynamic capacitated lot sizing with random demand and dynamic safety stocks". *OR Spectrum* 35.1, pp. 75–105. DOI: 10.1007/s00291-012-0283-6.

Hrabec, D., Hvattum, L. M., and Hoff, A. (2022). "The value of integrated planning for production, inventory, and routing decisions: A systematic review and meta-analysis".

International Journal of Production Economics 248, p. 108468. DOI: https://doi.org/10.1016/j.ijpe.2022.108468.

Jans, R. (2009). "Solving Lot-Sizing Problems on Parallel Identical Machines Using Symmetry-Breaking Constraints". *INFORMS Journal on Computing* 21.1, pp. 123–136. DOI: 10.1287/ijoc.1080.0283.

Kermani, A., Cordeau, J.-F., and Jans, R. (2024). "A progressive hedging-based matheuristic for the stochastic production routing problem with adaptive routing". *Computers & Operations Research* 169, p. 106745. DOI: https://doi.org/10.1016/j.cor.2024.106745.

Kermani, A., Cordeau, J.-F., and Jans, R. (2025). "The Stochastic Production Routing Problem with Adaptive Routing and Service Level Constraints".

Kilic, O. A. and Tarim, S. A. (2011). "An investigation of setup instability in non-stationary stochastic inventory systems". *International Journal of Production Economics* 133.1, pp. 286–292. DOI: 10.1016/j.ijpe.2010.04.021.

Koca, E., Yaman, H., and Aktürk, M. S. (2018). "Stochastic lot sizing problem with nervousness considerations". *Computers & Operations Research* 94, pp. 23–37. DOI: 10.1016/j.cor.2018.01.021.

Kumar, R. S., Kondapaneni, K., Dixit, V., Goswami, A., Thakur, L. S., and Tiwari, M. K. (2016). "Multi-objective modeling of production and pollution routing problem with time window: A self-learning particle swarm optimization approach". *Computers & Industrial Engineering* 99, pp. 29–40. DOI: 10.1016/j.cie.2015.07.003.

Li, Y., Chu, F., Chu, C., and Zhu, Z. (2019). "An efficient three-level heuristic for the large-scaled multi-product production routing problem with outsourcing". *European Journal of Operational Research* 272.3, pp. 914–927. DOI: 10.1016/j.ejor.2018.07.018.

Løkketangen, A. and Woodruff, D. L. (1996). "Progressive hedging and tabu search applied to mixed integer (0, 1) multistage stochastic programming". *Journal of Heuristics* 2.2, pp. 111–128. DOI: 10.1007/BF00247208.

Lu, D., Ding, Y., Asian, S., and Paul, S. K. (2018). "From supply chain integration to operational performance: The moderating effect of market uncertainty". *Global Journal of Flexible Systems Management* 19.1, pp. 3–20. DOI: 10.1007/s40171-017-0161-9.

Manousakis, E. G., Kasapidis, G. A., Kiranoudis, C. T., and Zachariadis, E. E. (2022). "An infeasible space exploring matheuristic for the Production Routing Problem". *European Journal of Operational Research* 298.2, pp. 478–495. DOI: 10.1016/j.ejor.2021.05.037.

Mladenović, N. and Hansen, P. (1997). "Variable neighborhood search". *Computers & Operations Research* 24.11, pp. 1097–1100. DOI: https://doi.org/10.1016/S0305-0548(97)00031-2.

Mousavi, R., Bashiri, M., and Nikzad, E. (2022). "Stochastic production routing problem for perishable products: Modeling and a solution algorithm". *Computers & Operations Research* 142, p. 105725. DOI: 10.1016/j.cor.2022.105725.

Neves-Moreira, F., Almada-Lobo, B., Cordeau, J.-F., Guimarães, L., and Jans, R. (2019). "Solving a large multi-product production-routing problem with delivery time windows". *Omega* 86, pp. 154–172. DOI: 10.1016/j.omega.2018.07.006.

Perboli, G., Tadei, R., and Vigo, D. (2011). "The two-echelon capacitated vehicle routing problem: Models and math-based heuristics". *Transportation Science* 45.3, pp. 364–380. DOI: 10.1287/trsc.1110.0368.

Qiu, Y., Ni, M., Wang, L., Li, Q., Fang, X., and Pardalos, P. M. (2018). "Production routing problems with reverse logistics and remanufacturing". *Transportation Research Part E: Logistics and Transportation Review* 111, pp. 87–100. DOI: 10.1016/j.tre.2018.01.009.

Qiu, Y., Qiao, J., and Pardalos, P. M. (2017). "A branch-and-price algorithm for production routing problems with carbon cap-and-trade". *Omega* 68, pp. 49–61. DOI: 10.1016/j.omega.2016.06.001.

Qiu, Y., Qiao, J., and Pardalos, P. M. (2019). "Optimal production, replenishment, delivery, routing and inventory management policies for products with perishable inventory". *Omega* 82, pp. 193–204. DOI: 10.1016/j.omega.2018.01.006.

Qiu, Y., Zhou, D., Du, Y., Liu, J., Pardalos, P. M., and Qiao, J. (2021). "The two-echelon production routing problem with cross-docking satellites". *Transportation Research Part E: Logistics and Transportation Review* 147, p. 102210. DOI: doi.org/10.1016/j.tre.2020.102210.

Rockafellar, R. T. and Wets, R. J. B. (1991). "Scenarios and Policy Aggregation in Optimization Under Uncertainty". *Mathematics of Operations Research* 16.1, pp. 119–147. DOI: 10.1287/moor.16.1.119.

Rodrigues, L. F., Santos, M. O. D., and Almada-Lobo, B. (2023). "A Memetic Algorithm for the multi-product Production Routing Problem". *Computers & Industrial Engineering* 182, p. 109388. DOI: 10.1016/j.cie.2023.109388.

Russell, R. A. (2017). "Mathematical programming heuristics for the production routing problem". *International Journal of Production Economics* 193, pp. 40–49. DOI: 10.1016/j.ijpe.2017.06.033.

Schenekemberg, C. M., Scarpin, C. T., Pécora, J. E., Guimarães, T. A., and Coelho, L. C. (2021). "The two-echelon production-routing problem". *European Journal of Operational Research* 288.2, pp. 436–449. DOI: 10.1016/j.ejor.2020.05.054.

Schenekemberg, C. M., Guimarães, T. A., Augusto, C. A., and Coelho, L. C. (2023). "A Three-Front Parallel Branch-and-Cut Algorithm for Production and Inventory Routing Problems". *Transportation Science*. DOI: 10.1287/trsc.2022.0261.

Schneider, H. (1981). "Effect of service-levels on order-points or order-levels in inventory models". *International Journal of Production Research* 19.6, pp. 615–631. DOI: 10.1080/00207548108956694.

Sereshti, N., Adulyasak, Y., and Jans, R. (2021). "The value of aggregate service levels in stochastic lot sizing problems". *Omega* 102, p. 102335. DOI: 10.1016/j.omega.2020.102335.

Sereshti, N., Adulyasak, Y., and Jans, R. (2024). "Managing flexibility in stochastic multi-level lot sizing problem with service level constraints". *Omega* 122, p. 102957. DOI: https://doi.org/10.1016/j.omega.2023.102957.

Shuang, Y., Diabat, A., and Liao, Y. (2019). “A stochastic reverse logistics production routing model with emissions control policy selection”. *International Journal of Production Economics* 213, pp. 201–216. DOI: 10.1016/j.ijpe.2019.03.006.

Silver, E. A., Pyke, D. F., Peterson, R., et al. (1998). *Inventory management and production planning and scheduling*. Vol. 3. Wiley New York.

Sluijk, N., Florio, A. M., Kinable, J., Dellaert, N., and Van Woensel, T. (2023). “Two-echelon vehicle routing problems: A literature review”. *European Journal of Operational Research* 304.3, pp. 865–886. DOI: doi.org/10.1016/j.ejor.2022.02.022.

Solyalı, O. and Süral, H. (2017). “A multi-phase heuristic for the production routing problem”. *Computers & Operations Research* 87, pp. 114–124. DOI: 10.1016/j.cor.2017.06.007.

Solyalı, O. and Süral, H. (2022). “An Effective Matheuristic for the Multivehicle Inventory Routing Problem”. *Transportation Science* 56.4, pp. 1044–1057. DOI: 10.1287/trsc.2021.1123.

Stadtler, H. and Meistering, M. (2019). “Model formulations for the capacitated lot-sizing problem with service-level constraints”. *OR Spectrum* 41.4, pp. 1025–1056. DOI: 10.1007/s00291-019-00552-1.

Stevens, G. C. (1989). “Integrating the Supply Chain”. *International Journal of Physical Distribution & Materials Management* 19.8, pp. 3–8. DOI: 10.1108/eum0000000000329.

Tarim, S. A. and Kingsman, B. G. (2004). “The stochastic dynamic production/inventory lot-sizing problem with service-level constraints”. *International Journal of Production Economics* 88.1, pp. 105–119. DOI: 10.1016/S0925-5273(03)00182-8.

Tempelmeier, H. (2007). “On the stochastic uncapacitated dynamic single-item lotsizing problem with service level constraints”. *European Journal of Operational Research* 181.1, pp. 184–194. DOI: 10.1016/j.ejor.2006.06.009.

Tempelmeier, H. (2013). “Stochastic Lot Sizing Problems”. *International Series in Operations Research & Management Science*. Vol. 192. Springer, pp. 313–344. DOI: 10.1007/978-1-4614-6777-9_10.

Tempelmeier, H. and Herpers, S. (2010). “ABC β–a heuristic for dynamic capacitated lot sizing with random demand under a fill rate constraint”. *International Journal of Production Research* 48.17, pp. 5181–5193. DOI: 10.1080/00207540903179782.

Tempelmeier, H. and Herpers, S. (2011). “Dynamic uncapacitated lot sizing with random demand under a fillrate constraint”. *European Journal of Operational Research* 212.3, pp. 497–507. DOI: 10.1016/j.ejor.2011.02.012.

Tomazella, C. P., Santos, M. O., Alem, D., and Jans, R. (2023). “Service-level-driven procurement and production lot-sizing problem with demand fulfilment”. *International Journal of Production Research*, pp. 1–22. DOI: 10.1080/00207543.2023.2204958.

Tunc, H., Kilic, O. A., Tarim, S. A., and Eksioglu, B. (2013). “A simple approach for assessing the cost of system nervousness”. *International Journal of Production Economics* 141.2, pp. 619–625. DOI: 10.1016/j.ijpe.2012.09.022.

Tunc, H., Kilic, O. A., Tarim, S. A., and Eksioglu, B. (2014). “A reformulation for the stochastic lot sizing problem with service-level constraints”. *Operations Research Letters* 42.2, pp. 161–165. DOI: 10.1016/j.orl.2014.01.010.

Tunc, H., Kilic, O. A., Tarim, S. A., and Rossi, R. (2018). “An Extended Mixed-Integer Programming Formulation and Dynamic Cut Generation Approach for the Stochastic Lot-Sizing Problem”. *INFORMS Journal on Computing* 30.3, pp. 492–506. DOI: 10.1287/ijoc.2017.0792.

Vadseth, S. T., Andersson, H., Stålhane, M., and Chitsaz, M. (2023). “A multi-start route improving matheuristic for the production routeing problem”. *International Journal of Production Research* 61.22, pp. 7608–7629. DOI: 10.1080/00207543.2022.2154402.

Wang, C., Ni, Y., and Yang, X. (2021). “The production routing problem under uncertain environment”. *IEEE Access* 9, pp. 15375–15387. DOI: 10.1109/ACCESS.2021.3053046.

Zhang, Y., Alshraideh, H., and Diabat, A. (2018). “A stochastic reverse logistics production routing model with environmental considerations”. *Annals of Operations Research* 271.2, pp. 1023–1044. DOI: 10.1007/s10479-018-3045-2.

Appendix A: A Progressive Hedging-based Matheuristic for the Stochastic Production Routing Problem with Adaptive Routing

A.1 Detailed CPU time of the PH-M algorithm

In this section, we present the computation time of each phase of the PH-M algorithm. Tables A.1 and A.2 provide the details of the CPU time for each phase of the algorithm for different ε values under the static-dynamic strategy and static-static strategy, respectively.

A.2 Computational experiments on larger instances

In this section, we present the results of experiments conducted on larger instances with up to 50 customers. Due to the high computational cost associated with determining the Value of Stochastic Solution (VSS) and Expected Value of Perfect Information (EVPI), as well as resource constraints, our focus was solely on solving the SPRP-AR using the PH-M and BC algorithms. We provide a summary of the results for the static-dynamic and static-static strategies in Tables A.3 and A.4, respectively. For the BC algorithm, we utilized the solution from the PH-M algorithm as a warm start.

Table A.1: Average computation time of the PH-M under the Static-Dynamic Strategy

T	K	ε	#Ins	SPRP-AR (PH-M)			
				Phase One (secs)	Phase Two (secs)	Phase Three (secs)	Total (secs)
6	1	0.2	6	0.2	113.1	1.4	114.7
6	2	0.2	6	0.1	708.8	264.5	973.5
6	3	0.2	4	0.1	1,182.0	30.6	1,212.7
9	1	0.2	6	0.1	641.7	2.3	644.0
9	2	0.2	4	0.1	1,672.7	21.4	1,694.2
Total			26	0.1	776.9	69.9	847.0
6	1	0.4	6	0.3	117.1	1.7	119.2
6	2	0.4	6	0.2	893.4	246.1	1,139.7
6	3	0.4	4	0.1	1,217.0	24.7	1,241.8
9	1	0.4	6	0.2	794.2	2.1	796.5
9	2	0.4	4	0.2	1,941.7	29.7	1,971.6
Total			26	0.2	902.4	66.0	968.7
6	1	0.6	6	0.1	119.8	1.5	121.3
6	2	0.6	6	0.1	846.1	167.6	1,013.7
6	3	0.6	4	0.0	1,220.0	37.4	1,257.5
9	1	0.6	6	0.1	752.1	1.7	753.9
9	2	0.6	4	0.0	1,946.3	39.2	1,985.5
Total			26	0.1	883.6	51.2	934.8
6	1	0.8	6	1.0	159.1	2.7	162.9
6	2	0.8	6	1.2	1,063.6	37.3	1,102.1
6	3	0.8	4	0.1	1,385.4	17.9	1,403.4
9	1	0.8	6	0.1	838.0	3.1	841.2
9	2	0.8	4	0.1	2,332.2	26.8	2,359.0
Total			26	0.6	1,047.3	16.8	1,064.9

Each setting ranged from 35 to 50 customers, with a step of 5, resulting in four instances per row of the table. The columns **T** and **K** represent the number of periods and vehicles in the configuration, while ε denotes the uncertainty level, and **#Ins** indicates the number of instances. Under the **PH-M** section, the **#Best** column illustrates the number of instances where the Upper Bound (UB) of the BC algorithm was equal to that obtained by the PH-M algorithm. The **CPU** column displays computation time in seconds. For the **BC** section, the **CPU** column shows the computation time of the BC algorithm plus the CPU time of the PH-M algorithm. Finally, the **Gap** column is the relative gap of the solution using the Lower Bound (LB) obtained from the BC algorithm. It is important

Table A.2: Average computation time of the PH-M under the Static-Static Strategy

T	K	ε	#Ins	SPRP-AR (PH-M) Phase One (secs)	Phase Two (secs)	Phase Three (secs)	Total (secs)
6	1	0.2	6	0.1	138.4	1.4	140.0
6	2	0.2	6	0.1	879.2	355.1	1,234.4
6	3	0.2	4	0.0	1,523.7	30.5	1,554.3
9	1	0.2	6	0.1	871.1	1.8	873.1
9	2	0.2	4	0.1	1,690.4	26.5	1,717.0
Total			26	0.1	930.3	91.5	1,021.9
6	1	0.4	6	1.0	174.2	2.4	177.6
6	2	0.4	6	0.2	936.1	265.1	1,201.5
6	3	0.4	4	0.1	1,332.2	62.1	1,394.4
9	1	0.4	6	0.1	771.4	2.0	773.5
9	2	0.4	4	0.1	1,913.2	27.2	1,940.4
Total			26	0.3	933.5	75.9	1,009.8
6	1	0.6	6	0.1	176.1	1.8	178.1
6	2	0.6	6	0.1	1,059.9	224.6	1,284.6
6	3	0.6	4	0.1	1,233.3	30.5	1,263.9
9	1	0.6	6	0.2	887.3	3.0	890.5
9	2	0.6	4	0.1	2,590.5	31.8	2,622.5
Total			26	0.1	1,078.3	62.5	1,140.9
6	1	0.8	6	0.1	189.5	1.9	191.5
6	2	0.8	6	0.1	1,213.9	32.4	1,246.5
6	3	0.8	4	0.1	1,312.9	15.5	1,328.5
9	1	0.8	6	0.1	851.3	2.6	854.0
9	2	0.8	4	0.2	2,517.4	48.0	2,565.6
Total			26	0.1	1,109.6	18.3	1,128.0

to note that in Section 1.5, we explored the possibility of utilizing the wait-and-see (WS) problem to obtain LBs. However, in these experiments we only relied on the LB provided by the BC algorithm.

As observed in Table A.3, for the static-dynamic strategy, the PH-M algorithm successfully solved all instances with an average gap of 8.4%. However, the BC algorithm failed to improve the initial solution within the time limit of four hours for any of the instances. Similarly, for the static-static strategy, as shown in Table A.4, the PH-M algorithm found a feasible solution for all instances with an average gap of 8.2%. The BC algorithm was unable to enhance the warm-start solution for any of the instances in this

Table A.3: Summary of the SPRP-AR results for different uncertainty levels large instances under the Static-Dynamic Strategy

T	K	ε	#Ins	PH-M #Best	PH-M CPU (secs)	BC CPU (secs)	Gap (%)
6	1	0.2	4	4	579.7	14,983.0	4.0
6	2	0.2	4	4	4,716.7	19,124.3	12.2
9	1	0.2	4	4	4,373.3	18,778.3	8.3
Total			12	12	3,223.2	17,628.5	8.2
6	1	0.4	4	4	450.0	14,853.1	4.2
6	2	0.4	4	4	4,594.9	19,003.3	16.8
9	1	0.4	4	4	2,847.0	17,251.5	7.5
Total			12	12	2,630.6	17,035.9	9.5
6	1	0.6	4	4	480.5	14,883.5	4.0
6	2	0.6	4	4	4,371.7	18,780.9	12.0
9	1	0.6	4	4	3,418.4	17,822.6	7.5
Total			12	12	2,756.9	17,162.4	7.8
6	1	0.8	4	4	581.8	14,985.3	4.6
6	2	0.8	4	4	5,186.7	19,595.1	11.3
9	1	0.8	4	4	3,245.1	17,650.2	8.0
Total			12	12	3,004.5	17,410.2	8.0

case.

A.3 Factors contributing to the routing costs difference

The disparity in routing costs across various distribution functions and uncertainty levels raises the question of which factors contribute to this difference. After investigating three factors, including: Average Number of Routes (ANR), Average Route Length (ARL), and Average per-route Number of Visited Customers (ANVC), it became evident that the ANR serves as the primary contributor to these differences in routing costs. Table A.5 presents the average values of each factor under different distribution functions and ε values.

Table A.4: Summary of the SPRP-AR results for different uncertainty levels large instances under the Static-Static Strategy

T	K	ε	#Ins	PH-M #Best	PH-M CPU (secs)	BC CPU (secs)	Gap (%)
6	1	0.2	4	4	643.6	15,047.0	5.2
6	2	0.2	4	4	5,505.7	19,914.8	11.8
9	1	0.2	4	4	2,005.1	16,410.5	8.1
Total			12	12	2,718.1	17,124.1	8.4
6	1	0.4	4	4	457.6	14,860.8	4.7
6	2	0.4	4	4	4,865.1	19,273.6	11.3
9	1	0.4	4	4	3,028.2	17,432.2	7.5
Total			12	12	2,783.6	17,188.9	7.8
6	1	0.6	4	4	564.3	14,968.3	4.8
6	2	0.6	4	4	4,789.4	19,198.9	12.3
9	1	0.6	4	4	4,358.8	18,764.1	7.2
Total			12	12	3,237.5	17,643.8	8.1
6	1	0.8	4	4	613.6	15,018.1	6.2
6	2	0.8	4	4	4,025.5	18,435.3	10.7
9	1	0.8	4	4	3,288.8	17,693.5	7.8
Total			12	12	2,642.6	17,048.9	8.2

Table A.5: Average ANR, ARL, and ANVC for different probability functions and under different uncertainty levels

DF	ε	ANR	ARL	ANVC
Uniform	0.2	0.461	4.489	3.827
	0.4	0.506	4.382	3.876
	0.6	0.517	4.356	3.840
	0.8	0.538	4.358	3.782
Normal	0.2	0.461	4.089	3.627
	0.4	0.498	4.322	3.823
	0.6	0.508	4.333	3.825
	0.8	0.535	4.346	3.811
Gamma	0.2	0.463	4.083	3.619
	0.4	0.494	4.300	3.809
	0.6	0.515	4.311	3.795
	0.8	0.542	4.361	3.819

Appendix B: The Stochastic Production Routing Problem with Adaptive Routing and Service Level Constraints

B.1 Details of the Branch-and-Cut algorithm

In this section, we provide the formulation for the SPRP-FR and details on the implementation of the BC algorithm. For this discussion, we use $\alpha_c^{customer}$ as the base formulation. To present the SPRP-FR, we first highlight the main difference compared to SPRP-AR, which is the way routing is handled. While in SPRP-AR the routes are determined based on the realized scenario, in SPRP-FR the routes are fixed and remain unchanged regardless of scenario realization. As a result, the node visit (z_{ikt}) and edge visit (x_{ijkt}) variables have one less dimension since the index s is no longer required. The formulation for SPRP-FR$_{\alpha_c^{customer}}$ is as follows:

(SPRP-FR$_{\alpha_c^{customer}}$)

$$\min \sum_{t \in \mathscr{T}} \left(F y_t + \sum_{(i,j) \in E} \sum_{k \in \mathscr{K}} c_{ij} x_{ijkt} + \sum_{s \in \phi} \xi_s \left(u p_t^s + \sum_{i \in \mathscr{N}} h_i I_{it}^s \right) \right) \tag{B.1}$$

s.t.

$$p_t^s \leq \mathscr{M}_t^s y_t \qquad \forall t \in \mathscr{T}, s \in \phi \tag{B.2}$$

$$I^s_{0t} = I^s_{0,t-1} + p^s_t - \sum_{i\in\mathscr{N}_c}\sum_{k\in\mathscr{K}} q^s_{ikt} \qquad \forall t \in \mathscr{T}, s \in \phi \quad \text{(B.3)}$$

$$I^s_{it} = b^s_{it} + I^s_{i,t-1} + \sum_{k\in\mathscr{K}} q^s_{ikt} - b^s_{i,t-1} - d^s_{it} \qquad \forall i \in \mathscr{N}_c, t \in \mathscr{T}, s \in \phi \quad \text{(B.4)}$$

$$I^s_{0t} \leq L_0 \qquad \forall t \in \mathscr{T}, s \in \phi \quad \text{(B.5)}$$

$$I^s_{it} + d^s_{it} \leq L_i \qquad \forall i \in \mathscr{N}_c, t \in \mathscr{T}, s \in \phi \quad \text{(B.6)}$$

$$\sum_{i\in\mathscr{N}_c} q^s_{ikt} \leq Q z_{0kt} \qquad \forall k \in \mathscr{K}, t \in \mathscr{T}, s \in \phi \quad \text{(B.7)}$$

$$\sum_{k\in\mathscr{K}} z_{ikt} \leq 1 \qquad \forall i \in \mathscr{N}_c, t \in \mathscr{T} \quad \text{(B.8)}$$

$$q^s_{ikt} \leq \mathscr{W}^s_{it} z_{ikt} \qquad \forall i \in \mathscr{N}_c, k \in \mathscr{K}, t \in \mathscr{T}, s \in \phi \quad \text{(B.9)}$$

$$D^s_{it} - \sum_{l=1}^{t}\sum_{k\in\mathscr{K}} q^s_{ikl} - I_{i0} \leq D^s_{it}\, o^s_{it} \qquad \forall i \in \mathscr{N}_c, t \in \mathscr{T}, s \in \phi \quad \text{(B.10)}$$

$$\sum_{s\in\phi} \xi_s o^s_{it} \leq 1 - \alpha^{customer}_c \qquad \forall i \in \mathscr{N}_c, t \in \mathscr{T} \quad \text{(B.11)}$$

$$\sum_{(j,j')\in\varepsilon(\{i\})} x_{jj'kt} = 2 z_{ikt} \qquad \forall i \in \mathscr{N}, k \in \mathscr{K}, t \in \mathscr{T} \quad \text{(B.12)}$$

$$\sum_{(i,j)\in E(\eta)} x_{ijkt} \leq \sum_{i\in\eta} z_{ikt} - z_{ekt} \qquad \forall \eta \subseteq \mathscr{N}_c, |\eta| \geq 2, e \in \eta, k \in \mathscr{K}, t \in \mathscr{T} \quad \text{(B.13)}$$

$$y_t \in \{0,1\} \qquad \forall t \in \mathscr{T} \quad \text{(B.14)}$$

$$p^s_t \geq 0 \qquad \forall t \in \mathscr{T}, s \in \phi \quad \text{(B.15)}$$

$$q^s_{ikt} \geq 0 \qquad \forall i \in \mathscr{N}_c, k \in \mathscr{K}, t \in \mathscr{T}, s \in \phi \quad \text{(B.16)}$$

$$I^s_{it} \geq 0 \qquad \forall i \in \mathscr{N}, t \in \mathscr{T}, s \in \phi \quad \text{(B.17)}$$

$$b^s_{it} \geq 0 \qquad \forall i \in \mathscr{N}_c, t \in \mathscr{T}, s \in \phi \quad \text{(B.18)}$$

$$z_{ikt} \in \{0,1\} \qquad \forall i \in \mathscr{N}, k \in \mathscr{K}, t \in \mathscr{T} \quad \text{(B.19)}$$

$$x_{ijkt} \in \{0,1\} \qquad \forall (i,j) \in E, i \neq 0, k \in \mathscr{K}, t \in \mathscr{T} \quad \text{(B.20)}$$

$$x_{0jkt} \in \{0,1,2\} \qquad \forall j \in \mathscr{N}_c, k \in \mathscr{K}, t \in \mathscr{T} \quad \text{(B.21)}$$

$$o^s_{it} \in \{0,1\} \qquad \forall i \in \mathscr{N}_c, t \in \mathscr{T}, s \in \phi. \quad \text{(B.22)}$$

The number of SECs in the above problem increases exponentially with the number of customers. To address this, we first relax the SECs and employ a BC algorithm to add them incrementally during the solution process whenever violations are detected. This approach ensures that SECs are added only when necessary, improving computational efficiency. To identify violated subtours for each vehicle, we solve a separation algorithm. To do this, we use the minimum s-t cut problem from the Concorde library, which is known for its computational efficiency (Applegate et al., 2020). At the root node, SECs are added for both fractional and integer solutions. However, after the root node, we check SEC violations only for integer solutions to maintain algorithmic efficiency.

The key distinction between first-stage routing and adaptive routing lies in how SECs are applied. For the SPRP-AR, SEC constraints (2.13) must be added for every scenario. Consequently, the separation problem is solved independently for each scenario, and violated constraints are added to the specific scenario where they are detected. While it is theoretically possible to add violated constraints from one scenario to all other scenarios, this approach is highly memory-intensive and proved impractical during our algorithm implementation. Instead, we restrict the SECs of each scenario to that scenario alone. In contrast, for the SPRP-FR, since the routing decisions are made in the first stage, SEC constraints (B.13) are applied in the first stage and do not have to be generated for each scenario separately.

B.2 Evaluating The Value of SPRP-AR with Service Level Constraints

A common approach for assessing the benefit of solving a stochastic problem instead of a deterministic problem with mean demand is to compute the Value of the Stochastic Solution (VSS). This process begins by solving a deterministic version of the problem using the expected demand, known as the Expected Value (EV) problem. The first-stage decisions from the EV solution are then fixed, and the stochastic problem with fixed first stage decisions is solved. By comparing the expected outcome (e.g., total cost) of this solution against the optimal solution of the original stochastic problem, VSS quantifies the potential improvement gained by explicitly considering uncertainty. In settings where a penalty cost for unmet demand can be assigned, VSS provides a clear metric for how stochastic recourse improves the objective function. However, in our case, the VSS approach is not applicable due to the presence of service-level constraints modeled as chance constraints. Determining first-stage decisions solely based on expected demand may lead to violations of the required service levels. Therefore, a conventional VSS analysis does not directly apply to our service-level-driven problem.

To support the previous argument, we solve the EV and then the stochastic problem by fixing the values of the first-stage variables to those obtained from the EV. We performed experiments under all service levels using the customer level-single period strategy. The stochastic problem with fixed first-stage variables was infeasible for all instances with six and nine periods. However, for some instances with three periods, we observed feasible solutions. The average VSS values for these instance were 21%, 27%, 18%, and 50% for the $\alpha_c^{customer}$, $\beta_c^{customer}$, $\gamma_c^{customer}$, and $\delta_c^{customer}$ service levels, respectively.

To address this issue and ensure feasibility, we propose an alternative approach to evaluate the benefit of the stochastic solution. First, we solve a Full Demand Problem (FDP), where each customer's demand is set to the maximum observed across all scenarios. This effectively enforces a 100% service level, ensuring that no shortages occur. Next, we fix the first-stage setup decisions from the FDP solution and reinstate our original stochastic framework with service-level requirements under the actual demand distribution. We refer to this second problem as the Expected of the Full Demand (EFD) problem.

By comparing the EFD solution with the original stochastic problem incorporating service level constraints, we define the Value of Service Level over Full Demand (VSLFD) as a measure of how service-level-based stochastic optimization performs relative to a full-demand deterministic baseline. Given that we have an upper bound on the fully stochastic solution (since it is not necessarily solved to optimality due to the time limit), we can only provide a lower bound for this metric as a percentage relative to the lower bound of the EFD problem, denoted as LB_{EFD}. Mathematically, the lower bound of VSLFD, LB_{VSLFD}, is computed as: $\frac{\text{LB}_{\text{EFD}}-\text{UB}_{\text{SPRP-AR}}}{\text{LB}_{\text{EFD}}}$, where $\text{UB}_{\text{SPRP-AR}}$ represents the

Table B.1: Average VSLFD for different service levels for customer level-single period

SL Type	T	S	#INS	LB_{VSLFD} (%)
$\alpha_c^{customer}$	3	100	108	14.9
	6	100	96	8.2
	9	100	60	7.4
Total			264	11.0
$\beta_c^{customer}$	3	100	108	18.7
	6	100	96	10.6
	9	100	60	9.1
Total			264	13.0
$\gamma_c^{customer}$	3	100	108	13.0
	6	100	96	4.2
	9	100	60	3.3
Total			264	7.8
$\delta_c^{customer}$	3	100	108	29.1
	6	100	96	26.6
	9	100	60	24.4
Total			264	27.1

upper bound of the SPRP-AR obtained via BC. Table B.1 reports the LB_{VSLFD} values under different service level metrics and for customer level-single period granularity.

The average cost savings achieved by solving the SPRP-AR with service level constraints, rather than using the VSLFD, range from 7.8% to 27.1% across different metrics. Notably, the δ service level exhibits the largest difference, whereas the γ service level shows the smallest. As noted in Section 2.5.3, the γ service level is the strictest, requiring a higher proportion of demand to be met, followed by the β- and δ service levels, while the α service level differs in nature. The high demand fulfillment required under the γ service level results in higher total costs. Thus, $UB_{SPRP\text{-}AR}$ is closer to LB_{EFD} and that leads to a lower VSLFD, though an average difference of 7.8% is still observed. In contrast, the δ service level, which permits greater unmet demand, leads to a larger cost gap, reaching up to 27.1%.

Additionally, instances with fewer planning periods (e.g., three periods) tend to exhibit larger percentage differences. In shorter horizons, setup costs typically constitute a more significant share of the objective function, amplifying the impact of even minor solution changes. As a result, deciding to produce in just one additional period can substantially affect the overall objective value.

B.3 Details of the CPU time for the IMH algorithm

In this section, we present the CPU time associated with each phase of the IMH algorithm. Table B.2 details the CPU time for each phase of the IMH algorithm applied to the per customer-per period variant of the problem. This breakdown highlights the time attributed to each phase of the algorithm, offering insights into its computational efficiency.

The second phase of the algorithm, which primarily focuses on satisfying the service level constraints, exhibits significantly higher CPU times for the α and especially the β service levels. This is because these service levels involve more complex constraints and require additional binary variables. In contrast, the second phase requires the least time for the γ and δ service levels.

The type of service level also influences the first phase of the algorithm, as a simplified version of the service level constraints is incorporated during this phase. While the third phase is generally expected to have balanced CPU times across all service levels, this is particularly true for the γ and δ service levels. For these service levels, the algorithm shows a very slow convergence as it approaches a 1% optimality gap, and the stopping criterion is often triggered by the specified CPU time limit. This ensures computational efficiency without compromising solution quality.

For the α and β service levels, however, the third phase has lower CPU times compared to the γ and δ service levels. This is because a time limit is imposed on each intensification phase, and the more complex constraints of the α and β service levels result in longer CPU times during the second phase. Consequently, less time remains for the third phase, and it reaches the intensification time limit earlier. It is important to emphasize that in these cases, meeting the service level requirements in the second phase takes precedence over further route optimization in the third phase. This prioritization ensures that the algorithm provides feasible and high-quality routing solutions, as reducing the time limit for the second phase could lead to infeasibility.

Table B.2: Average CPU Time for Different Phases of the IMH Algorithm

SL Type	T	S	#INS	CPU-IMH Phase One (secs)	CPU-IMH Phase Two (secs)	CPU-IMH Phase Three (secs)
$\alpha_c^{customer}$	3	100	108	574.4	229.7	1,834.7
	6	100	96	1,680.3	2,159.2	2,048.9
	9	100	60	2,023.6	2,688.9	1,734.3
Total			264	1,305.9	1,490.3	1,889.8
$\beta_c^{customer}$	3	100	108	1739.3	3185.4	1228.1
	6	100	96	2167.3	3098.4	1459.3
	9	100	60	2148.3	2875.7	1692.4
Total			264	1915.7	3113.0	1362.7
$\gamma_c^{customer}$	3	100	108	653.2	47.8	1846.9
	6	100	96	1203.7	616.3	3481.0
	9	100	60	1582.7	1265.8	2950.4
Total			264	1064.6	531.4	2691.9
$\delta_c^{customer}$	3	100	108	801.9	34.5	2012.4
	6	100	96	1528.1	841.7	3221.8
	9	100	60	1790.8	1073.5	3073.7
Total			264	1290.7	564.2	2693.4

Appendix C: The Stochastic Two-Echelon Production Routing Problem with Adaptive Routing

C.1 Summary of Sets, Parameters, and Decision Variables

Table C.1 presents the descriptions of all sets and parameters, and Table C.2 provides the descriptions of the decision variables used in the problem introduced in Section 3.2.

C.2 Mathematical formulation for the deterministic 2EPRP

In this section, we present the mathematical formulation for the deterministic two-echelon PRP, where the demand is assumed to be known in advance. As a result, we no longer consider scenarios to represent possible customer demands, and the scenario index s is no longer needed in the model. All parameters and decision variables are similar to those introduced in Table C.1 and Table C.2, except that the scenario index is removed. In addition, for the deterministic case, shortages are no longer allowed, and the variable l_{it}^{s} is therefore excluded from the formulation. The objective function minimizes

Table C.1: Sets and parameters.

Sets:	
$\mathcal{N}_o$	Plant node, $\mathcal{N}_o = \{0\}$.
$\mathcal{N}_w$	Set of warehouses, $\mathcal{N}_w = \{1, ..., N_w\}$.
$\mathcal{N}_c$	Set of customers, $\mathcal{N}_c = \{N_w + 1, ..., N_w + N_c\}$.
$\mathcal{N}$	Set of nodes, $\mathcal{N} = \mathcal{N}_o \cup \mathcal{N}_w \cup \mathcal{N}_c$.
$\mathcal{E}_{pw}$	Set of edges of the first echelon, $\mathcal{E}_{pw} = \{(i, j) : i, j \in \mathcal{N}_0 \cup \mathcal{N}_w, i < j\}$.
$\mathcal{E}_{wc}$	Set of edges of the second echelon, $\mathcal{E}_{wc} = \{(i, j) : i \in \mathcal{N}_w, j \in \mathcal{N}_c \vee i, j \in \mathcal{N}_c, i < j\}$.
$\mathcal{E}$	Set of edges, $\mathcal{E} = \mathcal{E}_{pw} \cup \mathcal{E}_{wc}$.
$\mathcal{R}$	Set of all possible routes in the first echelon, where $\mathcal{R} = \{1, \ldots, R\}$ and $R = 2^{N_w} - 1$.
$\mathcal{T}$	Set of time periods, $\mathcal{T} = \{1, ..., T\}$.
$\mathcal{K}_p$	Set of vehicles in the plant, $\mathcal{K}_p = \{1, ..., K_p\}$.
$\mathcal{K}_w$	Set of vehicles in warehouse $\mathcal{K}_w = \{\sum_{l=1}^{w-1} K_l + 1, ..., \sum_{l=1}^{w} K_w\}$.
$\mathcal{S}$	Set of scenarios, $\mathcal{S} = \{1, ..., S\}$.
$\tilde{\mathcal{E}}_{wc}(\eta)$	Subset of edges $(i, j) \in \mathcal{E}_{wc}$ such that $i, j \in \eta$ and $\eta \subseteq \mathcal{N}_w \cup \mathcal{N}_c$ is a given set of nodes.
$\tilde{\varepsilon}_{wc}(\{i\})$	Subset of edges incident to node $i \in \mathcal{N}_w \cup \mathcal{N}_c$.
Parameters:	
F	Setup cost.
u	Unit production cost.
$\tilde{c}_r$	Optimal cost of route $r \in \mathcal{R}$.
c_{ij}	Cost of visiting edge $(i, j) \in \mathcal{E}_{wc}$.
h_i	Per unit holding cost at node $i \in \mathcal{N}$.
α_i	Unit penalty cost of unmet demand for customer $i \in \mathcal{N}_c$.
C	Production capacity.
$\mathcal{Q}^p$	Capacity of vehicles located at the plant.
$\mathcal{Q}^w$	Capacity of vehicles located at warehouses.
a_{wr}	Binary parameter, equal to 1 if warehouse w is visited in route r, 0 otherwise.
ξ_s	Probability of scenario $s \in \mathcal{S}$.
d_{it}^s	Demand of customer i, in period t, under scenario s.
I_{i0}	Initial inventory at node $i \in \mathcal{N}$.
L_i	Storage capacity at node $i \in \mathcal{N}$.
$\mathcal{M}_w$	Upper bound on delivery quantity to warehouse w. $\mathcal{M}_w = \min\{\mathcal{Q}^p, L_w\}, \forall w \in \mathcal{N}_w$.
$\mathcal{M'}_{it}^{s}$	Upper bound on delivery quantity to customer i in period t under scenario s. $\mathcal{M'}_{it}^{s} = \min\{\mathcal{Q}^w, L_i, \sum_{l=t}^{T} d_{il}^s\}, \forall i \in \mathcal{N}_c, t \in \mathcal{T}, s \in \mathcal{S}$.

the total cost of production, storage, and transportation across all nodes and both echelons. The UB on deliveries to customers is also revised and should be replaced by $\mathcal{M''}_{it} = \min\left\{\mathcal{Q}^w, L_i, \sum_{l=t}^{T} d_{il}\right\} \quad \forall i \in \mathcal{N}_c, t \in \mathcal{T}$. The full mathematical formulation for the deterministic 2EPRP is given below.

Table C.2: Decision variables

Decision variables:	
First-stage Decisions:	
y_t	Setup decision, equal to 1 if a setup takes place in period $t \in \mathscr{T}$, and 0 otherwise.
p_t	Production quantity in period $t \in \mathscr{T}$.
I_{0t}	Inventory of plant at the end of each period t.
q_{wrt}	Quantity of products delivered to warehouse $w \in \mathscr{N}_w$ using route $r \in \mathscr{R}$ in period t.
o_{rt}	Binary variable, equal to 1 if route r is selected in period t in the first echelon, 0 otherwise.
Second-stage Decisions:	
I^s_{it}	Inventory of node $i \in \mathscr{N}_w \cup \mathscr{N}_c$ at the end of each period t in scenario s.
l^s_{it}	Unsatisfied demand of customer i in period t and scenario s.
w^s_{ikt}	Quantity of products delivered to customer $i \in \mathscr{N}_c$ by vehicle $k \in \bigcup_{w \in \mathscr{N}_w} \mathscr{K}_w$ in period t in scenario s.
z^s_{wkt}	Warehouse visiting decision, equal to 1 if node $w \in \mathscr{N}_w$ is visited by vehicle $k \in \mathscr{K}_w$ in period t in scenario s, 0 otherwise.
z^s_{ikt}	Customer visiting decision, equal to 1 if node $i \in \mathscr{N}_c$ is visited by vehicle $k \in \bigcup_{w \in \mathscr{N}_w} \mathscr{K}_w$ in period t in scenario s, 0 otherwise.
x^s_{ijkt}	Number of times edge $(i,j) \in \mathscr{E}_{wc}$ is traveled by vehicle $k \in \bigcup_{w \in \mathscr{N}_w} \mathscr{K}_w$ in period t and in scenario s.

$$\text{Min} \sum_{t \in \mathscr{T}} \Big(Fy_t + up_t + h_0 I_{0t} + \sum_{r \in \mathscr{R}} \tilde{c}_r o_{rt} + \sum_{w \in \mathscr{N}_w} I_{wt} + \sum_{w \in \mathscr{N}_w} \sum_{k \in \mathscr{K}_w} \sum_{(i,j) \in \mathscr{E}_{wc}} c_{ij} x_{ijkt} + \sum_{i \in \mathscr{N}_c} h_i I_{it} \Big) \tag{C.1}$$

s.t.

$$p_t \le C y_t \qquad \forall t \in \mathscr{T} \tag{C.2}$$

$$I_{0t} = I_{0,t-1} + p_t - \sum_{w \in \mathscr{N}_w} \sum_{r \in \mathscr{R}} q_{wrt} \qquad \forall t \in \mathscr{T} \tag{C.3}$$

$$I_{0t} \le L_0 \qquad \forall t \in \mathscr{T} \tag{C.4}$$

$$I_{wt} = I_{w,t-1} + \sum_{r \in \mathscr{R}} q_{wrt} - \sum_{i \in \mathscr{N}_c} \sum_{k \in \mathscr{K}_w} w_{ikt} \qquad \forall w \in \mathscr{N}_w, t \in \mathscr{T} \tag{C.5}$$

$$I_{wt} \le L_w \qquad \forall w \in \mathscr{N}_w, t \in \mathscr{T} \tag{C.6}$$

$$\sum_{r \in \mathscr{R}} a_{wr} o_{rt} \le 1 \qquad \forall w \in \mathscr{N}_w, t \in \mathscr{T} \tag{C.7}$$

$$\sum_{r \in \mathscr{R}} o_{rt} \le K_p \qquad \forall t \in \mathscr{T} \tag{C.8}$$

$$q_{wrt} \leq \mathscr{M}_w \, a_{wr} \qquad \forall w \in \mathscr{N}_w, r \in \mathscr{R}, t \in \mathscr{T} \quad \text{(C.9)}$$

$$\sum_{w \in \mathscr{N}_w} q_{wrt} \leq Q^p \, o_{rt} \qquad \forall r \in \mathscr{R}, t \in \mathscr{T} \quad \text{(C.10)}$$

$$I_{it} = I_{i,t-1} + \sum_{k \in \bigcup\limits_{w \in \mathscr{N}_w} \mathscr{K}_w} w_{ikt} - d_{it} \qquad \forall i \in \mathscr{N}_c, t \in \mathscr{T} \quad \text{(C.11)}$$

$$I_{it} + d_{it} \leq L_i \qquad \forall i \in \mathscr{N}_c, t \in \mathscr{T} \quad \text{(C.12)}$$

$$w_{ikt} \leq \mathscr{M}''_{it} \, z_{ikt} \qquad \forall i \in \mathscr{N}_c, k \in \bigcup_{w \in \mathscr{N}_w} \mathscr{K}_w, t \in \mathscr{T} \quad \text{(C.13)}$$

$$\sum_{i \in \mathscr{N}_c} w_{ikt} \leq Q^w \, z_{wkt} \qquad \forall w \in \mathscr{N}_w, k \in \mathscr{K}_w, t \in \mathscr{T} \quad \text{(C.14)}$$

$$\sum_{k \in \bigcup\limits_{w \in \mathscr{N}_w} \mathscr{K}_w} z_{ikt} \leq 1 \qquad \forall i \in \mathscr{N}_c, t \in \mathscr{T} \quad \text{(C.15)}$$

$$\sum_{(j,j') \in \tilde{\varepsilon}_{wc}(\{i\})} x_{jj'kt} = 2 \, z_{ikt} \qquad \forall i \in \mathscr{N}_w \cup \mathscr{N}_c, k \in \bigcup_{w \in \mathscr{N}_w} \mathscr{K}_w, t \in \mathscr{T} \quad \text{(C.16)}$$

$$\sum_{(i,j) \in \tilde{\mathscr{E}}_{wc}(\eta)} x_{ijkt} \leq \sum_{i \in \eta} z_{ikt} - z_{ekt} \qquad \forall \eta \subseteq \mathscr{N}_c, |\eta| \geq 2, e \in \eta, k \in \bigcup_{w \in \mathscr{N}_w} \mathscr{K}_w, t \in \mathscr{T} \quad \text{(C.17)}$$

$$y_t \in \{0,1\} \qquad \forall t \in \mathscr{T} \quad \text{(C.18)}$$

$$p_t \geq 0 \qquad \forall t \in \mathscr{T} \quad \text{(C.19)}$$

$$q_{wrt} \geq 0 \qquad \forall w \in \mathscr{N}_w, r \in \mathscr{R}, t \in \mathscr{T} \quad \text{(C.20)}$$

$$I_{0t} \geq 0 \qquad \forall t \in \mathscr{T} \quad \text{(C.21)}$$

$$o_{rt} \in \{0,1\} \qquad \forall r \in \mathscr{R}, t \in \mathscr{T} \quad \text{(C.22)}$$

$$I_{it} \geq 0 \qquad \forall i \in \mathscr{N}_w \cup \mathscr{N}_c, t \in \mathscr{T} \quad \text{(C.23)}$$

$$w_{ikt} \geq 0 \qquad \forall i \in \mathscr{N}_c, k \in \bigcup_{w \in \mathscr{N}_w} \mathscr{K}_w, t \in \mathscr{T} \quad \text{(C.24)}$$

$$z_{wkt} \in \{0,1\} \qquad \forall w \in \mathscr{N}_w, k \in \mathscr{K}_w, t \in \mathscr{T} \quad \text{(C.25)}$$

$$z_{ikt} \in \{0,1\} \qquad \forall i \in \mathscr{N}_c, k \in \bigcup_{w \in \mathscr{N}_w} \mathscr{K}_w, t \in \mathscr{T} \quad \text{(C.26)}$$

$$x_{ijkt} \in \{0,1\} \qquad \forall (i,j) \in \mathscr{E}_{wc}, i \notin \mathscr{N}_w, k \in \bigcup_{w \in \mathscr{N}_w} \mathscr{K}_w, t \in \mathscr{T} \quad \text{(C.27)}$$

$$x_{ijkt} \in \{0,1,2\} \qquad \forall (i,j) \in \mathscr{E}_{wc}, i \in \mathscr{N}_w, k \in \mathscr{K}_w, t \in \mathscr{T}. \quad \text{(C.28)}$$

C.3 Mathematical formulation for the deterministic 2EPRP with cross-docking satellites

In this section, we present the mathematical formulation for the deterministic two-echelon PRP with cross-docking satellites. Similar to the standard deterministic variant, the demand is assumed to be known in advance. All parameters and decision variables are consistent with those in the 2EPRP. However, inventory holding is no longer allowed at the satellites, which are used solely for transferring goods from the first-echelon vehicles to the second-echelon vehicles for delivery to customers. As mentioned earlier, we use the same notation for satellites to maintain consistency; thus, satellites are denoted by $w \in \mathscr{N}_w$. We assume a unit handling cost $\tilde{h}_w \ \forall w \in \mathscr{N}_w$, which is calculated as $\tilde{h}_w = \theta h_0$. The full mathematical formulation for the deterministic 2EPRP with cross-docking satellites is as follows.

$$\text{Min} \sum_{t \in \mathscr{T}} \Big(F y_t + u p_t + h_0 I_{0t} + \sum_{r \in \mathscr{R}} \tilde{c}_r o_{rt} + \sum_{w \in \mathscr{N}_w} \tilde{h}_w \sum_{i \in \mathscr{N}_c} \sum_{k \in \mathscr{K}_w} w_{ikt} + \sum_{w \in \mathscr{N}_w} \sum_{k \in \mathscr{K}_w} \sum_{(i,j) \in \mathscr{E}_{wc}} c_{ij} x_{ijkt} + \sum_{i \in \mathscr{N}_c} h_i I_{it} \Big) \tag{C.29}$$

s.t.

$$p_t \leq C y_t \qquad \forall t \in \mathscr{T} \tag{C.30}$$

$$I_{0t} = I_{0,t-1} + p_t - \sum_{w \in \mathscr{N}_w} \sum_{r \in \mathscr{R}} q_{wrt} \qquad \forall t \in \mathscr{T} \tag{C.31}$$

$$I_{0t} \leq L_0 \qquad \forall t \in \mathscr{T} \tag{C.32}$$

$$\sum_{r \in \mathscr{R}} q_{wrt} = \sum_{i \in \mathscr{N}_c} \sum_{k \in \mathscr{K}_w} w_{ikt} \qquad \forall w \in \mathscr{N}_w, t \in \mathscr{T} \tag{C.33}$$

$$\sum_{r \in \mathscr{R}} a_{wr} o_{rt} \leq 1 \qquad \forall w \in \mathscr{N}_w, t \in \mathscr{T} \tag{C.34}$$

$$\sum_{r \in \mathscr{R}} o_{rt} \leq K_p \qquad \forall t \in \mathscr{T} \tag{C.35}$$

$$q_{wrt} \leq Q^p a_{wr} \qquad \forall w \in \mathscr{N}_w, r \in \mathscr{R}, t \in \mathscr{T} \tag{C.36}$$

$$\sum_{w \in \mathscr{N}_w} q_{wrt} \leq Q^p \, o_{rt} \qquad \forall r \in \mathscr{R}, t \in \mathscr{T} \quad \text{(C.37)}$$

$$I_{it} = I_{i,t-1} + \sum_{k \in \bigcup_{w \in \mathscr{N}_w} \mathscr{K}_w} w_{ikt} - d_{it} \qquad \forall i \in \mathscr{N}_c, t \in \mathscr{T} \quad \text{(C.38)}$$

$$I_{it} + d_{it} \leq L_i \qquad \forall i \in \mathscr{N}_c, t \in \mathscr{T} \quad \text{(C.39)}$$

$$w_{ikt} \leq \mathscr{M}''_{it} \, z_{ikt} \qquad \forall i \in \mathscr{N}_c, k \in \bigcup_{w \in \mathscr{N}_w} \mathscr{K}_w, t \in \mathscr{T} \quad \text{(C.40)}$$

$$\sum_{i \in \mathscr{N}_c} w_{ikt} \leq Q^w \, z_{wkt} \qquad \forall w \in \mathscr{N}_w, k \in \mathscr{K}_w, t \in \mathscr{T} \quad \text{(C.41)}$$

$$\sum_{k \in \bigcup_{w \in \mathscr{N}_w} \mathscr{K}_w} z_{ikt} \leq 1 \qquad \forall i \in \mathscr{N}_c, t \in \mathscr{T} \quad \text{(C.42)}$$

$$\sum_{(j,j') \in \tilde{\varepsilon}_{wc}(\{i\})} x_{jj'kt} = 2 \, z_{ikt} \qquad \forall i \in \mathscr{N}_w \cup \mathscr{N}_c, k \in \bigcup_{w \in \mathscr{N}_w} \mathscr{K}_w, t \in \mathscr{T} \quad \text{(C.43)}$$

$$\sum_{(i,j) \in \tilde{\mathscr{E}}_{wc}(\eta)} x_{ijkt} \leq \sum_{i \in \eta} z_{ikt} - z_{ekt} \qquad \forall \eta \subseteq \mathscr{N}_c, |\eta| \geq 2, e \in \eta, k \in \bigcup_{w \in \mathscr{N}_w} \mathscr{K}_w, t \in \mathscr{T} \quad \text{(C.44)}$$

$$y_t \in \{0,1\} \qquad \forall t \in \mathscr{T} \quad \text{(C.45)}$$

$$p_t \geq 0 \qquad \forall t \in \mathscr{T} \quad \text{(C.46)}$$

$$q_{wrt} \geq 0 \qquad \forall w \in \mathscr{N}_w, r \in \mathscr{R}, t \in \mathscr{T} \quad \text{(C.47)}$$

$$I_{0t} \geq 0 \qquad \forall t \in \mathscr{T} \quad \text{(C.48)}$$

$$o_{rt} \in \{0,1\} \qquad \forall r \in \mathscr{R}, t \in \mathscr{T} \quad \text{(C.49)}$$

$$I_{it} \geq 0 \qquad \forall i \in \mathscr{N}_c, t \in \mathscr{T} \quad \text{(C.50)}$$

$$w_{ikt} \geq 0 \qquad \forall i \in \mathscr{N}_c, k \in \bigcup_{w \in \mathscr{N}_w} \mathscr{K}_w, t \in \mathscr{T} \quad \text{(C.51)}$$

$$z_{wkt} \in \{0,1\} \qquad \forall w \in \mathscr{N}_w, k \in \mathscr{K}_w, t \in \mathscr{T} \quad \text{(C.52)}$$

$$z_{ikt} \in \{0,1\} \qquad \forall i \in \mathscr{N}_c, k \in \bigcup_{w \in \mathscr{N}_w} \mathscr{K}_w, t \in \mathscr{T} \quad \text{(C.53)}$$

$$x_{ijkt} \in \{0,1\} \qquad \forall (i,j) \in \mathscr{E}_{wc}, i \notin \mathscr{N}_w, k \in \bigcup_{w \in \mathscr{N}_w} \mathscr{K}_w, t \in \mathscr{T} \quad \text{(C.54)}$$

$$x_{ijkt} \in \{0,1,2\} \qquad \forall (i,j) \in \mathscr{E}_{wc}, i \in \mathscr{N}_w, k \in \mathscr{K}_w, t \in \mathscr{T}. \quad \text{(C.55)}$$

C.4 The Branch-and-Cut algorithm

In this section, we provide the details of the branch-and-cut (BC) algorithm used to solve both the stochastic and deterministic versions of the problem. Since the BC structure is similar for both cases, we present the algorithm for the stochastic version. With minor modifications, the same approach can be applied to the deterministic version. The problem is solved with CPLEX, and a callback function is employed to generate subtour elimination constraints (SECs).

It is important to emphasize that, as mentioned earlier, the BC algorithm is primarily used to evaluate the quality of solutions obtained by the hybrid heuristic algorithm (HHA), rather than serving as the main solution method. In all cases, the solution produced by the HHA is provided to CPLEX as an initial feasible solution.

As the first-echelon routes are explicitly defined in the model, there is no need to generate SECs for that part of the problem. For each subset of warehouses, a traveling salesman problem (TSP) is solved to determine the optimal visiting sequence and corresponding cost. All possible combinations of warehouse visits are enumerated and added to the model. Since the number of warehouses is assumed to be small, the total number of possible combinations does not exceed $2^W - 1$ (for instances with W warehouses), which can be processed in a few seconds, since $W \leq 5$. The Concorde algorithm is used to solve these TSP problems efficiently.

In contrast, the number of SECs required in the second echelon grows exponentially with the number of customers. For the stochastic case, these constraints must be generated for each scenario, which adds significant complexity. Therefore, constraints 3.17 are relaxed at the beginning and added to the model only when violated. To identify violated SECs, we solve a separation problem formulated as a minimum *s*–*t* cut for each scenario. We use the algorithm provided by the Concorde library, which is known for its efficiency. Once violations are detected, the corresponding constraints are added to exclude the infeasible solution, and the BC algorithm continues from there. SECs are checked only at the root node and for integer solutions, as this has proven to be more efficient in practice than checking at every node in the branch-and-bound tree.

For solving the problem, we set the CPLEX MIP emphasis parameter to prioritize optimality, and allocate 32 GB of memory and 10 threads.

In addition to SECs, we add several classes of valid inequalities to improve the efficiency of the BC algorithm. The first set consists of logical inequalities that ensure a vehicle $k \in \mathscr{K}_w$ can only visit customer $i \in \mathscr{N}_c$ if it is dispatched from warehouse w:

$$z_{ikt}^s \leq z_{wkt}^s \qquad \forall i \in \mathscr{N}_c, w \in \mathscr{N}_w, k \in \mathscr{K}_w, t \in \mathscr{T}, s \in \mathscr{S}. \quad \text{(C.56)}$$

The second set of logical inequalities imposes that an edge $(i, j) \in \mathscr{E}_{wc}$ can only be used if both nodes i and j are visited:

$$x_{ijkt}^s \leq z_{ikt}^s \qquad \forall i, j \in \mathscr{N}_c \cup \mathscr{N}_w, k \in \bigcup_{w \in \mathscr{N}_w} \mathscr{K}_w, t \in \mathscr{T}, s \in \mathscr{S}. \quad \text{(C.57)}$$

$$x_{ijkt}^s \leq z_{jkt}^s \qquad \forall i, j \in \mathscr{N}_c \cup \mathscr{N}_w, k \in \bigcup_{w \in \mathscr{N}_w} \mathscr{K}_w, t \in \mathscr{T}, s \in \mathscr{S}. \quad \text{(C.58)}$$

The third set includes symmetry-breaking constraints that prevent a vehicle with index k from being dispatched unless the vehicle with index $k-1$ is already dispatched from the same warehouse:

$$z_{wkt}^s \leq z_{w,k-1,t}^s \qquad \forall w \in \mathscr{N}_w, k \in \mathscr{K}_w, t \in \mathscr{T}, s \in \mathscr{S}. \quad \text{(C.59)}$$

The last set of valid inequalities are the lexicographic constraints, which further help break symmetry by imposing an ordering on customer visits across vehicles. These constraints ensure that the sequence of customers visited by vehicle k is lexicographically smaller than or equal to that of vehicle $k-1$:

$$\sum_{i=1}^{j} 2^{j-i} z_{ikt}^s \leq \sum_{i=1}^{j} 2^{j-i} z_{i,k-1,t}^s \qquad \forall j \in \mathscr{N}_c, w \in \mathscr{N}_w, k \in \mathscr{K}_w, t \in \mathscr{T}, s \in \mathscr{S}. \quad \text{(C.60)}$$

Lexicographic constraints are especially helpful when multiple vehicles are assigned to the same warehouse, as they eliminate redundant solutions that differ only in the indexing of vehicles. All of these valid inequalities are also adjusted and added to the deterministic problem.