CORA Cork Open Research Archive
Cartlann Taighde Oscailte Chorcaí

Title	Computing the policy parameters for perishable inventory management and cold supply chains under stochastic demand
Authors	Gulecyuz, Suheyl
Publication date	2025-10-31
Original Citation	Gulecyuz, S. 2025. Computing the policy parameters for perishable inventory management and cold supply chains under stochastic demand. PhD Thesis, University College Cork.
Type of publication	Doctoral thesis
Rights	© 2025, Suheyl Gulecyuz. - https://creativecommons.org/licenses/by-nc-nd/4.0/
Download date	2026-10-07 10:34:02
Item downloaded from	https://hdl.handle.net/10468/18799

UCC
University College Cork, Ireland
Coláiste na hOllscoile Corcaigh

Computing the Policy Parameters for Perishable Inventory Management and Cold Supply Chains Under Stochastic Demand

Suheyl Gulecyuz
MSc
120220024

NATIONAL UNIVERSITY OF IRELAND, CORK

SCHOOL OF COMPUTER SCIENCE AND INFORMATION TECHNOLOGY

COLLEGE OF SCIENCE, ENGINEERING AND FOOD SCIENCE

Thesis submitted for the degree of Doctor of Philosophy

Oct 2025

Head of Department: Prof. Dirk Pesch

Supervisors: Prof. Barry O'Sullivan
Dr. Andrea Visentin
Prof. S. Armagan Tarim (external)

Contents

List of Figures . . . iii
List of Tables . . . v
Abstract . . . viii
Acknowledgements . . . x

1 Introduction **1**
1.1 Motivation . . . 1
1.2 Thesis Statement . . . 6
1.3 Contributions . . . 7
1.4 Thesis Structure . . . 8

2 Literature Review and Background **10**
2.1 Inventory Control . . . 10
2.1.1 Inventory Control Concepts . . . 10
2.1.2 Lot-Sizing Problem . . . 20
2.1.3 Perishability in Inventory Control . . . 32
2.1.4 Cold Food Supply Chains . . . 40
2.2 Solution Techniques . . . 45
2.2.1 Dynamic Programming . . . 47
2.2.2 Mixed Integer Programming . . . 50
2.2.3 Inventory Routing Problem . . . 53

3 Determining the Policy Parameters to Improve the Sustainability of Cold Supply Chains Under Stochastic Demand **56**
3.1 Introduction . . . 56
3.2 Problem Description . . . 58
3.2.1 Our Problem . . . 58
3.2.2 CO_2 Emissions . . . 59
3.2.3 Soysal et al.'s (R, Q) Model . . . 61
3.2.4 Proposed (R, S) Model . . . 63
3.2.5 Deterministic-Equivalent Approximation . . . 65
3.3 Numerical Experiments . . . 66
3.3.1 Experimental Design . . . 67
3.3.2 Base Case Results . . . 68
3.3.3 Performance of the (R, S) Model . . . 73
3.3.4 Modified Larger Case Study . . . 79
3.3.5 Planning Horizon Length vs Cumulative Ending Inventory and Order Size . . . 80
3.4 Concluding Remarks and Managerial Insights . . . 82

4 Determining the Policy Parameters for Perishable Inventory Management Under Penalty Cost **84**
4.1 Introduction . . . 84
4.2 Problem Description and Our Model . . . 85
4.2.1 Problem Definition . . . 85

4.2.2 Mathematical Model 86
4.2.3 Stochastic Dynamic Programming (SDP) Formulation . . 87
4.3 Network Representation 89
4.4 Replenishment Cycle Order-Up-To Level Calculation 91
4.4.1 Re-Formulation of the Total Cost Function 91
4.4.2 Calculating the Expected Outdating 92
4.4.3 Multi-Period Newsvendor Setting and Calculating the Optimal Parameters 93
4.5 Heuristic Approach for the Penalty Cost Case 95
4.5.1 Overview 96
4.5.2 Step 1: Determining the Replenishment Periods 97
4.5.3 Step 2: Determining the Order Quantities 99
4.5.4 An Example 101
4.6 Numerical Experiments 104
4.6.1 Experimental Setup 104
4.6.2 Results and Discussion 106
4.7 Conclusion 111

5 Determining the Policy Parameters for Perishable Inventory Management Under Service Level Constraints 113
5.1 Introduction 113
5.2 Problem Description and Our Model 114
5.2.1 Mathematical Model 115
5.2.2 Stochastic Dynamic Programming Approach 117
5.3 Network Representation 119
5.4 Heuristic Approach for the Service Level Case 120
5.4.1 Calculating the Order-up-to Levels 120
5.4.2 Step 1: Determining the Replenishment Schedule 121
5.4.3 Step 2: Calculating the Order Quantities 122
5.4.4 An Example 123
5.5 Numerical Experiments 126
5.5.1 Experimental Setup 126
5.5.2 Results and Discussion 127
5.6 Conclusion 132

6 Conclusions & Future Work 135
6.1 Conclusions 135
6.2 Future Directions 136
6.2.1 Cold Food Supply Chains 137
6.2.2 Perishable Inventory Management 138

List of Figures

2.1 Various examples of demand mean patterns. 14
2.2 The expected inventory level under the base-stock policy 29
2.3 The expected inventory level under the (s, S) policy 30
2.4 The expected inventory level under the (R, Q) policy 31
2.5 The expected inventory level under the (R, S) policy 32
2.6 The expected inventory level under the (R, s, S) policy 33

3.1 Schematic representation of the supply chain network. 58
3.2 Comparison of the cumulative ending inventory levels and percentage differences found via the (R, Q) and (R, S) models. . . 81

4.1 The network representation of all possible replenishment cycles of a perishable inventory model with $T = 4$ and $M = 3$. Notice that each Node $i = 1, ..., T - 1$ represent the end of Period i and the beginning of Period $i + 1$, Node 0 and Node T represent the beginning and the ending of the planning horizon, respectively, and each arc $(i, j) : i = 0, ..., T - 1, j = 1, ..., T, i < j$, represents $R(i + 1, j)$. 91
4.2 The only possible path to the end of period 1. 98
4.3 The possible paths to the end of period 2. 98
4.4 The possible paths to the end of period 3. 98
4.5 The network representation of the example problem with all possible cycle costs. The best path is highlighted in red. 101
4.6 The network representation of the example problem with all possible inventory compositions. The best path is highlighted in red. The first and the second row represent the starting inventory at the beginning of the replenishment cycle and the inventory composition after the order is placed, respectively. The remaining rows show the inventory compositions at the end of each period in the replenishment cycle. The columns represent the number of items of each age; the first and the second columns show the number of items of Age 0 and 1, respectively. 101
4.7 The network representation of the example problem with the previous replenishment cycle information. The best path is highlighted in red. 102
4.8 The best path at the end of period 1 with the corresponding cycle cost values. 102
4.9 The best path at the end of period 2 with the corresponding cycle cost values. 103
4.10 Demand patterns for each demand set. 105

5.1 The network representation of all possible replenishment cycles of a perishable inventory model with $T = 3$ and $M = 2$. Notice that each Node $i = 1, ..., T - 1$ represent the end of Period i and the beginning of Period $i + 1$, Node 0 and Node T represent the beginning and the ending of the planning horizon, respectively, and each arc $(i, j) : i = 0, ..., T - 1, j = 1, ..., T, i > j$, represents $R(i + 1, j)$. 120
5.2 The best path at the end of period 1 with the corresponding cycle cost values. 124
5.3 The best path at the end of period 2 with the corresponding cycle cost values. 126

List of Tables

2.1 Summary of the non-stationary demand literature in perishable inventory control. D, S, and NS represent deterministic, stationary stochastic, and non-stationary stochastic demand, respectively. 41
2.2 Summary of the CFSC literature review. D, S, and NS represent deterministic, stationary stochastic, and non-stationary stochastic, respectively. 46

3.1 Input parameters for the experiments. 68
3.2 Demand patterns and parameters. 69
3.3 Vehicle and emission parameters. 70
3.4 Refrigeration parameters. 70
3.5 Distances between the nodes (km). Retailers 5-11 are used in the extended case. 71
3.6 Summary of the base case optimisation results. 74
3.7 Summary of the base case simulation results, 10,000 reps. . . . 75
3.8 The replenishment plan and the order-up-to levels for the base case. 75
3.9 The order quantities and the ending inventory levels for the base case. 76
3.10 Achieved service levels (%), $\alpha = 0.95$, 10,000 reps. 77
3.11 Assigned routes for each vehicle and the number of assigned vehicles. 77
3.12 Comparison of the cumulative order quantities and ending inventory levels 77
3.13 Avg percentage increases of RQ above RS $(e.g. 100*(RQ/RS-1))$. All the other parameter values of the instances marked with (*) are the same as of the base case. 78
3.14 Demand parameters for the extended case. 79
3.15 Summary of the extended case optimisation results. 80
3.16 Assigned routes for each vehicle and the number of assigned vehicles for the extended case. 80
3.17 Comparison of the cumulative order quantities and ending inventory levels for the extended case. 80

4.1 Input parameters for the experiments 105
4.2 Demand sets, demand patterns, and arrival rates 105
4.3 Comparison of the SDP and the heuristic, penalty cost case, with respect to each parameter and the average computation times (in seconds) per instance (5,000 simulation reps each). Recall that the heuristic was implemented in MATLAB and the SDP model was implemented in C, and the SDP runtimes would be much longer in MATLAB. 109
4.4 Input parameters for the experiments 110
4.5 Demand means. Modified from the parameters in [PWHHvdV14] 110

4.6 Comparison of the SDP and the heuristic, penalty cost case, with respect to each parameter, penalty cost case. 110
4.7 Demand means taken from [PWHHvdV14] 111
4.8 Comparison of the average heuristic runtimes (seconds) for different product lifetimes, penalty cost case. 111

5.1 Input parameters for the experiments 127
5.2 Demand sets, demand patterns, and arrival rates 127
5.3 Comparison of the SDP and the heuristic, service level case, with respect to each parameter and the average computation times (in seconds) per instance (5,000 simulation reps each). 131
5.4 Input parameters for the experiments 132
5.5 Comparison of the SDP and the heuristic, service level case, with respect to each parameter. 133
5.6 Comparison of the average heuristic runtimes (seconds) for different product lifetimes, service level case. 134

I, Suheyl Gulecyuz, certify that this thesis is my own work, AI has not been used, any work of others that has been used or referenced is fully cited in the text, and I have not obtained a degree in this university or elsewhere on the basis of the work submitted in this thesis.

Suheyl Gulecyuz

Abstract

The objective of the work in this dissertation to introduce techniques to compute the near-optimal policy parameters for the perishable inventory problem and sustainable cold food supply chains with environmental concerns under stochastic demand, as well as tackling and dealing with the demand uncertainty and fluctuations arising from the nature of demand patterns. The solutions we present in this dissertation for the perishability problem hybridises the well-known Wagner-Whitin algorithm, the Silver-Meal heuristic on a network graph setting. Our solution for sustainable cold supply chains employs and combines the inventory routing problem and several formulations found in the literature to calculate environmental effects.

We consider a perishable inventory system under a finite planning horizon, periodic review, non-stationary stochastic demand, zero lead time, first in first out issuing policy, and a fixed shelf life. The inventory system has a fixed setup cost and linear ordering, holding, and outdating costs per item. We consider the penalty cost and the service level cases separately and develop two different versions for each case. In addition, we propose a mixed-integer programming model for cold food supply chains to solve a multi-period inventory routing problem under non-stationary stochastic demand, route-dependent costs, and environmental concerns. The study aims to find the replenishment and vehicle routing plans to minimise the total expected cost while producing a minimum amount of CO_2 emissions.

The first contribution presented herein is a mixed-integer linear programming model that computes near-optimal (R, S) policy parameters for the sustainable cold food supply chain problem with environmental concerns, while maintaining flexibility in ordering decisions under a pre-determined replenishment schedule. Our numerical experiments show that the (R, S) policy reduces inventory costs significantly, since it solves the excess inventory issue caused by higher excess ending inventory levels arising from the pre-determined and inflexible order quantities in the (R, Q) policy. However, the CO_2 emission levels and routing costs remain similar in both models. The reduced inventory costs make the (R, S) policy significantly reduce the total cost. In addition, our numerical experiments show that the difference between the cumulative ending inventory levels for the (R, Q) and (R, S) policies increasingly grows as the time horizon gets longer, and it results in increasingly larger differences in the total cost values for both policies.

The second contribution is a heuristic for the penalty cost case of the perishable inventory problem. The problem can be solved to optimality by the stochastic dynamic programming technique; however, due to the increasing dimensionality of the inventory state which is defined by the inventory levels of each item age, it becomes impractical for longer shelf lives. Our heuristic represents the inventory model as a network graph, and then calculates the shortest path in the graph in a recursive way, based on the hybridisation of the Wagner-Whitin algorithm and the Silver-Meal heuristic. It firstly determines the replenishment periods and cycles using the deterministic-equivalent shortest path approach. Using the replenishment plan determined in the first step, it calculates the order quantities based on the demand observations as a second step. We hereby employ Bookbinder and Tan's static-dynamic uncertainty strategy to respond to demand fluctuations. Our extensive numerical studies show that the computation time is significantly reduced and the heuristic finds near-optimal policy parameters.

The final contribution is the modification of our proposed heuristic for the service-level case. To our knowledge, there is no procedure in the literature that calculates the optimal solution. Our extensive computational experiments that use the near-optimal stochastic dynamic programming solution as a benchmark show that the heuristic has a similar performance in terms of near-optimality and significantly reduced computation times.

Acknowledgements

Firstly, I would like to express my sincere gratitude to my primary supervisor Prof. Barry O'Sullivan, second supervisor Dr. Andrea Visentin, and external and former primary supervisor Prof. S. Armagan Tarim, for the incredible opportunity they offered me and for their endless support, guidance, and patience during my PhD study. It has always been a pleasure to work with them. Undertaking this PhD under their supervision has been a truly life-changing experience. I sincerely thank them for being so supportive and encouraging. I learned so much from them. I hope and believe that this work is the beginning of a longer collaboration with them.

I would like to thank the internal and external examiners, Prof. Steven Prestwich and Dr. Xiyuan Wu, for their insightful feedback and interesting discussion for the improvement of this work.

I wish to express my sincere gratitude to all the Computer Science Department and the Insight Centre For Data Analytics. I would like to thank my supervisor and the head of Insight Prof. Barry O'Sullivan again for offering this research opportunity at Insight. My thanks go to all the Insight staff for their help, support, professionalism and kindness.

I would also like to thank Dr. Selja Seppälä and Dr. Hadi Karimikia from the Business Information Systems Department, and Prof. S. Armagan Tarim, for offering teaching opportunities and allowing me to improve my teaching skills in my PhD journey. I also want to thank all the fellow tutors I worked with during this experience, for their contributions, collaboration, and friendship.

I would like to thank all the colleagues in our lab and the other labs. We shared exciting discussions, challenges, events, lunches, travels and much more. I am proud to call many of them not only colleagues but friends.

I am incredibly fortunate to have such wonderful parents. I want to thank all the Gulecyuz family. They have been a source of inspiration, laughs, love and kindness. My family is the dearest thing to me. Without their efforts and support, I would never have achieved all my accomplishments.

There are so many people that I should thank for making my Irish experience unforgettable. I cannot thank all the people that have been important to me in these years, but I will try to express my gratitude to them as much as I can.

Chapter 1

Introduction

In this chapter, we briefly discuss the focus and contributions of this dissertation. We present the motivation behind our work in Section 1.1. Then we present our goal and thesis statement in Section 1.2. We summarise the contributions of our work and the list of the publications based on this work in Section 1.3. The structure of this dissertation is given in Section 1.4.

1.1 Motivation

One of the major challenges for leading practitioners in supply chain management, such as Wal-Mart, is to deal with the significant amount of perishable products in their portfolio, which consists of food items (dairy, bakery, fruit, vegetable, etc.), pharmaceuticals (drugs, cosmetics, etc.), chemicals (cleaning products etc.), and cut flowers, [KSWD11]. Perishability brings operational challenges and safety concerns due to the deterioration in product quality over time. For instance, spoiled food items might affect fresh items placed next to them and speed up their spoilage process, or result in serious health issues such as poisoning due to spoilage. Similarly, expired medical drugs might have serious affects on health. Apart from outdating risks itself, the impact of outdating in society is another concern to take into account, [KSWD11]. According to a survey conducted in 2003, 22% of overall unsalable costs of $2.57 billion from distributors to retailers (supermarkets, pharmacies, etc.), which was over $500 million, were due to expired products, [Ame04]. Similarly, Webb [Web06] states that spoilage results in $300 million annual loss in the $1.7 billion US apple industry. Moreover, blood is subject to perishability and one of the most critical resources in healthcare supply chains. According to Whitaker

[WS05], 5.8% of blood processed for transfusion were outdated in 2004 in the USA. In addition to consumer goods and healthcare resources, several industrial products, such as plywood which loses strength within one week of production, are subject to perishability, [Che06]. In contrast to non-perishable products, perishable products will become outdated and no longer be available to satisfy customer demand unless they meet customers within a finite time period. The limited time window to meet customer demand and the risks and the economic impacts of outdating make perishable inventory management a critical subject in supply chain management. Additionally, the product age is another concern in the area of perishable inventory management and plays a crucial role in making inventory decisions. Moreover, the increase in product lifetime enhances the complexity of the problem in terms of computational requirements. Our motivation in this study is led by the need for a practical solution that is minimally affected by the complexity of the perishability problem and makes near-optimal decisions.

Another concern in supply chain management, which can be directly or indirectly related to perishability, is the transportation of goods from suppliers to retailers. Transporting goods from suppliers and requires careful planning for efficient use and distribution of vehicles. In addition, since vehicles are dependent on fuel consumption during transportation activities, fuel costs and environmental impacts caused by fuel consumption, such as CO_2 emissions, are other significant factors to develop sustainable models. In case when refrigeration activities are included, such as in cold food supply chains, fuel consumption and emissions must also be considered for inventory decisions. According to the UN Environmental Agency [SG17], freight storage and transportation activities will generate 36% of all permissible emissions by 2050. Moreover, transporting and storing temperature-sensitive products requires additional energy consumption to keep them at an ideal temperature, resulting in higher CO_2 emissions, [BSA$^+$20]. A well-designed cold food supply chain model is required to manage these costs and CO_2 emissions. The need for a practical cold food supply chain model that explores the impact of non-stationary stochastic demand on operational decisions of storage and transportation in cold food supply chains leads to our motivation hereby.

The challenges related to perishability, transportation, and sustainability, which is discussed above, highlight the critical importance of inventory control in supply chain management. Effective inventory control systems must consider the limited shelf life of goods in the perishability problem and the environmental

and financial impacts of transportation in cold food supply chains. Therefore, keeping track of inventory and customer demand becomes a vital point in supply chain systems. Inventory control systems aim to monitor the amount of items ready to meet customer demand and used to make ordering decisions to satisfy customer demand. Optimal inventory decisions lead to significant cost savings and adequate customer satisfaction.

Inventory control is one of the fundamental and well-studied areas in operations research. The inventory control literature began with the work of Harris [Har13]. Since then, a significant body of literature has emerged in the inventory control area. The inventory control literature includes various methods and techniques that aim to manage and control production/inventory systems with different structures, constraints, and environments, and subject to deterministic (static or dynamic) or stochastic (stationary or non-stationary) customer demand. The timing and the amount of orders are two of the most well-known issues in the inventory theory. Among these issues, the problem of determining the appropriate order quantity, also known as the lot-sizing problem, has received considerable attention in the area of inventory control.

The lot-sizing problem aims to calculate the amount of the replenishment orders. The EOQ model [Edg88] is one of the oldest classical models considering the inventory systems in which demand is constant. The dynamic lot-sizing problem focuses on inventory systems where demand may change over time. There are many efficient methods in the literature such as the Wagner-Whitin algorithm, [WW58], the Silver-Meal heuristic, [Sil73], the various modifications of these techniques, etc., which were developed to deal with the dynamic lot-sizing problem. In most real life inventory systems, customer demand is random. Several strategies have been proposed in the literature, such as the static, dynamic, and static-dynamic uncertainty strategies of [BT88] to tackle uncertainties due to the random nature of demand. The static uncertainty allows no flexibility in ordering decisions while the dynamic uncertainty strategy allows flexibility and the static-dynamic uncertainty strategy offers a balanced approach.

An interesting subject in the inventory theory is the one that studies the production/inventory systems with perishable goods which are called perishable production/inventory systems. In contrast to most inventory models in the literature assuming that products or items have an infinite shelf life and they stay in the inventory indefinitely unless they are not subject to customer demands,

perishable inventory models consist of the items or products which become outdated after a fixed or variable shelf life or due to the environmental factors. Dairy products, meat products, fruits and vegetables, blood products, medical drugs, etc. are well-known perishable products and items in both real life problems and the literature, [KSWD11]. In perishable inventory systems, replenishment policies must take several parameters into account (e.g., the amount of estimated waste, the item lifetime and the number of perishable items of a specific age) in order to minimise the total cost, waste, etc. In addition, various policies based on the item age or remaining shelf life are issued in order to determine which items should be used first to meet the customer demand. FIFO (first in, first-out), LIFO (last in, first-out), FEFO (first expired, first-out), LSFO (least shelf life, first out), and LEFO (last expired, first out) are several examples of these issuing policies, [Nah11], [MCS$^+$20], [ÖRSVdH15].

The perishable inventory problem has challenges that cannot be solved by the techniques developed for non-perishable inventory systems, in particular the multi-dimensionality of the inventory state consisting of the amount of items of each age and the calculating expected waste. In perishable inventory systems, the inventory levels of items of each age has a critical role for making ordering decisions. Considering the multi-dimensionality of the inventory state, we have to handle a very large number of possible inventory compositions that consist of the inventory levels of each age, which is extremely hard to deal with. For instance, if the lifetime is 3 periods and the maximum inventory level for each age is 100, we have $100^3 =$ possible inventory compositions. If we increase the lifetime to 4, the number of possible inventory compositions will be 100^4. Moreover, since items do not stay in the inventory indefinitely, we need to take the expected waste into account when making ordering decisions. However, the expected number of outdated items (waste) has no explicit formulation and is computationally challenging, [Nah75], and can only be solved several approximation algorithms and heuristics (e.g. the EWA policy of Broekmeulen and Donselaar [BVD09] in the literature.

The perishability problem can be solved to optimality by using the stochastic dynamic programming (SDP) approach for the penalty cost case. For the service level case, the SDP does not necessarily generate optimal solutions. Since we have to take all the possible combinations of the amount of items of different ages, computation times increase exponentially as the item lifetime grows and the SDP approach quickly becomes computationally infeasible in terms of both the computation times and the amount of memory needed.

As a contribution to this area, we develop a mixed integer linear programming (MILP) model for the inventory routing problem (IRP) for cold food supply chains (CFSCs), which also considers environmental factors (CO_2 emissions). In traditional food logistics, reducing costs and increasing responsiveness were the primary objectives of a supply chain network. However, sustainability studies are becoming increasingly important to protect the environment and also to improve food supply chains' social and economic aspects, see Bottani et al. [BCNT19]. Amorim et al. [AAL14] focus on environmental measures of CO_2 emissions within supply chain models, and Shashi et al. [SCS$^+$18] state that one of the primary sources of CO_2 emissions in a food supply chain is the transportation and inventory activities. Moreover, transporting and storing temperature-sensitive products requires additional energy consumption to keep them at an ideal temperature, resulting in higher CO_2 emissions, [BSA$^+$20]. A well-designed CFSC model is required to manage these costs and CO_2 emissions. The need for a practical CFSC model that explores the impact of non-stationary stochastic demand on operational decisions of storage and transportation in CFSCs leads to our motivation here. Our proposed IRP model aims to assist decision-makers in reducing various costs associated with CFSCs—such as holding, transportation, unit ordering, fixed replenishment, and refrigeration—while also focusing on CO_2 emission reduction.

The well-known IRP allows combining both inventory management and routing problems. The NP-hard inventory routing problem handles inventory management and vehicle routing simultaneously, [JRK13]. However, a practical model could be achieved using the IRP. Limited studies have been conducted on inventory routing problems with non-stationary stochastic demand consideration. In most existing studies, the demand is stationary. However, this is not compatible with most real life cases, and it also creating a gap in the IRP and CFSC which creates an opportunity to investigate IRP in a non-stationary stochastic environment. Soysal et al. [SBRHVDV15] presented a model to handle inventory and routing decisions using the (R, Q) policy. This policy pre-determines replenishment timing and order quantities before the start of the planning horizon, and it does not allow enough flexibility in making replenishment decisions. One of our aims in our IRP and CFSC study is to fill this gap and make a significant contribution to the field.

The IRP model we developed in this dissertation employs the formulation for (R, S) policy, presented in [TK04]. Using the (R, S) policy, replenishment periods are pre-determined and cannot be modified, while order quantities are

determined after observing the demand realisations, in a similar manner to the static-dynamic strategy of [BT88]. The presented model under the (R, S) policy provides a more effective way to deal with non-stationary demand uncertainty.

1.2 Thesis Statement

In most real life inventory problems, demand uncertainty must taken into account when making ordering decisions, in order to ensure customer demand is satisfied. Since the static policies do not respond demand uncertainty well, they tend to find suboptimal solutions with higher total inventory costs. Therefore, employing ordering policies that tackle demand uncertainty is crucial. However, fully dynamic policies might have some drawbacks, such as computational complexity or higher development and operating cost.

The (R, S) policy is a balanced approach to maintain flexibility while reducing operational costs and computational challenges in the area of perishable inventory management and cold food supply chains with non-stationary stochastic demand. Moreover, hybridising several well-known techniques in the literature helps dealing with the challenges caused by the state space dimensionality in perishable inventory systems, and allow developing computationally fast methods which are minimally affected by the dimensionality. Applying the well-known IRP is an efficient way to combine inventory and transportation decisions.

Based on the claims given above, we assert that employing the (R, S) ordering policy is an effective way to handle demand uncertainties in non-stationary stochastic demand cases, while developing a timely and computationally efficient heuristic for perishable inventory systems and developing an IRP model for cold food supply chains which takes environmental concerns and refrigeration costs into account.

The work in this dissertation present heuristics and models in the next chapters that are:

- *Innovative in the problem configuration.* The non-stationary stochastic demand case is under-studied in the literature for both perishable inventory management and cold food supply chains. We aim to fill this gap in the literature by studying this case and developing efficient models and methods.

- *Computable in reasonable time.* Our proposed heuristic for perishable inventory systems is minimally affected by the dimensionality caused by the item lifetime and it outperforms the optimal SDP solution which has exponentially growing solution times as the dimensionality increases. Moreover, our IRP model can be solved in terms of minutes which is a reasonable computation time.
- *Novel in terms of hybridisation of different techniques.* Our proposed heuristic for perishable inventory systems employs different algorithms, heuristics and methods such as Wagner-Whitin, Silver-Meal and [RTHP11]'s state space network graph representation. Based on the hybridisation of these techniques, it proposes a novel approach. Moreover, the IRP model for CFSCs incorporate and hybridise environmental concerns, fuel consumption and refrigeration formulas, transportation and inventory costs, and the static-dynamic (R, S) ordering policy, in order to present a novel mathematical model.
- *Near-optimal.* The heuristics and mathematical models presented in this dissertation result in near-optimal solutions with little optimality gap.

1.3 Contributions

Our contributions to the literature with the work in this dissertation are listed and summarised as follows:

- Developing a computationally and time-efficient heuristic with little optimality gap, in order to calculate the replenishment schedule and order quantities in perishable inventory systems with non-stationary stochastic demand.
- Incorporating the well-known heuristics and algorithms in the lot-sizing literature, such as the Wagner-Whitin and Silver-Meal, into perishable inventory management by developing a heuristic inspired by them.
- Dealing with the *curse of dimensionality* arising as the item lifetime increases, by employing the network representation approach which allows to develop a fast heuristic.
- Employing the heuristic for both the penalty cost and service level cases.
- Formulating a MILP IRP model for the problem with environmental con-

cerns, such as CO_2 emissions occurring due to fuel consumption.

- Incorporating the costs and CO_2 emissions happening due to the refrigeration activities into the CFSC problem studied by [SBRHVDV15].
- Employing the (R, S) ordering policy which is responsive to non- stationary demand uncertainties, in order to avoid the inflexibility of the pre-determined ordering decisions of the static uncertainty while avoiding possible costs and computational requirements arising due to the dynamic uncertainty.
- Solving the excess cumulative ending inventory level issue resulted in [SBRHVDV15]'s (R, Q) model and caused by the pre-determined order quantities.
- Developing a framework to prove the optimality of the (s, S) policies in inventory systems with service level constraints.

We have published the work we conducted in this dissertation in international journals. The list of the published, submitted, and ongoing papers are listed below:

- Suheyl Gulecyuz and Barry O'Sullivan and S. Armagan Tarim: *A Heuristic Method for Perishable Inventory Management Under Non-Stationary Demand*, in Omega.
- Suheyl Gulecyuz and Barry O'Sullivan and S. Armagan Tarim: *A Heuristic Method for Perishable Inventory Management Under Non-Stationary Demand and Service Level Constraints*, in progress.
- Soodeh Jahdi and Suheyl Gulecyuz and Seamus O'Reilly and Barry O'Sullivan and S. Armagan Tarim: *An IRP Model to Improve the Sustainability of Cold Food Supply Chains under Stochastic Demand*, in the Journal of Cleaner Production.
- Suheyl Gulecyuz and Barry O'Sullivan and S. Armagan Tarim: *Optimality of (s, S) Policies in Inventory Control under Service Level Constraints*, in progress.

1.4 Thesis Structure

The remainder of this dissertation is organised as follows:

- In Chapter 2, we provide a comprehensive analysis and discussion on inventory control, perishability, cold food supply chains, and solution techniques, providing a foundation for the research questions addressed in this study and reviewing the relevant literature.
- In Chapter 3, we present the IRP model works under the (R, S) policy for CFSCs, considering transportation, inventory, and refrigeration costs and highlighting its contributions to sustainability through reduced CO_2 emissions and improved routing and inventory management. We then compare the performance of model to the existing (R, Q) model in the literature, [SBRHVDV15], pointing out that allowing flexibility in the ordering decisions helps to reduce excess inventory caused by pre-determined order quantities.
- In Chapter 4, we introduce the proposed heuristic to determine the policy parameters in perishable inventory systems under stochastic demand and penalty cost, which pre-determines replenishment timings and calculates order quantities based on observed demand information. We then provide extensive numerical studies to compare its performance to the optimal SDP solution.
- In Chapter 5, we extend the scope of the heuristic to include minimal α-service level constraints. We conduct extensive numerical experiments to compare the performance of our proposed heuristic with the near-optimal SDP formulation since there is no optimal SDP formulation to our knowledge in the literature.
- Chapter 6, summarises and concludes the findings or the research we conduct in this dissertation, then suggests future research ideas and directions.

Chapter 2

Literature Review and Background

In this chapter, our goal is to describe all the work relevant to our study. In Section 2.1, discuss and analyse the fundamental concepts of the inventory control problem, lot-sizing, perishability in inventory control, and cold food supply chains (SFCS), in an extensive way. In Section 2.2, we discuss the solution techniques and frameworks relevant to our work in this dissertation in detail, which are dynamic programming (DP), mixed integer programming (MIP), and the inventory routing problem (IRP).

2.1 Inventory Control

We provide a comprehensive definition and analysis for the inventory control concepts in Section 2.1.1. In Section 2.1.2, we discuss the lot-sizing problem, the key heuristics and algorithms in the literature, the strategies developed to handle uncertainty, and the ordering policies. We then examine perishability in inventory control in Section 2.1.3, analysing the key terms, perishable goods and inventory models, classifications, and issuing policies, as well as the perishability literature. Finally, we discuss cold food supply chain terms and literature in Section 2.1.4.

2.1.1 Inventory Control Concepts

Effective inventory management is crucial for optimising supply chain operations. Although recent studies provide advanced methods for inventory control in complex and uncertain environments, their foundations trace back to classical models that established the theoretical foundations of inventory theory, such

as Arrow et al. [AHM51], [WW58], [SAKS60], and Clark and Scarf [CS60]. They introduced key concepts such as the (s, S) policies, dynamic lot-sizing, and multi-echelon systems. Nahmias [Nah75], Fries [Fri75], and Nahmias [Nah82] established the theoretical basis and introduced key concepts for perishable inventory management.

This section introduces the main principles and practices of inventory control. We discuss key concepts such as fundamental terms and definitions, measures of effectiveness, and inventory configuration in to establish a framework for our study. We also discuss demand patterns, various inventory cost factors, replenishment lead time, and inventory review. Moreover, we analyse stockouts and the different approaches to stockouts, examine the concepts related to stock levels, and discuss methods to deal with stockouts.

Fundamental Terms and Definitions

In this section, we provide the fundamental terms of inventory control based on the definitions in [Ros21].

Supply chain: A system of organisations, people, activities, information, resources, technology, etc., involved in moving a product or service from supplier to customer.

Supply chain management: Management of the supply chain flow.

Inventory: A current asset on the balance sheet for all the goods, materials, etc., which can be sold and turned into cash.

Inventory review: The process to determine inventory quantity on hand.

Inventory system: The process to keep track of the flows of storage, supplies, sales, etc.

Warehouse: A stock keeping unit for items in which they are stored and kept until being sold to customer.

Inventory state: The state that is denoted by the current inventory quantity and their status (e.g. expiration).

Items: The good, asset, material units subject to be stored in a warehouse and sold to meet customer demand.

The definitions above are commonly used in the general inventory management literature. The concepts introduced above might have different implications on

cold food supply chains and perishable inventory systems. In cold food supply chains, additional factors such as refrigeration, transportation activities, and environmental considerations influence how inventory is managed. For perishable inventory systems, the finite lifetime of items requires careful timing in ordering, usage, and distribution to minimise waste or shortage. More detailed aspects specific to perishability and CFSCs are provided in Section 2.1.3 and Section 2.1.4, respectively.

Measures of Effectiveness

According to [SPT16], analysts in the production/inventory area have mostly focused on a single measure of effectiveness, the expected cost. Some other measures such as limited space, desired customer service, and nervousness have mostly been defined and modelled as constraints. In actual fact, the impact of decisions is not restricted to a single measure of effectiveness. Some of the following objectives, listed by [SPT16], are very challenging to quantify:

1. Minimising internal political conflicts within the organization.
2. Preserving flexibility to adapt to uncertain future scenarios.
3. Enhancing the probability of the firm's survival or keeping managerial positions safe.
4. Maintaining an acceptable level of human effort in decision-making and operational planning.

Despite expected cost being the most commonly used metric in the literature, several alternative objectives have been considered. [ABTM08] considered objectives such as minimising current investment costs, reducing total cost variability, and lowering financial risk or the probability of exceeding a set budget. In this dissertation, we focus on minimising the expected total cost as the measure of effectiveness, in line with the most of the literature.

In this dissertation, the metrics we evaluate are to minimise the expected cost and to preserve flexibility to handle uncertain future demand realisation scenarios.

Inventory Configuration

Inventory configuration plays an important role in the efficiency, cost effectiveness, and service levels of a supply chain model. There are two main configura-

tions to manage stock across different stages of a supply chain: Single-echelon and multi-echelon. A single-echelon system focuses on inventory management and optimising stock levels individually and independently. For instance, the single-echelon system can be used to model a retail store in a retail environment. Orders arrive at the retailer store and distributed to customers from the store. The relationship between specific suppliers and retailers are not modeled in single-echelon systems. However, some single-echelon inventory models might consider multiple warehouses for various reasons such as capacity restrictions or different products. In contrast, multi-echelon systems integrate inventory control across multiple stages and different nodes, in order to ensure coordination and efficiency throughout the supply chain network and determine the correct inventory levels in different supply chain nodes. An inventory system consisting of a distribution centre and the retailers it supplies as an example of a two-echelon problem, [SR91, SRK95]. In this problem, distribution centres and retailers represent one echelon of the two-echelon supply chain model. The relationships between each echelon are also taken into account when modeling the problem. Orders arrive at distribution centres and distributed to retailers, then customer demand is met from retailers.

In this dissertation, we consider a single-item and single-echelon inventory setting. In the perishability study, we consider a single location model, and in the cold supply chain management study we work on a model consisting of a supplier and multiple warehouses in different locations.

Demand Pattern

According to [SPT16], various types of demand patterns in inventory systems might be considered over a finite or infinite time interval which is called the planning horizon. Different control policies and procedures are appropriate for different types items. The nature of the item is also an important factor to influence the demand pattern. In terms of randomness, customer demand can be classified as deterministic or stochastic. The quantity of deterministic demand is known while stochastic demand takes a random value from a probability distribution. Customer demand can also be categorised by the fluctuations across time (seasonality trends, etc.), such as stationary and non-stationary. In case of stationary demand, the demand pattern do not show variability over time. In non-stationary demand, the demand pattern and distributions subject to vary as the time passes. Moreover, demand distributions and patterns

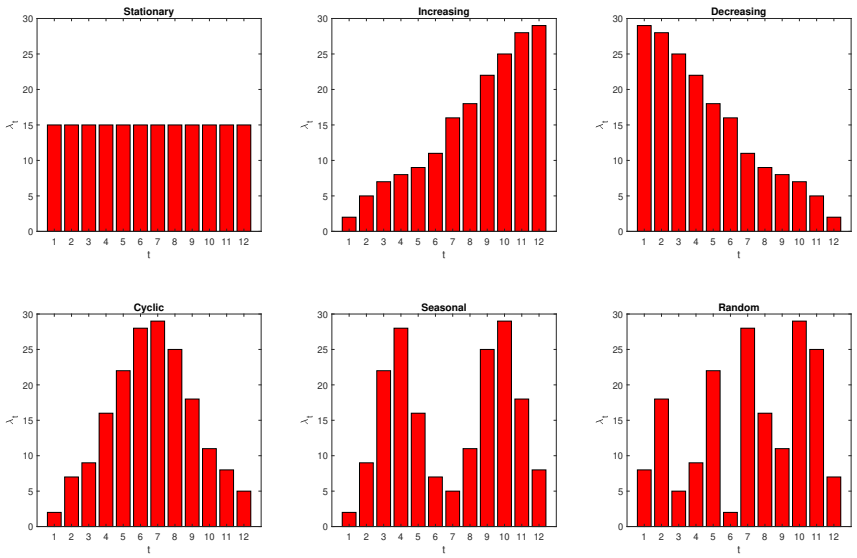

Figure 2.1: Various examples of demand mean patterns.

can be either known or unknown to the decision maker. Figure 2.1 illustrate various stochastic demand mean patterns over a finite planning horizon of 12 periods (e.g., 12 months of a year): stationary, increasing, decreasing, cyclic, seasonal, and random. For instance, the cyclic demand pattern might represent the non-stationary customer demand for summer clothing trending in the summer months and resulting in higher demand means.

In this dissertation we consider a known, non-stationary, and stochastic demand pattern over a finite planning horizon. In the following, we explain non-stationary stochastic known demand in more detail.

Non-Stationary Stochastic Known Demand: This is the demand setting in our study over a finite number of discrete time periods. Based on [Ros21], demand in one period is random and follows a probability distribution in non-stationary demand. Across periods, demand distributions may change. Cyclic and seasonal demand patterns or changes in demand distribution based on the customer trends are several examples of non-stationary demand patterns in practice and real life.

Stochastic demand helps address the uncertainties commonly encountered in real-world applications. It is modeled using stochastic variables that represent demand for each time period. This approach generalises the deterministic case, as stochastic solutions can also handle deterministic problems. However, dealing with non-deterministic demand significantly increases the complexity

of finding an optimal policy. The situation in which the inventory is inadequate to satisfy customer demand is called stockout. Under stochastic demand, stockouts may happen when the demand is higher than expected, while low demand can lead to excess inventory. Ensuring a minimum customer service level becomes particularly challenging in such scenarios, often referred to as non-stationary stochastic or time-varying probabilistic demand. Here, the policy's cost is determined by its expected cost over the planning horizon.

In stochastic demand scenarios, the probability distribution of demand may be either known or unknown. When the distribution is known, it enables more accurate modeling and optimization, allowing planners to create effective policies. When the distribution is unknown, the problem becomes more challenging and often requires adaptive or data-driven methods to estimate and respond to demand patterns over time. Managing demand with unknown distributions typically involves techniques like machine learning, Bayesian updating, or heuristics to keep the system efficient and reliable despite the uncertainty.

Inventory Cost Factors

Inventory costs are a key part of managing a supply chain and can be grouped into several main categories, as described by [SPT16] and [Ros21]:

Inventory review cost: The cost of physically checking inventory to determine stock levels and condition.

Fixed setup cost: A fixed cost incurred each time an order is placed, regardless of the order quantity.

Purchase or ordering cost per unit: A variable cost charged for each unit purchased or ordered during replenishment.

Inventory holding cost per unit: The cost of carrying one unit of inventory from one period to the next. This includes expenses such as storage, insurance, and depreciation.

Fixed penalty cost: A one-time cost charged whenever inventory levels fall below zero. This cost is independent of how many units are short and reflects the impact of stockouts, whether the demand is lost or backordered.

Penalty cost per unit: A cost applied to each unit of demand that is either backordered or lost. It is charged only once per unit and does not depend on how long it takes to fulfill a backorder.

Penalty cost per unit per period: A recurring cost charged for each unit of backordered demand, applied for every period the shortage continues.

System control costs: These are the costs related to running the chosen decision system. They include expenses for gathering and storing data, maintaining the system, and performing calculations.

Unit outdating cost: These are the costs specific to perishable inventory systems. This cost is incurred due to expenses such as waste disposal, recycling activities, environmental concerns due to waste, etc.

Understanding these costs helps to develop effective and efficient inventory policies and reducing overall supply chain expenses. Each cost points out a specific aspect of inventory management, helping businesses balance service levels, storage capacity, and ordering efficiency. In our perishable inventory study, we consider fixed setup cost and variable ordering, holding, penalty (no penalty cost in the α-service level case), and outdating cost per item. Additionally, we consider refrigeration, fuel, transportation costs in the cold supply chain IRP model.

Replenishment Lead Time

According to [Sil81] replenishment lead time is the amount of time elapses from the moment at which it is decided to place an order until it is physically on the shelf ready to satisfy customers' demands. It involves all the time spent by the activities from the transshipment decision by the supplier until the item is ready on the shelf to meet customer demand. In this dissertation, we assume that the replenishment lead time is zero.

Inventory Review

Inventory is reviewed in certain times, and based on these reviews inventory decisions are made. A review interval is the period of time between two consecutive inventory reviews. In the literature, two types of inventory reviews have been studied, which are explained based on [SPT16] and [Ros21] as follows:

Periodic review: In periodic review inventory systems, inventory reviews take place in fixed time intervals (e.g., day, month, year). The type of items, demand distributions and fluctuations, etc, might be crucial to determine the length of

review periods. In most cases, periodic review has lower costs than continuous review.

Continuous review: On the contrary to periodic review, inventory is reviewed continuously and at any time in continuous review inventory systems. The inventory status is continuously updated and always known. In practical life, continuous review has higher costs compared to periodic review. This is one of the main disadvantages of continuous review. One of the advantages of continuous review is that it generally requires less safety stocks.

Based on their importance and priority in inventory decisions and from the upper end of the spectrum to the lower end, items are classified as A-type, B-type, or C-type items, respectively, [SPT16]. Continuous review is more suitable for A-type most critical items which require constant monitoring and attention. Periodic review is more suitable for B-type and C-type items which have less effect in inventory decisions.

In this dissertation, we assume the inventory reviews take place for discrete time periods and a planning horizon with a finite number of periods.

Stock Levels

Based on [SPT16], we hereby introduce the definitions of several terms related to stock levels:

On-hand inventory: It is the physical amount of inventory which is on the shelf and ready to meet customer demand. On-hand inventory cannot be negative since it is the physical quantity of available inventory.

Net inventory (net stock): It is the balance of the on-hand inventory and backorders. It can be negative if there are backordered items.

$$\text{Net inventory} = \text{On-hand inventory} - \text{Backorders}$$

Inventory position: It consists of the net inventory plus the ordered items that are on the way (on order) minus the committed demand realisation. If there is zero lead time, the amount of items on order is zero since instant order placement is assumed.

$$\text{Inventory position} = \text{On-hand inventory} + \text{On order} - \text{Backorders}$$

Safety stock: Safety stock is the average level of the net stock just before a replenishment arrives.

Stockouts

Inventory levels may often drop to a level that the system can no longer meet customer demand. For this situation, two main approaches has been adopted in the literature, which are lost sales and backlogging, [SPT16, Ros21]. Most studies prefer only one of the models, however there are some studies such as [GCB02, VPRT21, VPRT23] which consider a combination of both backlogging and lost sales.

Backlogging: Also called backordering, the amount of customer demand that could not met in the current period is carried on the following periods and met later. Most studies in the inventory control literature assume full backlogging, that is backlogging all the unmet demand. In order to deal with backlogging situations, penalty costs or service level constraints are commonly assigned and applied, [SPT16, Ros21]. Full backlogging is commonly studied in classical inventory control, [SPT16, Zip00, SAKS60], and lot-sizing literature, [RTHP08, DSKTR16, Tar96].

Lost Sales: In the lost sales approach, all the unmet demand is lost and will not be met in the future. Lost sales models have been studied in the literature in different settings, [BV12b, Zip08, HJMR09]. Bijvank and Vis [BV12a] provide a complete literature review on such systems.

In this dissertation, we assume full backlogging in all parts of the work.

Dealing with Stockouts

There are several approaches in the literature to deal with and minimise the risk of stockouts. Most studies in the inventory literature assign a penalty cost, followed by applying a service constraint to ensure that the probability of stockouts must be above a certain threshold depending on the service level type. These are the two main methods to avoid stockouts.

Penalty Cost: A cost is assigned to the inventory system per stockout. It is dependent on the amount and/or the duration of stockouts, e.g., penalty cost per item or per time unit.

Service Level: There are three main types of service level constraints.

1. α***-service level (Type 1):*** This is the event-oriented service level. It is imposed to guarantee that the probability of non-stockout is α, such that

$$\mathbb{P}\{I_t \geq 0\} \geq \alpha$$

where I_t denotes the ending inventory level in Period t, e.g. 90% α-service level means that the probability of non-stockout in each period is greater than or equal to 0.9.

2. β***-service level (Type 2):*** This is the quantity-oriented service level. The probability that the number of backordered items over the total number must be less than $1-\beta$, such as

$$\beta = 1 - \frac{\text{expected number of backorders per period}}{\text{expected demand per period}}.$$

For instance, 90% β-service level means that if the expected demand per period is 100, the number of backordered items should be no more than 10.

3. γ***-service level (Type 3):*** This is the time- and quantity-related service level criterion. This is rarely applied compared to α and β service levels.

$$\gamma = 1 - \frac{\text{expected backorder level per period}}{\text{expected demand per period}}.$$

Moreover, service level constraints can be implemented as the mean or minimum service levels in periodic review inventory systems. In the minimum service level approach, in all the periods the inventory system must meet the minimum service level criteria. In the mean service level approach, the average service level per period throughout the planning horizon must meet the required service level threshold. The average service level througout the planning horizon can be approximated as

$$\frac{p}{p-h},$$

where h and p are the inventory holding and penalty costs per item, see [Vis20].

In this dissertation, we consider the penalty cost approach and the α type of service level separately. However, the work in this dissertation can be extended to the other service level types as a future work.

2.1.2 Lot-Sizing Problem

In this section we provide an overview of key algorithms, uncertainty strategies, and ordering policies associated with lot-sizing and specialised with stochastic lot-sizing. We first provide a lot-sizing literature survey. We introduce the key methods in the lot-sizing problem, that are the Wagner-Whitin algorithm and the Silver-Meal heuristic, also gave an inspiration to develop our proposed heuristic. Then we analyse the uncertainty strategies introduced by [BT88] . Finally, we examine the ordering policies relevant to the work in this dissertation.

The study of the dynamic lot-sizing problem began with the work of [WW58], who introduced the Wagner-Whitin algorithm. This algorithm uses a dynamic programming approach to calculate the optimal cost for ordering and holding inventory over time. Since then, dynamic lot-sizing has been a well-studied topic. Most of the early work focuses on the deterministic demand case. For instance, the Silver-Meal heuristic, developed by [Sil73], determines the production or order quantities to minimise the average cost per period. [DBGVW84, PVW92, KSVW94, DK97] provide comprehensive early lot-sizing literature reviews.

As we mentioned earlier in this dissertation, we study non-stationary stochastic demand cases. Therefore, we mainly focus on the stochastic lot-sizing problem. The newsvendor problem in Edgeworth [Edg88] is the simplest example of the single-item stochastic lot-sizing problem, e.g., calculating the optimal order/production quantity of a single item over a single period planning horizon. However, studies considering stochastic demand became widely studied much later. For instance, Silver [Sil78] and Askin [Ask81] proposed heuristic procedures for stochastic lot-sizing problems. Moreover, [Ask81] formulated a method for calculating the critical fractile in multi-period newsvendor problems, in order to provide a way to balance ordering decisions under uncertainty.

In the literature, various strategies have been proposed to deal with the uncertainty. The static-dynamic strategy works in two steps: first, the ordering periods are determined before the start of the planning horizon (static uncertainty stage), and then the order quantities are calculated dynamically as demand is realised over time (dynamic uncertainty stage). Bookbinder and Tan [BT88] introduced the static, dynamic, and static-dynamic uncertainty strategies for single-item, single-location, finite-horizon stochastic lot-sizing problems. They consider fixed setup costs, unit ordering costs, holding costs, and service level

constraints. However, the unit ordering cost is not explicitly considered in these strategies.

Tarim and Kingsman [TK04] re-formulated the static-dynamic strategy introduced by [BT88] as a mixed integer linear programming (MILP) model and using the (R, S) ordering policy. This re-formulation allows to determine both optimal ordering periods and order quantities. Their model also incorporates unit ordering costs and simplifies these costs into a function of the order-up-to level for the last period oft the planning horizon. This advancement made the strategy more practical and comprehensive.

Similarly, [Var09] examined non-stationary stochastic inventory systems and modeled them as shortest-path network problems. They extended the Wagner-Whitin algorithm to handle stochastic demand, adapting it to identify the shortest path while incorporating uncertainty into the ordering decisions. Both approaches highlight how advancements in modeling and algorithm design have improved the practical applicability and computational efficiency of lot-sizing solutions.

Some studies extended [TK04]'s approach to deal with additional complexities, such as varying service levels, non-zero lead times, and capacity constraints (e.g., [RTHP10, RTHP11, RTB12, TKTE11]). For instance, [RTHP11] developed a deterministic-equivalent DP method for the (R, S) policy under non-stationary stochastic demand. They reformulated the problem as a shortest-path model to reduce the computational complexity by minimising the state space. Their approach included an algorithm to filter out unnecessary paths and adjust the graph to handle infeasible cases such as negative order quantities.

Single-Period Newsvendor Problem: The origins of the newsvendor problem traces back to Edgeworth [Edg88]. This is the simplest stochastic inventory problem in the literature. In the newsvendor problem, the aim is to find the optimal order quantity in a single-period inventory model with a stochastic demand following a known cumulative distribution function F. Order must be placed before demand is realised and at the beginning of the period, and there are unit purchasing cost c, selling price r, and salvage value s. Based on these parameters, the overage cost o and the underage cost u are defined as $c - s$ and $r - c$, respectively. Then the optimal order quantity is

$$Q^* = F^{-1}\Big(\frac{u}{o+u}\Big).$$

.

Multi-Period Newsvendor Problem: The multi-period newsvendor problem is a generalisation of the single-period classical newsvendor problem for a T-period inventory model, $T > 1$. The aim is to find the optimal order quantity that minimises the total cost over T periods. Using the definitions and the formulations in the single-period newsvendor problem, the optimal order quantity is calculated by

$$\sum_{t=1}^{T} F_{1,\dots,t} = \frac{Tu}{o+u},$$

where $F_{1,\dots,t}$ is the CDF of the total demand $d_1 + d_2 + ... + d_T$ over T periods.

The multi-period newsvendor problem forms a basis to develop a procedure to find the optimal order quantities in the penalty cost version of our proposed heuristic for perishable inventory systems, discussed in detail in Chapter 4. In this chapter, we provide analytical expressions for the multi-period newsvendor problem for the perishable inventory case.

Key Lot-Sizing Algorithms and Heuristics

We hereby describe two of the most well-known algorithms and heuristics in the lot-sizing literature: The Wagner-Whitin algorithm and the Silver-Meal heuristic.

Wagner-Whitin Algorithm: The Wagner-Whitin algorithm, also known as the *dynamic lot-size model* and introduced by [WW58], is a DP-based approach designed for deterministic lot-sizing problems. Its adaptability to stochastic settings makes it an important foundational method. The algorithm works by iteratively calculating the optimal solutions for increasingly larger planning horizons. The Wagner-Whitin algorithm determines the optimal production schedule that minimises total cost which consist of fixed setup and holding costs, while meeting customer demand over a finite planning horizon. Despite its computational efficiency for smaller problems, it might become less practical for larger, stochastic instances due to the exponentially growing complexity. The dynamic lot-sizing problem addressed by [WW58] assumes the following:

- *Finite planning horizon:* The algorithm operates over a fixed and finite number of periods as discrete time intervals.
- *Known demand:* The customer demand in each period is assumed to be deterministic and known.

- *No backorders:* All demand must be met and no stockouts are allowed.
- *Cost structure:* There is a fixed setup cost per order or production, and a linear inventory holding cost per item.

The functional equation that follows the same notation as in [WW58] that represents the minimal cost policy is given in the following:

$$f_t(I) = \min_{x_t \geq 0, I_t + x_t \geq d_t} [i_{t-1} I_t + \mathbb{1}(x_t) s_t + f_{t+1}(I + x_t - d_t)]$$

where s_t is the fixed setup cost, i_t is the unit holding cost, x_t is the order quantity, the beginning inventory

$$I_t = I_0 + \sum_{j=1}^{t-1} x_j - \sum_{j=1}^{t-1} d_j \geq 0,$$

and

$$\mathbb{1}(x_t) = \begin{cases} 1, & \text{if } x_t > 0 \\ 0, & \text{if } x_t = 0. \end{cases}$$

Let us assume a 2-period planning horizon. At the end of Period 1, there is only one decision: placing an order at the beginning of Period 1 that covers 1-period demand. At the end of Period 2, there are two possible decisions: placing an order at the beginning of Period 1 that covers Periods 1 and 2, or placing separate orders that cover one period at the beginning of each period. At the end of each period, the previous decision schedule is updated recursively, in order to minimise the total cost.

Algorithm 1 describes the pseudocode of the Wagner-Whitin algorithm. It evaluates all the possible scenarios and calculates the optimal schedule up to Period t, then updates the optimal schedule at the end of each recursion, in order to find the optimal schedule and order/production quantities that minimise the total cost for the entire planning horizon.

Silver-Meal Heuristic: The Silver-Meal heuristic, introduced by [Sil73], offers a simple, straightforward, and computationally efficient approach to the lot-sizing problem by focusing on minimising the average cost per period, instead of finding an optimal solution. While it was initially developed for deterministic cases, it can be extended to stochastic cases. The heuristic proposed by [Sil73] assumes the same conditions as the Wagner-Whitin algorithm:

Algorithm 1 The Wagner-Whitin Pseudocode

```
1: C[0, ..., T] ← 0
2: P[1, ..., T] ← 0
3: for t = 1 to T do
4:     C[t] ← ∞
5:     for j = 1 to t do
6:         setup ← a_t
7:         holding ← 0
8:         for k = j to t do
9:             holding ← holding + h[k] Σ_{i=j}^{k} d_i
10:            TC = C[j − 1] + setup + holding
11:            if TC < C[t] then
12:                C[t] = TC
13:                P[t] = j
```

In an T period planning horizon, the average cost is calculated as follows. Let

$$AC(t) = \frac{a + hQ_2 + 2hQ_3 + ... + (t-1)Q_t}{t}$$

be the average cost per period function that spans $t \leq T$ periods, a and h are the fixed setup and unit holding costs, respectively. Moreover, Q_t is the required order/production quantity to satisfy the demand in Period t.

- $t = 1$:
$$AC(1) = a$$
- $t = 2$:
$$AC(2) = \frac{a + hQ_2}{2}$$
- $t = 3$:
$$AC(3) = \frac{a + hQ_2 + 2hQ_3}{3}$$
- $t = i$:
$$AC(i) = \frac{a + hQ_2 + 2hQ_3 + ... + (i-1)Q_i}{i}$$
- $t = T$:
$$AC(T) = \frac{a + hQ_2 + 2hQ_3 + ... + (T-1)Q_T}{T}$$

Based on the t value with the minimum $AC(.)$, the heuristic determines the number of periods that are covered by the next order/production batch.

Let us assume a 5-period planning horizon and $AC(2) > AC(3) > AC(1)$.

Then we place an order that covers Periods 1 and 2, $Q_1 + Q_2$. Next, we start the heuristic from Period 3 and select an order quantity that cover Period 3, 3 and 4, or between 3-5. Assume the average cost between Periods 3 and 5 is minimum. So, we order $Q_3 + Q_4 + Q_5$ items.

Algorithm 2 describes the pseudocode of the Silver-Meal heuristic. It evaluates all the possible replenishment cycles, calculates the average cost per Period t, and determines the next cycle based on the minimum average cost per period. This procedure is repeated until the end of the planning horizon.

Algorithm 2 The Silver-Meal Pseudocode

Input: Demand d_t for $t = 1, 2, \ldots, T$, setup cost a, unit holding cost h
Output: Lot sizes for each period
$t \leftarrow 1$
while $t \leq T$ **do**
    $k \leftarrow 1$
    $TotalCost \leftarrow 0$
    $AvgCost \leftarrow 0$
    $CurrentLot \leftarrow d_t$
    while $AvgCost \leq \dfrac{S + h \cdot \sum_{j=1}^{k}(j-1) \cdot d_{t+j-1}}{k}$ **and** $t + k - 1 \leq T$ **do**
        $CurrentLot \leftarrow CurrentLot + d_{t+k}$
        $TotalCost \leftarrow S + h \cdot \sum_{j=1}^{k}(j-1) \cdot d_{t+j-1}$
        $AvgCost \leftarrow \dfrac{TotalCost}{k}$
        $k \leftarrow k + 1$
    $t \leftarrow t + k - 1$

Stochastic Lot-Sizing Problem

We hereby provide the formulation of the lot-sizing problem below: It can be either formulated by assuming there is a penalty cost or there is a service level requirement. We consider a minimal α-service level requirement in the formulation below:

$$\min TC = \sum_{t=1}^{T} \Big(\mathbb{1}(Q_t) + h \max(0, I_t) \Big) \tag{2.1}$$

subject to

$$\mathbb{1}(Q_t) = \begin{cases} a + cQ_t, & \text{if } Q_t > 0 \\ 0, & \text{otherwise} \end{cases} \qquad t = 1, 2, ..., T. \tag{2.2}$$

$$I_t = I_0 + \sum_{i=1}^{t}(Q_t - d_t), \quad t = 1, 2, ..., T. \tag{2.3}$$

$$Q_t, I_t, d_t \geq 0, \quad t = 1, 2, ..., T. \tag{2.4}$$

$$\geq 0, \quad t = 1, 2, ..., T. \tag{2.5}$$

$$\geq 0, \quad t = 1, 2, ..., T. \tag{2.6}$$

where Q_t is the order quantity and I_t is the ending inventory level, , respectively, h is the unit holding cost, a is the fixed setup cost, and c is the unit ordering cost. d_t is the demand realisation in Period t with known density function $f_t(d_t)$ and the density functions may vary across periods as a result of the non-stationary demand pattern. Equation 2.1 is the objective function. In Equation 2.2, the ordering cost function is given. Equation 2.3 gives the inventory balance equation which denotes the ending inventory levels for each Period t with respect to the order quantities and customer demand. 2.4, 2.5, and 2.6 are the non-negativity constraints of the order quantity, ending inventory, and customer demand for each Period t, respectively.

In the stochastic lot-sizing problem, the objective function becomes

$$\min \mathbb{E}\left\{TC\right\} = \int_{d_T} ... \int_{d_1} \sum_{t=1}^{T} \Big(\mathbb{1}(Q_t) + hI_t \Big) f_1(d_1)...f_T(d_T) \mathrm{d}d_1 ... \mathrm{d}d_t \tag{2.7}$$

For the penalty cost case of the deterministic and the stochastic models, we add a penalty cost component $p\text{max}(0, -I_t)$ to the objective functions 2.1 and 2.7, respectively, and relax 2.3. For the α-service level case of the stochastic lot-sizing problem, we add the α-service level constraint given below:

$$\mathbb{P}\left\{I_t \geq 0\right\} \geq \alpha, \quad t = 1, 2, ..., T. \tag{2.8}$$

The deterministic-equivalent version of the stochastic lot-sizing model is given in the following:

$$\min \mathbb{E}(TC) = \int_{d_T} ... \int_{d_1} \sum_{t=1}^{T} \Big(a\delta(t) + cQ_t + h\mathbb{E}(I_t) \Big) f_1(d_1)...f_T(d_T) \mathrm{d}d_1 ... \mathrm{d}d_t \tag{2.9}$$

subject to

$$\delta(t) = \begin{cases} 1a + cQ_t, & \text{if } Q_t > 0 \\ 0, & \text{otherwise} \end{cases} \quad t = 1, 2, ..., T. \tag{2.10}$$

$$\mathbb{E}(I_t) = I_0 + \sum_{i=1}^{t} Q_t - \sum_{i=1}^{t} \mathbb{E}(d_t), \quad t = 1, 2, ..., T. \tag{2.11}$$

$$Q_t \geq F^{-1}_{D(t)}(\alpha) - I_0, \quad t = 1, 2, ..., T. \tag{2.12}$$

$$\sum_{i=1}^{t} Q_t, \mathbb{E}(I_t) \geq 0, \quad t = 1, 2, ..., T. \tag{2.13}$$

where $F^{-1}_{d_t}$ is the inverse cumulative distribution function of demand in Period d_t and $F^{-1}_{D(t)} = F^{-1}_{d_1+d_2+...+d_t}(.)$ is the inverse cumulative probability distribution function of $d_1 + d_2 + ... + d_t$. Note that Equation 2.12 is binding for $t = T$ at optimality. Therefore, the unit holding cost component of the objective function 2.9 is constant and c has no effect on calculating the optimal parameters.

Bookbinder and Tan's Uncertainty Strategies

In inventory control, dealing with uncertainty is a challenging issue. In order to manage uncertainty situations, Bookbinder and Tan [BT88] introduced a framework for dealing with uncertainty in lot-sizing in three strategies: The static, static-dynamic, and dynamic uncertainty. These strategies guide decision making for the timing and flexibility of information updates (e.g., order placement) during the planning process and horizon.

Static Uncertainty: The static uncertainty strategy involves determining both the order quantities and the timing of orders at the beginning of the planning horizon. This approach assumes that initial forecasts is sufficient to represent future demand uncertainties. It lacks flexibility and may produce results far from optimality in case of high deviations from the expectations. The static uncertainty strategy is often used when the variance is low and computational simplicity is prioritised, such as in cases when considerable preparation time is needed to place an order. In 2025, determining that 100 items will be produced in January, April, and August 2026 is an example for the static uncertainty.

Dynamic Uncertainty: The dynamic uncertainty strategy allows to make changes and modifications on both the timing and the quantity of orders based on updated information. For instance, the dynamic uncertainty strategy has advantages in pharmaceutical supply chains since vaccine and drug demand is highly uncertain and might spike suddenly. However, the enhanced flexibility has several drawbacks in terms of cost and computational requirements. The approaches based on the dynamic uncertainty strategy tend to be computationally intensive and require advanced modeling and coding techniques. This strategy fits the most for highly uncertain environments where demand pat-

terns subject to have large variations and the forecast accuracy improves over time. The dynamic uncertainty strategy works as in the following example: Let us assume in January 2026, the annual ordering schedule is determined to be January, April, and August and the quantity will be based on the observations such that they meet a certain order-up-to level. In March 2026, the demand and inventory observations show that the schedule must be updated and an order placement should be made in March 2026.

Static-Dynamic Uncertainty: In contrast to the static uncertainty strategy, the static-dynamic uncertainty strategy determines the timing of order placements before the beginning of the planning horizon, but allows order quantities to be flexible and determined based on new observations and information. It balances the stability of the static uncertainty with the flexibility of the dynamic uncertainty. It is the most effective strategy for cases in which timing constraints are critical, but flexibility is necessary when dealing with demand variability. This approach may reduce costs significantly compared to purely static methods while maintaining manageable computational complexity. Moreover, it lacks costs that arise due to new setups and decisions in the dynamic uncertainty strategy. For instance, the static-dynamic uncertainty strategy is advantageous for coordinating the order quantities of related items in material requirement planning. For instance, it might be necessary to fill a container in order to reduce the unit shipping cost when ordering from overseas. In 2025, determining that items will be produced in January, April, and August 2026, but the production quantities are determined during the production months and based on the observations such that they meet a certain order-up-to level, is an example for the static-dynamic uncertainty.

In this dissertation, we focus on the static-dynamic strategy of [BT88], in order to preserve the computational simplicity and speed while aiming to develop a cost-efficient approach suitable for most real-life cases by making a trade-off from the full flexibility of the dynamic uncertainty.

Ordering Policies for Stochastic Inventory Control

In this section, we discuss the ordering policies for stochastic inventory control relevant to our study, based on [Ros21] and [MRA19].

Base-Stock Policy: We consider the setting in the multi-period newsvendor problem, and relax the single-order assumption. The aim is to control inventory of a single item type over a T-period planning horizon that comprises In the

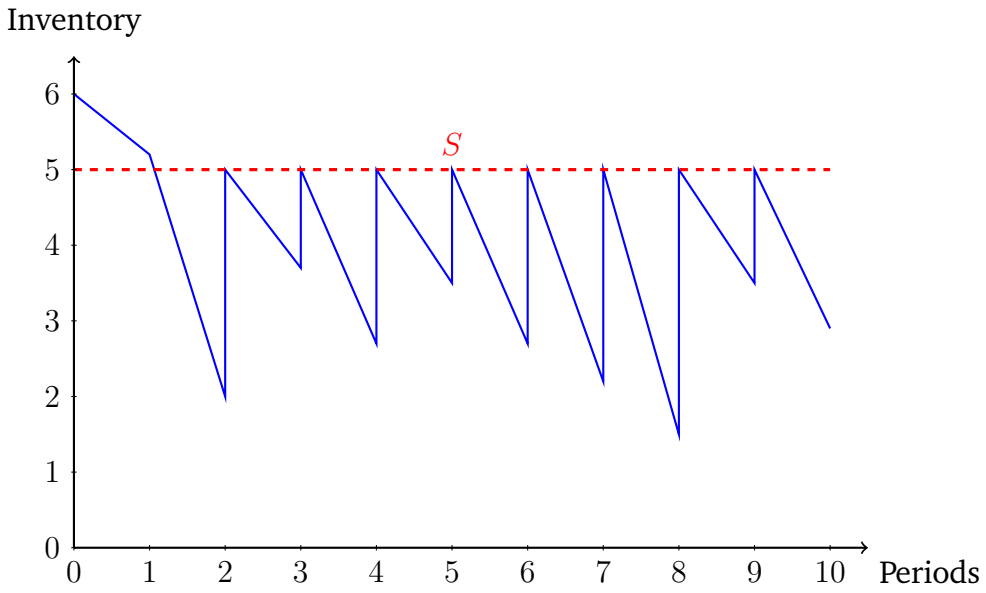

Figure 2.2: The expected inventory level under the base-stock policy

base-stock policy, if the inventory level I_t drops below a certain threshold S at the beginning of Period t, the decision maker places an order $Q = S - I$. When an order placement is made, the inventory level is raised to S. S is called the order-up-to level. The optimal policy determines the S^* value which minimises the total expected cost. The cost structure consists of a unit holding and penalty cost h and p, respectively. The base-stock policy adopts the dynamic uncertainty strategy since the timing and the amount of orders are decided at the beginning of each period, depending on the observed inventory level and the realised demand. In case of non-stationary demand, the order-up-to levels for each Period t, S_t, may vary depending of the demand distribution in Period t. Figure 2.2 illustrates an example for the base-stock policy.

(s, S) **Policy:** Similar to the base-stock policy, the inventory system is reviewed at the beginning of each period. If the inventory level drops below a certain threshold s, which is called the re-order point, the decision maker places an order that satisfies the order-up-to level S, that is $S - I$. The cost structure consists of a unit holding and penalty cost h and p, respectively, and a fixed setup cost K. The optimal parameters s^* and S^* minimise the total expected cost. In case of non-stationary demand, re-order points and order-up-to levels subject to change depending on the demand distribution in each Period t. In these cases, the (s, S) policy is referred to as the (s_t, S_t) policy which aims to find the optimal s_t and S_t values for each period. Scarf [SAKS60] proved the optimality of the (s, S) policy. However, computing the optimal s and S param-

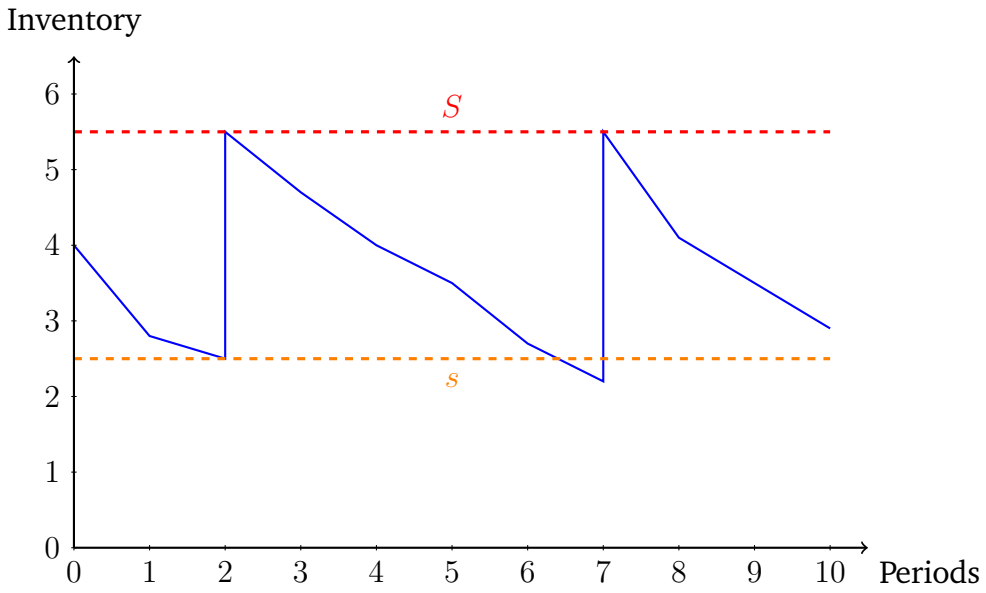

Figure 2.3: The expected inventory level under the (s, S) policy

eters is computationally challenging. In order to overcome these challenges, several studies such as Xiang et al. [XRMBT18] developed a mixed integer linear programming approach to compute the near-optimal s and S parameters. Another disadvantage of this policy is the variability in order quantity. Figure 2.3 which is taken from Visentin [Vis20] illustrates an example for the (s, S) policy .

For the minimum service level case, there is no study to our knowledge that proves the optimality of the (s, S) policy.

(R, Q) **Policy:** The (R, Q) policy aims to find the optimal replenishment Periods R and the order quantities Q before the start of the planning horizon. It works under the static uncertainty strategy of [BT88] and allows no flexibility on the decisions based on the demand observed over time. Therefore, the observed demand has no effect on decision making. In case of non-stationary demand, the R and Q parameters can vary over time. The variation of the (R, Q) policy for this case is referred to as the (R_t, Q_t) policy. For most real life situations, the lack of flexibility makes the (R, Q) policy less cost-efficient. However, for the environments with a low degree of flexibility, e.g. when considerable preparation time is needed to place an order, this strategy might be beneficial and advantageous. It provides a stable production environment and allows better price negotiations with suppliers. Figure 2.4, taken from [Vis20], illustrates an example for the (R, Q) policy.

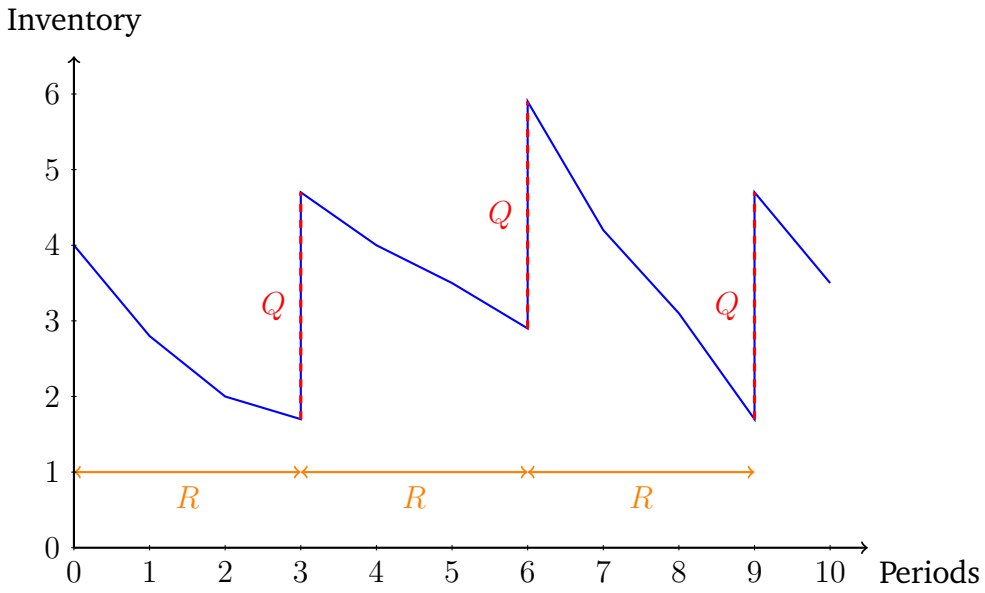

Figure 2.4: The expected inventory level under the (R, Q) policy

(R, S) **Policy:** The (R, S) policy works under the static-dynamic uncertainty strategy and allows some degree of flexibility on the decisions such that replenishment periods are pre-determined and cannot change, however ordering decisions are made based on the demand observations over time. This policy uses the order-up-to-level model. Therefore, in each replenishment Period R, the order quantity is determined to meet the order-up-to level S, such that $Q = S - I$. In case of non-stationary demand, the (R, S) policy is referred to as the (R_t, S_t) policy. The (R, S) policy deals with uncertainty in a better way compared to the static (R, Q) policy, since it enables flexible ordering decisions. Figure 2.5 illustrates an example for the (R, S) policy [Vis20].

Bookbinder and Tan [BT88] proposed the first algorithm to compute (R, S) parameters. Tarim and Kingsman [TK04] developed a mixed integer programming formulation for this problem.

In our study, we employ the (R, S) policy to an IRP model for cold food supply chains with environmental concerns and compare the results with the (R, Q) model developed by Soysal et al. [SBRHVDV15].

(R, s, S) **Policy:** The (R, s, S) policy is the combination of the (R, S) and (s, S) policies. In the (R, s, S) models, inventory reviews and replenishment decisions are made in pre-determined replenishment periods. At the beginning of a replenishment period, if the inventory level is below the re-order point s, an order placement is made to fulfill the order-up-to level S. In case of non-stationary

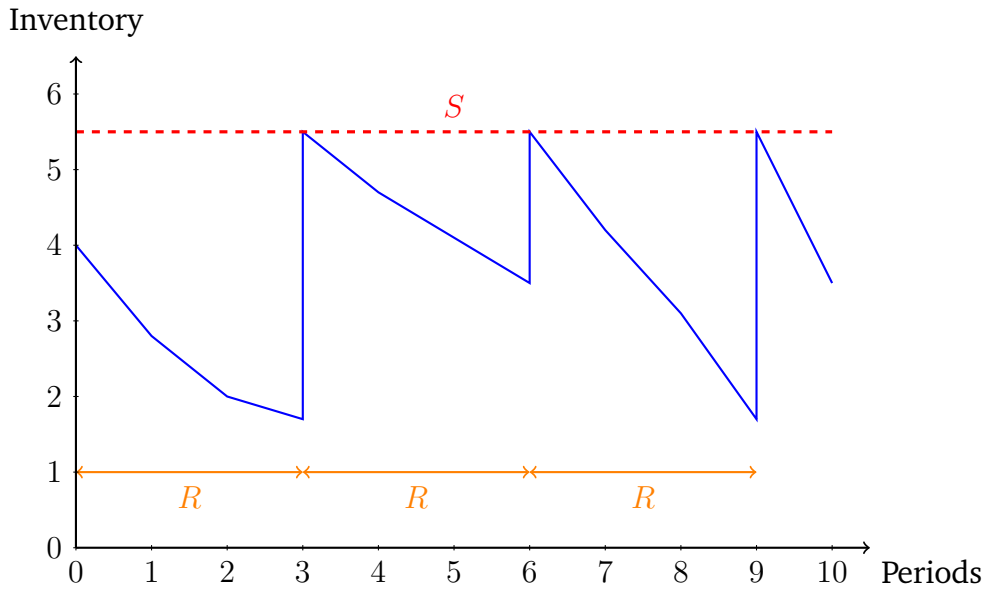

Figure 2.5: The expected inventory level under the (R, S) policy

demand, the (R, s, S) policy is referred to as the (R_t, s_t, S_t) policy.

The (s, S) and (R, S) policies are the special cases for the (R, s, S) policy. If the inventory is reviewed at the beginning of each period, it reduces to the (s, S) policy, particularly in cases when inventory review costs are zero. When $s = S - 1$, the policy reduces to the (R, S) policy, and there is an order in all review periods, see Visentin et al. [VPRT21]. This is particularly valid in cases under the assumption that the inventory levels very rarely exceed S. Computationally, determining the parameters of this policy requires greater effort than the other inventory control policies, see [Vis20].

2.1.3 Perishability in Inventory Control

In contrast to the classic inventory models which assume items stay in the inventory until being subject to customer demand, perishable inventory control deals with the items that could expire. Replenishment policies for perishable inventory systems differ from that of non-perishable inventory models such that they must take the amount of estimated waste, the product lifetime and the number of perishable items of a specific age into account in order to minimise the total cost, waste, etc. Moreover, various issuing policies need to be taken into account to determine which items should be used first, with respect to the age or remaining lifetime.

In the following, we discuss the fundamental perishability terms in an extensive

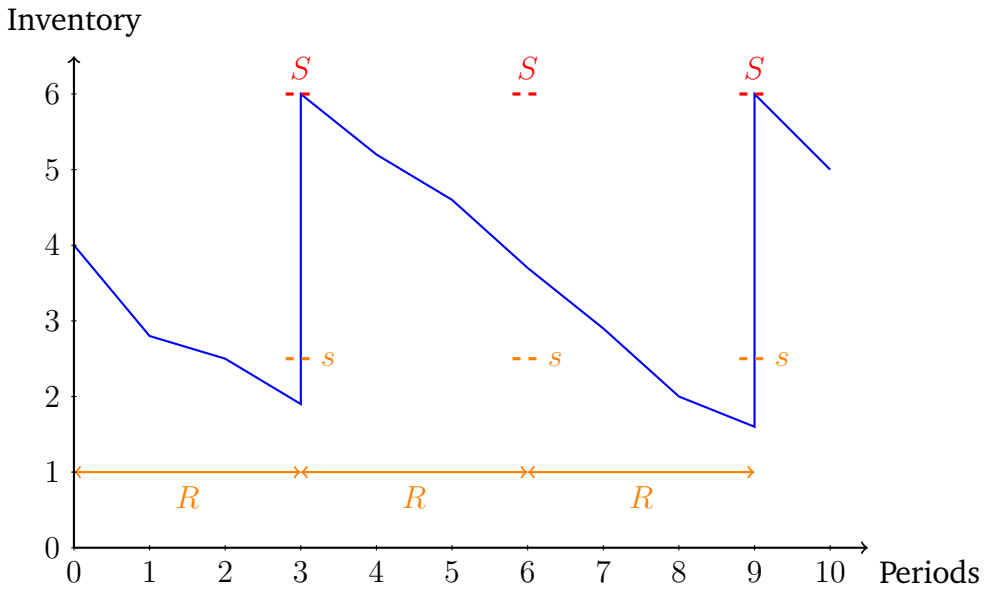

Figure 2.6: The expected inventory level under the (R, s, S) policy

way: We first discuss perishable goods and the classification of perishable goods and issuing policies. Then, we provide a literature review on perishability in inventory control.

Perishable Goods

Perishable goods are items or products subject to perishability. In the literature, the terms deteriorating goods, items, or products are also commonly used. Dairy products, meat products, fruits and vegetables, blood products, medical drugs, etc., are well-known real-life perishable products and items that are widely studied in the literature, see Karaeşmen et al. [KSWD11]. In the following, we introduce several properties related to perishable goods:

Lifetime: The amount of total time that an item is held in the inventory until it becomes expired. Lifetime can be fixed or variable, deterministic or stochastic. In the literature, lifetime is also commonly referred to as shelf life.

Age: The amount of time passed since an item entered the inventory. It is also defined by the remaining lifetime in some studies in the literature. Lifetime is the maximum possible age of an item.

Outdating: If an item stays in the inventory until the end of its lifetime, it becomes unavailable for usage. This process is called outdating, expiration, or perishing.

Waste: Waste is the outdated item which needs to be discarded. In the literature, waste is also referred to as outdated, expired, deteriorated, or perished item.

In the literature, there are several different classifications for perishable goods:

Shelf life uncertainty: Nahmias [Nah82] classifies perishable goods into two different categories; based on having fixed or uncertain shelf lives.

Outdating type: According to Goyal and Giri [GG01], inventory goods can be divided into three categories based on obsolescence, deterioration, or no obsolescence/deterioration.

Decay type: Perishable goods are categorised based on having a fixed shelf life or continuous decay [Raa91].

Issuing Policies

Issuing policies are applied to determine which items should be used first to satisfy customer demand, based on their age. We hereby introduce and discuss the most commonly used issuing policies:

FIFO: The FIFO (first-in, first-out) issuing policy considers the oldest items first when meeting customer demand. Newer items are used if and only if there is no older item.

Let us assume the lifetime of all items is fixed and M in a perishable inventory model, the age of a new item is 0, and an item becomes outdated when its age reaches M. The items with Age $m < M$ cannot be used until the inventory level for all Ages b, $m - 1 \leq b \leq M$, are zero.

LIFO: In contrast to the FIFO issuing policy, LIFO (last-in, first-out) gives priority to the newest items while meeting customer demand. The items of Age $m > M$ are not allowed to be used before the inventory level for all Ages b, $0 \leq b \leq m - 1$, are zero.

Other issuing policies: Other than FIFO and LIFO, notable less commonly used issuing policies can be listed as FEFO (first-expired, first-out), LSFO (least-shelf-life, first-out), and LEFO (last-expired, first-out), [Nah11], [ÖRSVdH15], [MCS+20].

Most papers in the single-item perishable inventory literature study the FIFO or LIFO issuing policies, and the other issuing policies have received little at-

tention. Mendes et al. [MCS+20] compare FIFO, FEFO, and LSFO policies and discusses their advantages to each other. Sazvar er al. [SMAehGB16] study a multi-product perishable inventory model with the FEFO issuing policy. Moreover, Önal et al. [ÖRSVdH15] study the economic lot sizing problem with perishable items for the FIFO, LIFO, FEFO, and LEFO issuing policies. Another paper in the perishability literature that study the FEFO policy is La Scalia et al. [LSMMT19].

Expected Waste Calculation

In perishable inventory systems, there is no explicit formula of the expected waste and the calculation of the exact expected waste is extremely difficult, [Nah75], and to our knowledge no other attempts to formulate it in the literature. This is one of the major challenges in the perishable inventory problem. In order to deal with this challenge, several waste estimation procedures and heuristics were developed in previous studies. Broekmeulen and Donselaar's [BVD09] EWA policy is one of the well-known heuristics which aim to estimate the expected outdating. We will discuss the EWA policy in Section 4.4.2 in more detail.

Inventory State in Fixed Lifetime Perishable Inventory Systems

In perishable inventory systems with a fixed lifetime of $M > 1$ periods, inventory state is denoted by an M-dimensional state vector

$$\mathbf{x} = \{x_{M-1}, x_{M-2}, ..., x_m, ..., x_1\}$$

in which x_m represents the amount of items with a remaining lifetime of m periods for all $m = 1, ..., M-1$, [Nah82], and t represents the current period. Alternatively, the state vector may consist of the elements representing the amount of items of each age, similar to that of [PWHHvdV14], e.g., $I_{m,t}$ represents the amount of items of Age m in Period t. Since the items in the last period of their lifetime do not affect future decisions and they will be waste soon if they are not used, they are not included in the state vector. Therefore, the inventory state vector consists of $M-1$ elements for the inventory levels.

Literature Review on Perishability

Stationary Demand: Inventory systems with perishable goods have been studied in the literature since early 1960s, pioneered by [Vei60], [Bul64], and [VZ64]. According to [AK58], if the lifetime is exactly one period the problem reduces to a sequence of newsvendor problems. Veinott [Vei60] considers perishable inventory models with a single product, one-period shelf life, and deterministic demand, then shows that an optimal policy orders an amount equal to demand for the FIFO issuing policy and the when the lifetime is non-increasing. Bulinskaya [Bul64] considers an inventory model in which the items perish immediately with a probability of p and after 1 period with a probability of $1-p$. Van Zyl [VZ64] derives the DP functional equations describing the optimal policy for a perishable inventory system with a product lifetime of 2 periods. Nahmias and Pierskalla [NP73] assume full backlogging of unsatisfied demand, then generalise Van Zyl's findings, derive the single-period and multi-period cost functions, and formulate the optimal policy for the lifetime $M = 2$ case. Nahmias [Nah75] and Fries [Fri75] independently and simultaneously extend [NP73]'s work into the M-period lifetime problem, derive the single-period and multi-period cost functions, and analyse the structure of the optimal policy for the fixed lifetime of $M \geq 2$ periods. While [Fri75] assumes lost sales and per unit per period shortage cost, [Nah75] considers per unit per period shortage costs and full backlogging of unmet demand. [Nah78] extends the previous studies by including a fixed setup cost into perishability for the first time in the literature. Nahmias [Nah78] shows that the (s, S) policy is the optimal policy for the one-period problem when the lifetime is 2 periods. However, the optimal policy structure becomes too complicated for the general M-period lifetime and one-period problem and the fact that the costs except the fixed setup cost is not convex, [NP76], makes the application of the K-convexity in Scarf [SAKS60] quite difficult. Even though they do not prove the structure of optimal policy in this paper they provide an (s, S) approximation for the multi-period problem and shows it is possible that the optimal policy for the multi-period problem may have the same structure as the (s, S) policy. To our knowledge, there is no paper in the literature that proves the optimal policy for the multi-period problem studied in [Nah78]. Recently, Zhang et al. [ZSC16] propose several approximation algorithms to the perishable inventory problem with setup costs.

Review Papers: A comprehensive review, summary, and classification about

the earliest literature can be found in [Nah82]. [GG01, BRT12] classify the perishable inventory literature in since 1990 and 2001, respectively. [Nah11, KSWD11] conduct extensive literature reviews about perishable inventory systems. [Raa91, PV14, JCS16, CKR18] are notable studies that review the perishability literature. All these reviews consider papers with stationary stochastic or deterministic demand, finite or infinite horizon, periodic or continuous review, single-echelon or multi-echelon, single-product or multi-product, fixed or stochastic lifetime, setup cost or no setup cost, fixed or stochastic lead time or no lead time, shortage or lost sales, and pricing in perishable inventory systems.

According to Raafat [Raa91], perishable inventory models can be classified in terms of the following aspects:

- Single item or multiple items.
- Deterministic or stochastic demand.
- Stationary or non-stationary demand.
- Single period or multiple periods.
- Purchase or production models.
- Quantity discount or without quantity discount.
- Shortage or no shortage.
- Constant or changing deterioration rate.

Goyal and Giri [GG01] classify perishable inventory models based on the properties of the lifetime, given in the following:

- Fixed lifetime.
- Random lifetime.
- Proportional decay in terms of utility or physical quantity.

Moreover, Bakker et al. [BRT12] give two different classification schemes based on deterioration and demand:

- Deterioration type:
    1. Fixed lifetime
    2. Age dependent deterioration rate
    3. Time or inventory (but not age) dependent deterioration rate.

- Demand type:
    1. Uniform demand, e.g. demand is a constant, fixed number of items.
    2. Stock-dependent demand.
    3. Time-varying demand.
    4. Price-dependent demand.

Perishable Inventory Systems with Non-Stationary Demand: Perishable inventory systems with non-stationary demand have received little attention and all the papers studying non-stationary stochastic demand were published in recent years. Haijema et al. [HvdWvD07] study a single-item, single-location, discrete-time, periodic review blood platelet inventory with linear production, holding, and shortage costs, and consider a weekly cyclic demand pattern, e.g., demand distributions over 7 days of a week but stationary across weeks. They propose a combined Markov decision process (MDP) and simulation approach. Haijema et al. [HvDvdWS09] extend this study into the non-stationary stochastic demand. Broekmeulen and Donselaar [BVD09] develop a heuristic for the perishable inventory problem with weekly cyclic demand, fixed review invervals, batch orders, and a fixed positive lead time. They also develop a waste estimation procedure called EWA. Haijema et al. [Hai13] deals with an extension of the problem in [HvdWvD07], includes a fixed setup cost, considers both the FIFO and LIFO issuing policies, and proposes a weekday-dependent (s_t, S_t, q_t, Q_t) ordering policy. Minner and Transchel [MT10] consider non-stationary stochastic demand, weekly demand pattern, FIFO and LIFO issuing policies, deterministic lead time, multiple service levels varying within a period, and no fixed setup cost.

Hendrix et al. [HHRPW12] study a finite horizon, single-item, single-location, discrete-time, periodic review perishable inventory system with non-stationary stochastic demand, a fixed setup cost and variable ordering, holding, outdating costs, a fixed lifetime, zero lead time, and a service level requirement. They formulate the problem as an SP model and propose an SDP approach. They show that the SDP model does not necessarily generate the optimal parameters for the service level version of the problem. Rossi [Ros13] works on the same problem, applies the CP model in Rossi et al. [RTHP10], and compare the performance of model with a brute-force approach. Both formulations are under the static-dynamic uncertainty of [BT88]. Pauls-Worm et al. [PWHHvdV14] extend [TK04]'s work and incorporate their MILP formulation in the perishability

problem with an α-service level constraint. Since the model finds a solution based on the deterministic-equivalent inventory compositions which consist of the amount of items of each age, it has several issues with fulfilling the service level requirement in most instances due to the incompatibility with the deterministic-equivalent and the actual inventory composition values. In order to fix this issue, Hendrix et al. [HPWR$^+$15] propose an MINLP approach based on the smoothed Monte Carlo method. Pauls-Worm et al. [PWHAH16] develop an MILP formulation for a different version of this problem. They consider long lead times, lost sales, and a fill-rate constraint. Gutierrez-Alcoba et al. [GAHG$^+$15] formulate an MINLP approximation to the lost sales and β-service level version of the problem. Gutierrez-Alcoba et al. [GARMBH17] extend the work and the discussion in [Ros13]. They include penalty cost instead of the service level constraint and introduce two new heuristics that extend Silver's heuristic, which are the analytical approximation and the Monte Carlo simulation optimisation approaches.

Table 2.1 compares and summarises the perishable inventory literature with non-stationary demand.

Other Approaches in Perishability: In recent literature, machine learning approaches have gained attention in perishable inventory management under uncertainty. While traditional optimisation methods rely on explicit modelling and tractable formulations, machine learning techniques such as deep reinforcement learning allow learning effective inventory policies without requiring full problem specification. Prestwich [Pre24] which employs machine learning techniques to solve complex optimisation problems. De Moor et al. [DMGB22] address the instability of deep reinforcement learning in perishable inventory problems and improve performance through reward shaping. Selukar et al. [SJK22] study multi-item perishable inventory systems and use deep reinforcement learning to minimise waste and cost in a sustainable supply chain setting. Nomura et al. [NLN23] propose a method based on deep reinforcement learning in order to optimise ordering policies with fixed product lifetimes and uncertain demand. Yavuz and Kaya [YK24] apply deep reinforcement learning to joint pricing and inventory control of perishable products under stochastic demand. Onyeaka et al. [OAM$^+$25] develop an AI-based approach to minimise food waste and spoilage. These studies show that learning-based approaches are increasingly used as scalable alternatives to traditional optimisation in perishable inventory management.

The Literature Gap and Our Contribution to Perishable Inventory Management

Our study aims to address the gap in the perishable inventory problem with non-stationary demand and proposes a novel solution method inspired by and employing the well-known approaches in the literature, such as the shortest path, the Wagner-Whitin algorithm, and the Silver-Meal heuristic. We consider both the penalty cost and the α-service level cases in this dissertation. Our study differs from that of Gutierrez-Alcoba et al. [GARMBH17] in the following ways: they do not employ the static-dynamic strategy, do not generate a pre-determined replenishment plan, and propose two heuristics based on using the observed inventory state at the beginning of a replenishment period and determining the next replenishment cycle with the lowest average cost. In our approach, we develop a pre-determined replenishment plan based on the static-dynamic uncertainty of Bookbinder and Tan [BT88], formulate the problem as a network model and find the near-optimal path which represents the near-optimal replenishment plan.

2.1.4 Cold Food Supply Chains

Cold food supply chains (CFSCs) are special supply chain systems designed to transport food products which are subject to losing their quality due to temperature. Fruits, vegetables, dairy, meat, and seafood are several examples of these kind of food products. In order to preserve their quality as food and for safety, they should be kept within specific temperature ranges, depending on the food type, during production, storage, transportation, and distribution stages. Several main concerns in the area of CFSCs can be listed as energy consumption, sustainability, waste, environmental factors, and technology issues. Due to transportation and refrigeration activities, CFSCs require plenty of energy consumption. In order to make energy consumption more efficient and sustainable, renewable energy sources and energy-efficient refrigeration technologies could be applied. Moreover, disruptions in CFSCs and refrigeration systems might result in food spoilage. Dealing with spoiled food and minimising waste due to spoilage should be taken into account in order to manage a CFSC successfully. In addition, energy sources used in CFSCs, such as fuel used in transportation and refrigeration, often tend to have significant impact on the environment, e.g., greenhouse gas emissions. Reducing CO_2 emissions and carbon footprints can be achieved by using more sustainable and environmental-friendly energy

Table 2.1: Summary of the non-stationary demand literature in perishable inventory control. D, S, and NS represent deterministic, stationary stochastic, and non-stationary stochastic demand, respectively.

#	Author	Year	Demand Type	Issuing Policy	Setup Cost	Lead Time	Service Level / Penalty	Method
1	Haijema et al. [HvdWvD07]	2007	S (weekly cyclic)	FIFO		+	+	MDP + Simulation
2	Haijema et al. [HvDvdWS09]	2009	NS	FIFO		+	+	Simulation
3	Broekmeulen & Donselaar [BVD09]	2009	S (weekly cyclic)	FIFO		+		Heuristic (EWA)
4	Haijema et al. [Hai13]	2013	NS	FIFO, LIFO	+	+	+	(s_t, S_t, q_t, Q_t) policy
5	Minner & Transchel [MT10]	2010	NS	FIFO, LIFO		+	+	Periodic Review
6	Hendrix et al. [HHRPW12]	2012	NS		+		+	SDP + SP model
7	Rossi [Ros13]	2013	NS		+		+	Constraint Programming
8	Pauls-Worm et al. [PWHHvdV14]	2014	NS	FIFO (implicit)	+		+	MILP
9	Hendrix et al. [HPWR+15]	2015	NS		+		+	MINLP + Monte Carlo
10	Pauls-Worm et al. [PWHAH16]	2016	NS		+	+	+	MILP
11	Gutiérrez-Alcoba et al. [GAHG+15]	2015	NS		+		+	MINLP Approximation
12	Gutiérrez-Alcoba et al. [GARMBH17]	2017	NS				+	Heuristics (Analytical, MC)
13	Gulecyuz et al. [GOT25]	2025	NS	FIFO	+		+	Heuristic (Shortest-path + Silver-Meal + Wagner-Whitin)

resources, building efficient transportation networks and systems, and efficient and advanced refrigeration techniques, [BSA+20]. Another concern is the variation of food types which require their own refrigeration settings and must be kept together with other food types. Table 2.2 summarises the literature review on CFSCs that we conducted.

Sustainable Cold Food Supply Chain Models

Since preserving the temperature of cold products within the recommended limits comes at the cost of higher energy consumption and further emission, most existing literature on CFSCs pays particular attention to sustainability. James and James [JJ10] studied the impact of CFSCs on global climate change and discovered that cold storage facilities account for approximately 50% of carbon emissions in these chains. According to Waltho et al. [WEG19], transportation and storage activities account for more than one-third of the total emission generated by a CFSC. This prompts an increasing research interest in the sustainability aspects of CFSCs. Numerous researchers introduced mathematical models for CFSC to incorporate environmental considerations into the traditional supply chain management models. These models broaden the focus from merely the business aspects of supply chains to include sustainability issues (Al Theeb et al. [ATSAHA20], Allaoui et al. [AGCB18]).

A review of the relevant literature identifies the parameters to be measured and lays the groundwork for our work. A variety of metrics have been used to evaluate carbon emissions, including the energy used by vehicles, distance travelled, and load transferred. Validi et al. [VBB14] developed a multi-objective green distribution model for a two-layer food supply chain system. Hsiao et al. [HCC17] focused on the cold chain distribution planning problem for perishable foods. Bottani et al. [BCNT19] investigated the issue of economic and environmental sustainability evaluation in CFSCs, focusing on reverse logistics. Babagolzadeh et al. [BSA+20] developed a two-stage stochastic programming model to investigate the CO_2 emissions under uncertain demand. These studies evaluated carbon emissions only in terms of the quantity of energy that was used by vehicles. Mallidis et al. [MDV12] developed a multi-objective supply chain design to minimise cost and CO_2 and particulate matters emissions. Pan et al. [PBF13] proposed a supply network pooling to reduce CO_2 emissions. Soysal et al. [SBRVDV14] presented a model for logistical network problems of perishable products under emissions consideration, specifically for beef logistics. Dan-

loup et al. [DMA+15] developed a model to reduce transportation greenhouse gas emissions with collaborative distribution. Moheb-Alizadeh and Handfield [MAH18] developed a multi-objective mixed-integer non-linear programming (MINLP) model for efficient and sustainable supplier selection and order allocation under stochastic demand. Musavi and Bozorgi-Amiri [MBA17] studied a multi-objective sustainable hub location-scheduling problem for perishable CFSC. Hariga et al. [HAaS17] studied the multi-stage cold supply chain problem under carbon tax regulation. Zhen et al. [ZHW19] proposed a bi-objective optimisation model is proposed for a closed-loop supply chain network under demand uncertainty and focused on trade-off between operating costs and CO_2 emissions. Li et al. [LSZ19] applied the ant colony optimisation algorithm into the multi-depot green vehicle routing problem with multiple objectives. These studies calculated an empty vehicle's emission in terms of the unit of distance travelled. Elhedhli and Merrick [EM12] proposed a Langrangian-based heuristic for green supply chain network design to reduce carbon emissions. Galal and El-Kilany [GEK16] studied sustainable two-echelon agri-food supply chains with uncertain demand and lead time, and developed a discrete-event simulation model. Ghahremani et al. [GNPN20] studied and proposed a robust optimisation model for multi-objective, multi-product, multi-period, closed-loop green supply chains under uncertain demand and discount. Saif and Elhedhli [SE16] modeled the cold supply chain design problem as a concave mixed integer program and proposed a simulation based approach. [MDV12], [EM12], [GEK16], [GNPN20], and [SE16] calculated CO_2 emissions according to the quantity of load transfer. As a result, distance and load units are considered essential factors in the computation of CO_2 emission. Bozorgi et al. [BPN14] and Bozorgi [Boz16] analysed transportation emissions based on the number of fully loaded refrigerated trucks that travelled during each period, the average load, and the distance travelled for transferring various cold products. Similarly, [MAH18] considered time, distance, and load quantity to be the critical components of CO_2 emission. This implies that distance and load unit are essential factors in CO_2 emission computations. Stellingwerf et al. [SLC+18] adopted a broader perspective. They determined the fuel consumption in a CFSC based on the load weight, road slope, distance travelled, and vehicle speed. The carbon emission generated by the chill transport is also addressed in their study by considering the amount of fuel used to refrigerate the load. They discovered that including chill transport emission in their calculations resulted in considerable route adjustments. [GEK16] also addressed the emission of chill transport

in their studies, and Fichtinger et al. [FRGB15] studied the emission of chill inventory management. Chen et al. [CH15] and [SLC+18] were unique in considering the CO_2 emissions resulting from opening vehicle doors and unloading products. However, recent studies such as Zhang and Chen [ZC14] suggest that offloading in high-temperature environments significantly impacts CO_2 emission. Recently, in order to enhance resilience and sustainability of food supply chains, Singh et al. [SRDM23] employ the digital twin technology. Aazami et al. [ASM21] considers a multi-level supply chain for perishable products and develop a heuristic. Turan and Ozturkoglu [TO22] develop a conceptual framework for CFSCs. Shi et al. [SLL+22] propose an intelligent green scheduling system, proposes a multi-objective optimisation model, and an optimisation algorithm for a cold supply chain with deterministic demand, by taking CO_2 emission concerns into account. Moreover, Chen et al. [CWF22] study a CFSC model with deterministic demand and proposes an optimal replenishment schedule by considering CO_2 emissions.

Cold Food Supply Chain Models and Demand Uncertainty

Uncertainty is an inherent aspect of decision-making in real-world scenarios. Most supply chain and inventory management research assumes that the demand parameter is known, or the probability distribution is known and static if the parameter is unknown. [BSA+20] recently proposed a two-stage stochastic programming model for minimising the economic and environmental impacts of a temperature-sensitive food supply chain with stationary stochastic input data. Similarly, [MZA20] developed a multi-objective mixed-integer linear programming model and applied several decision trees to analyse stationary uncertainty. Soysal et al. [SBRHVDV15] presented an optimisation model for minimising total cost while meeting CO_2 emission and service level requirements and uncertain stationary demand. They employed the fuel consumption parameters for transportation and refrigeration derived in Barth et al. [BYS05], Franceschetti et al. [FHVW+13], Demir et al. [DBL12], Stellingwerf et al. [SLC+18], Bektas and Laporte [BL11], and Tassou et al. [TDLG09]. [SBRHVDV15] might be the most relevant research regarding the extent and objectives; thus, uncertain demand is considered stationary, and emissions generated by cold facilities are neglected. Recently, [KSÇS23] combined refrigeration activities with the IRP model and took CO_2 emissions generated by refrigeration into account. Both [SBRHVDV15] and [KSÇS23] calculated the CO_2 emissions using the amount of fuel consumption.

Demand often has a varying probability distribution in practice, see Graves and Willems [GW08]. Stationary demand has a static distribution, though this is not the case with non-stationary demand. Zhang and Chen [ZC14] explain that if the probability distribution is not steady, demand is non-stationary stochastic. However, research on non-stationary demand in CFSCs is limited, with most studies focusing on stationary demand. Considering a stationary demand rather than a non-stationary parameter significantly increases the costs of a supply chain Tunc et al. [TKTE11]. Therefore, conducting research on CFSCs with non-stationary stochastic demand is critical.

The Literature Gap and Our Contribution to Cold Food Supply Chains

The main contributions of our study could be summarised as follows: Firstly, we propose a mixed-integer programming (MIP) IRP model that enables significant cost savings for CFSCs, allows flexibility in ordering decisions and solves the excess buffer stock and higher holding cost issue caused by the pre-determined order quantities found by the (R, Q) policy. Moreover, we incorporate CO_2 emissions caused by various refrigeration activities to the non-perishable version of the inventory routing problem in Soysal et al. [SBRHVDV15] by employing the findings of Köseli et al. [KSÇS23] related to refrigeration. In this dissertation we simultaneously cover environmental concerns regarding CO_2 emissions, refrigeration and transportation activities, and inventory control under non-stationary stochastic demand and a service level constraint. Our aim is to fill this gap in the literature.

2.2 Solution Techniques

Various solution techniques and approaches have been developed in order to enable the formulation and solution of complex problems to maximise operational efficiency, minimise costs, and enhance overall system performance. This section examines four main solution approaches and frameworks: Dynamic Programming (DP) in Section 2.2.1, Mixed Integer Programming (MIP) in Section 2.2.2, and the Inventory Routing Problem (IRP) in Section 2.2.3. Each subsection explores the theoretical background and the specific relevance to perishable inventory management and cold supply chains with non-stationary stochastic demand.

Table 2.2: Summary of the CFSC literature review. D, S, and NS represent deterministic, stationary stochastic, and non-stationary stochastic, respectively.

#	Author	Year	Demand type	Fixed dispatching cost	Fixed ordering cost	Linear travel costs	Penalty cost/ Service level	Holding cost	Empty vehicle travelling emission	Load transferring emission	Primary refrigeration emission	Chill transfer emission	Unloading emission
1	Al Theeb et al. [ATSAHA20]	2020	D			*	P	*					
2	Allaoui et al. [AGCB18]	2018	D			*							
3	Amorim & AlmadaLobo [AAL14]	2014	D			*							
4	Bortolini et al. [BFF+16]	2016	D			*		*		*			
5	Bottani et al. [BCNT19]	2019	D			*	P		*	*			
6	Bozorgi et al. [BPN14]	2014	D	*		*			*	*			
7	Bozorgi [Boz16]	2016	D			*		*		*			
8	Chen & Hsu [CH15]	2015	D							*			*
9	Danloup et al. [DMA+15]	2015	D			*			*				
10	Elhedhli & Merrick [EM12]	2012	D			*				*			
11	Fichtinger et al. [FRGB15]	2015	D		*	*							
12	Hariga et al. [HAaS17]	2017	D	*				*		*			
13	Hsiao et al. [HCC17]	2017	D	*		*				*			
14	Li et al. [LSZ19]	2019	D			*				*			
15	Mallidis, I. et al. [MDV12]	2012	D			*		*		*			
16	Meneghetti & Ceschia [MM15]	2015	D										
17	Musavi & Bozorgi-Amiri [MBA17]	2017	D			*			*	*			
18	Torabia et al. [TBM15]	2015	D			*	P			*			
19	Pan et al. [PBF13]	2013	D			*			*	*			
20	Tordecilla et al. [TMPC18]	2018	D	*	*	*		*		*			
21	Saif & Elhedhli [SE16]	2016	D		*	*		*		*			
22	Shamayleh et al. [SHAaD19]	2019	D		*	*		*		*			
23	Soysal et al. [SBRVDV14]	2014	D	*		*	P	*	*	*			
24	Stellingwerf et al. [SLC+18]	2018	D			*		*		*		*	
25	Validi et al. [VBB14]	2014	D		*	*				*			
26	Babagolzadeh et al. [BSA+20]	2020	S	*			P	*	*	*			
27	Jiang et al. [JZML20]	2020	S			*	P			*			
28	Solyalı et al. [SCL12]	2012	S	*		*		*					
29	Galal & El-Kilany [GEK16]	2016	S			*	S	*		*		*	
30	Ghahremani et al. [GNPN20]	2020	S		*	*	P	*		*			
31	Hassanzade-Amin & Zhang [AZ13]	2013	S			*							
32	Moheb-Alizadeh & Handfield [MAH18]	2018	S		*	*		*		*			
33	Mohebalizadehgashti et al. [MZA20]	2020	S			*				*			
34	Yavari & Geraeli [YG19]	2019	S		*	*		*					
35	Zhen et al. [ZHW19]	2019	S			*		*		*			
36	Soysal et al. [SBRHVDV15]	2015	S			*	S	*		*			
37	Tarim & Kingsman [TK06]	2006	NS		*		P	*					
38	Tarim & Smith [TS08]	2008	NS		*		S	*					
39	Seth & Pandey [SP09]	2009	NS				P	*					
40	Tunc et al. [TKTE11]	2011	NS	*	*		P	*					
41	Zhang & Chen [ZC14]	2014	NS				S						
42	Pauls-Worm et al. [PWHHvdV14]	2014	NS		*		S	*					
43	Purohit et al. [PCS16]	2016	NS	*	*		S	*					
44	Gutierrez-Alcoba et al. [GARMBH17]	2017	NS			*	P	*					
45	Sinaga et al. [SPA16]	2016	NS			*	S	*					
46	Chen et al. [CWF22]	2022	D	*	*	*		*	*	*	*	*	*
47	Shi et al. [SLL+22]	2022	D	*		*	P	*	*	*	*	*	*
48	Jahdi et al. [JGO+24]	2024	NS		*	*	S	*	*	*	*	*	*

2.2.1 Dynamic Programming

Dynamic programming (DP) is a mathematical optimisation technique used to solve problems which can be divided into smaller sub-problems. Introduced by [Bel54, Bel66], DP is particularly effective for multi-stage decision processes. The working principle of DP is recursively solving these sub-problems which are part of the larger problem, then combining the solutions of the sub-problems to address the larger problem. An optimal solution can be constructed by combining the optimal solutions of its sub-problems. Dynamic programming is often applied to problems with sequential decision making, in which decisions at one stage affect subsequent stages.

The *Principle of Optimality* is the basic principle of dynamic programming, and it is a prerequisite that must be held in order to apply DP. It is stated by Bellman as follows:

An optimal policy has the property that whatever the initial state and initial decision are, the remaining decisions must constitute an optimal policy with regard to the state resulting from the first decision."

We will explain the key concepts of DP and the relevant aspects with our study. We then discuss a specific version of dynamic programming which takes uncertainty and stochasticity into consideration.

Key Concepts

Similar to that of [XRT19], we define the key elements of DP algorithms as follows:

Stage: In DP, decision making takes place at the moments called decision epochs which can be discrete periods or moments in continuous time. Since the problem we consider in this dissertation deals with discrete time periods, we consider each period to be one stage of the DP algorithm.

State: A state represents the system's current status at each stage and before any decision is made. It is an indicator a sub-problem of the larger main problem contains the information for future decision making. It can consist of one or more state variables, hence there can be multi-dimensional states based on the number of state variables. In our study, we deal with $I \times M \times T$ states since we consider perishable items with M-period fixed lifetime. Moreover, DP states can have continuous or discrete values. We perform our numerical exper-

iments on the inventory systems with discrete or discretised demand. However, the results can be applied to continuous demand cases. An optimal cost of the set of recursive sub-problems is associated with its initial state. It is repeated recursively until the whole problem is covered, then the optimal cost of the problem is obtained. In the perishable inventory system that we study in this dissertation, each combination of the amount of inventory levels of each age at the beginning of each period is considered to be one DP state.

Action: Actions are the decisions that are made, given a system state at a decision epoch and selected from a set of all allowable actions. Actions can be deterministic or stochastic, as well as it can involve one or more decision variables.

Decision variable: A decision variable represents the choices available to the decision maker. It can take values based on the actions selected. **Transition function:** A transition function describes the evolution of the system from one state to another, based on the actions taken and the observations. **Policy:** The action selection procedure based on the state of each stage.

Perishable inventory control involves the actions of the order timings and quantities at a given stage, taking the number of items of each age and the number of outdated items into account.

Immediate cost: Immediate cost is immediately incurred to the decision maker as a result of taking a specific action. It is the function of the state-action pair. In perishable inventory control with non-stationary stochastic demand, the immediate cost is comprised of the single-period expected ordering, holding, penalty (if a penalty is applied for shortages), and outdating costs.

Objective function: This is the function to be maximised or minimised. Also referred to as functional equation, [Bel54]. It considers the transition function to the next state given an action and the current state, and evaluates all the possible actions and chooses the optimal action that minimises or maximises the objective function. Objective function can be expressed by forward or backward induction, depending on starting the recursion from the beginning or the end of the problem. For instance, the objective function of a discrete-time minimisation problem can be written as a Bellman equation with backward induction which follows the structure given below:

$$V_t(x_t) = \min_{u_t}\{C_t(x_t, u_t) + V_{t+1}(x_{t+1})\}$$

where x_t and u_t represent the state and the taken action at the beginning of stAge t, respectively, $C_t(.)$ represents the immediate cost function in stAge t, $V_t(.)$ represents the objective function in stAge t, and x_{t+1} is the state in stAge $t+1$. It is calculated by the state transition function $\Phi_t(x_t)x_{t+1}$ which transforms the current state x_t into the next state x_{t+1}.

Stochastic Dynamic Programming

Stochastic dynamic programming (SDP) is a type of dynamic programming designed for situations where uncertainty is involved. Unlike regular DP, which assumes everything is known and predictable, SDP works with problems where some elements, such as demand or costs, are random and follow a probability distribution. This makes SDP useful for solving real-world problems, such as inventory management or financial planning, where outcomes are not always certain. In SDP, randomness affects the decisions and their outcomes. For example, in an inventory system, the demand for products might vary unpredictably. SDP takes this into account by calculating the expected value of the total cost or reward, based on the probabilities of these random events. The goal of SDP is to find the best decision-making rule, called a policy, that minimises (or maximises) the expected total cost (or reward) over time. This process uses a recursive equation that includes the probability of different outcomes. For a given stage, the relationship can be written as:

$$\mathbb{E}\{V_t(x_t)\} = \min_{u_t} \mathbb{E}\{C_t(x_t, u_t) + V_{t+1}(x_{t+1})\}$$

where $\mathbb{E}$ represents the expectation.

This section is inspired by [Put14, Pow07].

SDP in Stochastic Demand

In the context of stochastic inventory management, SDP is a powerful technique for managing inventory over multiple time periods under uncertain demand. It models inventory decisions as a sequential decision-making process in which several elements such as demand are the stochastic elements. For instance, it can be used to determine replenishment policies that minimise total cost while meeting customer demand. SDP is particularly useful in complex inventory systems such as multi-echelon supply chain models and perishable inventory systems. In case of perishable inventory systems, the perishability of items adds

additional complexity to the problem, as stock must be consumed or sold before it expires, necessitating dynamic adjustments to inventory levels, [HHRPW12].

Challenges and Limitations

Despite the fact that DP is a powerful optimisation tool, its application is often hindered by the *curse of dimensionality*, where the computational complexity increases exponentially with the number of state and decision variables, [Pow07]. For instance, as the item lifetime increases the dimensionality increases in perishable inventory systems, [HHRPW12]. Therefore, dynamic programming easily becomes computationally infeasible and intractable for larger lifetimes. Dealing with the curse of dimensionality is one of our primary rationales to develop the heuristics in this study.

2.2.2 Mixed Integer Programming

We first analyse mixed integer programming (MIP). Then we provide the MIP approach proposed by [TK04] for the static-dynamic uncertainty strategy of [BT88], which forms a basis to our study.

Fundamentals of MIP

Based on [Win04], we analyse MIP. An integer programming problem (IP) is an optimisation problem in which at least one of its variables must be non-negative integers, which leads not to satisfy the divisibility assumption, such that the decision variables are allowed to take fractional values. According to [Win04], IPs are more challenging to solve compared to linear programming problems (LP) in which all the decision values are allowed to have fractional values. IPs can be classified according to the requirements that all or some of its decision variables must be integers. If all the decision variables in an IP are required to take integer values, it is called *pure integer program*, as in the example given below and taken from [Win04]:

$$\max z = 3x_1 + 2x_2$$

subject to

$$x_1 + x_2 \leq 6$$

$$x_1, x_2 \text{ integer.}$$

If at least one of the decision variables can take integer values and the others are allowed to take fractional values, it is called *mixed integer program*, as given below:

$$\max z = 3x_1 + 2x_2$$

subject to

$$x_1 + x_2 \leq 6$$

$$x_1 \text{ integer.}$$

Mixed integer linear programming (MILP) problems combines linear programming with the constraints which enforce at least one of the decision variables to be non-negative integers, making it particularly useful to model discrete decision variables in supply chain management, such as indicating the order placements, replenishment periods, used vehicles, together with continuous decision variables such as the optimal order quantity.

MILP problems consist of a linear objective function and a set of constraints. Decision variables in MILP can be classified as continuous (real numbers) or discrete (non-negative integers). Discrete variables can be defined and used as follows:

Binary variables: They are used as the indicative variables, such as indicating whether an action should be/have been taken or not, e.g., opening a new warehouse, if a particular vehicle is used or not. **Integer variables:** They are used to represent the optimal discrete quantities, e.g., the number of vehicles used, the number of customers visited by a single vehicle.

Tarim and Kingsman's MILP Formulation

We introduce and discuss the MILP formulation by [TK04] in order to find the optimal replenishment schedule and the order-up-to levels for the service level case that minimise the total finite horizon cost. This model operates according to the (R, S) ordering policy and re-formulates [BT88]'s static dynamic uncertainty strategy.

In the deterministic-equivalent version of the model given in Section 2.1.2, the order quantity variables Q_t are deterministic decision variables, their values are determined before the start of the planning horizon, and cannot be modified later. By examining the analogy between the deterministic-equivalent model and the static uncertainty model and considering the dynamic uncer-

tainty model which allows to modify the replenishment timings and quantities later, [BT88] develop a static-dynamic uncertainty formulation which consists of 2 steps: Determining the replenishment periods using various techniques such as DP or IP, and determining the order sizes with respect to the pre-determined schedule and the observations over the planning horizon.

According to [TK04], the Bookbinder and Tan static-dynamic method solves problems in two independent stages without considering how they interact each other and does not take the unit ordering cost into account and the unit ordering cost component is a constant, since the constraint is binding at optimality. The problem needs to be formulated to find the optimal solution under the static-dynamic uncertainty strategy. This approach determines the timing and size of orders in order to minimise expected costs while meeting demand forecasts and meeting the service level requirements. Their model allows the simultaneous determination of the timing and quantity of the replenishment orders as a single step. The initial decisions depend on the expected demand information that can be available over time and how to react to this information. Therefore, the replenishment schedule is determined before the planning horizon starts. The actual order quantities are determined only at the beginning of the pre-determined replenishment periods and depending on the realised demand.

Based on the motivation discussed above, they develop an MILP formulation in terms of a new decision variables R_t which represent the order-up-to level in each Period t. Their formulation is given in the following:

$$\min \mathbb{E}(TC) = \sum_{t=1}^{T} \Big(a\delta_t + h\mathbb{E}(I_t) + c\mathbb{E}(S_t) - c\mathbb{E}(I_{t-1}) \Big) \tag{2.14}$$

subject to

$$\mathbb{E}(I_t) = \mathbb{E}(S_t) - \mathbb{E}(d_t), \quad t = 1, 2, ..., T.$$

$$\mathbb{E}(S_t) \geq \mathbb{E}(I_{t-1}), \quad t = 1, 2, ..., T.$$

$$\mathbb{E}(S_t) - \mathbb{E}(I_{t-1}) \leq M\delta_t, \quad t = 1, 2, ..., T.$$

$$\mathbb{E}(I_t) = \sum_{j=1}^{t} \Big(F^{-1}_{d_{t-j+1}+d_{t-j+2}+...+d_t}(\alpha) - \sum_{k=t-j+1}^{t} \mathbb{E}(d_k) \Big) P_{tj}, \quad t = 1, 2, ..., T.$$

$$\sum_{j=1}^{t} P_{tj} = 1, \quad t = 1, 2, ..., T.$$

$$P_{tj} \geq \delta_{t-j+1} - \sum_{k=t-j+2}^{t} \delta_k, \quad t = 1, 2, ..., T, \quad j = 1, ..., t.$$

$$\mathbb{E}(I_t) \geq 0, \mathbb{E}(I_t) \geq 0, \delta_t, P_{tj} \in 0, 1 \quad t = 1, 2, ..., T, \quad j = 1, ..., t.$$

The binary integer decision variable P_{tj} is 1 if the most recent order prior to Period t was in Period $t-j+1$, and 0 otherwise. The binary variable δ_t indicates if Period t is a replenishment period or not. M is a very large number. I_t is the ending inventory level for Period t. Moreover, the unit ordering cost is denoted by c. Since the unit ordering cost component of 2.14 can be expressed as a function of $\mathbb{E}(S_T)$, see [TK04], Tarim and Kingsman's MILP formulation does not ignore the unit ordering cost and incorporates it into determining the optimal parameters.

In this dissertation, we formulate the inventory management component of the model for CFSCs, discussed in Chapter 3, based on the MILP formulation of [TK04].

2.2.3 Inventory Routing Problem

The well-known Inventory Routing Problem (IRP) offers a framework to combine both inventory management and vehicle routing decisions simultaneously, see Jemai et al. [JRK13]. Despite the fact that IRP is NP-hard, it is possible to develop a practical model. The IRP focuses on repeated distribution of a single product from a single distributor to a set of a finite number of customers over a finite planning horizon of T periods or an infinite planning horizon, and using a fleet consisting of a finite number of vehicles, see Campbell et al. [CCKS98]. The aim of IRP models is to determine when and how much to serve customers and finding the optimal route. The IRP is typically formulated as an MILP model and the objective function is defined to minimise costs related to transportation, inventory management, or both transportation and inventory management costs.

The IRP tackles both the inventory management and vehicle routing problems together. The inventory management component of the IRP focuses on inventory decisions which we discussed in the previous sections, such as replenishment timing and quantity and ensuring adequate stock levels to meet customer demand.

Vehicle Routing Problem

The vehicle routing problem (VRP) is an NP-hard IP problem that aims to determine the optimal set of routes for a vehicle fleet in order to deliver to customers from a given set, [DR59]. Consider the weighted graph $G = (V, E)$ in which $V = \{0, ..., n\}$ represents nodes and $E\{(i, j) : i, j \in V\}$ represents edges, which are interpreted as the depot, customers, and routes in the VRP. The assumptions in the VRP can be listed as follows:

- Each customer is served exactly once by exactly one vehicle.
- Each route starts and ends at the depot (the node 0).
- The capacity of the vehicles cannot be exceeded.

There are several variations of the VRP, such as the capacited vehicle routing (CVRP), vehicle routing with time windows (VRPTW), vehicle routing with backhauls (VRPB), and vehicle routing with pickup and delivery (VRPPD).

Literature Review on Stochastic and Sustainable IRP

In the earlier IRP literature, IRP models with deterministic demand and without sustainability concerns have been extensively studied, see Moin and Salhi [MS07] and Andersson et al. [AHC$^+$10]. However, real-world applications often require accounting for demand uncertainty, leading to the development of stochastic IRP models.

The IRP with stochastic demand has been studied since the early works of Federgruen and Zipkin [FZ84]. They developed dynamic models for periodic review systems under uncertainty. Kleywegt et al. [KNS02] formulated the problem as a Markov decision process. Adelman [Ade04] introduced an approach to stochastic inventory/routing to approximates the future costs of current actions using optimal dual prices of a linear program. Mirzapour Al-e-hashem et al. [MAEHMA11] proposed a multi-objective robust optimization model for a multi-product multi-site aggregate production planning problem under demand uncertainty. Solyalı and Süral [SS12] proposed a new transportation and shortest path based integer programming reformulation for the one-warehouse multi-retailer problem. Coelho and Laporte [CL13] proposed a branch-and-cut algorithm for the exact solution of several classes of IRPs. Devika et al. [DJN14] designed a mixed integer programming model based on triple bottom line approach for the multi-objective closed-loop supply chain network problem.

Bertazzi et al. [BBL15] provided a mathematical formulation of the IRP with transportation procurement and proved the sub-optimality of a policy based just on the average demand. Roldán et al. [RBC16] proposed robust customer selection methods for a dynamic and stochastic IRP. Hiassat et al. [HDR17] introduced a genetic algorithm approach for the inventory-routing problem with perishable products. Govindan et al. [GS17] provided a review of reverse logistics and closed-loop supply chains. [DHS+19] discussed the basic principles of the VRP.

Sustainability in IRP has also gained attention due to environmental and social concerns. Jabali et al. [JVDK12] presented a model that considers travel time, fuel, and CO_2 emissions costs in a time-dependent vehicle routing context. Demir et al. [DBL14] developed several multi-objective heuristics for the bi-objective pollution routing problem (PRP) which is a variation of VRP. Rahimi et al. [RBR16] proposed a green IRP model considering customer satisfaction and greenhouse gas emissions. Tirkolaee et al. [TAW21] applied a fuzzy chance-constrained programming technique to address the uncertainty and implemented a weighted goal programming method for the multi-trip location-routing problem for medical waste management during the COVID-19 pandemic.

Limited studies have been proposed for the IRP with non-stationary stochastic demand setting. In most existent studies, the probability distribution function for uncertain demand is static. However, this does not reflect most real life cases. Soysal et al. [SBRHVDV15] presented a model to handle inventory and routing decisions based on the static uncertainty and using the (R, Q) policy. This policy pre-determines replenishment timing and order quantities before the start of the planning horizon, and it does not allow enough flexibility in making replenishment decisions. Moreover, most sustainable IRP models only consider emissions and open-loop settings. Broader sustainability aspects, such as regulatory compliance, reverse logistics, or long-term resource efficiency are still understudied. In addition, there are only a few models that combine stochastic demand and sustainability objectives in CFSCs. Our publication Jahdi et al. [JGO+24], also discussed in Chapter 3 of this dissertation in detail, aims to address these gaps and make a significant contribution to the field.

Chapter 3

Determining the Policy Parameters to Improve the Sustainability of Cold Supply Chains Under Stochastic Demand

In this chapter, we present a novel inventory routing problem (IRP) approach that we studied in our publication [JGO+24], in order to find the policy parameters that minimise the total cost and CO_2 emissions. We develop a model considering the static-dynamic (R, S) policy proposed by [TK04] and by taking the total CO_2 emission concerns into account. In Section 3.1 we introduce the problem and our motivation briefly. We then describe the problem in detail as well as the emission formulations, and propose our IRP model that follows the (R, S) policy in Section 3.2. We present the extensive numerical analysis in Section 3.3. Finally, we conclude our study, provide managerial insights, and discuss further possible directions in Section 3.4.

3.1 Introduction

The global food system faces environmental sustainability challenges, and in addressing these challenges, the food supply chain (FSC) is an area that has received increasing attention. In traditional food logistics, reducing costs and increasing responsiveness were the primary objectives of a supply chain network. However, sustainability studies are becoming increasingly crucial to protect the

environment and also to improve food supply chains' social and economic aspects (see [BCNT19, AAL14]).

Improved routing of food logistics can reduce time, costs, and related emissions. The well-known NP-hard IRP offers a framework to address both inventory management and routing challenges by handling inventory management and vehicle routing simultaneously ([JRK13]). However, it is possible to develop a practical IRP model.

Soysal et al. [SBRHVDV15] have developed a multi-period IRP model to compute all inventory and routing decisions under the (R, Q) policy. The (R, Q) policy has been one of the most well-known policies used for handling inventory systems. According to this policy, Q units are ordered as the inventory level is at the reorder point R. Then the exact timing and quantities of future orders are known to the suppliers in advance. Therefore, it is not possible to change the pre-determined parameters with respect to the realised demand. Since the pre-determined parameters are not responsive to demand uncertainty under the (R, Q) policy, it is essential to keep more safety stocks which results in higher inventory holding costs for the system.

In order to deal with these types of uncertainties dynamically, we employ the (R, S) policy which allows us to develop recourse actions to be able to deal with such uncertainties as they arise. The (R, S) policy works in the same manner as [BT88]'s static-dynamic strategy. According to the (R, S) policy, the replenishment timings are fixed at the beginning of the horizon. Order-up-to-levels are also decided at the beginning of the planning horizon. However, the order quantities are determined later as demand become known. Thus, the distribution centre knows the timing of future orders and prepares to fulfill them, although the order quantities are unknown.

Note that some other methods, e.g., the dynamic uncertainty strategy of [BT88] and the rolling horizon approach, can provide optimal results, but not necessarily recommended in practice, since not knowing when an order takes place might be expensive, decisions need to be reviewed each period as the realised demand value becomes known, and the outcome of each decision may need a set-up in the next period. This approach could require a decision set-up almost every period ([BT88]). We aim to develop a model which is computationally efficient and easy to implement in practice, and provides near-optimal results while allowing flexibility in ordering decisions. Therefore, we apply the static-dynamic strategy and the (R, S) policy.

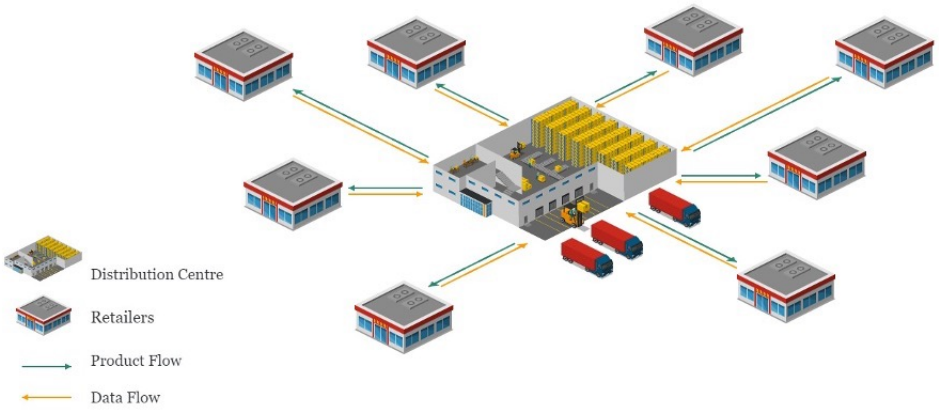

Figure 3.1: Schematic representation of the supply chain network.

3.2 Problem Description

We define our problem, introduce the formulations we use to calculate the CO_2 emissions occurring due to the transportation and refrigeration emissions, then formulate our proposed model, and finally explain the deterministic-equivalent approach we use to deal with the stochastic nature of the problem.

3.2.1 Our Problem

We study a single-item cold food supply chain (CFSC) problem with a single distribution centre and $V \geq 1$ retailers, as visualised in Figure 3.1. Demand realisations for each retailer i in each Period t, d_{it}, are independent random variables with known probability density functions $f_{it}(d_{it})$ and cumulative distribution functions $F_{it}(d_{it})$.

We define the problem on a complete graph $\mathcal{G} = \{\mathcal{V}, \mathcal{A}\}$, where $\mathcal{V} = \{0, ..., V\}$ is the set of nodes in which Node 0 represents the distribution centre and the remaining nodes represent the retailers, and $\mathcal{A} = \{(i, j) : i, j, \in V, i \neq j\}$ is the set of arcs representing the path between Nodes i and j. We denote $\mathcal{V}' = \{1, ..., V\}$ by the set of retailers. Items are shipped from the distribution centre to each retailer over a finite planning horizon of T periods, and using at most $K \geq 1$ available identical vehicles with a known finite capacity C per vehicle. We define the set of vehicles by $\mathcal{K} = \{1, ..., K\}$ and the set of periods by $t \in \mathcal{T}' = \{1, ..., T\}$. We assume the distribution centre and each retailer have unlimited capacity. The load carried by vehicle k from retailer i to retailer j, F_{ijkt}, cannot exceed the vehicle capacity at any time. The binary variable δ_{it} indicates if a replenishment order for retailer i is scheduled at the beginning of Period

t or not. S_{it} denotes the order-up-to level for retailer i in Period t, and the ending inventory level for retailer i in Period t is denoted by I_{it}. Orders for the same retailer can be shipped by multiple vehicles. The part of the order for retailer i which is carried by vehicle k is denoted by Q_{ikt}. The binary decision variable X_{ijkt} indicates if vehicle k traverses the arc (i, j) in Period t or not. In each Period t, vehicle k can only travel a single route starting and ending at the distribution centre, visits retailer i, and carries order for retailer i, e.g., $Q_{ikt} > 0$, if and only if a replenishment order is scheduled for that retailer, e.g., $\delta_{it} = 1$. Thus, it is not required to visit all the retailers in each period, and it is not required to operate all the K vehicles in a single period. The inventory level for all retailers at the beginning of the planning horizon, I_{i0}, is set to be zero. Replenishment lead times are not addressed in this study. However, lead times could be incorporated without loss of any generality. There is an α-service level requirement, therefore the customer demand in each period and for each retailer must be satisfied with a probability of at least α. Unsatisfied demand is backordered in the next period. Items have an indefinite shelf life and no waste occurs due to perishability.

In our problem, the costs are classified as the inventory, transportation, and refrigeration costs. For any retailer i, ordering costs consist of a fixed setup cost per order a_i and a variable ordering cost c_i which is linear in the order quantity $\sum_{k=1}^{K} Q_{ikt}$ are incurred at the beginning of each replenishment period. In addition, a linear unit holding cost h_i is incurred at the end of Period t. Transportation costs consist of fuel consumption costs and driver wages per second r. Furthermore, there are refrigeration costs as the result of the amount of fuel used for cooling the vehicles while transportation and due to the heat entering while opening the vehicle doors to load or unload the items. We determine the replenishment periods, order-up-to levels, and shipment routes that minimise the total inventory, transportation, and refrigeration costs while producing minimal CO_2 emissions.

3.2.2 CO_2 Emissions

In this section, we explain the procedure that we apply to calculate the total fuel consumption and emissions due to transportation and refrigeration activities.

Transportation Emissions

We provide the formulation to estimate the amount of fuel used during transportation activities. We employ the same approach as in [SBRHVDV15] and [KSÇS23]. Total fuel consumption is estimated as follows:

$$f_t = \lambda\bigg(y(a/f) + \gamma\beta a f^2 + \gamma s(\mu + F)a\bigg), \tag{3.1}$$

where a is the distance (m), f is the constant speed (m/s), F is the load (kg), $\lambda = \xi/(\kappa\psi)$, $y = k_e N_e V_e$, $\gamma = 1/(1000\epsilon\omega)$, $\beta = 0.5 C_d A_e \rho$, $s = g\sin\phi + gC_r\cos\phi$ are the technical parameters ([BYS05]), k_e is the engine friction factor (kJ/rev/l), N_e is the engine speed (rev/s), V_e is the engine displacement (l), μ is the curb weight (kg), g is the constant of gravitation (9.81 m/s^2), ϕ is the road angle, C_d is the coefficient of aerodynamic drag, C_r is the coefficient of rolling resistance, A_e is the frontal surface area (m^2), ρ is the air density (kg/m^3), ϵ is the vehicle drive train efficiency, ω is the diesel engine efficiency parameter, ξ is the fuel to air mass ratio, κ is the heating value of a typical diesel fuel (kJ/g), and ψ is the conversion factor from (g/s) to (l/s). After we estimate the fuel consumption, we use a conversion factor u in terms of (kg/l) in order to calculate the amount of the CO_2 emission produced.

Refrigeration Emissions

We present the formulations to calculate the CO_2 emission produced during refrigeration activities. We apply the formulation described in [KSÇS23]. The energy used to maintain and control the temperature is dependent on the heat entering through vehicle surfaces and the heat entered inside the vehicle when the doors are opened for loading or unloading. Equation (3.2) calculates the total amount of heat (kWh) entering inside a vehicle in Period t through vehicle surfaces:

$$H_w = \frac{\sum_{(i,j)\in\mathcal{A}} a_{ij} X_{ijkt} U S_k \Delta T}{3.6 * 10^6 f}. \tag{3.2}$$

We divide the power by $3.6 * 10^6$ to convert Joules (J) to kilowatt-hours (kWh). a_{ij} is the distance between retailers i and j (m), U is the heat transfer coefficient (W/m^2/°K), S_k is the surface area of vehicle k (m^2), and ΔT is the temperature difference (°K) between the inside and the outside of the vehicle. Equation (3.3) calculates the total amount of heat enters inside the vehicle through the

opened doors during loading or unloading activities:

$$H_s = \sum_{(i,j)\in\mathcal{A}} H_i X_{ijkt}, \tag{3.3}$$

where H_i is the heat (kWh) entering inside the vehicle through the opened doors at the location of retailer i. Then the amount of fuel consumed for refrigeration, denoted by f_r, is calculated as follows:

$$f_r = \frac{H_w + H_s}{\mu_e \mu_p P_f},$$

where μ_e denotes the efficiency for converting the chemical energy from the fuel to electricity to supply power to the refrigeration system, μ_p is the coefficient of performance, and P_f is the energy content of the fuel (kWh/l). Finally, by using the conversion factor u, we calculate the total CO_2 emission produced while refrigerating the vehicles.

3.2.3 Soysal et al.'s (R, Q) Model

Before introducing our proposed (R, S) model, we hereby provide the (R, Q) model developed by [SBRHVDV15] in order to find the optimal ordering parameters R and Q which correspond to the timing and the quantity of replenishment orders. They consider four different variations for the problem: Perishable items with explicit fuel consumptions, perishable items without explicit fuel consumptions, non-perishable items with explicit fuel consumptions, and non-perishable items without explicit fuel consumptions. They calculate the fuel consumptions in 2 different ways. In the explicit fuel consuption version, they apply several formulations from previous studies which we have introduced in Equation 3.1. In the other version, they calculate the fuel consumption in a much simpler way and based on only traveled distance. Since our study does not include perishability, we consider the third variation of their problem, the non-perishable version with explicit fuel consumption. Moreover, they did not include refrigeration, and we add the refrigeration component to their model as well for our numerical study and comparisons. Unlike [SBRHVDV15]'s study, we also assume there are fixed setup and variable linear ordering costs and modify their model with respect to the ordering cost parameters.

In the following, we give the modified version of the (R, Q) model they propose for the non-perishable inventory model with explicit fuel consumption calcu-

lations. We have added the refrigeration component and totally removed the waste component in order to make the model fit our study.

$$\min\ TC = \sum_{i\in\mathcal{V}'}\sum_{t\in\mathcal{T}} h_i I_{it}^{+} \tag{3.4a}$$

$$+ \sum_{(i,j)\in\mathcal{A}}\sum_{k\in\mathcal{K}}\sum_{t\in\mathcal{T}} \lambda\Big(y(a_{ij}/f)X_{ijkt} + \gamma\beta a_{ij} f^2 X_{ijkt} + \gamma s(\mu X_{ijkt} + F_{ijkt})a_{ij}\Big) l \tag{3.4b}$$

$$+ \sum_{(i,j)\in\mathcal{A}}\sum_{k\in\mathcal{K}}\sum_{t\in\mathcal{T}} (a_{ij}/f)X_{ijkt} r \tag{3.4c}$$

$$+ \sum_{(i,j)\in\mathcal{A}}\sum_{k\in\mathcal{K}}\sum_{t\in\mathcal{T}} f_r X_{ijkt} \tag{3.4d}$$

subject to

$$I_{i0} = 0, \quad i \in \mathcal{V}'. \tag{3.5}$$

$$I_{it} = \sum_{s\in 1,\dots,t, k\in\mathcal{K}'} Q_{iks} - \sum_{s\in 1,\dots,t} d_{is}, \quad i \in \mathcal{V}', t \in \mathcal{T}. \tag{3.6}$$

$$\sum_{s\in 1,\dots,t, k\in\mathcal{K}'} Q_{iks} \geq F^{-1}_{d_{i,t-s+1}+d_{i,t-s+2}+\dots+d_{i,t}}(\alpha), \quad i \in \mathcal{V}', t \in \mathcal{T}. \tag{3.7}$$

$$\sum_{i\in\mathcal{V}, i\neq j} X_{ijkt} = \sum_{i\in\mathcal{V}, i\neq j} X_{jikt}, \quad j \in \mathcal{V}', k \in \mathcal{K}, t \in \mathcal{T}. \tag{3.8}$$

$$\sum_{j\in\mathcal{V}, i\neq j} X_{ijkt} \leq 1, \quad i \in \mathcal{V}, k \in \mathcal{K}, t \in \mathcal{T}. \tag{3.9}$$

$$\sum_{j\in\mathcal{V}, i\neq j} F_{ijkt} = \sum_{j\in\mathcal{V}, i\neq j} F_{jikt} - Q_{ikt}, \quad i \in \mathcal{V}', k \in \mathcal{K}, t \in \mathcal{T}. \tag{3.10}$$

$$F_{ijkt} \leq C X_{ijkt}, \quad (i,j) \in \mathcal{A}, k \in \mathcal{K}, t \in \mathcal{T}. \tag{3.11}$$

$$\sum_{j\in\mathcal{V}'} F_{j0kt} \leq 0, \quad k \in \mathcal{K}, t \in \mathcal{T}. \tag{3.12}$$

$$\sum_{(i,j)\in\mathcal{A}}\sum_{k\in\mathcal{K}}\sum_{t\in\mathcal{T}} \lambda\Big(y(a_{ij}/f)X_{ijkt} + \gamma\beta a f^2 X_{ijkt} + \gamma s(\mu X_{ijkt} + F_{ijkt})a_{ij}\Big) \geq 0. \tag{3.13}$$

$$\sum_{(i,j)\in\mathcal{A}}\sum_{k\in\mathcal{K}}\sum_{t\in\mathcal{T}} f_r X_{ijkt} \geq 0. \tag{3.14}$$

Note that $(x)^+ = \max(x, 0)$ and $(x)^- = \min(x, 0)$.

The objective function given in Equations (3.4a)-(3.4d) is composed of four

parts: Equations (3.4a), (3.4b), (3.4c), and (3.4d) minimises the total inventory, transportation, routing, and refrigeration costs throughout the planning horizon, respectively. Constraints (3.5)-(3.6) are related to inventory decisions. Constraint (3.5) initialises the inventory levels at the beginning of the planning horizon to zero. Constraint (3.6) is the inventory balance equation. Constraint (3.7) is the service level constraint. Note that $F^{-1}_{d_{i,t-s+1}+d_{i,t-s+2}+\ldots+d_{i,t}}$ is the inverse cumulative distribution function of the total demand for retailer i as of Period t since the last replenishment in Period $t-s+1$. Constraints (3.8)-(3.12) are related to routing and flow control. In particular, constraint (3.8) ensures the flow conservation between nodes. According to constraint (3.9), each vehicle can be assigned to at most one route in each period. Constraint (3.10) states that the total load decreases after each retailer is visited. According to constraint (3.11), the load on each vehicle cannot exceed the vehicle capacity, and constraint (3.12) is zero when returning to the distribution centre. Constraint (3.10) also performs as a subtour elimination constraint since the load is monotonically decreasing after each visit, see [BN09], [TNJ14], [SBRHVDV15]. Finally, constraints (3.13) and (3.14) enforce the total fuel consumption to be positive.

3.2.4 Proposed (R, S) Model

We hereby introduce the chance-constrained model that formulates our problem:

$$\min\ TC = \sum_{i\in\mathcal{V}'}\sum_{t\in\mathcal{T}}\Big(h_i I^+_{it} + a_i\delta_{it} + c_i\delta_{it}(S_{it} - I_{it-1})^+\Big) \tag{3.15a}$$

$$+ \sum_{(i,j)\in\mathcal{A}}\sum_{k\in\mathcal{K}}\sum_{t\in\mathcal{T}}\lambda\Big(y(a_{ij}/f)X_{ijkt} + \gamma\beta a_{ij}f^2X_{ijkt} + \gamma s(\mu X_{ijkt} + F_{ijkt})a_{ij}\Big)l \tag{3.15b}$$

$$+ \sum_{(i,j)\in\mathcal{A}}\sum_{k\in\mathcal{K}}\sum_{t\in\mathcal{T}}(a_{ij}/f)X_{ijkt}r \tag{3.15c}$$

$$+ \sum_{(i,j)\in\mathcal{A}}\sum_{k\in\mathcal{K}}\sum_{t\in\mathcal{T}}f_rX_{ijkt} \tag{3.15d}$$

subject to

$$I_{i0} = 0, \quad i\in\mathcal{V}'. \tag{3.16}$$

$$I_{it} = S_{it} - d_{it}, \quad i\in\mathcal{V}', t\in\mathcal{T}. \tag{3.17}$$

$$\sum_{k\in\mathcal{K}'}Q_{ikt} = S_{it} - I_{it-1}, \quad i\in\mathcal{V}', t\in\mathcal{T}. \tag{3.18}$$

$$S_{it} \geq I_{it-1}, \quad i \in \mathcal{V}', t \in \mathcal{T}. \tag{3.19}$$

$$S_{it} - I_{it-1} \leq \mathcal{M}\delta_{it}, \quad i \in \mathcal{V}', t \in \mathcal{T}. \tag{3.20}$$

$$I_{it} \geq \sum_{s=1}^{t} \left(F^{-1}_{d_{i,t-s+1}+d_{i,t-s+2}+\ldots+d_{i,t}}(\alpha) - \sum_{m=t-s+1}^{t} d_{im} \right) P_{its}, \quad i \in \mathcal{V}', t \in \mathcal{T}. \tag{3.21}$$

$$\sum_{s=1}^{t} P_{its} = 1, \quad i \in \mathcal{V}', t \in \mathcal{T}. \tag{3.22}$$

$$P_{its} \geq \delta_{i,t-s+1} - \sum_{m=t-s+2}^{t} \delta_{im}, \quad i \in \mathcal{V}', t \in \mathcal{T}, s = 1, ..., t. \tag{3.23}$$

$$\sum_{i \in \mathcal{V}, i \neq j} X_{ijkt} = \sum_{i \in \mathcal{V}, i \neq j} X_{jikt}, \quad j \in \mathcal{V}', k \in \mathcal{K}, t \in \mathcal{T}. \tag{3.24}$$

$$\sum_{j \in \mathcal{V}, i \neq j} X_{ijkt} \leq 1, \quad i \in \mathcal{V}, k \in \mathcal{K}, t \in \mathcal{T}. \tag{3.25}$$

$$\sum_{j \in \mathcal{V}, i \neq j} F_{ijkt} = \sum_{j \in \mathcal{V}, i \neq j} F_{jikt} - Q_{ikt}, \quad i \in \mathcal{V}', k \in \mathcal{K}, t \in \mathcal{T}. \tag{3.26}$$

$$F_{ijkt} \leq CX_{ijkt} - Q_{ikt}, \quad (i,j) \in \mathcal{A}, k \in \mathcal{K}, t \in \mathcal{T}. \tag{3.27}$$

$$\sum_{j \in \mathcal{V}'} F_{j0kt} \leq 0, \quad k \in \mathcal{K}, t \in \mathcal{T}. \tag{3.28}$$

$$\sum_{(i,j) \in \mathcal{A}} \sum_{k \in \mathcal{K}} \sum_{t \in \mathcal{T}} \lambda \Big(y(a_{ij}/f) X_{ijkt} + \gamma\beta a f^2 X_{ijkt} + \gamma s(\mu X_{ijkt} + F_{ijkt}) a_{ij} \Big) \geq 0. \tag{3.29}$$

$$\sum_{(i,j) \in \mathcal{A}} \sum_{k \in \mathcal{K}} \sum_{t \in \mathcal{T}} f_r X_{ijkt} \geq 0. \tag{3.30}$$

The objective function given in Equations (3.15a)-(3.15d) is composed of four parts: Equations (3.15a), (3.15b), (3.15c), and (3.15d) minimises the total inventory, transportation, routing, and refrigeration costs throughout the planning horizon, respectively. Constraints (3.16)-(3.23) are related to inventory decisions. Constraint (3.16) initialises the inventory levels at the beginning of the planning horizon to zero. Constraint (3.17) is the inventory balance equation and reflects that the expected inventory level for each customer at the end of each period is equivalent to the expected inventory level at the beginning of the period subtracting the expected demands in that period. The expected inventory level at the beginning of periods could be interpreted as order-up-to-levels or the level that the inventory could be raised to at the beginning of periods. If there is no replenishment in that period, S_{it} is considered as the opening stock level. Constraint (3.18) states that in each Period t, the sum of all the amount of order for each retailer i carried by the operating vehicles must

be equal to the order-up-to level for that retailer. Constraints (3.19) and (3.20) ensures if there is no replenishment in Period t for retailer i, the order-up-to level is equal to the opening inventory level. Constraint (3.21) is the service level constraint. Note that $F^{-1}_{d_{i,t-s+1}+d_{i,t-s+2}+\ldots+d_{i,t}}$ is the inverse cumulative distribution function of the total demand for retailer i as of Period t since the last replenishment in Period $t-s+1$. Constraints (3.22) and (3.23) track the last replenishment. Constraints (3.24)-(3.28) are related to routing and flow control. In particular, constraint (3.24) ensures the flow conservation between nodes. According to constraint (3.25), each vehicle can be assigned to at most one route in each period. Constraint (3.26) states that the total load decreases after each retailer is visited. According to constraint (3.27), the load on each vehicle cannot exceed the vehicle capacity, and constraint (3.28) ensures the total load on each vehicle cannot be non-negative when returning to the distribution centre. Constraint (3.26) also performs as a subtour elimination constraint since the load is monotonically decreasing after each visit, see [BN09], [TNJ14], [SBRHVDV15]. Finally, constraints (3.29) and (3.30) enforce the total fuel consumption to be non-negative. Recall that f_r in Eq. (3.30) represents the fuel usage for refrigeration and the notation is the same as in [KSÇS23].

Different from the (R, Q) formulation presented in Section 3.2.3, we apply [TK04]'s formulation in this model. We define inventory balance constraints (3.17)-(3.20) in terms of the order-up-to level S_{it}. In addition, we employ [TK04]'s service level and replenishment tracking constraints, (3.21) and (3.22)-(3.23), respectively.

3.2.5 Deterministic-Equivalent Approximation

We find the replenishment schedule before any demand realisation occurs. Therefore, we consider a deterministic-equivalent approximation to the model. We replace the stochastic variables S_{it}, I_{it}, Q_{ikt}, and d_t with their expectations $\mathbb{E}(S_{it})$, $\mathbb{E}(I_{it})$, $\mathbb{E}(Q_{ikt})$, and $\mathbb{E}(d_t)$, respectively. Then, the objective function in

terms of the expectations is rewritten as follows:

$$\min\mathbb{E}(TC) = \sum_{i\in\mathcal{V}'}\sum_{t\in\mathcal{T}}\left(h_i\mathbb{E}(I_{it}^{+}) + A_i\delta_{it} + c_i\delta_{it}\Big(\mathbb{E}(S_{it}) - \mathbb{E}(I_{it-1})\Big)^{+}\right) \tag{3.31a}$$

$$+ \sum_{(i,j)\in\mathcal{A}}\sum_{k\in\mathcal{K}}\sum_{t\in\mathcal{T}}\lambda\Bigg(y(a_{ij}/f)X_{ijkt} + \gamma\beta af^2X_{ijkt} + \gamma s(\mu X_{ijkt} + F_{ijkt})a_{ij}\Bigg)l \tag{3.31b}$$

$$+ \sum_{(i,j)\in\mathcal{A}}\sum_{k\in\mathcal{K}}\sum_{t\in\mathcal{T}}(a_{ij}/f)X_{ijkt}r \tag{3.31c}$$

$$+ \sum_{(i,j)\in\mathcal{A}}\sum_{k\in\mathcal{K}}\sum_{t\in\mathcal{T}}f_rX_{ijkt} \tag{3.31d}$$

Each part of the deterministic-equivalent objective function defined in Equations 3.31a-3.31d correspond to the deterministic-equivalent version of Equations (3.15a), (3.15b), (3.15c), and (3.15d), respectively. We finally rewrite constraints (3.16)-(3.21) and (3.26)-(3.27) by replacing the stochastic variables with their expectations, respectively:

$$\mathbb{E}(I_{i0}) = 0, \quad i \in \mathcal{V}'.$$

$$\mathbb{E}(I_{it}) = \mathbb{E}(S_{it}) - \mathbb{E}(d_{it}), \quad i \in \mathcal{V}', t \in \mathcal{T}.$$

$$\sum_{k\in\mathcal{K}'}\mathbb{E}(Q_{ikt}) = \mathbb{E}(S_{it}) - \mathbb{E}(I_{it-1}), \quad i \in \mathcal{V}', t \in \mathcal{T}.$$

$$\mathbb{E}(S_{it}) \geq \mathbb{E}(I_{it-1}), \quad i \in \mathcal{V}', t \in \mathcal{T}.$$

$$\mathbb{E}(S_{it}) - \mathbb{E}(I_{it-1}) \leq \mathcal{M}\delta_{it}, \quad i \in \mathcal{V}', t \in \mathcal{T}.$$

$$\mathbb{E}(I_{it}) \geq \sum_{s=1}^{t}\left(G^{-1}_{d_{i,t-s+1}+d_{i,t-s+2}+\ldots+d_{i,t}}(\alpha) - \sum_{m=t-s+1}^{t}\mathbb{E}(d_{im})\right)P_{its}, \quad i \in \mathcal{V}', t \in \mathcal{T}.$$

$$\sum_{j\in\mathcal{V},i\neq j}F_{ijkt} = \sum_{j\in\mathcal{V},i\neq j}F_{jikt} - \mathbb{E}(Q_{ikt}), \quad i \in \mathcal{V}', k \in \mathcal{K}, t \in \mathcal{T}.$$

$$F_{ijkt} \leq CX_{ijkt} - \mathbb{E}(Q_{ikt}), \quad (i,j) \in \mathcal{A}, k \in \mathcal{K}, t \in \mathcal{T}.$$

3.3 Numerical Experiments

In this section we discuss the numerical experiments that we conducted to test and evaluate our model. The models we compare in these experiments are given below:

- **RQ:** The modified version of the (R, Q) model proposed by [SBRHVDV15]

for the non-perishability case, and with the extensions of the refrigeration costs and emissions and

- **RS:** Our proposed (R, S) model in this study.

We implemented both models using IBM ILOG CPLEX 22.1.0 software. We then performed simulation runs of 10,000 replications using MATLAB to find the order quantities and to measure the service levels that we achieved. We conducted our numerical experiments on a Macbook Pro with an Apple M1 Max 10 core CPU and a 64 GB RAM.

3.3.1 Experimental Design

We selected a similar experimental configuration to [SBRHVDV15]. We consider different combinations of demand patterns, fixed setup and unit ordering costs, service levels, coefficient of variations, fuel prices, and vehicle speeds in our testbed. In total, there are 146 instances. The fixed setup cost $A \in \{10, 25, 100\}$ and the unit ordering cost $c \in \{0.05, 0.1\}$. The service levels $\alpha \in \{0.95, 0.975\}$. There are $K = 2$ identical vehicles with a capacity of $C = 4000$ kg which operate between $V = 4$ retailers and a distribution centre in a finite planning horizon of $T = 8$ periods. The fuel price is $l = 1.7$ Euros per liter and all the vehicles has an average speed of $f = 80$ km/h. We also test the base case for $l = 1.2$ and $f = 40$. We then test our model on a larger instance in which $V = 11$, $K = 2$, and $C = 15000$. All these parameter values are summarised in Table 3.1. Moreover, we run the numerical experiments for 4 normally distributed different demand sets with the coefficients of variations taking values from the set $cv \in \{0.05, 0.1, 0.2\}$. There are random, seasonal, cyclic, and stationary demand distribution patterns for each retailer, which are referred to as the demand set 1, 2, 3, and 4, respectively. In order to construct our demand set 1, we take the base case demand set of of [SBRHVDV15] for the first 4 periods and the demand set 1 of [SBRHVDV15] for the last 4 periods of our planning horizon. Since our aim is to observe the differences between the performances of RQ and RS in a long enough planning horizon and a large enough retailer set rather than in a larger retailer set, and in order to speed up the computation, we take $V = 4$. The computational requirements increasingly grows as V gets larger. Therefore, we consider a subset of their retailer set and only take the first four retailers for our base case study and sensitivity analysis. We then sort the same demand means to obtain our demand sets 2, 3, and 4. Table 3.2 summarises the demand distribution means for each demand

Table 3.1: Input parameters for the experiments.

Input parameters	**Values**
Fixed setup cost (a)	{10, 25, 100}
Unit ordering cost (c)	{0.05, 0.1}
Unit holding cost (h)	0.06
Service level (α)	{0.95, 0.975}
Coef. of variation	{0.05, 0.1, 0.2}
Vehicle speed (f)	80 km/h
Number of vehicles (K)	2
Vehicle capacity (C)	4 tonnes
Planning horizon (T)	8

pattern, retailer, and period. Finally, we perform another numerical experiment by running our model for different planning horizon lengths to analyse the relationship between the planning horizon length and the cumulative safety stock differences between RS and RQ. In this study we consider the same vehicle and emission parameter settings as of [SBRHVDV15] and the same refrigeration parameter settings as of [KSÇS23]. Table 3.3 illustrates the values for the vehicle and emission parameters, and Table 3.4 illustrates the refrigeration parameter values, respectively. Table 3.5 shows the distances between the distribution centre and each retailer. We use the same distance set as in [SBRHVDV15] in our study. Note that we consider the retailers 5-11 only in the extended case.

3.3.2 Base Case Results

We present the results of the base case scenario and compare the performance of RS with RQ, in terms of total cost, CO_2 emissions, vehicle routes, service levels, and cumulative safety stock levels. As the base case of our study, we consider the scenario in which the demand pattern is random, the fixed setup cost is $a = 25$, the unit ordering cost is $c = 0.1$, the service level is $\alpha = 0.95$, and the coefficient of variation is $cv = 0.1$.

Table 3.6 summarises the base case optimisation results. The routing plan proposed by RS has a 32.27 hours of total driving time while that of RQ, 34.58 hours. Due to the constant vehicle speed, the routing policy of RS results in a lesser amount of total travel distance. Therefore, RS results in a lower total fuel cost due to the transportation activities, 808.20, in comparison to 861.37 found by RQ. Similarly, the total wage paid to the drivers is less in RS than in RQ, 348.48 and 373.42, respectively. The lesser amount of the total traveling

Table 3.2: Demand patterns and parameters.

Demand Patterns	Retailers	Period							
		1	2	3	4	5	6	7	8
Random	1	900	400	1000	600	500	1400	1300	300
	2	1400	1200	1700	1200	800	1100	1500	1800
	3	500	500	1250	600	1300	600	1000	1900
	4	1050	900	1500	1100	800	700	900	900
Seasonal	1	400	600	900	1000	1400	1300	500	300
	2	1200	1200	1400	1700	1800	1500	1100	800
	3	500	500	600	1250	1900	1300	1000	600
	4	400	500	1100	2500	2200	1600	800	600
Cyclic	1	400	900	1000	600	500	1400	1300	300
	2	1200	1400	1700	1200	800	1500	1800	1100
	3	500	600	1250	500	600	1300	1900	1000
	4	500	2500	1100	400	800	2200	1600	600
Stationary	1	800	800	800	800	800	800	800	800
	2	1350	1350	1350	1350	1350	1350	1350	1350
	3	950	950	950	950	950	950	950	950
	4	1200	1200	1200	1200	1200	1200	1200	1200

Table 3.3: Vehicle and emission parameters.

Notation	Description	Value
u	Emission conversion factor (kg/l)	2.63
ξ	Fuel-to-air mass ratio	1
κ	Heating value of a typical diesel fuel (kJ/g)	44
ψ	Conversion factor (g/l)	737
k_e	Engine friction factor (kJ/rev/l)	0.2
N_e	Engine speed (rev/s)	33
V_e	Engine displacement (l)	5
ρ	Air density (kg/m^3)	1.2041
A_e	Frontal surface area (m^2)	3.912
μ	Curb-weight (kg)	6350
g	Gravitational constant (m/s^2)	9.81
Φ	Road angle	0
C_d	Coefficient of aero dynamic drag	0.7
C_r	Coefficient of rolling resistance	0.01
ϵ	Vehicle drive train efficiency	0.4
ω	Efficiency parameter for diesel engines	0.9
l	Fuel price per liter (€)	1.7
r	Driver wage (€/s)	0.003

Table 3.4: Refrigeration parameters.

Notation	Description	Value
S_k	Surface area for the vehicle k (m^2)	165
ΔT	Temperature difference (°K)	18
U	Heat transfer coefficient (W/m^2/°K)	0.7
μ_e	The chemical to refrigeration energy conversion efficiency	0.3
μ_p	Coefficient of performance	0.67
P_f	Energy content of the fuel (kWh/l)	8.8
H_i	Heat entering during service time at stop i (kWh)	4

distance reduces the total fuel consumption to refrigerate the vehicles, and the refrigeration cost incurred by RS is 165.38 which is less than that of 172.73 incurred by RQ. The total routing cost consists of the transportation, refrigeration and wage costs, so RS leads to a lower amount of total routing cost compared to RQ which are 1322.06 and 1407.52. Moreover, since CO_2 emissions occur as a result of fuel consumption, it is directly related to the total fuel cost. Therefore, our RS model generates a lesser amount of CO_2 emission of 1506.19 kg in comparison to 1599.81 kg of RQ. The deterministic-equivalent solution results in a total inventory cost of 4923.38 and 5341.30 according to RS and RQ, respectively. Thus, the total deterministic-equivalent cost is 6245.44 for RS and

Table 3.5: Distances between the nodes (km). Retailers 5-11 are used in the extended case.

	DC	1	2	3	4	5	6	7	8	9	10	11
DC	0	67	89.2	126	78.1	70.6	106	66.3	64.4	156	151	35.5
1	73.2	0	154	191	143	144	141	101	74.3	166	176	61.5
2	70.8	136	0	65.9	62.9	113	158	118	126	218	212	97.2
3	126	192	69.5	0	98.9	171	233	193	182	274	268	153
4	78.4	144	63.2	99.7	0	123	185	145	134	226	220	105
5	70.9	144	105	163	115	0	50.2	58.4	120	155	220	105
6	106	131	161	222	175	50.9	0	41.6	75.3	105	199	84.4
7	66.5	91.2	121	182	135	58.2	40.1	0	35.3	117	159	44.4
8	67.4	149	185	137	92.7	0	74.4	34.5	0	92.1	120	34.8
9	158	166	239	276	228	155	106	116	92.4	0	69.6	126
10	150	176	232	268	220	221	192	152	119	70	0	119
11	35	60.3	116	153	105	106	83.6	43.7	30.6	123	118	0

6748.82 for RQ.

Table 3.7 presents a summary of the simulation results. According to Table 3.7, the average inventory cost over 10,000 replications is 4939.32 for RS and 5339.88 for RQ, which are very close to the deterministic-equivalent inventory cost values. Notice that even though all the routing decisions are predetermined and are not affected by random demand realisations, there is a slight difference of 0.11 in total routing costs between the optimisation and simulation results. This is due to the amount of vehicle load depends on order sizes and the fuel consumption is affected by the amount of vehicle load. In Table 3.10, the achieved service levels for each retailer and in each period are given, and both RS and RQ are sufficient to meet the 95% service level requirement.

The resulting ordering policy found by RS is given in Table 3.8. In the Replenishment Periods section, 1 indicates there is a replenishment order and 0 states that there is no ordering. For instance, for Retailer 1 the replenishment periods are 1, 3, 6, and 8. At the beginning of each replenishment Period t for each retailer i, the actual inventory starting inventory level I_{it-1} is observed and an amount of order that is sufficient to meet the order-up-to level S_{it} is placed. If t is not a replenishment period, even if I_{it-1} is less than S_{it} no order is placed. Similarly, if there is sufficient inventory to meet S_{it} at the beginning of replenishment Period t no order placement occurs. Unlike RQ and as a result of the static-dynamic strategy, there is a flexibility in ordering decisions such that order quantities depend on the actual inventory level and are determined after the demand realisations. Table 3.9 visualises the order quantities and the ending inventory levels for the base case and with respect to both RS and RQ.

In Table 3.11, the assigned routes for each vehicle and the number of assigned vehicles in each period with respect to RS and RQ are given. Except in periods 1, 3, and 7, both RS and RQ propose the same routes even though the assigned vehicles might differ. While RS suggests employing only Vehicle 2 in period 7, RQ suggests employing both vehicles. In both RS and RQ, the same number of vehicles are employed for all the other periods.

Table 3.12 shows that RQ yields a greater number of cumulative order quantities and ending inventory levels compared to RS due to its static nature. Thus the unit ordering and inventory holding costs are higher in RQ. Moreover, since the amount of vehicle load affects the fuel consumption it is possible that the larger amount of order size might increase the fuel consumption and the total

CO_2 emission. On the contrary, the flexibility in order size decisions in RS allows lower number of cumulative order quantities and ending inventory levels, hence lower inventory costs. We will provide a detailed analysis on the relationship between the planning horizon length and the cumulative ending inventory levels and order quantities in Section 3.3.5.

3.3.3 Performance of the (R, S) Model

We present the results of the extensive sensitivity analysis to compare the performance of RS with RQ under various parameter settings and scenarios. Table 3.13 summarises the average percentage increases of RQ above RS, e.g., $100(RQ/RS-1)$, in terms of costs, cumulative order sizes and ending inventory levels, and CO_2 emissions. We observe from Table 3.13 that in all the instances, RS has a significant advantage over RQ in terms of total cost. The average total cost of RQ is 9.35% worse than that of RS with a 95% confidence interval of [8.43, 10.26]. Although the average total routing costs for both models are similar, the average inventory cost is 12.55% higher for RQ than that of RS, as a result of higher inventory holding and variable ordering costs caused by the accumulated ending inventory and larger order sizes. This observation explains why RS performs better than RQ in terms of total cost. On average, the cumulative ending inventory levels in RQ are 52.76% greater than in RS and the cumulative order quantities are found to be 4.82% greater in RQ compared to in RS. In other words, the inventory holding costs and the total variable ordering costs are 52.76% and 4.82% greater in RQ than in RS. The amount of CO_2 emission is in line with fuel consumption, therefore both models generate similar levels of emission and have similar environmental impact. Even though RQ generates 0.54% more CO_2 than RS, the 95% confidence interval is [-0.19, 1.27], and therefore we cannot conclude that RS has less environmental impact, or vice versa.

The performance of RS over RQ is similar for all the 4 demand sets in terms of routing costs. However, RS performs better than RQ in terms of total cost and inventory costs with similar levels of percentage differences for all the demand patterns.

In case of the cyclic demand pattern, RS tends to have less environmental impact. For all the other demand patterns, the environmental impact is similar. The effect of α on the general performance of RS over RQ is similar between each α values, and also similar to the overall performance results. RS performs

Table 3.6: Summary of the base case optimisation results.

Model	Emission (kg)	Driving Time	Transportation Fuel Cost	Refrigeration Cost	Total Wage	Total Routing Cost	Inventory Cost	Total Cost
(R, S)	1506.19	32.27	808.20	165.38	348.48	1322.06	4923.38	6245.44
(R, Q)	1599.81	34.58	861.37	172.73	373.42	1407.52	5341.30	6748.82

Table 3.7: Summary of the base case simulation results, 10,000 reps.

Model	Total Routing Cost	Avg Inventory Cost	Avg Total Cost	95% CI for Avg Total Cost
(R, S)	1321.95	4939.32	6261.27	[6260.61, 6261.94]
(R, Q)	1407.52	5339.88	6747.40	[6743.45, 6751.35]

Table 3.8: The replenishment plan and the order-up-to levels for the base case.

	Period							
Retailer	1	2	3	4	5	6	7	8
	Replenishment Periods							
1	1	0	1	0	0	1	0	1
2	1	1	1	1	0	1	1	1
3	1	1	1	0	1	0	1	1
4	1	1	1	1	1	1	1	0
	Order-Up-To Levels							
1	1462	562	2309	1309	709	3049	1649	349
2	1630	1397	1980	2237	1037	1281	1747	2096
3	1082	582	2078	828	2136	836	1165	2213
4	1281	2911	582	466	1863	2562	1565	765

better than RQ in terms of total cost and inventory cost for all the coefficient of variation values. As the coefficient of variation increases, RS has a tendency to generate less CO_2 emission and to incur less routing cost compared to RQ. For $cv = 0.2$, RS leads to produce less emission than RQ, and incurs less number of total routing cost. Moreover, as the coefficient of variation increases, the percentage cumulative ending inventory and order size increases of RQ above RS gets larger, so RS proposes a much better solution in terms of order quantities and ending inventory levels, and inventory costs.

The average percentage difference in total cost increases from 4.18% and to 15.54%, yielding a much better total cost improvement since the confidence intervals for each coefficient of variation do not overlap each other. In contrast to the coefficient of variation, as the fixed setup cost a increases the percentage differences of RQ over RS tend to decrease. Despite the total cost incurred by RS still being lower than by RQ, reducing a results in much better cost improvement and lower number of accumulated ending inventory levels and ordered items. While RS generates less CO_2 emission for $a = 10$, both models tend to generate similar levels of emission for the larger fixed setup cost values. As the unit ordering cost c increases, the percentage total cost increase of RQ above RS tend to decrease, but we cannot conclude that it decreases since the confidence intervals overlap. For both the unit ordering cost parameters $c = 0.05$ and $c = 0.1$, the total cost and the order quantities are lower in RS than in RQ. As c increases, the changes in the percentage increases wo not be significant.

Table 3.9: The order quantities and the ending inventory levels for the base case.

		Order Quantities Period								Ending Inventory Levels Period								
Policy	**Retailer**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**
(R,S)	1	1462	0	2147	0	0	1487	1663	0	0	562	162	1309	709	209	1649	349	49
	2	1630	1167	1783	1957	0	1044	1566	1849	0	230	197	280	1037	237	181	247	296
	3	1082	0	1996	0	1908	0	929	2048	0	582	82	828	228	836	236	165	313
	4	1281	2730	171	384	1797	2299	1203	0	0	181	411	82	66	263	362	765	165
(R,Q)	1	1462	0	2202	0	0	1487	1663	0	0	562	162	1364	764	264	351	714	414
	2	2179	724	1810	2063	0	1148	1542	1873	0	779	303	413	1276	476	524	566	639
	3	1116	0	1990	0	1992	0	1037	2011	0	616	116	856	256	948	348	385	496
	4	1281	3276	0	404	1670	2312	1421	0	0	181	957	457	461	531	643	1264	664

Table 3.10: Achieved service levels (%), $\alpha = 0.95$, 10,000 reps.

		α-Service Levels							
		Period							
Policy	**Retailer**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**
(R,S)	1	1.0000	0.9493	1.0000	1.0000	0.9489	1.0000	0.9655	0.9740
	2	0.9512	0.9469	0.9486	1.0000	0.9496	0.9495	0.9526	0.9513
	3	1.0000	0.9662	1.0000	0.9507	1.0000	0.9501	0.9491	0.9521
	4	0.9532	0.9494	0.9597	0.9527	0.9492	0.9531	1.0000	0.9529
(R,Q)	1	1.0000	0.9493	1.0000	1.0000	0.9460	0.9479	0.9973	0.9468
	2	1.0000	0.9469	0.9476	1.0000	0.9462	0.9514	0.9480	0.9478
	3	1.0000	0.9462	1.0000	0.9486	1.0000	0.9510	0.9494	0.9497
	4	0.9532	1.0000	0.9482	0.9470	0.9469	0.9479	0.9994	0.9505

Table 3.11: Assigned routes for each vehicle and the number of assigned vehicles.

		Routes							
		Period							
Policy	**Vehicle**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**
(R,S)	V1	0-1-0	-	0-1-0	-	0-4-3-0	0-4-2-0	-	-
	V2	0-4-3-2-0	0-4-2-0	0-4-3-2-0	0-4-2-0	-	0-1-0	0-4-3-2-0	0-3-2-0
(R,Q)	V1	0-1-4-0	-	0-1-0	-	0-4-3-0	0-4-2-0	0-4-3-2-0	0-3-2-0
	V2	0-2-3-0	0-4-2-0	0-3-2-0	0-4-2-0	-	0-1-0	0-1-0	-
No of Vehicles vs Policy	(R,S)	2	1	2	1	1	2	1	1
	(R,Q)	2	1	2	1	1	2	2	1

Table 3.12: Comparison of the cumulative order quantities and ending inventory levels

	Period							
Policy	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**
	Cumulative Order Quantity							
(R,S)	5455	9352	15449	17790	21495	27678	31376	35273
(R,Q)	6038	10038	16040	18507	22169	27116	32779	36663
	Cumulative Ending Inventory							
(R,S)	1555	2407	4906	6946	8491	10919	12445	13268
(R,Q)	2138	3676	6766	9523	11742	13608	16537	18750

Finally, decreasing the fuel price to $l = 1.2$ and the vehicle speed to $f = 40$ leads to poorer RS performance over RQ in terms of the average routing costs and the emissions, compared to the base case.

Note that due to higher holding costs incurring in RQ, RS outperforms all instances by total cost. However, since RQ is based on the static uncertainty strategy of [BT88], it might be more suitable for such cases in which a considerable preparation is required before the time horizon begins, see [BT88]. In addition, since RS outperforms RQ in terms of reducing the excess ending inventory and the inventory costs, there is no threshold or certain critical values that would change the superiority of RS over RQ.

Table 3.13: Avg percentage increases of RQ above RS ($e.g. 100 * (RQ/RS - 1)$). All the other parameter values of the instances marked with (*) are the same as of the base case.

Parameter	Avg Emission	Avg Emission 95% CI	Avg Cum Order Quantity	Avg Cum Ending Inventory	Avg Cum End Inv + Ord Q	Avg Inv Cost	Avg Transport Fuel Cost	Avg Refrg Fuel Cost	Avg Total Wage	Avg Total Routing Cost	Avg Total Cost	Avg Total Cost 95% CI
Overall	0.54	[-0.19 1.27]	4.82	52.76	18.84	12.55	0.63	0.12	0.49	0.52	9.35	[8.43, 10.26]
Random	1.45	[-0.22, 3.12]	4.46	51.96	19.27	11.88	1.55	1.00	1.66	1.51	9.41	[7.49, 11.34]
Cyclic	1.67	[0.21, 3.13]	5.66	52.16	19.12	13.04	1.92	0.42	1.79	1.70	9.99	[8.06, 11.93]
Seasonal	-0.79	[-2.48, 0.90]	4.87	52.29	18.53	13.12	-1.00	0.33	-1.12	-0.88	9.31	[7.49, 11.26]
Stationary	-0.19	[-0.81, 0.43]	4.29	54.63	18.42	12.16	0.03	-1.27	-0.36	-0.23	8.68	[7.08, 10.28]
$\alpha = 0.95$	0.19	[-0.75, 1.13]	4.40	51.13	17.61	11.66	0.28	-0.21	0.17	0.18	8.58	[7.40, 9.76]
$\alpha = 0.975$	0.88	[-0.24, 2.00]	5.25	54.39	20.07	13.43	0.97	0.45	0.82	0.86	10.11	[8.74, 11.49]
$cv = 0.05$	-1.24	[-2.50, 0.02]	2.05	38.91	9.64	6.46	-1.24	-1.18	-1.35	-1.27	4.18	[3.81, 4.55]
$cv = 0.1$	0.79	[-0.21, 1.79]	4.17	50.30	16.87	11.12	0.89	0.35	0.92	0.82	8.33	[7.59, 9.07]
$cv = 0.2$	2.06	[0.72, 3.40]	8.25	69.10	29.91	20.06	2.23	1.19	1.91	2.02	15.54	[14.28, 16.79]
$a = 10$	1.71	[0.61, 2,81]	5.31	65.02	20.41	16.35	1.79	1.38	1.64	1.69	11.83	[10.10, 13.56]
$a = 25$	1.16	[-0.33, 2.66]	5.14	58.91	19.74	13.75	1.17	1.16	1.09	1.14	10.24	[8.73, 11.74]
$a = 100$	-1.27	[-2.27, 0.27]	4.02	34.35	16.36	7.54	-1.08	-2.18	-1.25	-1.26	5.98	[5.07, 6.89]
$c = 0.05$	0.47	[-0.53, 1.48]	4.79	52.29	18.69	14.23	0.56	0.08	0.42	0.46	10.07	[8.68, 11.46]
$c = 0.1$	0.60	[-0.47, 1.67]	4.85	53.23	18.98	10.86	0.69	0.16	0.57	0.59	8.62	[7.45, 9.80]
Base Case	6.22	-	3.71	39.71	13.61	8.11	6.58	4.44	7.16	6.46	7.76	-
$l = 1.2$*	-4.77	-	3.78	102.41	23.72	11.18	-4.08	-8.17	-3.75	-4.43	8.07	-
$f = 40$*	-2.03	-	3.71	63.55	20.16	10.47	-1.08	-5.66	-0.86	-1.54	7.42	-

Table 3.14: Demand parameters for the extended case.

Retailers	Periods			
	1	**2**	**3**	**4**
1	900	400	100	600
2	1400	1200	1700	1200
3	500	500	1250	600
4	1100	2500	500	400
5	1050	900	1500	1100
6	1200	500	400	1400
7	800	700	500	500
8	1900	400	300	1300
9	800	400	700	1300
10	1100	1600	400	300
11	2600	3200	2500	3200

3.3.4 Modified Larger Case Study

We present the results of the modified larger case study. In order to test the performance of our model for larger network setting, we increase the number of retailers to $V = 11$. For this case, we use the demand pattern given in Table 3.14 and we extend the distance matrix provided in Table 3.5 so as to include Retailers 5-11. The total load on all vehicles might increase as V or T grows, therefore we need to increase either K or C. For convenience, we employ $K = 2$ vehicles with a capacity of $C = 15000$ and set the time horizon as $T = 4$ periods. The demand pattern and the retailers provided in Table 3.14 are the same as the base case demand set of [SBRHVDV15].

Increasing V while keeping the total capacity the same greatly increase the size (number of constraints, decision variables, etc) of the model. Increasing K without changing C has no effect on the routes, costs, etc, and many vehicles would stay intact. Moreover, in the larger case we obtained similar results to the base case.

Tables 3.15 and 3.16 give the deterministic-equivalent optimisation results and the routing decisions for the larger case, respectively. In general, the performance of RS over RQ is similar to the base case. As seen from Table 3.16, both models propose the same routes and employs the same number of vehicles, although the assigned vehicles for a particular route might be different. Hence, both models offer the same total driving time, refrigeration cost, and total wage. There is a minor difference between the transportation fuel costs caused by the amount of vehicle load since RQ has larger order quantities compared to RS, as provided in Table 3.17. In addition, the number of cumulative ending inventory levels is about 8.81% greater in RQ than in RS, and it is less than the base case in which the percentage difference is 41.31%.

Table 3.15: Summary of the extended case optimisation results.

Model	Emission (kg)	Driving Time (h)	Transport Fuel Cost	Refr Cost	Total Wage	Total Routing Cost	Total Inv Cost	Total Cost
(R,S)	1363.10	26.99	748.24	132.84	291.49	1172.57	5286.72	6459.29
(R,Q)	1370.80	26.99	753.02	132.84	291.49	1177.35	5558.83	6736.18

Table 3.16: Assigned routes for each vehicle and the number of assigned vehicles for the extended case.

		Routes			
		Periods			
Policy	Vehicle	1	2	3	4
(R,S)	*V1*	0-1-8-9-10-0	0-11-4-2-0	0-1-11-7-5-2-3-0	-
	V2	0-11-7-6-5-2-3-4-0	-	-	0-11-8-9-6-5-0
(R,Q)	*V1*	0-11-7-6-5-2-3-4-0	-	-	-
	V2	0-1-8-9-10-0	0-11-4-2-0	0-1-11-7-5-2-3-0	0-11-8-9-6-5-0
No of Vehicles vs Policy	(R,S)	2	1	1	1
	(R,Q)	2	1	1	1

Table 3.17: Comparison of the cumulative order quantities and ending inventory levels for the extended case.

	Period			
Policy	1	2	3	4
	Cumulative Order Quantity			
(R,S)	23152	30902	42385	50658
(R,Q)	23433	31836	43828	52591
	Cumulative Ending Inventory			
(R,S)	23152	65712	71697	74055
(R,Q)	23433	68860	76288	80579

3.3.5 Planning Horizon Length vs Cumulative Ending Inventory and Order Size

We analyse the relationship between the length of the planning horizon and the difference in the cumulative ending inventory levels of RS and RQ. In order to analyse that relationship, we extend the planning horizon to $T = 40$ periods. We consider the demand set 1 and modify it such that after every 8 periods the 8-period demand mean sequence is recurring. As seen from Figure 1, the increase rate of cumulative ending inventory levels for RQ is greater than that of RS. Therefore, as t increases the difference between the cumulative inventory levels for both models increases. Hence, the percentage differences increase. For instance, when $t = 8$, the cumulative ending inventory levels for RS and RQ are found to be 15684 and 20408, respectively, with a percentage difference of 30.12%. These values become 40014, 69381, and 73.39% for $t = 20$. At

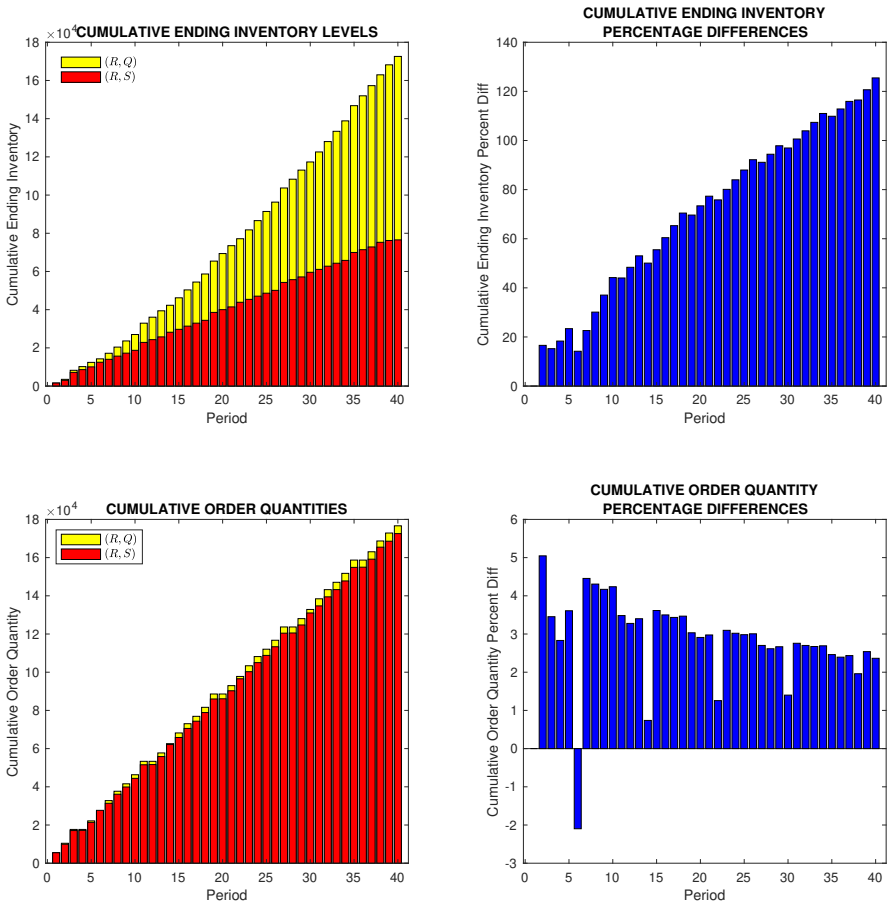

Figure 3.2: Comparison of the cumulative ending inventory levels and percentage differences found via the (R, Q) and (R, S) models.

the end of the planning horizon when $t = 40$, we obtain the cumulative ending inventory levels of 172595 for RQ and 76543 for RS, and a percentage difference of 125.49%. This phenomenon is the main factor that makes the model of [SBRHVDV15] find high inventory holding and total inventory costs compared to our (R, S) model. Therefore, we can conclude that the static-dynamic structure of RS helps avoiding placing unnecessary orders and holding unnecessary items in the inventory. On the contrary, even though RQ calculates larger order quantities than RS in most periods, the percentage differences in the order quantities between RQ and RS do not differ significantly, as seen in Figure 1. The percentage difference is 4.31% when $t = 8$, while it is 2.91% for $t = 20$ and 2.36% for $t = 40$.

In this study, a planning horizon with $T = 40$ periods has been taken to represent a long enough planning horizon in which the differences between the

ending inventory levels in RQ and RS are significantly large. Although a 40-period planning horizon being taken as more of a theoretical exercise, it also applies in medium-term tactical planning or long-term strategic planning, etc., see [OLFD16], depending on the cold supply chain structure.

3.4 Concluding Remarks and Managerial Insights

We have studied a cold supply chain model with refrigerated vehicles, multiple retailer, and a single distribution centre. We formulated the problem as an IRP model with the refrigeration cost component and which employs the (R, S) policy proposed by [TK04]. Then we compared the performance of our model with the refrigeration component added and no perishability version of [SBRHVDV15]'s IRP model which employs the (R, Q) policy. Our numerical experiments show that by applying the (R, S) policy instead of the (R, Q) policy, although we did not make a significant improvement to reduce CO_2 emissions and total routing costs, we reduced inventory costs significantly. Since the majority of the incurred costs consists of inventory costs, we made a significant improvement on the total cost. This work is premised on a recognition that impact on sustainability needs to be considered when seeking to reduce other costs in CFSCs. Hence the inclusion of sustainability in the optimisation model developed in this study. The results are encouraging, as the model reduces inventory costs together with a positive impact on harmful emissions.

Our model also solves the excess cumulative ending inventory level issue resulted in [SBRHVDV15]'s model and caused by pre-determined order quantities. Since our (R, S) formulation calculates order-up-to levels rather than order quantities it handles demand uncertainties well, allows flexible order size decisions that consider the actual inventory level and the realised demand, and prevents excess ending inventory and excess inventory holding costs.

This study contributes to the body of knowledge via the (R, S) policy and by developing a mixed-integer programming model responsive to non-stationary demand uncertainties inventory, transport, and regrigeration costs, and CO_2 emissions in terms of cost. The extensive computational results indicate that the proposed model can significantly reduce the total inventory costs compared to the more conventional RQ model while dealing with stochastic demand. However, RS performs similarly to RQ in terms of CO_2 emissions. Given the growing global concern for environmental sustainability, this finding could be seen as a

missed opportunity. Another possible way to reduce the carbon footprint could be developing a multi-objective model in order to allow trade-offs between the total cost and the total CO_2 emissions. Adding carbon capacity constraints, e.g., limiting CO_2 emissions per vehicle, etc., could be considered some other improvements in this model.

Since RS provides highly accurate solutions, allows flexibility on ordering decisions, and avoids replenishment timing decision set-ups and reviews which might be required for every period in the dynamic strategy, it could be easily applied to most real-world food logistics networks and utilised by executives and decision-makers concerned with the CFSC logistics.

Several possible future directions related to this study might be modifying this formulation for the multi-supplier and multi-retailer problem setting, adding a positive lead time, or considering a fleet consisting of non-identical vehicles with various properties in terms of fuel consumption, refrigeration activities, capacity, etc. The model could also be extended to consider additional environmental and social sustainability measures, such as perishability, food quality and safety, waste, shelf life, food expiration date, etc., as well as monitoring the perishability of products by adding new variables to the current model and keeping track of the age of products.

Chapter 4

Determining the Policy Parameters for Perishable Inventory Management Under Penalty Cost

In this chapter, we present the study we conducted in our publication Gulecyuz et al. [GOT25]. We propose a computationally efficient heuristic to determine the replenishment plan and the order quantities for the penalty cost version of the perishable inventory problem with the setting given in Section 4.2. Then we explain the network representation of the problem in Section 4.3 then we describe our proposed procedure to determine the optimal order-up-to levels in Section 4.4. Section 4.5 describes our proposed heuristic and our motivation to develop this heuristic in detail. In Section 4.6 we provide the numerical experiment results and compare our proposed heuristic with the optimal SDP solution. Finally, we conclude our study in Section 4.7.

4.1 Introduction

The perishability problem has several challenges that we cannot deal with by using the techniques developed for non-perishable inventory systems, in particular the multi-dimensionality of the inventory state consisting of the amount of items of each age. Considering the multi-dimensionality of the inventory state, we have to deal with a very large and unmanageable state space which is extremely hard to deal with and to employ the traditional approaches developed for non-perishable inventory systems.

We study a perishable inventory system with a finite planning horizon, non-stationary and stochastic demand with known parameters, penalty costs, a FIFO issuing policy, and a fixed shelf life. Penalty cost is an important component in perishable inventory systems since it represents the cost incurred when demand cannot be met. It can be interpreted as the consequence of shortages, such as lost sales or decreased customer satisfaction. This becomes particularly useful in settings with stochastic demand, where the risk of stockouts is present. By incorporating penalty cost, we aim to develop policies that balance service level requirements with inventory and wastage costs.

This problem can be solved to optimality by using the SDP approach. However, due to dimensionality, computation times and the inventory state space grow exponentially as the lifetime grows, and the SDP approach quickly becomes computationally infeasible in terms of computation times and the amount of memory needed. We employ a static-dynamic uncertainty strategy of [BT88] and propose a computationally efficient heuristic based on the Wagner-Whitin algorithm and Silver's heuristic in order to find near-optimal policy parameters that minimise the total cost throughout the planning horizon. In the first step, we determine the replenishment schedule before the beginning of the planning horizon. In the second step, we calculate the order quantities based on the observed demand and the pre-determined replenishment schedule.

4.2 Problem Description and Our Model

In this section, we define the problem in Section 4.2.1. Then we provide a mathematical model in Section 4.2.2 and an SDP formulation in Section 4.2.3.

4.2.1 Problem Definition

We hereby describe the problem setting based on the inventory setting and the cost structure, and explain it in more detail.

Inventory Settings

We consider the single-item and single-location perishable inventory system with non-stationary stochastic demand over a finite planning horizon of T periods. Demand is non-stationary and stochastic, hence demand distributions subject to change over time and demand realisations in each Period t, d_t, are

independent random variables with known probability density functions $f_t(d_t)$ and cumulative distribution functions $F_t(d_t)$. Customers are not allowed to return demand due to safety regulations, so we assume we have non-negative demand. The lead time is zero and replenishment orders Q_t are placed immediately at the beginning of each replenishment period. Demand realisations occur immediately after the order is placed in replenishment periods or at the beginning of each non-replenishment period. All items have a fixed lifetime of M periods and the age of a newly-ordered item is $b = 0$ periods. When the age of an item is $b \geq M$, it becomes outdated. Unmet demand is backlogged and met when available. $I_{t,b}$ denotes the inventory level for the items of Age b at the end of Period t. The inventory system operates with respect to the FIFO issuing policy, therefore older items are used first to meet the customer demand.

Costs

Ordering costs consist of a fixed setup cost per order a and a variable linear unit ordering cost c which are incurred at the beginning of each replenishment period. Inventory costs consist of a linear unit holding cost h and a penalty cost p. Inventory costs are incurred at the end of Period t. Finally, a linear unit outdating cost w is incurred if there is waste at the end of each Period t. We assume that outdated items (e.g. food past expiration date in supermarkets) incur no additional costs except the loss of revenue due to unsold and disposed outdated items. Hence, we assume $w = 0$.

4.2.2 Mathematical Model

Since demand is uncertain, we formulate our problem as a stochastic model in which d_t is the stochastic element of our model. Recall that Q_t is the order quantity in Period t, $I_{t,b}$ is the ending inventory level of Age b at the end of Period t, and a, c, h, p, and w are fixed setup, unit ordering, holding, penalty, and outdating costs, respectively. Our stochastic model is given below:

$$\min \mathbb{E}(TC) = \int_{d_T} \dots \int_{d_1} \sum_{t=1}^{T} \Bigg(\mathbb{1}(Q_t) + h(I_{t,0})^+ + h \sum_{b=1}^{M-2} I_{t,b} - p(I_{t,0})^- + wI_{t,M-1} \Bigg) f_1(d_1) \dots f_T(d_T) \mathrm{d}d_1 \dots \mathrm{d}d_t \tag{4.1}$$

subject to

$$\mathbb{1}(Q_t) = \begin{cases} a + cQ_t, & \text{if } Q_t > 0 \\ 0, & \text{otherwise} \end{cases} \qquad t = 1, 2, ..., T. \tag{4.2}$$

$$I_t = \sum_{b=0}^{M-1} I_{t,b} = \sum_{b=0}^{M-2} I_{t-1,b} + Q_t - d_t, \quad t = 1, 2, ..., T. \tag{4.3}$$

$$I_{t,0} = Q_t - \left(d_t - \sum_{b=0}^{M-2} I_{t-1,b} \right)^+, \quad t = 1, 2, ..., T. \tag{4.4}$$

$$I_{t,b} = \left(I_{t-1,b-1} - (d_t - \sum_{j=b}^{M-2} I_{t-1,j})^+ \right)^+, \quad t = 1, 2, ..., T; b = 1, ..., M-1. \tag{4.5}$$

$$I_{0,b} = 0, \quad b = 0, ..., M-1. \tag{4.6}$$

$$I_{t,b} \geq 0, \quad t = 1, 2, ..., T; b = 1, ..., M-1. \tag{4.7}$$

$$Q_t \geq 0, d_t \geq 0, \quad t = 1, 2, ..., T. \tag{4.8}$$

Note that $(x)^+ = \max(x, 0)$ and $(x)^- = \min(x, 0)$.

The model given above is the modified version of the mathematical model described in [PWHHvdV14] such that a penalty cost component is added to the objective function and the service level constraint is removed. The objective function given in (4.1) minimises the expected total cost throughout the planning horizon. In the constraint (4.2), the total ordering cost function in Period t is formulated, and the inventory levels of all ages at the end of Period t are calculated by using (4.3). The equations (4.4) and (4.5) are the FIFO constraints. The initial inventory levels for all ages are zero and the ending inventory levels are non-negative, as described in (4.6) and (4.7), respectively. According to (4.7), the ending inventory level for the items of Age 1 can be negative if the stock on hand is too small to meet demand. The last constraint in (4.8) states that the order quantity and the demand cannot be negative.

4.2.3 Stochastic Dynamic Programming (SDP) Formulation

The problem can be solved to optimality by using stochastic dynamic programming (SDP) formulation under the dynamic policy. We modify the SDP formulation developed by [HHRPW12] for the penalty cost case. We can formulate the problem as SDP as follows:

1. **Inventory State:** The $M-1$-dimensional inventory state vector at the

beginning of Period t is denoted by $\mathbf{I}_{t-1}$, such that

$$\mathbf{I}_{t-1} = (I_{t-1,0}, I_{t-1,1}, ..., I_{t-1,M-2}), t = i, ..., j \tag{4.9}$$

represents the inventory composition at the beginning of Period t. It consists of the amount of the inventory level of each age at the beginning of Period t. Since $I_{t-1,M-1}$ is equal to the number of items of Age $M-1$ that expire at the end of Period $t-1$, it is excluded in $\mathbf{I}_{t-1}$.

2. **State Transition Function:** $\Phi_t(\mathbf{I}_{t-1}, Q_t, d_t) = \mathbf{I}_t$ is the inventory state transition function to calculate the state at the beginning of Period $t+1$. It takes the observed inventory state, the order quantity, and the realised demand as inputs, then calculates the next inventory state by using Eqs (4.3)-(4.5).

3. **Waste Function:** The amount of waste at the end of Period t, $I_{t,M-1}$, is affected by d_t when the initial amount of items of Age $M-1$ is sufficient to cover all the demand, such that $I_{t-1,M-2} \geq d_t$. Therefore, it can be written as a function of the initial inventory state; $o(\mathbf{I}_{t-1}, d_t) = I_{t,M-1}$.

4. **Immediate Cost Function:** We formulate the single-period immediate cost function for Period t, $C_t(.)$, as follows:

$$\mathbb{E}C_t(\mathbf{I}_{t-1}, Q_t) = q(Q_t) + \mathbb{E}\left\{h(I_{t,0})^+ + h\sum_{b=1}^{M-2} I_{t,b} - p(I_{t,0})^- + wo(\mathbf{I}_{t-1}, d_t)\right\}. \tag{4.10}$$

5. **Objective Function:** We can write the SDP objective function via the Bellman equation using the value function for Period t, $V_t(.)$, as follows:

$$V_t(\mathbf{I}) = \min_{Q_t}\left(\mathbb{E}C_t(\mathbf{I}, Q) + \mathbb{E}(V_{t+1}(\Phi(\mathbf{I}, Q_t, d_t)))\right). \tag{4.11}$$

Example:

Since M determines the number of elements in I_{t-1} and therefore the dimensionality of the state space, the number of possible inventory states increases exponentially as M increases. Let us assume that the number of different possible inventory levels of each Age $b \in \{0, ..., M-1\}$ is 100 and the planning horizon is $T = 10$ periods. For $M = 2$, we have $100 \times 10 = 1000$ possible inventory states throughout the planning horizon. For $M = 3$, the total number of possible inventory states becomes $100^2 \times 10 = 100000$. For $M = 4$ and $M = 5$,

there are 10^7 and 10^9 possible inventory states for the whole the planning horizon, respectively.

As seen from the example, the SDP approach immediately generates a very large, multi-dimensional, and unmanageable state space graph. Therefore the SDP solution quickly becomes computationally intractable even for relatively small M values.

Pseudocode:

Algorithm 3 describes the pseudocode of the SDP formulation. It searches though all the possible starting inventory compositions and the order quantities in order to find the optimal parameters that minimise the total cost.

Algorithm 3 SDP – Penalty Cost

for $b_1 = min_inv_1$ **to** max_inv_1 **do**
  for $b_2 = 0$ **to** max_inv_2 **do**
    $\vdots$
    for $b_{M-1} = 0$ **to** max_inv_{M-1} **do**
      $C_{T+1}(b_1, b_2, \ldots, b_{M-1}) \leftarrow 0$
    $\vdots$
for $t = T$ **down to** 1 **do**
  for $b_1 = min_inv_1$ **to** max_inv_1 **do**
    for $b_2 = 0$ **to** max_inv_2 **do**
      $\vdots$
      for $b_{M-1} = 0$ **to** max_inv_{M-1} **do**
        $\mathbf{I}_{t-1} \leftarrow (b_1, b_2, \ldots, b_{M-1})$
        $x \leftarrow b_1 + b_2 + \cdots + b_{M-1}$
        $C_t(\mathbf{I}_{t-1}) \leftarrow \mathbb{E}\{wo(\mathbf{I}_{t-1}, d_t)\}$
        for $q = 0$ **to** max_order **do**
          $expected_cost \leftarrow \mathbb{E}\left\{h(I_{t,0})^+ + h\sum_{b=1}^{M-2} I_{t,b} - p(I_{t,0})^-\right\}$
            $+\mathbb{E}\left\{C_{t+1}(\Phi(\mathbf{I}_{t-1}, q, d_t))\right\}$
          if $expected_cost \leq C_t(\mathbf{I}_{t-1})$ **then**
            $C_t(\mathbf{I}_{t-1}) \leftarrow expected_cost$
            $S_t \leftarrow x + q$
            $q_t \leftarrow q$

4.3 Network Representation

Since the state space graph generated by the SDP solution quickly becomes unmanageable and computationally intractable, we employ the relaxation strategy

proposed by [Tar96] and modify it for the perishability problem. This relaxation strategy reduces the model to a shortest path problem by relaxing the constraint which enforces the order quantities to be non-negative. Hence, the model can be represented as a network on a state space graph with a certain number of arcs and nodes depending on the horizon length T. [RTHP11] propose a filtering and augmenting algorithm that combines the deterministic-equivalent DP and the state space relaxation procedure proposed by [Tar96]. They formulate a shortest path approach to find an optimal solution and to increase the computational efficiency.

We employ and modify the approach of [RTHP11] for our problem. Each Node $i \in \{0, ...T-1\}$ represents the inventory state at the beginning of Period i and the Node T represents the ending inventory state of the T-period planning horizon. Arcs represent the replenishment cycles and they are associated with cycle costs.

Replenishment Cycle: $R(i,j)$ is the replenishment cycle starting at the beginning of Period i and ending at the end of Period j, $i = 1, 2, ..., T$ and $j = i, i+1, ..., (T, i+M-1)^-$.

Expected Cycle Cost: $c(i,j)$ is the expected total cost for the replenishment cycle $R(i,j)$ with the order quantity Q_i, such that

$$c(i,j) = \mathbb{1}(Q_i) + \sum_{t=i}^{j} \left(h \sum_{b=1}^{M-1} \mathbb{E}(I_{t,b})^+ - p\mathbb{E}(I_{t,0})^- + w\mathbb{E}(I_{t,M-1}) \right) \tag{4.12}$$

Figure 4.1 illustrates a network representation of an inventory system with $T = 4$ periods. If there is no perishability, all the arcs between the nodes 0-4 are associated with the possible replenishment cycles. However, in perishable inventory systems the arcs longer than M are excluded. Let us consider $M = 3$ and for simplicity assume all inventory compositions with the same sum are represented by a single node. Since there are no possible replenishment cycles longer than M periods due to perishability, the dashed arc $(0, 4)$ which represents the 4-period replenishment cycle $R(1, 4)$ is excluded from the graph.

For non-perishable inventory systems, Node t represents the opening inventory level for Period $t+1$ and the closing inventory level for Period t. On the contrary, the inventory state is defined by the composition of the amount of items of each Age $b = 0, ..., M-1$ in a perishable inventory model. Unlike non-perishable inventory systems, the same amount of the total opening inventory level may

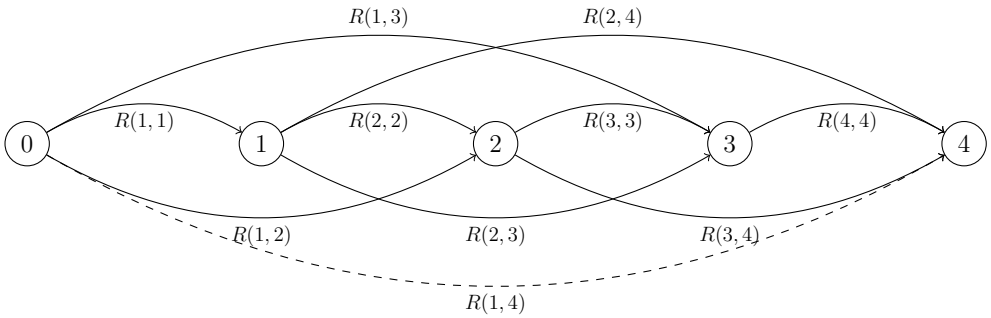

Figure 4.1: The network representation of all possible replenishment cycles of a perishable inventory model with $T = 4$ and $M = 3$. Notice that each Node $i = 1, ..., T - 1$ represent the end of Period i and the beginning of Period $i + 1$, Node 0 and Node T represent the beginning and the ending of the planning horizon, respectively, and each arc $(i, j) : i = 0, ..., T - 1, j = 1, ..., T, i < j$, represents $R(i + 1, j)$.

correspond to many different inventory compositions. In addition, the starting inventory level at the beginning of Period t and the ending inventory level at the end of Period $t - 1$ are different from each other if $I_{t-1,M-1} > 0$. Therefore, representing the perishable inventory models as a network graph is a challenging issue.

4.4 Replenishment Cycle Order-Up-To Level Calculation

We describe the procedure to calculate the order-up-to levels and the order quantities for the replenishment cycle $R(i, j)$.

4.4.1 Re-Formulation of the Total Cost Function

The ending inventory level in Period t in $R(i, j)$, $i \leq t \leq j$, is given below:

$$I_t = S_{i,j} - \sum_{k=i}^{t} d_k - \sum_{k=i}^{t-1} I_{k,M-1}. \tag{4.13}$$

where $S_{i,j}$ is the order-up-to level for $R(i, j)$. The number of non-expiring items in Period t, e.g. $b < M - 1$, in $R(i, j)$ is given as follows:

$$B_{i,t} = S_{i,j} - \sum_{k=i}^{t} d_k - \sum_{k=i}^{t-1} I_{k,M-1}. \tag{4.14}$$

The outdated items in Period t included in the ending inventory. [TK06] reformulates the total cost function for the non-perishable version of our model in terms of replenishment periods and cycles. We modify [TK06]'s formulation by including the waste component and apply Eq. (4.13) to calculate the holding and penalty costs incurred at the end of Period t, $I_{t,M-1}$. Notice that $I_{t,M-1}$ is excluded since it is the amount of waste at the end of Period t. Thus, we write Eq. (4.1) as

$$\mathbb{E}(TC) = \sum_{i=1}^{m} \Bigg[a\mathbb{1}(Q_{T_i}) + \sum_{t=T_i}^{T_{i+1}-1} \Bigg(h \int_{-\infty}^{S_{i,j}} B_{T_i,t} f_{T_i,t}(d_{T_i,t}) \mathrm{d}d_{T_i,t} - p \int_{S_{i,j}}^{\infty} B_{T_i,t} f_{T_i,t}(d_{T_i,t}) \mathrm{d}d_{T_i,t} + w\mathbb{E}(I_{t,M-1}) \Bigg) \Bigg] + cI_T + c \sum_{t=1}^{T-1} \mathbb{E}(I_{t,M-1}) + c \int_{d_{1,T}} d_{1,T} f_{1,T}(d_{1,t}) \mathrm{d}d_{1,t}, \quad (4.15)$$

where $d_{i,t} = \sum_{k=i}^{t} d_k$, T_i is the start of the i-th replenishment period, $1 \leq T_i \leq T$, m is the number of order placements, and

$$\mathbb{1}(Q_t) = \begin{cases} 1, & \text{if } Q_t > 0 \\ 0, & \text{otherwise} \end{cases} \qquad t = 1, 2, ..., T.$$

Hence, the reformulation of Eq. (4.12) is given by

$$c(i,j) = a + \sum_{t=i}^{j} \Bigg(h \int_{-\infty}^{S_{i,j}} B_{i,t} f_{i,t}(i,t) \mathrm{d}d_{i,t} - p \int_{S_{ij}}^{\infty} B_{i,t} f_{i,t}(i,t) \mathrm{d}d_{i,t} + w \sum_{k=i}^{t} \mathbb{E}(I_{k,M-1}) \Bigg), \quad (4.16)$$

where $\mathbb{E}(I_{j,M-1}) = \int^{S-I0} H(S-I0)$ if $j = i+M-1$, in other words if the length of $R(i,j)$ is equal to M periods. $I0 = \sum_{b=1}^{M-1} I_{b,i-1}$ is the opening inventory level for $R(i,j)$, $\int^{y} H(y)$ is the expected outdating given the order quantity $Q_i = S - I0$ at the beginning of Period i, and $H(Q_i)$ is the probability of outdating at the end of Period $t + M - 1$.

4.4.2 Calculating the Expected Outdating

Calculating the exact expected outdating given the replenishment order Q_t in Period t is extremely challenging. To our knowledge, there is no known proce-

dure in the literature that simplifies the calculation of the exact expected outdating, except several estimation methods such as [BVD09]'s EWA policy and [GARMBH17]'s approach for Poisson demand. For the details of the expected outdating and the proofs, see [Nah75].

Note that the number of expected outdated items between Periods i and $i+M-2$, $\mathbb{E}(I_{t,M-1}, t) = i \leq t < i+M-1$, are independent from Q_i and $S_{i,j}$, depend on the previous replenishment orders, and remain constant for any different values of Q_i and $S_{i,j}$. Therefore, we take them as the constants within the replenishment cycle $R(i,j)$, under the deterministic-equivalent setting.

For the expected outdated items from the previous replenishment orders within the current replenishment cycle, we apply the EWA procedure of [BVD09]. Their approach estimates the expected outdating at the end of each period by assuming that the expected demand values occur in each period, and in a deterministic-equivalent framework.

4.4.3 Multi-Period Newsvendor Setting and Calculating the Optimal Parameters

We apply the multi-period newsvendor setting described in [Ros21] to calculate the optimal order-up-to level S^*_{ij} and the optimal order quantity Q^*_i for $R(i,j)$. Let us consider the stochastic component of $c(i,j)$ and denote it by $C_{i,j}(S_{i,j})$ such that

$$C_{i,j}(S_{i,j}) \quad = \quad \sum_{t=i}^{j} \left(h \int_{-\infty}^{S_{ij}} B_{i,t} f_{i,t}(i,t) \mathrm{d}d_{i,t} \; - \; p \int_{S_{ij}}^{\infty} B_{i,t} f_{i,t}(i,t) \mathrm{d}d_{i,t} \right). \quad (4.17)$$

We then determine $S^*_{i,j}$ for $R(i,j)$ that minimises (4.17) in the multi-period newsvendor setting. The overage cost occurring per unit of unused inventory is equal to the holding cost h and the underage cost occurring per unmet demand is equal to the penalty cost p. Let us consider a random variable ω and a scalar x.

Definition 1. $\mathbb{E}(\omega - x)^+$ is the first order loss function, see Rossi et al. [RTPH14].

Definition 2. $\mathbb{E}(x-\omega)^+$ is the complementary first order loss function, see [RTPH14].

Lemma 1. $\mathbb{E}(\omega - x)^+ = \mathbb{E}(x-\omega)^+ - (x - \mathbb{E}(\omega))$.

For the proof, see [RTPH14].

The single-period inventory cost function

$$h\int_{-\infty}^{S}(S-d_t-I_{t,M-1})f_t(t)\mathrm{d}t - p\int_{S}^{\infty}(S-d_t-I_{t,M-1})f_t(t)\mathrm{d}t$$

is convex.

Proof. Note that the observed waste at the beginning of Period t, $I_{t,M-1}$ are not dependent on d_t and S. Thus, $x = S - I_{t,M-1}$ is constant. $\mathbb{E}(\omega - x)^+$ and $\mathbb{E}(x-\omega)^+$ are convex, see [RTPH14]. From Definitions 1 and 2, we can see that $h\int_{-\infty}^{S}(S-d_t-I_{t,M-1})f_t(t)\mathrm{d}t$ is the complementery first order loss function and $p\int_{S}^{\infty}(d_t-S+I_{t,M-1})f_t(t)\mathrm{d}t$ is the first order loss function. Hence, the single-period cost function is convex since it is the sum of two convex functions, the holding cost $h > 0$, and the penalty cost $p > 0$. □

Lemma 3. $\mathbb{E}(x-\omega)^+ = \int_{-\infty}^{x} F(t)\mathrm{d}t$, where $F(.)$ is the cdf of ω.

For the proof, see [RTPH14].

Using Lemma 1 and Lemma 3, we rewrite Eq. (4.17) as follows:

$$C_{i,j}(S_{i,j}) \;=\; \sum_{t=i}^{j}\left(\Big(h\;+\;p\Big)\int_{-\infty}^{S_{ij}-\sum_{k=i}^{t}I_{k,M-1}} F_{i,t}(d_{i,t})\mathrm{d}d_{i,t}\;-\;pB_{i,t}\right). \quad (4.18)$$

Lemma 4. $C_{i,j}(S_{i,j})$ is convex.

Proof. From Lemma 2, the sum of the convex functions is convex. Since Eq. (4.18) is the sum of the convex single-period inventory functions, it is convex. □

We take the derivative of $C_{i,j}(S_{i,j})$ with respect to $S_{i,j}$ in order to find the optimal order-up-to level:

$$\frac{dC_{i,j}(S_{i,j})}{dS_{ij}} = \sum_{k=i}^{j}\left(\Big(h+p\Big)F_{i,t}\left(S_{i,j}-\sum_{k=i}^{t}I_{k,M-1}\right)-p\right)$$

Then we obtain

$$\sum_{t=i}^{j}F_{i,t}\left(S^*_{i,j}-\sum_{k=i}^{t}I_{k,M-1}\right)=\frac{Np}{h+p} \quad (4.19)$$

where the length of the replenishment cycle is denoted by $N = j - i + 1$.

The optimal order-up-to level for Eq. (4.19) has no explicit formulation. Therefore, several techniques such as the Nelder-Mead algorithm need to be applied.

Moreover, we must consider the expected total number of outdated items from the previous replenishment cycle orders, $\sum_{k=i}^{min(j,i+M-2)} I_{k,M-1}$, and calculate it by employing the EWA procedure of [BVD09] as described in Section 4.4.2, and incorporate it into our modified newsvendor formulation.

4.5 Heuristic Approach for the Penalty Cost Case

In this section, we introduce and explain our proposed heuristic, the decision rule, and the motivation to develop this heuristic in detail.

Recall that the state space consists of many possible inventory compositions, and the cycle cost $c(i,j)$ is dependent on the inventory composition denoted by the vector $\mathbf{I}_{t-1} = (I_{t-1,0}, I_{t-1,1}, ..., I_{t-1,M-2}), t = i, ..., j$ and the previous path information from period 1 to Period $i-1$. Therefore, the problem becomes harder to be represented as a single-weight shortest path problem, the number of possible total cost and inventory composition values increases exponentially as T increases, and the model becomes computationally intractable. In order to deal with these issues, we propose a computationally efficient heuristic that determines a near-optimal path and a total cost up to Period $j = 1, ..., T$ recursively.

Instead of dealing with the corresponding nodes and arcs in the state space graph for each possible inventory composition and replenishment cycles in order to find the shortest path, we can employ the dynamic programming approach and find the best path in a recursive way. To do so, it is possible to apply and modify the well-known Wagner-Whitin algorithm [WW58] which finds the optimal solution for the dynamic lot sizing problem via dynamic programming. However, we still have several issues: Firstly, the Wagner-Whitin solution takes only the cost into account and the actual value of the starting inventory composition in Period t, $\mathbf{I}_{t-1}$, does not have any effect when determining the path. Moreover, the path found via the Wagner-Whitin algorithm does not guarantee that it is the near-optimal path, which we refer to as the "best" path, we can find, since the best path found may differ for each $\mathbf{I}_{t-1}$ having the same total items $\sum_{b=0}^{M-2} I_{t,b}$. Therefore, we need to set up a decision rule which considers both the inventory composition and cost values and combines them with the dynamic programming approach. In our decision rule we are inspired by the Silver-Meal heuristic [Sil73], since we aim to keep the average cost per period at minimum when determining the next replenishment cycle if the inventory on

hand is not sufficient for the following periods given $\mathbf{I}_t$, or under the assumption that we use the remaining items.

Our heuristic aims to determine the replenishment Periods R_t and the order quantities Q_t in a similar manner to the static-dynamic uncertainty strategy of [BT88]. Since the optimal order quantity in Period t greatly depends on the inventory composition $\mathbf{I}_{t-1}$ at the beginning of Period t, it does not necessarily fit the order quantity for the deterministic-equivalent path. Using the deterministic-equivalent approach, we firstly run our algorithm and find the replenishment Periods $R_t, t = 1, ..., T$. Then we perform a simulation run by taking the vector that indicates the replenishment Periods

$$\mathbf{Y}_t = (Y_1, ..., Y_t, ..., Y_T), Y_t \in 0, 1$$

as an input. Y_t is defined by

$$Y_t = \begin{cases} 1, & \text{if } t = R_t \\ 0, & \text{otherwise} \end{cases} \qquad t = 1, 2, ..., T.$$

We calculate Q_t by performing a simulation run and using the observed inventory state composition at the beginning of each Period t. Our heuristic calculates the replenishment plan and the total cost at the end of Period T, $TC(T)$, and works in a recursive manner.

4.5.1 Overview

The first step involves determining the replenishment periods by evaluating all possible replenishment cycles that end in each Period t in a similar manner to the Wagner-Whitin [WW58] algorithm. This process takes both the inventory compositions at the end of each period and the replenishment cycle costs given these compositions. Using a modified Silver-Meal heuristic [Sil73], the method selects the replenishment cycle that minimises the average total cost over time, considering both future inventory availability and demand coverage. By iterating through all feasible replenishment cycles and selecting the most cost-effective path, the heuristic calculates a near-optimal replenishment schedule for the planning horizon.

Once the replenishment periods are determined, the corresponding order quantities are calculated in the second step, based on the observed inventory compo-

sitions. Since they can differ from the deterministic-equivalent model, we use a simulation approach to determine order quantities dynamically. By employing a multi-period newsvendor framework and estimating waste for each replenishment cycle, the heuristic determines the optimal order quantity that minimises cycle costs in the second step.

4.5.2 Step 1: Determining the Replenishment Periods

At the end of each Period $t = 1, ..., T$, we firstly calculate the corresponding cycle costs $c(i,t)$ for each replenishment cycle $R(i,t), i = \max(1, t-M+1), ..., t$, starting in periods from $t-M+1$ to t and ending at the end of Period t. If the ending inventory composition is sufficient for meeting the next periods' demand without any new replenishment order placement, we then calculate the total cost of the remaining inventory as long as it suffices the future demand. Otherwise, we check all the next possible replenishments given the composition $\mathbf{I}_t$, then select the replenishment order with the least average cost by using the modified Silver-Meal heuristic of [GARMBH17]. Our goal is to ensure that we keep the average total cost at minimum after Period t by involving the inventory composition states as the determining factors. We denote the remaining inventory cost or the possible future replenishment cost given $\mathbf{I}_t$ at the end of Period t by

$$c'(t+1, T_{i,t}) = \mathbb{1}(Q_i) + \sum_{k=t+1}^{T_{i,t}} \left(h \sum_{b=1}^{M-1} \mathbb{E}(I_{k,b})^+ - p\mathbb{E}(I_{k,0})^- + w\mathbb{E}(I_{k,M-1}) \right),$$

and we denote the last period in which the expected demand is sufficed by the remaining inventory at the end of $R(i,t)$ by $T_{i,t}$. In case there is not enough demand, $T_{i,t}$ represents the last period that is covered by the next replenishment cycle (e.g. $R(t+1, T_{i,t})$) selected via the modified Silver-Meal heuristic.

Next, we choose the path that minimises the total cost up to the end of Period t, inspired by the Wagner-Whitin algorithm. We call this path the "best path". Let us define the set of the replenishment cycles on the best path up to the end of Period t by $\mathcal{R}_t$. We define the best path in terms of replenishment cycles.

Suppose that $M = 2$ and $T = 3$. We initially set $\mathcal{R}_1 = \{R(1,1)\}$ and $\mathbf{Y}_1 = (1)$, since it is the only possible path before period 1, as seen in Figure 4.2. Similarly, Figure 4.3 illustrates $\mathcal{R}_2$ would be either $\mathcal{R}_2 = \{R(1,1), R(2,2)\}$ highlighted in red or $\mathcal{R}_2 = \{R(1,2)\}$ highlighted in blue, and the corresponding replenishment

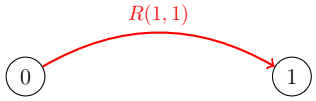

Figure 4.2: The only possible path to the end of period 1.

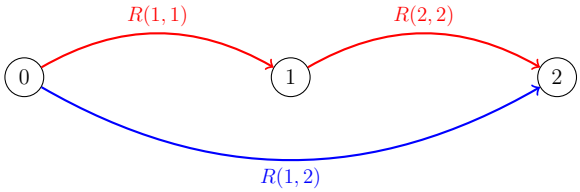

Figure 4.3: The possible paths to the end of period 2.

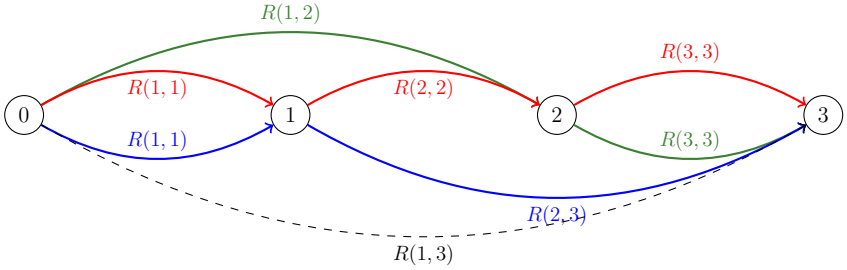

Figure 4.4: The possible paths to the end of period 3.

vector is $\mathbf{Y}_2 = (1, 1)$ or $\mathbf{Y}_2 = (1, 0)$, respectively. According to Figure 4.4, when $t = 3$, $\mathcal{R}_3 = \{\mathcal{R}_1, R(2, 3)\}$ highlighted in blue, or $\{\mathcal{R}_2, R(3, 3)\}$ which can be either $\{R(1, 1), R(2, 2), R(3, 3)\}$ highlighted in red or $\{R(1, 2), R(3, 3)\}$ highlighted in green. Then the corresponding vector would be $\mathbf{Y}_3 = (\mathbf{Y}_1, 1, 0) = (1, 1, 0)$ or $\mathbf{Y}_3 = (\mathbf{Y}_2, 1)$. Note that since $M = 2$ we do not include $R(1, 3)$ which is illustrated as a dashed line in 4.4.

Since we mentioned earlier that our aim is to employ the dynamic programming approach while using both cycle cost $c(t + 1, T_{i,t})$ and inventory state $\mathbf{I}_t$ as the decision parameters and keeping the average cost at a minimum level, our decision rule is inspired the Wagner-Whitin algorithm and the Silver-Meal heuristic which are among the fundamental and most well-known methods in the literature. We choose the best path prior to the end of Period t recursively, and by selecting the path that results in an inventory composition which keeps the average total costs at minimum after Period t. We determine the best path with respect to the least average cost up to Period $T_{i,t}$, using the formula given below:

$$AC(T_{i,t}) = \frac{TC(i-1) + c(i,t) + c'(t+1, T_{i,t})}{T_{i,t}}, \quad i \in \max(1, t - M + 1), ..., t, \tag{4.20}$$

where $AC(T_{i,t})$ denotes the average total cost at the end of Period $T_{i,t}$. We select the replenishment cycle which starts at the beginning of Period i and ends at the end of Period t such that

$$i^* = \arg\min_i AC(T_{i,t}). \tag{4.21}$$

Then the total cost up to the end of Period t can be expressed as follows:

$$TC(t) = TC(i^* - 1) + c(i^*, t), \tag{4.22}$$

and

$$TC(0) = 0.$$

We repeat this process for Period $t+1$ and continue until the end of the planning horizon.

The pseudocode of the algorithm for process of determining the replenishment periods is given in Algorithm 4. A brief explanation of Algorithm 4 is as follows: We firstly take the opening inventory levels and the demand expectations as inputs, and calculate a binary vector indicating the replenishment periods. We initialise the total cost to be zero and the replenishment cycle set to be an empty set. Each iteration of the outermost for loop represents the process in each period. In the inner for loop we compare all the possible previous replenishment cycles ending at the end of Period j and their resulting inventory compositions, then we select the next possible replenishment cycle with the lowest average cost per period in the inner if else statement. In the last if statement, we update the total cost vector and the replenishment cycle set if we keep the average cost at minimum.

4.5.3 Step 2: Determining the Order Quantities

In order to determine the order quantities, we employ a dynamic strategy and perform a simulation run. Recall that the state vector represents the inventory composition of each age. Hence, the order quantities found in the previous stage of our heuristic corresponds to the resulting inventory compositions and states from the deterministic-equivalent model. Due to the mismatch between the inventory states, applying the static uncertainty strategy may result in much higher costs. In order to deal with this issue, we need to consider the instantaneous inventory compositions and states during simulation while determining

the order quantities.

Consider the replenishment cycle $R(i,j) \in \mathcal{R}_T, i, j, = 1, ..., T$ determined by the procedure described in Section 4.5.2. Given the observed starting inventory composition in Period i, $\mathbf{I}_{i-1}$, we calculate the order quantity via local search. In order to determine the order quantity that minimises $c(i,j)$, we employ the multi-period newsvendor approach explained in detail in Section 4.4, using the expected total demand $\sum_{k=i}^{j} d_k$ and total waste $\sum_{k=i}^{j-1} I_{k,M-1}$, and the inverse CDF $F_{i,j}^{-1}$ for $R(i,j)$. We repeat this process for each pre-determined replenishment cycle until the end of the planning horizon.

Algorithm 4 Heuristic – Penalty Cost

Input: Opening inventory $\mathbf{I}_0 = (I_{1,0}, \ldots, I_{M-1,0})$, expected demand $d = (d_1, \ldots, d_T)$
Data: Directed state space graph with arcs representing at most M-period replenishment cycles
Result: Binary replenishment cycle vector $\mathbf{Y}_T$
$TC \leftarrow zeros(T)$ ▷ Total cost vector
$\mathcal{R} \leftarrow \emptyset$ ▷ Replenishment cycle set
$TC(0) \leftarrow 0$
for $j = 1$ to T **do**
    $avgcost \leftarrow \infty$
    for $i = \max(1, j - M + 1)$ to j **do**
        Calculate $S_{i,j}^*$ based on multi-period newsvendor formulation
        $c(i,j) = \mathbb{1}(Q_i) + \sum_{t=i}^{j} \left(h \sum_{b=1}^{M-1} \mathbb{E}(I_{t,b})^+ - p\mathbb{E}(I_{t,0})^- + w\mathbb{E}(I_{t,M-1}) \right)$
        Calculate $T_{i,j}$
        if $T_{i,j} > j$ **then**
            $c'(j+1, T_{i,j}) \leftarrow$ total cost of remaining inventory (if sufficient)
        else if $T_{i,j} = j$ **then**
            $T_{i,j}^* \leftarrow \underset{T'_{i,j} > j}{\arg\min}\, c(j+1, T'_{i,j})/(T'_{i,j} - j)$
            $T_{i,j} \leftarrow T_{i,j}^*$
            $c'(t+1, T_{i,j}) \leftarrow c(i+1, T_{i,j})$
        $temp \leftarrow (TC(i-1) + c(i,j) + c'(j+1, T_{i,j}))\,/T_{i,j}$
        if $temp < avgcost$ **then**
            $avgcost \leftarrow temp$
            $TC(j) \leftarrow TC(i-1) + c(i,j)$
            $\mathcal{R}(j) \leftarrow \{\mathcal{R}(i-1), R(i,j)\}$
            $\mathbf{Y}_i \leftarrow (\mathbf{Y}_{i-1}, 1)$
        else
            $\mathbf{Y}_i \leftarrow (\mathbf{Y}_{i-1}, 0)$
return $\mathbf{Y}_T$

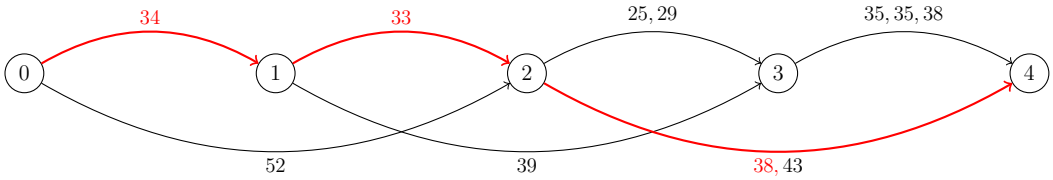

Figure 4.5: The network representation of the example problem with all possible cycle costs. The best path is highlighted in red.

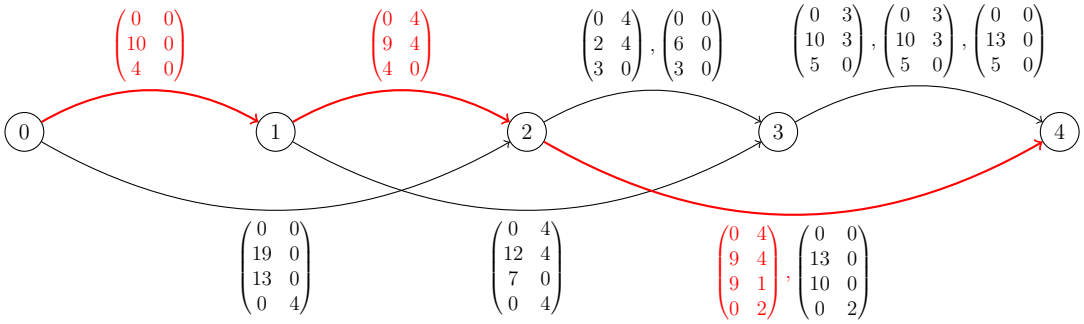

Figure 4.6: The network representation of the example problem with all possible inventory compositions. The best path is highlighted in red. The first and the second row represent the starting inventory at the beginning of the replenishment cycle and the inventory composition after the order is placed, respectively. The remaining rows show the inventory compositions at the end of each period in the replenishment cycle. The columns represent the number of items of each age; the first and the second columns show the number of items of Age 0 and 1, respectively.

4.5.4 An Example

We shall explain our heuristic via a simple example in detail, and clarify how it is possible to apply the heuristic in practice. We generate a small inventory model with $M = 2$, $N = 4$, $I_{0,b} = 0, \forall b = 0, ..., M-1$, the demand in each Period t has a Poisson distribution with rates $\lambda_t = (6, 9, 4, 8)$. We have a fixed ordering cost $a = 20$, unit ordering, holding, penalty, and outdating costs $c = 1$, $h = 1$, $p = 10$ and $w = 0$, respectively. This procedure follows the deterministic-equivalent approximation of the EWA procedure, where order quantities are determined using average demand and the resulting expected waste. Figures 4.5-4.7 illustrate the network representation for the problem. In Figure 4.5, each arc are labeled with all the corresponding cycle costs. Figure 4.6 shows the inventory composition values for the same problem. In Figure 4.7, the previous path information for the same problem is given. At the end of the planning horizon, we obtain the path highlighted in Figures 4.5-4.7.

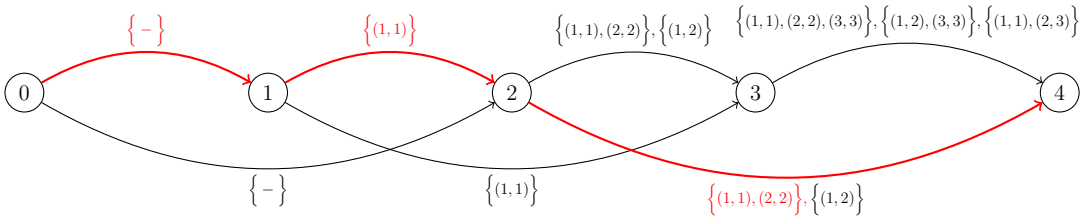

Figure 4.7: The network representation of the example problem with the previous replenishment cycle information. The best path is highlighted in red.

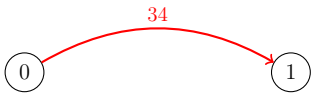

Figure 4.8: The best path at the end of period 1 with the corresponding cycle cost values.

Period 1:

We check all the possible options at the end of each Period t. Let us start from $t = 1$. Consider the node 1 which corresponds to the end of period 1. In this node, placing a one-period order covering period 1 is the only option. Therefore, the best path between the nodes 0 and 1 always consists of the arc $(0, 1)$ which represents $R(1, 1)$. Hence, the set $\mathcal{R}_1 = \{R(1, 1)\}$. According to the formulation given in Section 4.4 in detail, the order-up-to level for $R(1, 1)$ is $S_{1,1} = 10$. Since $\mathbf{I}_0 = (0)$, we order $Q_1 = 10$ items. Then the expected demand realisation $\mathbb{E}\{d_1\} = 6$ occurs. Finally, we have 4 items on hand and $\mathbf{I}_1 = (7)$. Hence, the total cost for $\mathcal{R}_1$ is calculated as $TC(1) = a + cQ_1 + h\sum_{b=0}^{M-2} I_{1,b} + wI_{1,M-1} = 20 + 10 + 4 + 0 = 34$. At the end of period 1, we highlight the arc $(0, 1)$ as seen in Figure 4.8.

Period 2:

For $t = 2$, we consider the node 2 which represents the end of period 2. Now we have two options: Placing a 2-period order covering periods 1 and 2 at the beginning of period 1, or placing two one-period orders at the beginning of periods 1 and 2. In other words, we need to choose the arc $(0, 2)$, or the arcs $(0, 1)$ and $(1, 2)$, respectively.

In the first option, we order up to $S_{1,2} = 19$. At the end of the first period, all the items in the inventory are of Age 0, thus we have $I_{1,0} = 19 - 6 = 13$ and $I_{1,1} = 0$ items on hand. At the end of period 2 all the items on hand are of Age 1 and since $M = 2$, $I_{2,1} = 13 - 9 = 4$ items are outdated, therefore no inventory is held for the following periods. We do not have enough inventory to cover the next

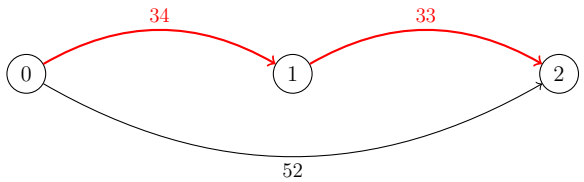

Figure 4.9: The best path at the end of period 2 with the corresponding cycle cost values.

period's demand. Therefore, we choose the replenishment decision with the least average cost to calculate $c'(3, T_{i,j})$ using the modified Silver-Meal heuristic. The selected decision covers the next two periods and sufficient for the demand until the end of Period $T_{1,2} = 4$. Hence, $TC(2) = 20 + 19 + 13 + 0 = 52$, $c'(3, T_{i,j}) = 45$, and $AC(T_{12}) = (TC(2) + AC(4))/T_{1,2} = (52+45)/(2+2) = 24.25$ for $\mathcal{R}_2 = \{R(1,2)\}$. Note that if we select $R(3,3)$, $AC(T_{12}) = (52+25)/(2+1) = 25.67$. Therefore it is not the following cycle which keeps the average cost at minimum.

In the second option, we place a new 1-period order after $R(1,1)$. Recall that $\mathbf{I}_1 = (4)$, so we start period 2 with 4 items of Age 1. From the procedure discussed in Section 4.4, $S_{2,2} = 13$ and we order $Q_2 = 9$ items. Then the demand $\mathbb{E}\{d_2\} = 9$ is realised. At the end of period 2, $I_{2,0} = 4$ and there is no outdating since $I_{2,1} = 0$. In this scenario, the resulting set of replenishment cycles is $\mathcal{R}_2 = \{R(1,1), R(2,2)\}$, the total cost for $\mathcal{R}_2$ is calculated as $TC(1) + 20 + 9 + 4 = 67$. Since we have sufficient inventory for $\mathbb{E}\{d_3\} = 3$, $T_{i,j} = 3$ and $c'(3, T_{i,j}) = 1$. We then have $AC(T_{2,2}) = (TC(2) + AC(3))/TC_{2,2})(67 + 1)/3 = 22.67$.

Since $AC(T_{2,2}) < AC(T_{1,2})$, we choose the 2nd option for the best path at the end of period 2, then set $\mathcal{R}_2 = \{R(1,2)\}$. Notice that we did not select the option 1 despite having less total cost (52) than that of option 2 (67), since we would not keep the average cost at minimum. Finally, we update the graph, remove the highlight from the arc $(0,1)$, and select the arc $(1,3)$ as in Figure 4.9. Recall that the best path is updated in each period, the selected arcs may not necessarily remain in the best path permanently, and they can be removed from the path, depending on the updates on the path in the following periods.

Following periods:

We repeat the same procedure for the following periods recursively. Note that the previous best paths may change due to the future inventory compositions and the path decisions (e.g. $\mathcal{R}_3 = \{R(1,2), R(3,3)\}$ can be possible).

At the end of the planning horizon, we obtain the replenishment cycle set $\mathcal{R}_4 = \{R(1,1), R(2,2), R(3,4)\}$ and the vector $\mathbf{Y}_4 = (1,1,1,0)$. Using $\mathbf{Y}_4$ as an input, we find the order quantities via performing a simulation run with 5000 replications and calculate the mean of the total cost during the planning horizon as $TC(4) = 116.2$.

4.6 Numerical Experiments

In this section, we discuss the numerical experiments that we performed to evaluate the performance of our heuristic. We implement our heuristic in MATLAB for our inventory problem. We implemented the SDP model in C since we realised that MATLAB was not as time-efficient as C to calculate the SDP model even for $M = 3$. We perform our numerical experiments on a machine with an Intel Core i7-1165G7 2.80 GHz processor and 16 GB RAM. We first generate the replenishment period vectors via our heuristic, then we determine the order quantities with respect to the instantaneous starting levels while performing simulation runs in order to calculate the total cost.

4.6.1 Experimental Setup

We perform the numerical experiments over an instance set of the combinations of various demand patterns, shelf lives, and cost parameters. There are three different fixed ordering cost parameters, $a \in \{25, 50, 75\}$, two different unit ordering cost parameters, $c \in \{1, 3\}$, and two different unit penalty cost parameters, $p \in \{10, 20\}$. We consider the unit holding and outdating costs to be $h = 1$ and $w = 0$, respectively. The inventory consists of items having $M \in \{2, 3, 4\}$ periods of shelf lives. The horizon length is set to $T = 12$ periods (e.g., the planning horizon may be thought as the entire year and each period may represent months). All the parameters are summarised in Table 4.1. There are also 6 different Poisson demand sets with stationary, increasing, decreasing, seasonal, cyclic, and randomly patterned demand rates which are given Table 4.2 and visualised in Figure 4.10. There are 216 randomly generated instances in total.

We use the optimal SDP results as a benchmark to our heuristic. Since the dimensionality of the perishable inventory problem severely limits the performance of the SDP model, we study with the demand sets with relatively small arrival rates.

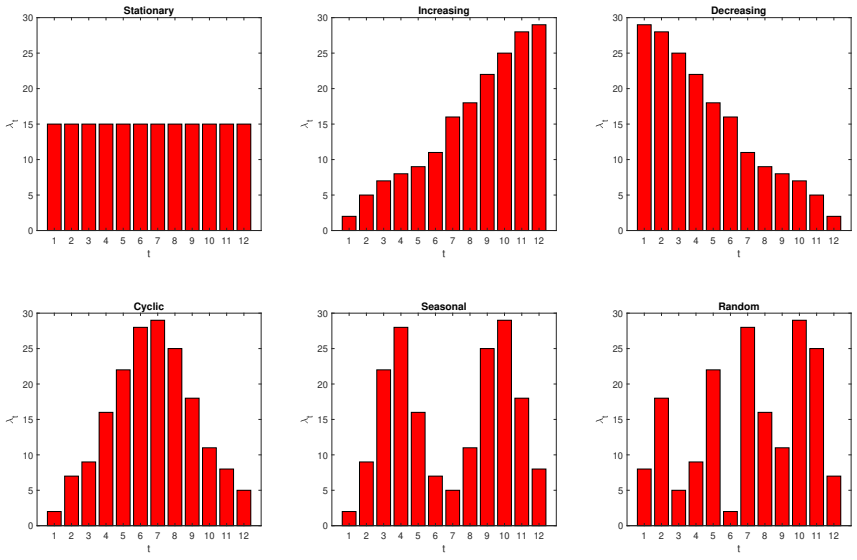

Figure 4.10: Demand patterns for each demand set.

Table 4.1: Input parameters for the experiments

Input parameters	**Values**
Fixed setup cost (a)	{25, 50, 75}
Unit ordering cost (c)	{1, 3}
Unit holding cost (h)	1
Unit penalty cost (p)	{10, 20}
Unit outdating cost (w)	0
Shelf life (M)	{2, 3, 4}
Planning horizon (T)	12

Table 4.2: Demand sets, demand patterns, and arrival rates

Demand patterns	**Arrival rates in period t**											
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**
Stationary	15	15	15	15	15	15	15	15	15	15	15	15
Increasing	2	5	7	8	9	11	16	18	22	25	28	29
Decreasing	29	28	25	22	18	16	11	9	8	7	5	2
Seasonal	2	7	9	16	22	28	29	25	18	11	8	5
Cyclic	2	9	22	28	16	7	5	11	25	29	18	8
Random	8	18	5	9	22	2	28	16	11	29	25	7

4.6.2 Results and Discussion

In order to guarantee that the SDP model is sufficient to generate the optimal order quantities, we construct a grid with a size of $151^{M-1} \times T$ possible states, so that the amount of items of each age can take 151 different values, between -50 and 100. For each of our instances and by using the simulation model with 5,000 replications, we calculate the expected finite horizon total cost.

Costs

Table 4.3 summarises the average percentage difference of the optimality gap between the heuristic and the SDP with respect to each parameter. The percentage difference of the optimality gap is calculated by the following formula:

$$100 \times \frac{TC_H - TC_{SDP}}{TC_{SDP}}$$

where TC_H and TC_{SDP} denote the total cost for the heuristic and the SDP solution, respectively. On average, the optimality gap tends to be very small, only 1.9%. The average total cost for the whole instance set is found to be 837.6 and 852.9 for the SDP and the heuristic, respectively. An average replenishment cycle covers 2.26 periods.

We notice that the sets with the seasonal and the random demand patterns tend to have slightly higher optimality gaps which are greater than 2%. One possible explanation might be the sharp fluctuations between the demand distributions in some of the subsequent periods benefit from more dynamic control. We also observe that the optimality gap increases as the amount of shelf life increases. Notice that the average replenishment cycle is 1.81 periods for $M = 2$ and it becomes equal to 2.60 when $M = 4$. Hence, most replenishment cycles cover 2 periods rather than 1 period in case $M = 2$. As M increases, the proportion of the M-period cycles tends to decrease due to the decrease in the ratio between the average replenishment cycle length. In addition, the total cost tends to drop as M increases, since longer shelf life allow less number of replenishment orders.

A similar relationship between the fixed setup cost and the replenishment cycle length can be observed. When the fixed setup cost a is sufficiently low, placing replenishment orders more frequently does not have a significant effect on the increase in the total cost. If a gets higher, ordering decisions become more costly, and therefore we need to keep the replenishment cycles as long as

possible to avoid additional ordering costs. We can see that in Table 4.3, as a increases from 25 to 75, the average cycle length increases from 1.74 to 2.62. On the contrary, the optimality gap percentage and the fixed cost tend to be inversely proportional, but for this experiment we cannot say that they are strictly inversely proportional since their 95% confidence intervals overlap. Moreover, we notice that the optimality gap tends to increase as M increases, from 1.07% when $M = 2$ to 2.66$ when $M = 4$, while there is no significant difference in the optimality gap percentages for the different values of p and c. However, increasing c significantly increases the total cost. Also, there are no significant changes in the average replenishment cycle lengths due to the changes in p and c.

Computation Times

As seen from Table 4.3, our proposed heuristic has a significant advantage over the SDP model in terms of the computation time. The heuristic determines the replenishment periods in the order of just milliseconds (7.9ms on average per instance) regardless of the shelf life and determines the order quantities in seconds (7.59s on average per instance) with small increases in terms of seconds as M increases. On the contrary, even though the SDP model works very fast (0.19s on average per instance) for $M = 2$ and reasonably fast (33.90s on average per instance) for $M = 3$, it slows down exponentially when $M = 4$ and generates the order quantity values in more than 1 hour (4,953.6s on average) per instance. Using the grid size we mentioned above, in order to compute the parameters for all the instances, the SDP model requires a computation time of 14 seconds, 40 minutes, and 4 days for $M = 2$, $M = 3$, and $M = 4$, respectively. If the shelf life increased to $M = 5$, the computation times would continue to increase exponentially, and it would take more than 1 week to calculate the order quantities for only 1 instance. That would result in about 1.7 years of required computation time for all the instances having a 5 periods of shelf life. In contrast to the SDP model, our proposed heuristic can generate the replenishment period vectors for all the 216 instances in just 1.7 seconds. Taking the simulation runs that determine the order quantity into account, the calculation of all the parameters for the whole instance set takes less than 30 minutes. Since we calculate the order quantities during the simulation runs, the average simulation runtime for our heuristic is longer (7.59s) compared to that of the SDP model (4.3s). When we conduct a sensitivity analysis on the algorithm and the simulation runtimes for both the SDP and our proposed

heuristic, we clearly see from Table 4.3 that all of them are subject to increase as M increases and there are slight differences in the simulation runtimes between different demand patterns, while changes in all the other parameters do not cause a significant change in the runtimes. Similar to the increase in the average algorithm runtimes, the average simulation runtimes increase from 4.6s to 9.54s for our proposed heuristic, respectively.

Testing the Heuristic on Known Instances in the Literature

We test our heuristic on a modified version of the another instance set in the literature, taken from [PWHHvdV14]. Since their mean demand parameters are too large (e.g., 800, 950), it is practically intractable to implement the optimal SDP model with these parameters and without using some techniques and approximations (e.g., extrapolation). Therefore, we scale down the demand mean and the fixed setup cost parameters by 40 to implement the SDP model without any technique. They consider normally distributed demand in their study, so we assume the demand follows a normal distribution. For this experiment, we select a subset of their instance set and consider the parameters given in Tables 4.4 and 4.5. In total, there are 12 instances.

Table 4.6 shows that the average optimality gap is slightly higher at 3.21%. $M = 3$ does not tend to have a greater optimality gap on the contrary to the main experimental setup. This observation shows that the increase in the shelf life does not always result in deviations from the optimality. It might be explained by the observation that the average cycle length-to-shelf ratio remains similar for both M values. Considering both the experimental setups, we can conclude that the gap tends to increase as the average cycle length-to-shelf ratio decreases. Moreover, $a = 37.5$ has the least average optimality gap and the gap does not tend to increase as a increases or decreases. Finally, the heuristic has a tendency to perform better when the coefficient of variation is lower.

Testing the Runtimes on Instances with Larger Demand and Lifetime

We test the performance of the heuristic in terms of computation times for longer shelf lives and larger demand means. We consider the same holding, penalty, outdating cost parameters as in Table 4.4. We take the fixed setup cost parameters $a = 1500$ and the demand means given in Table 4.7 from [PWHHvdV14], which we scaled down by 40 in Section 4.6.2. The shelf lives take value from the set $M \in \{2, 4, 6, 8, 10\}$. The time horizon lengths take val-

Table 4.3: Comparison of the SDP and the heuristic, penalty cost case, with respect to each parameter and the average computation times (in seconds) per instance (5,000 simulation reps each). Recall that the heuristic was implemented in MATLAB and the SDP model was implemented in C, and the SDP runtimes would be much longer in MATLAB.

Parameter	Avg. Opt. Gap Percent Diff	95% CI Percent Diff	Avg. Heuristic Total Cost	Avg. SDP Total Cost	Avg. Heur. Cycle Length	Avg. SDP Runtime (s)	Avg. SDP Sim. Runtime (s)	Avg. Heuristic Runtime (s)	Avg. Heur. Sim. Runtime (s)
Total avg	1.87%	[1.73, 2.02]	852.92	837.58	2.26	1,663	4.31	0.008	7.59
Stationary	1.61%	[1.27, 1.95]	867.19	853.51	2.42	1,662	4.48	0.009	7.60
Increasing	1.48%	[1.23, 1.74]	867.33	854.35	2.19	1,663	4.15	0.007	6.65
Decreasing	1.64%	[1.39, 1.89]	834.57	821.65	2.28	1,662	4.60	0.009	8.38
Cyclic	1.68%	[1.40, 1.97]	849.37	835.79	2.20	1,663	4.29	0.008	7.56
Seasonal	2.59%	[2.18, 2.99]	856.41	835.20	2.13	1,664	4.12	0.007	7.84
Random	2.25%	[1.82, 2.67]	842.67	824.94	2.32	1,661	4.24	0.007	7.52
$M = 2$	1.07%	[0.89, 1.26]	883.46	873.85	1.81	0.19	2.96	0.002	4.69
$M = 3$	1.88%	[1.68, 2.08]	840.20	824.86	2.36	33.9	4.96	0.007	8.54
$M = 4$	2.66%	[2.44, 2.88]	835.12	814.02	2.60	4,954	5.02	0.013	9.54
$a = 25$	2.13%	[1.91, 2.35]	713.10	698.59	1.74	1,663	4.25	0.008	7.43
$a = 50$	1.86%	[1.60, 2.12]	860.61	844.82	2.41	1,663	4.30	0.008	7.95
$a = 75$	1.64%	[1.38, 1.90]	985.06	969.32	2.62	1,662	4.39	0.008	7.39
$c = 1$	1.85%	[1.61, 2.09]	656.12	644.74	2.33	1,663	4.32	0.008	7.20
$c = 3$	1.90%	[1.74, 2.06]	1,049.73	1,030.41	2.18	1,662	4.30	0.008	7.98
$p = 10$	1.89%	[1.68, 2.09]	837.66	822.41	2.30	1,662	4.29	0.007	7.15
$p = 20$	1.86%	[1.66, 2.07]	868.19	852.74	2.21	1,663	4.34	0.008	8.04

Table 4.4: Input parameters for the experiments

Input parameters	**Values**
Fixed setup cost (a)	{12.5, 37.5, 50}
Unit ordering cost (c)	2
Unit holding cost (h)	0.5
Unit penalty cost (p)	5
Unit outdating cost (w)	0
Shelf life (M)	{2, 3}
Planning horizon (T)	12
Coefficient of variation (CV)	{0.25, 0.333}

Table 4.5: Demand means. Modified from the parameters in [PWHHvdV14]

	Average in period t											
.	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**
	20	23.75	5	22.5	20	3.75	16.25	20	22.5	7.5	3.75	15

Table 4.6: Comparison of the SDP and the heuristic, penalty cost case, with respect to each parameter, penalty cost case.

Parameter	**Avg. Opt. Gap Percent Diff**	**95% CI Percent Diff**	**Avg. Heuristic Total Cost**	**Avg. SDP Total Cost**	**Avg. Cycle Length**
Total avg	3.21%	[2.54, 3.89]	691.34	669.77	1.95
M = 2	3.52%	[2.60, 4.44]	719.42	694.38	1.64
M = 3	2.91%	[1.90, 3.92]	663.26	645.16	2.40
a = 12.5	3.58%	[2.67, 4.50]	559.90	540.51	1.60
a = 37.5	2.75%	[1.96, 3.54]	722.68	703.28	2.18
a = 50	3.31%	[1.55, 5.07]	791.43	765.50	2.18
CV = 0.25	2.61%	[1.68, 3.54]	675.33	658.09	1.95
CV = 0.333	3.82%	[3.05, 4.58]	707.34	681.45	1.95

ues from $T \in \{12, 24, 48\}$. In total there are 30 instances.

As seen from Table 4.8, the heuristic runtimes tend to increase as M grows, from 0.0013 seconds for $M = 2$ to 0.0777 seconds for $M = 10$. As T increases, the runtimes tend to increase linearly for smaller M values, but they tend to increase exponentially as M gets larger. The increase in M has a tendency to result in exponential increases in runtimes, e.g., from 0.0019 seconds when $M = 2$ to 0.1635 seconds when $M = 10$ for $T = 48$. However, the heuristic outperforms the optimal SDP solution in terms of order of magnitude of the growth in computation times (e.g. when $M = 5$ only 8 milli-seconds for the

Table 4.7: Demand means taken from [PWHHvdV14]

	Average in period t											
.	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**
	800	950	200	900	800	150	650	800	900	300	150	600

Table 4.8: Comparison of the average heuristic runtimes (seconds) for different product lifetimes, penalty cost case.

Parameter	**Lifetime** (M)				
	2	**4**	**6**	**8**	**10**
Total avg	0.0013	0.0049	0.0156	0.0382	0.0777
T = 12	0.0005	0.0020	0.0041	0.0073	0.0106
T = 24	0.0015	0.0041	0.0133	0.0302	0.0591
T = 48	0.0019	0.0087	0.0294	0.0771	0.1635
CV = 0.25	0.0014	0.0049	0.0157	0.0395	0.0797
CV = 0.333	0.0012	0.0049	0.0154	0.0369	0.0758

heuristic and more than 1 week for the SDP). Moreover, the heuristic tends to have similar performance for different instances of the fixed setup cost and the coefficient of variation as M increase.

4.7 Conclusion

We have studied a perishable inventory system consisting of items which outdate after a fixed number of periods. Due to the dimensionality of the inventory composition state, it becomes technically impossible to find the optimal parameters for longer shelf lives via the SDP, considering that the SDP model might be extremely demanding in terms of the computation time and also the memory requirements. In addition, several complexity issues related to formulating the problem as a network graph arise, Therefore, we develop time-efficient and fast heuristic inspired by the well-known Wagner-Whitin algorithm and the Silver-Meal heuristic, in order to obtain the best path on the network while using the ending inventory composition state as a parameter to maintain the lowest average cost in the ordering decision. Our numerical experiments show that using the order timings and quantities generated by our proposed heuristic, we obtain the total cost values with a very small optimality gap percentage (1.87% error on average) just in milliseconds.

By modeling the problem as a deterministic-equivalent network graph, we relax

the state space and make it more manageable in terms of runtimes and state space size. Since the network graph approach may not be sufficient to represent the different multi-dimensional inventory composition states, we employ additional procedures and make decisions based on the inventory states which help keeping the average cost since the beginning of the planning horizon at minimum. Moreover, the effect of the state space dimensionality caused by the amount of shelf life has a minimal effect on our heuristic.

Our heuristic quickly generates near-optimal results even for the items with much longer shelf lives. It is very convenient in terms of implementation and computational complexity. It simultaneously allows flexibility to make ordering decisions by taking the observed demand into account and avoids replenishment timing decision set-ups and reviews which might be required for every period in the rolling-horizon frameworks. Therefore, it could be easily applied to most real-world inventory systems for perishable goods and utilised by executives and decision-makers.

Chapter 5

Determining the Policy Parameters for Perishable Inventory Management Under Service Level Constraints

In this chapter, we analyse the minimal α-service level variation of the heuristic we propose in Chapter 4. We study this variation of the perishable inventory problem in our paper currently work in process. We modify our proposed heuristic to determine the replenishment plan and the order quantities for the minimal α-service level case version of the perishable inventory problem. After we provide an introduction in Section 5.1, we describe our problem Section 5.2, then provide a mathematical model and an SDP formulation. In Section 5.3 we explain the network representation for this case. In Section 5.4 we describe and explain the service level version of our heuristic. We provide a numerical study in Section 5.5 and conclude this chapter in Section 5.6.

5.1 Introduction

The minimal α-service level is a widely used measure in inventory control to ensure that the probability of no stockout in each period is at least α. Instead of applying a penalty for shortages, it indirectly controls stockouts by enforcing a minimum probability threshold to be satisfied. This is particularly useful in practical settings to guarantee customer satisfaction to a high level of degree,

in comparison to assigning a penalty for each unmet demand.

This type of service level is especially important in practical applications, such as retail, healthcare, and food supply chains, where unmet demand may result in high customer dissatisfaction or operational issues. For instance, grocery stores often operate under high service level targets to maintain shelf availability, and hospitals need to ensure a reliable supply of perishable medical products, such as blood or vaccines. Unlike the penalty cost approach, the α-service level provides a clearer and more interpretable target, which makes it easier to align with service-oriented objectives and operational performance metrics used in practice.

A policy focused only on minimising total cost may not work well in these cases. It might accept less service in periods with high uncertainty if it leads to lower total costs. In critical settings such as healthcare or food supply chains, even short periods of high stockout might result in serious problems. The α-service level prevents these kind of consequences by ensuring that the probability of stockout is no less than a specific level.

We study the variation of the perishable inventory problem under a minimal α-service level constraint. We modify the heuristic proposed in Chapter 4 to determine the replenishment periods and order quantities while ensuring the service level requirement is satisfied in each period. Our extensive computational experiments that use the near-optimal SDP solution as a benchmark show that the heuristic finds near-optimal solutions in significantly reduced computation times.

5.2 Problem Description and Our Model

In this chapter, we consider a similar setting to the problem setting described in Chapter 4, except the way to handle stockouts. Different from Chapter 4 in which we represented each backlogged item in terms of cost, our aim in this chapter is to ensure that the customer demand is met no less than a certain probability in every period. Therefore, we remove the penalty cost from the problem description and add a minimal α-service level constraint to guarantee that the non-stockout probability is no less than α.

We study a single-item single-location perishable inventory system with periodic review and over a finite planning horizon of T periods. Demand distributions

are stochastic, subject to change over time, and demand realisations in each Period t, d_t, are independent random variables with known probability density and cumulative distribution functions $f_t(d_t)$ and $F_t(d_t)$, respectively. In each replenishment period, there is zero lead time. All items have a fixed shelf life of $M \geq 2$ periods and the age of a new item is $b = 0$. An item is outdated and must be discarded when its age is $b \geq M$. We assume full backlogging. The inventory level for the items of Age b at the end of Period t is denoted by $I_{t,b}$. The inventory system works according to the FIFO issuing policy.

There is a fixed setup cost per order a and a variable unit ordering cost c linear in the order quantity Q_t, incurred at the beginning of each replenishment period. Moreover, a variable unit holding cost h, which is linear in the ending inventory except outdated items $\sum_{b=0}^{T-2} I_{t,b}$, is charged at the end of each Period t. Finally, a linear outdating cost per outdated items w is charged if there is waste at the end of Period t. Positive outdating cost might require expected waste estimation techniques or further exact expected waste calculations which have no explicit formulation in the literature. For convenience, we assume outdated items (e.g. food in supermarkets which are past their expiration date) incur no costs $w = 0$ except the loss of revenue due to unsold and disposed outdated items.

5.2.1 Mathematical Model

We formulate our problem as a stochastic programming model in which the random variable is d_t, demand realisations in each Period t. The model is formulated as follows:

$$\min E\{TC\} = \int_{d_T} \dots \int_{d_1} \sum_{t=1}^{T} \Bigg(\mathbb{1}(Q_t) + h(I_{t,0})^+ + h \sum_{b=1}^{M-2} I_{t,b} + w I_{t,M-1} \Bigg) f_1(d_1) \dots f_T(d_T) \mathrm{d}d_1 \dots \mathrm{d}d_t \tag{5.1}$$

subject to

$$\mathbb{1}(Q_t) = \begin{cases} a + cQ_t, & \text{if } Q_t > 0 \\ 0, & \text{otherwise} \end{cases} \qquad t = 1, 2, \dots, T. \tag{5.2}$$

$$\mathbb{P}\left\{ \sum_{b=0}^{M-1} I_{t,b} \geq 0 \right\} \geq \alpha, \quad t = 1, 2, \dots, T. \tag{5.3}$$

$$I_t = \sum_{b=0}^{M-1} I_{t,b} = \sum_{b=0}^{M-2} I_{t-1,b} + Q_t - d_t, \quad t = 1, 2, ..., T. \tag{5.4}$$

$$I_{t,0} = Q_t - \left(d_t - \sum_{b=0}^{M-2} I_{t-1,b} \right)^+, \quad t = 1, 2, ..., T. \tag{5.5}$$

$$I_{t,b} = \left(I_{t-1,b-1} - (d_t - \sum_{j=b}^{M-2} I_{t-1,j})^+ \right)^+, \quad t = 1, 2, ..., T; b = 1, ..., M-1. \tag{5.6}$$

$$I_{0,b} = 0, \quad b = 0, ..., M-1. \tag{5.7}$$

$$I_{t,b} \geq 0, \quad t = 1, 2, ..., T; b = 1, ..., M-1. \tag{5.8}$$

$$Q_t \geq 0, d_t \geq 0, \quad t = 1, 2, ..., T. \tag{5.9}$$

Note that $(x)^+ = \max(x, 0)$ and $(x)^- = \min(x, 0)$.

The model has previously been studied by [PWHHvdV14]. Different from the model given in Chapter 4, the penalty cost component of the objective function (5.1) is removed and a service level constraint (5.3) is added. The objective function given in Equation (5.1) minimises the expected total cost throughout the planning horizon. The constraint (5.2) defines the total ordering cost function in Period t. The constraint (5.3) formulates the service level requirement such that the probability that there is no stockout at the end of each Period t is at least α. The inventory levels of all ages at the end of Period t are calculated according to (5.4). The FIFO constraints are given in constraints (5.5) and (5.6). Constraint (5.7) and (5.8) state that the planning horizon starts with zero inventory levels for all ages. Constraint (5.8) implies the ending inventory levels of Ages $b > 0$ are non-negative and only the number of newly-ordered items can be negative at the end of Period t if the stock on hand is too small to meet demand. According to constraint (5.9), the order quantity and the demand must be non-negative.

From (5.8), we can reformulate the service level constraint given in (5.3) as

$$\mathbb{P}\{I_{t,0} \geq 0\} \geq \alpha, \quad t = 1, 2, ..., T. \tag{5.10}$$

since fresh items are the only items allowed to be negative. It is due to the FIFO issuing policy such that if no older item is left we start using newer items.

5.2.2 Stochastic Dynamic Programming Approach

We employ the SDP approach which was proposed by [HHRPW12]. However, the SDP approach is not necessarily optimal. According to [PWH15, PWHHvdV14], it could be explained as the SDP generates ordering policies that overachieve the service level requirements due to the conditional behaviour of the service level constraints, such that

$$\mathbb{P}\left\{\left(\sum_{b=0}^{M-1} I_{t,b}|\mathbf{I}_{t-1}\right) \geq 0\right\} \geq \alpha, \quad t = 1, 2, ..., T$$

where $\mathbf{I}_{t-1}$ is defined in Equation 4.9. The SDP does not and cannot take the difference on probability of occurrence of all inventory states into account. Regardless of the probability of the occurrence of each possible $\mathbf{I}_{t-1}$, the SDP meets the required service level in each period. This limitation arises because SDP computes decisions based on expected future costs conditioned on the current state, rather than explicitly differentiating long-term probabilities of transitioning into each individual state. Therefore, it tends to overachieve the service levels and that might yield higher total costs. This also explains as α grows, the SDP tends to approach optimality.

In the service level case, the definitions of the inventory composition, inventory state, state transition function, and waste function are the same as in the penalty cost case, which are explained in Section 4.2.3 in detail.

1. **Immediate Cost Function:** The immediate cost function differs in this case from the penalty cost case. We exclude the penalty cost component and formulate the single-period immediate cost function as follows:

$$EC(\mathbf{I}_{t-1}, Q_t) = q(Q_t) + \mathbb{E}\left\{h(I_{t,0})^+ + h\sum_{b=1}^{M-2} I_{t,b} + wo(\mathbf{I}_{t-1}, d_t)\right\}. \quad (5.11)$$

2. **Objective Function:** Using the immediate cost function in Eq. 5.11, we write the objective function as given below via the Bellman equation with the value function $V(.)$:

$$V_t(\mathbf{I}) = \min_{Q}\left(EC(\mathbf{I}, Q) + \mathbb{E}(V_{t+1}(\Phi(\mathbf{I}, Q, d_t)))\right). \quad (5.12)$$

Pseudocode:

Algorithm 5 describes the pseudocode of the SDP formulation based on the service level constraint. It searches though all the possible starting inventory compositions and the order quantities in order to find the optimal parameters that minimise the total cost. This SDP formulation differs from that of the penalty cost case. If the initial inventory is sufficient to meet the service level, no order is placed, as seen from Line 18 of Algorithm 5. Otherwise, an ordering decision is made and the algorithm searches for the order quantity that minimises the total cost while meeting the service level requirement.

Algorithm 5 SDP – Service Level

1: **Input:** inventory bounds $min_inv_b,\ max_inv_b\ (b = 1, \ldots, M-1)$; service level α; demand distribution d_t; maximum order max_order
2: **Output:** order–up-to levels S_t and order quantities q_t for $t = 1, \ldots, T$
3: **for** $b_1 = min_inv_1$ **to** max_inv_1 **do**
4: **for** $b_2 = 0$ **to** max_inv_2 **do**
5: $\vdots$
6: **for** $b_{M-1} = 0$ **to** max_inv_{M-1} **do**
7: $C_{T+1}(b_1, b_2, \ldots, b_{M-1}) \leftarrow 0$
8: $\vdots$
9: **for** $t = T$ **downto** 1 **do**
10: **for** $b_1 = min_inv_1$ **to** max_inv_1 **do**
11: **for** $b_2 = 0$ **to** max_inv_2 **do**
12: $\vdots$
13: **for** $b_{M-1} = 0$ **to** max_inv_{M-1} **do**
14: $\mathbf{I}_{t-1} \leftarrow (b_1, b_2, \ldots, b_{M-1})$
15: $x \leftarrow b_1 + b_2 + \cdots + b_{M-1}$
16: $C_t(\mathbf{I}_{t-1}) \leftarrow \mathbb{E}\{wo(\mathbf{I}_{t-1}, d_t)\}$
17: **if** $x \geq F^{-1}(\alpha)$ **then**
18: $q \leftarrow 0$ ▷ sufficient inventory to meet service level
19: **else**
20: **for** $q = 0$ **to** max_order **do**
21: $expected_cost \leftarrow \mathbb{E}\Big\{h(I_{t,0})^+ + h\sum_{b=1}^{M-2} I_{t,b}\Big\}$
22: $+ \mathbb{E}\Big\{C_{t+1}\Big(\Phi(\mathbf{I}_{t-1}, q, d_t)\Big)\Big\}$
23: **if** $expected_cost \leq C_t(\mathbf{I}_{t-1})$ **then**
24: $C_t(\mathbf{I}_{t-1}) \leftarrow expected_cost$
25: $S_t \leftarrow x + q$
26: $q_t \leftarrow q$

5.3 Network Representation

Recall the example in Section 4.2.3. As the shelf life M increases, the dimensionality of $\mathbf{I}_{t-1}$ increases and it leads to an exponential growth in the number of possible inventory states. For instance, assume there are 100 possible values for the number of items of each age. The number of possible states grows 100 times when we increase the shelf life by 1, e.g., it is 100^2 when $M = 2$ and becomes 100^3 for $M = 3$. As a result of the immediate growth in the state space it quickly becomes unmanageable and computationally intractable.

We employ the same network state space representation technique as in Chapter 4, which is the modified version of the work in [RTHP11] for perishable inventory systems. Let us consider a T-period planning horizon. Each Node $i \in \{0, ...T-1\}$ represents the initial inventory state in Period $i+1$ and the Node T represents the ending inventory state of the planning horizon.

Expected Cycle Cost: The expected cycle cost for a given replenishment cycle $R(i,j)$, $i = 1, 2, ..., T$ and $j = i, i+1, ..., (T, i+M-1)^-$, is denoted by $c(i,j)$, such that

$$c(i,j) = \mathbb{1}(Q_i) + \sum_{t=i}^{j} \left(h \sum_{b=1}^{M-1} \mathbb{E}(I_{t,b})^+ + w\mathbb{E}(I_{t,M-1}) \right) \tag{5.13}$$

Note that we remove the penalty cost component from the formulation of $c(i,j)$.

Figure 5.1 illustrates a network representation of a perishable inventory system with $T = 3$ periods and $M = 2$ periods with all possible replenishment cycles. Note that the arc $(0, 3)$ is not included in the graph since it represents a 3-period replenishment cycle $R(1, 3)$. For this case, let us assume each node represents the total inventory level of all ages and there is no outdating.

Different from non-perishable inventory systems, the total inventory level might correspond to different inventory compositions which give the same total inventory level. Moreover, the starting inventory level in Period $t+1$ and the ending inventory level in Period t may not be equal if there are outdating items at the end of Period t. These issues make the state space network representation for perishable inventory systems more complicated. We also aim to overcome these challenges in our heuristic.

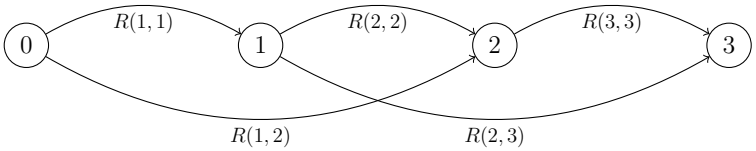

Figure 5.1: The network representation of all possible replenishment cycles of a perishable inventory model with $T = 3$ and $M = 2$. Notice that each Node $i = 1, ..., T - 1$ represent the end of Period i and the beginning of Period $i + 1$, Node 0 and Node T represent the beginning and the ending of the planning horizon, respectively, and each arc $(i, j) : i = 0, ..., T - 1, j = 1, ..., T, i > j$, represents $R(i + 1, j)$.

5.4 Heuristic Approach for the Service Level Case

In this section, we introduce the variation of our heuristic for the minimal α-service level case and explain it in detail.

Our heuristic aims to determine the replenishment Periods R_t and the order quantities Q_t in a similar manner to the static-dynamic uncertainty strategy of [BT88]. The procedure for determining the replenishment schedule in the service level variation is structurally identical to the penalty cost variation, introduced in Chapter 4, and follows the same two-stage heuristic logic. However, it incorporates the modified order-up-to level definition introduced in Section 5.4.1, which accounts for the service level requirement and the expected outdating within each replenishment cycle. Since the Wagner-Whitin algorithm makes a cost-based decision and may not take the effects of different inventory compositions with the same total, we need another decision rule based on the Silver-Meal heuristic. Hence, we decide to keep the average cost per period throughout the planning horizon at a minimum level while determining the replenishment schedule.

5.4.1 Calculating the Order-up-to Levels

Calculating the order-up-to levels is the key difference in the service level variation of the heuristic we proposed in Chapter 4 and one of the main contributions of the service level variation. In this variation of our heuristic, the order-up-to level for the replenishment cycle $R(i, j)$ must meet the total amount of expected demand with a probability of at least α. We denote the total demand for $R(i, j)$ by d_{ij}. Then the probability density and the cumulative distribution functions for d_{ij} are denoted by $f_{ij}(d_{ij})$ and $F_{ij}(d_{ij})$, respectively. Note that $d_{ij} = d_i + d_{i+1} + ... + d_j$ and $F_{ij}^{-1}(.)$ is the inverse cdf of α. Since outdated items

might cause additional stockouts and result in failure to meet the required service level, we have to take it into account.

In contrast to non-perishable inventory models under service level constraints, ordering up to $F_{ij}^{-1}(\alpha)$ might not be sufficient. This is due to outdating that occurs within the replenishment cycle. If there are outdated items at the end of each period, they lower the available inventory to meet the required service level, and it results in stockouts. Therefore, we must take the expected outdating into account while calculating the optimal order-up-to level $S_{i,j}^*$. If the number of the oldest items at the beginning of Period j is less than customer demand they will be used, otherwise all the remaining items are outdated. Therefore, the outdated items at the end of Period j, the last period of $R(i,j)$, do not affect stockouts.

The optimal order-up-to level for $R(i,j)$ is expressed by

$$S_{i,j}^* = F_{ij}^{-1}(\alpha) + \sum_{k=i}^{j-1} \mathbb{E}(I_{k,M-1}) \tag{5.14}$$

However, as we stated earlier in this dissertation and according to [Nah75], there is no explicit formulation to calculate the expected outdating, and there is no recent study to formulate it. Therefore, we employ the EWA heuristic proposed in [BVD09], as in the penalty cost variation of our heuristic. Recall that their approach estimates the expected outdating at the end of each period in a deterministic-equivalent framework.

5.4.2 Step 1: Determining the Replenishment Schedule

At the end of each Period $t = 1, \ldots, T$, we consider all the replenishment cycles $R(i,t)$ that can end in Period j, with lengths of at most M periods (e.g., $i = \max(1, t - M + 1), \ldots, t$). For each possible cycle, we calculate the expected cycle cost $c(i,t)$ using Eq. (5.13), based on the estimated inventory composition and expected outdating. The order-up-to level $S_{i,t}$ for each cycle is determined using Eq. (5.14), ensuring both the service level constraint and the expected outdating are satisfied.

Next, we estimate if the inventory at the end of Period t is sufficient to meet future demand for several periods without placing a new order or not. If so, we compute the holding and outdating costs until we use all the inventory. Otherwise, we select the replenishment cycle that provides the minimum average

cost per period by applying a modified version of the Silver-Meal heuristic.

The best path up to Period t is updated recursively by selecting the replenishment cycle $R(i^*, t)$ that minimises the average total cost up to a coverage horizon $T_{i,t}$:

$$AC(T_{i,t}) = \frac{TC(i-1) + c(i,t) + c'(t+1, T_{i,t})}{T_{i,t}}, \tag{5.15}$$

$$i^* = \arg\min_i AC(T_{i,t}), \tag{5.16}$$

$$TC(t) = TC(i^* - 1) + c(i^*, t), \tag{5.17}$$

where

$$c'(t+1, T_{i,t}) = \mathbb{1}(Q_i) + \sum_{k=t+1}^{T_{i,t}} \left(h \sum_{b=1}^{M-1} \mathbb{E}(I_{k,b})^+ + w\mathbb{E}(I_{k,M-1}) \right).$$

We continue until the end of the planning horizon. As a result, we obtain the complete replenishment cycle set $\mathcal{R}_T$ and the binary replenishment vector $\mathbf{Y}_T$ which define the replenishment schedule used in the second step of the heuristic.

The pseudocode of the algorithm for the process of determining the replenishment periods is given in Algorithm 6. In the service level case, the order-up-to level is determined by Equation 5.14, given in Line 10. In addition, the penalty cost component is removed from the cost function, as seen from Line 11.The differences from the penalty cost case can be listed as the procedure to calculate order-up-to levels and the replenishment cycle cost formulation (Lines 10 and 11 in Algorithm 4, respectively, Page 100). Recall that in the penalty cost case, order-up-to levels are determined in the multi-period newsvendor setting, and there is a penalty cost component in the replenishment cycle cost formulation.

5.4.3 Step 2: Calculating the Order Quantities

After we determine the replenishment schedule, we calculate the actual order quantities based on the observed inventory state in each replenishment period. While the schedule is pre-determined under a deterministic-equivalent setting, demand uncertainty may result in deviations from the expected inventory compositions. Therefore, we use a dynamic strategy to compute the order quantities based on the realised inventory composition $\mathbf{I}_{t-1}$ at the beginning of each replenishment period.

The procedure we employ is given as follows: For each replenishment cycle $R(i,j) \in \mathcal{R}_T$ and given the observed $\mathbf{I}_{i-1}$, we determine the order quantity Q_i that is sufficient to meet the order-up-to level $S^*_{i,j}$, which meets both the minimal α-service level and the expected outdating within the cycle that is calculated via the EWA heuristic of [BVD09]. Hence, we ensure that we avoid stockouts with a probability of at least α throughout each replenishment cycle while taking both the expected demand and outdating into account.

We repeat this process for each cycle $R(i,j)$ in the replenishment schedule $\mathcal{R}_T$ until the end of the planning horizon. The combined result of dynamically determined order quantities and the pre-determined replenishment schedule yields a near-optimal solution under the α-service level constraint.

5.4.4 An Example

We provide a simple example for our heuristic. Let us assume $M = 2$, $N = 4$, $I_{0,b} = 0, \forall b = 0, ..., M-1$, and the demand in each Period t has a Poisson distribution with rates $\lambda_t = (6, 9, 4, 8)$. The fixed ordering cost is $a = 20$, the unit ordering, holding, and outdating costs are $c = 1$, $h = 1$, and $w = 0$, respectively. In this problem, $\mathbf{I}_t$ has only one element and it is the number of items of Age 0, since $M = 2$. Note that the parameters are the same as in the example given in Chapter 4. Similar to the example for the penalty cost case, the deterministic-equivalent approximation of the EWA heuristic [BVD09] is applied in the service level case.

The differences between the two examples are given as follows: The penalty cost is replaced by an $\alpha = 0.95$ service level requirement. Therefore, the replenishment cycle order-up-to level calculation is performed by the Equation 5.14, while the example given in Section 4 follows the multi-period newsvendor procedure discussed in Section 4.4.

Period 1:

For period 1, the best path between the nodes 0 and 1 is the arc $(0,1)$ which is the only arch between these nodes and represents $R(1,1)$. Hence, the set $\mathcal{R}_1 = \{R(1,1)\}$. The order-up-to level for $R(1,1)$ is $S_{11} = 10$ which is sufficient to meet the 95% service level. Since $\mathbf{I}_0 = (0)$, we order $Q_1 = 10$ items. After the expected demand realisation $\mathbb{E}\{d_1\} = 6$ occurs, we have four items of Age 0 on hand at the end of period 1 and $\mathbf{I}_1 = (4)$. The total cost for $\mathcal{R}_1$ is calculated as

Algorithm 6 Heuristic – Service Level

Data: Directed state–space graph with arcs for at most M-period cycles
Input: Opening inventory $\mathbf{I}_0 = (I_{1,0}, \ldots, I_{M-1,0})$, expected demand $d = (d_1, \ldots, d_T)$
Result: Binary replenishment-cycle vector $\mathbf{Y}_T$
$TC \leftarrow$ zeros(T) ▷ total-cost vector
$\mathcal{R} \leftarrow \emptyset$ ▷ set of chosen cycles
$TC(0) \leftarrow 0$
for $j = 1$ **to** T **do**
    $avgcost \leftarrow \infty$
    for $i = \max\big(1,\, j - M + 1\big)$ **to** j **do**
        $S^*_{i,j} \leftarrow F^{-1}_{ij}(\alpha) \;+\; \sum_{k=i}^{j-1} \mathbb{E}(I_{k,M-1})$
        $c(i,j) \leftarrow \mathbb{1}(Q_i) \;+\; \sum_{t=i}^{j}\Big(h \sum_{b=1}^{M-1} \mathbb{E}(I_{t,b})^+ + w\,\mathbb{E}(I_{t,M-1})\Big)$
        Compute $T_{i,j}$
        if $T_{i,j} > j$ **then**
            $c'(j+1, T_{i,j}) \leftarrow$ cost of holding remaining inventory (stock is sufficient)
        else if $T_{i,j} = j$ **then**
            $T^*_{i,j} \leftarrow \arg\min_{T'_{i,j} > j} \dfrac{c(j+1, T'_{i,j})}{T'_{i,j} - j}$
            $T_{i,j} \leftarrow T^*_{i,j}$
            $c'(j+1, T_{i,j}) \leftarrow c(i+1, T_{i,j})$
        $temp \leftarrow \Big(TC(i-1) + c(i,j) + c'(j+1, T_{i,j})\Big)\Big/T_{i,j}$
        if $temp < avgcost$ **then**
            $avgcost \leftarrow temp$
            $TC(j) \leftarrow TC(i-1) + c(i,j)$
            $\mathcal{R}(j) \leftarrow \{\mathcal{R}(i-1), R(i,j)\}$
            $\mathbf{Y}_i \leftarrow (\mathbf{Y}_{i-1}, 1)$
return $\mathbf{Y}_T$

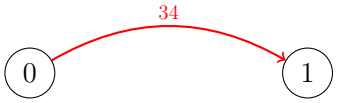

Figure 5.2: The best path at the end of period 1 with the corresponding cycle cost values.

$TC(1) = a + cQ_1 + h\sum_{b=0}^{M-2} I_{1,b} + wI_{1,M-1} = 20 + 10 + 4 + 0 = 34$. At the end of period 1, we highlight the arc $(0, 1)$ as seen in Figure 5.2.

Period 2:

For $t = 2$, we look for all the arches ending in the node 2. There are two possible arches: $(0, 2)$ and $(1, 2)$. Then, placing a 2-period order covering periods 1 and 2 at the beginning of period 1, or placing two one-period orders at the beginning of periods 1 and 2 are our two possible options.

In the first option, the order-up-to level that satisfies the service level requirement is $S_{1,2} = 22$. At the end of the first period and after the expected demand $d_1 = 6$ is realised, $\mathbf{I}_1 = (16)$. We start period 2 with 16 items of Age 1. At the end of period 2 there are $I_{2,1} = 16 - 9 = 7$ items, and all these 7 items are outdated since $M = 2$. therefore no inventory is held for the following periods. Hence, $TC(2) = 20 + 22 + 16 + 0 = 58$. We do not have enough inventory to cover the next period's demand. Therefore, we choose the replenishment decision with the least average cost to calculate $c'(3, T_{i,j})$ using the modified Silver-Meal heuristic. The selected decision covers the next 2 periods and sufficient for the demand until the end of Period $T_{1,2} = 4$. , $c'(3, T_{i,j}) = 52$, , and $AC(T_{1,2}) = (TC(2)+AC(4))/T_{1,2} = (58+52)/(2+2) = 27.25$ for $\mathcal{R}_2 = \{R(1, 2)\}$.

In the second option, we place a new 1-period order after $R(1, 1)$ and using $\mathbf{I}_1 = (4)$, so we start period 2 with 4 items of Age 1. The inventory position to meet the service level requirement is $S_{2,2} = 14$ and we order $Q_2 = 10$ items. After $\mathbb{E}\{d_2\} = 9$ is realised, $I_{2,0} = 5$ and $I_{2,1} = 0$. The resulting set of replenishment cycles is $\mathcal{R}_2 = \{R(1, 1), R(2, 2)\}$ and the total cost for $\mathcal{R}_2$ becomes $TC(2) = TC(1) + 20 + 10 + 5 = 69$. We have sufficient inventory for $\mathbb{E}\{d_3\} = 4$. Then $T_{i,j} = 3$ and $c'(3, T_{i,j}) = 1$. The final average cost is $AC(T_{2,2}) = (TC(2) + AC(3))/T_{2,2})(69 + 1)/3 = 23.33$.

Since $AC(T_{2,2}) \leq AC(T_{1,2})$, we choose the second option for the best path at the end of period 2 and the replenishment cycle set $\mathcal{R}_2 = \{R(1, 1), R(2, 2)\}$, despite the fact that the first option has less total cost. If we chose the first option, the previous path $R(1, 1)$ would be updated to $R(1, 2)$. Similarly, the future inventory compositions and the path decisions may alter the previous path (e.g. $\mathcal{R}_3 = \{R(1, 2), R(3, 3)\}$ is possible).

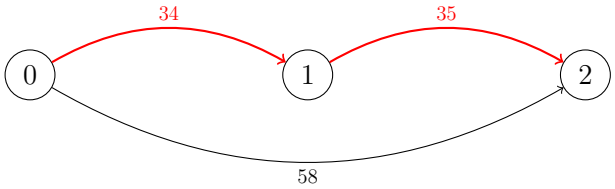

Figure 5.3: The best path at the end of period 2 with the corresponding cycle cost values.

Following periods:

We repeat the same procedure for the following periods recursively. At the end of the planning horizon, the final replenishment cycle set is

$$\mathcal{R}_4 = \{R(1,1), R(2,2), R(3,4)\},$$

and the vector is $\mathbf{Y}_4 = (1,1,1,0)$. Using $\mathbf{Y}_4$ as an input, we find the order quantities via performing a simulation run with 5000 replications and find the total cost is $TC(4) = 117.6$ on average. The order quantities for a specific period might differ in each replication since it is dependent on actual demand observations and a pre-determined order-up-to level.

5.5 Numerical Experiments

In this section, we discuss the numerical experiments that we conducted to measure the performance of our heuristic under service level constraints. We implement our heuristic and the SDP model in MATLAB for our inventory problem. We perform our numerical experiments on a Macbook Pro with an Apple M1 Max 10 core CPU and a 64 GB RAM. Similar to the penalty cost version discussed in Section 4.6, we first generate the pre-determined replenishment schedule, and using this schedule and observed demand values we determine the order quantities.

5.5.1 Experimental Setup

In the service level version of our numerical experiments, we use the same set of parameters and demand patterns as in Chapter 4 for $M \in \{2,3\}$, given in Tables 5.1 and 5.2. The fixed ordering cost parameters are $a \in \{25,50,75\}$ and the unit ordering cost values are $c \in \{1,3\}$. The unit holding and outdating costs are $h = 1$ and $w = 0$, respectively. The horizon length is set to $T =$

Table 5.1: Input parameters for the experiments

Input parameters	Values
Fixed setup cost (a)	{25, 50, 75}
Unit ordering cost (c)	{1, 3}
Unit holding cost (h)	1
Unit outdating cost (w)	0
Service level (α)	{0.90, 0.95, 0.99}
Shelf life (M)	{2, 3}
Planning horizon (T)	12

Table 5.2: Demand sets, demand patterns, and arrival rates

Demand patterns	Arrival rates in period t											
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**
Stationary	15	15	15	15	15	15	15	15	15	15	15	15
Increasing	2	5	7	8	9	11	16	18	22	25	28	29
Decreasing	29	28	25	22	18	16	11	9	8	7	5	2
Seasonal	2	7	9	16	22	28	29	25	18	11	8	5
Cyclic	2	9	22	28	16	7	5	11	25	29	18	8
Random	8	18	5	9	22	2	28	16	11	29	25	7

12 periods. We consider the service levels $\alpha \in \{0.90, 0.95, 0.99\}$. There are 6 different Poisson demand sets with stationary, increasing, decreasing, seasonal, cyclic, and random demand patterns. In total, we have 432 instances.

Since there is no optimal solution in the literature to our knowledge, we cannot evaluate the optimality gap. We use the near-optimal SDP approach of [HHRPW12], which is the only approach in the literature to our knowledge, as the benchmark to our heuristic. Unlike the penalty cost case, we implement the same model as in [HHRPW12] which was originally developed for the service level case. We compare and evaluate the performance of our heuristic and the SDP approach in terms of cost calculations and computational efficiency. The dimensionality of the perishable inventory problem severely limits the performance of the SDP model. Therefore, we study with the demand sets with relatively small arrival rates.

5.5.2 Results and Discussion

Costs

Table 5.3 summarises the average percentage difference between the heuristic and the SDP solution for each parameter. The percentage difference is calcu-

lated by the following formula:

$$100 \times \frac{TC_H - TC_{SDP}}{TC_{SDP}}$$

where TC_H and TC_{SDP} denote the total cost for the heuristic and the SDP solution, respectively. On average, the heuristic has a performance advantage of 3.17% over the SDP solution. The average total cost for the instance set is found to be 898.1 and 870.1 for the SDP and the heuristic, respectively.

From Table 5.3, it is seen that the heuristic performs better than the SDP for all the demand patterns, and the performance tends to be the greater for the stationary demand pattern, 5.03% better on average. For the seasonal and the random demand patterns, the heuristic tend to have slightly lower performance, such that 2.35% and 1.70% lower cost values. One possible explanation might be the sharp fluctuations between the demand distributions in some of the subsequent periods benefit from more dynamic control, similar to the way that it might result in higher optimality gaps in the penalty cost case. We also observe that the performance tends to increase as the amount of shelf life decreases. Notice that the average replenishment cycle is 1.70 periods for $M = 2$ and the average cycle length becomes equal to 2.13 when $M = 3$. Hence, the percentage of 3-period replenishment cycles tends to be less than that of 2-period cycles in case $M = 2$. In addition, the total cost tends to drop as M increases, since longer shelf life allow less number of replenishment orders.

A similar relationship between the fixed setup cost and the replenishment cycle length can be observed, such that when the fixed setup cost a is sufficiently low, more frequent replenishment orders do not have a significant effect on the increase in the total cost. If a gets higher, ordering decisions become more costly and therefore we need to keep the replenishment cycles as long as possible to avoid additional ordering costs. We can observe this situation in Table 5.3, as a increases from 25 to 75, the average cycle length increases from 1.50 to 2.21. The performance of the heuristic tends to be higher than that of the SDP solution as a increases. Moreover, we notice that the performance gap tends to decrease as M increases, from -3.70% when $M = 2$ to -2.64% when $M = 3$, therefore the heuristic tends to perform better when the amount of shelf life is smaller. As c increases, the performance of the heuristic over the SDP tends to decrease as well.

Service levels

Tables 5.3 and 5.5 illustrate the average service levels over the whole planning horizon and the values for the period in which the minimum service levels were observed, based on 5,000 simulation replications and for $\alpha \in \{0.9, 0.95, 0.99\}$. We also include the ser service level estimate It is clearly seen that the SDP tends to overachieve service levels in comparison to the heuristic. One possible explanation is that the SDP generates ordering policies overachieving the service level requirements, due to the conditional behaviour of the service level constraints. Regardless of the probability of their occurrence, the SDP meets the service level requirement per period for all possible starting inventory compositions, see Pauls-Worm and Hendrix [PWH15]. This also explains as the service level requirement grows, the performance gap between the heuristic and the SDP gets smaller, since the SDP generates the solutions closer to the optimal levels as α approaches 1. Our numerical experiments support this observation, since the heuristic performs 4.85% better when $\alpha = 0.9$ and it drops to 1.11% when $\alpha = 0.99$.

Computation Times

We implemented the SDP in MATLAB for $M \in \{2, 3\}$. Therefore, the SDP computation times are lower than that of the penalty cost case. As seen from Table 5.3, our proposed heuristic has a significant advantage over the SDP model in terms of the computation time, specifically when $M = 3$.

On average, the service level variation of the heuristic determines the replenishment periods in 4 milliseconds and the order quantities in 3.82 seconds. On the contrary, even though the SDP model works very fast (1.27 s on average per instance) for $M = 2$ and reasonably fast (376.54 s on average per instance) for $M = 3$, it slows down exponentially when $M = 4$ and generates the order quantity values in more than 1 day (more than 31 hours on average) per instance. Since the heuristic simulation includes order quantity calculation and the SDP simulation uses the parameters found by the SDP model, the heuristic simulation runtimes tend to be longer. If the shelf life increased to $M = 5$, the computation times would continue to increase exponentially, and it would take more than 1 year to calculate the order quantities for only one instance. In contrast to the SDP model, our proposed heuristic can generate the replenishment period vectors for all the 216 instances in just 0.9 seconds. Taking the simulation runs that determine the order quantity into account, the

calculation of all the parameters for the whole instance set takes less than 15 minutes. Since we calculate the order quantities during the simulation runs, the average simulation runtime for our heuristic is longer (3.82 s) compared to that of the SDP model (2.88 s). When we conduct a sensitivity analysis on the algorithm and the simulation runtimes for both the SDP and our proposed heuristic, we clearly see from Table 5.3 that all of them are subject to increase as M increases and there are slight differences in the simulation runtimes between different demand patterns, while changes in all the other parameters do not cause a significant change in the runtimes.

Testing the Heuristic on Known Instances in the Literature

We herein test the service level version of the heuristic approach on a modified version of the another instance set in the literature, taken from [PWHHvdV14], and follows a normal distribution with the coefficients of variances given in Table 5.4. Their mean demand parameters are too large (e.g., 800, 950), it is practically intractable to implement the SDP model without using some techniques and approximations such as extrapolation. Therefore, we scale down the demand mean and the fixed setup cost parameters by 40 and implement the SDP model without any technique. We select a subset of their instance set and take the parameters given in Tables 5.4 and 4.5. There are 36 instances in total.

According to Table 5.5, the average gap between the SDP solution and the heuristic is 1.27%. As M increases, the gap does not tend to drop. In addition, as a increases the gap tends to decrease, the performance of the heuristic increases, and for $a = 50$ the heuristic shows as good performance as the SDP. Since higher setup costs lead to less number of replenishment order, the average cycle length incrases as a increases. For lower service level values such as $\alpha = 0.9$, the heuristic tends to perform better than the SDP solution. As α increases the SDP has a tendency to outperform the heuristic. Finally, the heuristic performs worse when the coefficient of variation is larger.

As seen from Table 5.5 that while the heuristic meets the service level requirements, the SDP tends to overachieve them. For higher α values, the overachievement is limited and the SDP solution approaches the optimal levels. It is one possible explanation to the quick increase in the gap between the SDP and the heuristic when we approach $\alpha = 1$.

Table 5.3: Comparison of the SDP and the heuristic, service level case, with respect to each parameter and the average computation times (in seconds) per instance (5,000 simulation reps each).

Parameter	Avg. Opt. Gap Percent Diff	95% CI Percent Diff	Avg. Heuristic Total Cost	Avg. SDP Total Cost	Avg. Heur. Cycle Length	Avg. SDP Runtime (s)	Avg. SDP Sim. Runtime (s)	Avg. Heuristic Runtime (s)	Avg. Heur. Sim. Runtime (s)	Avg. Heur. Service Level (5,000 reps)	Min. Heur. Service Level (5,000 reps)	Avg. SDP. Service Level (5,000 reps)	Min. SDP. Service Level (5,000 reps)
Total avg	-3.17%	[-3.55, -2.79]	870.06	898.08	1.91	188.91	2.88	0.004	3.82	0.9746	0.9539	0.9864	0.9820
Stationary	-5.03%	[-5.93, -4.12]	792.92	833.77	2.05	190.57	3.46	0.004	4.30	0.9763	0.9520	0.9922	0.9842
Increasing	-3.54%	[-4.24, -2.83]	898.76	932.06	1.84	187.79	2.82	0.005	3.86	0.9748	0.9531	0.9844	0.9666
Decreasing	-2.97%	[-3.87, -2.06]	873.42	900.16	1.92	188.56	2.77	0.005	3.66	0.9760	0.9572	0.9863	0.9774
Cyclic	-3.43%	[-4.29, -2.58]	882.51	913.91	1.89	188.33	2.75	0.004	3.76	0.9757	0.9597	0.9853	0.9740
Seasonal	-2.35%	[-3.28, -1.41]	888.71	910.21	1.87	188.46	2.75	0.004	3.53	0.9754	0.9542	0.9869	0.9753
Random	-1.70%	[-2.61, -0.79]	884.05	898.37	1.91	189.73	2.74	0.004	3.81	0.9723	0.9484	0.9849	0.9739
$M = 2$	-3.70%	[-4.23, -3.16]	925.10	891.55	1.70	1.27	3.36	0.003	3.88	0.9727	0.9532	0.9841	0.9717
$M = 3$	-2.64%	[-3.16, -2.13]	848.57	871.06	2.13	376.54	2.41	0.005	3.75	0.9775	0.9542	0.9892	0.9808
$\alpha = 0.9$	-4.85%	[-5.40, -4.29]	828.19	870.08	1.97	187.87	2.89	0.004	3.77	0.9531	0.9125	0.9754	0.9666
$\alpha = 0.95$	-3.55%	[-4.08, -3.02]	859.64	891.02	1.93	188.98	2.89	0.005	3.85	0.9766	0.9590	0.9872	0.9825
$\alpha = 0.99$	-1.11%	[-1.68, -0.54]	922.35	933.14	1.84	189.87	2.87	0.004	3.84	0.9956	0.9922	0.9974	0.9966
$a = 25$	-1.23%	[-1.81, -0.66]	708.51	714.95	1.50	188.84	2.89	0.004	4.20	0.9697	0.9547	0.9754	0.9681
$a = 50$	-3.68%	[-4.27, -3.09]	880.45	911.06	2.03	188.86	2.88	0.004	3.70	0.9766	0.9543	0.9902	0.9847
$a = 75$	-4.59%	[-5.14, -4.05]	1,021.23	1,068.23	2.21	189.02	2.88	0.004	3.56	0.9790	0.9548	0.9944	0.9892
$c = 1$	-4.18%	[-4.71, -3.64]	662.14	692.54	2.03	188.91	2.88	0.005	3.72	0.9764	0.9545	0.9896	0.9849
$c = 3$	-2.16%	[-2.63, -1.70]	1,077.98	1,103.62	1.80	188.91	2.89	0.004	3.92	0.9738	0.9546	0.9837	0.9789

Table 5.4: Input parameters for the experiments

Input parameters	Values
Fixed setup cost (a)	{12.5, 37.5, 50}
Unit ordering cost (c)	2
Unit holding cost (h)	0.5
Unit outdating cost (w)	0
Shelf life (M)	{2, 3}
Service level (α)	{0.90, 0.95, 0.99}
Planning horizon (T)	12
Coefficient of variation (CV)	{0.25, 0.333}

Testing the Runtimes on Instances with Larger Demand and Lifetime

We test the computation times of the heuristic when shelf lives are longer and demand means are larger. We use the same parameter set given in Table 5.4 except the fixed setup cost parameter $a = 1500$ and with the demand means given in Table 4.7 taken from [PWHHvdV14], which we scaled down by 40 in Section 4.6.2. The shelf lives take value from the set $M \in \{2, 4, 6, 8, 10\}$. We consider the time horizon $T \in \{12, 24, 48\}$. In total there are 90 instances.

As seen from Table 5.6, the heuristic runtimes tend to increase as M grows, from 0.0009 seconds for $M = 2$ to 0.0429 seconds for $M = 10$. As T increases, the runtimes tend to increase linearly for smaller M values, but they have a tendency to increase exponentially for larger M. The increase in M tends to result in exponential increases in runtimes, e.g., from 0.0015 seconds when $M = 2$ to 0.0895 seconds when $M = 10$ for $T = 48$. The heuristic outperforms the optimal SDP solution in terms of order of magnitude of the growth in computation times. The heuristic tends to have similar performance for different instances of the fixed setup cost increases. As the coefficient of variation as M increases, the computation times tend to decline slightly, and this becomes more noticeable as M grows. In addition, as α grows, the computation times tend decrease and it becomes more noticeable as M increases.

5.6 Conclusion

We have studied a perishable inventory system operating under minimal α-service level constraints. Similar to the penalty cost variation of the problem, the dimensionality of the state space makes the SDP solution and computationally intractable and several complexity issues related to formulating the prob-

Table 5.5: Comparison of the SDP and the heuristic, service level case, with respect to each parameter.

Parameter	Avg. Opt. Gap Percent Diff	95% CI Percent Diff	Avg. Heuristic Total Cost	Avg. SDP Total Cost	Avg. Cycle Length	Avg. Heur. Service Level (5,000 reps)	Min. Heur. Service Level (5,000 reps)	Avg. SDP. Service Level (5,000 reps)	Min. SDP. Service Level (5,000 reps)
Total avg	1.27%	[0.30, 2.23]	739.78	731.72	1.72	0.9720	0.9506	0.9790	0.9660
M = 2	1.95%	[0.34, 3.55]	772.76	759.81	1.54	0.9702	0.9498	0.9753	0.9557
M = 3	0.59%	[-0.43, 1.60]	706.81	703.64	1.89	0.9738	0.9513	0.9827	0.9654
a = 12.5	3.38%	[1.96, 4.80]	585.61	565.85	1.39	0.9665	0.9487	0.9688	0.9499
a = 37.5	0.55%	[-0.88, 1.99]	775.53	770.63	1.77	0.9739	0.9513	0.9808	0.9640
a = 50	-0.14%	[-1.68, 1.39]	858.22	858.68	2.00	0.9755	0.9513	0.9874	0.9724
$\alpha = 0.90$	-1.07%	[-2.26, 0.13]	700.08	710.04	1.83	0.9468	0.9038	0.9614	0.9388
$\alpha = 0.95$	0.84%	[-0.23, 1.90]	730.94	726.24	1.70	0.9740	0.9547	0.9797	0.9662
$\alpha = 0.99$	4.03%	[2.74, 5.32]	788.33	758.89	1.61	0.9952	0.9918	0.9958	0.9919
CV = 0.25	0.82%	[-0.47, 2.12]	719.49	714.99	1.77	0.9740	0.9511	0.9815	0.9674
CV = 0.333	1.71%	[0.28, 3.14]	760.07	748.45	1.66	0.9700	0.9501	0.9765	0.9616

Table 5.6: Comparison of the average heuristic runtimes (seconds) for different product lifetimes, service level case.

Parameter	**Lifetime** (M)				
	2	**4**	**6**	**8**	**10**
Total avg	0.0009	0.0041	0.012	0.0238	0.0429
T = 12	0.0004	0.0014	0.0030	0.0048	0.0071
T = 24	0.0008	0.0034	0.0092	0.0189	0.0319
T = 48	0.0015	0.0074	0.0215	0.0478	0.0895
α *= 0.90*	0.0010	0.0042	0.0121	0.0279	0.0537
α *= 0.95*	0.0054	0.0041	0.0116	0.0254	0.0444
α *= 0.99*	0.0008	0.0040	0.0099	0.0182	0.0304
CV = 0.25	0.0009	0.0042	0.0119	0.0260	0.0479
CV = 0.333	0.0009	0.0040	0.0105	0.0217	0.0378

lem as a network graph arise as the shelf life grows. Therefore, we adapted the time-efficient and fast heuristic we developed for the penalty cost case to the service level case. The heuristic is inspired by the well-known Wagner-Whitin algorithm and the Silver-Meal heuristic. It aims to obtain the best path on the network while using the deterministic-equivalent ending inventory composition state as a parameter to maintain the lowest average cost in the ordering decision. Our numerical experiments show that using the order timings and quantities generated by our proposed heuristic, the heuristic has a similar performance to the SDP solution and often outperforms it. This is due to the overachievement of the service level constraints in the SDP solution. Moreover, while it meets the service level constraints in every period, it also avoids overachieving them, in contrast to the SDP solution.

Our heuristic works fast and generates near-optimal results even for even for much longer shelf lives. Moreover, in terms of implementation and computational complexity, it is very convenient. It allows flexibility to make ordering decisions based on observed demand and inventory states. It also prevents replenishment timing decision set-ups and reviews that might be required for every period, as seen in the dynamic strategy and rolling-horizon frameworks. Therefore, it could be easily applied to most real-world inventory systems for perishable goods with service level requirements, and utilised by executives and decision-makers.

Chapter 6

Conclusions & Future Work

6.1 Conclusions

In this chapter, we conclude the thesis and summarise our findings for each chapter. We applied the (R, S) policy to ensure the flexibility and the responsiveness of ordering decisions for uncertainties that might arise due to the non-stationary stochastic nature of demand. Moreover, we presented a novel and time-efficient approach that is minimally affected by the amount of item lifetime, in order to handle the state space dimensionality, which is determined by the item lifetime. We also employed the (R, S) policy in the IRP model that we developed for CFSCs and takes CO_2 emissions and refrigeration costs into account. Demonstrating the optimality of the (s, S) policies for inventory systems under service level constraints will fill an important gap in the literature and will enable us to compare our proposed heuristic for perishable inventory models with the optimal values, after small modifications for perishability.

In Chapter 1, we briefly summarised our focus, goal, and motivation for the study in this dissertation, and listed our contributions to the inventory control literature.

In Chapter 2, we provided a detailed analysis for the inventory control, lot-sizing, perishability, and cold food supply chain concepts, as well as the relevant optimisation techniques and models such as dynamic programming, mixed integer programming, and the inventory routing problem.

Chapter 3 includes the discussion of our first study, the IRP model we developed for CFSCs, and under the (R, S) policy. We first described the problem, intro-

duced a deterministic-equivalent approach, then provided the formulas we used for transportation and refrigeration calculations, and presented the model. In the extensive numerical analysis we presented, we compared our (R, S) model with the (R, Q) model of [SBRHVDV15]. Finally, we discussed that the flexibility in ordering decisions allowed by the (R, S) policy significantly reduces the cumulative inventory holding costs arising due to the excess inventory caused by pre-determined order quantities of the (R, Q) policy.

In Chapter 4, we defined the problem and presented the heuristic that we developed for the penalty case of our perishability problem. We also provided a detailed explanation about the network graph representation of the state space and the procedure to calculate the optimal replenishment cycle order-up-to levels, which is based on the multi-period newsvendor problem. We discussed the steps of the heuristic and the procedure to determine the replenishment cycle with a static approach and to determine the order quantities with a dynamic approach. We then compared our study with the optimal SDP solution by including an extensive numerical study and showed that our heuristic has an optimality gap between 0%-5% for all the instances of the instance set. We also compared the timing and showed that our proposed heuristic is affected minimally by the increase in item lifetime, while its effect on the SDP solution is huge and exponentially growing.

Chapter 5 provides the service level version of the heuristic we analysed in Chapter 4. We first gave the modified problem description. In this case, we replaced the penalty cost with a service level constraint and compared the heuristics with the near-optimal SDP approach of [HHRPW12]. Since the SDP approach is not necessarily optimal, our heuristic slightly outperforms the SDP heuristic, and on average, they have very similar results.

To sum up, this dissertation work contributed to the inventory literature by providing novel approaches for perishable inventory systems and CFSCs that calculate near-optimal solutions at a reasonable time and operate under the static-dynamic (R, S) ordering policy.

6.2 Future Directions

The methods and our findings in the work we conducted in this dissertation open several possible future research paths and further research questions. We hereby briefly discuss these directions and give several insights.

6.2.1 Cold Food Supply Chains

Our IRP model for CFSCs can be applied in the problem settings given in the following in the future:

Multi-supplier: In this study, we consider a single supplier. A possible future direction could be considering multiple suppliers. It would allow the model to better reflect the real-world CFSCs. Several challenges such as supplier selection and coordination might arise in this extension.

Positive lead time: The IRP model can be extended and modified to handle non-zero deterministic or stochastic lead time cases. In real-world supply chain models, there is a delay between ordering and placing items, e.g., due to transportation times. Incorporating lead times would allow the model to better reflect the real-world systems.

Non-identical vehicles: We assume a fixed number of identical vehicles. However, as in most real-life cases, we can consider non-identical vehicles with different properties in terms of fuel consumption, refrigeration activities, capacity, etc. Incorporating fleets with non-identical vehicles could make routing and fuel consumption calculations more realistic and cost-effective.

Capacitated warehouses: The CFSC IRP model can be extended to settings with limited warehouse storage capacity. Capacity constraints might need to be applied into replenishment, routing, and allocation decisions. This extension would result in challenges such as coordinating vehicle routing with storage limits and managing trade-offs between transportation cost, emissions, and capacitated storage.

Additional sustainability measures: In this work, we only take CO_2 emissions into account. Additional environmental and social sustainability measures, food quality and safety, waste, shelf life, food expiration date, etc., could be added to the problem definition and studied. It would align the model with global sustainability goals and more responsible supply chains.

Perishable inventory: In this work, we only focused on non-perishable inventory systems. The study can be extended to perishable inventory models studied in [SBRHVDV15] and the performance of the (R, S) model in terms of cost, emission reduction, and excess cumulative inventory levels could be compared if the perishability case is considered. Incorporating perishability would make the model more relevant for CFSCs since food spoilage is a key concern in real

life in most cases. Perishability also allows incorporating shelf life-dependent replenishment and issuing policies, as well as waste management policies.

6.2.2 Perishable Inventory Management

Our proposed heuristic for perishable inventory systems can be applied in the problem settings given below:

Positive outdating cost: In this work, we assume there is zero outdating cost (e.g., no additional unit cost for discarding spoiled food in supermarkets). In many real-life cases, outdated items might have significant effects such as speeding up the outdating process of nearby fresh items or resulting in serious health issues. Those kind of effects could be modeled in terms of a positive outdating cost and incorporated into the heuristic. In the positive outdating cost case, estimating expected waste cost is a challenging problem. The study can be extended to handle this task. An exact expected waste formulation could be developed and the heuristic could be modified for the exact formulation. There might be several additional challenges such as minimising and handling the amount of waste, balancing waste and inventory cost minimisation in the positive outdating cost case.

Positive lead time: In this dissertation, we assume zero lead time. The models and heuristics can be modified to consider non-zero lead times. As mentioned previously, most real-life order placements do not happen within the same time unit and might require delays. For further generalisation of the problem, the lead time can be modelled as a random variable. This extension would add complexity, such as the development of forecasting methods, lead time demand calculations, or buffer stock policies in order to manage uncertainty.

Random shelf life: The heuristics can be modified and adapted to the cases in which the item/product lifetime is a random variable. The study can be extended to the existing problem settings in the literature with random lifetime. This extension would better represent biological or processed food products, since they often have variable shelf lives depending on the production or the environmental conditions.

Multi-product/item inventory: Extending the model to multi-product, multi-item, and/or multi-echelon inventory systems is another possible future direction. This would make the heuristic more suitable for complex and multi-layer retail or distribution settings. It would also take cross-product interactions and

constraints into account, such as shared storage space or joint replenishment.

Capacitated warehouses: The perishability models and heuristics could be extended to capacitated warehouses. This extension might require incorporating capacity constraints into ordering and allocation decisions. Possible challenges include balancing capacity usage with minimisation of waste, and handling priority rules when demand exceeds capacity, Moreover, designing efficient allocation methods for each product/item type is another possible challenge if it is applied to multi-product/item inventory systems.

Joint replenishment: The perishability study in this dissertation can be combined with the joint replenishment problem. This would allow for the coordination of orders, transportation, or handling for multiple items. It might also introduce several challenges in balancing perishability with joint replenishment policies.

Pricing: The study can be extended to the problem settings in the literature in which pricing is considered. Incorporating pricing decisions would better link inventory management with revenue optimisation, It would also apply dynamic pricing strategies to meet customer demand and minimise waste.

References

[AAL14] Pedro Amorim and Bernardo Almada-Lobo. The impact of food perishability issues in the vehicle routing problem. *Computers & Industrial Engineering*, 67:223–233, 2014.

[ABTM08] Amir Azaron, KN Brown, S Armagan Tarim, and Mohammed Modarres. A multi-objective stochastic programming approach for supply chain design considering risk. *International Journal of Production Economics*, 116(1):129–138, 2008.

[Ade04] Daniel Adelman. A price-directed approach to stochastic inventory/routing. *Operations Research*, 52(4):499–514, 2004.

[AGCB18] Hamid Allaoui, Yuhan Guo, Alok Choudhary, and Jacqueline Bloemhof. Sustainable agro-food supply chain design using two-stage hybrid multi-objective decision-making approach. *Computers & Operations Research*, 89:369–384, 2018.

[AHC+10] Henrik Andersson, Arild Hoff, Marielle Christiansen, Geir Hasle, and Arne Løkketangen. Industrial aspects and literature survey: Combined inventory management and routing. *Computers & operations research*, 37(9):1515–1536, 2010.

[AHM51] Kenneth J Arrow, Theodore Harris, and Jacob Marschak. Optimal inventory policy. *Econometrica: Journal of the Econometric Society*, pages 250–272, 1951.

[AK58] Kenneth Arrow and Samuel Karlin. Herbert scarf. *Studies in the Mathe-matical Theory of Inventory and Production*, 1958.

[Ame04] GM America. of.(2004). 2004 unsalables benchmark report (inf. téc.). *Grocery Manufacturers of America*, 2004.

[AN87] Yash Aneja and A Hamid Noori. The optimality of (s, s) policies for a stochastic inventory problem with proportional and lump-sum penalty cost. *Management Science*, 33(6):750–755, 1987.

[Ask81] Ronald G Askin. A procedure for production lot sizing with probabilistic dynamic demand. *Aiie Transactions*, 13(2):132–137, 1981.

[ASM21] Adel Aazami and Mohammad Saidi-Mehrabad. A production and distribution planning of perishable products with a fixed lifetime under vertical competition in the seller-buyer systems: A real-world application. *Journal of manufacturing systems*, 58:223–247, 2021.

[ATSAHA20] Nader Al Theeb, Hazem J Smadi, Tarek H Al-Hawari, and Manar H Aljarrah. Optimization of vehicle routing with inventory allocation problems in cold supply chain logistics. *Computers & Industrial Engineering*, 142:106341, 2020.

[AZ13] Saman Hassanzadeh Amin and Guoqing Zhang. A multi-objective facility location model for closed-loop supply chain network under uncertain demand and return. *Applied Mathematical Modelling*, 37(6):4165–4176, 2013.

[BBL15] Luca Bertazzi, Adamo Bosco, and Demetrio Laganà. Managing stochastic demand in an inventory routing problem with transportation procurement. *Omega*, 56:112–121, 2015.

[BCNT19] Eleonora Bottani, Giorgia Casella, Majcol Nobili, and Letizia Tebaldi. Assessment of the economic and environmental sustainability of a food cold supply chain. *IFAC-PapersOnLine*, 52(13):367–372, 2019.

[Bee67] Paul R Beesack. A finite horizon dynamic inventory model with a stockout constraint. *Management Science*, 13(9):618–630, 1967.

[Bel54] Richard Bellman. The theory of dynamic programming. *Bulletin of the American Mathematical Society*, 60(6):503–515, 1954.

[Bel57] Richard Bellman. A Markovian decision process. *Journal of mathematics and mechanics*, pages 679–684, 1957.

[Bel66] Richard Bellman. Dynamic programming. *science*, 153(3731):34–37, 1966.

[BFF+16] Marco Bortolini, Maurizio Faccio, Emilio Ferrari, Mauro Gamberi, and Francesco Pilati. Fresh food sustainable distribution: cost, delivery time and carbon footprint three-objective optimization. *Journal of Food Engineering*, 174:56–67, 2016.

[Bij14] Marco Bijvank. Periodic review inventory systems with a service level criterion. *Journal of the Operational Research Society*, 65(12):1853–1863, 2014.

[BL11] Tolga Bektaş and Gilbert Laporte. The pollution-routing problem. *Transportation Research Part B: Methodological*, 45(8):1232–1250, 2011.

[BM99] Srinivas Bollapragada and Thomas E Morton. A simple heuristic for computing nonstationary (s, S) policies. *Operations research*, 47(4):576–584, 1999.

[BN09] Jonathan F Bard and Narameth Nananukul. Heuristics for a multiperiod inventory routing problem with production decisions. *Computers & Industrial Engineering*, 57(3):713–723, 2009.

[Boz16] Ali Bozorgi. Multi-product inventory model for cold items with cost and emission consideration. *International Journal of Production Economics*, 176:123–142, 2016.

[BPN14] Ali Bozorgi, Jennifer Pazour, and Dima Nazzal. A new inventory model for cold items that considers costs and emissions. *International Journal of Production Economics*, 155:114–125, 2014.

[BRT12] Monique Bakker, Jan Riezebos, and Ruud H Teunter. Review of inventory systems with deterioration since 2001. *European Journal of Operational Research*, 221(2):275–284, 2012.

[BSA+20] Mahla Babagolzadeh, Anup Shrestha, Babak Abbasi, Yahua Zhang, Alice Woodhead, and Anming Zhang. Sustainable cold supply chain management under demand uncertainty and carbon tax regulation. *Transportation Research Part D: Transport and Environment*, 80:102245, 2020.

[BT88] James H Bookbinder and Jin-Yan Tan. Strategies for the probabilistic lot-sizing problem with service-level constraints. *Management Science*, 34(9):1096–1108, 1988.

[Bul64] Ekaterina Vadimovna Bulinskaya. Some results concerning optimum inventory policies. *Theory of Probability & Its Applications*, 9(3):389–403, 1964.

[BV12a] Marco Bijvank and Iris FA Vis. Inventory control for point-of-use locations in hospitals. *Journal of the Operational Research Society*, 63(4):497–510, 2012.

[BV12b] Marco Bijvank and Iris FA Vis. Lost-sales inventory systems with a service level criterion. *European Journal of Operational Research*, 220(3):610–618, 2012.

[BVD09] Rob ACM Broekmeulen and Karel H Van Donselaar. A heuristic to manage perishable inventory with batch ordering, positive lead-times, and time-varying demand. *Computers & Operations Research*, 36(11):3013–3018, 2009.

[BW02] Dirk Beyer and Julie Ward. *Network server supply chain at HP: A case study*. Springer, 2002.

[BYS05] Matthew Barth, Theodore Younglove, and George Scora. Development of a heavy-duty diesel modal emissions and fuel consumption model. *California PATH Research Report*, 2005.

[CCKS98] Ann Campbell, Lloyd Clarke, Anton Kleywegt, and Martin Savelsbergh. The inventory routing problem. In *Fleet management and logistics*, pages 95–113. Springer, 1998.

[CH15] Wei-Ting Chen and Chaug-Ing Hsu. Greenhouse gas emission estimation for temperature-controlled food distribution systems. *Journal of Cleaner Production*, 104:139–147, 2015.

[Che06] Zhi-Long Chen. Integrated production and outbound distribution scheduling in a supply chain: Review and extensions. Technical report, Working paper, Robert H. Smith School of Business, University of Maryland . . . , 2006.

[CK01] Frank Y Chen and Dmitry Krass. Inventory models with minimal service level constraints. *European journal of operational research*, 134(1):120–140, 2001.

[CKR18] Vaibhav Chaudhary, Rakhee Kulshrestha, and Srikanta Routroy. State-of-the-art literature review on inventory models for perishable products. *Journal of Advances in Management Research*, 2018.

[CL13] Leandro C Coelho and Gilbert Laporte. The exact solution of several classes of inventory-routing problems. *Computers & Operations Research*, 40(2):558–565, 2013.

[CM10] Sunil Chopra and Peter Meindl. Supply chain management: Strategy, planning, and operation, 2010.

[CS60] Andrew J Clark and Herbert Scarf. Optimal policies for a multi-echelon inventory problem. *Management science*, 6(4):475–490, 1960.

[CS62] Andrew J Clark and Herbert Scarf. Approximate solutions to a simple multi-echelon inventory problem. *Studies in applied probability and management science*, pages 88–110, 1962.

[CWF22] G Chen, MIM Wahab, and L Fang. Optimal replenishment strategy for a single-manufacturer multi-retailer cold chain considering multi-stage quality degradation. *Applied Mathematical Modelling*, 104:96–113, 2022.

[DBGVW84] Marc A De Bodt, Ludo F Gelders, and Luk N Van Wassenhove. Lot sizing under dynamic demand conditions: A review. *Engineering Costs and Production Economics*, 8(3):165–187, 1984.

[DBL12] Emrah Demir, Tolga Bektaş, and Gilbert Laporte. An adaptive large neighborhood search heuristic for the pollution-routing problem. *European journal of operational research*, 223(2):346–359, 2012.

[DBL14] Emrah Demir, Tolga Bektaş, and Gilbert Laporte. The bi-objective pollution-routing problem. *European journal of operational research*, 232(3):464–478, 2014.

[DHS+19] Emrah Demir, Katy Huckle, Aris Syntetos, Andrew Lahy, and Mike Wilson. Vehicle routing problem: Past and future. *Contemporary operations and logistics: Achieving excellence in turbulent times*, pages 97–117, 2019.

[DJN14] Kannan Devika, Ahmad Jafarian, and Vahid Nourbakhsh. Designing a sustainable closed-loop supply chain network based on triple bottom line approach: A comparison of metaheuristics hybridization techniques. *European journal of operational research*, 235(3):594–615, 2014.

[DK97] Andreas Drexl and Alf Kimms. Lot sizing and scheduling—survey and extensions. *European Journal of operational research*, 99(2):221–235, 1997.

[DMA+15] Nicolas Danloup, Vahid Mirzabeiki, Hamid Allaoui, Gilles Goncalves, Denyse Julien, and Carlos Mena. Reducing transportation greenhouse gas emissions with collaborative distribution: a case study. *Management Research Review*, 38(10):1049–1067, 2015.

[DMGB22] Bram J De Moor, Joren Gijsbrechts, and Robert N Boute. Reward shaping to improve the performance of deep reinforcement learning in perishable inventory management. *European Journal of Operational Research*, 301(2):535–545, 2022.

[DR59] George B Dantzig and John H Ramser. The truck dispatching problem. *Management science*, 6(1):80–91, 1959.

[DSKTR16] Gozdem Dural-Selcuk, Onur A Kilic, S Armagan Tarim, and Roberto Rossi. A comparison of non-stationary stochastic lot-sizing strategies. *arXiv preprint arXiv:1607.08896*, 2016.

[Edg88] Francis Y Edgeworth. The mathematical theory of banking. *Journal of the Royal Statistical Society*, 51(1):113–127, 1888.

[EM12] Samir Elhedhli and Ryan Merrick. Green supply chain network design to reduce carbon emissions. *Transportation Research Part D: Transport and Environment*, 17(5):370–379, 2012.

[FHVW+13] Anna Franceschetti, Dorothée Honhon, Tom Van Woensel, Tolga Bektaş, and Gilbert Laporte. The time-dependent

pollution-routing problem. *Transportation Research Part B: Methodological*, 56:265–293, 2013.

[FRGB15] Johannes Fichtinger, Jörg M Ries, Eric H Grosse, and Peter Baker. Assessing the environmental impact of integrated inventory and warehouse management. *International Journal of Production Economics*, 170:717–729, 2015.

[Fri75] Brant E Fries. Optimal ordering policy for a perishable commodity with fixed lifetime. *Operations research*, 23(1):46–61, 1975.

[FZ84] Awi Federgruen and Paul Zipkin. The combined vehicle routing and inventory allocation problem. *Operations Research*, 32(5):1019–1037, 1984.

[GAHG+15] Alejandro Gutierrez-Alcoba, Eligius MT Hendrix, Inmaculada García, Gloria Ortega, Karin GJ Pauls-Worm, and Rene Haijema. On computing order quantities for perishable inventory control with non-stationary demand. In *International Conference on Computational Science and Its Applications*, pages 429–444. Springer, 2015.

[GARMBH17] Alejandro Gutierrez-Alcoba, Roberto Rossi, Belen Martin-Barragan, and Eligius MT Hendrix. A simple heuristic for perishable item inventory control under non-stationary stochastic demand. *International Journal of Production Research*, 55(7):1885–1897, 2017.

[GCB02] Thomas W Gruen, Daniel S Corsten, and Sundar Bharadwaj. Retail out-of-stocks: A worldwide examination of extent. *Causes, and Consumer Responses, Grocery Manufacturers of America, Washington DC*, 2002.

[GEK16] NM Galal and KS El-Kilany. Sustainable agri-food supply chain with uncertain demand and lead time. *International Journal of Simulation Modelling*, 15(3):485–496, 2016.

[GG01] Suresh Kumar Goyal and Bibhas Chandra Giri. Recent trends in modeling of deteriorating inventory. *European Journal of operational research*, 134(1):1–16, 2001.

[GKRL+16] David A Goldberg, Dmitriy A Katz-Rogozhnikov, Yingdong Lu, Mayank Sharma, and Mark S Squillante. Asymptotic optimality of constant-order policies for lost sales inventory models with large lead times. *Mathematics of Operations Research*, 41(3):898–913, 2016.

[GNPN20] Javid Ghahremani Nahr, Seyed Hamid Reza Pasandideh, and Seyed Taghi Akhavan Niaki. A robust optimization approach for multi-objective, multi-product, multi-period, closed-loop green supply chain network designs under uncertainty and discount. *Journal of industrial and production engineering*, 37(1):1–22, 2020.

[GOT25] Suheyl Gulecyuz, Barry O'Sullivan, and S Armagan Tarim. A heuristic method for perishable inventory management under non-stationary demand. *Omega*, 133:103267, 2025.

[GS17] Kannan Govindan and Hamed Soleimani. A review of reverse logistics and closed-loop supply chains: a journal of cleaner production focus. *Journal of cleaner production*, 142:371–384, 2017.

[GW08] Stephen C Graves and Sean P Willems. Strategic inventory placement in supply chains: Nonstationary demand. *Manufacturing & service operations management*, 10(2):278–287, 2008.

[HAaS17] Moncer Hariga, Rami As' ad, and Abdulrahim Shamayleh. Integrated economic and environmental models for a multi stage cold supply chain under carbon tax regulation. *Journal of Cleaner Production*, 166:1357–1371, 2017.

[Hai13] René Haijema. A new class of stock-level dependent ordering policies for perishables with a short maximum shelf life. *International Journal of Production Economics*, 143(2):434–439, 2013.

[Har13] Ford W Harris. How many parts to make at once. 1913.

[HCC17] Yu-Hsiang Hsiao, Mu-Chen Chen, and Cheng-Lin Chin. Distribution planning for perishable foods in cold chains with quality concerns: Formulation and solution procedure. *Trends in food science & technology*, 61:80–93, 2017.

[HDR17] Abdelhalim Hiassat, Ali Diabat, and Iyad Rahwan. A genetic algorithm approach for location-inventory-routing problem with perishable products. *Journal of manufacturing systems*, 42:93–103, 2017.

[HHRPW12] Eligius MT Hendrix, Rene Haijema, Roberto Rossi, and Karin GJ Pauls-Worm. On solving a stochastic programming model for perishable inventory control. In *International Conference on Computational Science and Its Applications*, pages 45–56. Springer, 2012.

[HJMR09] Woonghee Tim Huh, Ganesh Janakiraman, John A Muckstadt, and Paat Rusmevichientong. Asymptotic optimality of order-up-to policies in lost sales inventory systems. *Management Science*, 55(3):404–420, 2009.

[HPWR+15] Eligius MT Hendrix, Karin GJ Pauls-Worm, Roberto Rossi, Alejandro G Alcoba, and Rene Haijema. A sample-based method for perishable good inventory control with a service level constraint. In *International Conference on Computational Logistics*, pages 526–540. Springer, 2015.

[HvDvdWS09] René Haijema, Nico van Dijk, Jan van der Wal, and Cees Smit Sibinga. Blood platelet production with breaks: optimization by sdp and simulation. *International Journal of Production Economics*, 121(2):464–473, 2009.

[HvdWvD07] René Haijema, Jan van der Wal, and Nico M van Dijk. Blood platelet production: Optimization by dynamic programming and simulation. *Computers & Operations Research*, 34(3):760–779, 2007.

[HW63] George Hadley and Thomson M Whitin. Analysis of inventory systems. 1963.

[Igl63] Donald L Iglehart. Optimality of (s, s) policies in the infinite horizon dynamic inventory problem. *Management science*, 9(2):259–267, 1963.

[Iid02] Tetsuo Iida. A non-stationary periodic review production–inventory model with uncertain production capacity and un-

certain demand. *European Journal of Operational Research*, 140(3):670–683, 2002.

[JCS16] Larissa Janssen, Thorsten Claus, and Jürgen Sauer. Literature review of deteriorating inventory models by key topics from 2012 to 2015. *International Journal of Production Economics*, 182:86–112, 2016.

[JGO+24] Soodeh Jahdi, Suheyl Gulecyuz, Seamus O'Reilly, Barry O'Sullivan, and S Armagan Tarim. An IRP model to improve the sustainability of cold food supply chains under stochastic demand. *Journal of Cleaner Production*, 462:142615, 2024.

[JJ10] SJ James and CJFRI James. The food cold-chain and climate change. *Food Research International*, 43(7):1944–1956, 2010.

[JRK13] Zied Jemai, Yacine Rekik, and Rim Kalaï. Inventory routing problems in a context of vendor-managed inventory system with consignment stock and transshipment. *Production Planning & Control*, 24(8-9):671–683, 2013.

[JVDK12] Ola Jabali, Tom VanWoensel, and AG De Kok. Analysis of travel times and co2 emissions in time-dependent vehicle routing. *Production and Operations Management*, 21(6):1060–1074, 2012.

[JZML20] Jiehui Jiang, Dezhi Zhang, Qiang Meng, and Yajie Liu. Regional multimodal logistics network design considering demand uncertainty and CO_2 emission reduction target: A system-optimization approach. *Journal of Cleaner Production*, 248:119304, 2020.

[Kap70] Robert S Kaplan. A dynamic inventory model with stochastic lead times. *Management Science*, 16(7):491–507, 1970.

[KNS02] Anton J Kleywegt, Vijay S Nori, and Martin WP Savelsbergh. The stochastic inventory routing problem with direct deliveries. *Transportation science*, 36(1):94–118, 2002.

[KSÇS23] İlker Köseli, Mehmet Soysal, Mustafa Çimen, and Çağrı Sel. Optimizing food logistics through a stochastic inventory routing problem under energy, waste and workforce concerns. *Journal of Cleaner Production*, page 136094, 2023.

[KSVW94] Roelof Kuik, Marc Salomon, and Luk N Van Wassenhove. Batching decisions: structure and models. *European journal of operational research*, 75(2):243–263, 1994.

[KSWD11] Itir Z Karaesmen, Alan Scheller-Wolf, and Borga Deniz. Managing perishable and aging inventories: Review and future research directions. *Planning production and inventories in the extended enterprise*, pages 393–436, 2011.

[LSMMT19] Giada La Scalia, Rosa Micale, Pier Paolo Miglietta, and Pierluigi Toma. Reducing waste and ecological impacts through a sustainable and efficient management of perishable food based on the Monte Carlo simulation. *Ecological Indicators*, 97:363–371, 2019.

[LSZ19] Yongbo Li, Hamed Soleimani, and Mostafa Zohal. An improved ant colony optimization algorithm for the multi-depot green vehicle routing problem with multiple objectives. *Journal of cleaner production*, 227:1161–1172, 2019.

[MAEHMA11] SMJ Mirzapour Al-E-Hashem, Hooman Malekly, and Mir Bahador Aryanezhad. A multi-objective robust optimization model for multi-product multi-site aggregate production planning in a supply chain under uncertainty. *International journal of production economics*, 134(1):28–42, 2011.

[MAH18] Hadi Moheb-Alizadeh and Robert Handfield. An integrated chance-constrained stochastic model for efficient and sustainable supplier selection and order allocation. *International Journal of Production Research*, 56(21):6890–6916, 2018.

[MBA17] MirMohammad Musavi and Ali Bozorgi-Amiri. A multi-objective sustainable hub location-scheduling problem for perishable food supply chain. *Computers & Industrial Engineering*, 113:766–778, 2017.

[MCS+20] Adriana Mendes, João Cruz, Tiago Saraiva, Tània M Lima, and Pedro D Gaspar. Logistics strategy (FIFO, FEFO or LSFO) decision support system for perishable food products. In *2020 International Conference on Decision Aid Sciences and Application (DASA)*, pages 173–178. IEEE, 2020.

[MDV12] Ioannis Mallidis, Rommert Dekker, and Dimitrios Vlachos. The impact of greening on supply chain design and cost: A case for a developing region. *Journal of Transport Geography*, 22:118–128, 2012.

[MM15] Antonella Meneghetti and Luca Monti. Greening the food supply chain: an optimisation model for sustainable design of refrigerated automated warehouses. *International Journal of Production Research*, 53(21):6567–6587, 2015.

[MRA19] Xiyuan Ma, Roberto Rossi, and Thomas Archibald. Stochastic inventory control: A literature review. *Ifac-papersonline*, 52(13):1490–1495, 2019.

[MS07] Noor Hasnah Moin and Said Salhi. Inventory routing problems: a logistical overview. *Journal of the Operational Research Society*, 58(9):1185–1194, 2007.

[MT10] Stefan Minner and Sandra Transchel. Periodic review inventory-control for perishable products under service-level constraints. *OR spectrum*, 32(4):979–996, 2010.

[MZA20] Fatemeh Mohebalizadehgashti, Hossein Zolfagharinia, and Saman Hassanzadeh Amin. Designing a green meat supply chain network: A multi-objective approach. *International Journal of Production Economics*, 219:312–327, 2020.

[Nah75] Steven Nahmias. Optimal ordering policies for perishable inventory—ii. *Operations research*, 23(4):735–749, 1975.

[Nah78] Steven Nahmias. The fixed-charge perishable inventory problem. *Operations Research*, 26(3):464–481, 1978.

[Nah82] Steven Nahmias. Perishable inventory theory: A review. *Operations research*, 30(4):680–708, 1982.

[Nah11] Steven Nahmias. *Perishable inventory systems*, volume 160. Springer Science & Business Media, 2011.

[NLN23] Yusuke Nomura, Ziang Liu, and Tatsushi Nishi. Deep reinforcement learning for perishable inventory optimization problem. In *2023 IEEE International Conference on Industrial Engineering*

and Engineering Management (IEEM), pages 0370–0374. IEEE, 2023.

[NP73] Steven Nahmias and William P Pierskalla. Optimal ordering policies for a product that perishes in two periods subject to stochastic demand. *Naval Research Logistics Quarterly*, 20(2):207–229, 1973.

[NP76] Steven Nahmias and William P Pierskalla. A two-product perishable/nonperishable inventory problem. *SIAM Journal on Applied Mathematics*, 30(3):483–500, 1976.

[OAM+25] Helen Onyeaka, Adenike Akinsemolu, Taghi Miri, Nnabueze Darlington Nnaji, Keru Duan, Gu Pang, Phemelo Tamasiga, Samran Khalid, Zainab T Al-Sharify, and Chinenye Ugwa. Artificial intelligence in food system: innovative approach to minimizing food spoilage and food waste. *Journal of Agriculture and Food Research*, 21:101895, 2025.

[OLFD16] Uche Okongwu, Matthieu Lauras, Julien François, and Jean-Christophe Deschamps. Impact of the integration of tactical supply chain planning determinants on performance. *Journal of Manufacturing Systems*, 38:181–194, 2016.

[ÖRSVdH15] Mehmet Önal, H Edwin Romeijn, Amar Sapra, and Wilco Van den Heuvel. The economic lot-sizing problem with perishable items and consumption order preference. *European Journal of Operational Research*, 244(3):881–891, 2015.

[PBF13] Shenle Pan, Eric Ballot, and Frédéric Fontane. The reduction of greenhouse gas emissions from freight transport by pooling supply chains. *International journal of production economics*, 143(1):86–94, 2013.

[PCS16] Arun Kr Purohit, Devendra Choudhary, and Ravi Shankar. Inventory lot-sizing with supplier selection under non-stationary stochastic demand. *International Journal of Production Research*, 54(8):2459–2469, 2016.

[Pow07] Warren B Powell. *Approximate Dynamic Programming: Solving the curses of dimensionality*, volume 703. John Wiley & Sons, 2007.

[Pre24] Steven Prestwich. Solving complex optimisation problems by machine learning. *AppliedMath*, 4(3):908–926, 2024.

[Put14] Martin L Puterman. *Markov decision processes: Discrete stochastic dynamic programming*. John Wiley & Sons, 2014.

[PV14] Julia Pahl and Stefan Voß. Integrating deterioration and lifetime constraints in production and supply chain planning: A survey. *European Journal of Operational Research*, 238(3):654–674, 2014.

[PVW92] Chris N Potts and Luk N Van Wassenhove. Integrating scheduling with batching and lot-sizing: A review of algorithms and complexity. *Journal of the Operational Research Society*, 43(5):395–406, 1992.

[PWH15] Karin GJ Pauls-Worm and Eligius MT Hendrix. SDP in inventory control: non-stationary demand and service level constraints. In *Computational Science and Its Applications–ICCSA 2015: 15th International Conference, Banff, AB, Canada, June 22-25, 2015, Proceedings, Part II 15*, pages 397–412. Springer, 2015.

[PWHAH16] Karin GJ Pauls-Worm, Eligius MT Hendrix, Alejandro G Alcoba, and Rene Haijema. Order quantities for perishable inventory control with non-stationary demand and a fill rate constraint. *International Journal of Production Economics*, 181:238–246, 2016.

[PWHHvdV14] Karin GJ Pauls-Worm, Eligius MT Hendrix, René Haijema, and Jack GAJ van der Vorst. An MILP approximation for ordering perishable products with non-stationary demand and service level constraints. *International Journal of Production Economics*, 157:133–146, 2014.

[Raa91] Fred Raafat. Survey of literature on continuously deteriorating inventory models. *Journal of the Operational Research society*, 42(1):27–37, 1991.

[RBC16] Raúl F Roldán, Rosa Basagoiti, and Leandro C Coelho. Robustness of inventory replenishment and customer selection poli-

cies for the dynamic and stochastic inventory-routing problem. *Computers & Operations Research*, 74:14–20, 2016.

[RBR16] Mohammad Rahimi, Armand Baboli, and Yacine Rekik. Inventory routing problem for perishable products by considering customer satisfaction and green criteria. In *Dynamics in Logistics: Proceedings of the 5th International Conference LDIC, 2016 Bremen, Germany*, pages 445–455. Springer, 2016.

[Ros13] Roberto Rossi. Periodic review for a perishable item under non stationary stochastic demand. *IFAC Proceedings Volumes*, 46(9):2021–2026, 2013.

[Ros21] Roberto Rossi. *Inventory analytics*. Open Book Publishers, 2021.

[RTB12] Roberto Rossi, S Armagan Tarim, and Ramesh Bollapragada. Constraint-based local search for inventory control under stochastic demand and lead time. *INFORMS journal on computing*, 24(1):66–80, 2012.

[RTHP08] Roberto Rossi, S Armagan Tarim, Brahim Hnich, and Steven Prestwich. A global chance-constraint for stochastic inventory systems under service level constraints. *Constraints*, 13(4):490–517, 2008.

[RTHP10] Roberto Rossi, S Armagan Tarim, Brahim Hnich, and Steven Prestwich. Computing the non-stationary replenishment cycle inventory policy under stochastic supplier lead-times. *International Journal of Production Economics*, 127(1):180–189, 2010.

[RTHP11] Roberto Rossi, S Armagan Tarim, Brahim Hnich, and Steven Prestwich. A state space augmentation algorithm for the replenishment cycle inventory policy. *International Journal of Production Economics*, 133(1):377–384, 2011.

[RTPH14] Roberto Rossi, S Armagan Tarim, Steven Prestwich, and Brahim Hnich. Piecewise linear lower and upper bounds for the standard normal first order loss function. *Applied Mathematics and Computation*, 231:489–502, 2014.

[SAKS60] Herbert Scarf, K Arrow, S Karlin, and P Suppes. The optimality of (S, s) policies in the dynamic inventory problem. *Optimal*

pricing, inflation, and the cost of price adjustment, pages 49–56, 1960.

[SBRHVDV15] Mehmet Soysal, Jacqueline M Bloemhof-Ruwaard, Rene Haijema, and Jack GAJ Van Der Vorst. Modeling an inventory routing problem for perishable products with environmental considerations and demand uncertainty. *International Journal of Production Economics*, 164:118–133, 2015.

[SBRM+12] Mehmet Soysal, Jacqueline Bloemhof-Ruwaard, Miranda PM Meuwissen, Jack GAJ van der Vorst, et al. A review on quantitative models for sustainable food logistics management. *International Journal on Food System Dynamics*, 3(2):136–155, 2012.

[SBRVDV14] Mehmet Soysal, Jacqueline M Bloemhof-Ruwaard, and Jack GAJ Van Der Vorst. Modelling food logistics networks with emission considerations: The case of an international beef supply chain. *International Journal of Production Economics*, 152:57–70, 2014.

[SCL12] Oğuz Solyalı, Jean-François Cordeau, and Gilbert Laporte. Robust inventory routing under demand uncertainty. *Transportation Science*, 46(3):327–340, 2012.

[SCS+18] Shashi Shashi, Roberto Cerchione, Rajwinder Singh, Piera Centobelli, and Amir Shabani. Food cold chain management: From a structured literature review to a conceptual framework and research agenda. *The International Journal of Logistics Management*, 29(3):792–821, 2018.

[SE16] Ahmed Saif and Samir Elhedhli. Cold supply chain design with environmental considerations: A simulation-optimization approach. *European Journal of Operational Research*, 251(1):274–287, 2016.

[SG17] UN Secretary-General. Agriculture development, food security and nutrition: report of the secretary-general. 2017.

[SHAaD19] Abdulrahim Shamayleh, Moncer Hariga, Rami As' ad, and Ali Diabat. Economic and environmental models for cold prod-

ucts with time varying demand. *Journal of cleaner production*, 212:847–863, 2019.

[Sil73] Edward A Silver. A heuristic for selecting lot size quantities for the case of a deterministic time-varying demand rate and discrete opportunities for replenishment. *Prod. Inventory Manage.*, 2:64–74, 1973.

[Sil78] Edward Silver. Inventory control under a probabilistic time-varying, demand pattern. *AIIE Transactions*, 10(4):371–379, 1978.

[Sil81] Edward A Silver. Operations research in inventory management: A review and critique. *Operations Research*, 29(4):628–645, 1981.

[SJK22] Mayur Selukar, Pooja Jain, and Tapan Kumar. Inventory control of multiple perishable goods using deep reinforcement learning for sustainable environment. *Sustainable Energy Technologies and Assessments*, 52:102038, 2022.

[SLC+18] Helena M Stellingwerf, Gilbert Laporte, Frans CAM Cruijssen, Argyris Kanellopoulos, and Jacqueline M Bloemhof. Quantifying the environmental and economic benefits of cooperation: A case study in temperature-controlled food logistics. *Transportation Research Part D: Transport and Environment*, 65:178–193, 2018.

[SLL+22] Yuhe Shi, Yun Lin, Ming K Lim, Ming-Lang Tseng, Changlu Tan, and Yan Li. An intelligent green scheduling system for sustainable cold chain logistics. *Expert Systems with Applications*, 209:118378, 2022.

[SMAehGB16] Z Sazvar, SMJ Mirzapour Al-e hashem, K Govindan, and B Bahli. A novel mathematical model for a multi-period, multi-product optimal ordering problem considering expiry dates in a FEFO system. *Transportation Research Part E: Logistics and Transportation Review*, 93:232–261, 2016.

[SP09] Dinesh Seth and Manjit Kumar Pandey. A multiple-item inventory model for a non-stationary demand. *Production Planning and Control*, 20(3):242–253, 2009.

[SPA16] Syamsuriadi Sinaga, Liza Setyaning Pertiwi, and Toni Ardian. Inventory simulation optimization under non stationary demand. *International Journal of Applied Engineering Research*, 11(1):524–529, 2016.

[SPT16] Edward A Silver, David F Pyke, and Douglas J Thomas. *Inventory and production management in supply chains*. CRC press, 2016.

[SR91] Helmut Schneider and Dan B Rinks. Empirical study of a new procedure for allocating safety stock in a wholesale inventory system. *International Journal of Production Economics*, 24(1-2):181–189, 1991.

[SRDM23] Gaurvendra Singh, R Rajesh, Yash Daultani, and Subhas Chandra Misra. Resilience and sustainability enhancements in food supply chains using digital twin technology: A grey causal modelling (GCM) approach. *Computers & Industrial Engineering*, 179:109172, 2023.

[SRK95] Helmut Schneider, DAN B RNKS, and Peter Kelle. Power approximations for a two-echelon inventory system using service levels. *Production and Operations Management*, 4(4):381–400, 1995.

[SS12] Oğuz Solyalı and Haldun Süral. The one-warehouse multi-retailer problem: Reformulation, classification, and computational results. *Annals of Operations Research*, 196:517–541, 2012.

[Tar96] S Armagan Tarim. *Dynamic lotsizing models for stochastic demand in single and multi-echelon inventory systems*. PhD thesis, Ph. D. thesis, Lancaster University, 1996.

[TAW21] Erfan Babaee Tirkolaee, Parvin Abbasian, and Gerhard-Wilhelm Weber. Sustainable fuzzy multi-trip location-routing problem for medical waste management during the covid-19 outbreak. *Science of the Total Environment*, 756:143607, 2021.

[TBM15] SA Torabi, Mansouri Baghersad, and SA Mansouri. Resilient supplier selection and order allocation under operational and

disruption risks. *Transportation Research Part E: Logistics and Transportation Review*, 79:22–48, 2015.

[TDLG09] SA Tassou, Gauthier De-Lille, and YT Ge. Food transport refrigeration–approaches to reduce energy consumption and environmental impacts of road transport. *Applied Thermal Engineering*, 29(8-9):1467–1477, 2009.

[TK04] S Armagan Tarim and Brian G Kingsman. The stochastic dynamic production/inventory lot-sizing problem with service-level constraints. *International Journal of Production Economics*, 88(1):105–119, 2004.

[TK06] S Armagan Tarim and Brian G Kingsman. Modelling and computing (R_n, S_n) policies for inventory systems with non-stationary stochastic demand. *European Journal of Operational Research*, 174(1):581–599, 2006.

[TKTE11] Huseyin Tunc, Onur A Kilic, S Armagan Tarim, and Burak Eksioglu. The cost of using stationary inventory policies when demand is non-stationary. *Omega*, 39(4):410–415, 2011.

[TM17] Shahrzad Tamjidzad and S Hamid Mirmohammadi. Optimal (r, q) policy in a stochastic inventory system with limited resource under incremental quantity discount. *Computers & Industrial Engineering*, 103:59–69, 2017.

[TMPC18] Rafael Tordecilla-Madera, Andrés Polo, and Adrián Cañón. Vehicles allocation for fruit distribution considering co2 emissions and decisions on subcontracting. *Sustainability*, 10(7):2449, 2018.

[TNJ14] Stefan Treitl, Pamela C Nolz, and Werner Jammernegg. Incorporating environmental aspects in an inventory routing problem. a case study from the petrochemical industry. *Flexible Services and Manufacturing Journal*, 26:143–169, 2014.

[TO22] Cansu Turan and Yucel Ozturkoglu. A conceptual framework model for an effective cold food chain management in sustainability environment. *Journal of Modelling in Management*, 17(4):1262–1279, 2022.

[TS08] S Armagan Tarim and Barbara M Smith. Constraint programming for computing non-stationary (R, S) inventory policies. *European Journal of Operational Research*, 189(3):1004–1021, 2008.

[Var09] Vicente Vargas. An optimal solution for the stochastic version of the wagner–whitin dynamic lot-size model. *European Journal of Operational Research*, 198(2):447–451, 2009.

[VBB14] Sahar Validi, Arijit Bhattacharya, and PJ Byrne. A case analysis of a sustainable food supply chain distribution system—a multi-objective approach. *International Journal of Production Economics*, 152:71–87, 2014.

[Vei60] CG Veinott. Synthesis of induction motor designs on a digital computer. *Transactions of the American Institute of Electrical Engineers. Part III: Power Apparatus and Systems*, 79(3):12–18, 1960.

[Vei66] Arthur F Veinott, Jr. On the opimality of (s,s) inventory policies: New conditions and a new proof. *SIAM Journal on Applied Mathematics*, 14(5):1067–1083, 1966.

[Vis20] Andrea Visentin. Computing policy parameters for stochastic inventory control using stochastic dynamic programming approaches. 2020.

[VJW65] Arthur F Veinott Jr and Harvey M Wagner. Computing optimal (s, S) inventory policies. *Management science*, 11(5):525–552, 1965.

[VPRT21] Andrea Visentin, Steven Prestwich, Roberto Rossi, and S Armagan Tarim. Computing optimal (R, s, S) policy parameters by a hybrid of branch-and-bound and stochastic dynamic programming. *European Journal of Operational Research*, 294(1):91–99, 2021.

[VPRT23] Andrea Visentin, Steven Prestwich, Roberto Rossi, and S Armagan Tarim. Stochastic dynamic programming heuristic for the (R, s, S) policy parameters computation. *Computers & Operations Research*, 158, 2023.

[VZ64] GJJ Van Zyl. Inventory control for perishable commodities, Unpublished Ph. D. Dissertation, University of North Carolina, Chapel hill, NC, 1964.

[Web06] A Webb. Ripeness sticker takes the guesswork out of picking and eating. *The Albuquerque Journal*, 1, 2006.

[WEG19] Cynthia Waltho, Samir Elhedhli, and Fatma Gzara. Green supply chain network design: A review focused on policy adoption and emission quantification. *International Journal of Production Economics*, 208:305–318, 2019.

[Win04] Wayne L Winston. *Operations research: applications and algorithm*. Thomson Learning, Inc., 2004.

[WS05] BI Whitaker and M Sullivan. The 2005 nationwide blood collection and utilization survey report. *Washington, DC: US Department of Health and Human Services*, pages 1–24, 2005.

[WW58] Harvey M Wagner and Thomson M Whitin. Dynamic version of the economic lot size model. *Management science*, 5(1):89–96, 1958.

[XRMBT18] Mengyuan Xiang, Roberto Rossi, Belen Martin-Barragan, and S Armagan Tarim. Computing non-stationary (s, S) policies using mixed integer linear programming. *European Journal of Operational Research*, 271(2):490–500, 2018.

[XRT19] Mengyuan Xiang, Roberto Rossi, and S Armagan Tarim. An (r, s) based heuristic model for the stochastic joint replenishment problem. *arXiv preprint arXiv:1902.11025*, 2019.

[YG19] Mohammad Yavari and Mohaddese Geraeli. Heuristic method for robust optimization model for green closed-loop supply chain network design of perishable goods. *Journal of Cleaner Production*, 226:282–305, 2019.

[YK24] Tuğçe Yavuz and Onur Kaya. Deep reinforcement learning algorithms for dynamic pricing and inventory management of perishable products. *Applied Soft Computing*, 163:111864, 2024.

[ZC14] Y Zhang and XD Chen. An optimization model for the vehicle routing problem in multi-product frozen food delivery. *Journal of applied research and technology*, 12(2):239–250, 2014.

[ZHW19] Lu Zhen, Lufei Huang, and Wencheng Wang. Green and sustainable closed-loop supply chain network design under uncertainty. *Journal of Cleaner Production*, 227:1195–1209, 2019.

[Zip00] Paul Herbert Zipkin. Foundations of inventory management. *(No Title)*, 2000.

[Zip08] Paul Zipkin. Old and new methods for lost-sales inventory systems. *Operations research*, 56(5):1256–1263, 2008.

[ZSC16] Huanan Zhang, Cong Shi, and Xiuli Chao. Approximation algorithms for perishable inventory systems with setup costs. *Operations Research*, 64(2):432–440, 2016.