The Retail Location Problem Under Uncertain Demand

Eng. Cristian David Ramírez Pico
Advisor: Ph.D. Michel De Lara
Co-Advisor: Ph.D. William J. Guerrero R.

University: Escuela Colombiana de Ingeniería Julio Garavito

Master in Industrial Engineering

August 17, 2017

ESCUELA COLOMBIANA DE INGENIERÍA JULIO GARAVITO

Acknowledgments

I would like to thank specially the Escuela Colombiana de Ingeniería Julio Garavito for its financial support to attend multiple scientific events about this topic and for giving me technological resources during job's development. Also, I am so grateful with my advisors Ph.D. Michel de Lara and Ph.D. William Guerrero who gave me different tools and sustained motivation to finish this project.

I would like to thank my master student colleagues Eng. Rocío Pérez and Eng. Nicolás Giedelman for their support and their work with me. Also, thanks to the professor Ph.D. Ignacio Pérez who introduced me to the passion for operations research.

Special dedication: To my mother Mary and my siblings Giovanni and Paola who are the reason of my happiness. They always are supporting me and I am so grateful with them because, without their help, I never would have achieved my professional and master titles.

Abstract

We study the problem of a retailer facing uncertainty on the demand. The main objective is to maximize his profit by optimizing the inventory policy and sales, also considering the option to open new selling points. We propose an integrated framework to jointly optimize the strategic and tactical decisions. First, we formulate a *deterministic optimization problem* (with demand known in advance) and we analyze its outcomes. The optimal solution is not satisfying because it suffers from being anticipative. Secondly, *multi-stage stochastic optimization* is considered. We formulate the problem in three different versions with increasing complexity. The first version considers a single retailer (SRLP) and ignores the strategic decision for opening a new selling point. We solve it by *stochastic dynamic programming* and we discuss results. Second and third versions are: a $N-$retailer (NRLP) case where transshipments between retailers are possible; a case where opening decisions of retailers might be made only at the beginning of the time span. Here we propose a new resolution method gathering Stochastic Dual Dynamic Programming and Progressive Hedging algorithms.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

In this chapter, we present the problem (classic in the operations research field). We classify it as a facility location problem combined with tactical and operational decisions. We will look at the general context that arises in our approach, as well as the objectives and state of the art regarding similar problems.

1.1 Motivation

Frequently, entrepreneurs put a product on the market creating retail business, making decisions mostly based on experience, particular observations of the market, personal interests or other qualitative methods.

Once a retailer is beginning to show grow tendencies, some questions appear for the company's manager. E.g. "What is the order sized triggered to suppliers?", "Where is the better place to open a new selling point?", "What is the proper size for the warehouse?". These questions impact directly the strategic and operational decisions of the business.

Retailers are companies that can be found at the last echelon of the supply chain, just where the product is made available for the client. Those retailers are oftentimes working under uncertainty on the demand, given that the customer is who attends the selling point to buy their products. Despite the manager can find some demand behavior, it is unlikely to establish accurately the quantities demanded in future time periods.

This thesis is focused on assessing retailer companies in the process of making decisions related with the above questions and context. The research on this topic has been addressed both to find the optimal performance for the entire supply chain and for first or intermediate echelons of it (plants, warehouses, collections points, etc). Often, researchers develop mathematical models to decide about inventory management, transshipments at the same echelon or between them. Also, researchers have studied widely the facility location theory, as we will see in §1.3. Nonetheless, rarely we find studies with simultaneous decisions.

Recently, Laporte et al. [1] carried out a survey about facility location tendencies under stochasticity conditions in one or more components of the problem. This text highlights several proposes around multi-stage model optimizing opening and closing decisions at any moment over a previous defined time span.

We propose a mathematical model denoted as *Retail Location Problem* or (RLP), wherein we consider multiple stages with decisions about inventory management, sales and location of new possible selling points. The objective is to maximize the retailer total profit, taking into account *Net Present Value* of the cash flows across the time. In order to provide a better comprehension of the problem and its components, we have developed two phases that progressively increase both the real-life case representation and the complexity to solve the model:

1. A deterministic version whereby the demand can be anticipated, it means that the manager makes decisions knowing the future in advance.

2. A multi-stage stochastic problem, where the demand is a random variable. This proposal in turn will be decomposed into others:

   (a) A model in which we only consider one retailer that manages its inventory policy and sales (SRLP)

   (b) The previous proposal evolves when we have two or more selling points. Here the decisions are made about particular inventory policies, transshipments between them and sales (NRLP)

   (c) Finally, we present the problem that includes strategic (location) decisions added to the NRLP and this is what we call *Retail Location Problem* (RLP)

1.2 Objectives and research questions

1.2.1 Objectives

General

Formalize mathematically the RLP aimed to optimize the total profit of a retailer company working under uncertain demand and apply an adapted method to solve its different versions

Specifics

- Transform the deterministic RLP in several stochastic programs
- Solve the stochastic optimization programs (SRLP, NRLP and RLP) using adapted method for each one of them
- Carry out a sensitivity analysis of the problem using several random instances

1.2.2 Research Questions

- How can we transform the deterministic RLP in multiple stochastic programs?
- Which adapted methods exist and how do we apply it to solve the stochastic programs?
- How can we do a sensitivity analysis for the SRLP, NRLP and RLP?

1.3 Literature review

In this section, we present a brief discussion about three main topics: inventory management, sales policies and location decision over the supply chain and its different echelons. Furthermore, we present the solution methods used on the problems described and related with our proposal.

Initially, we talk about facility location problems. Snyder [2] shows multiple historical approaches around this problem, specially for our interest, cases with simultaneous strategic and tactical or operational decisions, which had not been widely studied before 2006. The first authors considering stochasticity for this kind of problem were Alonso-Ayuso et al [3], Santoso et al. [4], Snyder et al. [5], Öszen et al. [6] and Atamtürk et al. [7] who consider: two-stage stochastic programs for supply chain design [3], [4], while [5], [6] and [7] consider only one echelon of the supply (warehouses).

To our knowledge only one paper studies the problem of facility location decisions under uncertainty for retailers, Fernández et al. [8] wherein their aim is to maximize the market participation of the new selling point, inventory and sales management are out of consideration in this study.

Hinojosa et al. [9] propose a multi-stage stochastic program studying opening and closing decisions, also they include inventory management acquiring pretty similar features regarding our proposal. However, they considered whole supply chain forbidding transshipments between facilities at the same echelon.

Zadeh et al. [10], Albareda-Sambola et al. [11], Ghaderi et al. [12], Álvarez-Miranda et al. [13] and Shiina et al. [14] also present related literature. In problems of location and minimization costs, they used several exact resolution methods as L-Shape method, branch-and-bound, Benders decomposition, as well metaheuristic methods, greedy algorithms or Fixed-and-relax coordination [11], [10], [12] and [13] present study cases for supply chains of steel, health care and disaster management. The main concepts of our literature review so far are in Table 1.1.

There is no a related model in Table 1.1 solved through *Dynamic Programming* or *Stochastic Dynamic Programming (SDP)*, this is a special characteristic as we will see in future paragraphs and sections. Hinojosa et. al. [9] present a multi-stage stochastic model solved by lagrange-ralaxation, considering location and inventory decisions.

Now, we are going to review approaches regarding inventory management in only one facility, also trying to find considerations on lost sales and policies over orders and sales. Morton in [16] and [17], Nahmias [18], [19] and [20] propose heuristics methods to solve the inventory problems for periodic review systems under different conditions such as lead times, backlogged orders, lost sales and others. Zipkin [21] introduces several ways to manage inventory in facilities and also, stochastic cases where the complexity increases when lost sales instead of

Author	Year	Location	Inventory	Transshipments	Objective
Shiina et al. [14]	2014	×	✓	✓	Costs Minimization
Álvarez-Miranda et al. [13]	2014	✓	×	×	Costs Minimization
Ghaderi et al. [12]	2013	✓	×	×	Costs Minimization
Albareda-Sambola et al. [11]	2013	✓	×	×	Costs Minimization
Nickel et. al. [15]	2012	✓	×	×	Profit maximization
Atamtürk et al. [7]	2012	✓	✓	×	Costs Minimization
Zadeh et al. [10]	2011	✓	✓	×	Costs Minimization
Hinojosa et al. [9]	2008	✓	✓	×	Costs Minimization
Oszen et al. [6]	2008	✓	✓	×	Costs Minimization
Snyder et al [5]	2007	✓	✓	×	Costs Minimization
Fernández et al [8]	2006	✓	×	×	Market participation maximization
Snyder [2]	2005	NA	NA	NA	Minimize, maximize minimax, minisum
Santoso et al [4]	2005	✓	×	✓	Costs Minimization
Alonso-Ayuso et al. [3]	2003	✓	No	✓	Profit maximization

Table 1.1: Literature review strategic and tactical decisions

backorders are considered. Beyer et. al [22] introduce a treatment for Markovian demands and non-linear costs.

Close to our matter, we can find [23] introducing an effective method for models with lost sales. Nonetheless, we still have significant differences regard his work, due to the maximization of profit (sales random variables) and not the minimization of costs posed by Zipkin (not sales random variables included). In the same way, Levi [24] conducts dual balancing methods for problem in Zipkin proposal. In 2011, Bijvank et. al [25] introduce the basic theory to manage lost sales under different policies, always considering costs minimization. A last approach by Li [26] uses stochastic dynamic programming for a minimization problem with lost sales including returned orders.

At the end, we review bibliography regarding resolution methods for stochastic multi-stage programs, particularly on stochastic dual dynamic programming (SDDP). Bertsekas [27] and Shapiro et. al. [28] tackle the problem of decisions about inventory management for one facility under uncertainty and show its theoretical solution. However, there is no a proposal for transshipments or facility open decisions. Also, Shapiro et. al [28] give the principles of SDDP. On the other hand, King et. al [29] present the complexity to work with SDP to get exact solutions, even when we solve "small" size instances and a "little" set of constraints (those instances will be shown further on); they also give basic concepts to take into account net present value of money in time, mainly in objective functions.

Last, Ross [30] defines a concept of Positive Dynamic Programming showing how to apply that in gambling theory and its implementations.

Chapter 2

The Retail Location Problem Statement

This chapter presents the general structure of the Retail Location Problem (RLP) including statement, variables, dynamics, criterion, logistics considerations and assumptions. Across the document we will often refer to the problem context in order to apply and modify the main concepts, according to each model posed, one deterministic and three stochastic.

2.1 Problem context

Here, we consider a supply chain composed of a set of retailers, with centralized management, selling a single product. This kind of business is at the end of a chain serving the final consumer. We study a case with only one supplier to all retailers. The problem in consideration takes into account the optimization of location decisions, inventory management and transshipments for each selling point.

The decisions are framed in some important assumptions in terms of logistics and supply chain management:

- If the total demand is greater than the stock plus incoming orders, those are Lost Sales
- If the company has a customer with demand greater than the current stock Partial Sales are allowed

2.1.1 Assumptions

We present some convenient assumptions to tackle the problem:

1. Transshipments are allowed between selling points
2. Lead times are neglected, once you put an order to the supplier it will be available for the same time period
3. The demand is treated as a random variable during the second phase
4. We consider price and costs as constants over whole time horizon
5. The fixed costs triggering an order to the supplier is assumed as zero
6. The investment budget to open new facilities is unlimited
7. There is a finite and fixed set of candidates facilities
8. We assume a known discount rate for the time value of money (NPV)

In addition, we have some considerations very useful to define the problem size:

- the time span is finite and discrete, where

  $$t \in \overline{\mathbb{T}} = \{t_0, t_0 + 1, \dots, T\}, \tag{2.1}$$

  and

  $$t \in \mathbb{T} = \{t_0, t_0 + 1, \dots, T - 1\}, \tag{2.2}$$

  and t denotes the beginning of the period $[t, t+1[$; we call t_0 the *initial time* and T denotes the *last time* period.

- we denote the set of facilities as

  $$i \in \mathbb{I} = \{1, \dots, I\} \tag{2.3}$$

  this set is split in two disjoint subsets, $\mathbb{I}^o$ represents the subset of *open facilities* and $\mathbb{I}^c$ is the subset that contains the *candidate facilities*, with

  $$\mathbb{I} = \mathbb{I}^o \cup \mathbb{I}^c \quad \text{and} \quad \mathbb{I}^o \cap \mathbb{I}^c = \emptyset. \tag{2.4}$$

2.1.2 Parameters

As we mentioned in §2.1.1, *Net Present Value* is considered in order to tackle the problem as an investment project for the company. Also, we establish associated parameters to compute incomes and costs

Hereafter, whenever we speak of indexes t and i, it is assumed that each one are included in sets $\overline{\mathbb{T}}$ and $\mathbb{I}$ respectively:

- r is a discount rate used in discounted cash flow,
- c_i^f fixed costs arise to open a new selling point, it will be zero when the facility i is already open
- p_t unitary selling price at the time t
- c_t^p unitary purchase price at the time t
- c_{it}^d unitary cost for the transportation from the supplier to the retailer i at the time t
- c_{it}^{so} unitary cost of stockout at the retailer i and time t
- c_t^h unitary holding cost at time t
- c_{ijt}^e unitary cost to transshipment product from the retailer i to retailer j at time t
- $S_i^\sharp$ storage capacity in the retailer i

2.2 Variables

We are going to present two kind of variables that intervene in our mathematical formulation.

2.2.1 Decision variables

Here, we present the decision variables of the problem. Later, we will classify them, based on standards frequently used to tackle stochastic programs:

- $\mathbf{Y}_i$ *Location* are the decisions about to keep open or closed a selling point i during the time span; these are boolean variables wherein 1 means to open the selling point and 0 the opposite choice
- $\mathbf{S}_{it}$ *Stock* is the quantity of product storage at the beginning of period $[t, t+1[$ at the retailer i, belonging to the set $\mathbb{S} = [0, S^\sharp]$

- $\mathbf{Q}_{it}$ *Order Size* is the quantity of product bought by the retailer i to the supplier, available at the beginning of the period $[t, t+1[$, belonging to the set $\mathbb{Q} = [0, Q^\sharp]$
- $\mathbf{M}_{ijt}$ *Transshipment* is the quantity of product delivered from the retailer i to retailer j at the beginning of the period $[t, t+1[$
- $\mathbf{O}_{it}$ *Sales* at the retailer i during the time period $[t, t+1[$
- $\mathbf{F}_{it}$ *Stockout* at the retailer i during the time period $[t, t+1[$

2.2.2 Uncertain variables

We represent the sequence of demands as a stochastic process denoted $\{(\mathbf{W}_{it})_{t\in\overline{\mathbb{T}},\ i\in\mathbb{I}}\}$. Particularly, $\mathbf{W}_{it}$ is the random variable of the demand at location i during the period $[t, t+1[$ belonging to the set $\mathbb{W} = \left[W^\flat, W^\sharp\right]$, where $W^\flat$ is the minimum level of the demand and $W^\sharp$ is the maximum.

2.3 Mathematical propositions

We now introduce mathematical expressions for inventory flow and the decision maker's objective as below.

2.3.1 Dynamics

The dynamics help us to hold a correct inventory flow across consecutive time periods

$$\mathbf{S}_{i,t+1} = f_{it}(\mathbf{S}_{it}, \mathbf{Q}_{it}, \mathbf{M}_{ijt}, \mathbf{O}_{it}),\ \forall\ i \in \mathbb{I},\ t \in \mathbb{T}. \tag{2.5}$$

Both in deterministic and stochastic RLP versions, the dynamics has some special modifications. We present a generic function of the future stock depending of the previous stock, order size, transshipments and sales.

2.3.2 Criterion

The decision maker's problem is to *maximize the profit* taking into account price and costs presented in §2.1.2 for each period over the time horizon. Total profit is computed as follows:

$$\sum_{i\in\mathbb{I}} \left[-c_i^f \mathbf{Y}_i - \sum_{t\in\overline{\mathbb{T}}} \left((p_t - c_t^p)\mathbf{O}_{it} - c_{it}^{so}\mathbf{F}_{it} - c_{it}^d \mathbf{Q}_{it} - c_t^h \mathbf{S}_{it} - \sum_{j\in\mathbb{I}} c_{ijt}^e \mathbf{M}_{ijt} \right) \right]. \tag{2.6}$$

The criterion computes tidily: location decision costs, gross income, minus stockout costs, minus delivery costs, minus inventory holding costs minus transshipments costs.

Why did we introduce all this? As we mentioned in §1.1, there is a series of different problems related to the Retail Location Problem (RLP). All concepts introduced in this chapter will be used in several ways across the document, in such way that this is a general framework to tackle the problem.

Chapter 3

Deterministic Retail Location Problem

This chapter present our first approach to the Retail Location Problem (RLP). We begin with the deterministic problem statement, then we formulate the mathematical model and finally we show an useful analysis for next chapters.

3.1 Optimization problem ingredients

We remember that the main motivation to do this project is studying the *Retail Location Problem.* Our first approximation is a Mixed Integer Linear Program (MILP) where we consider location decisions for new selling points under deterministic demands. Now, We present some considerations with respect to statements introduced in chapter 2.

3.1.1 Definitions

We use the decision variables described in §2.2.1. However, we need a special consideration to distinguish deterministic and stochastic versions.

Remark 1 *In terms of problem's notation, we present random variables in bold type, while in deterministic version those variables are typed in small. For example, the demand* W_{it} *is a parameter and not a random variable* $\mathbf{W}_{it}$.

The following proposal of RLP can be formulated as a Mixed Integer Program where the demand can be anticipated and it is just one input for the model.

3.1.2 Criterion

We consider the criterion defined in equation (2.6) changing only the boldface on variables

$$-\sum_{i\in\mathbb{I}}\left[c_i^f Y_i - \sum_{t\in\overline{\mathbb{T}}}\left((p_t - c_t^p)O_{it} - c_{it}^{so}F_{it} - c_{it}^d Q_{it} - c_t^h S_{it} - \sum_{j\in\mathbb{I}} c_{ijt}^e M_{ijt}\right)\right], \tag{3.1}$$

3.2 Optimization problem

Before explaining the optimization problem's structure, we clarify the notation used above: $O_{(i.)}$, $Q_{(i.)}, S_{(i.)}$, $F_{(i.)}, M_{(ij.)}$ are sequences that represent periodic decisions over the time span:

$$Z_{(i.)} = \{Z_{i,t_0}, \ldots, Z_{i,T}\},\ \forall\ i \in \mathbb{I}. \tag{3.2}$$

Our goal is to maximize the profit using the criterion described previously under a set of constraints. Then, the optimization problem is performed as follows:

$$\max_{Y_i,O_{(i.)},Q_{(i.)},S_{(i.)},F_{(i.)},M_{(ij.)}} -\sum_{i\in\mathbb{I}}\left[c_i^f Y_i - \left(\sum_{t\in\overline{\mathbb{T}}}((p_t - c_t^p)O_{it} - c_{it}^{so}F_{it} - c_{it}^d Q_{it} - c_t^h S_{it} - \sum_{j\in\mathbb{I}} c_{ijt}^e M_{ijt})\right)\right], \tag{3.3a}$$

s.t.

$$S_{it} \leq Y_i S_i^\sharp, \quad \forall\ i \in \mathbb{I},\ t \in \overline{\mathbb{T}} \tag{3.3b}$$

$$O_{it} + F_{it} = Y_i W_{it}, \quad \forall\ i \in \mathbb{I},\ t \in \overline{\mathbb{T}} \tag{3.3c}$$

$$Y_i = 1, \quad \forall\ i \in \mathbb{I}^o \tag{3.3d}$$

$$S_{i,t_0} = 0, \quad \forall\ i \in \mathbb{I} \tag{3.3e}$$

$$M_{iit} = 0, \quad \forall\ i \in \mathbb{I},\ t \in \overline{\mathbb{T}} \tag{3.3f}$$

$$\sum_{j\in\mathbb{I}} M_{ijt} \leq S_{it}, \quad \forall\ i \in \mathbb{I},\ t \in \overline{\mathbb{T}} \tag{3.3g}$$

$$S_{i,t+1} = S_{it} - O_{it} + Q_{it} + \sum_{j\in\mathbb{I}} M_{jit} - \sum_{j\in\mathbb{I}} M_{ijt},\ \forall\ i \in \mathbb{I},\ t \in \mathbb{T} \tag{3.3h}$$

$$O_{i,T} = S_{i,T} + Q_{i,T}, \quad \forall\ i \in \mathbb{I} \tag{3.3i}$$

$$\begin{gathered} Y_i \in \{0,1\} \\ O_{it},\ Q_{it}, S_{it},\ F_{it}, M_{ijt} \geq 0 \quad \forall\ i,j \in \mathbb{I},\ t \in \overline{\mathbb{T}} \end{gathered} \tag{3.3j}$$

The objective function (3.3a) is the *Net Present Value* of the profit for all retailers, accumulated through the time span.

The constraints (3.3b) ensure that the inventory in each retail does not exceed the maximum capacity of storage for every period. Constraints (3.3c) keep the balance between sales, stock-out and demand for every period. The constraints (3.3d) force the retailers (in the set $\mathbb{I}^o$) to remain open. Constraints (3.3f) and (3.3g) prohibit the shipments from a retail to itself and ensure that the transshipments are done only from open selling points. (3.3e) is used to set the initial inventory in every selling point.

Moreover, the dynamics in equation (2.5) become in the constraints (3.3h) for our MIP, guarantying the inventory flow conservation through all periods, and at the horizon (last time period) the sales are calculated by (3.3i).

Finally, constraints (3.3j) denote the nature of the decision variables.

3.3 Computational experiments

After displaying the mathematical model, we proceed to solve some instances through classic resolution methods. Then, we conclude and analyze the results.

3.3.1 Numerical data

We have developed 50 instances with $T = 72$, $\overline{\mathbb{T}} = \{1, 2, \ldots, 72\}$ periods for the planning horizon and $I = 10$, $\mathbb{I} = \{1, 2, \ldots, 10\}$ facilities whereby, between 2 and 5 were opened at the beginning of the time span $\mathbb{I}^o$ and the other were candidate facilities $\mathbb{I}^c$. The features for these instances were:

- $\mathbb{I} = 10$ facilities opened and candidates
- For the NPV we use a discount rate $r = 6\%$ per year or 0.5%
- Some parameters for the model were randomly generated using uniform distributions as follows[1]:

[1] Naturally we have used some particular probability distributions as a strategy to build our instances, specifically to generate parameters as costs and demand. This does not mean that we are working with random variables, but that we develop instances under those assumptions.

- Fixed cost to open a new selling point c_i^f between $\$50,000,000$ and $\$250,000,000$ COP
- Delivery costs c_{it}^d between \$600 and $\$1,200$ COP
- Transshipment costs c_{ijt}^e between \$200 and \$400 COP
- Retailers' capacity $S_i^\sharp$ between 50 and 200 units of product

- Each retailer had an associated demand W_{it} created randomly for the instances by a normal distribution with a mean between 80 and 120, and a standard deviation between 8 and 12 units of product.

- The unity gain, given by the difference between price p_t and Unitary cost c_t^p is $\$15,000$ COP

- Inventory holding cost c_t^h was $\$1,000$ COP

- Stockout cost c_{it}^{so} was $\$1,500$ COP

The size of the set $\overline{\mathbb{T}}$, chosen by 72 months, is one of the most common time horizon which general managers and/or investors impose to recover the money invested in some project (open new selling points in this case).

3.3.2 Resolution method

To solve our linear model (3.3), we use Mixed Integer Linear Programming solver CPLEX in GAMS Software Version 23.5. CPLEX uses branch and cut algorithm to find the optimal solution for this kind of problems.

For all numerical experiments given in this paper we use a computer with the following features: Intel® Core(TM) i7-5500U CPU 2.40Ghz Processor and 16Gb RAM.

The deterministic RLP does not generate problems in terms of computational times, given that we work with small size (for deterministic problems) instances $(I * T)$. The largest instance studied had a total of $3,610$ variables.

3.3.3 Results analysis

Computational times

The time to solve the instances is less than 1 second, a very small time regarding the time span.

Results

After solving all instances using the data showed in §3.3.1 we found:

- Since remark 1, the variables and parameters are deterministic, then:

  Conjecture 1 *The demand can be anticipated and the inventory, stockout and transshipments increase the cost. Thus, the maximum of the problem* (3.3) *comes when these variables are equal to zero for every period at each facility, provided that the retailers' capacity is not violating. Mathematically it means that, at the optimum, we expect:*

  $$\sum_{i\in\mathbb{I}}\sum_{t\in\mathbb{T}}(c_{it}^{so}F_{it} + c_t^h S_{it} + \sum_{j\in\mathbb{I}} c_{ijt}^e M_{ijt}) = 0. \qquad (3.4)$$

  The mathematical proof of (3.4) is not available. We admit (3.4) as true and its demonstration is a pendant issue for future research.

- Related with the previous assertion, in Figure 3.1 you can see how, in practice: orders, demand and sales are exactly equal, leaving zero transshipments, inventory and stockout

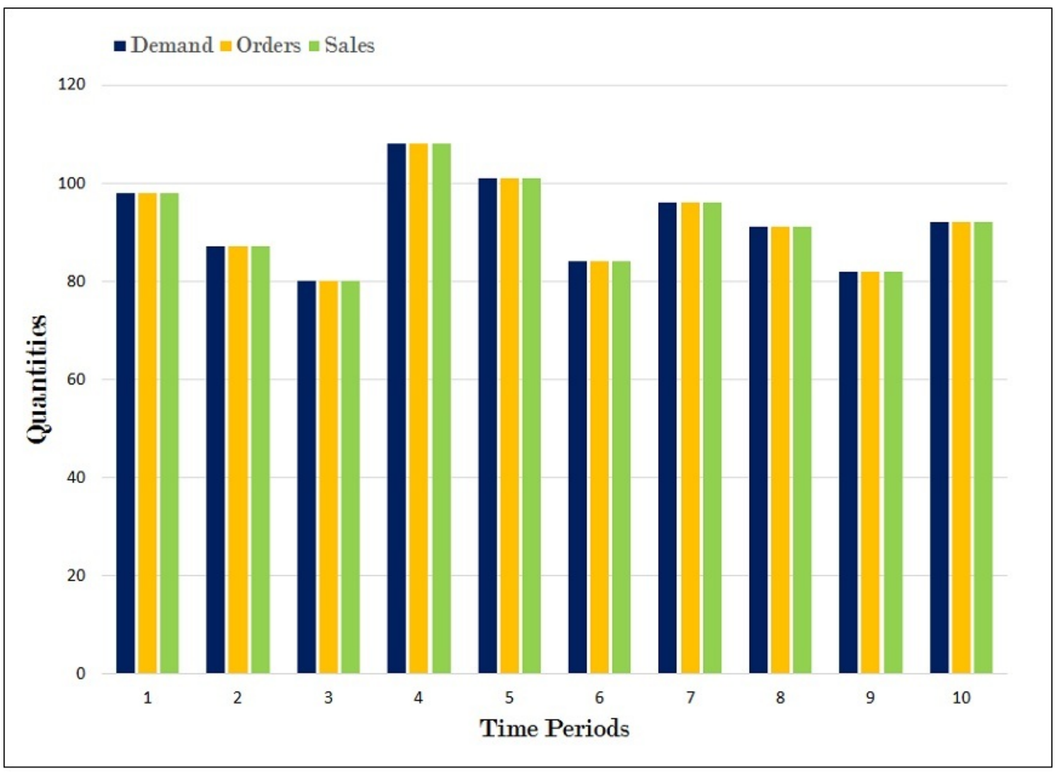

Figure 3.1: Deterministic RLP. Instance 4, selling point 2, time periods 1 to 10

- Based on Conjecture 1 we can propose the rule to open any new facility as a main result of the program

  1. Firstly, we clarify that this rule is true iff sales are no limited to reach the same demand quantity, that is to say, the orders do not have a limit (supplier's capacity is not restricted):

  $$O_{(i.)} = Y_i W_{(i.)}, \quad \forall\ i \in \mathbb{I}$$

  2. Given that the incomes are defined by the sales and the inventory is zero always:

  $$O_{(i,.)} = Q_{(i,.)} \quad \forall\ i \in \mathbb{I}$$

Using all previous conclusions, the optimal decision rule for opening decisions should be:

forall facilities $i \in \mathbb{I}$

if:

$$\sum_{t\in\overline{\mathbb{T}}} (p_t - c_t^p) O_{it} \geq c_i^f + \sum_{t\in\overline{\mathbb{T}}} c_{it}^d Q_{it}$$

then:

$Y_i = 1;$ and
$O_{(i.)} = W_{(i.)}$

else

$Y_i = 0;$ and
$O_{(i.)} = 0$

The above procedure implies that, if the total gross income is greater than the fixed costs for opening location plus total delivery costs, the facility i must be opened, but in contrary case, it must remain closed.

3.4 Final considerations

The conclusions presented here are valid under a classical cost's structure, where the stockout cost is greater than unitary profit, other cases are outside of our main interests.

The Deterministic Model is a good initial approach to understand the system's behavior. Nonetheless, the proposed formulation is not the best representation of real-life situations, given that knowing the demand in advance, the optimal strategy ensures that inventory, stockout and transshipments never occur, which is rarely the case in real life.

Henceforth, we will propose stochastic programs in order to improve the representation of real problems.

Chapter 4

Single Retailer Stochastic Problem

This chapter presents the first stochastic approach to RLP. We tackle the problem with only one selling point, and we aim to optimize tactical decisions, orders and sales, which impact directly on inventory holding and stockout. First, the mathematical framework is posed, then we formulate the optimization problem and outline the resolution method chosen. Finally, we present numerical tests including analysis and conclusions respectively.

4.1 Optimization problem ingredients

As in the deterministic problem, we present some mathematical considerations with respect to the general framework introduced in Chapter 2.

Remark 2 *We introduce a probability space* $(\Omega, \mathcal{A}, \mathbb{P})$ *which describes the behavior of the uncertain variables, and* $\mathbb{E}$ *the mathematical expectation.*

4.1.1 Parameters

We can use the parameters defined in §2.1.2, considering that index i is not necessary here, due to we only are studying the operation of one selling point.

4.1.2 Information's structure

Earlier, we have considered the period $[t, t+1[$ as a time slot wherein simultaneously occurs orders, sales, transshipments, stockouts and demands. From now on we propose a model that represents the chronology of events. Thus, we establish a neat calendar of the events during $[t, t+1[$:

1. **At the beginning** of the interval time we observe a stock at the retailer
2. **Immediately**, we put an order to the supplier, which is available instantly
3. The initial stock plus the incoming order is the product available to satisfy the demand that comes **during** the interval
4. The sales come **during** the interval
5. **At the end** of the time slot we are sure about the total lost sales (stockout)

The manager must make decisions at the beginning of $[t, t+1[$, before knowing the demand that will occur during this period. But, he also will be making decisions once the uncertainty has been revealed.

A decision with index t means that it can only depend on what happened before t.

4.1.3 Problem variables

The decision variables suffer some changes with respect to previous definitions in §2.2.1. We classify them in two big categories: state and control variables. State are partially controllable variables, since they are a result of the random and control variables. Controls are variable which we can decide in order to change the system state. Then, we modify the variables index based on considerations of §4.1.2.

- State variables:
  - $\mathbf{S}_t$ stock of product **at the beginning** of $[t, t+1[$ and belonging to the set $\mathbb{S} = \left[0, S^\sharp\right]$
- Control variables: they are split in two different groups depending if they are made **at the beginning** or **during and at the end** of the time slot $[t, t+1[$:

  The controls made before to the realization of the noise (identified by t index) are

  - $\mathbf{Q}_t$ ordered product **decided at the beginning** of $[t, t+1[$ belonging to $\mathbb{Q} = \mathbb{Z}^+ \cup 0$

  Moreover, we add the control made once the uncertainty has been revealed (identified by $t+1$ index):

  - $\mathbf{O}_{t+1}$ sales made **during** $[t, t+1[$,

– $\mathbf{F}_{t+1}$ stockout occurred **at the end** of $[t, t+1[$. We also can present this variable as the difference between demand and sales:

$$\mathbf{F}_{t+1} = \mathbf{W}_{t+1} - \mathbf{O}_{t+1}. \tag{4.1}$$

For practicality on notation, we replace $\mathbf{F}_{t+1}$ in the rest of the document with the expression in equation (4.1).

Before closing this section, we have to highlight that state and control variables are also random, due to the direct relations between *Demand, Orders, Sales, Inventory* and *Stockout,* as we will show in §4.2.1.

4.1.4 Criterion

The decision maker aims to maximize the *expected profit* over the time span, but its profit depends on the price and costs mentioned in §2.1.2. We sum over $\overline{\mathbb{T}}$ and consider revenue minus operational costs

$$\mathbb{E}\left[\sum_{t\in\mathbb{T}} L_t(\mathbf{S}_t, \mathbf{Q}_t, \mathbf{O}_{t+1}, \mathbf{W}_{t+1}) + K(\mathbf{S}_T)\right], \tag{4.2}$$

which K is the final profit, obtained according to the stock at the final period of the horizon, and L_t is called the instantaneous profit:

$$\begin{aligned} K(\mathbf{S}_{iT}) &= \text{Final profit, period time } T, \\ L_t(\mathbf{S}_t, \mathbf{Q}_t, \mathbf{O}_{t+1}, \mathbf{W}_{t+1}) &= \underbrace{(p_t - c_t^p)\mathbf{O}_{t+1} - c_t^{so}(\mathbf{W}_{t+1} - \mathbf{O}_{t+1}) - c_t^d\mathbf{Q}_t - c_t^h\mathbf{S}_t}_{\text{Instaneous profit}}. \end{aligned} \tag{4.3}$$

L_t is composed of: revenue, stockout, ordering and inventory holding costs.

4.1.5 Uncertain demand

Stochastic Process: We define the demand as a stochastic process, measurable with respect to probability space in Remark (2), and denoted by $(\mathbf{W}_t)_{t\in\overline{\mathbb{T}}}$.

In addition, a *Scenario Demand* is a sequence:

$$\mathbf{W}(\cdot) := (\mathbf{W}_{t_0+1}, \mathbf{W}_{t_0+2}, \dots, \mathbf{W}_T). \tag{4.4}$$

Also, we introduce

$$\mathcal{F}_t = \sigma\left(\mathbf{W}_{t_0+1}, \mathbf{W}_{t_0+2}, \dots, \mathbf{W}_t\right), \tag{4.5}$$

as the σ-field generated by the stochastic process of the demand between the periods $\{t_0, t_0+1, \dots, t\} \in \overline{\mathbb{T}}$

White noise assumption: As we will see, Stochastic Dynamic Programming (SDP) is the resolution method chosen to solve the problem. To use SDP we need some probabilistic assumptions over the random variables:

- the random variables $(\mathbf{W}_{t_0+1}, \mathbf{W}_{t_0+2}, \ldots, \mathbf{W}_T)$ are independent between them. Independence is a key assumption on SDP implementations.

- the following marginal distributions describe the behavior of the demand over the set $\mathbb{W}$:
$$\mathbb{P}\{\mathbf{W}_t = w\} = \xi_{tw}, \tag{4.6}$$

we have noticed that random variables are independent, but not necessarily identically distributed. This allows us to take into account, for example, seasonal effects on the demand.

4.1.6 Dynamics

The behavior of the retailer's stock described in equation (2.5) must be modified as:

$$\mathbf{S}_{t+1} = f_t(\mathbf{S}_t, \mathbf{Q}_t, \mathbf{O}_{t+1}) \quad \forall\, t \in \mathbb{T}. \tag{4.7}$$

In this case, the following equations guarantee the flow of the inventory between subsequent periods of time:

$$\begin{aligned} \mathbf{S}_{t_0} &= S_{in} && (4.8a)\\ \mathbf{S}_{t+1} &= \mathbf{S}_t + \mathbf{Q}_t - \mathbf{O}_{t+1},\ \forall\, t \in \mathbb{T} && (4.8b) \end{aligned}$$

Where S_{in} is the inventory at the beginning of the time horizon (Initial inventory)

4.1.7 Constraints

On the other hand, we have multiple linear constraints helping us to hold a good representation of a real-life case

$$\begin{aligned} \mathbf{O}_{t+1} \leq{} & \mathbf{W}_{t+1}, && \forall\, t \in \overline{\mathbb{T}} && (4.9a)\\ \mathbf{O}_{t+1} \leq{} & \mathbf{S}_t + \mathbf{Q}_t, && \forall\, t \in \overline{\mathbb{T}} && (4.9b)\\ \mathbf{S}_t + \mathbf{Q}_t \leq{} & S^\sharp, && \forall\, t \in \overline{\mathbb{T}} && (4.9c)\\ 0 \leq \mathbf{Q}_t \leq{} & Q^\sharp, && \forall\, t \in \overline{\mathbb{T}} && (4.9d)\\ 0 \leq{} & \mathbf{O}_{t+1}, && \forall\, t \in \overline{\mathbb{T}} && (4.9e)\\ 0 \leq{} & \mathbf{S}_t, && \forall\, t \in \overline{\mathbb{T}} && (4.9f) \end{aligned}$$

The constraints (4.9a) ensure sales less or equal to the demand of the period $[t, t+1[$. (4.9b) guarantee that sales do not exceed the available quantity of product. (4.9c) keep the stock under the retailer's capacity. (4.9d) do not allow ordering negative or most that the limit quantities of product. (4.9e) and (4.9f) ensure positives or zero quantities of stock and sales.

4.2 Optimization problem formulation

Using §4.1, we propose the following optimization problem and some considerations about its solution.

4.2.1 Optimization problem

The manager of the company aims to maximize the criterion

$$\max_{\mathbf{S},\mathbf{Q},\mathbf{O}} \mathbb{E}\left[\sum_{t\in\mathbb{T}} L_t(\mathbf{S}_t, \mathbf{Q}_t, \mathbf{O}_{t+1}, \mathbf{W}_{t+1}) + K(\mathbf{S}_T)\right], \tag{4.10a}$$

Subject to:

(4.8), (4.9) and

$$\sigma(\mathbf{Q}_t) \subset \mathcal{F}_t, \quad \forall\, t \in \mathbb{T} \tag{4.10b}$$

$$\sigma(\mathbf{O}_t) \subset \mathcal{F}_t, \quad \forall\, t \in \mathbb{T} \tag{4.10c}$$

The objective (4.10a) is to maximize the expected value of the profit under conditions established in dynamics and constraints presented. Finally, (4.10b) and (4.10c) are the non-anticipativity constraints expressed as measurability of the controls with respect to the history of the demand. Recall that $\sigma(\mathbf{X})$ is the σ-field generated by the random variable $\mathbf{X}$.

As we can see, this problem has a large number of constraints (4.9). Nonetheless, all mathematic expressions are linear. Non-anticipativity constraints are linear, since they can be written as:

$$\mathbf{X}_t = \mathbb{E}\left[\mathbf{X}_t|\mathcal{F}_t\right], \tag{4.11}$$

for any control variable, in our case orders and sales restricted in (4.10b) and (4.10c). the mathematical proof of (4.11) can be consulted in Shapiro et. al. [28].

For simplicity, we propose a more compact formulation that allows us to set our problem in any linear solver under algebraic modeling:

$$\max_{\mathbf{S},\mathbf{Q},\mathbf{O}} \mathbb{E}\left[\sum_{t\in\overline{\mathbb{T}}} L_t(\mathbf{S}_t,\mathbf{Q}_t,\mathbf{O}_{t+1},\mathbf{W}_{t+1}) + K(\mathbf{S}_T)\right] \tag{4.12a}$$

S.t.

$$\begin{aligned} \mathbf{S}_{t+1} &= f_t(\mathbf{S}_t,\mathbf{Q}_t,\mathbf{O}_{t+1}) && (4.12b)\\ G\ \mathbf{S}_t + H\ \mathbf{Q}_t + I\ \mathbf{O}_{t+1} &\leq b_t && (4.12c)\\ \mathbf{Q}_t &= \mathbb{E}\left[\mathbf{Q}_t|\mathcal{F}_t\right] && (4.12d)\\ \mathbf{O}_t &= \mathbb{E}\left[\mathbf{O}_t|\mathcal{F}_t\right] && (4.12e) \end{aligned}$$

in which (4.12a) and (4.12b) hold the sense shown in dynamics (4.8) and non-anticipativity constraints respectively. Moreover, the arrays in (4.12c) are defined as follows:

$$G = \begin{pmatrix} 0\\ -1\\ 1\\ 0\\ 0\\ -1\\ 0 \end{pmatrix} \quad H = \begin{pmatrix} 0\\ -1\\ 1\\ -1\\ 1\\ 0\\ 0 \end{pmatrix} \quad I = \begin{pmatrix} 1\\ 1\\ 0\\ 0\\ 0\\ 0\\ -1 \end{pmatrix} \quad b_t = \begin{pmatrix} w\\ 0\\ S^\sharp\\ 0\\ Q^\sharp\\ 0\\ 0 \end{pmatrix}$$

w represents a realization of the demand ($\mathbf{W}_t$) at time t.

4.2.2 Solution space

We tackle the problem using *Stochastic Dynamic Programming* (SDP). As a result of this, we get a *strategy* or a *policy*.

Policy [1] : is a decision rule (a function); also, a *policy*, can be shown as a sequence of measurable functions $\pi = \{\pi^q, \pi^o\}$, where $\pi^q = (\pi^q_1, \pi^q_2, \ldots, \pi^q_{T-1})$ and $\pi^o = (\pi^o_2, \pi^o_3, \ldots, \pi^o_T)$. For any stock s, we find the controls q and o.

Specifically, an admissible policy is mapping by $\pi^q : \mathbb{T}\times\mathbb{S}\to\mathbb{Q}$; $\pi^o : \mathbb{T}\times\mathbb{S}\times\mathbb{W}\times\mathbb{Q}\to\mathbb{O}$, this policy yields uncertain trajectories for states and controls:

$$\begin{aligned} \mathbf{S}(\cdot) &:= (\mathbf{S}_{t_0},\ldots,\mathbf{S}_{T-1},\mathbf{S}_T)\\ \mathbf{Q}(\cdot) &:= (\mathbf{Q}_{t_0},\ldots,\mathbf{Q}_{T-1})\\ \mathbf{O}(\cdot) &:= (\mathbf{O}_{t_0+1},\ldots,\mathbf{O}_T) \end{aligned} \tag{4.13}$$

[1] under the white noise assumptions 4.1.5, solutions can be looked after as policies

If we restrict the solution to policies, the optimization problem (4.12) can be re-written as:

$$\max_{\pi} \ \mathbb{E}\left[\sum_{t\in\overline{\mathbb{T}}} L_t(\mathbf{S}_t, \mathbf{Q}_t, \mathbf{O}_{t+1}, \mathbf{W}_{t+1}) + K(\mathbf{S}_T)\right] \tag{4.14a}$$

S.t.

$$\begin{aligned} \mathbf{S}_{t+1} &= f_t(\mathbf{S}_t, \mathbf{Q}_t, \mathbf{O}_{t+1}) && (4.14b)\\ G\ \mathbf{S}_t + H\ \mathbf{Q}_t + I\ \mathbf{O}_{t+1} &\leq b_t && (4.14c)\\ \mathbf{Q}_t &= \pi^q(\mathbf{S}_t) && (4.14d)\\ \mathbf{O}_{t+1} &= \pi^o(\mathbf{S}_t, \mathbf{W}_{t+1}, \mathbf{Q}_t) && (4.14e) \end{aligned}$$

Notice that (4.14d) and (4.14e) let us in a not necessarily linear program. Then, SDP approach is justified as we are going to show.

4.3 Resolution by Stochastic Dynamic Programming

The Stochastic Dynamic Programming method is an algorithm used to find an optimal solution of (4.14) through the Bellman equation. Here, we introduce the concept of *Value Function* and its usability to get optimal policies.

4.3.1 Additive stochastic dynamic programming equation

The optimization problem (4.14) can be solved using dynamic programming method if we introduce (4.15) the *Value Functions* $(V_t)_{t\in\overline{\mathbb{T}}}$ as follows:

$$\begin{aligned} V_T(s) &= K(s), && (4.15a)\\ \hat{V}_t(s,q,w) &= \max_{o_{t+1}\in\mathbb{O}^{ad}} L_t(s, q_t, o_{t+1}, w) + V_{t+1}\circ f_t(s, q_t, o_{t+1}), && (4.15b)\\ V_t(s) &= \max_{q_t\in\mathbb{Q}^{ad}} \mathbb{E}\left(\hat{V}_t(s, q, \mathbf{W}_{t+1})\right) && (4.15c) \end{aligned}$$

The solution spaces $\mathbb{Q}^{ad}$ and $\mathbb{O}^{ad}$ are defined as follows:

$$\mathbb{O}^{ad}(s,q,w) = \begin{cases} 0 \leq o_{t+1} \leq w \\ o_{t+1} \leq s + q_t \end{cases} \tag{4.16a}$$

$$\mathbb{Q}^{ad}(s) = \begin{cases} s + q_t \leq S^\sharp \\ 0 \leq q_t \leq Q^\sharp \\ 0 \leq s \end{cases} \tag{4.16b}$$

Optimal policies Based on (4.15) and (4.16), the optimal policies at interval $[t, t+1[$ are got as:

- For sales: $\pi^o(s,w,q) = $ argmax of (4.15b)
- For orders: $\pi^q(s) = $ argmax of (4.15c)

4.3.2 Sketch of the algorithm

We begin using the general structure of the SDP algorithm with a backward path, which for each possible state we compute the Bellman function defined in (4.15)

For computational experiments in §4.4, the set $\mathbb{W}$ is split in a finite number of elements (n) between the limits: inferior ($W^\flat$) and superior ($W^\sharp$). In that case, the set's cardinality is equal to n.

Based on the partitioning of the demand set $\mathbb{W}$ mentioned above, the Value function (4.15) can be written as:

$$\begin{cases} V_T(s) &= K(s), \\ V_t(s) &= \max_{q_t \in \mathbb{Q}^{ad}} \left[\sum_{w \in \mathbb{W}} \xi_w \left(\hat{V}_t(s,q,w) \right) \right] \end{cases} \tag{4.17}$$

Computing the Value Function (4.15)

The maximization problem (4.17) is solved using SDP algorithm performed as in Algorithm 1

input : Discretized number of states and time span. An initial stock, price and costs, A finite number of realizations of $\mathbf{W}$ into $\mathbb{W}$
output: The Value Function $V_t(s)$ at each period and each state

begin
  for *time* $t = T$ **to** t_0 **step** -1 **do**
    for *states* $s \in \mathbb{S}$ **do**
      for *controls* $q \in \mathbb{Q}$ **do**
        for *uncertain* $w \in \mathbb{W}$ **do**
          $\hat{V}_t(s,q,w) = \max_{o_w} \left[L_t(s,q,o_w,w) + V_{t+1} \circ f_t(s,q,o_w)\right]$
        end
        $\hat{\hat{V}}_t(s,q) = \sum_{w \in \mathbb{W}} \xi_w \left(\hat{V}_t(s,q,w_k)\right)$
      end
      $V_t(s) = \max_q \hat{\hat{V}}_t(s,q)$
    end
  end
end

Algorithm 1: Stochastic Dynamic Programming to SRLP

Obtaining the optimal policies

During the execution of SDP algorithm the optimal policies are taken as follows: at time t and state s

$$\pi^o(s,w,q) = \operatorname*{argmax}_{o_w} \left[L_t(s,q,o_w,w) + V_{t+1} \circ f_t(s,q,o_w)\right] \tag{4.18a}$$

and

$$\pi^q(s) = \operatorname*{argmax}_q \left[\hat{\hat{V}}_t(s,q)\right] \tag{4.18b}$$

Hence, having the Value Function $V_t(s)$ we can use it to obtain the best decisions (policies) about q_t and o_{t+1} at each time period over the planing horizon.

Simply, at the time period t we must solve (4.17) and take the corresponding argmax over q_t and o_{t+1}, even if some modifications have been done over the solution spaces $\mathbb{O}^{ad}$ and $\mathbb{Q}^{ad}$, e.g. It is possible that at any period $[t, t+1[$ the probability distribution of the demand change, we could replace $\xi_w, \forall\ w \in \mathbb{W}$ by its new values $\xi'_w, \forall\ w \in \mathbb{W}$ and solve (4.17) again, getting the corresponding argmax over q_t and o_{t+1}.

4.4 Computational experiments

After building an appropriate algorithm for the SDP implementation we create some instances that allow us to conclude about this optimization problem.

We begin by looking at the behavior of computational times regarding the number of loops and its sizes. If we look the Algorithm 1, the number of cycles needed to obtain the Value Function would be $|\mathbb{T}| \times |\mathbb{S}| \times |\mathbb{Q}| \times |\mathbb{W}|$. However, this is not the real number of cycles. Notice that the stock at time $t+1$ depends of stock at time t and also, it is limited by retailer's capacity. Thus, the number of cycles over order size also will depend of those data varying through the algorithm execution. We leave an example of this situation below.

E.g. at time period t, we suppose that the loop over states says that the stock is 100 units of product, and the control loop is saying that the order size is 90 units. When we go to the uncertainty's loop, the first run there says that demand is equal to 20 units. Now, suppose a retailer's capacity equal to 100. Here, the selling point can not sell more than 20 units of product, letting as minimum a future stock, at time $t+1$, a total of 170 units, which is impossible physically. So we wish to limit the size order respecting the previous situations.

We redesign our algorithm to enforce the capacity constraint using a simple trick. When, we go to the loop control, the algorithm looks as follows:

```
1 begin
2     ...
3     for controls q = 0 : [Q♯ − [Current stock]] do
4         ...
5     end
6     ...
7 end
```

Algorithm 2: Stochastic Dynamic Programming to RLP modified

The line 3 of the algorithm 2 ensures that we only cover the order sizes to bring the stock from current until its maximum (retailer's capacity). The ellipsis mean that the algorithm remains as was presented in Algorithm 1.

4.4.1 Numerical data

We have generated random instances in two different fronts. For the first we want to analyze the computational times to solve it. Secondly, we do a sensitive analysis varying the stockout cost, in that way we can conclude about an important *Key Performance Indicator* in supply chain as the service level.

In both cases we use the following parameters, expressed as a percentage with respect to the unitary product cost (c_t^p), given that the analysis is made on percentage terms too:

- Unitary selling price p_t is 200% of c_t^p
- Unitary transportation cost $\boldsymbol{c}_t^d$ is 50%
- Unitary stockout cost $\boldsymbol{c}_t^{so}$ is also 50%
- Unitary inventory holding cost $\boldsymbol{c}^h$ is 20%

First instances: we proceed changing the time span, the discrete number of states ($|\mathbb{S}|$), order sizes ($|\mathbb{Q}|$), amount of demand ($|\mathbb{W}|$). Hereafter, Table 4.1 shows the respective instances:

Instance	$\|\mathbb{T}\|$	$\|\mathbb{S}\|$	$\|\mathbb{Q}\|$
1	12	11	9
2	12	21	17
3	12	51	41
4	24	11	9
5	24	21	17
6	24	51	41
7	36	11	9
8	36	21	17
9	36	51	41

Table 4.1: Instances proposed to study Single RLP

We also work with larger instances, Table 4.2, to show the ”curse of dimensionality” [2] [31]

Instance	$\|\mathbb{T}\|$	$\|\mathbb{S}\|$	$\|\mathbb{Q}\|$
10	72	11	9
11	72	21	17
12	72	51	41
13	72	101	81

Table 4.2: Larger instances for Single RLP study

[2] *Curse of dimensionality:* refers to the exponential grow regarding the high-dimensional spaces, algorithms as Dynamic Programming have a poor performance in those spaces

We should highlight that the last state and size order represent $S^\sharp$ and $Q^\sharp$ respectively, for all proposed instances.

Second instances: Here, we work with only one size of instances, that is $|\mathbb{T}| = 72$, $|\mathbb{S}| = 51$, $|\mathbb{Q}| = 51$, $|\mathbb{W}| = 41$, and we change the stockout cost as a percentage of unitary product cost as follows:

$$c_t^{so} = \{\underbrace{0; 0,05; 0,1; 0,15; 0,2; 0,25; 0,3; 0,35; 0,4; 0,45; 0,5}_{\text{Each value generates a scenario}}\} \times c_t^p \tag{4.19}$$

Thus, we have 11 instances to analyze the service level at the selling point.

4.4.2 Results analysis

Based on the instances presented we have addressed the analysis in two fronts: computational times and practical considerations.

Computational times. We work with different size instances, recalling the machine characteristics presented in §3.3.2. To understand the increasing computation complexity, we calculate the number of loops L needed to solve a particular instance:

$$L = |\mathbb{T}| \times \frac{|\mathbb{S}| \times (|\mathbb{S}| + 1)}{2} \times |\mathbb{W}| \tag{4.20}$$

In our case $|\mathbb{S}| = |\mathbb{Q}|$, such that we can use any of them in calculation (4.20). In cases where $|\mathbb{S}| \neq |\mathbb{Q}|$, the above expressions must be recalculated.

Table 4.3 presents the summary of computational times for whole instances defined in §4.4.1, we present both case which the policies over $\mathbf{Q}(\cdot)$ and $\mathbf{O}(\cdot)$ are saved and not saved.

		Computational times (seconds)	
Instance	Loops needed	Saving policies	Not saving policies
1	6,534	30	17
2	43,197	246	105
3	598,026	10,670	1,770
4	13,662	63	39
5	90,321	577	263
6	1,250,418	38,885	4,616
7	20,790	102	59
8	137,445	992	397
9	1,902,810	insufficient memory over 64,800	8,985
10	42,174	222	108
11	278,817	2,864	816
12	3,859,986	insufficient memory over 64,800	20,603
13	29,623,401	insufficient memory over 64,800	406,500

Table 4.3: Computational times for instances presented in §4.4.1

Maximum sales policy. Intuitively, we might think that the best solution for the problem can come deciding to sell the maximum quantity of product at each time period as we can. This assumption means:

$$o_{t+1} = \min \{s_t + q_t, w_{t+1}\}, \tag{4.21}$$

we developed another version of the algorithm using this decision rule, this can be seen in Algorithm 3

Here, we only have one maximization operation and that decreases the computational times regarding the Algorithm 1, mainly that we do not have to keep on computer RAM a huge quantity of data as in the case of sales policy (hypermatrices). Table 4.4 presents the computational times using the maximum sales policy.

We highlight that, using optimal policies and maximum sales policy we get exactly the same results for Value Function $V_t(s)$ and policies over orders $\pi^o(s, q, w)$ and sales $\pi^q(s)$.

```
input  : Discretized number of states and time span. An initial stock, price
         and costs, A finite number of realizations of W into 𝕎
output: The Value Function V_t(s) at each period and each state
begin
    for time t = T to t_0 step −1 do
        for controls q ∈ ℚ do
            for states s ∈ 𝕊 do
                for uncertain w ∈ 𝕎 do
                    o = min {s + q, w}
                    V̂_t(q, s) = Σ_{w∈𝕎} ξ_w (L_t(s, q, o_w, w) + V_{t+1} ∘ f_t(s, q, o_w))
                end
            end
        end
        V_t(s) = max_q V̂_t(q, s)
    end
end
```

Algorithm 3: Stochastic Dynamic Programming to SRLP, maximum sales as policy

Instance	Loops needed	Computational times (seconds)
1	$6,534$	0.3
2	$43,197$	2.4
3	$598,026$	29.4
4	$13,662$	0.9
5	$90,321$	4.3
6	$1,250,418$	60.4
7	$20,790$	1.2
8	$137,445$	6.5
9	$1,902,810$	92
10	$42,174$	1.9
11	$278,817$	13.8
12	$3,859,986$	197
13	$29,623,401$	1.472

Table 4.4: Computational times for instances presented in §4.4.1 with max sales policy implementation

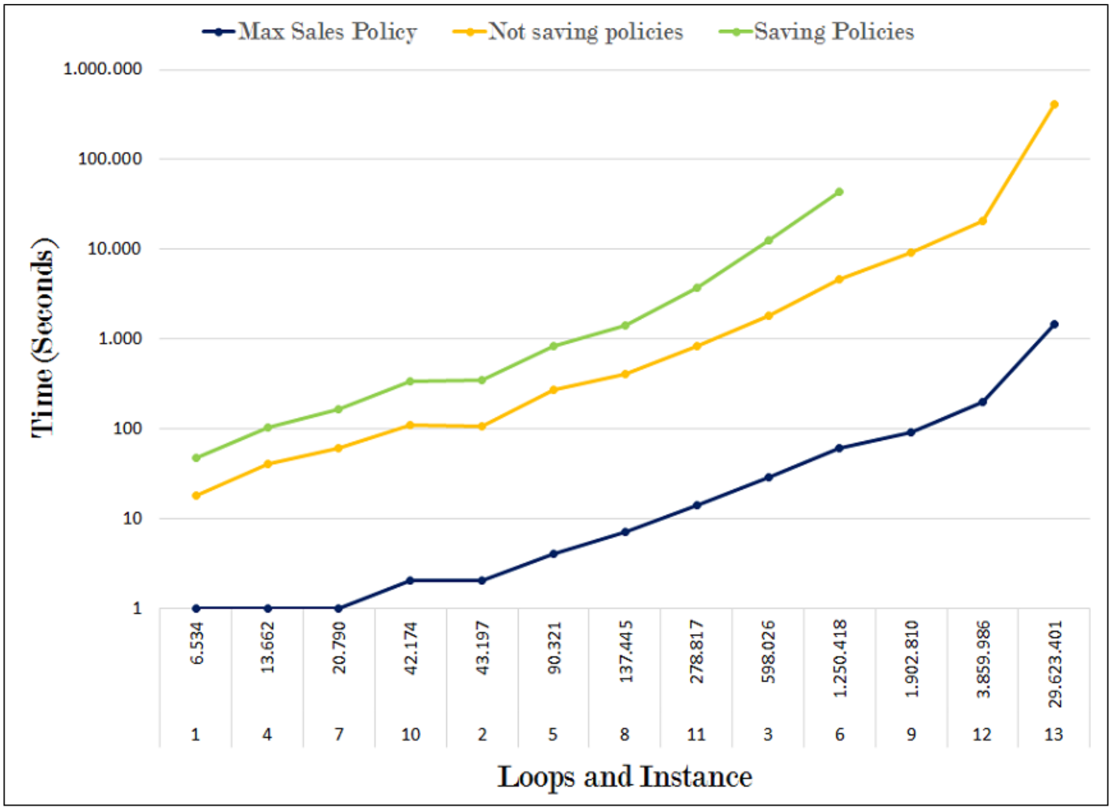

Figure 4.1: Computational times for SRLP

To complete analysis of computational times, the Figure 4.1 shows the relation between loops and time to find Value Functions for each instance. Also, we can note the behavior in logarithmic scale for the optimal computation (saving and not saving policies) and the policy of max sales at each time period.

Stockout and service level. We select a particular instance to analyze the effect of stockout cost over the service level; the details were presented in §4.4.1. The instance describes a selling point where the stock $\mathbf{S}_t$ can be into $[0, 50]$ and same case for orders $\mathbf{Q}_t \in [0, 50]$; on the other hand we consider the demand $\mathbf{W}_t \in [10, 50]$.

We carry out a Monte Carlo Simulation after to obtain the optimal Value Function. Then, the figures 4.2 and 4.3 show the averages of stockout by period and total service level. We can notice when the stockout cost increase, the quantity of lost sales is reduced, given that the order size for the policy also increase. That is logical result, because if you have a higher stockout cost, you will try to avoid it.

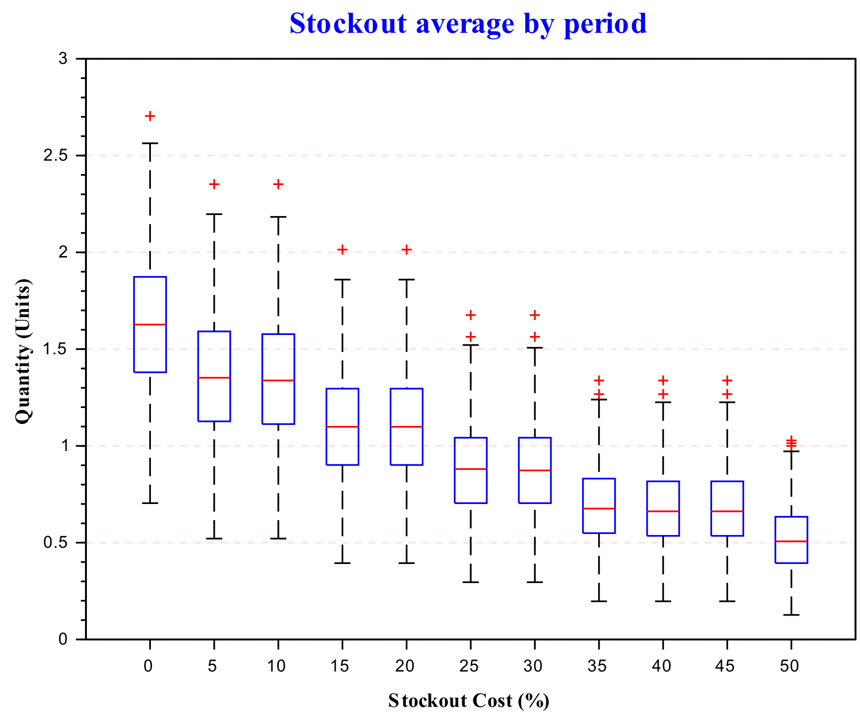

Figure 4.2: Stockout average for SRLP varying sotckout costs

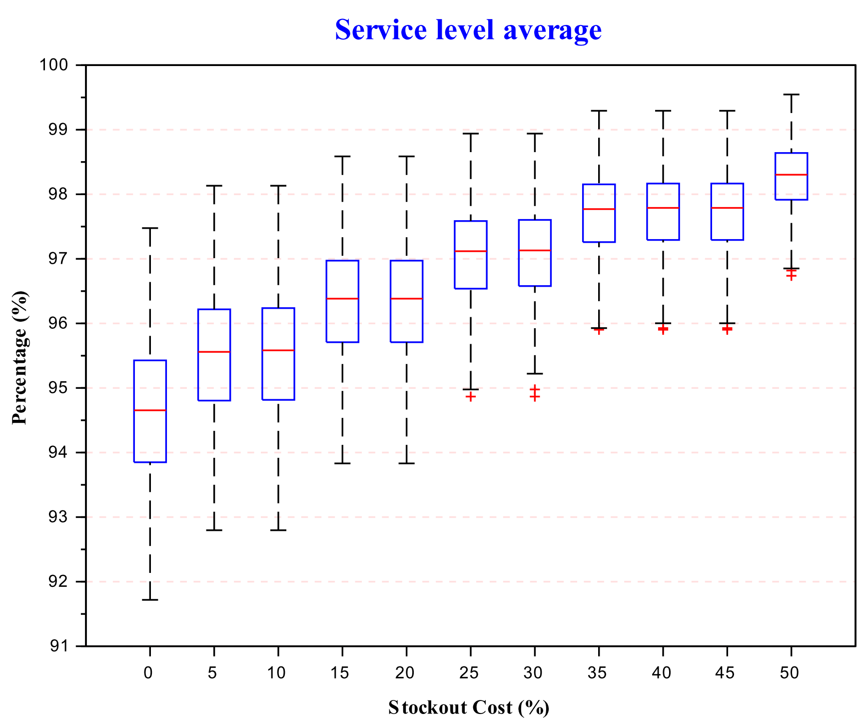

Figure 4.3: Service level comparison for SRLP varying stockout costs

Given that in some cases we can not find significant differences in service levels when the increments of stockout cost are made it, we have developed a statistical test to analyze the situation.

Taking into account that the same scenarios for Monte Carlo simulation were used for each stockout percentage cost, we develop a t-paired test with a significance level of 5% to compare (35% $\mu_{so} = 0.6923$ $\sigma = 0.2066$) and (40% $\mu_{so} = 0.6876$ $\sigma = 0.2055$) cases. Under those conditions there is no significant evidence to reject the null hypothesis of parity between means, even taking a higher significance level.

On the other hand, we have Figure 4.3 presenting the results about service level, when increasing stockout costs the lost sales amount decrease, which is equivalent to increase the service. Under conditions presented for the instance in §4.4.1 and varying the stockout cost between 0% and 50%, the service level is between $94,6\%$ and $98,2\%$. As in computational times, the simulation did it here give us exactly the same result with optimal and maximum sales policies.

4.5 Final Considerations

Regarding computational times we can note how the resolution method suffer the course of dimensionality even for SRLP, that lead us in trouble to tackle the N-retailers problem as we will see in chapter 5.

The maximum sales policy shows an excellent performance for cases where prices and costs hold constant over the time horizon. Nonetheless, it would be interesting to analyze its results in cases whit for example non-linear prices and costs, or even whenever those parameters are random.

Finally, we find very interesting results in practical implementations of the algorithm, since the averages of service level and lost sales can be established according the sotckout cost, keeping the remaining parameters constant. This model is easy implementable for a single selling point in order to manage its inventory policy, it is highly useful in order to replace rules as Economic Order Quantity (EOQ), Minimum Order Quantity (MOQ) or others, given that our model is versatile to tackle not identically distributed demands between time periods.

Chapter 5

N-retailers and RLP Stochastic Programs

In this chapter, we present the statement of problems that involve both more than one selling point and facility location decisions. Furthermore, we introduce a brief proposition of a mixed algorithm based upon both of Stochastic Dual Dynamic Programming and Progressive Hedging.

5.1 N-retailers stochastic program

In chapter 4 we worked with a RLP where only one selling point was taken into account. Here we present a problem with several selling points managed by the same company. The optimization problem ingredients and mathematical model are introduced below.

5.1.1 Optimization problem ingredients

In general, we hold a similar structure as in chapter 4, we will remind general sections of SRLP in order to highlight the main considerations when we talk about N-retailers Location Problem (NRLP from now on).

Parameters: These can be seen in §2.1.2

Information's Structure: We hold the structure presented in §4.1.2.

Variables: We take aspects defined in §2.2.1 and other shown in §4.1.3. In the SRLP we used variables without i index, but for NRLP we have to include it, recalling that each element of the set $\mathbb{I}$ represents one selling point. On the other hand, we also consider the general problem statement using transshipments variable.

- $\mathbf{M}_{ijt}$ as the quantity of product delivered from the retailer i to retailer j at time t, decided once we have received the order $\mathbf{Q}_{it}$ (**beginning of period** $[t, t+1[$)

Given that we can have control over those quantities of transsshipments, these are included in control variables.

State variables remain as in SRLP. However, the states' space grow up exponentially. We can explain the phenomenon considering a couple of selling points, in that case one state is defined by an ordered pair of numbers, each one representing the stock at one facility.

I.e. if we have 10 possible states for each retailer, the states' space would be:

$$s = \left\{ \{s^0, s^0\}, \{s^0, s^1\}, \dots, \{s^0, s^9\}, \{s^1, s^0\}, \{s^1, s^1\}, \dots, \{s^9, s^9\} \right\}$$

.

Where $s^0 = 0$ and $s^9 = S^\sharp$. In that way, the total number of possible states for NRLP is

$$|\mathbb{S}| = \prod_{i \in \mathbb{I}} |\mathbb{S}_i| \,,$$

wherein $|\mathbb{S}_i|$ represents the number of possible states at each retailer i.

Criterion: we use the mathematical expression presented in (2.6) removing the strategic decisions handled by variable $\mathbf{Y}_i$ and including the expected value considering the stochasticity on the demand

$$\mathbb{E}\left[\sum_{i \in \mathbb{I}} \sum_{t \in \overline{\mathbb{T}}} L_{it}(\mathbf{S}_{it}, \mathbf{Q}_{it}, \mathbf{O}_{i,t+1}, \mathbf{M}_{ijt}, \mathbf{W}_{i,t+1}) + K_i(\mathbf{S}_{iT}) \right], \tag{5.1}$$

we recall §4.1.4 that K is the final profit, obtained according to the stock at the final period of the horizon, and L_t is called the instantaneous profit:

$$\begin{aligned} K_i(\mathbf{S}_{iT}) &= \text{Final profit, facility } i \text{ period time } T, \\ L_{it}(\mathbf{S}_{it}, \mathbf{Q}_{it}, \mathbf{O}_{i,t+1}, \mathbf{M}_{ijt}, \mathbf{W}_{t+1}) &= \underbrace{\begin{aligned} &(p_t - c_t^p)\mathbf{O}_{i,t+1} - c_t^{so}(\mathbf{W}_{i,t+1} - \mathbf{O}_{i,t+1}) - \\ &\qquad c_t^d \mathbf{Q}_{it} - c_t^h \mathbf{S}_{it} - \sum_{j \in \mathbb{I}} c_{ijt}^e \mathbf{M}_{ijt} \end{aligned}}_{\text{Instantaneous profit}} \end{aligned} \tag{5.2}$$

Uncertain demand: we defined the stochastic process of demand in 4.1.5, the same concepts are applied in NRLP adding that we work with a set of stochastic process, one by each selling point i

$$\mathbf{W}_i(\cdot) := (\mathbf{W}_{i,t_0+1}, \mathbf{W}_{i,t_0+2}, \ldots, \mathbf{W}_{i,T}), \quad \forall i \in \mathbb{I} \tag{5.3}$$

Also, we need to clarify the σ-field generated by stochastic processes of the demand

$$\mathbf{W}_t = (\mathbf{W}_{1,t}, \mathbf{W}_{2,t}, \ldots, \mathbf{W}_{I,t}) \tag{5.4}$$

Then, using (5.4), notation introduced in 4.5 can be hold, keeping on mind that for this case at each time t, the demand $\mathbf{W}_t$ contains every random variable associated with locations $i \in \mathbb{I}$.

Further on, we propose a mixed SDDP algorithm, which we hold the assumptions of **white noise** in §4.1.5.

Dynamics: The NRLP dynamics go through some modifications generated by the transshipments variables

$$\mathbf{S}_{i,t+1} = f_t(\mathbf{S}_{it}, \mathbf{Q}_{it}, \mathbf{O}_{i,t+1}, \mathbf{M}_{ijt}) \quad \forall\, t \in \mathbb{T}, \tag{5.5}$$

or explicitly

$$\mathbf{S}_{i,t_0} \;=\; S_{i,in} \tag{5.6a}$$

$$\mathbf{S}_{i,t+1} \;=\; \mathbf{S}_{it} + \mathbf{Q}_{it} - \mathbf{O}_{i,t+1} - \sum_{j\in\mathbb{I}} [\mathbf{M}_{ijt} - \mathbf{M}_{jit}],\ \forall\, i \in \mathbb{I},\ t \in \mathbb{T} \tag{5.6b}$$

Constraints: Transshipments also have an effect here, shown as follows

$$\mathbf{O}_{i,t+1} \;\le\; \mathbf{W}_{i,t+1}, \qquad \forall\, i \in \mathbb{I},\ t \in \overline{\mathbb{T}} \tag{5.7a}$$

$$\mathbf{O}_{i,t+1} \;\le\; \mathbf{S}_{it} + \mathbf{Q}_{it} - \sum_{j\in\mathbb{I}} [\mathbf{M}_{ijt} - \mathbf{M}_{jit}], \quad \forall\, i \in \mathbb{I},\ t \in \overline{\mathbb{T}} \tag{5.7b}$$

$$-S_i^\sharp \;\le\; -\mathbf{S}_{it} - \mathbf{Q}_{it} + \sum_{j\in\mathbb{I}} [\mathbf{M}_{ijt} - \mathbf{M}_{jit}],\ \forall\, i \in \mathbb{I},\ t \in \overline{\mathbb{T}} \tag{5.7c}$$

$$0 \le \sum_{j\in\mathbb{I}} \mathbf{M}_{ijt} \;\le\; \mathbf{S}_{it} + \mathbf{Q}_{it}, \qquad \forall\, i \in \mathbb{I},\ t \in \overline{\mathbb{T}} \tag{5.7d}$$

$$0 \le \mathbf{Q}_{it} \;\le\; Q^\sharp, \qquad \forall\, i \in \mathbb{I},\ t \in \overline{\mathbb{T}} \tag{5.7e}$$

$$0 \;\le\; \mathbf{O}_{i,t+1}, \qquad \forall\, i \in \mathbb{I},\ t \in \overline{\mathbb{T}} \tag{5.7f}$$

$$0 \;\le\; \mathbf{S}_{it}, \qquad \forall\, i \in \mathbb{I},\ t \in \overline{\mathbb{T}} \tag{5.7g}$$

$$0 \;\le\; \mathbf{M}_{ijt}, \qquad \forall\, i, j \in \mathbb{I},\ t \in \overline{\mathbb{T}} \tag{5.7h}$$

Ultimately, each constraint holds the same sense that in SRLP; we only add index i and transshipments considerations in equations which require that. We only add (5.7d) to ensure that transshipments do not exceed quantity of product and (5.7h) to declare transshipments as positive variable.

5.1.2 Optimization problem formulation

In this section we take all elements defined in §5.1.1 until to present the *Value Function* to contextualize SDP and SDDP implementations.

Mathematical modeling: Gathering elements presented up to this point we can formulate NRLP as follows

$$\max_{\mathbf{S},\mathbf{Q},\mathbf{O},\mathbf{M}} \mathbb{E}\left(\sum_{i\in\mathbb{I}}\sum_{t\in\overline{\mathbb{T}}} L_{it}(\mathbf{S}_{it},\mathbf{Q}_{it},\mathbf{O}_{i,t+1},\mathbf{M}_{ijt},\mathbf{W}_{i,t+1}) + K(\mathbf{S}_{iT})\right), \tag{5.8a}$$

Subject to:

(5.6), (5.7) and

$$\sigma(\mathbf{Q}_{it}) \subset \mathcal{F}_t, \quad \forall\, i\in\mathbb{I},\ t\in\mathbb{T} \tag{5.8b}$$
$$\sigma(\mathbf{O}_{it}) \subset \mathcal{F}_t, \quad \forall\, i\in\mathbb{I},\ t\in\mathbb{T} \tag{5.8c}$$
$$\sigma(\mathbf{M}_{ijt}) \subset \mathcal{F}_t, \quad \forall\, i,j\in\mathbb{I},\ t\in\mathbb{T} \tag{5.8d}$$

Value Function: The optimization problem (5.8) is solved using the concept of Bellman function as was presented in SRLP

$$V_T(s) = \sum_{i\in\mathbb{I}} K_i(s_i), \tag{5.9a}$$

$$\hat{V}_t(s,q,m,w) = \max_{o_{i,t+1}\in\mathbb{O}^{ad}} \sum_{i\in\mathbb{I}} \left[L_{it}(s_{it},q_{it},o_{i,t+1},m_{ijt},w_{i,t+1}) + V_{t+1}\circ f_t(s_{it},q_{it},o_{i,t+1},m_{ijt})\right], \tag{5.9b}$$

$$V_t(s) = \max_{q_t,m_t\in\mathbb{Q}^{ad}} \mathbb{E}\left(\hat{V}_t(s,q,m,\mathbf{W}_{t+1})\right) \tag{5.9c}$$

The solution spaces $\mathbb{Q}^{ad}$ and $\mathbb{O}^{ad}$ are defined as follows:

$$\mathbb{O}^{ad}(s,q,m,w) = \begin{cases} 0 \le o_{i,t+1} \le w_i, \quad \forall\ i \in \mathbb{I} \\ o_{i,t+1} \le s_{it} + q_{it} - \sum_{j \in \mathbb{I}} \left[m_{ijt} - m_{jit} \right], \forall\ i \in \mathbb{I} \end{cases} \tag{5.10a}$$

$$\mathbb{Q}^{ad}(s) = \begin{cases} s_{it} + q_{it} - \sum_{j \in \mathbb{I}} \left[m_{ijt} - m_{jit} \right] \le S^{\sharp},\ \forall\ i \in \mathbb{I} \\ 0 \le \sum_{j \in \mathbb{I}} m_{ijt} \le s_{it} + q_{it},\ \forall\ i \in \mathbb{I} \\ 0 \le q_{it} \le Q^{\sharp}, \quad \forall\ i \in \mathbb{I} \\ 0 \le s_{it}, \quad \forall\ i \in \mathbb{I} \\ 0 \le m_{ijt}, \quad \forall\ i, j \in \mathbb{I} \end{cases} \tag{5.10b}$$

5.2 Retail Location Problem

At this point, we can use all definitions of SRLP and NRLP for the final multi-stage stochastic program to Retail Location Problem. We are going to introduce a stochastic location variable, as the possibility of opening a new selling point according to stochastic demands, and then the corresponding optimization problem.

Holding the structure: The NRLP in §5.1 has the scheme necessary to propose the RLP formulation. We hold parameters, information's structure, uncertainty and dynamics as before, and the remaining concepts suffer small modifications.

Variables: We recall the binary variable $\mathbf{Y}_i$ as in §2.2.1. $\mathbf{Y}_i$ or location decision can be classified as control variable. Nonetheless, we have to be careful considering that is made only once **at the beginning** of the time span t_0. Consequently, location variables are not indexed by time t.

The remain variables, state(stocks) and controls (orders, transshipments, sales) are handled in the same way as NRLP.

Criterion: We modify definition in §2.3.2 only adding the expected value attached to stochastic programs. Keeping NRLP notation, we present:

$$\mathbb{E}\left[\sum_{i\in\mathbb{I}}\left(J(\mathbf{Y}_i)+\sum_{t\in\overline{\mathbb{T}}}L_{it}(\mathbf{S}_{it},\mathbf{Q}_{it},\mathbf{O}_{i,t+1},\mathbf{M}_{ijt},\mathbf{W}_{i,t+1})+K_i(\mathbf{S}_{iT})\right)\right], \tag{5.11}$$

Each term in (5.11) is:

$$\left\{\begin{array}{ll} K_i(\mathbf{S}_{iT}) & = \text{Final profit, facility } i \text{ period time } T, \\ L_{it}(\mathbf{S}_{it},\mathbf{Q}_{it},\mathbf{O}_{i,t+1},\mathbf{M}_{ijt},\mathbf{W}_{t+1}) & = \begin{array}{r}(p_t-c_t^p)\mathbf{O}_{i,t+1}-c_t^{so}(\mathbf{W}_{i,t+1}-\mathbf{O}_{i,t+1})- \\ c_t^d\mathbf{Q}_{it}-c_t^h\mathbf{S}_{it}-\sum_{j\in\mathbb{I}}c_{ijt}^e\mathbf{M}_{ijt},\end{array} \\ J(\mathbf{Y}_i) & = -\,c_i^f\mathbf{Y}_i \end{array}\right. \tag{5.12}$$

Constraints: Some modification are needed here

$$
\begin{aligned}
\mathbf{O}_{i,t+1} &\leq \mathbf{Y}_i\mathbf{W}_{i,t+1}, & \forall\, i \in \mathbb{I},\ t \in \overline{\mathbb{T}} \quad &(5.13a)\\
\mathbf{O}_{i,t+1} &\leq \mathbf{S}_{it} + \mathbf{Q}_{it} - \sum_{j\in\mathbb{I}}[\mathbf{M}_{ijt} - \mathbf{M}_{jit}], & \forall\, i \in \mathbb{I},\ t \in \overline{\mathbb{T}} \quad &(5.13b)\\
-\mathbf{Y}_i S_i^\sharp &\leq -\mathbf{S}_{it} - \mathbf{Q}_{it} + \sum_{j\in\mathbb{I}}[\mathbf{M}_{ijt} - \mathbf{M}_{jit}], & \forall\, i \in \mathbb{I},\ t \in \overline{\mathbb{T}} \quad &(5.13c)\\
0 \leq \sum_{j\in\mathbb{I}} \mathbf{M}_{ijt} &\leq \mathbf{S}_{it} + \mathbf{Q}_{it}, & \forall\, i \in \mathbb{I},\ t \in \overline{\mathbb{T}} \quad &(5.13d)\\
\mathbf{M}_{ijt} &\leq \mathbf{Y}_i S_i^\sharp, & \forall\, i \in \mathbb{I},\ t \in \overline{\mathbb{T}} \quad &(5.13e)\\
0 \leq \mathbf{Q}_{it} &\leq \mathbf{Y}_i Q^\sharp, & \forall\, i \in \mathbb{I},\ t \in \overline{\mathbb{T}} \quad &(5.13f)\\
0 &\leq \mathbf{O}_{i,t+1}, & \forall\, i \in \mathbb{I},\ t \in \overline{\mathbb{T}} \quad &(5.13g)\\
0 &\leq \mathbf{S}_{it}, & \forall\, i \in \mathbb{I},\ t \in \overline{\mathbb{T}} \quad &(5.13h)\\
0 &\leq \mathbf{M}_{ijt}, & \forall\, i,j \in \mathbb{I},\ t \in \overline{\mathbb{T}} \quad &(5.13i)
\end{aligned}
$$

Basically, these are the same constraints as in SRLP. We only modify (5.13a), (5.13c), (5.13f) adding $\mathbf{Y}_i$ variable to take into account if the selling point is open or close. Furthermore, we add (5.13e) to forbid transshipments if the retailer is closed.

Optimization problem: The previous considerations lead us

$$
\max_{\mathbf{Y},\mathbf{S},\mathbf{Q},\mathbf{O},\mathbf{M}} \mathbb{E}\left[\sum_{i\in\mathbb{I}} J(\mathbf{Y}_i) + \sum_{t\in\overline{\mathbb{T}}} L_{it}(\mathbf{S}_{it}, \mathbf{Q}_{it}, \mathbf{O}_{i,t+1}, \mathbf{M}_{ijt}, \mathbf{W}_{i,t+1}) + K_i(\mathbf{S}_{iT})\right] \qquad (5.14a)
$$

Subject to:

(5.6), (5.13) and

$$
\begin{aligned}
\sigma(\mathbf{Q}_{it}) &\subset \mathcal{F}_t, \quad \forall\, i \in \mathbb{I},\ t \in \mathbb{T} & (5.14b)\\
\sigma(\mathbf{O}_{it}) &\subset \mathcal{F}_t, \quad \forall\, i \in \mathbb{I},\ t \in \mathbb{T} & (5.14c)\\
\sigma(\mathbf{M}_{ijt}) &\subset \mathcal{F}_t, \quad \forall\, i,j \in \mathbb{I},\ t \in \mathbb{T} & (5.14d)\\
& & (5.14e)
\end{aligned}
$$

5.3 New resolution method

We propose in a mixed resolution method using principles of Stochastic Dual Dynamic Programming [32] and Progressive Hedging (PH) [33] algorithms. This section presents the general framework to implement this kind of algorithm

As in NRLP we can write again the Value Function

$$V_T(s) = \sum_{i\in\mathbb{I}} K_i(s_i), \tag{5.15a}$$

$$V_t(s) = \max_{q_t, m_t \in \mathbb{Q}^{ad}} \mathbb{E}\left(\max_{o_{i,t+1}\in\mathbb{O}^{ad}} \sum_{i\in\mathbb{I}} [L_{it}(s_{it}, q_{it}, o_{i,t+1}, m_{ijt}, w_{it}) + V_{t+1}\circ f_t(s_{it}, q_{it}, o_{i,t+1}, m_{ijt})] \right) \tag{5.15b}$$

In fact, (5.15b) has a two stage stochastic program within itself. Thus, we can induce a SDDP with Progressive Hedging inside it to calculate the Value Function at time t and state s.

5.3.1 Generalization of notation:

Given that we have a set of control variables **at the beginning** of interval $[t, t+1[$ and other **during** and **at the end** of the same interval, we introduce a general notation for this problem and for any other that could be tackle using our proposed method.

Controls at the beginning of the interval: These controls are

$$\mathbf{U}_{it} = \{\mathbf{Q}_{it}, \mathbf{M}_{ijt}\}, \quad \forall\ i,j \in \mathbb{I} \tag{5.16}$$

Controls during or at the end of the interval: The controls are

$$\mathbf{U}^{+}_{i,t+1} = \{\mathbf{O}_{i,t+1}\}, \quad \forall\ i \in \mathbb{I} \tag{5.17}$$

Notice that (5.16) and (5.17) lead us a decision-hazard-decision framework. This is characterized to the decisions are made before to disclosure the uncertain demand (decision-hazard framework). Then, the realization of demand comes and we again make some decision (hazard-decision framework).

Generalized value function: We can generalize the optimization problem with its respective value function using states ($\mathbf{S}_t$), two types of controls ($\mathbf{U}_t$, $\mathbf{U}^{+}_{t+1}$) and stochasticity ($\mathbf{W}_{t+1}$).

In addition, the sets $\mathbb{U}_B^{ad}$ and $\mathbb{U}_A^{ad}$ contain linear constraints associated with special characteristics for a given problem.

$$V_T(s) = K(s), \tag{5.18a}$$

$$V_t(s) = \max_{u_t \in \mathbb{U}_B^{ad}} \mathbb{E}\left(\max_{u_{t+1}^{+} \in \mathbb{U}_A^{ad}} L_t(s_t, u_t, u_{t+1}^{+}, w_{t+1}) + V_{t+1} \circ f_t(s_t, u_t, u_{t+1}^{+}) \right) \tag{5.18b}$$

The equation (5.18b) is in fact a two stage stochastic program that can be written as

$$V_t(s) = \max_{u_t,\ u_{t+1,}^{+w} \in \mathbb{U}^{ad}} \sum_{w \in \mathbb{W}} \xi_w \left[L_t(s_t, u_t, u_{t+1,}^{+w}, w_{t+1}) + V_{t+1} \circ f_t(s_t, u_t, u_{t+1}^{+w}) \right], \tag{5.19}$$

where $\mathbb{U}^{ad} = \mathbb{U}\mathbb{B}^{ad} \cap \mathbb{U}\mathbb{A}^{ad}$. Here, we present and study u_t as reserve variable and u_{t+1}^{+w} as recourse variable.

5.3.2 Method presentation

Now, we are going to present a DOASA implementation (*Dynamic Outer Approximation Sampling Algorithm*) as well SDDP is structured in [34]. But in addition, we plug in a implementation of PH algorithm into the calculation of value function at time t and state s.

Foundations:

- **SDDP:** This method is based on the utilization of duality theory to build cuts, for the convex value function, in a stochastic case when the model is decomposed stage by stage, as was shown by [32] and [35].

- **PH:** We can explain the idea behind o fthis algorithm based on Figure 5.1. Progressive Hedging is also a resolution method for large scale optimization, where a second stage decision depends of the first one. The algorithm allows to make different decisions at first stage rather than only one, and step by step all those first decision are forced to converge the same result. For a detailed explanation [36] can be consulted.

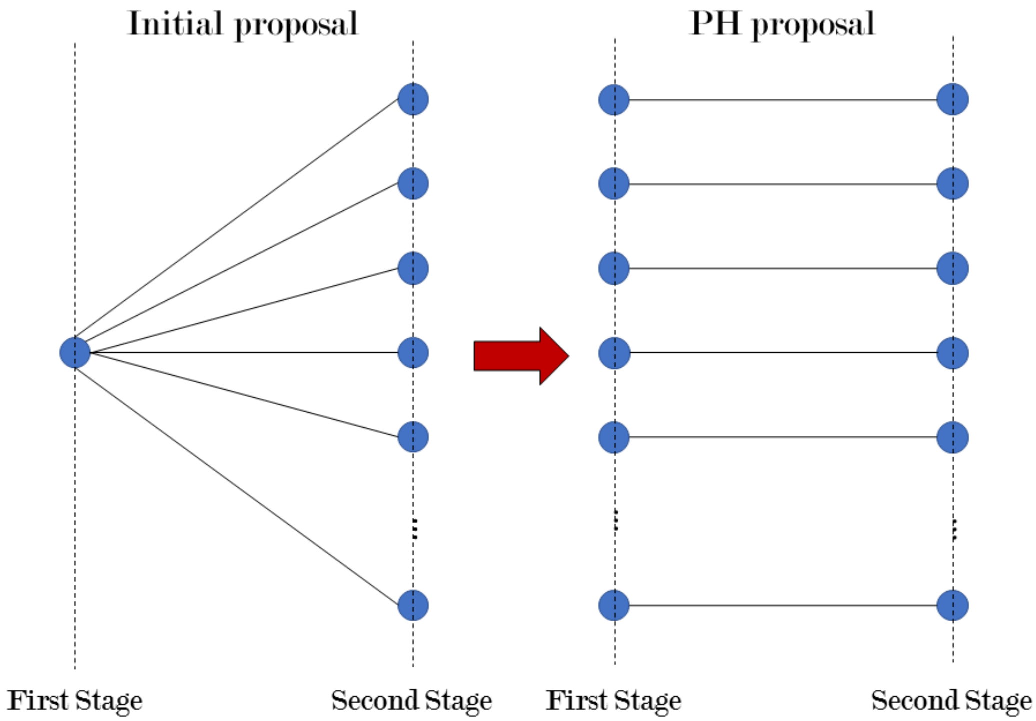

Figure 5.1: Progressive Hedging structure

Stochastic Dual Dynamic Programming framework

Recursively, we build lower approximations $\check{V}_t^{(k)}(s)$ of the value functions $V_t(s)$, defined in equation (5.18), as the supremum of a number of affine functions. Each one of this affine function is called a *cut*. Here, super index k represent the stage of the algorithm; we will give more details about it further on.

Furthermore, we assume the following necessary conditions for the algorithm:

- convexity on functions (5.1) and (5.4), and
- linearity on the constraints (5.7)

Bellman operator: We introduce the following operator to simplify notation and provide a better understanding of the resolution method. For any time t and some function B mapping the set of states and demands into $\mathbb{R}$, we define the stochastic Bellman operator $\mathcal{T}_t$:

$$\mathcal{T}_t(B)(s) = \max_{u_t, u_{t+1}^{+w} \in \mathbb{U}^{ad}} \mathbb{E}_w \left[L_t(s_t, u_t, u_{t+1}^{+w}, w_{t+1}) + B \circ f_t(s_t, u_t, u_{t+1}^{+w}) \right] \tag{5.20}$$

In that way, the value function simply reads:

$$\begin{cases} V_T(s) &= -K(s), \\ V_t(s) &= \mathcal{T}_t(V_{t+1})(s) \end{cases} \tag{5.21}$$

This Bellman operator respect the following properties.

Monotonicity. For any couple of functions $(V, \overline{V})$

$$\forall\ s \in \mathbb{S}, V(s) \leq \overline{V}(s) \Rightarrow \forall\ s \in \mathbb{S}, (\mathcal{T}V)(s) \leq (\mathcal{T}\overline{V})(s). \tag{5.22}$$

Convexity. For any function V, if L_t is jointly convex in (s, u_t, u_{t+1}^{+w}), if V is convex and if f_t is affine, then

$$s \mapsto (\mathcal{T}V)(s) \text{ is convex.} \tag{5.23}$$

Linearity. For any piecewise linear function V, if L_t is jointly convex in $(s, u_t, ^{+w})$ and is piecewise linear function, and if f_t is affine, then

$$s \mapsto (\mathcal{T}V)(s) \text{ is a piecewise linear.} \tag{5.24}$$

Duality theory: Assuming that we are at iteration k of the algorithm and having a lower approximation $\check{V}_{t+1}^{(k)}(s)$ of $V_{t+1}^{(k)}(s)$, we consider the problem

$$\max_{x_t, u_t, u_{t+1}^{+w}} \sum_{w \in \mathbb{W}} \xi_w \left[L_t(s_t, u_t, u_{t+1}^{+w}, w_{t+1}) + V_{t+1}^{(k)} \circ f_t(s_t, u_t, u_{t+1}^{+w}) \right], \tag{5.25a}$$

$$\text{s.t} \quad s = s_t^{(k)}, \tag{5.25b}$$

and $\beta_t^{(k+1)}$ the optimal value of (5.25a), and $\lambda_t^{(k+1)}$ an optimal multiplier of the constraint (5.25b), their corresponding mathematical expressions are:

$$\begin{cases} \beta_t^{(k+1)} &= \mathcal{T}_t(\check{V}_{t+1}^{(k)})(s^{(k)}), \\ \lambda_t^{(k+1)} &\in \partial \mathcal{T}_t(\check{V}_{t+1}^{(k)})(s^{(k)}), \end{cases} \tag{5.26}$$

Then, we can build an affine minorant function of Bellman equation based on monotonicity, convexity and linearity properties already presented

$$\beta_t^{(k+1)} + \langle \lambda_t^{(k+1)}, s - s_t^{(k)} \rangle \leq \mathcal{T}_t(\check{V}_{t+1}^{(k)})(s) \leq \mathcal{T}_t(V_{t+1})(s) = V_t(s), \quad \forall s \in \mathbb{S}. \tag{5.27}$$

This ensures that $s \mapsto \beta_t^{(k+1)} + \langle \lambda_t^{(k+1)}, s - s_t^{(k)} \rangle$ is a cut, i.e an affine function below $V_t(s)$, updating the approximation of value function we obtain:

$$\check{V}_t^{(k+1)} = \max \left\{ \check{V}_t^{(k)}, \beta_t^{(k+1)} + \langle \lambda_t^{(k+1)}, s - s_t^{(k)} \rangle \right\}. \tag{5.28}$$

In addition, we can go to the next time step using f_t function

$$s_{t+1}^{(k)} = f_t \left(s_t^{(k)}, u_t^{(k)}, u_{t+1}^{+w\ (k)} \right). \tag{5.29}$$

Progressive Hedging implementation

Going to equation (5.25) problem. We observe a two stage stochastic program with reserve and recourse variables, which can be decomposed and expressed as

$$\max_{x_t, u_t, u_{t+1}^{+w}} \sum_{w \in \mathbb{W}} \xi_w \left[L_t(s_t, u_t^w, u_{t+1}^{+w}, w_{t+1}) + V_{t+1}^{(k)} \circ f_t(s_t, u_t^w, u_{t+1}^{+w}) \right], \tag{5.30a}$$

$$\text{s.t} \quad s = s_t^{(k)}, \tag{5.30b}$$

$$u_t^w = \sum_{w' \in \mathbb{W}} \xi_{w'} u_t^{w'} \tag{5.30c}$$

This problem can be solved using the Progressive Hedging algorithm as in [35]. Furthermore, (5.30) allows us to keep the general structure of SDDP to get cuts with $\beta_t^{(k+1)}$ the optimal value of the problem, and $\lambda_t^{(k+1)}$ the optimal multiplier of the constraint (5.30b).

SDDP-PH algorithm

At the beginning of step k, we suppose that we have, for each time step t, an approximation of $\check{V}_t^{(k)}$ of $V_t^{(k)}$ satisfying

- $\check{V}_t^{(k)} \leq V_t^{(k)}$,
- $\check{V}_t^{k} = K$,
- $\check{V}_t^{(k)}$ is convex.

input : Discretized states and time span. An initial stock, Linear L_t function, A finite number of realizations of $\mathbf{W}$ into $\mathbb{W}$

output: The approximated Value Function $\check{V}_t(s)$ at each period and each state

begin

  Generate L initial cuts $\check{V}_t(s)$ of $V_t(s)$ (Backward phase)

  for *cuts* $l = 1$ *to* L **do**

    for *time* $t = T$ **to** t_0 **step** -1 **do**

      Solve the problem (5.30) with a PH implementation as 5

$$\beta_t^{(k+1)} = \text{optimal value of (5.30)}$$

$$\lambda_t^{(k+1)} = \text{ associate multiplier of constraint (5.30b)}$$

$$\check{V}_t^{(k+1)}(s) = \max\left\{\check{V}_t^{(k)}(s), \beta_t^{(k+1)} + \langle \lambda_t^{(k+1)}, s - s_t^{(k)} \rangle\right\}$$

    end

  end

  while ***Stopping rule*** *is not reached* **do**

    Forward phase

    for *time* $t = t_0$ **to** T **do**

      Generate a random scenario $w_{t_0}, \ldots, w_{T-1} \in \mathbb{W}$

      Define a trajectory $(s_t^{(k)})_{t=t_0,\ldots,T}$ solving by PH 5

$$\max_{x_t, u_t, u_{t+1}^{+w}} \sum_{w \in \mathbb{W}} \xi_w \left[L_t(s_t, u_t^w, u_{t+1}^{+w}, w_{t+1}) + V_{t+1}^{(k)} \circ f_t(s_t, u_t^w, u_{t+1}^{+w})\right],$$

$$\text{s.t} \quad u_t^w = \sum_{w' \in \mathbb{W}} \xi_{w'} u_t^{w'}$$

      Use $s_{t+1}^{(k)} = f_t\left(s_t^{(k)}, u_t^{(k)}, \overset{+w}{u_{t+1}}{}^{(k)}\right)$ to define the stock trajectory

    end

    Backward phase

    Using trajectory $(s_t^{(k)})_{t=t_0,\ldots,T}$ obtained in forward phase

    for *time* $t = T$ **to** t_0 **step** -1 **do**

      Solve the problem (5.30) with a PH implementation as 5

$$\beta_t^{(k+1)} = \max_{x_t, u_t, u_{t+1}^{+w}} (5.30)$$

$$\lambda_t^{(k+1)} = \text{ associate multiplier of constraint (5.30b)}$$

$$\check{V}_t^{(k+1)}(s) = \max\left\{\check{V}_t^{(k)}(s), \beta_t^{(k+1)} + \langle \lambda_t^{(k+1)}, s - s_t^{(k)} \rangle\right\}$$

    end

    Check the ***stopping rule*** for the algorithm

  end

end

Algorithm 4: Stochastic Dual Dynamic Programming with Progressive Hedging implementation

The Algorithm 4 needs two additional clarifications: The stopping rule of the algorithm and the Progressive Hedging implementation

The problem to must be solved in backward phase and every step of the algorithm is an adequate adaptation of (5.30) as PH suggests

$$\max_{s_t, u_t^w, u_{t+1}^w} \Bigg[L_t(s_t, u_t^w, u_{t+1}^w, w_{t+1}) + V_{t+1}^{(k)} \circ f_t(s_t, u_t^w, u_{t+1}^w) \tag{5.31a}$$

$$- \Lambda_w^{(\underline{k})} \left(u_t^{w(\underline{k}+1)} - \overline{u}_t^{w(\underline{k})} \right) - \frac{r}{2} \left\| u_t^{w(\underline{k}+1)} - \overline{u}_t^{w(\underline{k})} \right\|^2 \Bigg],$$

$$\text{s.t} \qquad s = s_t^{(k)}, \tag{5.31b}$$

input : penalty r defined as in [33], initial multipliers $\left\{ \Lambda_w^{(0)} \right\}_{w \in \mathbb{W}}$, discretized set $\mathbf{W}$ into $\mathbb{W}$

output: Optimal first decision u_t

begin
- **repeat**
  - **for** *uncertainty* $w \in \mathbb{W}$ **do**
    - Solve the deterministic problem (5.31),
    - and obtain optimal decision $u_t^{w(\underline{k}+1)}$;
  - **end**
  - Update the mean of reserve decisions

    $$\overline{u}_t^{w(\underline{k})} = \sum_{w \in \mathbb{W}} \xi_w u_t^{w(k+1)};$$

  - Update the multiplier of the measurability penalization by

    $$\Lambda_w^{(k+1)} = \Lambda_w^{(k)} + r \left(u_t^{w(k+1)} - \overline{u}_t^{w(\underline{k})} \right), \forall\ w \in \mathbb{W};$$

- **until** $u_t^{w(\underline{k}+1)} - \sum_{w' \in \mathbb{W}} \xi_{w'} u^{w(\underline{k}+1)} = 0,\ \forall\ w \in \mathbb{W}$;

end

Algorithm 5: Progressive Hedging implementation into SDDP

Stopping rule of SDDP-PH: Defining upper and lower bound we can get the stopping rule of our algorithm, we use the proposal presented by De Lara et. al [35].

Lower bound, for the approximation as step k is given by $\check{V}_0^{(k)}(s_0)$. On the other, hand, the upper bound is a more complicated to calculate. We do a Monte-Carlo simulation and computing the expected cost $(\hat{m}_n^{(k)})$ and its respective standard deviation e_n, build a confidence interval with $\alpha\%$ significant level $\left[\hat{m}_n^{(k)} - e_n, \hat{m}_n^{(k)} + e_n\right]$.

Finally, for the stopping rule, we choose an a priori error $\epsilon > 0$ and stop the algorithm when the upper bound $\hat{m}_n^{(k)} + e_n$ of the confidence interval, built as in the previous paragraph, is smaller than ϵ above the exact lower bound $\check{V}_0^{(k)}(s_0)$. In that way, we have a chance of $1 - \alpha$ that the approximate policy given by the algorithm yields a value less than ϵ over the optimal value.

5.3.3 Final considerations

The algorithm presented in §5.3 could be useful to tackle the NRLP problem, several instances should be studied to analyze its performance. Nonetheless, we need another kind of method to solve the RLP. Currently, a new research project is being proposed to extend the SDDP-PH 4 algorithm in order to gather and mix it with L-Shaped method, which has been proved to solve mixed integer stochastic programs as can be seen in Birge et. al. [37].

Chapter 6

Conclusions and Future Research

In this report we have outlined a model for the Retail Location Problem. The model which has been studied in different ways shows us features deserving several analysis. First, we have been able to show in what the deterministic problem is not a good representation of real life cases. Since the demand can be anticipated, the results of the model yield a certain degree of bias in the analysis of the final company's profit.

Turning into stochastic programs, we find that RLP is a highly complex model, considering its features as mixed integer program, time span and general information's structure decision. When we could build the value function for the SRLP, we tackle the problem with a special implementation of stochastic dynamic programming for this kind of problem, which includes a two stage stochastic program at each time period $[t, t+1[$ (decision-hazard-decision framework). Nonetheless, the results show a highly computational complexity for this implementation. Intuitively, we developed a policy (maximum sales policy) where the second stage is reduced and consequently, we get value functions with similar or equal profit results regarding the optimal proposed previously. Likewise, we study the behavior of stockout and service level versus established value of stockout cost. It is pretty clear that each real life case will have prices and costs that generate its proper service level, however, our proposed instances shown how we can produce levels between 93% and 98%.

On the other hand, we present the NRLP and RLP stochastic programs. We mentioned before the complexity of these problems. Thus, we are proposing a new resolution method that allows us to overcome the "curse of dimensionality" in [31], a mix between SDDP and Progressive Hedging (SDDP-PH) is posed to study frameworks of decision-hazard-decision structure. Currently, implementations of our proposal in §5.3 are being conducted to measure its performance in high dimension problems.

Furthermore, the RLP which involves strategic decisions to new locations is still an open issue, it could be interesting to create new resolution methods integrating e.g. L-Shaped and SDDP-PH algorithms. Nonetheless, this is no the only way to study that problem, other implementations can be proposed in future research projects using for example approximate dynamic programming.

Finally, we can present other possibles extensions of our research including for example: stochastic costs, prices or lead times. The RLP presented here, only is able to handle open decisions at the beginning of the time span, it would be interesting to propose a new model that considers these decisions at any time period across the whole time span. That model is interesting for business which work with rented facilities.

Bibliography

[1] G. Laporte, S. Nickel, and F. Saldanha da Gama, *Location science*. Springer, Feb. 2015.

[2] L. V. Snyder, "Facility location under uncertainty: a review," *IIE Transactions*, vol. 38, no. 7, pp. 547–564, 2006.

[3] A. Alonso-Ayuso, L. F. Escudero, A. Garìn, M. T. Ortuño, and G. Pérez, "An approach for strategic supply chain planning under uncertainty based on stochastic 0-1 programming," *Journal of Global Optimization*, vol. 26, no. 1, pp. 97–124, 2003.

[4] T. Santoso, S. Ahmed, M. Goetschalckx, and A. Shapiro, "A stochastic programming approach for supply chain network design under uncertainty," *European Journal of Operational Research*, vol. 167, no. 1, pp. 96–115, 2005.

[5] L. V. Snyder, M. S. Daskin, and C.-P. Teo, "The stochastic location model with risk pooling," *European Journal of Operational Research*, vol. 179, no. 3, pp. 1221–1238, 2007.

[6] L. Ozsen, C. R. Coullard, and M. S. Daskin, "Capacitated warehouse location model with risk pooling," *Naval Research Logistics (NRL)*, vol. 55, no. 4, pp. 295–312, 2008.

[7] A. Atamtürk, G. Berenguer, and Z.-J. Shen, "A conic integer programming approach to stochastic joint location-inventory problems," *Operations Research*, vol. 60, no. 2, pp. 366–381, 2012.

[8] J. Fernández, B. Pelegrı, F. Plastria, and B. Tóth, "Solving a huff-like competitive location and design model for profit maximization in the plane," *European Journal of Operational Research*, vol. 179, no. 3, pp. 1274–1287, 2007.

[9] Y. Hinojosa, J. Kalcsics, S. Nickel, J. Puerto, and S. Velten, "Dynamic supply chain design with inventory," *Computers & Operations Research*, vol. 35, no. 2, pp. 373–391, 2008.

[10] A. S. Zadeh, R. Sahraeian, and S. M. Homayouni, "A dynamic multi-commodity inventory and facility location problem in steel supply chain network design," *The International Journal of Advanced Manufacturing Technology*, vol. 70, no. 5-8, pp. 1267–1282, 2014.

[11] M. Albareda-Sambola, A. Alonso-Ayuso, L. F. Escudero, E. Fernández, and C. Pizarro, "Fix-and-relax-coordination for a multi-period location–allocation problem under uncertainty," *Computers & operations research*, vol. 40, no. 12, pp. 2878–2892, 2013.

[12] A. Ghaderi and M. S. Jabalameli, "Modeling the budget-constrained dynamic uncapacitated facility location–network design problem and solving it via two efficient heuristics: a case study of health care," *Mathematical and Computer Modelling*, vol. 57, no. 3, pp. 382–400, 2013.

[13] E. Álvarez-Miranda, E. Fernández, and I. Ljubić, "The recoverable robust facility location problem," *Transportation Research Part B: Methodological*, vol. 79, pp. 93–120, 2015.

[14] T. Shiina, M. Umeda, J. Imaizumi, S. Morito, and C. Xu, "Inventory distribution problem via stochastic programming," *Asian Journal of Management Science and Applications*, vol. 1, no. 3, pp. 261–277, 2014.

[15] S. Nickel, F. Saldanha-da Gama, and H.-P. Ziegler, "A multi-stage stochastic supply network design problem with financial decisions and risk management," *Omega*, vol. 40, no. 5, pp. 511–524, 2012.

[16] T. E. Morton, "Bounds on the solution of the lagged optimal inventory equation with no demand backlogging and proportional costs," *SIAM review*, vol. 11, no. 4, pp. 572–596, 1969.

[17] T. E. Morton, "The near-myopic nature of the lagged-proportional-cost inventory problem with lost sales," *Operations Research*, vol. 19, no. 7, pp. 1708–1716, 1971.

[18] S. Nahmias, "Simple approximations for a variety of dynamic leadtime lost-sales inventory models," *Operations Research*, vol. 27, no. 5, pp. 904–924, 1979.

[19] K. van Donselaar, T. de Kok, and W. Rutten, "Two replenishment strategies for the lost sales inventory model: A comparison," *International Journal of Production Economics*, vol. 46, pp. 285–295, 1996.

[20] M. Bijvank and S. G. Johansen, "Periodic review lost-sales inventory models with compound poisson demand and constant lead times of any length," *European Journal of Operational Research*, vol. 220, no. 1, pp. 106–114, 2012.

[21] P. H. Zipkin, *Foundations of inventory management*, vol. 2. McGraw-Hill New York, 2000.

[22] D. Beyer, S. P. Sethi, and M. Taksar, "Inventory models with markovian demands and cost functions of polynomial growth," *Journal of Optimization theory and Applications*, vol. 98, no. 2, pp. 281–323, 1998.

[23] P. Zipkin, "Old and new methods for lost-sales inventory systems," *Operations Research*, vol. 56, no. 5, pp. 1256–1263, 2008.

[24] R. Levi, G. Janakiraman, and M. Nagarajan, "A 2-approximation algorithm for stochastic inventory control models with lost sales," *Mathematics of Operations Research*, vol. 33, no. 2, pp. 351–374, 2008.

[25] M. Bijvank and I. F. Vis, "Lost-sales inventory theory: A review," *European Journal of Operational Research*, vol. 215, no. 1, pp. 1–13, 2011.

[26] X. Li, "Managing dynamic inventory systems with product returns: a markov decision process," *Journal of Optimization Theory and Applications*, vol. 157, no. 2, pp. 577–592, 2013.

[27] D. P. Bertsekas, *Dynamic programming and optimal control*, vol. 1. Athena Scientific Belmont, MA, 1995.

[28] A. Shapiro, D. Dentcheva, and A. Ruszczyński, *Lectures on stochastic programming: modeling and theory*, vol. 16. SIAM, 2014.

[29] A. J. King and S. W. Wallace, *Modeling with stochastic programming*. Springer Science & Business Media, 2012.

[30] S. M. Ross, *Introduction to stochastic dynamic programming*. Academic press, 2014.

[31] R. Bellman, *Dynamic Programming*. Princeton, NJ, USA: Princeton University Press, 1 ed., 1957.

[32] M. V. Pereira and L. M. Pinto, "Multi-stage stochastic optimization applied to energy planning," *Mathematical programming*, vol. 52, no. 1-3, pp. 359–375, 1991.

[33] R. T. Rockafellar and R. J.-B. Wets, "Scenarios and policy aggregation in optimization under uncertainty," *Mathematics of operations research*, vol. 16, no. 1, pp. 119–147, 1991.

[34] A. Philpott and G. Pritchard, "*EMI-DOASA. Stochastic Optimization Limited*," 2013.

[35] M. De Lara, P. Carpentier, J.-P. Chancelier, and V. Leclère, "Optimization methods for the smart grid," *Report commissioned by the Conseil Français de l'Énergie, École des Ponts ParisTech*, 2014.

[36] R. J.-B. Wets, "Lecture notes in aggregation principle in stochastic optimization." http://cermics.enpc.fr/~delara/TEACHING/CEA-EDF-INRIA_2012/Roger_Wets4.pdf, June 2012.

[37] J. R. Birge and F. Louveaux, *Introduction to stochastic programming*. Springer Science & Business Media, 2011.