UNIVERSIDADE DE LISBOA
FACULDADE DE CIÊNCIAS
DEPARTAMENTO DE ESTATÍSTICA E INVESTIGAÇÃO OPERACIONAL

# A Multi-Period Stochastic Location-Inventory Problem

Inês Mendes Coelho

**Mestrado em Estatística e Investigação Operacional**
Especialização em Investigação Operacional

Dissertação orientada por:
Professor Doutor Francisco Saldanha da Gama

2023

# Acknowledgements

First of all, I would to like express my extreme gratitude to my advisor, Professor Francisco Saldanha da Gama, for all I have learned from him since the first year of my undergraduate degree and for all the support and collaboration during the development of this dissertation.

I would like to thank all the Professors that I have met during my whole academic path for all I have learnt from them. Especially, a big thank you to Professor Sónia Lucas Candeias.

I would also like to thank all my friends and colleagues that have been part of this journey.

Finally, I thank my family for all the patience and support.

Thank you all very much.

Inês Mendes Coelho

# Abstract

A two-stage stochastic linear programming model is proposed to formulate the *Multi-Period Stochastic Location-Inventory Problem*, where both location, allocation and inventory management related decisions are considered. A variant, which adds the concept of lead times between suppliers and DCs, is also formulated as a two-stage stochastic linear programming model.

In order to solve it, the concept of demand scenarios is introduced as a means to capture the uncertainty of the customers' demand. This way, the *Multi-Period Stochastic Location-Inventory Problem* can be formulated as a mixed-integer linear programming model. This is the model that needs to be solved, which can be done, for example, through the use of a commercial solver.

A set of instances is computationally generated for the purpose of performing computational tests. Afterwards, two batches of computational tests are run. The first batch uses the generated instances as they are, while in the second batch those instances have their fixed costs for locating a DC at some site modified (the original values are multiplied by one hundred).

Some characteristics and metrics are chosen in an effort to evaluate the quality of the solving approach. Most instances are solved in a considered suitable time (the majority take less than a minute). Only a few (the largest ones in both number of decision variables and constraints) are not solved due to hardware constraints.

**Keywords:** location-inventory; stochastic demand; multi-period, two-stage stochastic linear programming;

# Resumo


Nos dias correntes em que as cadeias de abastecimento são globais, os problemas de Investigação Operacional aplicados a este âmbito são de extrema relevância. Estes problemas podem ser classificados em três grandes categorias: problemas de localização, problemas de roteamento ou problemas de gestão de aprovisionamento (ou de cadeias de stock). Na maioria das vezes, um problema logístico insere-se dentro de apenas uma das categorias supramencionadas. No entanto, recentemente começaram a ser estudados problemas que podem pertencer a duas ou até mesmo às três categorias.

Nesta dissertação, o *Multi-Period Stochastic Location-Inventory Problem* é definido, sendo que pertence às categorias de localização e de gestão de aprovisionamento. Neste problema, existem três entidades diferentes, sendo elas: fornecedores, centros de distribuição e clientes. Considera-se que o horizonte temporal é composto por um número finito de períodos e admite-se que a procura dos clientes pode ser modelada através de uma distribuição de probabilidade, sendo esta conhecida.

Em cada período, podem ser instalados novos centros de distribuição, sendo que ficam a funcionar desde esse mesmo período até ao fim do horizonte temporal. Desta forma, torna-se óbvio que a afetação entre fornecedores e centros de distribuição e entre estes e os clientes pode sofrer alterações ao longo dos períodos. No entanto, para satisfazer a procura de um determinado período só pode haver uma única origem. Tal aplica-se tanto na satisfação da procura dos clientes como dos centros de distribuição.

Ainda assim, no que diz respeito à satisfação da procura dos clientes, ruturas de stock podem ocorrer e consequentemente é permitido que as encomendas sejam entregues num período posterior ao que era suposto. Ou seja, num determinado período, um cliente pode receber a encomenda $A$ proveniente do centro de distribuição $X$ referente à procura desse mesmo período e simultaneamente receber a encomenda $B$ proveniente do centro de distribuição $Y$ referente à procura de um período anterior.

A capacidade que os fornecedores possuem para satisfazer as encomendas dos centros de distribuição não tem limite e o mesmo se verifica entre os centros de distribuição e os clientes. Além do mais, admite-se que as entregas são instantâneas.

A formulação proposta para o *Multi-Period Stochastic Location-Inventory Problem* diz-se que é *two-stage stochastic linear programming*. Além deste problema, é do mesmo modo formulada uma extensão sua, onde são considerados tempos de entrega entre os fornecedores e os centros de distribuição. Em ambos, pretende-se minimizar o custo total esperado.

Dado que problemas de programação estocásticos não são fáceis de resolver e, por forma de ainda assim conseguir capturar a incerteza da procura estocástica, são considerados cenários de procura. Por exemplo, se a procura de determinado cliente para um certo período pode ser caracterizada pela distribuição Normal com valor esperado de 100 unidades e de desvio-padrão 15, cinco cenários para a sua procura desse período podiam então ser 116, 82, 91, 106 e 102 unidades.

Partindo da distribuição de probabilidade de cada cliente e para cada período, os cenários podem ser gerados através, por exemplo, do uso de técnicas expeditas de predição. Deste modo, o *Multi-Period Stochastic Location-Inventory Problem* passa a poder ser formulado em programação linear inteira-mista.

Por consequência, a fim de resolver uma instância deste problema, basta utilizar uma solver comercial, como, por exemplo, o CPLEX da IBM.

A fim de poder avaliar o valor do método de resolução proposto, é necessário efetuar testes computacionais. No entanto, para tal é necessário ter instâncias do problema. Estas foram geradas computacionalmente, sendo que os algoritmos desenvolvidos para esse propósito são descritos. Relativamente à procura, são admitidas duas situações. Uma em que a procura dos clientes segue distribuições Uniformes discretas e outra em que segue distribuições Normais.

Nos testes computacionais existem duas fases. Na primeira, pretende-se obter a solução ótima para as instâncias tal como elas foram geradas. Na segunda etapa, o objetivo é exatamente o mesmo, mas os custos fixos para localizar um centro de distribuição num determinado local foram alterados (tendo sido todos multiplicados por cem).

No conjunto das duas etapas, a solução ótima só não é encontrada em apenas uma ínfima fração das instâncias (sendo que a maioria delas coincidem com as maiores instâncias em termos de quantidade de variáveis de decisão (superiores a 3 000 000) e de restrições (superiores a 3 500 000)), devido a constrangimentos de *hardware*. Na maioria das instâncias em que se encontra a solução ótima, esta é encontrada em menos de um minuto.

Além do tempo demorado até a solução ótima ser encontrada, o número de variáveis de decisão e de restrições, são definidas outras métricas para avaliar a solução obtida e/ou o método proposto. Nomeadamente, o rácio, em percentagem, entre o número total de centros de distribuição a abrir e o número de localizações disponíveis para esse efeito (batizado de *DC ratio (%)*) e a percentagem da procura dos clientes que foi entregue sem atrasos (batizado de *CDDOT (%)*) para cada cenário de procura.

Dados os resultados registados, constata-se que os valores dos custos fixos para instalar num determinado local um centro de distribuição afetam de forma significativa o valor do *DC ratio (%)*. Quando a segunda fase de testes computacionais é efetuada, observa-se um claro decrescimento do número de centros de distribuição a abrir comparativamente aos resultados obtidos durante a primeira fase. Contabilizando apenas as instâncias em que a solução ótima é encontrada, o *DC ratio (%)* de 52.75% das instâncias utilizadas na primeira fase está compreendido entre 80% e 100%; enquanto o *DC ratio (%)* de 67.02% das da segunda etapa estão compreendidas entre 0% e 20%.

Relativamente aos valores do *CDDOT (%)* para cada cenário de procura, não se observam-se diferenças significativas entres as duas etapas dos testes computacionais. No geral, verifica-se um ligeiro decréscimo dos resultados da primeira para com os da segunda fase. No entanto, tal não ocorre com todas as instâncias em que a comparação é possível.

Excluindo as instâncias em que a solução ótima não é encontrada, apenas 14.89% das instâncias da segunda etapa têm cenários de procura com valores do *CDDOT (%)* inferiores a 75%. Já entre as instâncias da primeira fase, nenhuma delas contém cenários de procura com valores do *CDDOT (%)* inferiores a 75%.

Assim, tendo em conta a informação apresentada, considera-se que o método de resolução escolhido pode-se afirmar ser bom.

No entanto, devem ser realizados mais testes computacionais, não só com base em instâncias do problema geradas computacionalmente, mas tabmém com base em instâncias provenientes de situações reais. Adicionalmente, devem ser também consideradas outras métricas com o propósito de avaliar a solução obtida.

Além demais, devem ser também estudadas novas variantes do *Multi-Period Stochastic Location-Inventory Problem* estudado no âmbito desta dissertação. Nomeadamente, variantes em que se considerem capacidades não só nos centros de distribuição como também nos fornecedores; variantes em que é possível não só abrir novos centros de distribuição, mas também se pode fechar outros que existiam aquando do início do horizonte temporal. Por fim, também podem ser consideradas variantes em que as capacidades dos centros de distribuição sejam modulares, podendo-se ao longo do horizonte temporal tanto aumentar como diminuir a capacidade dos mesmos.

Por fim, devem ser aplicadas outros métodos de resolução e/ou ser desenvolvidos novos. Consequentemente, é necessário realizar testes computacionais que permitam a comparação entre as diferentes abordagens para resolver o problema e eventuais extensões.

**Palavras-chave:** *location-inventory*; procura estocástica; multi-período, *two-stage stochastic linear programming*

# Contents

# List of Figures

# List of Tables

# List of Algorithms

# Chapter 1

# Introduction

Everyday goods are being produced, stored, transported, bought and sold around the world. So the extreme relevance of Operations Research to the domain of Logistics is not at all surprising. There are three major segments of Operations Research problems applied to this domain. These are the Facility Location, Routing and Inventory Management problems. All of them contribute to the design of efficient and reliable supply chain networks. And each of them has their own high degree of complexity, which it is further increased whenever more details are incorporated. The more details are included in the problems, the better they resemble reality.

In Facility Location problems, the usual goal is to find the "best" way to locate one or more facilities so that a set of communities can be properly served by them. In Routing problems, the usual goal is to find the "best" routes to serve all communities allocated to a facility (or more). In Inventory Management problems, the main goals are to discover the quantity a facility should order and when the order should be placed so as to satisfy the demand.

Normally, these segments of problems are studied separately. However, they are not truly independent from each other, since everything is connected in the real world. For instance, the location of some facilities will affect which routes to use.

Therefore, one way to encapsulate this inter-connectivity could be to solve different, but interconnected, problems sequentially. However, the sequence will impact the final solution. Here is an example. If at first it is decided where to locate some services and how to allocate the services to the communities and afterwards it is decided what routes to use and only then it is chosen how to manage those services' inventory, the solution reached will differ from the one obtained had the order been other.

Nonetheless, making these decisions separately might led to sub-optimal results. Therefore, recently researchers have began to explore problems that encompass decisions from two or from all three segments. This seems to be a better option, since it should lead to better solutions (for instance, less costly solutions).

Then why are these segments of problems still to this day, more often than not, studied separately? These kind of optimisation problems are frequently computationally hard to solve on their own. And combining them does not mean that they become easier to solve.

In fact, many Location, Routing and Inventory Management problems belong to NP-Hard class. On practice, this is reflected by not existing (or having yet to be discovered) an exact algorithm capable of finding the optimal solution in polynomial time, no matter the problem's instance. Consequently, research is conducted to improve existing exact and approximate algorithms as well as to discover new ones.

Despite this detail, over time the problems studied have grown more complex as new optimisation techniques have been developed alongside technological evolution.

On the other hand, some problems are simply arduous to model. Particularly, if it is intended to model entirely through the use of linear expressions. Consequently, finding the optimal solution becomes even more challenging. Inventory Management problems are an example of this situation, where the objective function is customarily non-linear.

Going back to the topic of the detail level included in the problems, Chapter 2 of this dissertation will illustrate the on-going increase of detail in Facility Location problems and Inventory Management problems, culminating in a brief overview of Location-Inventory problems.

In Location-Inventory problems, there usually are at least one supplier, a set of possible locations for distribution centres, abbreviated as DCs, (or warehouses) and a set of customers. While the available suppliers and the customers are known, the customers' demand might not be. The goal is to satisfy all the demand at the lowest cost. Decisions regarding the location of DCs and the allocation of DCs to customers are common in these problems, but the same does not happen with the decisions regarding the inventory management. For this reason, Amiri-Aref et al. (2018) distinguish between joint and integrated Location-Inventory problems.

Amiri-Aref et al. (2018) stated that integrated Location-Inventory problems (such as the one they studied) encompass "the location-allocation decisions with the hierarchical integration of the periodic inventory policy and inventory replenishment decisions".

On the other hand, the joint ones can be seen as an extension of the *Uncapacitated Fixed-Charged Facility Location Problem* as Amiri-Aref et al. (2018) noted. The costs associated to the inventory management part of the problem are added to the objective function of the *Uncapacitated Fixed-Charged Facility Location Problem*, and thus no actual decisions regarding the inventory management are made. One of the earliest joint Location-Inventory problem studied was introduced by Daskin et al. (2002).

This dissertation aims to model the *Multi-Period Stochastic Location-Inventory Problem* and then to solve it. This problem considers a finite planning horizon, where the customers' demand, which while unknown, follow a known probability distribution. At each period, DCs can be installed and therefore the allocation between DCs and customers (as well as between suppliers and DCs) may change along the planning horizon.

The capacity for the suppliers and DCs to deliver goods to the DCs and customers, respectively is unbound. For both cases, the deliver is instantaneous. At each period, both operating DCs and customers are single-sourced, but the source does not have to be the same throughout the planning horizon. Stock-outs can occur and for that reason the customers' demand for some period may be satisfied at a latter time.

Considering the classification seen previously, the *Multi-Period Stochastic Location-Inventory Problem* falls under the integrated label, since there are location and allocation decisions as well as inventory management related decisions to be made at each time period.

The goal is to minimize the total expected cost during the planning horizon. For that, this problem is modelled as a two-stage stochastic linear programming model.

The remainder of this dissertation is organized as follows. In the next chapter a literature review concerning Facility Location, Inventory Management and Location-Inventory problems is performed, as mentioned previously. In Chapter 3 the problem under study is presented and modelled, a variant considering lead times between suppliers and DCs is also introduced and modelled and, at last, a resolution method is proposed. Chapter 4 focuses on the computational tests. Hence the data generation process is explained and the computational results are discussed. Finally, some conclusions are drawn and future work is suggested in Chapter 5.

# Chapter 2

# Literature Review

This chapter intends to present a brief overview of the work that has been done within the scope of Location Problems (Section 2.1), Inventory Management Problems (Section 2.2) and Location-Inventory Problems (Section 2.3) in order to show the relevance of this dissertation. For that, some models will be enunciated and briefly explained.

## 2.1 Facility Location Problems

In most facility location problems there are two major decisions. These are where to locate the facilities and the assignment of users (which in this section they will be known as customers or retailers) to working facilities. The main objective is usually to minimize the costs regarding the decisions. Overall, facility location problems are not just relevant in the field of supply chain management. They may be also used to help decision makers locating facilities such as schools, fire stations, and even telecommunications hubs or bank accounts.

In this section, a short review on facility location problems is presented, namely the *Uncapacitated Fixed-Charged Facility Location Problem* (Section 2.1.2), the *Capacitated Fixed-Charged Facility Location Problem* (Section 2.1.3), the *Stochastic Uncapacitated Fixed-Charged Facility Location Problem* (Section 2.1.4), the *Multi-Echelon Facility Location Problem* (Section 2.1.5) and the *Multi-Period Uncapacitated Facility Location Problem* (Section 2.1.6).

All the necessary notation for sets, parameters and decision variables is presented in Section 2.1.1. However, to avoid confusion, the notation used for the *Multi-Echelon Facility Location Problem* and the *Multi-Period Uncapacitated Facility Location Problem* is instead included in Sections 2.1.5 and 2.1.6, respectively. The notation and formulations presented in throughout these sections were adopted from chapter eight of the book by Snyder and Shen (2019), except for the the *Multi-Period Uncapacitated Facility Location Problem*, whose notation and formulation were adopted from Nickel and da Gama (2019).

### 2.1.1 Notation

**Sets**

$I$**:** set of customers;

$J$**:** set of potential facility locations;

$S$**:** set of scenarios (capturing uncertainty in parameters).

**Parameters**

$h_i$**:** annual demand of customer $i \in I$;

$c_{ij}$**:** cost to transport one unit of demand from facility $j \in J$ to customer $i \in I$;

$f_j$**:** fixed annual cost to open a facility at site $j \in J$;

$v_j$**:** maximum demand that can be served by facility $j \in J$ per year;

$h_{is}$**:** annual demand of customer $i \in I$ under scenario $s \in S$;

$c_{ijs}$**:** cost to transport one unit of demand from facility $j \in J$ to customer $i \in I$ under scenario $s \in S$;

$q_s$**:** probability that scenario $s \in S$ occurs $\left(\sum_{s \in S} q_s = 1\right)$.

**Decision Variables**

$$x_j = \begin{cases} 1, & \text{if facility } j \in J \text{ is opened;} \\ 0, & \text{otherwise;} \end{cases}$$

$y_{ij}$ **=** fraction of the demand of customer $i \in I$ that is served by facility $j \in J$;

$y_{ijs}$ **=** fraction of the demand of customer $i \in I$ that is served by facility $j \in J$ in scenario $s \in S$.

### 2.1.2 The Uncapacitated Fixed-Charged Facility Location Problem

A popular facility location problem is the so-called *Uncapacitated Fixed-Charged Facility Location Problem*. In this problem, it is decided where to locate facilities and how to assign the customers to working facilities, while minimizing the costs incurred.

An assumption made is that the facilities' capacity to satisfy the customers' demand is non-binding. As constraints, all demand must be supplied (constraints (2.2)) and only from open facilities (constraints (2.3)). Thus, this problem can be formulated as a mixed-integer linear programming model as it follows, where constraints (2.4) and (2.5) are the decision variables' domain constraints.

$$\text{minimize} \quad \sum_{j \in J} f_j x_j + \sum_{j \in J} \sum_{i \in I} h_i c_{ij} y_{ij} \tag{2.1}$$

$$\text{subject to} \quad \sum_{j \in J} y_{ij} = 1 \qquad \forall i \in I \tag{2.2}$$

$$y_{ij} \le x_j \qquad \forall i \in I,\ j \in J \tag{2.3}$$

$$x_j \in \{0,1\} \qquad \forall j \in J \tag{2.4}$$

$$y_{ij} \ge 0 \qquad \forall i \in I,\ j \in J \tag{2.5}$$

### 2.1.3 The Capacitated Fixed-Charged Facility Location Problem

The *Capacitated Fixed-Charged Facility Location Problem* is an extension of the *Uncapacitated Fixed-Charged Facility Location Problem*, since the facilities' capacity to satisfy the customers' demand may now be binding (constraints (2.6)). This is the only difference between these two problems.

$$\text{minimize} \quad \sum_{j \in J} f_j x_j + \sum_{j \in J} \sum_{i \in I} h_i c_{ij} y_{ij} \tag{2.1}$$

$$\text{subject to} \quad \sum_{j \in J} y_{ij} = 1 \qquad \forall i \in I \tag{2.2}$$

$$y_{ij} \leq x_j \qquad \forall i \in I,\ j \in J \tag{2.3}$$

$$\sum_{i \in I} h_i y_{ij} \leq v_j \qquad \forall j \in J \tag{2.6}$$

$$x_j \in \{0,1\} \qquad \forall j \in J \tag{2.4}$$

$$y_{ij} \geq 0 \qquad \forall i \in I,\ j \in J \tag{2.5}$$

A variant of the above problem imposes that each customer is supplied from a single facility, which calls to replace constraints (2.5) with $y_{ij} \in \{0,1\},\ \forall i \in I,\ j \in J$.

### 2.1.4 The Stochastic Uncapacitated Fixed-Charged Facility Location Problem

The *Stochastic Uncapacitated Fixed-Charged Facility Location Problem* is another extension of the *Uncapacitated Fixed-Charged Facility Location Problem*. This problem allows for uncertainty to be considered.

In the model formulation exposed here, the uncertainty is present in the customers' demand as well as in the transportation costs. Both may vary according to a set of different scenarios previously identified. The more scenarios are considered the larger the model will be and the longer it is expected to take to be solved.

$$\text{minimize} \quad \sum_{j \in J} f_j x_j + \sum_{s \in S} \sum_{j \in J} \sum_{i \in I} q_s h_{is} c_{ijs} y_{ijs} \tag{2.7}$$

$$\text{subject to} \quad \sum_{j \in J} y_{ijs} = 1 \qquad \forall i \in I,\ s \in S \tag{2.8}$$

$$y_{ijs} \leq x_j \qquad \forall i \in I,\ j \in J,\ s \in S \tag{2.9}$$

$$x_j \in \{0,1\} \qquad \forall j \in J \tag{2.4}$$

$$y_{ijs} \geq 0 \qquad \forall i \in I,\ j \in J,\ s \in S \tag{2.10}$$

### 2.1.5 The Multi-Echelon Facility Location Problem

The *Multi-Echelon Facility Location Problem* is an extension of the *Capacitated Fixed-Charge Facility Location Problem*, where each echelon represents a kind of facilities. It could also be known as the *Multi-Echelon Multi-Commodity Capacitated Fixed-Charge Facility Location Problem* and is usually present when one needs to design a supply chain network.

The model formulated in this section considers a three-echelon system, namely plants, distribution centres (DCs) and customers. The customers are known and so are the locations for possible plants and DCs. Therefore, the locating decisions apply to two echelons (plants and DCs). Similarly to the *Capacitated Fixed-Charge Facility Location Problem*, the customers' demand for each commodity and the capacities of the available sites for the different facilities to locate are known.

The goal of this problem is to minimize the costs incurred, just like in the other problems that have been seen. As constraints, all demand for each commodity must be supplied (constraints (2.12)) and only from open DCs, whose capacity must not be exceeded (constraints (2.13)). For each commodity and each DC, all quantity that arrives from the suppliers must be delivered to the customers (constraints (2.14)). The DCs can only be supplied by open plants, whose capacity must not be exceeded (constraints (2.15)). Constraints (2.16) - (2.19) are the decision variables' domain constraints.

The notation for sets, parameters and decision variables required is presented next, followed by the model formulation.

**Sets**

$I$**:** set of customers;

$J$**:** set of potential DC locations;

$K$**:** set of potential plant locations;

$L$**:** set of products.

**Parameters**

$h_{il}$**:** annual demand of customer $i \in I$ for product $l \in L$;

$v_j$**:** capacity of DC $j \in J$;

$b_k$**:** capacity of plant $k \in K$;

$s_l$**:** units of capacity consumed by one unit of product $l \in L$;

$f_j$**:** fixed annual cost to open a DC at site $j \in J$;

$g_k$**:** fixed annual cost to open a plant at site $k \in K$;

$c_{ijl}$**:** cost to transport one unit of product $l \in L$ from DC $j \in J$ to customer $i \in I$;

$d_{jkl}$**:** cost to transport one unit of product $l \in L$ from plant $k \in K$ to DC $j \in J$.

**Decision Variables**

$$x_j = \begin{cases} 1, & \text{if DC } j \in J \text{ is opened;} \\ 0, & \text{otherwise;} \end{cases}$$

$$z_k = \begin{cases} 1, & \text{if plant } k \in K \text{ is opened;} \\ 0, & \text{otherwise;} \end{cases}$$

$y_{ijl}$ **=** number of units of product $l \in L$ shipped from DC $j \in J$ to customer $i \in I$;

$w_{jkl}$ **=** number of units of product $l \in L$ shipped from plant $k \in K$ to DC $j \in J$.

$$
\begin{aligned}
\text{minimize} \quad & \sum_{j\in J} f_j x_j + \sum_{k\in K} g_k z_k + \sum_{l\in L}\left[\sum_{j\in J}\sum_{i\in I} c_{ijl} y_{ijl} + \sum_{k\in K}\sum_{j\in J} d_{jkl} w_{jkl}\right] && && (2.11)\\
\text{subject to} \quad & \sum_{j\in J} y_{ijl} = h_{il} && \forall i \in I,\ l \in L && (2.12)\\
& \sum_{i\in I}\sum_{l\in L} s_l y_{ijl} \leq v_j x_j && \forall j \in J && (2.13)\\
& \sum_{k\in K} w_{jkl} = \sum_{i\in I} y_{ijl} && \forall j \in J,\ l \in L && (2.14)\\
& \sum_{j\in J}\sum_{l\in L} s_l w_{jkl} \leq b_k z_k && \forall k \in K && (2.15)\\
& x_j \in \{0,1\} && \forall j \in J && (2.16)\\
& z_k \in \{0,1\} && \forall k \in K && (2.17)\\
& y_{ijl} \geq 0 && \forall i \in I,\ j \in J,\ l \in L && (2.18)\\
& w_{jkl} \geq 0 && \forall j \in K,\ k \in K,\ l \in L && (2.19)
\end{aligned}
$$

### 2.1.6 The Multi-Period Uncapacitated Facility Location Problem

The *Multi-Period Uncapacitated Facility Location Problem* is yet another extension of the *Uncapacitated Fixed-Charged Facility Location Problem.* This problem allows for the time dimension to be considered, by having a finite planning horizon divided into periods.

The notation for sets, parameters and decision variables required is presented next, followed by the model formulation.

**Sets**

$I$**:** set of customers;

$J$**:** set of potential DC locations;

$T$**:** set of time periods.

**Parameters**

$f_{jt}$**:** the cost for operating the facility $j \in J$ in period $t \in T$;

$c_{ijt}$**:** the cost for satisfying all the demand of customer $i \in I$ in period $t \in T$ from facility $j \in J$.

**Decision Variables**

$$x_{jt} = \begin{cases} 1, & \text{if a facility is operating at site } j \in J \text{ in period } t \in T; \\ 0, & \text{otherwise;} \end{cases}$$

$y_{ijt}$ **=** fraction of demand of customer $i \in I$ in period $t \in T$ that is supplied by facility $j \in J$.

$$\begin{aligned}
\text{minimize} \quad & \sum_{t \in T} \sum_{j \in J} f_{jt} x_{jt} + \sum_{t \in T} \sum_{j \in J} \sum_{i \in I} c_{ijt} y_{ijt} && && (2.20)\\
\text{subject to} \quad & \sum_{j \in J} y_{ijt} = 1 && \forall i \in I,\ t \in T && (2.21)\\
& \sum_{i \in I} y_{ijt} \leq |I| x_{jt} && \forall j \in J,\ t \in T && (2.22)\\
& x_{jt} \in \{0,1\} && \forall j \in J,\ t \in T && (2.23)\\
& y_{ijt} \geq 0 && \forall i \in I,\ j \in J,\ t \in T && (2.24)
\end{aligned}$$

However, the model formulation shown for this problem can be decomposed into $|T|$ single-period problems (i.e., $|T|$ *Uncapacitated Fixed-Charged Facility Location Problems*). This happens since what happens at some period is independent from what happens at another. For instance, it is possible at some period for a facility to be operating in some location, but not at the subsequent period. If the constraints $x_{jt} \leq x_{j,t+1},\ \forall j \in J,\ t \in T \setminus \{|T|\}$ are added to this formulation, the previous example will not occur. These constraints ensure that, once a facility is opened at some period, it will work at all the subsequent ones and constitutes a variant for the problem presented here. Therefore, this new problem can no longer be decomposed.

Correia and Melo (2016), Correia and Melo (2017) and Sauvey et al. (2020) addressed variants of the *Multi-Period Uncapacitated Facility Location Problem*. The models developed by Correia and Melo (2016) tackle the time dimension, while taking into consideration delivery lead times. Modular capacities for the facilities are considered as so is the option to close initially existing facilities. The customers' demand are known, hence it is deterministic. Not only these settings are also addressed by the models developed by Correia and Melo (2017), the use of modular capacities for the facilities is further exploited by allowing their expansion and contraction over the planning horizon. A particular detail of all these models is that there are two segments of customers based on their sensitivity to the delivery lead times. Some customers must receive their orders on time, while the others can receive their with a delay as long as it does not surpass a pre-established limit. Sauvey et al. (2020) developed heuristics for the problem introduced by Correia and Melo (2016).

## 2.2 Inventory Management Problems

In Inventory Management Problems there are two major decisions. These are when to place an order and what the order size shall be, which compose the inventory policy.

These problems can be characterized regarding various factors. Either there is a single commodity (or a family of commodities) or multiple ones. The demand can be deterministic or stochastic. The planning horizon can be finite or infinite. The replenishment of the stock can either be instantaneous or not. Either the facility produces itself the amount needed (production policy) or orders it from a supplier (order policy).

The inventory level can be monitored continuously (continuous review) or periodically (periodic review). In continuous review models, the inventory level is constantly assessed and an order will be placed whenever the stock level falls bellow a certain value. In periodic review models, the inventory level is assessed at discrete intervals, which is when ordering may be decided.

When demand surpass supply, stock-outs occur. Then, either there will be back-orders (the unmet orders will be delivery at a later time) or lost sales (the unmet orders will never be fulfilled). Even so, it is naturally relevant to ensure a good service. Generally, the service level is said to be good when the proportion of satisfied demand is high.

Two types of service level are usually mentioned in the literature, which are type 1 (or cycle service level) and type 2 (or fill rate). Snyder and Shen (2019) define the cycle service level as "the percentage of order cycles during which no stock-out occurs" (often denoted by $\alpha$ or $A$), and fill rate as "the percentage of demand that is met from stock" (often denoted by $\beta$ or $B$).

Most inventory management problems deal with imperishable inventory, also mentioned as stable inventory. So, there has to be specific models to deal with perishable commodities. Some might get spoiled (such as foods and medicine), become obsolete (for instance, technological products) or have a deadline to be sold (such as newspapers or airline tickets).

For a variety of reasons, it may be impossible to know or even accurately forecast the demand. Yet, when it is possible to described it by a random variable with some known probability distribution, the inventory management problems are said to be stochastic. In this case, it is common to mention the term "safety stock", which, as the name suggests, refers to the quantity held in stock to buffer against uncertainty.

For the case when the demand is deterministic, three models will be presented: the *Economic Order Quantity Model* (Section 2.2.1), the *Economic Order Quantity Model with Planned Stock-Outs and Non-Instantaneous Replenishment* (Section 2.2.2) and the *Wagner-Whitin Model* (Section 2.2.3). There are other variants of the *Economic Order Quantity Model*, where, for instance, lead times are considered or there are discounts based on the quantity purchased, but those will not be presented here.

For the case when the demand is stochastic, inventory policies are required. An inventory policy is the rule that dictates when to place and order and how much to order. An inventory management problem following the $(r,\ Q)$ policy is presented in Section 2.2.4. The $(s,\ S)$ policy is introduced in Section 2.2.5. Finally, The *Newsboy Problem* is presented in Section 2.2.6, which considers a perishable commodity (such as newspapers).

The notation used to present the models was adapted from Snyder and Shen (2019) and Hillier and Lieberman (2015).

### 2.2.1 The Economic Order Quantity Model

The *Economic Order Quantity Model* is one of the most well known inventory management models. There is a product, whose demand rate is deterministic and constant. The product's delivery and replenishment are instantaneous. The inventory is continuously reviewed and stock-outs are not allowed. It is considered a infinite planning horizon.

For this model, the goal is to find the order (batch) size, $Q$, and the cycle length (time between two consecutive stock replenishments), $T$. The following parameters are required:

$d$**:** demand rate per time unit;

$c$**:** cost per unit produced or purchased;

$K$**:** fixed cost for placing an order (it is independent of the order's size);

$h$**:** holding cost per unit per time unit held in inventory.

Figure 2.1 illustrates the evolution of the inventory level over time, whose behaviour repeats every cycle length. The objective is to minimize the total cost per time unit (function (2.25)), which is given by the quotient between the total cost per cycle and the length of one cycle.

$$f(Q) = \frac{dK}{Q} + cd + \frac{hQ}{2} \tag{2.25}$$

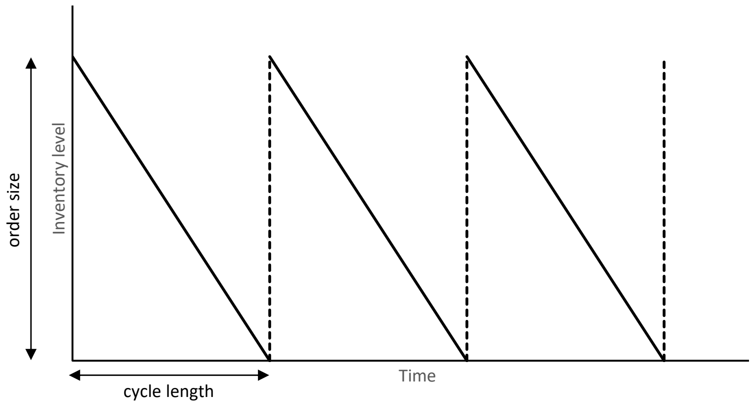



Figure 2.1: Evolution of the inventory level over time.

Ultimately, the optimal order size $Q^*$, cycle length $T^*$ and cost per unit of time $f(Q*)$ are given by the expressions (2.26) – (2.28), respectively.

$$Q^* = \sqrt{\frac{2dK}{h}} \tag{2.26}$$

$$T^* = \frac{Q^*}{d} \tag{2.27}$$

$$f(Q^*) = cd + \sqrt{2dKh} \tag{2.28}$$

### 2.2.2 The Economic Order Quantity Model with Planned Stock-Outs and Non-Instantaneous Replenishment

The *Economic Order Quantity Model with Planned Stock-Outs and Non-Instantaneous Replenishment* is an extension of the previous model. As the name suggests, back-orders are allowed when stock shortages occur and the replenishment of the inventory is not instantaneous.

In addition to the order size, $Q$, and the cycle length, $T$, it is relevant to find the value of the maximum stock-out, $S_{max}$, and the maximum inventory level, $I_{max}$. However, to achieve that goal, besides the parameters indicated in the previous model, the following ones are required:

**$p$:** stock-out cost per unit per time unit of time short;

**$r$:** replenishment rate per time unit ($r > d$).

Figure 2.2 illustrates the evolution of the inventory level over time, whose behaviour repeats every cycle length, just like in the former model. Once more, the objective is to minimize the total cost per time unit.

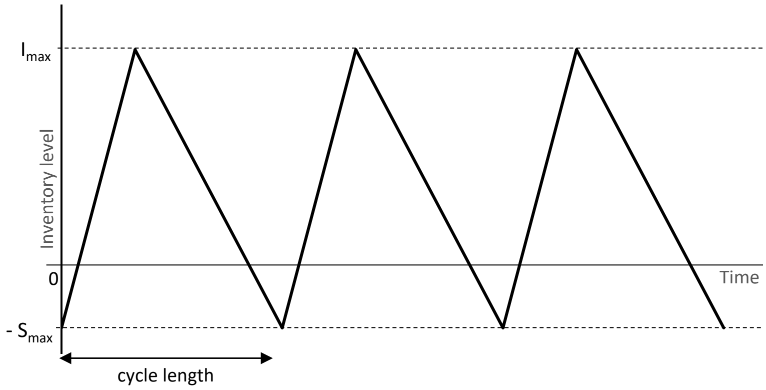



Figure 2.2: Evolution of the inventory level over time.

The optimal order size $Q^*$, cycle length $T^*$, cost per unit of time $f(Q*,\ S^*_{max})$, maximum stock-out $S^*_{max}$, maximum inventory level $I^*_{max}$ are given by the expressions (2.29) – (2.33), respectively.

$$Q^* = \sqrt{\frac{2dK}{h}}\sqrt{\frac{r}{r-d}}\sqrt{\frac{h+p}{p}} \tag{2.29}$$

$$T^* = \frac{Q^*}{d} = \sqrt{\frac{2K}{dh}}\sqrt{\frac{r}{r-d}}\sqrt{\frac{h+p}{p}} \tag{2.30}$$

$$f(Q^*,\ S^*_{max}) = cd + \sqrt{2dKh}\sqrt{\frac{r-d}{r}}\sqrt{\frac{p}{h+p}} \tag{2.31}$$

$$S^*_{max} = \sqrt{\frac{2dKh}{p(h+p)}}\sqrt{\frac{r-d}{r}} \tag{2.32}$$

$$I^*_{max} = Q^*(1-\frac{d}{r}) - S^*_{max} \tag{2.33}$$

Additionally, the fraction of time with no stock-outs is given by the expression (2.34).

$$1 - \frac{r}{r-d}\frac{S^*_{max}}{Q^*} \tag{2.34}$$

There are also the *Economic Order Quantity Model with Planned Stock-Outs* (the replenishment is instantaneous) and the *Economic Order Quantity Model with Non-Instantaneous Replenishment* (without planned stock-outs). From the model presented in this section, the optimal values for the previous models are obtained by considering $p \to +\infty$ and $r \to +\infty$, respectively.

### 2.2.3 The Wagner-Within Model

The *Wagner-Within Model*, also known as the *Uncapacitated Lot Sizing Model*, is another of most well known inventory management models. While in the previous ones the inventory review happens continuously, in the *Wagner-Within Model* it occurs periodically during a finite planning horizon. It is further assumed that replenishment occurs instantaneously and stock-outs are not allowed. At the beginning and at the end of the planning horizon there is no stock.

Let $T$ denote the set of time periods that compose the planning horizon. The parameters and decision variables required by this model are described below.

**Parameters**

$d_t$**:** demand in period $t$, $\forall t \in T$;

$c$**:** cost per unit produced or purchased;

$K$**:** fixed cost for producing or purchasing any units to replenish the inventory at the beginning of a period;

$h$**:** holding cost per unit per time unit held in inventory at the end of a period;

$M$**:** an arbitrarily large number.

**Decision Variables**

$x_t$ = amount to be purchased or produced in period $t$, $t \in T$;

$$y_t = \begin{cases} 1, & \text{if production or purchasing occurs in period } t \in T; \\ 0, & \text{otherwise;} \end{cases}$$

$q_t$ = inventory level at the end of period $t \in T$.

This model is formulated as a mixed-integer linear programming problem. While the decision variables $q_t$ are not necessarily required, because of equation (2.35), there are included so that the model is more easily understood. To further simplify it, let $q_0$ and $q_{|T|}$ denote the inventory level at the beginning and at the end of the planning horizon, respectively.

$$q_t = \sum_{s=1}^{t} (x_s - d_s) \tag{2.35}$$

Figure 2.3 illustrates how the relationship between the decision variables and the demand affect the inventory level at each period of the planning horizon. At each period, the existing stock comes from the amount produced or purchased at that time plus the amount held at the end of the prior period. However, part of that existing stock is destined to satisfy that period's demand and the rest will be hold for the next period. This situation is represented by constraints (2.37), which are often known as the inventory-balance constraints.

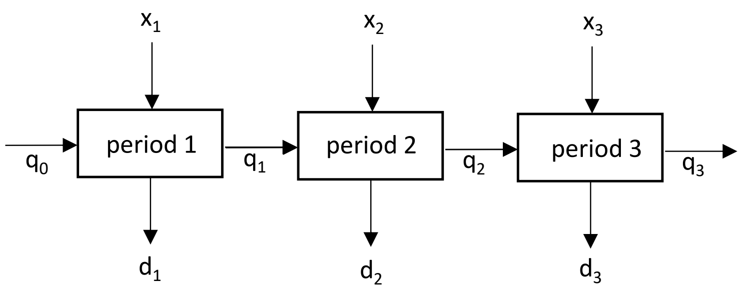



Figure 2.3: Stock evolution over a planning horizon composed by three time periods.

Since there is no stock at those times, let $q_0 = q_{|T|} = 0$, as indicated by constraint (2.38). This also facilitates modelling the circumstances where there is stock at any of those moments. Constraints (2.39) guarantees that the product will only be produced or purchased at some period if an order is placed at that same time. Finally, constraints (2.40) – (2.42) are the decision variables' domain constraints.

As seen in the other models, the objective is to minimize the total cost incurred with the decisions regarding the inventory management in the planning horizon, as seen in the objective function (2.36).

$$
\begin{aligned}
\text{minimize} \quad & \sum_{t \in T} (Ky_t + cx_t + hq_t) && && (2.36)\\
\text{subject to} \quad & q_{t-1} + x_t = q_t + d_t && \forall t \in T && (2.37)\\
& q_0 = q_{|T|} = 0 && && (2.38)\\
& x_t \leq My_t && \forall t \in T && (2.39)\\
& x_t \geq 0 && \forall t \in T && (2.40)\\
& y_t \in \{0,1\} && \forall t \in T && (2.41)\\
& q_t \geq 0 && \forall t \in T && (2.42)
\end{aligned}
$$

### 2.2.4 The $(r,\ Q)$ Policy

Consider a stochastic inventory management problem for a stable commodity, where the inventory level is checked continuously throughout a non-finite planning horizon. Lead times are taken into consideration, replenishment is instantaneous and stock-outs with back-orders are allowed.

In the *Economic Order Quantity Model*, an order was placed whenever there was not any stock left. Even though, in the new problem the demand is stochastic, the same reasoning is applied. thus, whenever the inventory level reaches the reorder point $r$, an order of size $Q$ must be placed. This is said to be a $(r,\ Q)$ inventory policy and these are the quantities to be determined. Figure 2.4 illustrates the evolution of the inventory level over time, which shows an unequal cyclic behaviour.

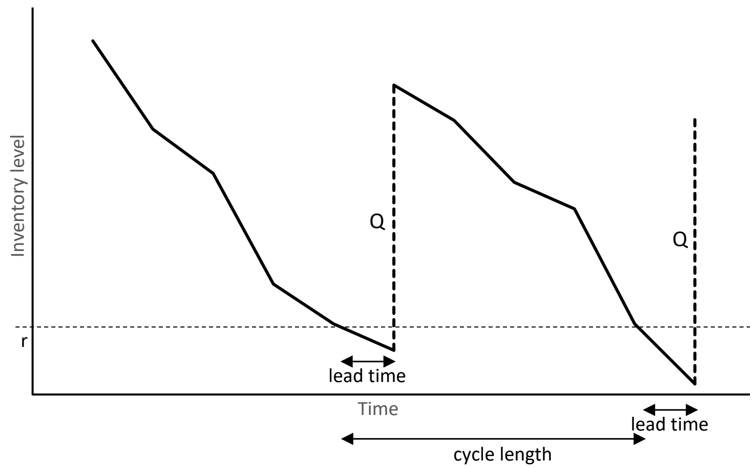



Figure 2.4: Evolution of the inventory level over time under the $(r,\ Q)$ policy.

Let $D$ be the continuous random variable that represents the demand per time unit. The expected demand per time unit is known and is denoted as $\mu_D$. Also, let $X$ be the random variable that represents the demand during the lead time, whose cumulative distribution function is denoted as $F_X(x) = P(X \leq x)$ and it is known as well. The following parameters are known as well:

$c$**:** cost per unit produced or purchased;

$K$**:** fixed cost for placing an order;

$h$**:** holding cost per unit per time unit held in inventory;

$p$**:** stock-out cost per unit short independently from the shortage time;

$l$**:** lead time.

The expected demand during the lead time is $\mu_X = \mu_D \times l$. The service level is the probability of stock-outs not occurring during the cycle. Hence, it is calculated as shown in equation (2.43), while the safety stock is calculated as shown in equation (2.44).

$$\text{Service level} = 1 - P(X > r) = F_X(r) \tag{2.43}$$

$$\text{Safety stock} = r - \mu_X \tag{2.44}$$

When the demand during the lead time is larger than the reorder point, it is said to have occurred outstanding orders. Its value is given by $max\{0,\ X - r\}$, so it is also a random variable. Therefore, its expected value, $\eta(r)$, is calculated by solving the integral in (2.45), where $f_X(x)$ represents the density distribution function of $X$.

$$\eta(r) = \int_0^{+\infty} max\{0,\ x - r\} f_X(x)dx = \int_r^{+\infty} (x - r) f_X(x)dx \tag{2.45}$$

On the other hand, the average on-hand inventory during a cycle, $\bar{I}$, is estimated by $Q/2 + r - \mu_X$, while the average cycle length, $\bar{T}$, is $Q/\mu_D$.

The total expected cost per cycle is the sum of the ordering cost (expression (2.46)), expected holding cost (expression (2.47)) and expected shortage cost (expression (2.48)).

$$\text{Ordering cost} = K + cQ \tag{2.46}$$

$$\text{Expected holding cost} = h\bar{I}\bar{T} = h(\frac{Q}{2} + r - \mu_X)\frac{Q}{\mu_D} \tag{2.47}$$

$$\text{Expected shortage cost} = p\eta(r) \tag{2.48}$$

However, the objective is to minimize the expected total cost per time unit (function (2.49)), which is given by the quotient between the total expected cost per cycle and the average cycle length.

$$f(r,\ Q) = \frac{K\mu_D}{Q} + c\mu_D + h(\frac{Q}{2} + r - \mu_X) + \frac{p\mu_D\eta(r)}{Q} \tag{2.49}$$

Contrary to what happens in the *Economic Order Quantity Model*, a closed form solution is not found. Instead, the system of equations (2.50) is obtained. Therefore, as a means to find the solution, an iterative procedure was defined. First, initialize it by having $Q_0 = \sqrt{2dK/h}$ and used it to find $r_0$ through the first equation, which is then used in the second equation to find $Q_1$. Then, use $Q_1$ in the first equation to find $r_1$ and $r_1$ in the second equation to find $Q_2$, and so on until the values stabilize.

$$\begin{cases} Q = & \sqrt{\dfrac{2\mu_D(K + p\eta(r))}{h}} \\ F_X(r) = & 1 - \dfrac{hQ}{p\mu_D} \end{cases} \tag{2.50}$$

### 2.2.5 The $(s,\ S)$ Policy

Consider an inventory management problem for a stable commodity, whose demand is stochastic, lead times can be taken into consideration and replenishment is instantaneous. The planning horizon is composed by multiple periods, where at each period the inventory level is observed (periodic-review policy). If it is lower than $s$, then an order is placed so that the inventory level rises to $S$. This constitutes the $(s,\ S)$ inventory policy, which is somewhat similar to the $(r,\ Q)$ inventory policy. In the $(s,\ S)$ policy, $s$ and $S$ are constants and not quantities to be determined as it is the case of the $(r,\ Q)$ policy. Yet, $s$ and $r$ are both known as the reorder points. $S$ is often known as the order-up-to level, while $Q$ is the order size.

In the $(s,\ S)$ policy, the values for both $s$ and $S$ may vary from period to period, although $s \leq S$. The orders' size do not have to be the same, even if $s$ and $S$ were fixed throughout the planning horizon.

Figure 2.5 illustrates the evolution of the inventory level over time, if the values for the reorder point and the order-up-to-level do not change across the planning horizon.

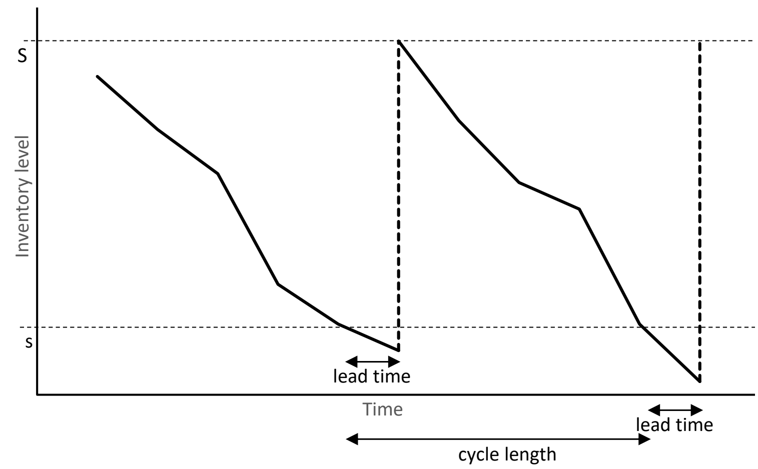



Figure 2.5: Evolution of the inventory level over time under the $(s,\ S)$ policy.

As it is the case for stochastic inventory management problems, minimizing the expected total cost for a cycle is not trivial.

### 2.2.6 The Newsboy Problem

The *Newsboy Problem*, also known as the *Newsvendor Problem*, is a classical example of stochastic optimisation as well of stochastic inventory management for a perishable product.

Consider a newsboy that each morning acquires $Q$ newspapers to sell during the day. That is the only moment he can buy newspapers to sell that day. Each newspaper costs the newsboy $p$ and he sells it for $c$. At the end of the day, he is able to return the unsold newspapers and gets $s$ for each copy (this is often known as the salvage cost). Let $D$ be the random variable that represents the newspapers demand for a day. Function (2.51) denotes the total cost for the newsboy if he buys $Q$ newspapers.

$$f(Q) = pQ - c\min\{D,\ Q\} - s\max\{Q - D,\ 0\} \tag{2.51}$$

The goal is to minimize the expected total cost, $E[f(Q)]$.

The *Newsboy Problem* is also a a particular case of the $S$ inventory policy, since it is a case of single-period inventory management problem.

In general, when the planning horizon is composed by multiple periods and under the $S$ inventory policy, the inventory level is observed at each period and an order is placed so that it rises to $S$. $S$ is often known as the base-stock level and it does not have to be the same for each period.

## 2.3 Facility Location and Inventory Management Problems

Location problems and inventory management have often been studied separately. Research combining these two areas is fairly recent. In fact, the articles by Daskin et al. (2002) and Shen et al. (2003) are two of the earliest related studies, which were prompted by a work on the blood bank system in the greater Chicago area.

Daskin et al. (2002) stated that platelets are "the most expensive and most perishable of all blood products". Each hospital supplied by the blood bank managed their own supply and their demand for platelets was irregular due to, as stated by Shen et al. (2003), "when they are needed, it is often the case that multiple units must be transfused at one time". These are some of the reasons that led to the existence of oversupplied hospitals (and therefore, greater amounts of expired platelets units), but also undersupplied hospitals (which would then need to place emergency orders at substantial costs).

The authors' idea was for the hospitals be supplied by a subset of them. Hence, the models proposed in both articles resemble a typical facility location model. The decisions to be made regard where to locate distribution centres (DCs) and the allocation of the retailers to a working DC. Therefore, the constraints required are straightforward: every retailer must be assigned to a single DC, and only functioning DCs can be assigned to retailers. The inventory management part of their models only shows up in the objective functions. Both papers aim at developing an objective function that includes the location and allocation related costs plus the costs regarding the inventory. In both articles, the terms that express the inventory costs are based on the Economic Order Quantity (EOQ) model. Thus, their objective functions are non-linear and the only explicit decisions made refer to the location and allocation part of the problem.

To solve their model, Daskin et al. (2002) proposed a Lagrangian relaxation based algorithm. Shen et al. (2003) proposed a set-covering approach to solve theirs, which was further explored by Shu et al. (2005).

Almost two decades later, these models served as the basis for the location-inventory model presented in the chapter dedicated to integrated models from the comprehensive book regarding Supply Chains by Snyder and Shen (2019). This model, which is non-linear, is presented next.

**Sets**

$I$**:** set of retailers;

$J$**:** set of potential DC sites.

**Parameters**

$\mu_i$**:** mean daily demand of retailer $i \in I$;

$\sigma_i^2$**:** variance daily demand of retailer $i \in I$;

$f_j$**:** fixed (daily) cost to open a DC at site $j \in J$;

$K_j$**:** fixed cost for DC $j \in J$ to place an order from the supplier, including fixed components of both ordering and transportation costs;

$c_j$**:** per-unit cost for each item ordered by DC $j \in J$ from the supplier, including per-unit inbound transportation;

$d_{ij}$**:** per-unit outbound transportation cost from DC $j \in J$ to retailer $i \in I$;

$h_j$**:** holding cost per unit per day at DC $j \in J$;

$L_j$**:** lead time (in days) to for orders placed by DC $j \in J$ to the supplier;

$\alpha$**:** desired fraction of DC order cycles during which no stockout occurs;

$z_\alpha$**:** $\alpha$-quantile from the Standard Normal Distribution.

**Decision Variables**

$$x_j = \begin{cases} 1, & \text{if site } j \in J \text{ is selected as a DC;} \\ 0, & \text{otherwise;} \end{cases}$$

$$y_{ij} = \begin{cases} 1, & \text{if retailer } i \in I \text{ is served by DC } j \in J; \\ 0, & \text{otherwise.} \end{cases}$$

The objective function is the sum of all costs and it is defined as shown in (2.52). Its first term, $\sum_{j \in J} f_j x_j$, gives the fixed location costs, while its second term, $\sum_{j \in J} \sum_{i \in I} \mu_i \left(c_j + d_{ij}\right) y_{ij}$, are the in-bound and out-bound per-unit costs. The objective function is non-linear, because of its last two terms, $\sqrt{2K_j h_j \sum_{i \in I} \mu_i y_{ij}} + h_j z_\alpha \sqrt{\sum_{i \in I} L_j \sigma_i^2 y_{ij}}$, which give the expected inventory costs (cycle stock cost plus safety stock cost) at the DCs, under the expected-inventory-level approximation and a $(r, Q)$ inventory policy with a type-1 service level.

$$Costs(\boldsymbol{x},\, \boldsymbol{y}) = \sum_{j \in J} \left[ f_j x_j + \sum_{i \in I} \mu_i \left(c_j + d_{ij}\right) y_{ij} + \sqrt{2K_j h_j \sum_{i \in I} \mu_i y_{ij}} + h_j z_\alpha \sqrt{\sum_{i \in I} L_j \sigma_i^2 y_{ij}} \right] \tag{2.52}$$

Thus, the formulation is as it follows:

$$\begin{aligned} \text{minimize} \quad & Costs(\boldsymbol{y},\, \boldsymbol{x}) \\ \text{subject to} \quad & \sum_{j \in J} y_{ij} = 1 \qquad \forall i \in I && (2.53) \\ & y_{it} \leq x_j \qquad \forall i \in I,\ j \in J && (2.54) \\ & x_j \in \{0, 1\} \qquad \forall j \in J && (2.55) \\ & y_{ij} \in \{0, 1\} \qquad \forall i \in I,\ j \in J && (2.56) \end{aligned}$$

Constraints (2.53) guarantees that each retailer is supplied by a DC, while constraints (2.54) ensures that each retailer will only have assigned a DC that is open. Constraints (2.55) and (2.56) are the decision variables' integrality and domain constraints. Note that single-allocation is being considered.

Amiri-Aref et al. (2018) studied stochastic location-inventory problem, which takes into consideration the time dimension and a decentralized periodic-review ($R$, $s$, $S$) inventory control policy. At the end of each period, the inventory is reviewed (hence $R = 1$) and, if the inventory is less or equal to the reorder point $s$, an order is placed so that it reaches the order-up-to-level $S$. The customers' demand follow a compound non-stationary stochastic process.

## 2. LITERATURE REVIEW

The problem is formulated as a two-stage stochastic model, where the goal is to maximize the total profit. Despite being taken into account, not all decisions are time dependent. Those that are not time dependent are known as the design decisions and regard the DCs location, the customers allocation and the inventory policy. The decisions regarding the inventory level, the order quantity and the transportation flow are known as the periodic planning decisions and are time dependent. In order to solve their model, Amiri-Aref et al. (2018) used a sample average approximation technique based on the linear approximation of the two-stage stochastic model.

# Chapter 3

# Problem Description

In this chapter the *Multi-Period Stochastic Location-Inventory Problem* is defined (Section 3.1). In Section 3.2, an optimisation model is proposed, while in Section 3.3 an extension is introduced. At last, a solution approach is proposed in Section 3.4.

## 3.1 Problem Statement

Consider a finite planning horizon divided into time periods. At any given time period, the flow of a single commodity goes according to the Figure 3.1. Suppliers deliver the goods to distribution centres (DCs) and these are responsible for supplying the customers.

The suppliers and customers are known and so are the possible locations for the distribution centres. The suppliers' production capacity is unlimited and so is the transportation capacity from the suppliers and DCs. The DCs' inventory capacity from a time period to the subsequent one is also unlimited. At the beginning of the planning horizon, there is no DC working. Thus, the system is being planned from scratch. Nevertheless, once a DC is installed it will stay so until the end of the planning horizon. Furthermore, for each period there is a given maximum number of DCs that can be installed.

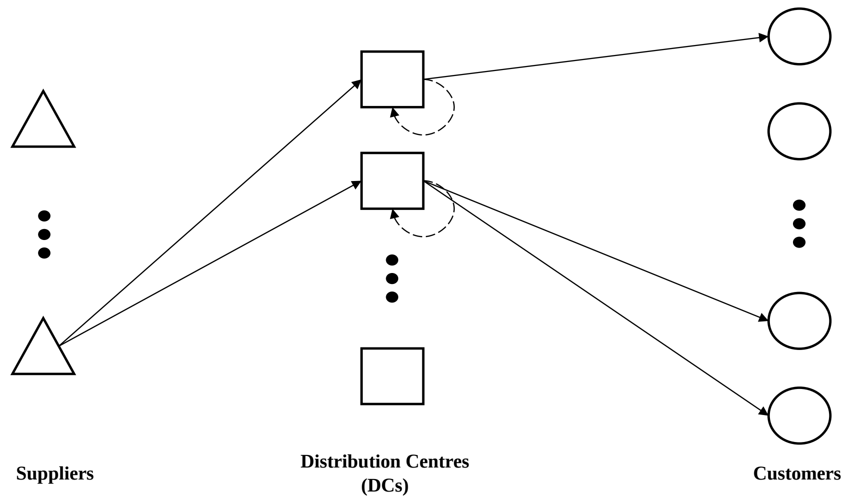



Figure 3.1: Product flow from suppliers to customers at some period from the planning horizon

It is assumed that for each period the DCs and customers are supplied from a single source. However, that source may change over the planning horizon. Stock-outs are allowed to occur and so it is back-ordering. In other words, a customer's demand for some period can be satisfied at a later time.

This problem involves different costs and revenues, whose values are given. They are the following:

1. cost for installing a DC;
2. cost for allocating a supplier to a DC (includes, for instance, the transportation costs);
3. cost for allocating a DC to a customer (includes, for instance, the transportation costs);
4. fixed cost for ordering by a DC from a supplier;
5. cost per unit ordered by a DC from a supplier;
6. unitary holding cost at a DC;
7. revenue per unit sold by a DC to a customer (revenues may differ if the order was delivered on time or with a delay).

What it is not known beforehand is the customers' demand for each period. Nonetheless, it is assumed to follow a given probability distribution (for example, Normal, Exponential or Log-Normal). It is further assumed that all the random variables (i.e., demands) are independent from each other.

The goal is to minimize the costs incurred, while satisfying the customers' demand throughout the planning horizon. The decisions to be made are time dependent and comprise of:

1. those typically found in facility location problems:
   (a) when and where to install DCs;
   (b) how to allocate the workings DCs to suppliers;
   (c) how to allocate the customers to working DCs;
2. and those commonly found in inventory management problems:
   (a) when a working DC should be supplied;
   (b) how much a DC should receive;
   (c) how much a DC should hold for the subsequent period;
   (d) how much the customers should receive on time;
   (e) how much the customers should receive due to back-ordering.

It is assumed that a DC is fully operational from the beginning of the time period in which it is installed.

## 3.2 Model Formulation

The problem previously described is here formulated as a two-stage stochastic linear model. In the first-stage, the location and allocation decisions are made, while the decisions regarding inventory management are tackled in the second-stage (thus adapting to the observed demand).

Consider a set $K$ of suppliers, a set $I$ of possible locations for distribution centres (DCs), a set $J$ of customers and a set $T$ of time periods that comprise the planning horizon. As parameters, besides seven types regarding costs and revenues (each is described below), there is also the number of DCs that can be installed at each time period.

**Sets**

$K$**:** set of suppliers;

$I$**:** set of potential distribution centres (DCs);

$J$**:** set of customers;

$T$**:** set of time periods;

$T_t = \{1, 2, \dots, t\} \subseteq T$**:** a subset of time periods.

**Parameters**

$F_{it}$**:** fixed cost of locating DC $i \in I$ in period $t \in T$, plus any other incurred costs (for instance, maintenance related) from period $t$ until the end of the planning horizon;

$C^1_{ikt}$**:** cost of allocating DC $i \in I$ to supplier $k \in K$ in period $t \in T$;

$C^2_{ijt}$**:** cost of allocating DC $i \in I$ to customer $j \in J$ in period $t \in T$;

$n_t$**:** maximum number of DCs that can be installed in period $t \in T$;

$g_{ikt}$**:** fixed cost incurred by DC $i \in I$ when placing an order at supplier $k \in K$ at the beginning of period $t \in T$;

$a_{ikt}$**:** cost for each item ordered by DC $i \in I$ from the supplier $k \in K$ at the beginning of period $t \in T$;

$h_{it}$**:** unit holding cost at DC $i \in I$ during period $t \in T$;

$p_{ijts}$**:** unit revenue for each item sold to customer $j \in J$ by DC $i \in I$ in period $t \in T$ to satisfy the demand from period $s \in T_t$;

$M$**:** an arbitrarily large number.

**Random Variables**

$\boldsymbol{\xi} = \left[\xi_{jt}\right]_{j \in J,\ t \in T}$**,** where $\xi_{jt}$ represents the demand of customer $j \in J$ at time period $t \in T$

**First-Stage Decision Variables**

$$y_{it} = \begin{cases} 1, & \text{if DC } i \in I \text{ is operating in period } t \in T \\ 0, & \text{otherwise} \end{cases}$$

$$w_{ikt} = \begin{cases} 1, & \text{if DC } i \in I \text{ is supplied from supplier } k \in K \text{ in period } t \in T \\ 0, & \text{otherwise} \end{cases}$$

$$x_{ijt} = \begin{cases} 1, & \text{if DC } i \in I \text{ supplies customer } j \in J \text{ in period } t \in T \\ 0, & \text{otherwise} \end{cases}$$

**Second-Stage Decision Variables**

$$r_{ikt}(\boldsymbol{\xi}) = \begin{cases} 1, & \text{if DC } i \in I \text{ places an order at supplier } k \in K \text{ in the beginning of period } t \in T \\ & \text{under customers' demand scenario } \boldsymbol{\xi}; \\ 0, & \text{otherwise;} \end{cases}$$

$q_{ikt}(\boldsymbol{\xi}) =$ quantity DC $i \in I$ receives from supplier $k \in K$ in the beginning of period $t \in T$ under customers' demand scenario $\boldsymbol{\xi}$;

$u_{it}(\boldsymbol{\xi}) =$ quantity that remains in DC $i \in I$ at the end of period $t \in T$ under customers' demand scenario $\boldsymbol{\xi}$;

$v_{ijts}(\boldsymbol{\xi}) =$ quantity DC $i \in I$ sends to customer $j \in J$ in period $t \in T$ to satisfy their demand from period $s \in T_t$ under customers' demand scenario $\boldsymbol{\xi}$.

Additionally, to simplify the model formulation, consider the following decision variables lookalikes, but they are in fact parameters:

$y_{i0} = 0, \;\; \forall i \in I$, because initially no DC is operating;

$u_{i0}(\boldsymbol{\xi}) = 0, \;\; \forall i \in I$, because initially DC $i$ does not hold anything.

The two-stage stochastic model is presented in the following page. The goal is to minimize the costs incurred, as shown in both objective functions (3.1) and (3.10).

Constraints (3.2) ensure that the maximum number of DCs that can be installed at each period is not surpassed, while constraints (3.3) guarantee that the DCs will be working all the subsequent periods from the moment they are installed.

Constraints (3.4) and (3.5) ensure for each period each customer's demand is fulfilled by a single DC and this DC must be a working one. Constraints (3.6) ensure that, for each period, working DCs are supplied by only one supplier and non-working DCs are not supplied at all.

Constraints (3.11) are the inventory-balance constraints.

For each period, an order can only arrive to a DC from a supplier if an order was placed in that same period, which is guaranteed by constraints (3.12). However, constraints (3.13) ensure that a DC can only place an order to a supplier if they are allocated in the same period.

Constraints (3.14) assure that, for each period, a DC can only hold anything in stock if it is a working DC. And constraints (3.15) guarantee that a DC can only satisfy a customer's demand from a period if they are allocated in that same period.

Each customer's demand for each period must always be completely satisfied, which is ensured by constraints (3.16).

Constraints (3.7)–(3.9) and (3.17)–(3.20) define the domain of the decision variables.

The first term of the objective function (3.1) is the total cost of opening and operating DCs. Its second term is the total cost of the allocation between suppliers and DCs, while the third term is the total allocation cost between DCs and customers. The fourth and final term represents the expected cost of the inventory management in all DCs for the whole planning horizon, given the decisions $\boldsymbol{y}$, $\boldsymbol{w}$, $\boldsymbol{x}$ and the customers' demand $\boldsymbol{\xi}$.

The first term of the objective function (3.10) is the total cost of the commodities' order, which includes both the fixed cost per order and the unitary cost per unit. The total holding cost for DCs is expressed by the second term of this objective function. Finally, the last term expresses the total revenue from satisfying the customers' demand.

$$\text{minimize} \quad \sum_{t\in T}\sum_{i\in I} F_{it}(y_{it}-y_{i,t-1}) \; + \; \sum_{t\in T}\sum_{i\in I}\sum_{k\in K} C^1_{ikt} w_{ikt} + \sum_{t\in T}\sum_{i\in I}\sum_{j\in J} C^2_{ijt} x_{ijt} + \mathscr{Q}(\boldsymbol{y},\boldsymbol{w},\boldsymbol{x}) \tag{3.1}$$

$$\text{subject to} \quad \sum_{i\in I}(y_{it}-y_{i,t-1}) \le n_t \qquad \forall t\in T \tag{3.2}$$

$$y_{it} \ge y_{i,t-1} \qquad \forall i\in I,\ t\in T \tag{3.3}$$

$$\sum_{i\in I} x_{ijt} = 1 \qquad \forall j\in J,\ t\in T \tag{3.4}$$

$$x_{ijt} \le y_{it} \qquad \forall i\in I,\ j\in J,\ t\in T \tag{3.5}$$

$$\sum_{k\in K} w_{ikt} = y_{it} \qquad \forall i\in I,\ t\in T \tag{3.6}$$

$$y_{it} \in \{0,1\} \qquad \forall i\in I,\ t\in T \tag{3.7}$$

$$w_{ikt} \in \{0,1\} \qquad \forall i\in I,\ k\in K,\ t\in T \tag{3.8}$$

$$x_{ijt} \in \{0,1\} \qquad \forall i\in I,\ j\in J,\ t\in T \tag{3.9}$$

$\mathscr{Q}(\boldsymbol{y},\boldsymbol{w},\boldsymbol{x}) = E[Q(\boldsymbol{y},\boldsymbol{w},\boldsymbol{x},\boldsymbol{\xi})]$ is called the recourse function, where $Q(\boldsymbol{y},\boldsymbol{w},\boldsymbol{x},\boldsymbol{\xi}) =$

$$\text{minimize} \quad \sum_{t\in T}\sum_{k\in K}\sum_{i\in I}\Big(g_{ikt} r_{ikt}(\boldsymbol{\xi}) + a_{ikt} q_{ikt}(\boldsymbol{\xi})\Big) + \sum_{t\in T}\sum_{i\in I} h_{it} u_{it}(\boldsymbol{\xi}) \; - \; \sum_{t\in T}\sum_{s\in T_t}\sum_{j\in J}\sum_{i\in I} p_{ijts} v_{ijts}(\boldsymbol{\xi}) \tag{3.10}$$

$$\text{subject to} \quad \sum_{k\in K} q_{ikt}(\boldsymbol{\xi}) + u_{i,t-1}(\boldsymbol{\xi}) = u_{it}(\boldsymbol{\xi}) + \sum_{j\in J}\sum_{s\in T_t} v_{ijts}(\boldsymbol{\xi}) \qquad \forall i\in I,\ t\in T \tag{3.11}$$

$$q_{ikt}(\boldsymbol{\xi}) \le M r_{ikt}(\boldsymbol{\xi}) \qquad \forall i\in I,\ k\in K,\ t\in T \tag{3.12}$$

$$r_{ikt}(\boldsymbol{\xi}) \le w_{ikt} \qquad \forall i\in I,\ k\in K,\ t\in T \tag{3.13}$$

$$u_{it}(\boldsymbol{\xi}) \le M y_{it} \qquad \forall i\in I,\ t\in T \tag{3.14}$$

$$v_{ijtt}(\boldsymbol{\xi}) + \sum_{s\in T\setminus T_t} v_{ijst}(\boldsymbol{\xi}) \le M x_{ijt} \qquad \forall i\in I,\ j\in J,\ t\in T \tag{3.15}$$

$$\sum_{i\in I}\left(v_{ijtt}(\boldsymbol{\xi}) + \sum_{s\in T\setminus T_t} v_{ijst}(\boldsymbol{\xi})\right) = \xi_{jt} \qquad \forall j\in J,\ t\in T \tag{3.16}$$

$$r_{ikt}(\boldsymbol{\xi}) \in \{0,1\} \qquad \forall i\in I,\ k\in K,\ t\in T \tag{3.17}$$

$$q_{ikt}(\boldsymbol{\xi}) \ge 0 \qquad \forall i\in I,\ k\in K,\ t\in T \tag{3.18}$$

$$u_{it}(\boldsymbol{\xi}) \ge 0 \qquad \forall i\in I,\ t\in T \tag{3.19}$$

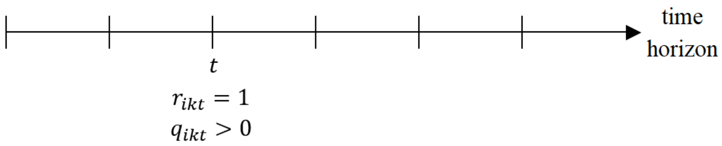
$$v_{ijts}(\boldsymbol{\xi}) \ge 0 \qquad \forall i\in I,\ j\in J,\ t\in T,\ s\in T_t \tag{3.20}$$

Given the structure of the model, for the problem to be feasible, at least one DC may be installed in the first period. Besides considering that capacity constraints were included in the problem being studied, it is highly unlike for a problem instance not to be feasible.

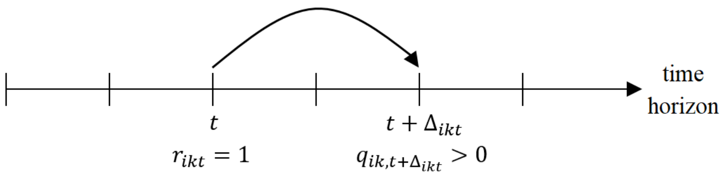
## 3.3 A Model Extension

The above model assumes negligible lead times. Suppose now that these need to be accounted for. In this case, only the second-stage model (and thus its related decision variables) requires some changes.

First, let $\chi_t$ be the number of days in time period $t\in T$ and denote by $l_{ikt}$ the lead time also in days, from supplier $k\in K$ to DC $i\in I$ in period $t\in T$. Define $\Delta_{ikt} = \left\lfloor \frac{l_{ikt}}{\chi_t} \right\rfloor$ as the number of periods it takes an order placed in the beginning of period $t$ to arrive to DC $i$ from supplier $k$. Also, let $T_{ik} = \{t + \Delta_{ikt} \in T : t \in T\}$ be the set of periods when DC $i \in I$ can receive an order from supplier $k \in K$.




(a) Case when there are no lead times between a supplier $k$ and a DC $i$.




(b) Case when there are lead times between a supplier $k$ and a DC $i$.

Figure 3.2: Comparison between the moment an order is placed at period $t$ and the moment of its delivery without (Figure 3.2a) and with lead times (Figure 3.2b).

Although there is no need for considering additional decision variables, those associated to the suppliers need some modifications. In the case of variables $r_{ikt}(\boldsymbol{\xi})$, they retain the same meaning, but now $t \in T_{ik}$ instead of $t \in T$. However, the variables $q_{ikt}(\boldsymbol{\xi})$ are now defined as it follows:

$q_{ikt}(\boldsymbol{\xi})$ = quantity DC $i \in I$ receives from supplier $k \in K$ during period $t \in T_{ik}$ from an order placed in the beginning of period $t - \Delta_{ikt}$ under customers' demand scenario $\boldsymbol{\xi}$.

To simplify the notation, let $q_{ikt}(\boldsymbol{\xi}) = 0,\ \forall i \in I,\ k \in K,\ t \in T \setminus T_{ik}$. This way, $q_{ikt}(\boldsymbol{\xi})$ is defined for all the periods of the planning horizon.

Neither do the objective function nor the constraints suffer major alterations and therefore their meaning remains the same. The second-stage model for the lead time variant of the problem is as it follows:

$$
\begin{aligned}
\text{minimize} \quad & \sum_{i \in I} \sum_{k \in K} \sum_{t \in T_{ik}} \left( g_{ik,t-\Delta_{ikt}} r_{ik,t-\Delta_{ikt}}(\boldsymbol{\xi}) + a_{ik,t-\Delta_{ikt}} q_{ikt}(\boldsymbol{\xi}) \right) - \sum_{t \in T} \sum_{s \in T_t} \sum_{j \in J} \sum_{i \in I} p_{ijts} v_{ijts}(\boldsymbol{\xi}) + \\
& + \sum_{t \in T} \sum_{i \in I} h_{it} u_{it}(\boldsymbol{\xi}) && (3.21) \\
\text{subject to} \quad & \sum_{k \in K} q_{ikt}(\boldsymbol{\xi}) + u_{i,t-1}(\boldsymbol{\xi}) = u_{it}(\boldsymbol{\xi}) + \sum_{j \in J} \sum_{s \in T_t} v_{ijts}(\boldsymbol{\xi}) && \forall i \in I,\ t \in T \quad (3.22) \\
& q_{ikt}(\boldsymbol{\xi}) \le M r_{ikt-\Delta_{ikt}}(\boldsymbol{\xi}) && \forall i \in I,\ k \in K,\ t \in T_{ik} \quad (3.23) \\
& r_{ikt-\Delta_{ikt}}(\boldsymbol{\xi}) \le w_{ikt-\Delta_{ikt}} && \forall i \in I,\ k \in K,\ t \in T_{ik} \quad (3.24) \\
& u_{it}(\boldsymbol{\xi}) \le M y_{it} && \forall i \in I,\ t \in T \quad (3.25) \\
& v_{ijtt}(\boldsymbol{\xi}) + \sum_{s \in T \setminus T_t} v_{ijst}(\boldsymbol{\xi}) \le M x_{ijt} && \forall i \in I,\ j \in J,\ t \in T \quad (3.26) \\
& \sum_{i \in I} \left( v_{ijtt}(\boldsymbol{\xi}) + \sum_{s \in T \setminus T_t} v_{ijst}(\boldsymbol{\xi}) \right) = \xi_{jt} && \forall j \in J,\ t \in T \quad (3.27) \\
& r_{ikt}(\boldsymbol{\xi}) \in \{0, 1\} && \forall i \in I,\ k \in K,\ t - \Delta_{ikt} \in T_{ik} \quad (3.28) \\
& q_{ikt}(\boldsymbol{\xi}) \ge 0 && \forall i \in I,\ k \in K,\ t \in T_{ik} \quad (3.29) \\
& u_{it}(\boldsymbol{\xi}) \ge 0 && \forall i \in I,\ t \in T \quad (3.30) \\
& v_{ijts}(\boldsymbol{\xi}) \ge 0 && \forall i \in I,\ j \in J,\ t \in T,\ s \in T_t \quad (3.31)
\end{aligned}
$$

## 3.4 Solution Approach

It is rare that the customers' future demand is properly known. However, through a variety of forecasting methods, one may think about defining a set of possible demand scenarios and also estimate their occurrence probability. This is the main reasoning leading to the solution approach for the problem being studied.

In the two-stage stochastic linear models introduced previously, $\boldsymbol{\xi}$ is a $|J| \times |T|$ matrix, where each entry $\xi_{jt}$ is a random variable representing the demand of customer $j \in J$ in period $t \in T$. Let $\Phi$ be the finite set of demand scenarios to be considered. For each scenario $\varphi \in \Phi$, there will be a matrix $\boldsymbol{\xi}^{\varphi}$, where each entry $\xi_{jt}^{\varphi}$ is the known demand of customer $j \in J$ for the period $t \in T$ under that scenario. Also let the parameter $\Pi^{\varphi}$ denote the probability that demand scenario $\varphi \in \Phi$ occurs, where $\sum_{\varphi \in \Phi} \Pi^{\varphi} = 1$.

Due to these considerations, the two-stage stochastic linear models above described can be rewritten as mixed-integer linear programming models. Notice that only the decisions variables and constraints involving the random matrix $\boldsymbol{\xi}$ require adaptations. All parameters introduced so far remain unchanged.

The mixed-integer linear programming model for the case where lead times between suppliers and DCs are not considered is discussed now. First, let all the decision variables be stated.

$$y_{it} = \begin{cases} 1, & \text{if DC } i \in I \text{ is operating in period } t \in T; \\ 0, & \text{otherwise;} \end{cases}$$

$$w_{ikt} = \begin{cases} 1, & \text{if DC } i \in I \text{ is supplied from supplier } k \in K \text{ in period } t \in T; \\ 0, & \text{otherwise;} \end{cases}$$

$$x_{ijt} = \begin{cases} 1, & \text{if DC } i \in I \text{ supplies customer } j \in J \text{ in period } t \in T; \\ 0, & \text{otherwise;} \end{cases}$$

$$r_{ikt}^{\varphi} = \begin{cases} 1, & \text{if DC } i \in I \text{ places an order at supplier } k \in K \text{ in the beginning of period } t \in T \text{ under} \\ & \text{customers' demand scenario } \varphi \in \Phi; \\ 0, & \text{otherwise;} \end{cases}$$

$q_{ikt}^{\varphi}$ = quantity DC $i \in I$ receives from supplier $k \in K$ in the beginning of period $t \in T$ under customers' demand scenario $\varphi \in \Phi$;

$u_{it}^{\varphi}$ = quantity that remains in DC $i \in I$ at the end of period $t \in T$ given customers' demand scenario $\varphi \in \Phi$;

$v_{ijts}^{\varphi}$ = quantity DC $i \in I$ sends to customer $j \in J$ in period $t \in T$ to satisfy their demand from period $s \in T_t$ under customers' demand scenario $\varphi \in \Phi$.

Just like it was done in Section 3.2, to simplify the model formulation, consider the following additional parameters:

$y_{i0} = 0, \ \forall i \in I$, because initially no DC is operating;

$u_{i0}^{\varphi} = 0, \ \forall i \in I, \varphi \in \Phi$, because initially DC $i$ does not hold anything.

In the mixed-integer linear programming model, the goal remains the same, which is to minimize the costs incurred. These are calculated as shown in the function (3.32).

$$Costs(\boldsymbol{y},\, \boldsymbol{w},\, \boldsymbol{x},\, \boldsymbol{r},\, \boldsymbol{q},\, \boldsymbol{u},\, \boldsymbol{v}) = \sum_{t\in T}\sum_{i\in I} F_{it}(y_{it} - y_{i,t-1}) + \sum_{t\in T}\sum_{i\in I}\sum_{k\in K} C^1_{ikt} w_{ikt} + \sum_{t\in T}\sum_{i\in I}\sum_{j\in J} C^2_{ijt} x_{ijt} +$$
$$+ \sum_{\varphi\in\Phi} \Pi^{\varphi} \left[ \sum_{t\in T}\sum_{i\in I}\sum_{k\in K} \left( g_{ikt} r^{\varphi}_{ikt} + a_{ikt} q^{\varphi}_{ikt} \right) + \sum_{t\in T}\sum_{i\in I} h_{it} u^{\varphi}_{it} - \right.$$
$$\left. - \sum_{t\in T}\sum_{s\in T_t}\sum_{i\in I}\sum_{j\in J} p_{ijts} v^{\varphi}_{ijts} \right] \tag{3.32}$$

The constraints are slightly changed, considering the two-stage stochastic linear model initially proposed (as it was mentioned previously), but their meaning remains the same. Therefore, the meaning of constraints (3.33)–(3.50) is the same as that for expressions (3.2)–(3.6), (3.11)–(3.16), (3.7)–(3.9) and (3.17)–(3.20), respectively.

$$\text{minimize} \quad Costs(\boldsymbol{y},\, \boldsymbol{w},\, \boldsymbol{x},\, \boldsymbol{r},\, \boldsymbol{q},\, \boldsymbol{u},\, \boldsymbol{v})$$
$$\text{subject to} \quad \sum_{i\in I}(y_{it} - y_{i,t-1}) \le n_t \qquad \forall t\in T \tag{3.33}$$
$$y_{it} \ge y_{i,t-1} \qquad \forall i\in I,\ t\in T \tag{3.34}$$
$$\sum_{i\in I} x_{ijt} = 1 \qquad \forall j\in J,\ t\in T \tag{3.35}$$
$$x_{ijt} \le y_{it} \qquad \forall i\in I,\ j\in J,\ t\in T \tag{3.36}$$
$$\sum_{k\in K} w_{ikt} = y_{it} \qquad \forall i\in I,\ t\in T \tag{3.37}$$
$$\sum_{k\in K} q^{\varphi}_{ikt} + u^{\varphi}_{i,t-1} = u^{\varphi}_{it} + \sum_{j\in J}\sum_{s\in T_t} v^{\varphi}_{ijts} \qquad \forall i\in I,\ t\in T,\ \varphi\in\Phi \tag{3.38}$$
$$q^{\varphi}_{ikt} \le M r^{\varphi}_{ikt} \qquad \forall i\in I,\ k\in K,\ t\in T,\ \varphi\in\Phi \tag{3.39}$$
$$r^{\varphi}_{ikt} \le w_{ikt} \qquad \forall i\in I,\ k\in K,\ t\in T,\ \varphi\in\Phi \tag{3.40}$$
$$u^{\varphi}_{it} \le M y_{it} \qquad \forall i\in I,\ t\in T,\ \varphi\in\Phi \tag{3.41}$$
$$v^{\varphi}_{ijtt} + \sum_{s\in T\setminus T_t} v^{\varphi}_{ijst} \le M x_{ijt} \qquad \forall i\in I,\ j\in J,\ t\in T,\ \varphi\in\Phi \tag{3.42}$$
$$\sum_{i\in I}\left( v^{\varphi}_{ijtt} + \sum_{s\in T\setminus T_t} v^{\varphi}_{ijst} \right) = \xi^{\varphi}_{jt} \qquad \forall j\in J,\ t\in T,\ \varphi\in\Phi \tag{3.43}$$
$$y_{it} \in \{0,1\} \qquad \forall i\in I,\ t\in T \tag{3.44}$$
$$w_{ikt} \in \{0,1\} \qquad \forall i\in I,\ k\in K,\ t\in T \tag{3.45}$$
$$x_{ijt} \in \{0,1\} \qquad \forall i\in I,\ j\in J,\ t\in T \tag{3.46}$$
$$r^{\varphi}_{ikt} \in \{0,1\} \qquad \forall i\in I,\ k\in K,\ t\in T,\ \varphi\in\Phi \tag{3.47}$$
$$q^{\varphi}_{ikt} \ge 0 \qquad \forall i\in I,\ k\in K,\ t\in T,\ \varphi\in\Phi \tag{3.48}$$
$$u^{\varphi}_{it} \ge 0 \qquad \forall i\in I,\ t\in T,\ \varphi\in\Phi \tag{3.49}$$
$$v^{\varphi}_{ijts} \ge 0 \qquad \forall i\in I,\ j\in J,\ t\in T,\ s\in T_t,\ \varphi\in\Phi \tag{3.50}$$

Since the problem is now formulated as a mixed-integer linear programming problem, then commercial solvers like IBM's CPLEX, FICO's Xpress or Gurobi's Gurobi Optimizer can be utilized in order to solve it.

# Chapter 4

# Computational Results

The model introduced in Section 3.4 was implemented using IBM ILOG CPLEX Optimization Studio, version 22.10. To assess its relevance and tractability, a set of instances was generated computationally. The procedures for accomplish that are detailed in Section 4.1. This section also details the implementation of these algorithms and their pre-requisites. The instances generated are briefly described in this section as well. Finally, the results obtained from the computational tests are presented and analysed in Section 4.2.

## 4.1 Data Generation

Recall the mixed-integer linear problem presented in the Section 3.4. In this model, there are a total of ten groups of parameters, where eight were already part of the two-stage stochastic linear model presented in Section 3.2. These original parameters are $n_t,\ F_{it},\ h_{it},\ C^1_{ikt},\ g_{ikt},\ a_{ikt},\ C^2_{ijt}$ and $p_{ijts}$, where $t \in T,\ i \in I,\ k \in K,\ j \in J$ and $s \in T_t$. The other two are $\Pi^\varphi$ and $\xi^\varphi_{jt}$, where $\varphi \in \Phi,\ j \in J$ and $t \in T$.

For these computational tests, it was decided that the customers' demand scenarios have the same occurring probability, that is, $\Pi^\varphi = 1/|\Phi|,\ \forall \varphi \in \Phi$. For the remainder parameters, the data generation is based in pseudo-random numbers from a discrete Uniform distribution in $\{a,\ \ldots,\ b\},\ a < b$. For each parameter, initial values for $a$ and $b$ were selected according to the table 4.1 and are from now on mentioned as *initalMin* and *initalMax*, respectively. By considering discrete Uniform distributions, it is ensured that the obtained values are integers.

Table 4.1: Initial parameters' minimum and maximum values

| Parameter | *initalMin* | *initialMax* |
|---|---|---|
| $n_t$ | 0 | $\lceil \lvert I \rvert /2 \rceil$ |
| $F_{it}$ | 10000 | 50000 |
| $C^1_{ikt}$ | 100 | 500 |
| $C^2_{ijt}$ | 100 | 500 |
| $g_{ikt}$ | 50 | 200 |
| $a_{ikt}$ | 1 | 25 |
| $h_{it}$ | 1 | 30 |
| $p_{ijts}$ | 10 | 60 |
| $\xi^\varphi_{jt}$ | 100 | 1000 |

The generation of the values $n_t,\ t \in T$, is accomplished according to Algorithm 4.1. For each period, a pseudo-random number is generated according to a discrete Uniform distribution in

$\{initalMin,\ \ldots,\ initialMax\}$, which, as stated in Table 4.1, is $\{0,\ \ldots,\ \lceil |I|/2 \rceil\}$. However, for the first period, if the possibility to open at least one DC is missing, then it is decided that one DC can be open at that period. Otherwise, given the structure of the problem, it will be impossible to solve it.

**Algorithm 4.1** Data Generator for $n_t$

**for** each $t \in T$ **do**
  $value \sim Uniform\{initalMin,\ initialMax\}$
  **if** $t = 1$ and $value = 0$ **then**
    $value \leftarrow 1$ ▷ so that there is one DC working at the first period
  **end if**
**end for**

The data generation method for $F_{it},\ h_{it},\ C^1_{ikt},\ g_{ikt},\ a_{ikt}$ and $C^2_{ijt}$ (Algorithm 4.3), although similar, it is not equal to the previous method. First, notice that all of the parameters are indexed in $t \in T$. To simplify the explanation, let the remaining indexes be known as the situational indexes from now on. For instance, the situational indexes of $C^1_{ikt}$ are $i \in I$ and $k \in K$.

Just like in the process for generating the values of $n_t$, for this group of parameters, their values will be pseudo-random numbers from a discrete Uniform distribution. However, it will not be the discrete Uniform distribution in $\{initalMin,\ \ldots,\ initialMax\}$. Instead, with the purpose of creating extra diversity, for each group of situational indexes, the values of *initalMin* and *initialMax* will be replaced by *newMin* and *newMax*, which are obtained according to the Algorithm 4.2. This algorithm is designed so that it is impossible that these new values are negative or that *newMax* is inferior to *newMin*. And its main goal is to introduce more variability into the data to be generated.

**Algorithm 4.2** New Limits Generator

**function** NEWLIMITS($oldMin,\ oldMax$)
  Generate $u_0,\ u_1,\ u_2 \sim Uniform\{0,\ 100\}$
  $v_1,\ v_2 \sim Uniform\left\{ \left\lfloor \frac{oldMin \times u_0}{100} \right\rfloor,\ \left\lceil \frac{oldMax \times u_0}{100} \right\rceil \right\}$
  $w \leftarrow 1$
  **if** $u_1 < 50$ **and** $v_1 < oldMin$ **then**
    $w \leftarrow -1$
  **end if**
  $newMin \leftarrow oldMin + w \times v_1$
  $w \leftarrow 1$
  **if** $u_2 < 50$ **and** $v_2 < oldMax - newMin$ **then**
    $w \leftarrow -1$
  **end if**
  $newMax \leftarrow oldMax + w \times v_2$
  **return** $newMin,\ newMax$
**end function**

Regarding the parameter $p_{ijts}$, the idea behind Algorithm 4.3 is used, but with a twist. Recall that $p_{ijts}$ refers to the unit revenue from DC $i \in I$ to customer $j \in J$ in period $t \in T$ to satisfy the customer's demand from period $s \in T_t$, where $T_t = \{1,\ \ldots,\ t\}$. When $t = s$, then the product is sold at the period it is requested. When $t \neq s$, the product is being sold with a delay of $t - s$ periods. For this reason, the values generated for this parameter follow the following rule: for each period that the delivery is delayed, the customer may be given a non-specified discount. Algorithm 4.4 reflects this decision.

When it comes to the generation procedure for data regarding the customers' demand, two different

**Algorithm 4.3** Data Generator for $F_{it}$, $h_{it}$, $C^1_{ikt}$, $g_{ikt}$, $a_{ikt}$ and $C^2_{ijt}$

**for** each situational index **do**<br>
    $(newMin,\ newMax) \leftarrow$ newLimits$(initialMin,\ initialMax)$<br>
    **for** each $t \in T$ **do**<br>
        $value \sim Uniform\{newMin,\ newMax\}$<br>
    **end for**<br>
**end for**

**Algorithm 4.4** Data Generator for $p_{ijts}$

**for** each $i \in I$ and $j \in J$ **do**<br>
    $(newMin,\ newMax) \leftarrow$ newLimits$(initialMin,\ initialMax)$<br>
    **for** each $t \in T$ and $s \in T_t$ **do**<br>
        **if** $t = s$ **then**<br>
            $value \sim Uniform\{newMin,\ newMax\}$<br>
        **else**<br>
            $value \sim Uniform\{newMin,\ p_{ij,t-1,s}\}$<br>
        **end if**<br>
    **end for**<br>
**end for**

possibilities were defined. In the first one, it is assumed that the demand follows an Uniform distribution and therefore the method applied for the parameters $F_{it}$, $h_{it}$, $C^1_{ikt}$, $g_{ikt}$, $a_{ikt}$ and $C^2_{ijt}$ (Algorithm 4.3) is used. Notice that $\varphi \in \Phi$ is considered to be part of the situational indexes.

In the second possibility, a Normal distribution is assumed. The expected value chosen is the average of the values obtained after applying Algorithm 4.2; the standard deviation is the absolute value of the subtraction of the chosen expected value with the average of $initialMin$ and $initialMax$. When generating the values, only those between the limits defined are accepted.

**Algorithm 4.5** Data Generator for $\xi^{\varphi}_{jt}$ – Normal Demand

**for** each $j \in J$ and $\varphi \in \Phi$ **do**<br>
    $(newMin,\ newMax) \leftarrow$ newLimits$(initialMin,\ initialMax)$<br>
    $initialMean \leftarrow \dfrac{initialMax + initialMin}{2}$<br>
    $newMean \leftarrow \dfrac{newMax + newMin}{2}$<br>
    $diff \leftarrow \mid initialMean - newMean \mid$<br>
    **for** each $t \in T$ **do**<br>
        **repeat**<br>
            $value \sim Normal(newMean,\ diff)$<br>
        **until** $newMin \leq value \leq newMax$<br>
    **end for**<br>
**end for**

The algorithms presented in this section were implemented using the C++ programming language and the Eclipse IDE. As inputs, the user only needs to indicate the dimensions of the sets $K$, $I$, $J$, $T$ and $\Phi$. The outputs are two .dat files and .txt. This means that two different instances are generated, but the only difference between them regards the costumers' demand data. In one case it is generated using the Uniform distribution and in the other using the Normal distribution, as described previously. The .txt file is only created so that a user can easily read the data. It contains both customers' demand cases.

## 4. COMPUTATIONAL RESULTS

The .dat files do not require any kind of modification in order to be read by CPLEX, given the implementation of the model from Section 3.4.

The dimensions of sets $K$, $I$, $J$, $T$ and $\Phi$ of the instances generated are listed in Table 4.2. The letters 'U' and 'N' next to the instances' number refer to the customers' demand generation process: Uniform and Normal distributions, respectively. In this table, the number of decision variables and constraints are also indicated. It is expected that the set whose dimension has the largest impact in the number of decision variables and constraints is $T$ (it is present in all of them). On the other hand, $K$ is expected to have the least impact, since it is only present in three groups of decision variables and in about a quarter of the constraints' groups.

Table 4.2: Instances' dimensions of sets and other characteristics

| **Instance** | $\|K\|$ | $\|I\|$ | $\|J\|$ | $\|T\|$ | $\|\Phi\|$ | **Decision variables** | **Constraints** |
|---|---|---|---|---|---|---|---|
| 1U, 1N | 3 | 10 | 50 | 3 | 2 | 8 040 | 13 533 |
| 2U, 2N | 3 | 10 | 50 | 3 | 5 | 17 670 | 28 833 |
| 3U, 3N | 3 | 10 | 100 | 3 | 2 | 15 540 | 25 983 |
| 4U, 4N | 3 | 10 | 100 | 6 | 3 | 70 500 | 98 466 |
| 5U, 5N | 3 | 10 | 100 | 12 | 3 | 249 000 | 304 932 |
| 6U, 6N | 3 | 10 | 250 | 6 | 3 | 174 000 | 241 566 |
| 7U, 7N | 3 | 25 | 50 | 3 | 3 | 28 125 | 45 678 |
| 8U, 8N | 3 | 25 | 50 | 6 | 5 | 144 600 | 197 706 |
| 9U, 9N | 3 | 25 | 100 | 12 | 2 | 425 400 | 524 412 |
| 10U, 10N | 3 | 25 | 250 | 3 | 3 | 133 125 | 213 078 |
| 11U, 11N | 3 | 25 | 250 | 12 | 3 | 1 545 500 | 1 864 812 |
| 12U, 12N | 3 | 50 | 100 | 12 | 5 | 2 033 400 | 2 425 812 |
| 13U, 13N | 3 | 50 | 250 | 3 | 5 | 418 350 | 654 153 |
| 14U, 14N | 5 | 10 | 50 | 3 | 2 | 8 340 | 14 073 |
| 15U, 15N | 5 | 10 | 50 | 12 | 2 | 87 360 | 110 292 |
| 16U, 16N | 5 | 10 | 50 | 12 | 5 | 208 320 | 255 372 |
| 17U, 17N | 5 | 10 | 250 | 3 | 2 | 38 340 | 63 873 |
| 18U, 18N | 5 | 10 | 250 | 3 | 3 | 53 670 | 87 813 |
| 19U, 19N | 5 | 10 | 250 | 3 | 5 | 84 330 | 135 693 |
| 20U, 20N | 5 | 10 | 250 | 6 | 3 | 174 840 | 243 126 |
| 21U, 21N | 5 | 25 | 50 | 12 | 3 | 319 200 | 393 012 |
| 22U, 22N | 5 | 25 | 100 | 3 | 2 | 39 600 | 64 953 |
| 23U, 23N | 5 | 25 | 100 | 6 | 2 | 124 200 | 174 906 |
| 24U, 24N | 5 | 25 | 100 | 6 | 5 | 286 650 | 389 556 |
| 25U, 25N | 5 | 25 | 100 | 12 | 2 | 428 400 | 529 812 |
| 26U, 26N | 5 | 25 | 100 | 12 | 3 | 626 700 | 762 912 |
| 27U, 27N | 5 | 25 | 250 | 6 | 5 | 702 900 | 946 206 |
| 28U, 28N | 5 | 50 | 50 | 3 | 2 | 41 700 | 68 553 |
| 29U, 29N | 5 | 50 | 50 | 3 | 5 | 91 650 | 146 853 |
| 30U, 30N | 5 | 50 | 50 | 6 | 3 | 184 200 | 256 806 |
| 31U, 31N | 5 | 50 | 250 | 6 | 5 | 1 405 800 | 1 883 406 |
| 32U, 32N | 5 | 50 | 250 | 12 | 5 | 5 061 600 | 6 016 812 |

Table 4.2: Instances' dimensions of sets and other characteristics

| Instance | $\|K\|$ | $\|I\|$ | $\|J\|$ | $\|T\|$ | $\|\Phi\|$ | Decision variables | Constraints |
|---|---|---|---|---|---|---|---|
| 33U, 33N | 10 | 10 | 50 | 3 | 2 | 9 090 | 15 423 |
| 34U, 34N | 10 | 10 | 50 | 6 | 5 | 62 460 | 88 986 |
| 35U, 35N | 10 | 10 | 50 | 12 | 5 | 214 920 | 267 972 |
| 36U, 36N | 10 | 10 | 100 | 3 | 5 | 36 480 | 59 643 |
| 37U, 37N | 10 | 10 | 100 | 6 | 5 | 117 960 | 164 286 |
| 38U, 38N | 10 | 25 | 50 | 6 | 5 | 156 150 | 219 756 |
| 39U, 39N | 10 | 25 | 50 | 12 | 3 | 329 700 | 412 512 |
| 40U, 40N | 10 | 25 | 100 | 6 | 2 | 127 950 | 181 656 |
| 41U, 41N | 10 | 25 | 100 | 12 | 2 | 435 900 | 543 312 |
| 42U, 42N | 10 | 25 | 250 | 3 | 5 | 214 950 | 340 353 |
| 43U, 43N | 10 | 50 | 50 | 3 | 2 | 45 450 | 75 303 |
| 44U, 44N | 10 | 50 | 50 | 3 | 3 | 63 600 | 104 403 |
| 45U, 45N | 10 | 50 | 50 | 6 | 5 | 312 300 | 437 706 |
| 46U, 46N | 10 | 50 | 100 | 3 | 2 | 82 950 | 135 753 |
| 47U, 47N | 10 | 50 | 100 | 6 | 2 | 255 900 | 361 506 |
| 48U, 48N | 10 | 50 | 250 | 6 | 5 | 1 422 300 | 1 914 906 |
| 49U, 49N | 10 | 50 | 250 | 12 | 3 | 3 119 400 | 3 772 212 |
| 50U, 50N | 10 | 50 | 250 | 12 | 5 | 5 094 600 | 6 079 812 |

Table 4.3 shows, for each set, the number of instances being considered per set dimension for a type of customers' demand generation method.

Table 4.3: Absolute frequencies of each set's dimension for each customers' demand generation method

| Set | $K$ | | | $I$ | | | $J$ | | | $T$ | | | $\Phi$ | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **Dimension** | **3** | **5** | **10** | **10** | **25** | **50** | **50** | **100** | **250** | **3** | **6** | **12** | **2** | **3** | **5** |
| **Frequency** | 13 | 19 | 18 | 18 | 17 | 15 | 19 | 16 | 15 | 19 | 16 | 15 | 16 | 14 | 20 |

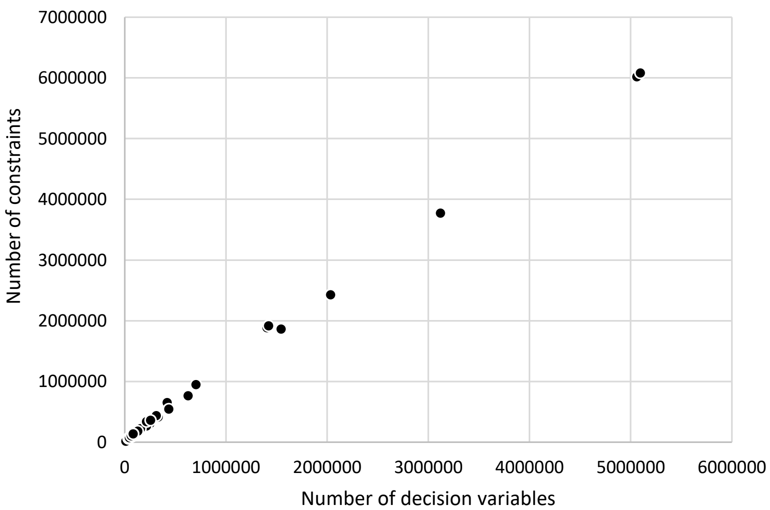



Figure 4.1: Scatter plot of the number of decision variables and number of constraints.

In order to observe the relationship between the number of decision variables and the number of constraints, these values are displayed in a scatter plot presented in Figure 4.1. It seems that the latter increases in an linear way with respect to the former, which it is not at all surprising.

## 4.2 Computational Tests

The characteristics of the computer and software used to run the one-hundred generated instances are the following ones:

**Processor:** 12th Gen Intel(R) Core(TM) i5-12500 3.00 GHz;

**Installed RAM:** 8.00 GB (7.68 GB usable);

**System type:** 64-bit operating system, x64-based processor;

**Windows edition:** Windows 10 Enterprise version 22H2;

**IBM ILOG CPLEX Optimization Studio:** version 22.1.0.

The standard parameters' values of the solver were adopted (therefore a time limit was not established). When running the computational tests, the most relevant characteristics regarding the instances, which are presented in Table 4.4, are the following ones:

**1.** whether the instance is, or not, solvable;

**2.** time it took to solve it, as given by cplex.getSolvedTime(), in seconds;

**3.** how many DCs are to be installed and when;

**4.** percentage of the customers' demand delivered on time (CDDOT (%)) for each demand scenario.

Instead of reporting on the number of DCs to be installed and when, the third column of Table 4.4 presents a metric, named 'DCs ratio (%)', which is the percentage ratio between the number of DCs opened and the number of potential locations available.

All the instances solved to proven optimality registered negative optimal objective function values, which indicates a profit.

Table 4.4: Main results

| Instance | Time (seconds) | DCs ratio (%) | CDDOT (%) |
|---|---|---|---|
| 1U | 0.312 | 70 | 94.78, 95.57 |
| 1N | 0.282 | 70 | 95.93, 94.70 |
| 2U | 0.766 | 60 | 93.32, 95.18 95.71, 94.59, 95.47 |
| 2N | 0.64 | 60 | 92.81, 95.97 95.00, 94.58, 95.26 |
| 3U | 0.688 | 60 | 89.24, 88.74 |
| 3N | 0.797 | 60 | 95.76, 94.61 |
| 4U | 1.781 | 100 | 94.06, 93.76, 93.98 |
| 4N | 1.813 | 100 | 93.97, 93.57, 94.19 |

Table 4.4: Main results

| Instance | Time (seconds) | DCs ratio (%) | CDDOT (%) |
|---|---|---|---|
| 5U | 13.594 | 100 | 90.03, 89.56, 90.73 |
| 5N | 10.656 | 100 | 89.74, 89.55, 90.55 |
| 6U | 36.328 | 100 | 94.81, 94.37, 94.32 |
| 6N | 328 | 100 | 94.62, 92.97, 94.87 |
| 7U | 2.828 | 40 | 98.47, 97.49, 98.62 |
| 7N | 1.687 | 44 | 99.47, 98.78, 99.40 |
| 8U | 5.594 | 76 | 91.47, 91.93, 91.50, 91.05, 91.84 |
| 8N | 5.204 | 72 | 91.79, 91.66, 92.26, 92.61, 91.74 |
| 9U | 35.125 | 96 | 94.08, 94.01 |
| 9N | 36.906 | 96 | 94.03, 93.55 |
| 10U | 15.094 | 44 | 93.72, 94.16, 94.27 |
| 10N | 10.813 | 44 | 95.33, 95.46, 95.38 |
| 11U | Error 1001: Out of memory | | |
| 11N | 295.047 | 100 | 94.93, 94.81, 95.05 |
| 12U | 571.828 | 50 | 94.54, 94.31, 94.14, 93.93, 94.64 |
| 12N | 508.937 | 50 | 94.95, 94.74, 94.77, 94.56, 94.83 |
| 13U | 41.515 | 60 | 88.82, 89.54, 88.07, 88.49, 88.90 |
| 13N | 41.719 | 60 | 88.54, 88.47, 87.22, 88.58, 88.87 |
| 14U | 0.406 | 50 | 83.52, 84.74 |
| 14N | 0.437 | 50 | 83.64, 84.62 |
| 15U | 3.812 | 100 | 92.01, 92.42 |
| 15N | 3.578 | 100 | 92.82, 92.41 |
| 16U | 10.672 | 100 | 93.07, 92.82, 92.64, 92.51, 91.88 |
| 16N | 10.562 | 100 | 93.17, 92.24, 92.34, 92.43, 92.03 |
| 17U | 4.516 | 90 | 96.31, 96.26 |
| 17N | 2.906 | 90 | 96.75, 96.61 |
| 18U | 3.672 | 90 | 96.15, 95.92, 96.36 |
| 18N | 7.406 | 90 | 96.08, 96.01, 96.87 |
| 19U | 6.36 | 90 | 93.69, 94.02, 93.74, 93.71, 93.66 |
| 19N | 7.672 | 80 | 91.33, 91.84, 91.60, 91.97, 91.58 |
| 20U | 18.516 | 100 | 90.14, 90.46, 90.83 |
| 20N | 21.078 | 100 | 90.11, 90.46, 90.78 |
| 21U | 6.86 | 100 | 94.15, 94.69, 94.17 |
| 21N | 7.312 | 100 | 94.73, 93.69, 93.98 |
| 22U | 1.203 | 52 | 96.47, 96.84 |
| 22N | 1.547 | 52 | 96.44, 96.85 |
| 23U | 2.437 | 100 | 95.86, 95.14 |
| 23N | 2.047 | 100 | 95.23, 94.85 |
| 24U | 23.328 | 92 | 84.99, 84.15, 83.73, 83.39, 84.51 |
| 24N | 25.516 | 92 | 85.28, 84.74, 84.21, 83.25, 83.70 |
| 25U | 7.719 | 40 | 94.75, 95.03 |

Table 4.4: Main results

| Instance | Time (seconds) | DCs ratio (%) | CDDOT (%) |
|---|---|---|---|
| 25N | 7.032 | 40 | 94.61, 94.96 |
| 26U | 10.844 | 40 | 94.73, 94.57, 94.30 |
| 26N | 11 | 40 | 94.79, 94.56, 94.47 |
| 27U | 64.375 | 100 | 94.32, 94.58, 94.46, 95.10, 94.37 |
| 27N | 54.484 | 100 | 87.76, 87.35, 87.48, 87.75, 87.28 |
| 28U | 1.032 | 34 | 97.84, 96.95 |
| 28N | 0.813 | 34 | 97.32, 97.05 |
| 29U | 3.719 | 34 | 97.76, 96.56, 96.86, 97.91, 97.78 |
| 29N | 3.844 | 34 | 97.29, 96.56, 96.83, 97.18, 97.13 |
| 30U | 12.891 | 52 | 94.84, 94.01, 94.08 |
| 30N | 13.844 | 52 | 95.70, 95.32, 95.12 |
| 31U | Error 1001: Out of memory | | |
| 31N | Error 1001: Out of memory | | |
| 32U | Error 1001: Out of memory | | |
| 32N | Error 1001: Out of memory | | |
| 33U | 0.515 | 30 | 97.53, 99.05 |
| 33N | 0.453 | 30 | 98.00, 98.73 |
| 34U | 2.781 | 90 | 97.43, 96.60, 96.29, 96.33, 96.15 |
| 34N | 2.765 | 90 | 97.12, 96.24, 96.53, 96.24, 95.96 |
| 35U | 9.36 | 100 | 93.38, 92.84, 92.36, 92.21, 91.58 |
| 35N | 9.5 | 100 | 92.74, 92.34, 92.62, 92.11, 91.84 |
| 36U | 2.016 | 100 | 95.86, 95.51, 96.13, 96.48, 95.71 |
| 36N | 3.187 | 90 | 97.44, 97.69, 97.86, 97.80, 97.94 |
| 37U | 5.89 | 100 | 95.15, 94.99, 95.11, 95.15, 94.91 |
| 37N | 6.093 | 100 | 95.00, 95.39, 95.27, 95.70, 95.19 |
| 38U | 4.812 | 84 | 98.15, 98.60, 98.69, 98.28, 98.24 |
| 38N | 3.797 | 84 | 97.86, 97.93, 97.68, 98.14, 98.18 |
| 39U | 17.797 | 100 | 97.30, 97.76, 97.85 |
| 39N | 20.734 | 100 | 97.81, 97.81, 97.85 |
| 40U | 2.969 | 92 | 96.65, 97.21 |
| 40N | 3.079 | 92 | 97.07, 96.79 |
| 41U | 19.187 | 100 | 95.94, 95.37 |
| 41N | 15.281 | 100 | 95.89, 95.98 |
| 42U | 26.125 | 88 | 95.90, 96.16, 95.81, 96.37, 95.75 |
| 42N | 19.438 | 92 | 95.31, 95.51, 95.37, 95.88, 96.11 |
| 43U | 1.485 | 42 | 97.12, 96.70 |
| 43N | 1.312 | 42 | 96.64, 94.91 |
| 44U | 2 | 40 | 95.70, 97.39, 95.60 |
| 44N | 2.094 | 44 | 96.25, 97.86, 96.18 |
| 45U | 7.797 | 58 | 97.66, 98.39, 97.19, 97.77, 96.93 |
| 45N | 5.703 | 56 | 98.01, 97.04, 97.72, 97.06, 96.95 |
| 46U | 4.609 | 46 | 97.36, 97.26 |

Table 4.4: Main results

| Instance | Time (seconds) | DCs ratio (%) | CDDOT (%) |
|---|---|---|---|
| 46N | 4.516 | 44 | 86.56, 85.26 |
| 47U | 17.922 | 70 | 96.98, 97.56 |
| 47N | 16.125 | 68 | 95.55, 96.08 |
| 48U | 246.422 | 100 | 84.24, 85.17, 85.40, 85.09, 85.18 |
| 48N | 227.797 | 100 | 84.16, 84.76, 84.97, 84.83, 84.68 |
| 49U | Error 1001: Out of memory | | |
| 49N | Error 1001: Out of memory | | |
| 50U | Error 1001: Out of memory | | |
| 50N | Error 1001: Out of memory | | |

The data in this table reveals that the problem's instances can be quickly solved by the chosen solver as long as there is available memory. In fact, the instance that took the longest time (instance 12U) was solved in less than ten minutes and had 2 033 400 decision variables and 2 425 812 constraints. On the other hand, from instances 11U and 11N, only one was solvable and yet they have the same number of decision variables (1 545 500) and constraints (1 864 812). The only difference in the parameters of these two instances were the values for the customers' demand.

Among the nine instances that were not solved due to lack of available memory (instances 11U, 31U, 31N, 32U, 32N, 49U, 49N, 50U and 50N), all of them had 250 as the dimension for the set of customers. Eight of them had 50 as the dimension for the set of possible location for DCs, while the remaining one had 25. Seven of them comprised 12 time periods, while the last two had six. Six of them had five as the dimension for the set of demand scenarios, while the remaining three had three. Finally, four had consisted of ten suppliers, four others had five, while the last one had three.

In the scatter plots from Figures 4.2 and 4.3, it is easily observed that the majority of the instances that were solved took less than a minute and half, comprising less than 500 000 decision variables and less than 1 000 000 constraints. It is not surprising that larger instances tend to take longer to be solved, but this is not a rule. For example, instance 6N took 328 seconds to be solved, but instance 6U, which has the same number of decision variables and constraints, was solved in 36.328 seconds.

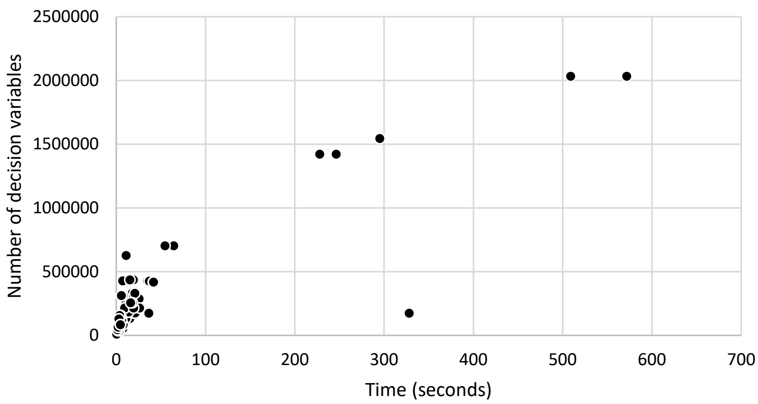



Figure 4.2: Scatter plot of the time it took to solve the instance by the number of decision variables.

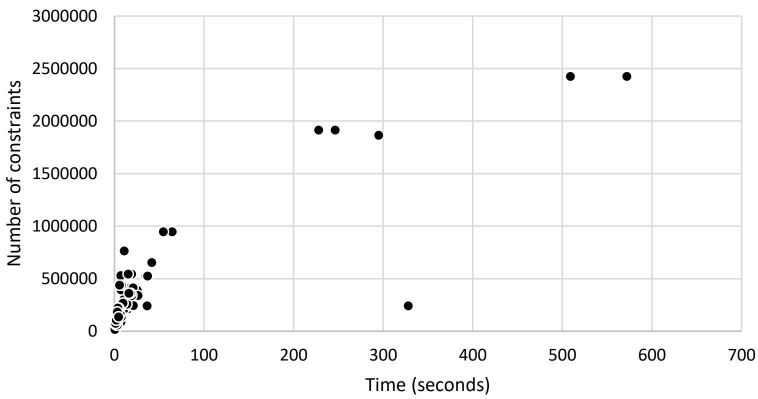



Figure 4.3: Scatter plot of the time it took to solve the instance by the number of constraints.

Regarding the values for the DCs ratio (%), from Table 4.4, it is observed that for the majority of instances, the results dictate that at least half of the possible locations for DCs should be open. In fact, 50 out of the 91 solved instances showed a DC ratio (%) of at least 75%. Moreover, for 30 instances that value is 100%. It is possible that these results are highly affected by the values for the fixed cost of locating a DC (denoted by $F_{it}$), particularly when compared to the total profit of the sold products. To investigate this further, it was decided to run every instance again, but the original values for $F_{it}$ were multiplied by 100. The results from the new tests are presented in Table 4.5 and analysed afterwards.

Note that if it is not desirable to open DCs in every possible location, then a new constraint is required. Constraint (4.1) limits the total number of DCs to be installed in the planning horizon, where the parameter $N$ is the value for that limit. $N$ must be a positive integer lower than the number of available locations for DCs ($|I|$). This constraint, if needed, should be added as it is to the first-stage models discussed in Sections 3.2 and 3.3 and to the linear programming model proposed in Section 3.4.

$$\sum_{i \in I} y_{i|T|} \leq N \tag{4.1}$$

The values for CDDOT (%) for each demand scenario from Table 4.4 are high. This is in some sense positive, since it means that the majority of the total customers' demand under each demand scenario was delivered on time. Overall, the lowest percentage registered was 83.25% for the fourth demand scenario in instance 24N and the highest was 99.47% for the first demand scenario in instance 7N. However, it does not imply that every customer in each demand scenario had most of their demand be delivered on time. For this reason, it might be beneficial to know these values by customer, even though they are not displayed here.

Table 4.5: Main results, where instead of $F_{it}$ it was used $100 \times F_{it}$

| Instance | Time (seconds) | DCs ratio (%) | CDDOT (%) |
|---|---|---|---|
| 1U | 0.172 | 10 | 87.14, 85.54 |
| 1N | 0.172 | 10 | 87.97, 87.22 |
| 2U | 0.36 | 10 | 85.25, 86.86, 87.76, 86.68, 85.45 |
| 2N | 0.407 | 10 | 86.46, 86.67, 86.17, 85.13, 86.35 |
| 3U | 0.594 | 20 | 87.45, 86.89 |
| 3N | 0.609 | 20 | 87.00, 86.21 |
| 4U | 5.375 | 100 | 47.95, 49.29, 50.92 |
| 4N | 6.906 | 20 | 88.07, 87.02, 87.64 |

Table 4.5: Main results, where instead of $F_{it}$ it was used $100 \times F_{it}$

| Instance | Time (seconds) | DCs ratio (%) | CDDOT (%) |
|---|---|---|---|
| 5U | 14.11 | 40 | 87.34, 85.79, 85.98 |
| 5N | 11.297 | 40 | 86.77, 86.53, 86.20 |
| 6U | 42.125 | 100 | 44.72, 44.08, 44.28 |
| 6N | 35.813 | 100 | 44.70, 44.10, 44.77 |
| 7U | 1.921 | 4 | 88.16, 90.36, 89.58 |
| 7N | 1.579 | 4 | 88.97, 88.29, 87.62 |
| 8U | 9.906 | 8 | 79.29, 77.71, 78.02, 78.42, 78.23 |
| 8N | 10.359 | 8 | 80.00, 78.56, 79.06, 79.22, 79.24 |
| 9U | 42.531 | 24 | 86.16, 86.74 |
| 9N | 41.704 | 24 | 86.75, 87.59 |
| 10U | 15.328 | 44 | 91.95, 92.13, 90.83 |
| 10N | 13.875 | 44 | 91.18, 92.55, 91.50 |
| 11U | 489.094 | 100 | 74.67, 74.83, 75.00 |
| 11N | 633.687 | 100 | 74.58, 74.57, 75.23 |
| 12U | 1824.406 | 20 | 92.69, 93.04, 93.01, 92.55, 92.79 |
| 12N | 1487.828 | 20 | 92.36, 92.26, 92.30, 92.04, 92.42 |
| 13U | 118.422 | 10 | 91.41, 91.36, 90.63, 92.02, 90.83 |
| 13N | 171.484 | 10 | 91.39, 91.64, 90.82, 91.29, 91.21 |
| 14U | 0.328 | 10 | 70.49, 71.39 |
| 14N | 0.329 | 10 | 69.02, 70.68 |
| 15U | 2.875 | 20 | 81.66, 81.63 |
| 15N | 3.062 | 20 | 81.57, 81.77 |
| 16U | 16.015 | 20 | 82.40, 84.07, 81.70, 81.85, 81.84 |
| 16N | 14.579 | 20 | 81.74, 82.63, 81.00, 81.86, 80.80 |
| 17U | 3.813 | 30 | 89.36, 89.00 |
| 17N | 3.656 | 30 | 87.80, 89.22 |
| 18U | 4.968 | 30 | 74.12, 73.11, 72.65 |
| 18N | 5.719 | 30 | 89.19, 89.80, 89.23 |
| 19U | 7.875 | 30 | 52.02, 53.33, 51.65, 51.33, 52.97 |
| 19N | 12.766 | 30 | 88.37, 89.28, 89.32, 89.15, 88.57 |
| 20U | 21.625 | 50 | 68.83, 68.17, 69.01 |
| 20N | 26.484 | 50 | 84.98, 85.61, 85.29 |
| 21U | 25.297 | 16 | 89.24, 89.61, 88.26 |
| 21N | 25.75 | 16 | 88.34, 88.90, 87.75 |
| 22U | 2.312 | 8 | 94.28, 93.10 |
| 22N | 1.906 | 8 | 95.19, 94.66 |
| 23U | 6.141 | 16 | 84.70, 83.49 |
| 23N | 8.422 | 16 | 86.65, 87.72 |
| 24U | 38.844 | 12 | 88.62, 87.45, 88.33, 88.65, 87.91 |
| 24N | 49.719 | 12 | 88.26, 87.34, 88.43, 88.77, 88.35 |
| 25U | 7.86 | 20 | 92.11, 91.08 |

Table 4.5: Main results, where instead of $F_{it}$ it was used $100 \times F_{it}$

| Instance | Time (seconds) | DCs ratio (%) | CDDOT (%) |
|---|---|---|---|
| 25N | 8.688 | 20 | 91.33, 91.25 |
| 26U | 13.109 | 20 | 91.45, 91.64, 91.44 |
| 26N | 12.015 | 20 | 91.32, 91.89, 91.46 |
| 27U | 197.766 | 48 | 88.66, 88.45, 88.85, 89.56, 88.65 |
| 27N | 208.735 | 36 | 88.10, 88.59, 89.29, 89.30, 88.94 |
| 28U | 3.484 | 6 | 92.15, 92.34 |
| 28N | 3.313 | 4 | 96.52, 97.83 |
| 29U | 16.297 | 4 | 97.18, 98.03, 98.66, 98.03, 98.05 |
| 29N | 17.844 | 4 | 97.42, 98.22, 97.51, 98.37, 97.06 |
| 30U | 29.094 | 6 | 92.97, 92.73, 91.59 |
| 30N | 34.171 | 6 | 92.56, 92.24, 91.83 |
| 31U | 2119.562 | 16 | 90.39, 91.12, 91.60, 91.20, 91.04 |
| 31N | 2081.844 | 16 | 91.24, 91.44, 91.10, 90.91, 91.28 |
| 32U | Error 1001: Out of memory | | |
| 32N | Error 1001: Out of memory | | |
| 33U | 0.328 | 10 | 96.16, 95.82 |
| 33N | 0.359 | 10 | 95.60, 96.43 |
| 34U | 5.859 | 20 | 85.13, 83.42, 83.98, 83.80, 83.65 |
| 34N | 4.625 | 20 | 63.51, 63.52, 61.64, 61.44, 60.51 |
| 35U | 27.656 | 60 | 86.67, 86.88, 86.07, 86.46, 85.68 |
| 35N | 22.406 | 60 | 83.12, 83.42, 83.63, 83.32, 82.56 |
| 36U | 2.485 | 20 | 94.69, 93.36, 94.31, 94.75, 92.63 |
| 36N | 2.312 | 20 | 93.49, 93.62, 93.64, 94.35, 92.51 |
| 37U | 11.093 | 40 | 79.40, 79.62, 78.62, 79.84, 79.15 |
| 37N | 11.797 | 30 | 90.63, 90.84, 90.28, 91.29, 90.81 |
| 38U | 8.969 | 8 | 91.86, 92.11, 92.84, 92.66, 92.41 |
| 38N | 9.375 | 8 | 92.06, 92.33, 93.25, 93.39, 92.38 |
| 39U | 43.203 | 24 | 68.17, 69.06, 67.56 |
| 39N | 41.046 | 16 | 90.65, 90.82, 91.58 |
| 40U | 7.391 | 16 | 93.55, 95.13 |
| 40N | 5.875 | 16 | 93.59, 94.82 |
| 41U | 38.578 | 32 | 90.72, 89.66 |
| 41N | 35.343 | 44 | 90.45, 90.37 |
| 42U | 88.531 | 40 | 93.55, 94.22, 94.21, 93.69, 94.10 |
| 42N | 118.906 | 40 | 93.50, 93.64, 93.72, 93.73, 93.15 |
| 43U | 2.36 | 4 | 90.12, 88.76 |
| 43N | 2.359 | 4 | 91.18, 90.58 |
| 44U | 5.297 | 4 | 89.46, 93.00, 92.52 |
| 44N | 5.328 | 4 | 91.82, 91.40, 90.50 |
| 45U | 149.843 | 8 | 90.93, 92.45, 89.84, 91.20, 91.87 |
| 45N | 95.531 | 10 | 77.17, 77.40, 74.75, 77.39, 76.06 |
| 46U | 16 | 8 | 90.04, 89.06 |

Table 4.5: Main results, where instead of $F_{it}$ it was used $100 \times F_{it}$

| Instance | Time (seconds) | DCs ratio (%) | CDDOT (%) |
|---|---|---|---|
| 46N | 9.688 | 6 | 90.39, 90.86 |
| 47U | 36.469 | 10 | 91.99, 91.90 |
| 47N | 43.266 | 10 | 92.19, 91.94 |
| 48U | 922.031 | 26 | 74.52, 74.49, 75.16, 75.47, 74.26 |
| 48N | 2460.703 | 14 | 95.09, 94.55, 94.51, 95.05, 94.52 |
| 49U | Error 1001: Out of memory | | |
| 49N | Error 1001: Out of memory | | |
| 50U | Error 1001: Out of memory | | |
| 50N | Error 1001: Out of memory | | |

By comparing the data in Tables 4.4 and 4.5, the differences are noticeable. First, it was possible to solve instances 11U, 31U and 31N (before, the result was 'Error 1001: Out of memory'). In summary, in the second batch of tests, instances seem to take longer to be solved and to have lower values for both DCs ratio (%) and CDDOT (%) for each demand scenario.

Out of the 91 instances that were solved in both situations, in the second run of tests, 73 took longer to be solved. In 77 instances, the difference between the run times are under one minute, while in only six instances, the difference is greater than five minutes.

In Figure 4.4, it is observed that in both situations, most of the instances were solved in less than a minute, but only in the second run of tests there were instances taking longer than ten minutes to be solved. In fact, instances 48N, 31U and 31N were the ones that took the longest to solve, taking between 34 and 42 minutes, while instances 1U and 1N were the ones that took the least time to be solved (0.172 seconds).

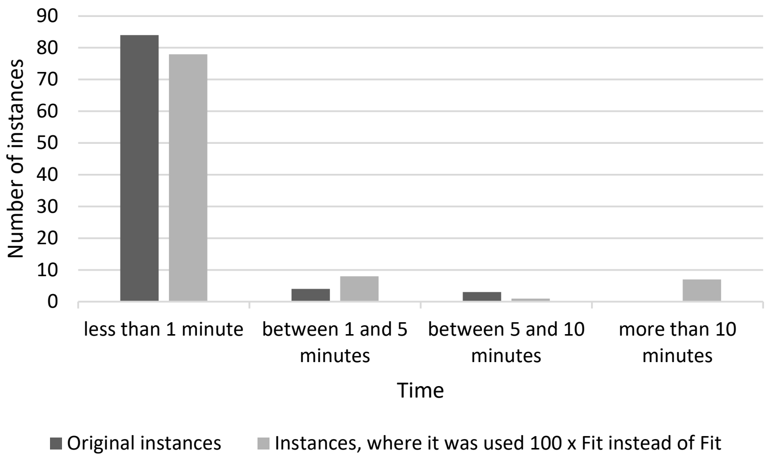



Figure 4.4: Bar graph of the time it took to solve the instances by their typology.

As seen before and as expected, the larger instances are the ones that tend to take longer to be solved, as seen in Figures 4.5 and 4.6. A large majority of instances were solved in less than five minutes and had at most 500 000 decision variables and 1 000 000 constraints. Among the eight instances that took longer than five minutes to be solved, six had 50 possible locations for DCs and five demand scenarios, while the remaining two had, respectively, 25 and three. Six had 250 customers, while the other two had 100. Four had 12 time periods and three suppliers, while the remaining four had six time periods. Out of these four instances, half had ten suppliers, while the other half had five.

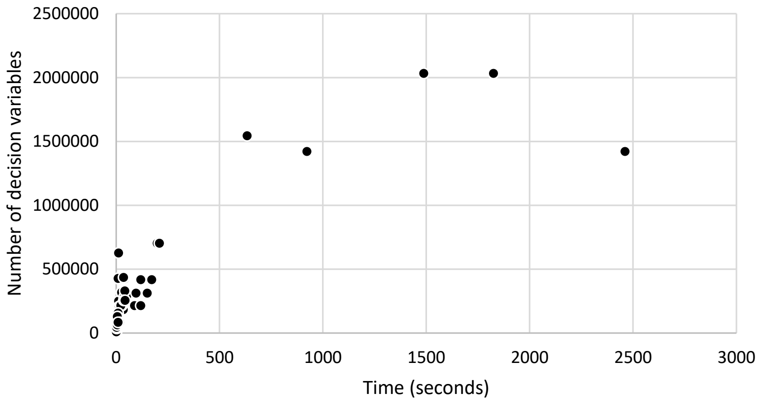



Figure 4.5: Scatter plot of the time it took to solve the instance and the number of decision variables.

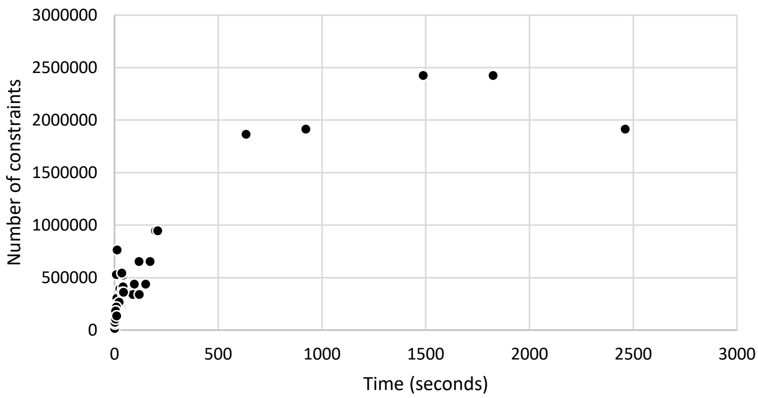



Figure 4.6: Scatter plot of the time it took to solve the instance and the number of constraints.

Regarding the DCs ratio (%) registered for the new tests, as hypothesised previously, they are lower than those registered in Table 4.4. In the first group of tests, the majority of instances (52.75%) had DC ratios in the interval ]80, 100], while ]0, 20] is the range with the majority of instances from the new group of tests (67.02%). This is a significant difference. Figure 4.7 details the variations between these groups.

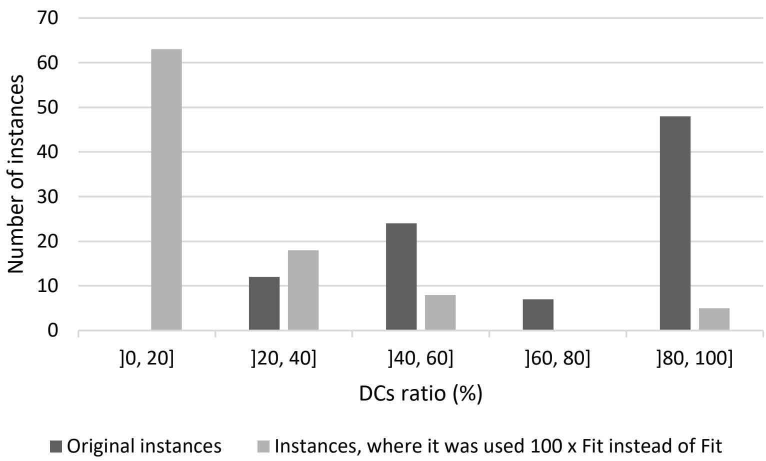



Figure 4.7: Bar graph of the DCs ratios (%) by the instances' typology.

Given that the DCs ratio (%) registered for the new tests is lower than the original tests, it is expected to see some decrease in the values for CDDOT (%) for each demand scenario. Despite the fact that this is what occurs, they remain generally high. In fact, only 14 out of the 94 solved instances have scenarios with CDDOT percentages lower than 75%, and three of them have scenarios with CDDOT percentages lower than 50%. The lowest percentage registered is 44.08% from the second scenario in instance 6U and 98.66% from the third scenario in instance 29U is the highest one. In the results from Table 4.4, in 85.7% of the instances, the worst scenario had a CDDOT (%) higher than 90%. In the results from the new tests (in Table 4.5), that percentage decreases to 41.5%. Nonetheless, in 81.9% of the instances, from the new tests, the worst scenario had a CDDOT (%) higher than 80%.

In the majority of the instances, the CDDOT percentages from Table 4.5 decreased for all scenarios, when compared to those from Table 4.4. However, instances 13U, 13N, 24U, 24N, 46N and 48N showed an increase for all scenarios, while only some scenarios from instances 28N, 29U and 29N showed an increase (and the remaining scenarios a decrease).

In Figure 4.8, it is possible to observe the differences between the CDDOT (%) from all scenarios given the group of tests. The majority demand scenarios from the first group of tests had a CDDOT (%) of at least 90%, while the majority from the second group had between 85% and 95%.

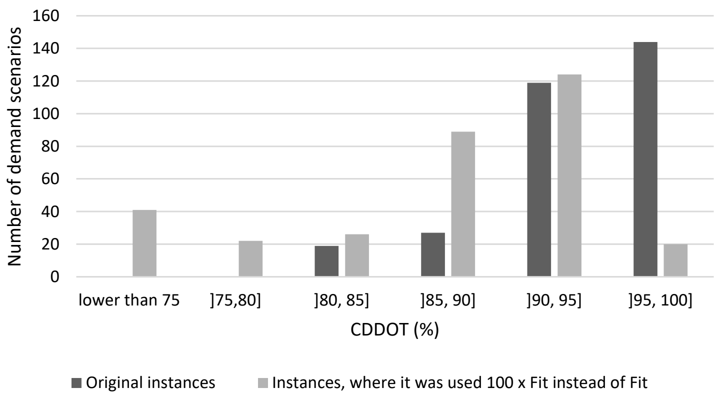



Figure 4.8: Bar graph of the CDDOT (%) of all demand scenarios by their instances' typology.

In short, the linear programming model proposed in Section 3.4 works well as it was expected. It was also seen that the values for $F_{it}$ affect primarily the percentage of DCs ratio, but the percentages from each demand scenario's CDDOT as well.

# Chapter 5

# Conclusions and Future Work

In this chapter, a brief summary on work done in this dissertation is presented in Section 5.1 as well as the main conclusions. In Section 5.2, some ideas for future work are identified.

## 5.1 Summary and Conclusions

In this dissertation, a two-stage stochastic linear programming model was proposed to formulate the *Multi-Period Stochastic Location-Inventory Problem*. In the first-stage model, the location and allocation decisions are addressed, while the decisions from the second-stage model regard the inventory management aspect of the problem. A two-stage stochastic linear programming model was also proposed to formulate a variant of the *Multi-Period Stochastic Location-Inventory Problem*, which added the concept of lead times between suppliers and DCs.

A mixed-integer linear programming model was proposed based on the concept of demand scenarios, which intends to capture the uncertainty of customers' demand. This is the basis of the proposed solution approach.

For the purpose of performing computational tests, a set of instances was computationally generated. Afterwards, computational tests were run through the use of a commercial solver. Later, a second batch of computational tests were performed by only modifying the values for the fixed cost of locating a DC at some site of all generated instances (the original values were multiplied by 100).

In an effort to evaluate the quality of the solving approach, a few characteristics and metrics were chosen. Namely, the number of decision variables and constraints, the time taken to obtain the optimal solution, the percentage ratio between the number of DCs to open and the number of potential locations available (known as DCs ratio (%)), and, at last, the percentage of customers' demand delivered on time (known as CDDOT (%)) for each demand scenario.

Most instances were solved in a considered suitable time, which led not to impose a maximum time limit in the solver settings. In fact, in the grand majority of computational tests, it took less than a minute to obtain the optimal solution.

Only in 15, out of the total 200 computational tests, it was not possible to find the optimal solution. And in those 15, the lack of memory space was the reason, taking into account these were performed in a computer with 8.00 GB of RAM. Out of the 15, nine were from the first out of two batches, while the remaining ones were from the second. Out of those 15, 12 involved the six largest instances in both number of decision variables (higher than 3 000 000) and number of constraints (higher than 3 500 000).

The results obtained suggest than the value of the fixed cost for locating a DC has a high impact on the DCs ratio (%) metric. In the second batch of tests, which had higher values for the group of parameters, the DCs ratios (%) were significantly lower than those from the first batch. In fact, excluding the tests that failed to find the optimal solution, the DC ratios in 52.75% of the tests from the first batch were between 80% and 100%, while in 67.02% of the tests from the second batch were between 0% and 20%.

The same level of impact was not observed in the CDDOT (%) metric for each demand scenario. While generally, these values decreased slightly, it was not a rule. Excluding the tests that failed to find the optimal solution, only 14.89% of the tests from the second batch had scenarios with CDDOT lower than 75%, while in first batch, not a single scenario registered a CDDOT lower than 75%.

The results that were highlighted here suggest that the solution approach chosen can be said to be good.

## 5.2 Future Work

As previously seen in Chapters 1 and 2, research concerning optimisation models applied to the domain of Logistics still is extremely relevant and continues to grasp the attention of both academics and professionals.

In the *Multi-Period Stochastic Location-Inventory Problem* studied in this dissertation, some of its possible variants are quite obvious. The two-stage stochastic linear model proposed in Section 3.2 is a first step, while the lead time extension proposed in Section 3.3 is the second step. The variants of this problem intend to capture details that have yet to be taken into consideration and represent the next steps.

For instance, there could be multiple variants dealing with the concept of capacity. Namely, the suppliers' capacity to provide the DCs, the DCs' capacity to hold inventory and to supply their customers. Under the same rationale, there could be variants where the DCs' capacities are modular, as it was considered, for instance, in the paper by Correia and Melo (2017).

The problem studied follows the assumption that initially there are no DCs working. Instead, there could be initially some DCs operating (i.e., it is already established an initial supply chain network), of which some could be afterwards closed (or changed, in the case that modular capacities are considered). Meanwhile, new DCs may be installed (and later be adapted, if their capacities are modular). Models from Correia and Melo (2016) and Correia and Melo (2017) dealt with this situation, which intends to improve the supply chain network in effect.

These are just a few examples of possible extensions of the *Multi-Period Stochastic Location-Inventory Problem* defined in this dissertation. Nonetheless, it is not only with respect to modelling that further work should be done. In fact, more research should be done regarding solving the model presented. This will be particular helpful to solve the original problem as well as its variants, since they will be more complex and thus likely be harder to solve. It is also expected that new solving approaches would be able to solve larger instances.

Alongside further development on new solving approaches, more computational tests should be performed. In this dissertation, fabricated instances were used. However, in order to better assess the quality of the solving approaches, it would be most helpful to use instances from real-world situations on the computational tests. In the case that fabricated instances must be used, the method applied in this dissertation for that purpose should be improved. For instance, for each parameter, a value $A$ could be generated for the situational indexes at the first period. Then, for the remaining periods, their values could be generated based on a percentage change to that value $A$ (or to the value of the previous period). Notwithstanding, even with computationally generated instances, an innumerable quantity of computational tests could be performed by fixing all parameters except one.

Lastly, besides the metrics that were considered in this dissertation (identified in Section 5.1), there are others that could be potentially interesting to known as well. For instance, for the orders that were not satisfied on time for each demand scenario, it could be valuable to known the time average between the moment an order is satisfied and the moment that same order was placed. This would give some insight of the tardiness average and could be compared to average of all orders. The percentage of the customers' demand not delivered on time for each demand scenario seen by the amount of delay as well as the ratio between the number of customers with back orders and the total number of customers for each demand scenario are other metrics that could be worthy to know. Still on the subject of lateness, for each demand scenario, it could be relevant to know the customers with the lowest and highest percentage of demand delivered on time as well as these values.

# References


Amiri-Aref, M., Klibi, W., & Babai, M. Z. (2018). The multi-sourcing location inventory problem with stochastic demand. *European Journal of Operational Research*, *266*(1), 72–87. https://doi.org/10.1016/j.ejor.2017.09.003

Correia, I., & Melo, T. (2016). Multi-period capacitated facility location under delayed demand satisfaction. *European Journal of Operational Research*, *255*(3), 729–746. https://doi.org/10.1016/j.ejor.2016.06.039

Correia, I., & Melo, T. (2017). A multi-period facility location problem with modular capacity adjustments and flexible demand fulfillment. *Computers & Industrial Engineering*, *110*, 307–321. https://doi.org/10.1016/j.cie.2017.06.003

Daskin, M. S., Coullard, C. R., & Shen, Z.-J. M. (2002). An inventory-location model: Formulation, solution algorithm and computational results. *Annals of Operations Research*, *110*(1), 83–106. https://doi.org/10.1023/A:1020763400324

Hillier, F. S., & Lieberman, G. J. (2015). *Introduction to operations research* (10th ed.). McGraw-Hill.

Nickel, S., & da Gama, F. S. (2019). Multi-period facility location. In G. Laporte, S. Nickel, & F. S. da Gama (Eds.), *Location science* (2nd ed., pp. 303–326). Springer.

Sauvey, C., Melo, T., & Correia, I. (2020). Heuristics for a multi-period facility location problem with delayed demand satisfaction. *Computers & Industrial Engineering*, *139*, 106171. https://doi.org/10.1016/j.cie.2019.106171

Shen, Z.-J. M., Coullard, C., & Daskin, M. S. (2003). A joint location-inventory model. *Transportation Science*, *37*(1), 40–55. https://doi.org/10.1287/trsc.37.1.40.12823

Shu, J., Teo, C.-P., & Shen, Z.-J. M. (2005). Stochastic transportation-inventory network design problem. *Operations Research*, *53*(1), 48–60. https://doi.org/10.1287/opre.1040.0140

Snyder, L. V., & Shen, Z.-J. M. (2019). *Fundamentals of supply chain theory* (2nd ed.). Wiley.