STUDIUM UNIVERSITATIS BERGOMENSIS
UNIVERSITY OF BERGAMO

Doctor of Philosophy in
ANALYTICS FOR ECONOMICS AND BUSINESS
XXX Cycle

Optimization Models in Logistics

Co–ordinator:
Marida Bertocchi †
Adriana Gnudi

Supervisors:
Francesca Maggioni
Luca Bertazzi

Candidate:
Sarem Deilami Moezi
Candidate ID: 1032923

To all my family,

who even in the distance made me feel their love and support.

To all those people who started as colleagues, classmates or just acquaintances,

who became friends, and now are my international family.

THANK YOU!!!

Declaration of Authorship

I, Sarem Deilami Moezi, declare that this thesis titled, Optimization Models in Logistics, and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a research degree at this University.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Signed:

Date:

Acknowledgements

It has been a long journey, full of stressful moments, but with many more joys and achievements. Before this chapter of my life comes to its end, looking forward to begin a new one, I would like to thank

Prof. Marida Bertocchi †, for her support for my PhD studies in Bergamo, Italy. Her memory will be with me always.

My supervisors, Prof. Francesca Maggioni and Prof. Luca Bertazzi, for blindly accepted me in their research group and for letting me learn from their passion, criticism, and dedication always to want to improve.

Prof. Vittorio Moriggia, for his support and brilliance in the lab.

The review committee of the thesis, Prof. Daniela Ambrosino and Prof. Guido Perboli, for their valuable contribution and dedication in reviewing this work.

All my friends, for making memorable the whole experience of living in Italy.

My mother Shahnaz and my father Alireza, for all your understanding and support in my decision to spend 3 long years away from you, making me always feel your presence.

SAREM

STUDIUM UNIVERSITATIS BERGOMENSIS

UNIVERSITY OF BERGAMO

Abstract

Optimization Models in Logistics

Sarem Deilami Moezi

In this thesis, we study three problems in logistics: *Single Link Shipping*, *Vendor-Managed Inventory* (VMI) and *Scheduling*. Firstly, we review the single link shipping problem in the case of multi-product and multi-modal shipping with given sets of frequencies and lead times. The contribution is in the single link case, that is due to the introduction of different sets of frequencies for each type of vehicles and to the introduction of lead times. Secondly, we introduce a two-echelon vendor-buyer supply chain model under VMI, which is more applicable to real-world inventory control systems. Then, considering that the objective function is very complex to solve exactly by using conventional techniques, some solution algorithms are used. Finally, we introduce the problem of scheduling on a single machine with focusing on Early/Tardy problem with distinct due dates. Then, an INLP model is formulated and an algorithm is developed in order to solve the single machine earliness and tardiness problem. For all the three problems, numerical results are discussed.

Introduction

It is notable that, whenever a problem should be optimized, the disintegration of the problem in sub-problems yields a sub-optimal solution. In any case, many problems are exceptionally hard to handle and optimization models go for optimizing a small part of a much more complex system. Though in a few circumstances the problem is generally autonomous of the rest of the system, in other circumstances this is not the case. In general, modeling more comprehensive problems creates the opportunity for more savings.

In transportation and logistics, a pattern toward considering an ever increasing number of comprehensive systems can be observed both in practice and in the academic world. The idea of logistics has advanced over time to incorporate increasingly company works and has recently evolved into the concept of supply chain management that goes for incorporating more companies into the joining procedure. The advance of information systems, the availability of data and the rise of the internet have favored this direction.

Transportation is an important domain of human activity. It supports and enables many other social and economic activities and exchanges. Every day a distribution company has to decide how to put goods onto pallets to serve the customers' orders and then how to efficiently distribute these pallets among the vehicles, in order to get the right goods to the right place, and in the desired conditions, while minimizing the number of vehicles used.

The *Inventory Routing Problem* (IRP) integrates inventory management, vehicle routing, and delivery scheduling decisions. Its study is rooted in the seminal paper of Bell et al. [7], published 30 years ago. The IRP arises in the context of VMI, a business practice aimed at reducing logistics costs and adding business value. Because VMI provides advantages to both the supplier and the customers, it is natural to think that integrating one more element of the supply chain may lead to an even better performance. In IRP the goal can be achieved by determining the optimal distribution policy that minimizes the total cost (see Cordeau et al. [28]). IRP has investigated in several studies that an exhaustive review of previous studies has been presented by Andersson et al. [2] and Coelho et al. [27].

In this thesis we introduce, motivate and survey the area of IRP, where the first contributions date back to the eighties and that has been constantly growing over time. The adoption by several companies of the so called VMI technique in supply chain management has partially driven the research in this area. Thousands of paper are available for the modeling and solution of routing problems and thousands of papers can be found on inventory management models. Inventory routing problems aim at integrating the two areas.

Logistics is concerned with getting the right product (or service) to the right place at the right time. While there is a variety of potential applications of logistics and transportation problems, we have focused on three areas that are of particular importance for companies. These are transportation problems, inventory control and scheduling problems. While the Single Link Shipping Problem is of very special structure that is specific for IRP problems, the others problems are certainly of general relevance. An important problem class that we have covered here is the single machine scheduling problem. This problem has certain aspects of scheduling problems under decision-making problems which play a critical role in manufacturing as well as in service industries.

In Chapter 1, we focus on the *Single Link Shipping Problem* in the case of multi-product and multi-modal shipping with given set of frequencies and lead times with the aim of minimizing the total cost. We consider a transportation problem where different products have to be shipped from an origin to a destination by means of a set of types of pallets and vehicles. For each type a set of shipping frequencies and a transportation capacity are given. The production rate of each product at the origin and the demand rate at the destination are deterministic, constant over time and identical. The problem consists in determining the number of pallets to ship at each frequency for each type of vehicles and the number of vehicles of each type to use at each frequency, with the objective of minimizing the sum of the transportation cost and the inventory cost per time unit.

In Chapter 2, an *Economic Production Quantity* (EPQ) model in the case of multi-product by considering a warehouse space limitation is presented. This model is developed for a two-level supply chain consisting of a single supplier and two buyers and examines the inventory management practices before and after implementation of VMI. It is shown that the model of the problem is a constrained non-linear integer program thus *Particle Swarm Optimization* (PSO) and *Genetic Algorithm* (GA) are used to solve it. Finally, in order to illustrate the applicability of the proposed methodology a numerical example is provided at the end of the chapter.

In Chapter 3, scheduling is examined. As *Just-In-Time* (JIT) production system prevails in modern production systems, the earliness and tardiness (ET) objectives have received much attention in scheduling theory. The objective of this chapter is to develop an algorithm, in order to solve the single machine earliness and tardiness problem. The parameters of Tabu search (TS) Method are optimized through experiments, and some analyses are concluded. Different sizes of sample tests are used to compare the solutions resulted from the two solution methods. Comparisons between the two approaches are discussed. Finally, conclusions indicate that both methods could be an effective and efficient technique to solve the problem with the TS method performing better.

Symbols and Notations

In this section, we include a description of major symbols and notations used in all the chapters of the current work. The additional notation may be needed within specific chapter/sections and it is explained when used. To the greatest extent possible, we have attempted to keep unique meanings for each item. In those cases where an item has additional uses, it should be clear from context.

Chapter1	
Symbol	Definition
H	time period (or horizon) of the policy
I	set of product indices
i	index of products
P	set of pallet indices
p	index of pallets
m_p	weight per type of pallets p
q_{ip}	number of pallets p carrying product i
h_{ip}	inventory holding cost of each pallet p carrying one unit of product i per time unit
K	set of types of vehicles (FLC)
k	index of vehicles (FLC)
Q_k	capacity of vehicle of type k (FLC)
c_k	cost of a single trip of vehicle of type k (FLC)
L_k	lead time vehicles of type k (FLC)
J_k	set of frequency indices of vehicle of type k (FLC)
j	index of frequencies (FLC)
t_{kj}	intershipment time associated to vehicle k at frequency j (FLC)
x_{ipkj}	number of pallets p loading product i to ship with vehicle k at frequency j (FLC)
y_{kj}	number of vehicles of type k to use at frequency j (FLC)
L	set of types of vehicles (LCL)
l	index of vehicles (LCL)
d_{lp}	transportation cost per pallet p of vehicle of type l (LCL)
F_l	lead time vehicles of type l (LCL)
S_l	set of frequency indices of vehicle of type l (LCL)
s	index of frequencies (LCL)
τ_{ls}	intershipment time associated to vehicle l at frequency s (FLC)
z_{ipls}	number of pallets p loading product i to ship with vehicle l at frequency s (LCL)
T	set of possible shipping time indices
t	index of possible shipping times

I^A_{ipt}	inventory level of pallets type p at time t of product i at the supplier A
I^B_{ipt}	inventory level of pallets type p at time t of product i at the customer B
I^A_{ip0}	starting inventory level of pallets type p and product i at the supplier A
I^B_{ip0}	starting inventory level of pallets type p and product i at the customer B
β_{kj}	number of shipments using vehicle k at frequency j (FLC)
α_{kjt}	cumulative number of shipments using vehicle k at frequency j (FLC) up to time t
α'_{kjt}	cumulative number of shipments received by vehicle k at frequency j (FLC) up to time t
δ_{ls}	number of shipments using vehicle l at frequency s (LCL)
γ_{lst}	cumulative number of using vehicle l at frequency s (LCL) up to time t
γ'_{lst}	cumulative number of shipments received by vehicle l at frequency s (LCL) up to time t

Chapter2

Symbol	Definition
i	index of buyers
j	index of products
A_j	supplier setup cost to produce product j per cycle
C_{ij}	buyer i setup cost to buy product j per cycle
D_{ij}	demand rate of product j by buyer i
h_{ij}	holding cost of product j at buyer i per unit and per time unit
P_j	product j output rate
f_j	space occupied by each unit of product j
F	available warehouse space at the supplier for all products
T	supplier optimum production cycle in VMI system
Q_{ij}	order quantity of product j by buyer i
TC	total inventory cost of all products
TC_{noVMI}	total costs of all products in traditional system
TC_{VMI}	total costs of all products in VMI system
KB_{0i}	buyer inventory cost before running VMI system
KB_{1i}	buyer inventory cost after running VMI system
KS_0	supplier inventory cost before running VMI system
KS_1	supplier inventory cost after running VMI system

Chapter3	
Symbol	Definition
i	index of counter of sequenced jobs
j	index of jobs
d_j	processing time of each job j
dd_j	due date of each job j
ds_j	completion time of each job j
e_j	earliness of each job j
t_j	and tardiness of each job j
α_j	unit earliness penalty of each job j
β_j	unit tardiness penalty of each job j

Contents

List of Figures

List of Tables

Chapter 1

A Multi–Modal Single Link Shipping Problem with Given Frequencies and Lead Times

1.1 Introduction

The *Vehicle Routing Problem* (VRP) is a classical combinatorial optimization problem. A set of routes is computed to serve the deterministic demand of a set of customers from one supplier, at minimum cost. Dantzig and Ramser [31] proposed the first formulation for this problem in 1959. Dantzig et al. [32] formulated a binary linear programming model. Clarke and Wright [25] proposed the *Saving method* to solve it heuristically. This method was the basis for many future studies, so that many methods have been developed based on their study (see for example Golden and Nguyen [49] and Toth and Vigo [81]). This problem is also used by shippers or bearers, as represented by the many papers focused on managing the booking and the delivery of vehicles (see for example Crainic [29], Crainic and Laporte [30]). We refer to Toth and Vigo [82] for a comprehensive description of the problem and its variants.

Companies and organizations have recognized for years the importance of integrating the activities carried out in an efficient supply chain management and therefore researchers have focused on it. One of the areas that many studies recently have noticed is the optimization of the integrated management of inventory and distribution in a supply chain, so that the total cost related to ordering, storage and transportation is minimized.

The *Inventory Routing Problem* (IRP) is an important extension of the VRP, where inventory control and routing decisions are entwined (see Campbell et al. [22], Cordeau et al. [28], Bertazzi et al. [18], Andersson et al. [2], Bertazzi and Speranza [11], Bertazzi and Speranza [12], Coelho et al. [27]). In this area, Bell et al. [7] were almost the first ones who introduced an integrated model of vehicle inventory and scheduling management. The IRP has several applications in *Vendor Managed Inventory* (VMI). In VMI, the supplier is able to control the timing and size of product delivery to retailers. In return for this freedom, the supplier guarantees that customers do not face a shortage (stock-out). In the traditional relationship between supplier and customers, where customers order products to the supplier, the performance drastically reduces and, on the other hand, the total cost sharply increases due to the timing of customers orders. However, using the VMI in practice is not easy, especially when the number and variety of customers are high. Achieving this goal is possible by using an IRP model.

In different IRP problems, the number of suppliers and customers may differ. Along these lines, the structure is *one-to-many* in the most widely recognized case, with one supplier and several customers, and *one-to-one*, when there is one supplier serving one customer. In the former case (one-to-many), a set of products has to be shipped from the supplier to the customers to fulfill the demand over a time horizon, that can be infinite or finite. Inventory costs are charged at the supplier and at the customers. The circumstance of inventory costs at the supplier just or at the customers just are extraordinary cases. Shipments are performed by a fleet of vehicles having given capacity. A transportation cost is paid for each route traveled by the vehicles. The case with deterministic demand has been studied, among the others, by Burns et al. [21], Bertazzi et al. [13], Archetti et al. [3], Bertazzi et al. [15], Bertazzi et al. [16]. Federgruen and Zipkin [42] proposed the first model of the IRP with stochastic demand. Several types of IRP models were improved by various researchers, for example the IRP with single product (see Shen and Honda [75]), the IRP with the assumption of separate deliveries (see Yu et al. [88]) and the IRP with transshipment, i.e., the assumption of legality of transferring products between retailers (see Coelho et al. [26]). The second case (one-to-one), referred to as the *Single Link Shipping Problem*, is very important in long-haul transportation logistics (international and overseas). The supplier and the customer are part of the same company or have an agreement to manage the overall system at minimum cost. Therefore, one decision-maker has to determine when to make the shipments in order to minimize the total cost. Since in long-haul transportation, the transportation cost is high, a policy often used in practice is to send full load vehicles. However, as shown in Bertazzi and Speranza [10]), this policy is suboptimal when the sum of the inventory cost and transportation cost is minimized. The concept of frequency is commonly used in the *Single Link Shipping Problem*. A service may be provided daily, or weekly, or monthly, sometimes more often, for example twice a day. The use of frequency is well accepted in practice because of the regularity of a frequency-based process that simplifies the organization of both the supplier and the customer. In some cases, a set of possible frequencies is given. This is of interest when the shipping frequencies are not defined by the decision–maker, but by third parties.

Different models for the *Single Link Shipping Problem*, based on different assumptions on shipping times are available: 1) The continuous case, in which it is assumed that a shipment can be performed at any time and is aimed at finding an optimal periodic policy (see for instance Harris [51], Blumenfeld et al. [19] and Bertazzi and Speranza [9]); 2) The continuous case with minimum intershipment time, in which there is a minimum time between consecutive shipments (see Bertazzi et al. [17]); 3) The discrete case, in which the basic assumption is that shipments are performed at multiples of the minimum intershipment time (see Speranza and Ukovich [77], Speranza and Ukovich [78], Bertazzi and Speranza [8], Bertazzi and Speranza [10], Bertazzi et al. [14], Bertazzi and Speranza [9]).

An extension of the *Single Link Shipping Problem* is given by the *Sequences of Links*, in which the management of one origin, one or a few intermediate depots and one destination are integrated. These logistic networks are ordinarily utilized as a part of the inter-modal systems, in which distinctive sorts of vehicles are used in different links.

Our research in based on the IRP with one origin and one destination (*Single Link Shipping Problem*) in the case of multi-product and multi-modal shipments with given set of frequencies and lead times aiming at minimizing the total cost (see Bertazzi and Speranza [11], Bertazzi and Speranza [12] and Bertazzi et al. [18]). We present here an optimization model that, given a set of possible frequencies, identifies the best ones. This problem is of interest when the possible shipment frequencies are not defined by the decision-maker, but by third parties. The practical relevance of this problem has been pointed out in Bertazzi and Speranza [11]. The problem without lead times was studied in Bertazzi and Speranza [9], Bertazzi et al. [14] and Speranza and Ukovich [78]. Exact and heuristic algorithms were proposed in Bertazzi et al. [14] and Speranza and Ukovich [78]. The problem we studied is new, so no benchmark instances are available. The novelty with respect to the literature is due to the introduction of different sets of frequencies for each type of vehicles and to the introduction of lead times.

The remainder of this chapter is organized as follows. In Sections 1.2 and 1.3 the problem is described and formulated, respectively. A numerical example is shown in Section 1.4, and a systematic computational experiment is presented in Section 1.5 and 1.6.

1.2 Problem Definition

We study the problem of sending a set I of items from a supplier A to a customer B in the time periods $t \in T = \{1, 2, \ldots, H\}$. A set P of types of pallets is available. Each type $p \in P$ has a different weight m_p, which is a submultiple of the weight of the largest type of pallets. A typical example is given by the following three types of EURO pallets: Full (1000 kg), Half (500 kg) and Quarter (250 kg). The deterministic demand rate of each item $i \in I$ at the customer is defined in terms of number of pallets q_{ip} of each type $p \in P$, carrying item i, required in each time period $t \in T$. The production rate at the supplier is equal to the demand rate q_{ip}. For each item $i \in I$, a unit inventory cost h_{ip} is paid for each pallet of type $p \in P$, carrying item $i \in I$, left at the supplier and at the customer at the end of each time period $t \in T$, while no stock-out can occur. Shipments are performed by loading pallets on ships and planes. The shipment can be FLC (*Full Load Containers*) or LCL (*Less than Container Load*). In the first case, a set K of FLC container types is given. Each type $k \in K$ is defined by the type of vehicles on which it is loaded and has a different capacity Q_k, a different transportation cost c_k, a different set J_k of shipping frequencies and a different lead time L_k. The transportation cost c_k is a fixed cost, paid to rent the container, independently of the number of pallets loaded on the container. Each frequency $j \in J_k$ has an intershipment time t_{kj} (number of time periods between one shipment and the next one). The lead time L_k is a multiple of the time unit able to take into account of all operations to carry out to send the items from the supplier to the customer (e.g., 5 days for planes, 23 days for ships). A set L of LCL shipment types is given. Each type $l \in L$ is defined by the type of vehicle and has a different unit transportation cost d_{lp} per pallet of type $p \in P$, a different set S_l of shipping frequencies and a different lead time F_l. The transportation cost d_{lp} is a proportional cost, paid to send 1 pallet of type $p \in P$ with LCL type $l \in L$. Each frequency $s \in S_l$ has an intershipment time τ_{ls}. No transportation capacity is given for LCL: pallets can be loaded on different containers or a container can be shared by several shippers, or containers are not used. The aim of this problem is to determine a periodic shipping strategy that minimizes the total cost in the time unit, given by the sum of the inventory cost at the supplier and at the customer and of the transportation cost, in the time unit. A periodic shipping strategy is given by:

- the initial inventory level I^A_{ip0} of pallets of type $p \in P$, carrying item $i \in I$, to make available at the supplier A;

- the initial inventory level I^B_{ip0} of pallets of type $p \in P$, carrying item $i \in I$, to make available at the customer B;
- the number x_{ipkj} of pallets of type $p \in P$, carrying item $i \in I$, to load on FLC containers of type $k \in K$ traveling at frequency $j \in J_k$;
- the number y_{kj} of FLC containers of type $k \in K$ to use at frequency $j \in J_k$;
- the number z_{ipls} of pallets of type $p \in P$, carrying item $i \in I$, to send with a LCL shipment of type $l \in L$ traveling at frequency $s \in S_l$.

Let us denote with I^A_{ipt} and I^B_{ipt} the inventory level of pallets of type $p \in P$, carrying item $i \in I$, at the end of time period $t \in T$, at the supplier A and at the customer B, respectively.

1.3 Model Formulation

In this section, we formulate an integer linear programming model, referred to as Problem $\mathcal{H}$, for the problem described in the previous section. We first define the following data:

The definition of data related to the model is the following: β_{kj} is total number of shipments using vehicle k at frequency j for FLC, α_{kjt} is related to the supplier and it is the cumulative number of shipments using vehicle k at frequency j for FLC up to time t, α'_{kjt} is related to the customer and it is the cumulative number of shipments received by vehicle k at frequency j for FLC up to time t, δ_{ls} is total number of shipments using vehicle l at frequency s for LCL, γ_{lst} is related to the supplier and it is the cumulative number of shipments using vehicle l at frequency s for LCL up to time t and γ'_{lst} is related to the customer and it is the cumulative number of shipments received by vehicle l at frequency s for LCL up to time t. Let us formally introduce these definitions:

- Total number of shipments over H for FLC containers of type $k \in K$ traveling at frequency $j \in J_k$:

$$\beta_{kj} = \begin{cases} \frac{H-L_k}{t_{kj}}, & \text{if } H-L_k \text{ is a multiple of } t_{kj} \\ \lfloor \frac{H-L_k}{t_{kj}} \rfloor + 1, & \text{otherwise.} \end{cases} \tag{1.1}$$

- Cumulative number of shipments up to time $t \in T$ for FLC containers of type $k \in K$ traveling at frequency $j \in J_k$:

$$\alpha_{kjt} = \begin{cases} \left\lfloor \frac{t-1}{t_{kj}} \right\rfloor + 1 & \text{if } t \leq H-L_k \\ \beta_{kj} & \text{otherwise.} \end{cases} \tag{1.2}$$

- Cumulative number of shipments received by the customer up to time $t \in T$ for FLC containers of type $k \in K$ traveling at frequency $j \in J_k$:

$$\alpha'_{kjt} = \begin{cases} 0 & \text{if } t \leq L_k \\ \left\lfloor \frac{t-1-L_k}{t_{kj}} \right\rfloor + 1 & \text{otherwise.} \end{cases} \tag{1.3}$$

- Total number of shipments over H for LCL shipments of type $l \in L$ traveling at frequency $s \in S_l$:

$$\delta_{ls} = \begin{cases} \frac{H-F_l}{\tau_{ls}}, & \text{if } H-F_l \text{ is a multiple of } \tau_{ls} \\ \lfloor \frac{H-F_l}{\tau_{ls}} \rfloor + 1, & \text{otherwise.} \end{cases} \tag{1.4}$$

- Cumulative number of shipments up to time $t \in T$ for LCL shipments of type $l \in L$ traveling at frequency $s \in S_l$:

$$\gamma_{lst} = \begin{cases} \left\lfloor \frac{t-1}{\tau_{ls}} \right\rfloor + 1 & \text{if } t \leq H - F_l \\ \delta_{ls} & \text{otherwise.} \end{cases} \tag{1.5}$$

- Cumulative number of shipments received by the customer up to time $t \in T$ for LCL shipments of type $l \in L$ traveling at frequency $s \in S_l$:

$$\gamma'_{lst} = \begin{cases} 0 & \text{if } t \leq F_l \\ \left\lfloor \frac{t-1-F_l}{\tau_{ls}} \right\rfloor + 1 & \text{otherwise.} \end{cases} \tag{1.6}$$

The integer linear programming model can be formulated as follows.

Problem $\mathcal{H}$

$$\min \sum_{i \in I} \sum_{p \in P} \sum_{t \in T} \frac{h_{ip}}{H} (I^A_{ipt} + I^B_{ipt}) + \sum_{k \in K} \sum_{j \in J_k} \frac{c_k}{H} \beta_{kj} y_{kj} + \sum_{l \in L} \sum_{s \in S_l} \sum_{i \in I} \sum_{p \in P} \frac{d_{lp}}{H} \delta_{ls} z_{ipls} \tag{1.7}$$

$$I^A_{ipt} = I^A_{ip0} + q_{ip}(t-1) - \sum_{k \in K} \sum_{j \in J_k} \alpha_{kjt} x_{ipkj} - \sum_{l \in L} \sum_{s \in S_l} \gamma_{lst} z_{ipls} \quad i \in I \quad p \in P \quad t \in T, \tag{1.8}$$

$$I^B_{ipt} = I^B_{ip0} + \sum_{k \in K} \sum_{j \in J_k} \alpha'_{kjt} x_{ipkj} + \sum_{l \in L} \sum_{s \in S_l} \gamma'_{lst} z_{ipls} - q_{ip} t \qquad i \in I \quad p \in P \quad t \in T, \tag{1.9}$$

$$\sum_{k \in K} \sum_{j \in J_k} \beta_{kj} x_{ipkj} + \sum_{l \in L} \sum_{s \in S_l} \delta_{ls} z_{ipls} = q_{ip} H \qquad i \in I \quad p \in P, \tag{1.10}$$

$$\sum_{i \in I} \sum_{p \in P} m_p x_{ipkj} \leq Q_k y_{kj} \qquad j \in J_k \quad k \in K, \tag{1.11}$$

$$I^A_{ipt} \geq 0 \qquad i \in I \quad p \in P \quad t \in T, \tag{1.12}$$

$$I^B_{ipt} \geq 0 \qquad i \in I \quad p \in P \quad t \in T, \tag{1.13}$$

$$x_{ipkj} \geq 0 \quad \text{integer} \qquad i \in I \quad p \in P \quad j \in J_k \quad k \in K, \tag{1.14}$$

$$z_{ipls} \geq 0 \quad \text{integer} \qquad i \in I \quad p \in P \quad s \in S_l \quad l \in L, \tag{1.15}$$

$$y_{kj} \geq 0 \quad \text{integer} \qquad j \in J_k \quad k \in K. \tag{1.16}$$

The objective function (1.7) expresses the minimization of the sum of the inventory and transportation costs in the time unit. Constraints (1.8) and (1.9) are the inventory level definition constraints in A and in B. Constraints (1.8) show that the inventory level at time t at the supplier is equal to the starting inventory level, plus the total quantity produced up to time $t-1$, minus the total quantity shipped to destination up to time t. Constraints (1.9) define that the inventory level at time t at the customer is equal to the starting inventory level, plus the total quantity received up to time t, minus the total demand up to time t. Constraints (1.10) guarantee that the pallets of type p carrying product i are totally shipped from the supplier to the customer by using one or several frequencies. Constraints (1.11) guarantee that the number of FLC containers of each type used at each frequency is sufficient for loading pallets. Finally, (1.12)-(1.16) define the decision variables. Note that (1.12) and (1.13) are also stock-out constraints.

1.4 A Numerical Example

In this section, we propose a numerical example, inspired by real data, to show the optimal solution of Problem $\mathcal{H}$.

Three items $i = 1, 2, 3$ are produced in Italy over a time horizon $H = 96$ days. Shipments are from Italy (Genoa) to USA (New York). The Italian unit sends the products to the USA unit by ships with FLC, by ships with LCL and by planes with LCL. The critical point in this logistic system is the physical distance between Italy and USA, and therefore the corresponding high lead times.

EURO pallets are considered in which the weight m_p is 1000 kg (full) for $p = 1$, 500 kg (half) for $p = 2$ and 250 kg (quarter) for $p = 3$.

Table 1.1 provides data concerning the items $i \in I$ and the pallets $p \in P$, namely the inventory cost h_{ip} and the production and demand rate q_{ip}:

i	p	h_{ip}	q_{ip}
1	1	0	25
2	2	250	15
3	3	250	10

Table 1.1: Data related to products and pallets

There are two different types of vehicles (ships and planes). Two types of FLC containers (20 ft, 40 ft) are assumed in which transportation capacity Q_k, with $k = 1, 2$, is:

- 20 ft: maximum 11 full pallets ($Q_1 = 11000$)
- 40 ft: maximum 25 full pallets ($Q_2 = 25000$).

Table 1.2 shows data concerning the set K of FLC containers, while Table 1.3 shows data concerning the set L of LCL shipments.

We compare the optimal solution and cost obtained in the cases without and with lead times, respectively. We also set the variables x, z and y in the model with lead times equal to the optimal value obtained in the case without lead times. This is a more fair comparison than the one comparing the cases without and with lead times. The computational results are shown in

	k	J_k	t_{kj}	c_k	Q_k	L_k
Ships with FLC (containers 20 ft)	1	1,2	6,12	2000	11000	23
Ships with FLC (containers 40 ft)	2	1,2	6,12	2500	25000	23

Table 1.2: Data related to FLC containers

	l	S_l	τ_{ls}	d_{lp}	F_l
Ships with LCL (1 full pallet)	1	1,2	12,24	300	23
Ships with LCL (1 half pallet)	2	1,2	12,24	400	23
Ships with LCL (1 quarter pallet)	3	1,2	12,24	200	23
Planes with LCL (1 full pallet)	4	1	2	3000	5
Planes with LCL (1 half pallet)	5	1	2	1500	5
Planes with LCL (1 quarter pallet)	6	1	2	1000	5

Table 1.3: Data related to LCL containers

Tables 1.4- 1.6. Table 1.4 corresponds to the situation in which lead times are equal to 0. Table 1.5 corresponds to the situation in which lead times are taken into account. Table 1.6 shows more complex situations in which results are obtained with lead times and fixed optimal value in the case without lead times. Moreover, Tables 1.7 and 1.8 show the model with lead times with all variables x and y equal to 0 and with all variables z equal to 0, respectively. The tables are organized as follows: the first column shows the number of pallets of type p, carrying product i, to ship with vehicle of type k at frequency j (FLC), the second the number of pallets of type p, carrying product i, to ship with vehicle of type l at frequency s (LCL), the third the number and type k of vehicles to use at each frequency j, and the last three columns the inventory, transportation and total cost, respectively.

Figure 1.1 shows the inventory level of product i at the supplier and at the costumer on each day $t \in T$.

x^*_{ipkj}	z^*_{ipls}	y^*_{kj}	Inventory cost	Transportation cost	Total cost per time unit
$x^*_{1121} = 100$ $x^*_{1122} = 100$ $x^*_{2221} = 90$	$z^*_{3321} = 20$	$y^*_{21} = 6$ $y^*_{22} = 4$	21250.0	13333.3	34583.3

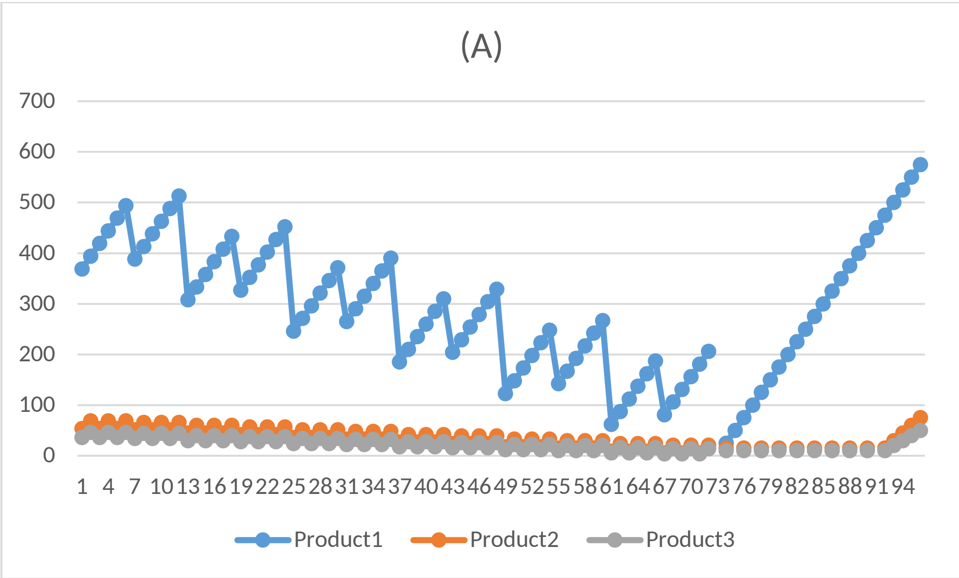
Table 1.4: Results with lead times equal to 0

x^*_{ipkj}	z^*_{ipls}	y^*_{kj}	Inventory cost	Transportation cost	Total cost per time unit
$x^*_{1111} = 33$ $x^*_{1121} = 98$ $x^*_{1122} = 98$ $x^*_{2221} = 3$ $x^*_{2222} = 3$ $x^*_{3321} = 2$ $x^*_{3322} = 2$	$z^*_{1111} = 1$ $z^*_{1112} = 1$ $z^*_{2221} = 30$ $z^*_{3321} = 20$	$y^*_{11} = 3$ $y^*_{21} = 4$ $y^*_{22} = 4$	26562.5	34076.0	60638.5

Table 1.5: Results with lead times

x^*_{ipkj}	z^*_{ipls}	y^*_{kj}	Inventory cost	Transportation cost	Total cost per time unit
$x^*_{1111} = 29$ $x^*_{1121} = 100$ $x^*_{1122} = 100$ $x^*_{2211} = 2$ $x^*_{2221} = 90$ $x^*_{3311} = 1$ $x^*_{3321} = 1$	$z^*_{1111} = 1$ $z^*_{1112} = 4$ $z^*_{2211} = 2$ $z^*_{2221} = 5$ $z^*_{3311} = 2$ $z^*_{3321} = 20$	$y^*_{11} = 3$ $y^*_{21} = 6$ $y^*_{22} = 4$	85156.2	16909.4	102065.6

Table 1.6: Results with lead times and fixed optimal value in the case without lead times



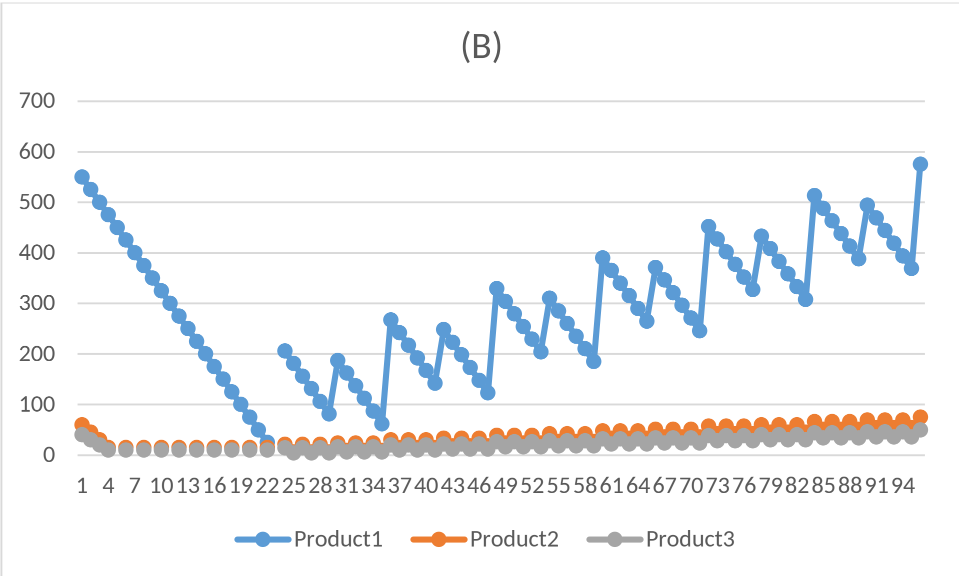


Figure 1.1: Set of inventory level for different products: (A) The inventory level at the supplier. (B) The inventory level at the customer.

x^*_{ipkj}	z^*_{ipls}	y^*_{kj}	Inventory cost	Transportation cost	Total cost per time unit
	$z^*_{1111} = 98$				
	$z^*_{1112} = 95$				
	$z^*_{1121} = 29$				
	$z^*_{2211} = 8$				
$All = 0$	$z^*_{2212} = 1$	$All = 0$	26562.5	76497.9	103060.4
	$z^*_{2221} = 30$				
	$z^*_{3311} = 4$				
	$z^*_{3312} = 3$				
	$z^*_{3321} = 20$				

Table 1.7: Results with lead times and all variables x and y equal to 0

x^*_{ipkj}	z^*_{ipls}	y^*_{kj}	Inventory cost	Transportation cost	Total cost per time unit
$x^*_{1111} = 42$					
$x^*_{1121} = 99$					
$x^*_{1122} = 81$					
$x^*_{2211} = 1$		$y^*_{11} = 4$			
$x^*_{2221} = 71$	$All = 0$	$y^*_{21} = 6$	143750.0	4026.0	147776.0
$x^*_{2222} = 72$		$y^*_{22} = 5$			
$x^*_{3311} = 5$					
$x^*_{3321} = 57$					
$x^*_{3322} = 22$					

Table 1.8: Results with lead times and all variables z equal to 0

The results show that the solution with minimum total cost is, as expected, the one corresponding to the model with lead times equal to 0. This is due to the fact that in this model the lead times is not imposed. However, this solution is not feasible if there are positive lead times. The total cost obtained by setting the value of the variables of the model without lead times in the model with lead times is 40.58% greater than the model with lead times. Instead, a significant increase in the total cost and inventory cost is obtained when fixing value of the variables of the model without lead times are imposed, while the transportation cost is decreasing. In fact, imposing all variables x and y equal to 0 and all variables z equal to 0, the total cost increases of about 41.16% and 58.96%, respectively. Let us now analyze the cost components. The result shows that inventory cost is constant in the case of all variables x and y equal to 0 while increases of 81.52% in the case of all variables z equal to 0. However, the transportation cost is increased in the first case, while is reduced in the second case. In the model with all variables z equal to 0, the inventory cost and the total cost have their maximum value, while the transportation cost is minimized. However, in the model without lead times the inventory cost and total cost have their minimum value and the transportation cost in the model with all variables x and y equal to 0 has the maximum value. If we compare the solution with maximum total cost with the solution with the model with lead times we can note that the increase in the total cost is mainly due to the inventory cost, while the transportation cost is reduced. The results in the cases of all variables x and y equal to 0 and all variables z equal to 0 show that using FLC and LCL together is more valuable. Finally, the number of shipments by ship with FLC does not vary significantly in different models, while the product is never sent by plane with LCL. Whenever a shipment of the product is performed, it is performed by ship with FLC and LCL.

1.5 Computational Results

In this section, we show the results obtained in a set of problem instances generated by systematically varying the lead times and the frequencies. The data related to the other parameters are the same provided in Section 1.4.

Tables 1.10- 1.13 and Tables 1.17- 1.20 show the results obtained for the first sets of frequencies and second sets of frequencies according to Table 1.9.

	t_{kj}	τ_{ls}	L_k	F_l	H
First sets of frequencies	6,12	12,24,2	0,23	0,5	96
Second sets of frequencies	6,12,18,24,36,48	12,24,36,48,2,4,6,8,12	0,23	0,5	144

Table 1.9: Data related to first and second sets of frequencies

Table 1.9 shows data related to the first sets of frequencies with t_{kj} = 6, 12 and τ_{ls} = 12, 24, 2 over a planning horizon $H = 96$ and the data related to the second sets of frequencies with t_{kj} = 6, 12, 18, 24, 36, 48 and τ_{ls} = 12, 24, 36, 48, 2, 4, 6, 8, 12 over a planning horizon H = 144. Tables 1.10- 1.13 and Tables 1.17- 1.20 correspond to a different combination of lead times: 0, 0 when $L_k = 0$ for all k and $F_l = 0$ for all l; 23, 0 when $L_k = 23$ for all k and $F_l = 0$ for all l; 0, 5 when $L_k = 0$ for all k and $F_l = 5$ for all l; 23, 5 when $L_k = 23$ for all k and $F_l = 5$ for all l. We compare the optimal solutions obtained by solving different combinations of lead times for both sets. In all cases, a shipment by ship with FLC is preferable to a shipment by ship or plane with LCL. In the following tables, we show the cost components and the total costs obtained in the different combinations of lead times and statistics about the shipments, number of pallets of FLC and LCL shipments and number of vehicles. The results show that the number of pallets type 2, carrying product 2, to ship with vehicle of type 2 at frequency 1 (FLC) and the number of pallets type 3, carrying product 3, to ship with vehicle of type 2 at frequency 1 (FLC) have a significant decrease in the case with lead times equal to 23, 5 in the first sets, while there are not impressive differences in the other cases. The average number of utilization of pallets type 1, carrying product 1, to ship with vehicle type 2 at frequency 1 (FLC) and pallets type 3, carrying product 3, to ship with vehicle type 2 at frequency 1 (LCL) in all models are higher than other cases. The results with lead times equal to 0, 5 in first sets show that the product is never sent by

LCL shipments. Finally, the number of shipments by ship with FLC does not vary significantly in different models, while the product is never sent by plane with LCL. Whenever a shipment of the product is performed, it is performed by ship with FLC and LCL.

First sets of frequencies

x^*_{ipkj}	z^*_{ipls}	y^*_{kj}	Inventory cost	Transportation cost	Total cost per time unit
$x^*_{1121} = 100$ $x^*_{1122} = 100$ $x^*_{2221} = 90$	$z^*_{3321} = 20$	$y^*_{21} = 6$ $y^*_{22} = 4$	21250.0	13333.3	34583.3

Table 1.10: Results with lead times equal to 0, 0

x^*_{ipkj}	z^*_{ipls}	y^*_{kj}	Inventory cost	Transportation cost	Total cost per time unit
$x^*_{1111} = 31$ $x^*_{1121} = 100$ $x^*_{1122} = 99$	$z^*_{1112} = 1$ $z^*_{2221} = 30$ $z^*_{3321} = 20$	$y^*_{11} = 3$ $y^*_{21} = 4$ $y^*_{22} = 4$	6250.0	35408.3	41658.3

Table 1.11: Results with lead times equal to 23, 0

x^*_{ipkj}	z^*_{ipls}	y^*_{kj}	Inventory cost	Transportation cost	Total cost per time unit
$x^*_{1111} = 10$ $x^*_{1121} = 90$ $x^*_{1122} = 100$ $x^*_{2221} = 90$ $x^*_{3321} = 60$	$All = 0$	$y^*_{11} = 1$ $y^*_{21} = 6$ $y^*_{22} = 4$	31250.0	3666.7	34916.7

Table 1.12: Results with lead times equal to 0, 5

x^*_{ipkj}	z^*_{ipls}	y^*_{kj}	Inventory cost	Transportation cost	Total cost per time unit
$x^*_{1111} = 33$ $x^*_{1121} = 98$ $x^*_{1122} = 98$ $x^*_{2221} = 3$ $x^*_{2222} = 3$ $x^*_{3321} = 2$ $x^*_{3322} = 2$	$z^*_{1111} = 1$ $z^*_{1112} = 1$ $z^*_{2221} = 30$ $z^*_{3321} = 20$	$y^*_{11} = 3$ $y^*_{21} = 4$ $y^*_{22} = 4$	26562.5	34076.0	60638.5

Table 1.13: Results with lead times equal to 23, 5

Instance	Computational Time
0, 0	0.378
23, 0	0.504
0, 5	0.407
23, 5	0.585

Table 1.14: Computational time in seconds for different distances

Instance	Objective Function
0, 0	102065.6
23, 0	60675.0
0, 5	134458.3
$x, y = 0$	103060.4
$z = 0$	147776.0

Table 1.15: Best cost in the model with lead times equal to 23, 5 after setting the cases

Instance	Percentage Error
0, 0	40.58%
23, 0	0.06%
0, 5	54.90%
$x, y = 0$	41.16%
$z = 0$	58.96%

Table 1.16: Percent cost increase with respect to the cost of the model with lead times equal to 23, 5

Tables 1.14 and 1.21 show the computational times in GAMS using commercial solver (CPLEX) as solution kernel for the first and the second sets of frequencies. The computational experiments have been carried out with a Windows 10 as operating system with an Intel processor (Core i3 - 1.90 GHz) and 4 GB of RAM.

We use the model with the lead times equal to 23, 5 in both sets as the model to solve after setting the value of the variables equal to the optimal value obtained in the other cases. Moreover, we set the model with the lead times equal to 23, 5 in both sets with all variables x and y equal to 0 and with all variables z equal to 0. The values of the objective function in both sets are shown in Tables 1.15 and 1.22 and the percent increase error with respect to the cost of the model with lead times equal to 23, 5 is shown in Tables 1.16 and 1.23.

Second sets of frequencies

x^*_{ipkj}	z^*_{ipls}	y^*_{kj}	Inventory cost	Transportation cost	Total cost per time unit
$x^*_{1121} = 80$ $x^*_{1122} = 100$ $x^*_{1123} = 50$ $x^*_{1124} = 60$ $x^*_{2221} = 90$	$z^*_{3321} = 20$	$y^*_{21} = 5$ $y^*_{22} = 4$ $y^*_{23} = 2$ $y^*_{24} = 3$	21250.0	13281.3	34531.3

Table 1.17: Results with lead times equal to 0, 0

x^*_{ipkj}	z^*_{ipls}	y^*_{kj}	Inventory cost	Transportation cost	Total cost per time unit
$x^*_{1121} = 100$ $x^*_{1122} = 75$ $x^*_{1123} = 100$ $x^*_{1124} = 25$	$z^*_{2221} = 30$ $z^*_{3321} = 20$	$y^*_{21} = 4$ $y^*_{22} = 3$ $y^*_{23} = 4$ $y^*_{24} = 1$	6250.0	35000.0	41250.0

Table 1.18: Results with lead times equal to 23, 0

x^*_{ipkj}	z^*_{ipls}	y^*_{kj}	Inventory cost	Transportation cost	Total cost per time unit
$x^*_{1121} = 92$					
$x^*_{1122} = 98$		$y^*_{21} = 6$			
$x^*_{1123} = 24$		$y^*_{22} = 4$			
$x^*_{1124} = 24$	$z^*_{3323} = 10$	$y^*_{23} = 1$	29166.7	5156.2	34322.9
$x^*_{2221} = 90$		$y^*_{24} = 1$			
$x^*_{3321} = 50$					

Table 1.19: Results with lead times equal to 0, 5

x^*_{ipkj}	z^*_{ipls}	y^*_{kj}	Inventory cost	Transportation cost	Total cost per time unit
$x^*_{1121} = 100$					
$x^*_{1122} = 99$		$y^*_{21} = 4$			
$x^*_{1123} = 19$	$z^*_{2221} = 30$	$y^*_{22} = 4$			
$x^*_{1124} = 99$	$z^*_{3321} = 20$	$y^*_{23} = 1$	28125.0	34131.9	62256.9
$x^*_{2223} = 10$		$y^*_{24} = 4$			
$x^*_{3322} = 2$					
$x^*_{3323} = 3$					

Table 1.20: Results with lead times equal to 23, 5

Instance	Computational Time
0, 0	0.625
23, 0	0.944
0, 5	0.748
23,5	1.065

Table 1.21: Computational time in seconds for different distances

Instance	Objective Function
0, 0	103110.4
23, 0	62277.7
0, 5	129475.6
$x, y = 0$	89944.4
$z = 0$	147256.9

Table 1.22: Best cost in the model with lead times equal to 23, 5 after setting the cases

Instance	Percentage Error
0, 0	39.62%
23, 0	0.03%
0, 5	51.91%
$x, y = 0$	30.78%
$z = 0$	57.72%

Table 1.23: Percent cost increase with respect to the cost of the model with lead times equal to 23, 5

Note that the computational time to obtain the optimal solutions is less than 1 second with exception to case 23, 5. Finally, let us compare the optimal costs after setting the cases with the cost obtained with lead times equal to 23, 5. The increase in the total cost is only about 0.06% and 0.03% in the case with a lead time 23, 0 in first and second sets, respectively. It becomes about 30.78% in the case with all variables x and y equal to 0 in the second set, while it is more than 39.00% in the remaining cases.

1.6 Increasing the dimension of the instance

In this section, our aim is to show the computational time required to solve larger and larger instances.

We randomly generate an instance using the same data as in Section 1.5, but with 3 products. Each product is packed into pallets of type 1, 2 and 3. Therefore, q_{ip} is a matrix having rows indexed by i and columns indexed by p. Each of these numbers is randomly generated in the following intervals:

0-10, 11-20, 21-30 and 0-30.

The corresponding inventory cost h_{ip} is 0.01, 0.5 and 0.1 of the value of one pallet of type p carrying product i. To compute the value of one pallet, we assume a price of the product per kg. For example: If the price is 2 euros per kg, 1 full pallet has a value of 2000 euros and the corresponding inventory cost is $h_{ip} = 0.01 \times 2000$ euros or 0.5×2000 euros or 0.1×2000 euros. The price of each product is generated in an interval from 1 to 1000 euros. Based on these random numbers, we generate 1 instance with number of products equal to 3. The corresponding total cost per time unit is equal to 339319.5 and the computational time is equal to 985.2 seconds.

Table 1.24 shows the sets of frequencies with t_{kj} = 6, 12, 18, 24, 36, 48 , 72 , 96, 120, 144, 180, 240 and τ_{ls} = 12, 24, 36, 48, 96, 120, 180, 240, 2, 4, 6, 8, 12, 24, 36, 48, 96 over a planning horizon $H = 720$.

	t_{kj}	τ_{ls}	H
Sets of frequencies	6,12,18,24,36,48,72,	12,24,36,48,96,120,180,240,	720
	96,120,144,180,240	2,4,6,8,12,24,36,48,96	

Table 1.24: Data related to larger instances

Table 1.25 shows the computational times (seconds) obtained when we increase the number of types of containers.

The results show that the models with less number of types of containers can be easily solved to optimality. However, the computational time increases more than proportionally in the number of container types.

Number of Types of Containers	Computational Time
3	364.62
4	984.41
5	2944.32
6	3858.05

Table 1.25: Computational time for different number of types of containers in GAMS (seconds)

1.7 Conclusion

In this chapter, we studied the problem of shipping products from a common origin to a common destination. The aim is to minimize the sum of the transportation cost and the inventory cost both at the origin and at the destination. We present here an optimization model that, given a set of possible frequencies and lead times, identifies the best frequencies to use. Finally, we computationally showed that using FLC and LCL together allows us to reduce the total cost significantly.

Chapter 2

A Vendor-Managed Inventory multi-product EPQ model with constrained space

2.1 Introduction

One of the issues raised in supply chains is inventory management at various levels of the chain. Since there is a conflict between the demands at different levels, the organization can achieve a certain level of inventories. The aim of the inventory control system is to determine the timing and the order quantity (see Tersine [80]). Inventory management at the end of the supply chain is one of the issues that have attracted the attention of researchers, and it has significant importance for organizations. Supply chain management is an integrated approach to plan and control materials and information. The supply chain includes operations related to manufacturing and distribution, transportation, warehouses, and customers (see Chopra and Meindl [24]). Some of this research has led the *Vendor Managed Inventory* (VMI) models. VMI is a set in which the buyer of a product provides certain information to a supplier of that product, and the supplier takes full responsibility for maintaining an agreed inventory of the material (see Kumar and Kumar [60]). VMI is an economic model in which the vendor monitors the retailer inventory level, so that the vendor sets the timing and the delivery quantity for the retailer. As a result, this approach leads to reduce inventory costs for the vendor (see Achabal et al. [1]). The profit derived from this approach is shown by famous suppliers such as Wal-Mart, Penny, and Dylard (see Yao et al. [87]). More general models of *Economic Production Quantity* (EPQ) and *Economic Order Quantity* (EOQ) are used in supply chains. These economic models are based on certain assumptions which limited their application in real-world (see Axster [5]).

The works done in the second half of the 90s based on VMI can be classified into two categories:

1. Addressing pure and applied management issues based on simulation approach (see Holmström [55], Tyan and Wee [83] and Disney and Towill [35]).

2. Addressing mathematical issues and the provision of economic analysis (see Dong and Xu [36] and Yao et al. [87]).

With the development of VMI policies, organizations use the impact of the VMI policy as a tool for reducing supply chain costs (see Dong and Xu [36]). The supremacy of the VMI approach toward the traditional approach of inventory management is expressed by some scholars such as Yao et al. [87] and Dong and Xu [36]. Yao et al. [87] used the same assumption as Dong and Xu [36] and the additional assumption that the order quantity for the supplier is likely to be an integer multiple of the buyer's replenishment quantity. In this case, they compared both traditional and VMI system. As in previous studies, supply chain was consisting of a supplier and a buyer, while Jasemi [57] presented this comparison with n buyers. He also created a system of price determination such that buyers and sellers benefit as a percentage of the advantages of this system. A supply chain with a vendor and a retailer for a product in EPQ environment is studied by Hill [53]. Dong and Xu [36] presented a two-level supply chain under VMI system in EPQ environment that would include a retailer and a supplier. Also, the optimal approach of replenishment policy for perishable products is investigated by Wee et al. [85]. Chen et al. [23] examined the supply chain inventory management model with a vendor and multi-retailer according to bullwhip effect. An inventory model for supply chain with a retailer, a vendor, and a price discount policy is studied by Duan et al. [38]. Also, other modes of two-level supply chain consisting of a supplier and multi-retailer were studied by Zavanella and Zanoni [90], Darwish and Odah [33], and Lu [63]. Elvander et al. [41] studied a framework for designing VMI. Vigtil [84] researched on demand forecasting for VMI models to make better decisions in the supply chain. Zammori et al. [89] developed a standard architecture for implementing VMI. Sari [74] showed VMI usefulness for the uncertain demands by using discrete simulation. Borade and Bansod [20] studied VMI implementation for Indian industry.

There are several interesting and relevant papers related to the application of VMI system in inventory by using meta-heuristic methods like GA (Genetic Algorithms) to solve complicated optimization problems (see Pasandideh et al. [68], Sadeghi et al. [72], Pasandideh et al. [69] and Sadeghi et al. [73]). Aryanezhad et al. [4] developed a supply chain model with a single-supplier and n buyers and compared the performances of a VMI system with the ones of the traditional types. They expanded a multi-product EOQ model with limited warehouse-space and budget limitation. They also formulated the problem as a non-linear integer-programming (NIP) model, and then a GA is proposed to solve it. Also, Moghadam et al. [66] extend the previous research in which a multi-product EPQ model is considered.

This research deals with the mathematical issues of a two-echelon vendor-buyer supply chain model under VMI. This research is divided into two cases. In the first case, we analytically compare VMI and the traditional model of inventory control without taking account the capacity constraint. The criterion for this comparison is the total cost of supply chain and is determined by a supplier and two buyers. In the second case, we study a multi-product EPQ model with a supplier and two buyers in which there is limited storage space under VMI system. Considering a new objective function and different constraints makes the model more applicable to real-world inventory control systems with respect to the ones proposed in the literature. In addition, the solution method, that is based on two parameter-tuned meta-heuristic algorithms, seems to be more compact and simpler than the ones provided in earlier studies.

VMI model induces a broad approach to supply chain distribution channels which helps manufacturers, suppliers and retailers to improve production, reduce inventory, increase inventory turnover and strengthen access to goods. Access to this information can help producers to make better plan for their production to adapt to customer needs. Given the importance of supply chain in inventory management at retailers, the VMI model can help to improve performance in the supply chain. Supply chain optimization requires accurate transfer of information between members of the chain. If VMI is properly implemented, it will have significant results in supply chain. VMI will facilitate the transfer of information and create opportunities for vendors to save costs. Increasing the competitiveness of the supply chain through mutual cooperation is an incentive for retailer and supplier to implement the VMI model. The most obvious advantage of the VMI model for the retailers is the reduction of the inventory costs and for supplier is cost reduction which improves the productivity and quality of service and will result in increased sales and profits.

Exploring the effects of collaborative supply chain initiatives, such as VMI, and determining the impact of important supply chain parameters on cost savings, the previous research are extended and a multi-product EPQ model with one supplier and two buyers is proposed, in which limited storage space is considered for constant demand rate. Under these conditions, the inventory management models before and after the implementations of VMI are examined. Finally, we develop GA and PSO to solve the problem. The main aim of this research is to show the improvement in supply chain performance and to show the performance of the new system in comparison with the traditional one. The problem we studied is new, so no benchmark instances are available.

Firstly, this study will examine and calculate the total inventory cost of the supply chain before and after the implementation of VMI in different modes. Secondly, after calculating the total costs in multi-product supply chain inventory and considering that the cost function is very complex and difficult to solve exactly by using conventional techniques, genetic algorithms will be used.

The rest of this chapter is organized as follows. In Section 2.2, the problem is defined precisely. Sections 2.3, 2.4 and 2.5 are dedicated to the mathematical formulation of the problem, the case with and without capacity constraint, respectively. Comprehensive explanation of the methodology proposed to solve the model is discussed in Section 2.6. In Sections 2.7 and 2.8 a numerical example and computational results are given, respectively. Finally, conclusions are provided and future research directions are proposed in Section 2.9.

2.2 Problem Definition

We study supply chains under VMI with one supplier and two buyers. The period of this policy is infinite (infinite time horizon).

The assumptions of the model are defined as follows:

- Customer demand is with a constant rate.
- Products demand is independent of each other.
- Supplier production rate is greater than customer demand.
- Shortage is not allowable.
- Delivery time is zero.
- Handling time is zero.
- Unit costs are constant over time.

The aim is to find an optimal order quantity that minimizes the total inventory cost of the supply chain and satisfy the capacity constraint.

2.3 Notation

2.3.1 Parameters

For products $j = 1, \ldots, n$ and buyers $i = 1, 2$ the parameters of the model are defined as follows:

A_j = supplier setup cost to produce product j per cycle;

C_{ij} = buyer i setup cost to buy product j per cycle;

D_{ij} = demand rate of product j by buyer i;

h_{ij} = holding cost of product j at buyer i per unit and per time unit;

P_j = product j output rate;

f_j = space occupied by each unit of product j;

F = available warehouse space at the supplier for all products.

2.3.2 Decision Variables

T = supplier optimum production cycle in VMI system;

Q_{ij} = order quantity.

Moreover, we introduce the following notation:

TC_{noVMI} = total costs of all products in traditional system;

TC_{VMI} = total costs of all products in VMI system;

KB_{0i} = buyer inventory cost before running VMI system;

KB_{1i} = buyer inventory cost after running VMI system;

KS_0 = supplier inventory cost before running VMI system;

KS_1 = supplier inventory cost after running VMI system.

2.4 The case without capacity constraint

2.4.1 Before implementing VMI

Before implementing VMI, the supplier and buyer act separately in a traditional system. Each party is responsible for controlling of its own inventory. The total inventory cost of all products (TC) is given by the sum of buyer inventory cost (KB) and supplier inventory cost (KS) of all products. Therefore, before running the VMI system:

$$TC_{noVMI} = KB_{01} + KB_{02} + KS_0, \tag{2.1}$$

where the 1st and 2nd buyer inventory cost before running VMI system are as follow:

$$KB_{01} = \sum_{j=1}^{n} \frac{D_{1j}}{Q_{1j}} C_{1j} + \frac{1}{2} \sum_{j=1}^{n} h_{1j} Q_{1j} \left(1 - \frac{D_{1j}}{P_j}\right), \tag{2.2}$$

$$KB_{02} = \sum_{j=1}^{n} \frac{D_{2j}}{Q_{2j}} C_{2j} + \frac{1}{2} \sum_{j=1}^{n} h_{2j} Q_{2j} \left(1 - \frac{D_{2j}}{P_j}\right), \tag{2.3}$$

while the supplier inventory cost is as follow:

$$KS_0 = \sum_{j=1}^{n} \frac{D_{1j}}{Q_{1j}} A_j + \sum_{j=1}^{n} \frac{D_{2j}}{Q_{2j}} A_j. \tag{2.4}$$

Given that each supplier acts separately, if we minimize the corresponding cost, we have:

$$Q_{1j} = \sqrt{\frac{2D_{1j}C_{1j}}{h_{1j}\left(1 - \frac{D_{1j}}{P_j}\right)}}, \tag{2.5}$$

$$Q_{2j} = \sqrt{\frac{2D_{2j}C_{2j}}{h_{2j}\left(1 - \frac{D_{2j}}{P_j}\right)}}. \tag{2.6}$$

These values are obtained by setting the first derivative of Eq. (2.2) and Eq. (3.3) with respect to Q_{1j} and Q_{2j} equal to 0, respectively.

Therefore, the total cost per time unit for the supplier and the two buyers in a traditional supply chain is:

$$
\begin{aligned}
TC_{noVMI} = {} & \sum_{j=1}^{n} \left(\sqrt{\frac{D_{1j}h_{1j}\left(1-\frac{D_{1j}}{P_j}\right)}{2C_{1j}}} + \sqrt{\frac{D_{2j}h_{2j}\left(1-\frac{D_{2j}}{P_j}\right)}{2C_{2j}}} \right) A_j \\
& + \sum_{j=1}^{n} \sqrt{\left(2C_{1j}D_{1j}h_{1j}\left(1-\frac{D_{1j}}{P_j}\right)\right)} + \sum_{j=1}^{n} \sqrt{\left(2C_{2j}D_{2j}h_{2j}\left(1-\frac{D_{2j}}{P_j}\right)\right)}.
\end{aligned} \tag{2.7}
$$

2.4.2 VMI

Now, the case in which the supplier and buyer have agreed to apply the VMI system is considered. In this case, the supplier has full information on demand and is responsible for managing the inventory for both parties. The corresponding total cost is:

$$
\begin{aligned}
TC_{VMI} = \sum_{j=1}^{n} \frac{(A_j + C_{1j} + C_{2j})}{T} & + \frac{1}{2}\sum_{j=1}^{n} h_{1j}D_{1j}T\left(1-\frac{D_{1j}}{P_j}\right) \\
& + \frac{1}{2}\sum_{j=1}^{n} h_{2j}D_{2j}T\left(1-\frac{D_{2j}}{P_j}\right).
\end{aligned} \tag{2.8}
$$

In this situation, it is assumed $T_1 = T_2 = ... = T_n = T$. Let us take the derivative with respect to T and set it to zero. Eq. (2.9) yields the optimal T:

$$
T = \sqrt{\frac{2(A_j + C_{1j} + C_{2j})}{h_{1j}D_{1j}\left(1-\frac{D_{1j}}{P_j}\right) + h_{2j}D_{2j}\left(1-\frac{D_{2j}}{P_j}\right)}}. \tag{2.9}
$$

After replacing T in Eq. (2.8) we have:

$$
TC_{VMI} = \sum_{j=1}^{n} \sqrt{2(A_j + C_{1j} + C_{2j})\left(h_{1j}D_{1j}\left(1-\frac{D_{1j}}{P_j}\right) + h_{2j}D_{2j}\left(1-\frac{D_{2j}}{P_j}\right)\right)}. \tag{2.10}
$$

2.4.3 Comparison

Now, for comparison of VMI and traditional system, we wonder when $TC_{VMI} \leq TC_{noVMI}$:

$$
\begin{aligned}
&\sum_{j=1}^{n} \sqrt{2(A_j + C_{1j} + C_{2j})\left(h_{1j}D_{1j}\left(1 - \frac{D_{1j}}{P_j}\right) + h_{2j}D_{2j}\left(1 - \frac{D_{2j}}{P_j}\right)\right)} \\
&\leq \sum_{j=1}^{n} \left(\sqrt{\frac{D_{1j}h_{1j}\left(1 - \frac{D_{1j}}{P_j}\right)}{2C_{1j}}} + \sqrt{\frac{D_{2j}h_{2j}\left(1 - \frac{D_{2j}}{P_j}\right)}{2C_{2j}}} \right) A_j \\
&+ \sum_{j=1}^{n} \sqrt{\left(2C_{1j}D_{1j}h_{1j}\left(1 - \frac{D_{1j}}{P_j}\right)\right)} + \sum_{j=1}^{n} \sqrt{\left(2C_{2j}D_{2j}h_{2j}\left(1 - \frac{D_{2j}}{P_j}\right)\right)}.
\end{aligned}
\tag{2.11}
$$

After squaring Eq. (2.11) and simplifying we have a quadratic equation $aA_j^2 + bA_j + c$ as follows:

$$
\begin{aligned}
&\left(\sqrt{\frac{D_{1j}h_{1j}\left(1 - \frac{D_{1j}}{P_j}\right)}{2C_{1j}}} + \sqrt{\frac{D_{2j}h_{2j}\left(1 - \frac{D_{2j}}{P_j}\right)}{2C_{2j}}} \right)^2 A_j^2 \\
&+ 2\left(\sqrt{\frac{C_{1j}}{C_{2j}}} + \sqrt{\frac{C_{2j}}{C_{1j}}}\right)\left(D_{1j}D_{2j}h_{1j}h_{2j}\left(1 - \frac{D_{1j}}{P_j}\right)\left(1 - \frac{D_{2j}}{P_j}\right)\right) A_j \\
&- \left(2C_{1j}D_{2j}h_{2j}\left(1 - \frac{D_{2j}}{P_j}\right) + 2C_{2j}D_{1j}h_{1j}\left(1 - \frac{D_{1j}}{P_j}\right)\right. \\
&\left. -4\left(C_{1j}C_{2j}D_{1j}D_{2j}h_{1j}h_{2j}\left(1 - \frac{D_{1j}}{P_j}\right)\left(1 - \frac{D_{2j}}{P_j}\right)\right)\right) \geq 0.
\end{aligned}
\tag{2.12}
$$

In $ax^2 + bx + c$ if $x \leq \frac{-b-\sqrt{b^2-4ac}}{2a}$ or $x \geq \frac{-b+\sqrt{b^2-4ac}}{2a}$ then $ax^2 + bx + c$ is same sign with a.

Then the $\Delta = b^2 - 4ac$ is:

$$
\begin{aligned}
b^2 - 4ac =& D_{1j}D_{2j}h_{1j}h_{2j}\left(1 - \frac{D_{1j}}{P_j}\right)\left(1 - \frac{D_{2j}}{P_j}\right)\left(\frac{C_{1j}}{C_{2j}} + \frac{C_{2j}}{C_{1j}}\right) \\
&+ \frac{C_{1j}}{C_{2j}}D_{2j}^2h_{2j}^2\left(1 - \frac{D_{2j}}{P_j}\right)^2 + \frac{C_{2j}}{C_{1j}}D_{1j}^2h_{1j}^2\left(1 - \frac{D_{1j}}{P_j}\right)^2.
\end{aligned}
\tag{2.13}
$$

So, result of proposed model shows that when supplier faces two buyers, VMI system acts better than traditional system only when

$$
\begin{aligned}
A_j \geq\ & \frac{-\left(\sqrt{\frac{C_{1j}}{C_{2j}}}+\sqrt{\frac{C_{2j}}{C_{1j}}}\right)\left(D_{1j}D_{2j}h_{1j}h_{2j}\left(1-\frac{D_{1j}}{P_j}\right)\left(1-\frac{D_{2j}}{P_j}\right)\right)\left(1-\frac{D_{1j}}{P_j}\right)^2}{\left(\sqrt{\frac{D_{1j}h_{1j}\left(1-\frac{D_{1j}}{P_j}\right)}{2C_{1j}}}+\sqrt{\frac{D_{2j}h_{2j}\left(1-\frac{D_{2j}}{P_j}\right)}{2C_{2j}}}\right)^2} \\
& + \frac{\sqrt{\left(D_{1j}D_{2j}h_{1j}h_{2j}\left(1-\frac{D_{1j}}{P_j}\right)\left(1-\frac{D_{2j}}{P_j}\right)\right)\left(\sqrt{\frac{C_{1j}}{C_{2j}}}+\sqrt{\frac{C_{2j}}{C_{1j}}}\right)+\frac{C_{1j}}{C_{2j}}D_{2j}^2h_{2j}^2\left(1-\frac{D_{2j}}{P_j}\right)^2+\frac{C_{2j}}{C_{1j}}D_{1j}^2h_{1j}^2\left(1-\frac{D_{1j}}{P_j}\right)^2}}{\left(\sqrt{\frac{D_{1j}h_{1j}\left(1-\frac{D_{1j}}{P_j}\right)}{2C_{1j}}}+\sqrt{\frac{D_{2j}h_{2j}\left(1-\frac{D_{2j}}{P_j}\right)}{2C_{2j}}}\right)^2}.
\end{aligned}
\tag{2.14}
$$

2.5 The case with capacity constraint

As it is mentioned earlier, our goal is to determine the optimum value of Q_{ij} such that the total cost is minimized and the constraint is satisfied. Hence, the problem after replacing $T = Q/D$ in Eq. (2.8) is formulated as follows:

$$\begin{aligned}
&\min \sum_{j=1}^{n} \frac{\big(A_j + C_{1j} + C_{2j}\big)\big(D_{1j} + D_{2j}\big)}{Q_{1j} + Q_{2j}} \\
&\quad + \frac{1}{2} \sum_{j=1}^{n} \frac{D_{1j}}{D_{1j} + D_{2j}} (Q_{1j} + Q_{2j}) h_{1j} \left(1 - \frac{D_{1j}}{P_j}\right) \\
&\quad + \frac{1}{2} \sum_{j=1}^{n} \frac{D_{2j}}{D_{1j} + D_{2j}} (Q_{1j} + Q_{2j}) h_{2j} \left(1 - \frac{D_{2j}}{P_j}\right) \qquad (2.15) \\
&s.t. \\
&\quad \sum_{j=1}^{n} f_j \left(Q_{1j} \left(1 - \frac{D_{1j} + D_{2j}}{P_j}\right) + Q_{2j} \left(1 - \frac{D_{1j} + D_{2j}}{P_j}\right) \right) \leq F, \\
&\quad Q_{ij} \geq 0 \quad j = 1, ..., n, i = 1, 2.
\end{aligned}$$

The objective function expresses the minimization of the total cost of the supply chain. Constraints show the warehouse space limitation. Finally, Q_{ij} is the non negative decision variable.

2.6 Solution Algorithms

GA and PSO are employed to solve the proposed nonlinear integer programming model.

2.6.1 Genetic Algorithms (GAs)

The genetic algorithm is an algorithm for intelligent search based on the Darwinian principle of natural selection. This technique was first introduced by Holland [54]. The usual form of GA was described by Goldberg [48]. Researchers observed adaptation, survival, self-healing, guidance, reproduction and other characteristics of natural systems and to note how nature solves its problems, thought to mimic natural methods to solve difficult problems and design systems. Genetic algorithm is one of the most innovative search algorithms that are used for optimizing different functions (see Gen and Cheng [45]. GA is a part of evolutionary computation that is part of the Artificial Intelligence. Genetic algorithms are stochastic search techniques which work based on natural selection and natural genetics. The genetic algorithm is the means by which the machine can simulate the mechanism of natural selection. The act of searching the problem space is used to find the optimal answer, but it is not necessarily better. GA typically consists of three genetic operators of reproduction, crossover and mutation. These operators have the capability to manage chromosomes toward better usability (see Man et al. [64]). Since in our problem the parameters are real numbers, a real coded GA is used, in which the chromosome is defined as an array of real numbers with the mutation (P_m) and crossover (P_c) operators. Here, the mutation (P_m) can change the value of a real number randomly, and the crossover (P_c) can take place only at the boundary of two real numbers. More details of proposed GA is shown in Figure 2.1 (see Raie and Rashtchi [71]).

Since in our problem the parameters are real numbers, a real coded GA is used, in which the chromosome is defined as an array of real numbers with the mutation and crossover operators. Here, the mutation can change the value of a real number randomly, and the crossover can take place only at the boundary of two real numbers.

2.6.2 Particle Swarm Optimization (PSO)

PSO is a collective intelligence based algorithm, either a solution to the optimization problem in search space or modeling social behavior when there is a target. PSO is a computer-based algorithm to solve the population problem. PSO is a kind of collective intelligence based on principles of social psychology; moreover, it provides insight into social behavior and helps to engineer applications (see Kennedy and Eberhart [59]). The PSO algorithm is inspired by social behavior of bird flocking or fish schooling. PSO shares many similarities with evolutionary computation techniques such as GA. The system is initialized with a population of random solutions and searches for optima by updating generations. However, unlike GA, PSO has no evolution operators such as crossover and mutation. In PSO, the potential solutions, called particles, fly through the problem space by following the current optimum particles. Compared to GA, the advantages of PSO are that PSO is easy to implement and there are few parameters to adjust. PSO has been successfully applied in many areas (see Kennedy and Eberhart [58], Eberhart and Kennedy [39] and Eberhart and Shi [40]).

The standard PSO algorithm employs a population of particles. The particles fly through the $n-$dimensional domain space of the function to be optimized (in this paper, minimization is assumed). The state of each particle is represented by its position $\underline{x_i} = (x_{i1}, x_{i2}, ..., x_{in})$ and velocity $\underline{v_i} = (v_{i1}, v_{i2}, ..., v_{in})$, the states of the particles are updated. The flow chart of the procedure is shown in Figure 2.2 (see Haupt and Haupt [52]). During every iteration, each particle is updated by following two best values. The first one is the position vector of the best fitness. This particle has achieved so far. The fitness value $\underline{p_i} = (p_{i1}, p_{i2}, ..., p_{in})$ is also stored. This position is called $p$best. Another best position that is tracked by the particle swarm optimizer is the best position, obtained so far, by any particle in the population. This best position is the current global best $\underline{p_g} = (p_{g1}, p_{g2}, ..., p_{gn})$ and is called $g$best. At each time step, after finding the two best values, the particle updates its velocity and position according to Eq. (2.16) and Eq. (2.17).

$$v_i^{k+1} = wv_i^k + s_1 r_1 (pbest_i - x_i^k) + s_2 r_2 (gbest_k - x_i^k), \tag{2.16}$$

$$x_i^{k+1} = x_i^k + v_i^{k+1}, \tag{2.17}$$

where v_i^{k+1} is the velocity of particle number i at the $(k+1)^{th}$ iteration, x_i^k is the current particle (solution or position). r_1 and r_2 are random numbers between 0 and 1. S_1 is the self confidence (cognitive) factor; S_2 is the swarm confidence (social) factor. Usually S_1 and S_2 are in the range from 1.5 to 2.5; w is the inertia factor that takes values downward from 1 to 0 according to the iteration number (see Haupt and Haupt [52]). When a predetermined termination condition is reached, p_g is returned as the optimal value found.

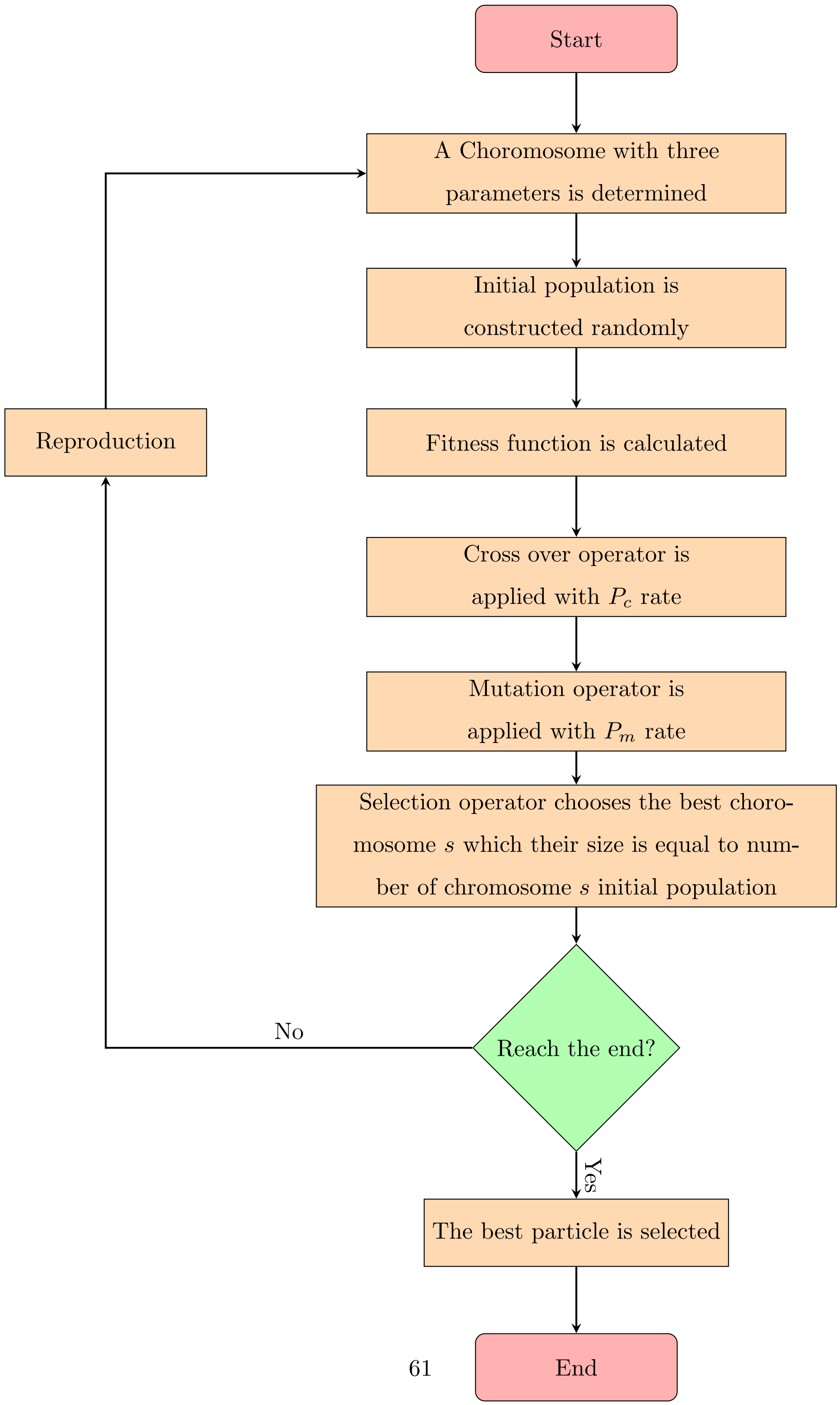


Figure 2.1: Genetic Algorithm flow chart.

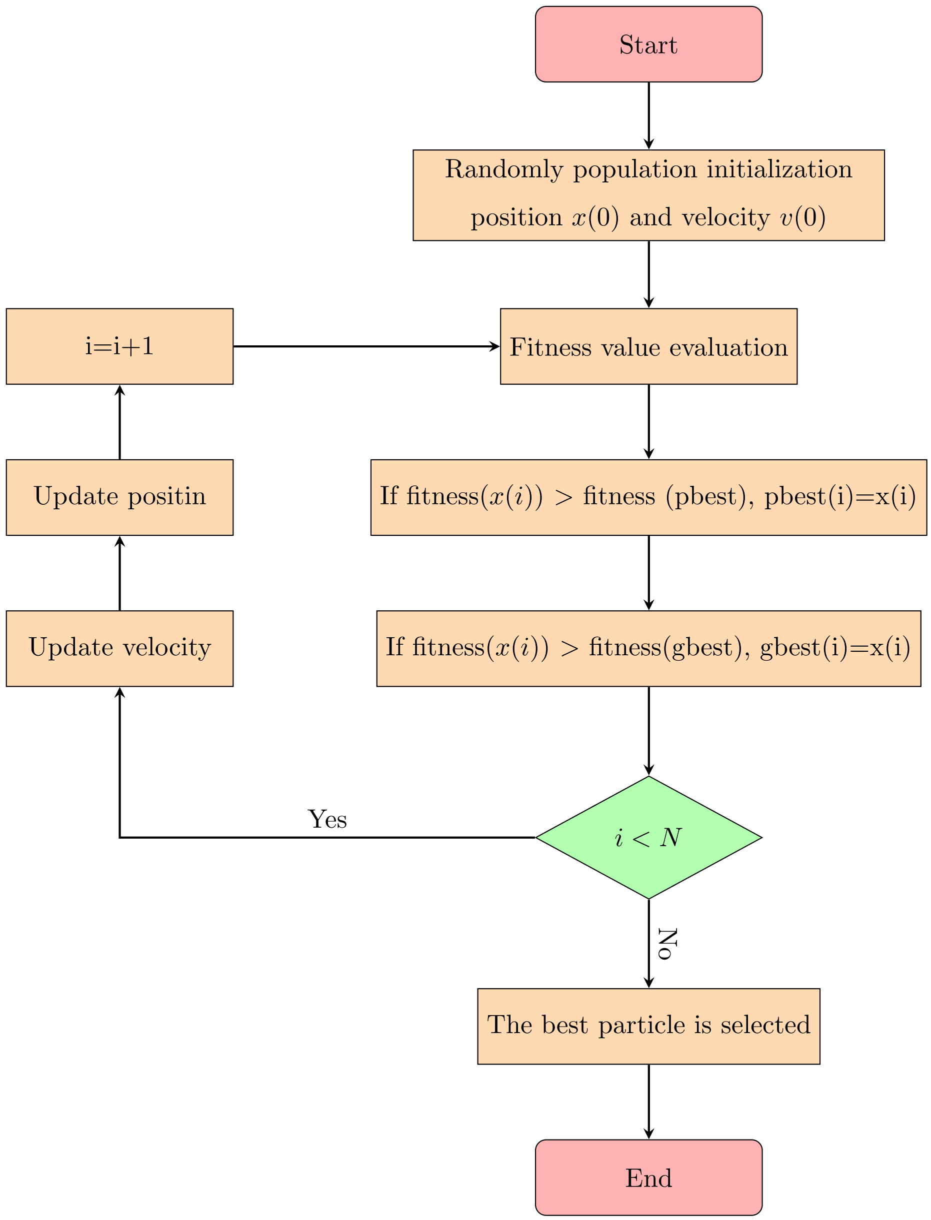


Figure 2.2: PSO flow chart.

2.7 A Numerical Example

In order to demonstrate the application of the proposed algorithms and study their performances, we propose a numerical example in this section. The data of the example is given in Table 2.1. Moreover, $F = 16000$.

Product	D_1	D_2	P	C_1	C_2	A	h_1	h_2	f
1	23	20	61	20	18	68	7	5	36
2	17	20	57	22	28	47	8	4	25
3	22	22	65	24	26	64	4	5	26
4	19	20	59	27	23	55	5	8	24
5	19	30	85	34	31	69	7	6	23
6	18	24	70	29	27	58	8	9	24
7	20	21	60	25	18	52	6	7	28
8	21	30	78	26	26	71	7	3	26
9	20	20	64	26	21	59	2	3	27
10	22	20	57	15	21	73	5	6	36
11	28	19	75	19	23	77	9	6	29
12	24	27	68	31	29	62	5	8	34

Table 2.1: Data

After trying different values of the parameters, we realized that the best results of the proposed algorithms are obtained with the parameters according to Table 2.2. The PSO results are compared with the GA. Number of population size is $n = 200$ and $n = 50$, respectively. For PSO, $S_1 = 1.8$, $S_2 = 2.2$, w starts at 1 and decreases until reaching 0 at the end of the run. For the sake of fair comparison of performance between the two algorithms, the stopping criterion is made the same. $P_m = 0.12$ and $P_c = 0.9$ are used to get the best results of GA. The mating is performed using single point cross over.

Optimization technique	Population size	Iteration number	Parameters used
GA	200	100	$P_m = 0.12$, $P_c = 0.9$
PSO	50	100	$S_1 = 1.8$, $S_2 = 2.2$

Table 2.2: Parameters adapted for GA and PSO

In this study, each chromosome represents the amount of goods ordered by the buyers. A matrix is assigned for each order of buyers. For example, chromosome is a 2×5 matrix for a problem which has 2 buyers and 5 products. A typical form of a chromosome is presented as follow:

$$\begin{bmatrix} 15 & 10 & 67 & 20 & 38 \\ 12 & 7 & 32 & 45 & 50 \end{bmatrix}.$$

The initial population is randomly generated regarding the population size that varies between 1 and maximum order. In this study, we use the additive form of the penalty function ($Pen(S)$) and the fitness function ($fitn(S)$) with the following form:

$$\begin{aligned} &fitn(S) = f(S) + Pen(S) \\ &Pen(S) = 0 \quad \text{if S is feasible,} \\ &Pen(S) > 0 \quad \text{otherwise.} \end{aligned} \tag{2.18}$$

where $f(S)$ is the objective function in Eq. (2.15) and S represents a solution. In this approach, we search for the solution that minimizes $fitn(S)$. A "roulette wheel selection" procedure is applied for the selection operator of this research. This selection approach is based on the concept of selection probability for each individual proportional to the fitness value. Mutation rate = 0.12

and Crossover rate = 0.9 are used to get the best results of GA. The mating is performed using single point crossover. Finally, the proposed GA is run for a fixed number of generations.

The best results obtained by the genetic algorithm and particle swarm optimization are as follows:

For GA:

$$\begin{bmatrix} 1 & 3 & 47 & 4 & 126 & 89 & 25 & 25 & 42 & 67 & 2 & 5 \\ 108 & 25 & 72 & 58 & 40 & 25 & 32 & 112 & 37 & 48 & 44 & 63 \end{bmatrix}.$$

For example, this solution indicates that order quantity for 5^{th} product by the 2^{th} buyer is 40 or order quantity for the 10^{th} product by the 1^{th} buyer is 67 and fitness is equal to 2951.

For PSO:

$$\begin{bmatrix} 11 & 22 & 81 & 4 & 32 & 92 & 28 & 66 & 62 & 42 & 31 & 40 \\ 111 & 71 & 114 & 37 & 31 & 64 & 20 & 11 & 92 & 45 & 74 & 105 \end{bmatrix},$$

having fitness equal to 2937.

Figure 2.3 shows the convergence of proposed PSO and GA algorithm. It is observed that the variation of the fitness during the PSO and GA run for the best case and shows the swarm of optimal variables. It can be seen that PSO is better than GA.

In order to have a benchmark the model (2.15) is solved by using LINGO which the best results obtained are as follows:

$$\begin{bmatrix} 24 & 21 & 29 & 21 & 27 & 20 & 21 & 31 & 44 & 25 & 23 & 27 \\ 24 & 21 & 29 & 21 & 27 & 20 & 21 & 31 & 25 & 25 & 23 & 27 \end{bmatrix},$$

having fitness equal to 2347.

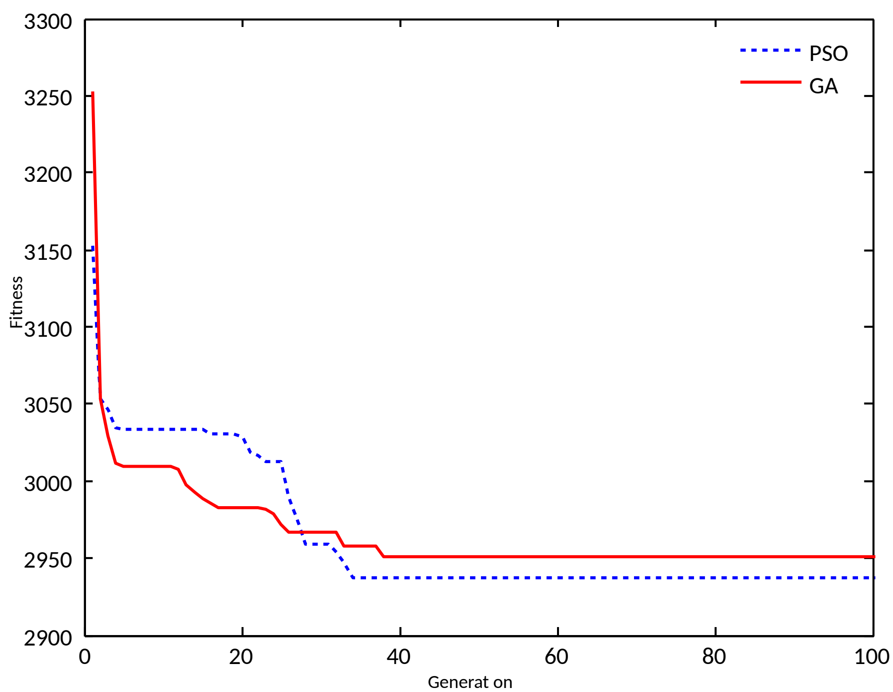


Figure 2.3: Convergence of optimization algorithms for an increasing number of generation.

2.8 Computational Results

In this section, a set of problem instances has been considered. The data related to parameters and the computational results are shown in Table 2.3.

The table is organized as follows. The second column shows the number of products, the third the available warehouse space at the supplier for all products, the forth, the fifth and the sixth the best results of the proposed approaches, respectively, the seventh, the eighth and the ninth the computational times (seconds) required by the three approaches, respectively and the last three columns the percent cost increase of LINGO with respect to GA and PSO method and GA method with respect to PSO method, respectively.

LINGO allows us to have a better solution, but the computational time is significantly larger than GA and PSO. LINGO requires a larger computational time than the heuristics just to provide a local optimum.

Instance	Number of Products	F	$LINGO$	GA	PSO	$LINGO$	GA	PSO	1-(GA/LINGO)	1-(PSO/LINGO)	1-(GA/LINGO)
1	3	1200	525	525	525	2	2	2	0.00%	0.00%	0.00%
2	4	2000	901	901	901	10	10	10	0.00%	0.00%	0.00%
3	5	3000	985	1010	1008	947	17	17	2.53%	2.33%	0.19%
4	6	4200	1289	1342	1335	1518	30	30	4.11%	3.56%	0.52%
5	7	5500	1456	1620	1609	2700	45	45	11.2%	10.50%	0.68%
6	8	7200	1723	1914	1902	3300	55	52	11.08%	10.38%	0.63%
7	10	12000	2159	2696	2683	3954	125	110	24.87%	24.27%	0.48%
8	12	16000	2347	2951	2937	4351	178	169	25.73%	25.13%	0.47%
9	14	18500	3079	3750	3730	5514	210	202	21.79%	21.14%	0.53%
10	15	22000	3825	4127	4103	9482	325	300	7.89%	7.26%	0.58%

Table 2.3: Best Results of the Proposed Algorithms

2.9 Conclusion

This research dealt with the mathematical issues of a two-echelon vendor-buyer supply chain model under VMI. A multi-product EPQ model is developed and this model is expanded by assuming that the warehouse space is limited for all products. Under these conditions, the problem is formulated as a non-linear integer-programming model and PSO and GA are proposed to solve it. In the end, a numerical example was presented to not only demonstrate the application of proposed methodology but also to evaluate the effectiveness of the PSO and GA algorithms in solving the multi-product EPQ model of this research. The results indicated the relative superiority of PSO against GA. Some recommendations for future research are as follows:

1. Other search-heuristic algorithms such as Simulated Annealing, Ant Colony, Bacterial Foraging, Frog Leaping Algorithm, Coco Algorithm and so on may be used to solve the problem at hand and compare the results with the available results of algorithms in the paper.

2. Some other limitations such as shortage, budget limitation, etc. can be added to the model.

3. The model can be extended to involve random variable or fuzzy natures of some parameters such as the demand rate.

Chapter 3

Minimizing Earliness and Tardiness Penalties in a Single Machine Scheduling Problem

3.1 Introduction

Sequencing and scheduling are decision-making problems which play a critical role in manufacturing as well as in service industries. Scheduling is one of the research topics that scholars have studied widely because the methodology for finding an optimal or a good schedule can be applied to many real cases. Companies have to meet shipping dates committed to the customers and they have to schedule activities to use the available resources in an efficient manner (see Pinedo [70]).

The study of scheduling dates back to 1950s. Researchers in Operations research, Industrial Engineering and Management were faced with the problem of managing various activities. With the advent of the Theory of Computational Complexity researchers began to realize that many of these problems may be inherently difficult to solve. In the 1970s, many scheduling problems were shown to be NP-hard. In the 1980s, several different directions were pursued in Academia and Industry. One direction was the development and analysis of approximation algorithms. Another direction was the increasing attention paid to stochastic scheduling problems. After almost 60 years, there is now an astounding body of knowledge in the field of scheduling (see Leung [62] and Hallah [50]).

Nowadays and in many practical situations, many manufacturing systems are adopting Just-In-Time (JIT) concepts. Companies that use JIT have developed scheduling policies with the objective of producing each customer order exactly at the required time, due to direct and indirect costs incurred by completing jobs before or after their due dates. In JIT manufacturing systems, jobs are needed to arrive and proceed just as they are needed, not earlier nor later. Earliness causes high inventory costs and lateness delays other jobs and gives a low service level.

The concept of penalizing both earliness and tardiness has opened a new and rapidly developing line of research in the scheduling field. Because the use of both earliness and tardiness penalties gives rise to a non-regular performance measure, it has led to new methodological issues in the design of solution procedures (see Szwarc [79]):

- Early/Tardy problem with common due date. Prescribing a common due date might represent a situation where several items represent a single customer's order, or it might be part of a final product and thus all the part should be ready at the same time for the final assembly to avoid undesired delay (see De et al. [34]).

- Early/Tardy problem with common due window. In many circumstances, jobs are sometimes associated with a period of time within which they incur no penalty since customers usually allow for a time interval within which products are delivered (see Mazzini and Armentano [65]).

- Early/Tardy problem with distinct due dates. Every job has its own distinct due date (see Baker and Scudder [6]).

In this chapter, we focus on the latter case.

The studied scheduling problem aims at finding a schedule of n jobs such that total Earliness/Tardiness is minimized. When each family contains only one job, the problem reduces to the traditional single machine total Earliness/Tardiness problem, known to be NP-hard (see Du and Leung [37]). An integer nonlinear programming model of the problem is proposed. The problem we studied is new, so no benchmark instances are available.

The remaining chapter is organized as follows. In Section 3.2 some issues in sequencing and scheduling are reviewed. Then, in Sections 3.3 and 3.4 the problem is described, the structural properties of the problem are introduced and a model is developed. Moreover, the computational results are presented in Section 3.5.

3.2 Literature Review

The earliest paper on the topic of distinct due dates was by Sidney [76], who provided an efficient algorithm to minimize the maximum earliness or tardiness penalty on a single machine where jobs have a restricted variety of due dates. This algorithm was improved by Lakshminarayanan et al. [61]. Panwalkar et al. [67] presented an analytical approach for single machine scheduling problem with due date assignment. They considered the problem of assigning individual job due dates and identifying a sequence to minimize weighted earliness, tardiness and lead time costs. Garey et al. [44] considered the problem of minimizing total (un-weighted) earliness and tardiness on a single machine where due dates may vary between jobs. This important paper contains one of the first proofs of NP-completeness for a single machine scheduling problem with a non regular performance measure. Yano and Kim [86] studied several dominance properties for sequencing jobs with penalty weights proportional to processing times. They gave a linear programming formulation of the problem and developed a dynamic programming algorithm to solve it.

3.3 Problem Definition

Classical assumptions in scheduling, according to French [43] and Pinedo [70], are:

1. No pre-emption or cancellation for jobs is allowed.
2. The processing times are independent of the schedule.
3. Machine never breakdowns and is available throughout the scheduling period.
4. There is no randomness. In particular:
   (a) The number of jobs is deterministic.
   (b) The processing times are deterministic.
   (c) The ready times are deterministic.
   (d) All other quantities needed to define any particular problem are known.
5. All jobs are ready to be processed at time $t = 0$.

In Single machine Earliness/Tardiness problem, n jobs ready at time zero on a single machine are scheduled with the objective to minimize the total penalties for earliness and tardiness. Each job j have a processing time d_j and a due date dd_j. The single machine is assumed to be available continuously and all the necessary information, such as processing time, is known advance. The completion time for a job j is defined as ds_j. The symbols e_j and t_j represent the earliness and tardiness of job j respectively, where these quantities are defined by the following:

$$e_j = max(0, dd_j - ds_j) \quad e_j = (dd_j - ds_j)^+, \tag{3.1}$$

$$t_j = max(0, ds_j - dd_j) \quad t_j = (ds_j - dd_j)^+. \tag{3.2}$$

Table 3.1 shows the parameters and variables of INLP model.

We introduce the following unit penalties: α_j is the unit earliness penalty and β_j is the unit tardiness penalty. With the assumption that the penalty functions are linear, the Earliness/Tardiness objective function can be written as follows:

$$f(s) = \sum_{j=1}^{n} [\alpha_j e_j + \beta_j t_j], \tag{3.3}$$

Parameters	Variables
j = Index of jobs ($j = 1, 2, ..., n$)	d_i = Processing time of a proposed job i
i = Index of counter of sequenced job ($i = 1, 2, ..., n$)	dd_i = Due date of a proposed job i
d_j = Processing time of job j	ds_i = Completion time of a proposed job i
dd_j = Due date of job j	e_i = Earliness penalty of a proposed job i
ds_j = Completion time of job j	t_i = Tardiness penalty of a proposed job i
e_j = Earliness penalty of job j (α)	
t_j = Tardiness penalty of job j (β)	

Table 3.1: The parameters and variables of INLP model

where s is a schedule (job sequence). The aim of the problem is to find a schedule that minimizes objective function (3.3).

3.4 Modeling

Below is presented an Integer Non-Linear Programming (INLP) model to solve the problem described in the previous section.

Let i and j represent the i^{th} job ($i = 1, 2, ..., n$) and j^{th} job ($j = 1, 2, ..., n$). We have two kind of jobs: Discrete jobs which are not similar to each other and have differences in terms of processing time and so on, sequential jobs in which each of them should be done continuously, a job is not started or, if is started, be sure is finished before starting the next one.

Variables:

$$q_i = \begin{cases} 1 & \text{if job } i \text{ is completed before the due date} \\ 0 & \text{otherwise.} \end{cases}$$

$$z_{ij} = \begin{cases} 1 & \text{if job } j \text{ is processed after job } i \\ 0 & \text{otherwise.} \end{cases}$$

The integer nonlinear programming model for E/T problems of single machine scheduling is:

$$\min z = \sum_{i=1}^{j} \big((-1)(ds_i - dd_i)(e_i)(q_i) + (ds_i - dd_i)(t_i)(1 - q_i)\big) \tag{3.4}$$

$$\sum_{j=1}^{n} z_{ij}.d_j = d_i \qquad \forall i, \tag{3.5}$$

$$\sum_{j=1}^{n} z_{ij} = 1 \qquad \forall i, \tag{3.6}$$

$$\sum_{i=1}^{j} z_{ij} = 1 \qquad \forall j, \tag{3.7}$$

$$\sum_{j=1}^{n} z_{ij}.dd_j = dd_i \qquad \forall i, \tag{3.8}$$

$$\sum_{j=1}^{n} z_{ij}.e_j = e_i \qquad \forall i, \tag{3.9}$$

$$\sum_{j=1}^{n} z_{ij}.t_j = t_i \qquad \forall i, \tag{3.10}$$

$$ds_i = \sum_{j=1}^{i} d_j \qquad \forall i, \tag{3.11}$$

$$(ds_i - dd_i)(q_i) \leq 0 \qquad \forall i, \tag{3.12}$$

$$(ds_i - dd_i)(1 - q_i) \geq 0 \qquad \forall i, \tag{3.13}$$

$$ds_i \geq 0, \quad dd_i \geq 0, \quad e_i \geq 0, \quad t_i \geq 0, \quad d_i \geq 0 \qquad \forall i, \tag{3.14}$$

$$q_i \in \{0, 1\}, \quad z_{ij} \in \{0, 1\} \qquad \forall i, j. \tag{3.15}$$

In this model, the goal of the objective function (3.4) is to minimize the total penalties of earliness and tardiness. Constraints (3.5) determines that sequenced job i has a processing time of d_i. It means that if job j is proceeded after job i or if job j with sequence i is accepted, then $z = 1$. For example, if there are 3 jobs in sequence, the whole process is collected. Constraints (3.6) and (3.7) ensure that each job is selected one time. The next sets of constraints (3.8)-(3.10) determine sequenced job i has due date, earliness penalty and tardiness penalty of dd_i, e_i and t_i. Also determine that completing jobs before or after their due dates have penalty. If earliness and tardiness happen, these constraints consider the penalties. If $z = 1$ this means that the job is on time and there is no penalty, otherwise, the penalty is considered. Constraints (3.11) is summation of processing time or completion time of a proposed job i and calculates the completion time from the first position to the last position of the proposed sequence. The next sets of constraints (3.12) and (3.13) ensure that if $(ds_i - dd_i)$ becomes negative then q_i will be equal to one and e_i would be selected, and if $(ds_i - dd_i)$ becomes positive then q_i will be equal to zero and t_i would be selected. Finally, the last sets of constraints (3.14) and (3.15) ensures that non-negativity and binary conditions are hold.

3.5 Computational results

3.5.1 Tabu Search

Tabu search (TS) is a local search based on optimization method: the search moves from one solution to another, choosing at each iteration the best not forbidden solution in a given neighborhood of the current solution. Tabu Search has a powerful search capability for many optimization problems. This method forbids solutions with certain attributes, with the goals of preventing cycling and guiding the search towards unexplored regions. Without using the technique of forbidden solutions, starting from a local optimum s (i.e. a solution such that all elements in its neighborhood $N(s)$ have a greater cost than s), the method chooses the best solution in $N(s)$ and then at the next step falls back into s again, i.e. s is a local optimum. However, storing a given number of the last visited solutions in a forbidden list and testing if a candidate solution belongs to the list is generally too expensive, both for memory and for computational time requirements (see James and Buchanan [56]). Usually, a tabu list is defined, which stores only a subset of the visited solutions. A solution s' is considered forbidden if the current solutions can be transformed into s'' by applying one of the moves in the tabu list. In addition to a tabu status, a so-called aspiration criterion is associated with each move. If a current tabu move satisfies the associated aspiration criterion, it is considered an admissible move. For a precise and complete description of this method, the interested reader can refer to the papers by Glover [46] and [47].

3.5.2 Working Mechanism

In this section, we will explain the work mechanism of the Tabu Search method. The following parameters will be used in the Tabu solution method:

- TLS: the size of Tabu list.
- T: number of allowed iterations.
- AL: aspiration level.

Let Best Cost be the cost of the best solution (Best sequence) obtained so far. The way that Tabu Search works can be described in the following steps:

Step 1:

Start with an initial sequence, specify the number of iterations (T), the Tabu List Size (TLS) and the Aspiration Level (AL)

Step 2:

Create new sequences by generating two random numbers K and L, where:

- K is an integer number between 0 and n.
- L is an integer number between 0 and n.
- $K \neq L$.

Then, we swap the jobs in the K^{th} and the L^{th} positions in the current sequence and compute the corresponding cost. If the new cost is better than Best Cost, update Best Cost and keep the new sequence as Best Sequence. Otherwise, reset the sequence of the jobs.

Step 3:

Create a Tabu List including the last selection of K and L. The Tabu List saves the last moves, i.e., swaps of jobs in L^{th} and K^{th} positions. Every time we generate the random numbers K and L, we check whether they are in the list or not. The size of the Tabu List (TLS) is specified by the user and in the Tabu List we keep a record for the last TLS moves.

Step 4:

Repeat 2 and 3 for T times.

Before rejecting a bad sequence, generate a random number λ and check if λ is less than the AL specified. After running T iterations, the Best Cost and the Best Sequence are considered as a solution. A difficult part of the implementation of Tabu Search is to find the right Tabu List Size. No single rule gives good sizes for all classes of problems. If the TLS is too small, the search process may start cycling and if it is too large, the computational time increases a lot.

3.5.3 Tabu Search Parameters Selection

There are several parameters defining the TS method including Tabu List Size (TLS), Aspiration Level (AL) defined by the probability of accepting bad solution, Stopping Criterion and Initial Solution. In the following, we will discuss the selection of these parameters:

1. Tabu List Size (TLS): Several values (1, 2, , 7) were tested in preliminary tests which did not show significant clear difference in the performance of the algorithm.

2. Stopping Criterion: the stopping criteria should be set based on the problem size. Experiments are conducted to find the best stopping criterion in which two alternatives were considered:

   (a) Alternative 1: In this alternative, we will report the cost of the objective function when we have 200 Tabu iterations without improvement in cost. The cost of the objective function is improving by iterations, we will observe the iteration where we do not get any improvement for the last 200 iterations and we will compare the cost obtained by the solution of INLP Method.

   (b) Alternative 2: In this alternative we try to find the steady state of the TS solution. The steady state is the iteration where we reach the best cost and there is no improvement in the cost after this iteration. The process of finding the steady state for different problems sizes is presented in the next section.

3. Aspiration Level (AL): the aspiration criterion is a devise used to override the Tabu status. Experiments have been done to find the best aspiration level.

We will try to find the best TS parameters: TLS, Stopping Criterion and AL. The TS parameters are investigated by solving 10 test problems with different sizes. The results of the experiments conducted are represented in the following sections. We would like to find a relation between the sizes of the problem and the number of iterations needed to reach the steady state situation. The steady state iterations in TS are the iteration where we reach the best cost and there is no improvement in the cost after this iteration. To define the steady state for different problems sizes, we selected 10 test problems with different sizes and we solved each problem 25 times and we try to find the steady state situations for these problems. We used TLS equals to 7 and we set the AL to zero. We have selected 10 test problems and we run the TS for each problem for 25 times. In each run, we observed the solution at the following iteration: 10, 25, 50, 75, 100, 125, 150, 175, 200, 250, 300, 350, 400, 450, 500, 600, 700, , 9000 and 10000.

Table 3.2 shows a summary of the number or iterations needed to reach the steady state for each problem and the average for each problem size.

We now try to find a relation between the size of the problem and the number of iterations needed to reach the steady state for that particular size, number of jobs in an instance. We

Number of Test Problem	Number of Jobs	Average number of iterations to reach steady state
10	5	100
10	10	295
10	15	760
10	20	1800
10	25	2000
10	30	3710

Table 3.2: Average Number of Iterations to Reach Steady

will apply Nonlinear Regression to find the best curve fitting for the data resulted from the test problems by SPSS15. Figure 3.1 is a graphical representation of the regression model. It shows the optimal solution in order to find best criteria by decreasing the searching time for a solution.

The result of regression model is:

$$Y = 90X - 2.95X^2 + 0.145X^3 - 347.$$

Thus, for a problem with size 100, the number of iteration needed to reach the steady state is about 124153 iterations.

Aspiration level is a method used to override the Tabu status. To find the best aspiration level, we have done experiments on the same 10 test problems used in the previous section. We solved each test problem 10 times using different aspiration levels. The aspiration levels used are 0.05, 0.06, 0.07, 0.08, 0.09 and 0.1. In each solution, we observed the cost of the objective function and we recorded the iteration where we got the solution of the problem. The Tabu List Size is fixed to 7. The Table 3.3 represents the results of our experiments and shows the number of iterations needed to reach the solutions of the test problems (TP) 1 to 10 using different aspiration levels.

The minimum number of iterations for each problem is marked with asterisk, we apply the weighted average formula for them:

$$\text{Weighted average} = [(3 \times 0.05) + (2 \times 0.06) + (3 \times 0.07) + (1 \times 0.08) + (1 \times 0.1)] \div 10 = 0.066 \approx 0.07.$$

Aspiration Level	TP1	TP2	TP3	TP4	TP5	TP6	TP7	TP8	TP9	TP10
	n=10	n=10	n=10	n=10	n=10	n=10	n=10	n=10	n=10	n=10
0.05	139	136	256*	276*	585*	790	1204	1110	1727	1989
0.06	105	103	261	502	807	557	1047*	2046	1363	1388*
0.07	88	85*	370	320	908	250*	1272	1543	1312*	1938
0.08	144	269	266	1011	1374	2255	6377	775*	1895	1876
0.09	142	187	299	478	761	721	1409	1108	2400	2234
0.1	59*	158	477	527	977	2680	1265	1107	1965	1449

Table 3.3: Results of the experiments for different aspiration levels.

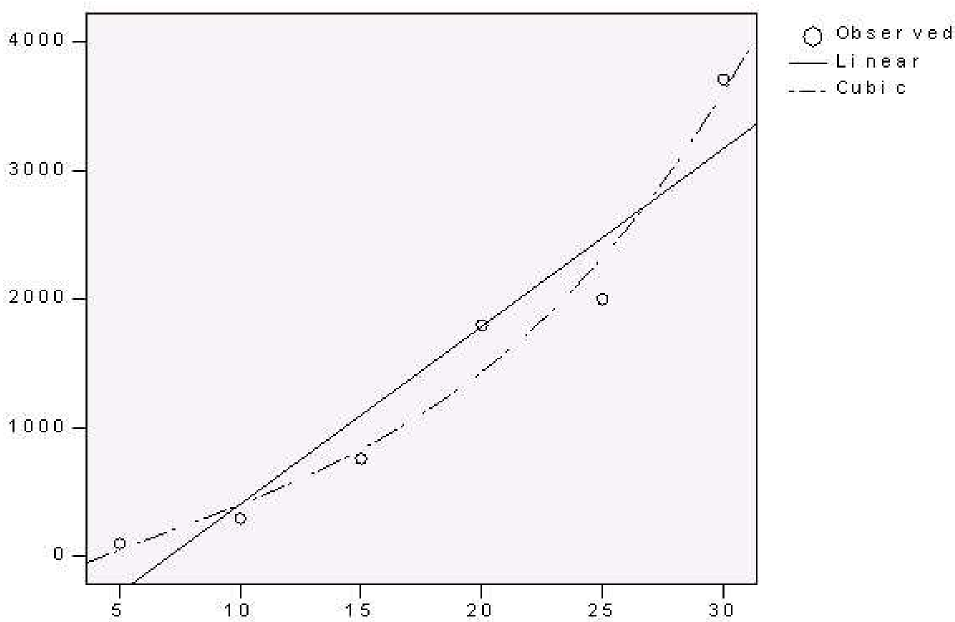


Figure 3.1: Comparison of Linear Regression and Cubic Regression.

This solution indicates the repeating numbers in each row which are marked with an asterisk and divided by the total sum of abundance.

In conclusion, the Aspiration Level (AL) of 0.07 gives a reasonable performance with respect to the number of iterations required to meet the optimal solution.

3.5.4 Experimental Results

In this paper, we generated 60 problems with different sizes: 5, 10, 15, 20, 25 and 30 jobs. We generated 10 problems for each size. We can classify our problems according to the number of jobs. The problems are small, medium and large. Small problems have 5 and 10 jobs, medium problems have 15 and 20 jobs and large problems have 25 and 30 jobs. An example of a small problem is shown in the Table 3.4. The average number of iterations of TS and INPL model in Lingo are shown in Table 3.5 and Table 3.6, respectively. For example, if the solution is obtained in 1000, 2000, 1000, 1000, 1800 iterations, then 1360 iterations is shown as the average number of iterations.

Jobs	1	2	3	4	5	6	7	8	9	10
Processing Time	15	18	9	16	3	11	10	21	7	25
Due Date	20	28	6	29	4	7	23	30	12	33
Earliness Penalty	4	7	8	1	12	6	3	7	8	2
Tardiness Penalty	4	2	5	6	9	5	5	6	2	5

Table 3.4: Sample of small problems

Instance	1	2	3	4	5	6	7	8	9	10
5	100	50	100	150	100	100	100	100	100	100
10	250	300	350	300	300	250	300	300	350	250
15	700	700	700	700	700	800	700	900	800	900
20	1800	1900	1800	1700	1700	1800	1800	1800	1900	1800
25	2000	1900	2000	2000	1900	2000	2000	2100	2000	2100
30	3600	3800	3800	3700	3800	3800	3700	3700	3500	3700

Table 3.5: Average number of iterations for the best cost in TS Method

Table 3.7 and Table 3.8 show the computational times in TS and INLP method, respectively. The values of the objective function in both methods are shown in Table 3.9 and Table 3.10 and the comparison of them respect to the percent of increase cost is shown in Table 3.11.

For small instances, the problem has been solved with Tabu Search Heuristic implemented under Lingo framework. For larger instances, the model has been solved by the INLP method. We considered that small problems in Lingo are solved rapidly, but medium and large problems are solved faster by Tabu Search method. Running INLP model took a lot of time, even more than 1 day. That is the reason why we put the stopping criteria equal to 200. Therefore, we have only local optimal solutions. Instead, we have global optimum for small size instances by using INLP.

Instance	1	2	3	4	5	6	7	8	9	10
5	35	23	52	89	28	30	45	41	31	39
10	276	349	678	312	574	298	323	586	1230	314
15	256322	315155	286530	415426	298855	345210	265560	484430	452330	569870
20	776525	806550	798562	665262	689508	745820	722350	759965	799860	691210
25	1218796	1181765	1212895	1465238	985274	1187410	1498563	1558960	1365220	1606987
30	2560254	3065205	2985463	2911562	2741089	3126508	2875428	2599600	2457418	2626599

Table 3.6: Average number of iterations for the best cost in INLP

Instance	1	2	3	4	5	6	7	8	9	10
5	2	2	2	2	2	2	2	2	2	2
10	2	2	2	2	2	2	2	2	2	2
15	3	3	3	3	3	3	3	3	3	3
20	7	7	7	6	6	7	7	7	7	7
25	7	7	7	7	7	7	7	8	7	8
30	13	14	14	14	14	14	14	14	13	14

Table 3.7: Time for the best cost in TS Method (seconds)

Instance	1	2	3	4	5	6	7	8	9	10
5	2	2	2	2	2	2	2	2	2	2
10	2	2	2	2	2	2	2	2	5	3
15	949	1167	1061	1539	1107	1279	984	1794	1675	2111
20	2876	2987	2958	2464	2554	2762	2675	2815	2962	2560
25	4514	4377	4492	5427	3649	4398	5550	5774	5056	5952
30	9482	11353	11057	10784	10152	11580	10650	9628	9102	9728

Table 3.8: Time for the best cost in INLP Method (seconds)

Instance	1	2	3	4	5	6	7	8	9	10
5	28	79	29	78	55	147	66	213	155	36
10	99	101	47	201	154	362	128	459	291	87
15	112	175	98	352	323	585	285	878	333	120
20	285	355	214	875	444	665	458	969	658	265
25	398	401	454	1620	589	898	556	1546	980	320
30	545	1111	665	1985	688	1005	787	2150	1450	451

Table 3.9: Best Cost in TS Method

Instance	1	2	3	4	5	6	7	8	9	10
5	28	79	29	78	55	147	66	213	155	36
10	99	101	47	201	154	362	128	459	291	87
15	122	191	109	382	350	635	311	951	362	130
20	326	405	244	999	508	760	529	1108	750	302
25	477	481	545	1942	709	1079	669	1855	1175	383
30	725	1478	887	2636	920	1336	1055	2859	1925	600

Table 3.10: Best Cost in INLP Method

Instance	1	2	3	4	5	6	7	8	9	10	Average
5	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
10	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
15	8.20%	8.38%	10.09%	7.85%	7.71%	7.87%	8.36%	7.68%	8.01%	7.69%	8.18%
20	12.58%	12.35%	12.30%	12.41%	12.60%	12.50%	13.42%	12.55%	12.27%	12.25%	12.52%
25	16.56%	16.63%	16.70%	16.58%	16.93%	16.77%	16.89%	16.66%	16.60%	16.45%	16.68%
30	24.83%	24.83%	25.03%	24.70%	25.22%	24.78%	25.40%	24.80%	24.68%	24.83%	24.91%

Table 3.11: Percent cost increase of INLP method with respect to TS method : 1 - (TS Method / INLP Method)

3.6 Conclusions

We considered the problem of scheduling n jobs that are ready at time zero on a single machine with the objective of minimizing the total penalties for earliness and tardiness. Each job has its own distinct due date. The problem is known as Single Machine Earliness/Tardiness scheduling problem with distinct due dates. We have formulated an INLP model and implemented a TS algorithm to solve it. In TS, we discussed the main steps and used random numbers generators and jobs swapping to find near-optimal solutions. We have done experiments to optimize the TS parameters: Tabu List Size (TLS), Stopping Criterion and the Aspiration Level. A nonlinear Regression model was used to estimate the number of iterations to reach the steady state situation. The best aspiration level resulted from the experiments is 0.07. In INLP model, we considered small problems in Lingo solved rapidly but medium and large problems solved very slower than TS method.

Test problems of different sizes have been created to test the proposed algorithm and compare between the two methods were conducted. Thus we can use two methods for small problems and propose Tabu Search for large problems. In the meantime, instances with large sizes, 100 jobs for example can be solved using proposed methods, but INLP is time consuming. The main results of this paper are: (1) develop Tabu Search model and present INLP model, (2) optimization of parameters in Tabu Search, experiments were conducted to find the best Stopping Criterion, Aspiration Level and Tabu List Size.

Appendix A

In this appendix we collect data, in order to demonstrate the application of the proposed algorithms in Chapter 2.

Product	D_1	D_2	P	C_1	C_2	A	h_1	h_2	f
1	22	21	66	20	18	52	4	7	28
2	18	20	57	24	28	47	5	4	25
3	20	22	65	22	26	64	6	5	26

Table A.1: Data with $F = 1200$

Product	D_1	D_2	P	C_1	C_2	A	h_1	h_2	f
1	22	22	70	24	26	64	4	7	26
2	17	20	57	22	28	47	8	4	25
3	28	19	75	19	23	77	9	5	29
4	20	21	65	25	18	52	6	6	28

Table A.2: Data with $F = 2000$

Product	D_1	D_2	P	C_1	C_2	A	h_1	h_2	f
1	24	30	98	34	31	69	9	7	23
2	20	20	80	24	22	47	8	8	25
3	22	19	90	26	30	77	6	9	29
4	18	25	95	25	19	64	7	4	26
5	25	20	92	18	20	52	5	6	28

Table A.3: Data with $F = 3000$

Product	D_1	D_2	P	C_1	C_2	A	h_1	h_2	f
1	30	18	95	23	27	58	5	9	24
2	19	20	85	24	26	64	4	5	26
3	22	17	87	21	18	52	6	7	28
4	18	22	97	28	31	69	9	6	23
5	28	27	99	19	28	47	7	4	25
6	26	25	100	35	23	77	8	6	29

Table A.4: Data with $F = 4200$

Product	D_1	D_2	P	C_1	C_2	A	h_1	h_2	f
1	26	27	130	29	24	62	4	8	34
2	28	20	110	28	35	47	9	4	25
3	19	24	105	27	22	58	6	9	24
4	17	19	115	23	26	77	7	6	29
5	20	30	120	31	21	69	5	6	23
6	22	22	117	26	34	64	5	5	26
7	30	21	125	18	19	52	4	7	28

Table A.5: Data with $F = 5500$

Product	D_1	D_2	P	C_1	C_2	A	h_1	h_2	f
1	30	24	125	24	26	71	8	7	26
2	21	19	115	19	25	52	9	6	28
3	22	20	117	26	24	64	6	4	26
4	19	25	120	18	19	58	5	9	29
5	24	23	116	22	29	62	6	8	24
6	27	22	135	25	31	64	7	5	34
7	30	26	127	35	34	69	4	7	23
8	20	30	110	20	22	47	3	8	25

Table A.6: Data with $F = 7200$

Product	D_1	D_2	P	C_1	C_2	A	h_1	h_2	f
1	22	30	125	18	15	73	5	9	36
2	22	25	120	22	24	64	4	6	26
3	28	26	122	32	19	77	9	7	29
4	19	17	130	24	34	69	7	7	23
5	20	19	105	26	25	52	6	8	28
6	24	28	119	19	31	62	5	4	34
7	21	21	124	20	26	71	7	5	26
8	20	24	115	28	26	59	2	4	27
9	17	22	110	21	22	47	8	3	25
10	18	20	107	25	29	58	8	9	24

Table A.7: Data with $F = 12000$

Product	D_1	D_2	P	C_1	C_2	A	h_1	h_2	f
1	23	20	61	20	18	68	7	5	36
2	17	20	57	22	28	47	8	4	25
3	22	22	65	24	26	64	4	5	26
4	19	20	59	27	23	55	5	8	24
5	19	30	85	34	31	69	7	6	23
6	18	24	70	29	27	58	8	9	24
7	20	21	60	25	18	52	6	7	28
8	21	30	78	26	26	71	7	3	26
9	20	20	64	26	21	59	2	3	27
10	22	20	57	15	21	73	5	6	36
11	28	19	75	19	23	77	9	6	29
12	24	27	68	31	29	62	5	8	34

Table A.8: Data with $F = 16000$

Product	D_1	D_2	P	C_1	C_2	A	h_1	h_2	f
1	30	25	147	18	22	72	3	7	34
2	19	20	179	21	32	53	9	8	29
3	17	24	168	25	24	62	8	4	36
4	22	22	144	29	25	51	6	5	27
5	28	17	154	26	21	65	7	7	22
6	34	21	180	20	28	55	5	8	25
7	21	30	192	23	19	45	4	6	32
8	26	19	211	24	20	70	2	7	26
9	20	23	135	22	30	68	3	2	34
10	18	21	190	19	15	77	9	5	25
11	28	19	226	19	23	65	6	9	23
12	24	27	184	31	29	59	8	5	36
13	30	27	195	15	26	63	5	6	22
14	25	18	155	29	25	69	7	8	21

Table A.9: Data with $F = 18500$

Product	D_1	D_2	P	C_1	C_2	A	h_1	h_2	f
1	22	18	155	20	25	55	4	2	21
2	25	22	195	24	31	64	5	7	33
3	19	24	184	21	15	52	6	6	26
4	24	30	135	25	22	66	7	8	32
5	18	23	211	18	19	75	3	8	24
6	27	27	169	22	25	63	2	5	31
7	20	18	133	15	15	54	9	3	22
8	21	25	170	26	30	56	6	9	25
9	22	20	158	23	23	77	8	2	28
10	25	22	172	19	26	57	8	4	29
11	23	26	116	32	20	66	5	9	21
12	30	24	145	26	24	63	4	6	36
13	24	22	137	31	29	52	5	6	27
14	17	30	124	25	32	50	9	7	20
15	22	18	151	20	18	67	7	5	37

Table A.10: Data with $F = 22000$

Bibliography

[1] D.D. Achabal, S.H. McIntyre, S.A. Smith, and K. Kalyanam. A decision support system for vendor managed inventory. *Journal of Retailing*, 76(4):430–454, 2000.

[2] H. Andersson, A. Hoff, M. Christiansen, G. Hasle, and A. Løkketangen. Industrial aspects and literature survey: Combined inventory management and routing. *Computers & Operations Research*, 37(9):1515–1536, 2010.

[3] C. Archetti, L. Bertazzi, G. Laporte, and M. G. Speranza. A branch-and-cut algorithm for a vendor-managed inventory-routing problem. *Transportation Science*, 41(3):382–391, 2007.

[4] M.B. Aryanezhad, S. Deilami Moezi, and H. Saiedy. A genetic algorithm to optimize a multi-product EOQ model with limited warehouse-space and capital limitation under VMI. *International Journal of Management Perspective*, 1(1):31–35, 2012.

[5] S. Axster. *Inventory Control*. Springer, New York, NY, USA, 2010.

[6] K.R. Baker and G.D. Scudder. Sequencing with earliness and tardiness penalties: A review. *Operations Research*, 38(1):22–36, 1990.

[7] W.J. Bell, L.M. Dalberto, M.L. Fisher, A.J. Greenfield, R. Jaikumar, P. Kedia, R.G. Mack, and P.J. Prutzman. Improving the distribution of industrial gases with an on-line computerized routing and scheduling optimizer. *Interfaces*, 13(6):4–23, 1983.

[8] L. Bertazzi and M.G. Speranza. Rounding procedures for the discrete version of the capacitated economic order quantity problem. *Annals of Operations Research*, 107(1):33–49, 2001.

[9] L. Bertazzi and M.G. Speranza. Continuous and discrete shipping strategies for the single link problem. *Transportation Science*, 36(3):314–325, 2002.

[10] L. Bertazzi and M.G. Speranza. Worst-case analysis of the full load policy in the single link problem. *International Journal of Production Economics*, 93:217–224, 2005.

[11] L. Bertazzi and M.G. Speranza. Inventory routing problems: an introduction. *EURO Journal on Transportation and Logistics*, 1(4):307–326, 2012.

[12] L. Bertazzi and M.G. Speranza. Inventory routing problems with multiple customers. *EURO Journal on Transportation and Logistics*, 2(3):255–275, 2013.

[13] L. Bertazzi, M.G. Speranza, and W. Ukovich. Minimization of logistic costs with given frequencies. *Transportation Research Part B: Methodological*, 31(4):327–340, 1997.

[14] L. Bertazzi, M.G. Speranza, and W. Ukovich. Exact and heuristic solutions for a shipment problem with given frequencies. *Management Science*, 46(7):973–988, 2000.

[15] L. Bertazzi, G. Paletta, and M.G. Speranza. Deterministic order-up-to level policies in an inventory routing problem. *Transportation Science*, 36(1):119–132, 2002.

[16] L. Bertazzi, G. Paletta, and M.G. Speranza. Minimizing the total cost in an integrated vendormanaged inventory system. *Journal of heuristics*, 11(5):393–419, 2005.

[17] L. Bertazzi, L.M.A. Chan, and M.G. Speranza. Analysis of practical policies for a single link distribution system. *Naval Research Logistics (NRL)*, 54(5):497–509, 2007.

[18] L. Bertazzi, M. Savelsbergh, and M.G. Speranza. Inventory routing. In *The vehicle routing problem: latest advances and new challenges*, volume 43, pages 49–72. Springer, 2008.

[19] D.E. Blumenfeld, L.D. Burns, J.D. Diltz, and C.F. Daganzo. Analyzing trade-offs between transportation, inventory and production costs on freight networks. *Transportation Research Part B: Methodological*, 19(5):361–380, 1985.

[20] A.B. Borade and S.V. Bansod. Study of vendor-managed inventory practices in indian industries. *Journal of Manufacturing Technology Management*, 21(8):1013–1038, 2010.

[21] L.D. Burns, R.W. Hall, D.E. Blumenfeld, and C.F. Daganzo. Distribution strategies that minimize transportation and inventory costs. *Operations Research*, 33(3):469–490, 1985.

[22] A. Campbell, L. Clarke, A. Kleywegt, and M. Savelsbergh. The inventory routing problem. In *Fleet management and logistics*, pages 95–113. Springer US, 1998.

[23] X. Chen, G. Hao, X. Li, and K.F.C. Yiu. The impact of demand variability and transshipment on vendor's distribution policies under vendor managed inventory strategy. *International Journal of Production Economics*, 139(1):42–48, 2012.

[24] S. Chopra and P. Meindl. Supply chain management. strategy, planning & operation. In *Das summa summarum des management*, pages 265–275. Gabler, 2007.

[25] G. Clarke and J. Wright. Scheduling of vehicles from a central depot to a number of delivery points. *Operations Research*, 12(4):568–581, 1964.

[26] L.C. Coelho, J.F. Cordeau, and G. Laporte. The inventory-routing problem with transshipment. *Computers & Operations Research*, 39(11):2537 – 2548, 2012.

[27] L.C. Coelho, J.F. Cordeau, and G. Laporte. Thirty years of inventory routing. *Transportation Science*, 48(1):1–19, 2013.

[28] J.F. Cordeau, G. Laporte, M.W.P. Savelsbergh, and D. Vigo. Vehicle routing. *Transportation*, 14:367–428, 2007.

[29] T.G. Crainic. Long-haul freight transportation. In *Handbook of transportation science*, pages 451–516. Springer US, 2003.

[30] T.G. Crainic and G. Laporte. Planning models for freight transportation. *European journal of operational research*, 97(3):409–438, 1997.

[31] G.B. Dantzig and J.H. Ramser. The truck dispatching problem. *Management Science*, 6(1): 80–91, 1959.

[32] G.B. Dantzig, D.R. Fulkerson, and S.M. Johnson. Solution of a large scale traveling salesman problem. Technical Report P-510, RAND Corporation, Santa Monica, California, USA, 1954.

[33] M.A. Darwish and O.M. Odah. Vendor managed inventory model for single-vendor multi-retailer supply chains. *European Journal of Operational Research*, 204(3):473–484, 2010.

[34] P. De, J.B. Ghosh, and C.E. Wells. Scheduling to minimize weighted earliness and tardiness about a common due-date. *Computers & Operations Research*, 18(5):465–475, 1991.

[35] S.M. Disney and D.R. Towill. A procedure for the optimization of the dynamic response of a vendor managed inventory system. *Computers & Industrial Engineering*, 43(1):27–58, 2002.

[36] Y. Dong and K. Xu. A supply chain model of vendor managed inventory. *Transportation Research Part E: Logistics and Transportation Review*, 38(2):75–95, 2002.

[37] J. Du and J.Y. Leung. Minimizing total tardiness on one machine is np-hard. *Mathematics and Operations Research*, 3(15):483–495, 1990.

[38] Y. Duan, J. Luo, and J. Huo. Buyervendor inventory coordination with quantity discount incentive for fixed lifetime product. *International Journal of Production Economics*, 128(1): 351–357, 2010.

[39] R.C. Eberhart and J. Kennedy. A new optimizer using particle swarm theory. In *Proc. Int. Conf. on micro machine and human science, Nagoya, Japan*, pages 39–43, 1995.

[40] R.C. Eberhart and Y. Shi. Particle swarm optimization: developments, applications and resources. In *Proc. Int. Conf. on Evolutionary Computation, Seoul, Korea*, pages 81–86, 2001.

[41] M.S. Elvander, S. Sarpola, and S.A. Mattsson. Framework for characterizing the design of VMI systems. *International Journal of Physical Distribution & Logistics Management*, 37 (10):782–798, 2007.

[42] A. Federgruen and P. Zipkin. A combined vehicle routing and inventory allocation problem. *Operations Research*, 32(5):1019–1037, 1984.

[43] S. French. *Sequencing and Scheduling: An Introduction to the Mathematics of Job Shop*. John Wiley and Sons, 1981.

[44] M.R. Garey, R.E. Tarjan, and G.T. Wilfong. One-processor scheduling with symmetric earliness and tardiness penalties. *Mathematics of Operations Research*, 13(2):330–348, 1988.

[45] M. Gen and R. Cheng. *Genetic Algorithm and Engineering Design*. John Wiley and Sons, New York, NY, USA., 1997.

[46] F. Glover. Tabu searchpart I. *ORSA Journal on Computing*, 1(3):190–206, 1989.

[47] F. Glover. Tabu searchpart II. *ORSA Journal on Computing*, 2(1):4–32, 1990.

[48] D. Goldberg. *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley, Reading, MA, USA., 1989.

[49] T.L. Golden, B.L. Magnanti and H.Q. Nguyen. Implementing vehicle routing algorithms. *Networks*, 7(2):113–148, 1977.

[50] R. Hallah. Minimizing total earliness and tardiness on a single machine using a hybrid heuristic. *Computers & Operations Research*, 34(10):3126–3142, 2007.

[51] F.W. Harris. How many parts to make at once. *Operations Research*, 38(6):947–950, 1991.

[52] R.L. Haupt and S.E. Haupt. *Practical Genetic Algorithms*. John Wiley & Sons, 2004.

[53] R.M. Hill. The single-vendor single-buyer integrated production-inventory model with a generalised policy. *European Journal of Operational Research*, 97(3):493–499, 1997.

[54] J.H. Holland. *Adoption in neural and artificial systems*. Ann Arbor, Michigan (USA): The University of Michigan Press., 1975.

[55] J. Holmström. Business process innovation in the supply chain–a case study of implementing vendor managed inventory. *European Journal of Purchasing & Supply Management*, 4(2-3): 127–131, 1998.

[56] R.J.W. James and J.T. Buchanan. Performance enhancements to tabu search for the early/tardy scheduling problem. *European Journal of Operational Research*, 106(2):254–265, 1998.

[57] M. Jasemi. A vendor-managed inventory. In *Master of Science Thesis, Department of Industrial Engineering, Sharif University of Technology, Tehran, Iran (in Persion)*. 2006.

[58] J. Kennedy and R.C. Eberhart. Particle swarm optimization. In *Proc. Int. Conf. on Neural Networks, Perth, Australia*, pages 1942–1948, 1995.

[59] J. Kennedy and R.C. Eberhart. *Swarm Intelligence*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2001. ISBN 1-55860-595-9.

[60] P. Kumar and M. Kumar. Vendor managed inventory in retail industry. In *Tata Consultancy Services*, pages 2–9. 2003.

[61] S. Lakshminarayanan, R. Lakshmanan, R.L. Papineau, and R. Rochette. Optimal single machine scheduling with earlinesstardiness penalties. *Operations Research*, 26(4):1079–1082, 1978.

[62] J.Y.T. Leung. *Handbook of scheduling: algorithms, models, and performance analysis*. CRC Press, 2004.

[63] L. Lu. A one-vendor multi-buyer integrated inventory model. *European Journal of Operational Research*, 81(2):312–323, 1995.

[64] K.F. Man, K.S. Tang, and S. Kwong. *Genetic algorithms for control and signal processing*. Springer Science and Business Media, 2012.

[65] R. Mazzini and V.A. Armentano. A heuristic for single machine scheduling with early and tardy costs. *European Journal of Operational Research*, 128(1):129–146, 2001.

[66] M.B. Moghadam, S. Deilami Moezi, H. Saiedy, S. Ghasemi, and V. Maleki. Optimization of VMI with multi-product and multi-constraint by using GA under EPQ. *Asian Journal of Research in Marketing*, 3(3):37–58, 2014.

[67] S.S. Panwalkar, M.L. Smith, and A. Seidmann. Common due date assignment to minimize total penalty for the one machine scheduling problem. *Operations Research*, 30(2):391–399, 1982.

[68] S.H.R. Pasandideh, S.T.A. Niaki, and A. Roozbeh Nia. A genetic algorithm for vendor managed inventory control system of multi-product multi-constraint economic order quantity model. *Expert Systems with Applications*, 38(3):2708–2716, 2011.

[69] S.H..Reza. Pasandideh, S.T.A. Niaki, and M. Hemmati Far. Optimization of vendor managed inventory of multiproduct EPQ model with multiple constraints using genetic algorithm. *The International Journal of Advanced Manufacturing Technology*, 71(1):365–376, 2014.

[70] M. Pinedo. *Scheduling: Theory, Algorithms and Systems*. Prentice Hall, Englewood Cliffs N.J, 1995.

[71] A. Raie and V. Rashtchi. Using a genetic algorithm for detection and magnitude determination of turn faults in an induction motor. *Electrical Engineering*, 84(5):275–279, 2002.

[72] J. Sadeghi, A. Sadeghi, and M. Saidi-Mehrabad. A parameter-tuned genetic algorithm for vendor managed inventory model for a case single-vendor single-retailer with multi-product and multi-constraint. *Journal of Optimization in Industrial Engineering*, 4(9):57–67, 2011.

[73] J. Sadeghi, S.M. Mousavi, S.T.A. Niaki, and S. Sadeghi. Optimizing a multi-vendor multi-retailer vendor managed inventory problem: Two tuned meta-heuristic algorithms. *Knowledge-Based Systems*, 50:159–170, 2013.

[74] K. Sari. Exploring the benefits of vendor managed inventory. *International Journal of Physical Distribution & Logistics Management*, 37(7):529–545, 2007.

[75] S.Y. Shen and M. Honda. Incorporating lateral transfers of vehicles and inventory into an integrated replenishment and routing plan for a three-echelon supply chain. *Computers & Industrial Engineering*, 56(2):754–775, 2009.

[76] J.B. Sidney. Optimal single machine scheduling with earliness and tardiness penalties. *Operations Research*, 25(1):62–69, 1977.

[77] M.G. Speranza and W. Ukovich. Minimizing transportation and inventory costs for several products on a single link. *Operations Research*, 42(5):879–894, 1994.

[78] M.G. Speranza and W. Ukovich. An algorithm for optimal shipments with given frequencies. *Naval Research Logistics (NRL)*, 43(5):655–671, 1996.

[79] W. Szwarc. Adjacent ordering in single machine scheduling with earliness and tardiness penalties. *Naval Research Logistics*, 40(2):229–243, 1993.

[80] R.J. Tersine. *Principles of inventory and materials management*. Englewood Cliffs, NJ, USA: Prentice Hall PTR., 1994.

[81] P. Toth and D. Vigo. Exact solution of the vehicle routing problem. In *Fleet management and logistics*, pages 1–31. 1998.

[82] P. Toth and D. Vigo. *Vehicle routing: problems, methods, and applications*. SIAM, 2014.

[83] J. Tyan and H.M. Wee. Vendor managed inventory: a survey of the taiwanese grocery industry. *Journal of Purchasing and Supply Management*, 9(1):11–18, 2003.

[84] A. Vigtil. Information exchange in vendor managed inventory. *International Journal of Physical Distribution & Logistics Management*, 37(2):131–147, 2007.

[85] H.M. Wee, M.C. Lee, J.C.P. Yu, and C.E. Wang. Optimal replenishment policy for a deteriorating green product: Life cycle costing analysis. *International Journal of Production Economics*, 133(2):603–611, 2011.

[86] C.A. Yano and Y.D. Kim. Algorithms for a class of single-machine weighted tardiness and earliness problems. *European Journal of Operational Research*, 52(2):167–178, 1991.

[87] Y. Yao, P.T. Evers, and M.E. Dresner. Supply chain integration in vendor-managed inventory. *Decision Support Systems*, 43(2):663–674, 2007.

[88] Y. Yu, H. Chen, and F. Chu. A new model and hybrid approach for large scale inventory routing problems. *European Journal of Operational Research*, 189(3):1022–1040, 2008.

[89] F. Zammori, M. Braglia, and M. Frosolini. A standard agreement for vendor managed inventory. . *Strategic Outsourcing: An International Journal*, 2(2):165–186, 2009.

[90] L. Zavanella and S. Zanoni. A one-vendor multi-buyer integrated production-inventory model: The consignment stock case. *International Journal of Production Economics*, 118(1):225–232, 2009.