



**HAL**  
open science

# Production and delivery integrated scheduling problems in multi-transporter multi-customer supply chain with costs considerations

Zakaria Hammoudan

► **To cite this version:**

Zakaria Hammoudan. Production and delivery integrated scheduling problems in multi-transporter multi-customer supply chain with costs considerations. Automatic. Université de Technologie de Belfort-Montbéliard, 2015. English. NNT : 2015BELF0267 . tel-01864063

**HAL Id: tel-01864063**

**<https://theses.hal.science/tel-01864063>**

Submitted on 29 Aug 2018

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# SPIM

## Thèse de Doctorat



école doctorale sciences pour l'ingénieur et microtechniques  
UNIVERSITÉ DE TECHNOLOGIE BELFORT-MONTBÉLIARD

Thèse Présentée en vue de l'obtention du titre de  
**DOCTEUR**

### **Production and Delivery Integrated Scheduling Problems in Multi-Transporter Multi-Customer Supply Chains with costs considerations**

Zakaria HAMMOUDAN

De

L'Université de Technologie de Belfort-Montbéliard

Laboratoire IRTES-SET

Spécialité: Automatique

ABDELLAH EL-MOUDNI    Directeur de thèse    Professeur à l'UTBM

Olivier GRUNDER    Co-Directeur de thèse    Maître de Conférence HDR à l'UTBM

# Contents

<b>List Of Publications</b>	<b>5</b>
<b>Acknowledgments</b>	<b>7</b>
<b>Abstract</b>	<b>9</b>
<b>Résumé</b>	<b>11</b>
<b>1 Introduction</b>	<b>13</b>
1.1 Context . . . . .	13
1.2 Methodology . . . . .	14
1.3 Outline of Dissertation . . . . .	15
<b>2 State-of-the-Art</b>	<b>19</b>
2.1 Introduction . . . . .	19
2.2 Description of a supply chain . . . . .	20
2.3 Lot-Sizing and Batch-Scheduling problems . . . . .	21
2.3.1 Single-level lot-sizing and batch-scheduling problems . . . . .	22
2.3.2 Multi-Level lot-sizing and batch-scheduling problems . . . . .	23
2.4 The integrated Production-Delivery-Inventory problems . . . . .	27
2.4.1 Basic studied models . . . . .	27
2.4.2 Extension of the basic models . . . . .	29
2.5 The integrated Delivery-Inventory problems . . . . .	31
2.5.1 Basic studied models . . . . .	33
2.5.2 Extended studied models . . . . .	38
2.6 Conclusion . . . . .	43
<b>3 Multi-Customer with Single-Transporter Integrated Lot Sizing and Delivery Scheduling Problem</b>	<b>45</b>
3.1 Introduction . . . . .	45
3.2 Problem Definition and Mathematical Formulation . . . . .	47
3.2.1 Notations . . . . .	47
3.2.2 Mixed Integer Programming Model . . . . .	48
3.3 Branch and Bound Algorithm . . . . .	50
3.4 Heuristic Algorithm . . . . .	52
3.5 Genetic Algorithm . . . . .	55
3.5.1 Gene Representation . . . . .	56

3.5.2	Detailed algorithm . . . . .	56
3.6	Experimental results . . . . .	59
3.6.1	Numerical example . . . . .	60
3.6.2	Test cases . . . . .	61
3.6.3	Comparison of the computational time of solutions . . . . .	61
3.6.4	Comparison of the quality of solutions . . . . .	64
3.7	Conclusion . . . . .	67
<b>4</b>	<b>Single-Customer with Multi-Transporter Integrated Lot Sizing and De-</b>	
	<b>livery Scheduling Problem</b>	<b>69</b>
4.1	Introduction . . . . .	69
4.2	Problem description . . . . .	71
4.2.1	Notations . . . . .	71
4.2.2	Problem definition . . . . .	72
4.2.3	Illustrative Example . . . . .	72
4.3	Mathematical Formulation and Dominance property . . . . .	74
4.3.1	General case . . . . .	75
4.3.2	Identical transporters case . . . . .	78
4.4	Solving methods . . . . .	80
4.4.1	Dynamic programming approach . . . . .	80
4.4.2	The Mixed Integer Programming Model . . . . .	82
4.4.3	Genetic Algorithm . . . . .	84
4.5	Experimental and Computational results . . . . .	87
4.5.1	Instances with Heterogeneous Transporters . . . . .	88
4.5.2	Instances with Homogeneous Transporters . . . . .	91
4.5.3	Large instances with Homogeneous Transporters . . . . .	93
4.6	Conclusion . . . . .	95
<b>5</b>	<b>Multi-Customers/Transporters Integrated Lot Sizing and Delivery Schedul-</b>	
	<b>ing Problem</b>	<b>97</b>
5.1	Introduction . . . . .	97
5.2	Problem definition and formulation . . . . .	98
5.2.1	General notations . . . . .	100
5.2.2	Assumptions . . . . .	101
5.3	The General studied Model . . . . .	101
5.3.1	The general model with batch size dependent time and cost . . . . .	102
5.3.2	Branch and Bound Algorithm . . . . .	102
5.4	Experimental Results . . . . .	111
5.4.1	Evaluation of the quality of different models . . . . .	111
5.4.2	Evaluation of the Computational Results . . . . .	115
5.5	Conclusion . . . . .	118
<b>6</b>	<b>Conclusions and Future Extensions</b>	<b>121</b>
6.1	Conclusions . . . . .	121

6.2	Summary of Contributions . . . . .	121
6.3	Possible Future Extensions . . . . .	123
	<b>Bibliography</b>	<b>125</b>



# List of Figures

- 2.1 Flow diagram of single supplier single customer with multiple transporters integrated supply chain . . . . . 20
- 2.2 Flow diagram of single supplier multiple customers with one transporters integrated supply chain . . . . . 21
- 2.3 Serial product structure . . . . . 24
- 2.4 Assembly product convergent structure . . . . . 24
- 2.5 Arborescent product divergent structure . . . . . 24
- 2.6 General product network structure . . . . . 24
- 2.7 Different product structures . . . . . 24
  
- 3.1 Single-supplier and multi-customer model . . . . . 47
- 3.2 Genetic algorithm . . . . . 57
  
- 4.1 The case of optimal solution (optimal cost = 1750.24) . . . . . 73
- 4.2 The case of fully loaded solution ( Cost = 2158) . . . . . 73
- 4.3 An example of the dominance relation . . . . . 77
- 4.4 An example of the dominance relation in the identical transporters case 78
- 4.5 A simple GA flowchart . . . . . 85
  
- 5.1 Multi-customer Multi-transporter general model . . . . . 99
- 5.2 Illustration of models for different delivery structures . . . . . 112
- 5.3 Illustration of models for different delivery structures . . . . . 113
- 5.4 Illustration of models for different delivery structures . . . . . 114
- 5.5 Illustration of models for different delivery structures . . . . . 115





# List of Tables

3.1	A chromosome representing a solution . . . . .	56
3.2	The two parents crossover operator . . . . .	58
3.3	A numerical example . . . . .	60
3.4	Main characteristics of the test cases . . . . .	61
3.5	Computational results for $\beta_h \in [1, 5]$ <i>Euro</i> . . . . .	62
3.6	Computational results for $\beta_h \in [10, 50]$ <i>Euro</i> . . . . .	63
3.7	Computational results for $\beta_h \in [100, 500]$ <i>Euro</i> . . . . .	64
3.8	The performance results for $\beta_h \in [1, 5], [10, 50], [100, 500]$ <i>Euro</i> for three classes respectively . . . . .	65
3.9	The performance results with $\beta_h \in [1, 5], [10, 50], [100, 500]$ <i>Euro</i> for three classes respectively . . . . .	66
4.1	A chromosome representing a solution . . . . .	85
4.2	The crossover operator . . . . .	87
4.3	Results of instances with heterogeneous transporters with $\beta = [0.01, 0.05]$ . . . . .	89
4.4	Results of instances with heterogeneous transporters with $\beta = [1, 5]$ .	89
4.5	Results of instances with heterogeneous transporters with $\beta = [10, 50]$ . . . . .	90
4.6	Comparison of the quality of the solutions found by the GA and the <i>GDP</i> . . . . .	91
4.7	Results of small-medium instances with identical transporters . . . . .	92
4.8	Comparison of the quality of the solutions found by the GA and the <i>GDPid</i> . . . . .	94
4.9	Results of large instances with identical transporters . . . . .	94
5.1	Data for a single-supplier and 2-customers problem . . . . .	112
5.2	Main characteristics of the test cases . . . . .	116
5.3	Total computational time for the three classes . . . . .	117



# LIST OF PUBLICATIONS

This thesis arising 8 international publications and 2 national publications and it contains some unpublished material.

## International journals [3]

- [1] GRUNDER, O., HAMMOUDAN, Z., & EL MOUDNI, A. (2015). Production and delivery scheduling problem with delivery time and cost dependent on batch-size. *International Journal of Logistics Systems and Management.*(**Accepted in press**)
- [2] HAMMOUDAN, Z., GRUNDER, O., BOUDOUH, T., & EL MOUDNI, A. (2015). A coordinated scheduling of delivery and inventory in a central pharmacy of multi-hospitals. *European Journal of Industrial Engineering.* (**Pre-accepted**)
- [3] HAMMOUDAN, Z., GRUNDER, O., & EL MOUDNI, A. (2015). A Branch and Bound Algorithm for One Supplier and Multiple Heterogeneous Customers to solve a Coordinated Scheduling Problem. *RAIRO Operations Research Journal.* (**Under review**)

## International conferences [5]

- [4] GRUNDER, O., HAMMOUDAN, Z., & EL MOUDNI, A. (2013) Production and Delivery Integrated Scheduling to Minimize the Costs with Heterogeneous Transporters. In *Manufacturing Modelling, Management, and Control (MIM 2013)*, 2013 International Conference. IFAC.
- [5] HAMMOUDAN, Z., GRUNDER, O., BOUDOUH, T., & El Moudni, A. (2014, November). A branch and bound algorithm for one supplier and multiple heterogeneous customers to solve a coordinated scheduling problem. In *Control, Decision and Information Technologies (CoDIT)*, 2014 International Conference on (pp. 213-218). IEEE.
- [6] HAMMOUDAN, Z., GRUNDER, O., & EL MOUDNI, A. (2014) Coordinated scheduling of delivery and inventory with one supplier and multiple heterogeneous customers. In *Modelling, Optimization and Simulation of systems (MOSim)*, 2014 International Conference.
- [7] HAMMOUDAN, Z., GRUNDER, O., & EL MOUDNI, A. (2015) A robust heuristic algorithm for multiple-customers coordinated scheduling of delivery and inventory problem. In *Symposium on Information Control in Manufacturing (INCOM 2015)*, International Conference. IFAC.

[8] HAMMOUDAN, Z., GRUNDER, O., & EL MOUDNI, A. (2015) A genetic algorithm approach to the single-vendor multi-customer integrated delivery-inventory problem. International Conference on Computers & Industrial Engineering (CIE45), International Conference. IFAC.

## **National conferences [2]**

[9] HAMMOUDAN, Z., GRUNDER, O., & EL MOUDNI, A. (2013) Minimisation des coûts dans une chaîne logistique simple, dans le cas multi-transporteur. 14<sup>ème</sup> conférence de la société Française de Recherche Opérationnelle et Aide à la Décision (ROADEF)

[10] HAMMOUDAN, Z., GRUNDER, O., & EL MOUDNI, A. (2014) Optimisation combinée des couts de transport et de stockage dans un réseau logistique, multi-clients. 15<sup>ème</sup> conférence de la société Française de Recherche Opérationnelle et Aide à la Décision (ROADEF)

# Acknowledgments

First and foremost, i would like to to express my deepest sense of gratitude to my chief supervisor Professor Abdellah EL-MOUDNI and my co-supervisor Dr. Olivier GRUNDER. Thanks for their continuous academically and personally support and mentorship in these three years of study. Thanks for their guidance, suggestions and dedication throughout this thesis. They are one of the few people I know that have not lost a bit of their glittering enthusiasm about their job. Working with them was an interesting challenge, made up of hard work and fun at the same time. I would like to thank specially Dr. Olivier GRUNDER for working closely with me. His unconditional support in solving many details surrounding this dissertation and his valuable feedback's are deeply appreciated. A special thank to Dr. Raed KOUTA for all the good advice. Many thanks to the Board of Examiners, Dr. Lionel AMODEO, Dr. Slim HAMMADI, Dr. Abdelhakim ARTIBA and Dr. Philippe LUTZ for their constructive comments and suggestions which lead to a significant improvement in the thesis.

I strongly felt the ceaseless faith and support of my family along the way, I deeply thank my parents, Ahmad HAMMOUDAN and Fadia HAMMOUDAN, my brothers (Youssef, Yasser, Yahya and Abdel-razzak), my sisters (Fatima, Najwa, Nisrine, Hanadi, Nour and Zaynab), and my stepbrother (Khaled HASHEM, Nassim EL-GHARIB, Abdallah BAKER, Raed AL-YOUSSEF and Mouhamad HAMMOUDAN). Their love and support are a constant inspiration to me in striving to improve myself. This dissertation is dedicated to you, specialy to my Mom and Dad. This dissertation would have not been possible without your supports. Further, I would like to express sincere thanks to my beloved one, Sarra EL-MAAZI, for her love, support and patience, allowing me to accomplish my goal and sharing with me all the happy and sad moments.

I also would like to express my appreciation to all my friends and colleagues at the IRTES-SET laboratory and at the UTBM university for lightening up my life every day and making graduate school more fun than I am sure it is supposed to be. In particular I would like to thank Dr. Imad MATRAJI, Dr. Mouhamad HARMOUSH, Dr. Fayez AHMED, Rabih ZAKARIA, Joseph EL-ASMAR, George GEORGES, Florant PERONET, Abdelrahim CHARJET, XUGUANG HAO, Dr. Jalil AB-BASS TURKI, Dr. Rachid BOUYEKHF, Dr. Mahjoub DIRIDI, Mouaz SAYED-AHMAD, Dr. Inaya LAHOUD, Dr. Mouhamad KABALO, Samer CHEHAIDI, Ghassan MOURAD, Malek SARMOUT, Zaher EL-CHAMI, Mouhamad KHALIL, Mouhamad SLEIMAN. Thanks also go out to the secretary of laboratory SET Ms.

Ariane GLATIGNY for her help with administrative matters.

I would like to dedicate also in this dissertation my friends in Lebanon. First, to my deceased friend Jackie SAAD for provided unconditional support and encouragement me when I needed most to be encouraged. Second to my friend Antoine SAAD, who provided encouraged me in moments of doubt. Third to my friend Ghassan EL-RIFI, who also encouraged me in moments of doubt. I extend my thanks to Assem YAMAK, Khaled EL-HALABI and his family, Rabih ANKLIS, Dr. Alfred RIACHI, Dr. Ahmad EL-RAFII, Dr. Mouhamad KHALIL, M. Rami RIFAI, M. Ahmad SAFADI, M. Nasser ADRA, M. Fadi EL-MASRI, for all your encouragement and support.

# Abstract

The coordination of logistics activities in a supply chain has recently received a lot of attention in operations management research. In a typical supply chain, finished products are produced and either shipped to be temporary storage or arrived directly on time to the customers. To achieve optimal operational performance, the coordination and integration of production, delivery, and storage is an important consideration. The recent study considered customer storage cost with fixed transportation cost or fixed batch size. In this thesis, we study the coordination of batching and scheduling activities, which include the customer(s) orders which require the delivery time and cost from the supplier and the storage of products at the customer(s). This study focus on the multi-customer scenario and multi-transporter scenario. For the first scenario with multi-customer, two models illustrate the transferring of batches to the customer. Where in the first model, we considered a multi-customer with one transporter available to serve the customers without the vehicle routing consideration. Then, in the second model, we considered a multi-customer with multi-transport available to serve the customers. Concerning the second scenario, we studied the case with multi-transporter available to serve a single customer. In this scenario, models with homogeneous and heterogeneous vehicles are studied. The costs and the distances depend of the transporters used or the customers according to the proposed scenario. The future extension of this research may involve considering the vehicle routing consideration with inventory in the multi-customer case.

Keyword: Coordinated scheduling; Production and Delivery; Delivery and Inventory; Integrated models; Supply chain management; Heuristic; Batching; Dynamic programming; Branch and Bound; Mixed integer programming





# Résumé

La coordination des activités dans une chaîne logistique a suscité de nombreux travaux de recherche dans le domaine du management et la recherche opérationnelle. Dans une chaîne logistique typique, les produits finis sont transportés pour être stockés temporairement ou arriver directement chez les clients. Pour obtenir la configuration opérationnelle optimale, la coordination et l'intégration de la production, de la livraison, et du stockage deviennent importantes. Des études récentes ont considéré le coût de stockage chez le client en fixant le coût de transport ou la taille des lots, mais dans la pratique cela se révèle plus complexe. Dans cette thèse, nous avons étudié la coordination entre le traitement et l'ordonnancement des lots, ce qui inclut les commandes des clients qui requièrent la livraison du fournisseur ainsi que le stockage des produits chez les clients. Ce travail se focalise sur les cas plusieurs-clients et plusieurs-transporteurs. Pour le premier scénario avec plusieurs-clients, deux modèles illustrent le transfert des lots aux clients. Dans le premier modèle, nous avons considéré plusieurs-clients avec un seul transporteur disponible pour servir les clients sans prendre en considération la tournée de véhicule. Puis, dans le deuxième modèle, nous avons étudié le cas de plusieurs-clients avec plusieurs transporteurs disponibles pour servir les clients. En ce qui concerne le deuxième scénario, nous avons étudié le cas avec plusieurs transporteurs disponibles pour servir un seul client. Dans ce scénario, des modèles avec les véhicules homogènes et hétérogènes sont étudiés. L'ensemble des coûts du système sont calculés en additionnant le coût de la livraison et de stockage pour les différents clients et transporteurs. Les coûts et les distances sont dépendants soient des transporteurs soient des clients suivant le scénario retenu. Dans chaque modèle, nous présentons les procédures de résolution, plusieurs exemples numériques pour soutenir les résultats mathématiques et pour clarifier le problème, ainsi que des comparaisons de performances entre les différents résultats. Les perspectives de recherche se situent actuellement dans la prise en compte de tournées entre les différents clients.

Mots clefs: Modèles intégré; Production et distribution; Distribution et stockage; modèles intégrés; Planification de la chaîne logistique; Meta-Heuristique; Lots; Programmation dynamique; Methodes de coupe (Séparation et évaluation); Programmation en nombres entiers mixte



# 1 Introduction

## 1.1 Context

The competitive environment in the industrial companies to accommodate the maximum number of customers is considered as a key factor in the evolution of the companies, in order to increase their profits. Today, the issues of cost and time dissipation due to delays are becoming more and more important in terms of customer's requirements. As well, the simultaneous optimization of the production, transport and storage activities becomes a key factor in the success of a company specifically, and the whole supply chain generally. The objective of each supplier is to satisfy its customer demands with high service level requirements to meet the optimal total cost, which will ensure that the products are delivered in time, at the right location, in the right condition, in the right quantity. This relation of coordination between the supplier and its customers assimilate the situation as an integrated problem with taking into account all the related indicators that may affect on the final cost, as the setup cost, the production cost, the storage cost at the supplier, the delivery cost and the storage cost at the customers.

The increased efficiency of foreign manufacturers in the late 1970s and early 1980s changed the thinking at the managers on the supply chain. The integrating decisions between the different members in the supply chain attracted the attention of companies, where they started to realize the potential cost benefits. At the beginning, the researchers failed to take an integrated view of the integral supply chain. They considered different sub-problems as production or distribution or distribution with inventory at the customer. These sub-problems were solved separately and the optimal solution were then joined together to establish an operating policies. Recently, the coordinated of logistics activities in a supply chain has recently received a lot of attention in operations management research. To achieve optimal operational performance, the coordination of production, delivery and storage is an important consideration.

This thesis studied different scheduling planning problems in various supply chain structures. We mainly consider an integrated Lot-Sizing Delivery-scheduling problems in two different axes: (1) Multiple Customers Lot Sizing Delivery Scheduling Problem (MCLSDSP) (real-world case from the pharmaceutical environment); (2) Multi-Transporter Lot Sizing and Delivery Scheduling Problem (MT-LSDSP) with batch-size-dependent delivery time and cost.

In the first axe, we investigated a delivery-inventory supply-chain problem with multiple products in a multiple customers and taken into consideration a different parameters related to the health care due to the our real-world studied case from the pharmaceutical environment. The studied systems is composed of a central medical stores which has to deserve pharmaceutical supplies to different hospital sites at given due dates. The objective is to reduce the overall cost which includes the delivery costs and an earliness penalty that is incurred for medical supplies which are delivered before their due dates. For this case, different scenario of the delivery stage are proposed. In the second axe, we studied an integrated scheduling problem for a single item, in a supply chain environment involving a fleet of heterogeneous or homogeneous transporters with batch-size-dependent delivery time and cost. We assumed that a job that arrives at the customer before its due date will incur an earliness penalty depending on the considered job, with the objective is to find a coordinated lot sizing and scheduling scheme such that the total cost is minimized while guaranteeing a certain service level.

## 1.2 Methodology

The methodology, aims to demonstrate that much can be gained in a systematic intervention. This is a structured set of activities to assist people in undertaking research. Generally, a methodology will develop, either implicitly or explicitly, within a particular paradigm and will embody the philosophical assumptions [Mingers and Brocklesby, 1997]. Methodologies may emerge as broad prescriptions for good practice in using particular techniques. A high relation is between methodology and technique, where the methodology specifies what type of activities should be undertaken, and the techniques are particular ways of performing these activities. Moreover, each methodology has a number of possible techniques. The optimization of a problem described in a specific variables, where the complexity of the problem is defined as its number of variables and the complexity of the best possible algorithm can solve it. The integrated problems are classified into polynomially solvable and Nondeterministic Polynomial hard (NP-hard) problems. The polynomially solvable problems is that which can be solved by a polynomial time algorithm. A problem is called (NP-hard) if it can be solved by a polynomial time by a nondeterministic machine. Two important way is used to solve an NP-hard problem, that by encoding and enumerating all its feasible solutions, then select the best solution.

The integrated problems belongs to the combinatorial optimization problems. It is a critical hard and it needs an efficient tools in order to formulate and solve problems to make a final decision. Some frequently techniques are used to study the integrated problems comprise dynamic programming algorithm, branch and bound algorithm as an exact methods and some approximation algorithms as an approach methods. For these two exact methods, they help to simplify the search process for an optimal solution it needs to exploring the dominant properties. In this thesis

we emphasize the modeling and algorithmic development of optimization models for different coordinated scheduling problems. Since the studied problems that we are dealing have high complexity, we focus on the development of efficient and effective heuristic algorithms for those problems. In chapter 3, we proposed two exact methods to evaluate the coordinated scheduling of delivery-inventory problems with a single supplier and multiple customers. The first method is a mixed integer programming model, and the second one is a branch and bound algorithm supported by a lower bound. Then, we developed an heuristic approach and a genetic algorithm to solve these problems. In chapter 4 we developed a dynamic programming algorithm supported by a dominance and a mixed integer programming approach as an exact methods to solve the integrated scheduling problem with multiple transporters, then we developed a genetic algorithm as an approach method, which could solve the large instances. However, in chapter 5 we studied the case of single-supplier/multi-customer with multiple transporters assumption. For this problem, we proposed a general model with multiple heterogeneous transporters available to serve the customer without any allotment, then we have proposed different models with different delivery structures. For this work, we have developed branch and bound algorithm in the general case, and MIP model for each case solved by CPLEX.

### 1.3 Outline of Dissertation

In this dissertation, we study the coordination of scheduling activities by taken into consideration the integrated production and delivery scheduling with batch-size-dependent on the delivery times and costs in the case of multiple transporters, the coordination of delivery and inventory lot-sizing and batch-scheduling problem with multiple customers in the case of single transporter, and the coordination of delivery-inventory scheduling of multiple customers with multiple transporters. This dissertation is divided into four chapters.

Chapter 2 deals with an extensive literature review on batch scheduling problems and the related cases studied in this dissertation.

In chapter 3, we study the MCLSDSP with a multiple products consideration. We specifically consider a real-world scheduling problem which is motivated by a local pharmaceutical industry. In this problem, there is a central pharmacy which has to deserve pharmaceutical supplies to different hospital sites with a single transporter at given due dates. It is assumed that directing delivery method is used for sending the batches to the hospitals, which means that the jobs are transmitted to each customer separately [Chen, 2010]. The objective is to determine the optimal solutions for inventory lot size, scheduling programs and the number of deliveries to achieve hospitals goal with a minimum total cost for the central pharmacy.

In chapter 4, we discuss the MT-LSDSP with batch-size-dependent delivery time and cost. Few research works have been conducted the multiple transporters characteristics in the context of the integrated production delivery problems, assuming

generally that a transporter is always available to deliver the products. In our work, we develop and analyze a scheduling model that takes into account the storage cost on both sides of the supply chain (supplier and customer) and the delivery time and cost which are dependent on batch-size. The consideration of batch-size-dependent delivery time and cost which take into account the loading and the unloading times, has not yet been studied as far as we know.

In chapter 5, we investigate the multiple transporters MCTLSDSP where different models for the supply of multi-items are proposed. In the first model which is a general case, the manufacturer transfers the jobs to a customer by a finite number of heterogeneous transporters without any allotment, whereas in the other extension models the batches are transferred to the customers by a number of transporters allotted to each customer. Under the first model, the general case of the single-supplier/multi-customer with multiple transporters is studied, whereas for the second model it is a fleet of transporters available to deliver the jobs from the supplier to the customers where each customer is served by his allotted transporter. The intention of this work is to find an optimal solution technique for each of these two models, and through a sensitivity analysis explore for the best outcome. In this work, we mixed the both previous studied cases in chapters 3 and 4 to produce a new view to the global case of the multiple customers with multiple transporters, in the objective is to minimize the delivery and storage costs.

Finally, in chapter 6 we summarize the work and recommend areas that need further study.







# 2 State-of-the-Art

## 2.1 Introduction

The objective of all effort spent on supply chain management is to devise an optimal strategy between the different stages of supplying a product to develop, manage and operate the supply chain in order to satisfy customer's demand by producing the appropriate product at the right time and places with suitable quantity for the least expenses without lowering any product quality. An efficient supply chain requires a smooth logistics flow in all activities from moving the raw materials into finished products and from moving the finished products to the end customers. Coordination between production, transportation and storage activities can be a major in reducing total costs. In this chapter, we present a literature review on the different lot-sizing and scheduling problems. Lot-sizing and scheduling are important parts of the production process system. It is necessary to know how many items and in which order they have to be produced, to minimize the total costs.

The simultaneous optimization of the production, transport and storage activities becomes a key factor in the success of a company in specific, and the whole supply chain in general. The requirements generation is a key distinction, they may be generated directly by customers orders or indirectly by the different mentioned stages (production, transport and storage). The production scheduling is a sequencing problem and a lot-sizing decision associated with the inventory and transporters processes. The scheduling criteria indicates the measures upon which schedules are to be evaluated in term of schedule cost. The cost associated includes the fixed costs associated with production setups, transporter costs, inventory holding costs at the supplier or the customers.

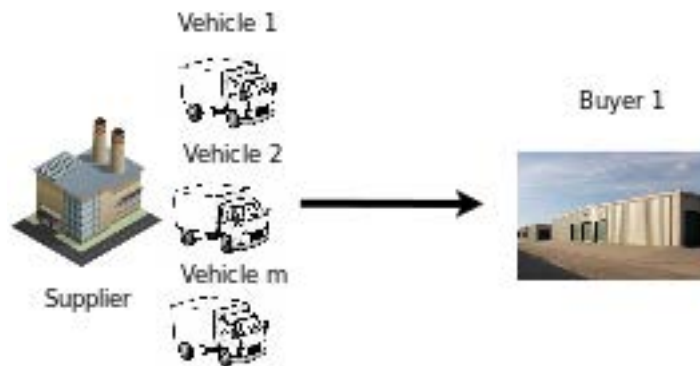
An efficient supply chain requires a smooth logistics flow in all activities from moving the raw materials into finished products and from moving the finished products to the end customers. In this thesis, a single supplier supplying items to one or multiple buyers. The supplier produce the item in batches and at a finite rate. The supplier then sends the finished items to the buyer(s). In this process, the studied system(s) incur(s) batch delivery and storage costs, in different assumptions. Meanwhile, the buyer(s) has his own deterministic demand. In the problem under study, the supplier and the customer have a problem of determining the shipping schedule and the lot-sizing of each batch, which minimizes the operating cost. During the last three decades, researchers have been searching for the solution to take an integrated view of the entire supply chain. They considered only one piece of the

overall problem, such as production or distribution sub-models. These sub-models were optimized separately and the solutions were then joined together to establish operating policies. The supplier and the buyer(s) have known as integrated supply chain system, rather than as separate individuals, total system cost can be reduced significantly. Here, we concentrate on developing optimization tools to enable companies to take advantage of opportunities to improve their supply chain.

This chapter gives the propoerties of the studied models, the definitions of Lot-Sizing and Batch-Scheduling problems, a short overview of each problem is noted and we reviewed the Multi-level Lot-Sizing and Batch-Scheduling problems. The review then shifts to the integrated production-delivery-inventory problems with batch-size-dependent scenario and the delivery-inventory problems with incompatible jobs families and due dates objectives.

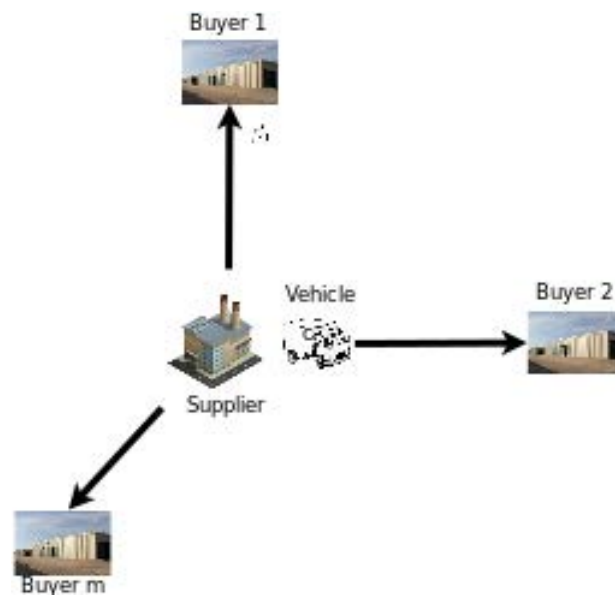
## 2.2 Description of a supply chain

In the integrated supply chain, the supplier needs to have a clear information about their customer(s) demand. Due to the great improvements in the electronic information, the exchange of the information between the supplier and the customer(s) becomes feasible. The diagrams in Figure 2.1 and Figure 2.2 show two integrated supply chain models, where in Figure 2.1 the flow of one/supplier multi/customer with one transporter, and the second diagram in Figure 2.2 shows the flow of sine/supplier single/customer with one transporter.



**Figure 2.1:** Flow diagram of single supplier single customer with multiple transporters integrated supply chain

The production planner is concerned with determining optimal production-inventory and distribution inventory levels for each product in every period so that the total cost of production, delivery and inventory holding is minimized. On the other hand, the distribution planner must determine schedules for distribution of products to customers so that the total transportation cost is minimized. The inventory and distribution planning problem decides on the replenishment policy at the warehouse



**Figure 2.2:** Flow diagram of single supplier multiple customers with one transporters integrated supply chain

and the distribution schedule for the customers, so that the total of inventory and distribution costs are minimized. In supply chain, transportation cost is a major part of operational cost. Transportation time, cost, and capacity constraint play a role in making decisions. In today's world, short life cycle and countless specialties of similar products have made the global market highly competitive. In order to survive market pressure, every company has to be highly competitive in terms of product quality, price and product supply. For example, shipping in smaller quantities and with high frequency reduces the inventory level at the warehouse, but causes higher transportation costs. The delivery and inventory activities can function independently if there is a sufficiently large inventory buffer that completely decouples the two. However, this would lead to increased holding costs and longer lead times, since on one side the distribution planner, in order to minimize transportation costs, would prefer full truck shipments and minimum number of stops; and on the other side the production planner would prefer less number of machine setups. The pressure of reducing inventory and lead times in the supply chain forced companies to explore the issue of closer coordination between production and distribution.

## 2.3 Lot-Sizing and Batch-Scheduling problems

The Lot Sizing problem (LSP) and Batch Scheduling Problem (BSP) are both parts of the most challenging subjects that arises in production planning for the supply chain managers. The LSP is the activity to transform customer demands into lots to minimize the total costs of setup costs, product change costs and inventory costs.

The objective of the LSP is to determine the periods in which production will take place and the quantities that have to be produced per period. The LSP depends on the setup times, which depend on the assignment of the machine and the product sequence which are fixed by the scheduling. The BSP is the activity when and which machine should be produced where there are no shortages in the production plan. Since each production series involves an additional cost of setup, thus the global production quantity has to satisfy the total demand of the customer and has to minimize the global cost of the system. The BSP needs the production volume to decide about the machine assignment and the sequence [Quadt, 2004]. The coordination between lot sizing and scheduling problems have the objective to make an optimal decision about the number of items by batch and also the sequence of the items, to make the production planning feasible.

### 2.3.1 Single-level lot-sizing and batch-scheduling problems

Historically, [Harris, 1913] predates the first work on LSP which is based on the economic order quantity (EOQ). He considered a model that assumes demands occur continuously over time. In the early fifties, the EOQ was followed by the dynamic model developed by [Wagner and Whitin, 1958]. [Wagner and Whitin, 1958] developed an extension of the work of [Harris, 1913], where they studied a model with a dynamic demand in each period and no capacity constraints with a finite time horizon. They find an optimal solution by using a dynamic programming algorithm to solve the problem. The [Wagner and Whitin, 1958] model has received a considerable attention and it was discussed in several hundred papers. [Wagner and Whitin, 1958] model is extended to Capacitated LSP, which is a single-level multi-item capacitated lot-sizing model. [Fleischmann, 1990] studied the CLSP with scheduling, where that is one setup per period for each item which is produced in the period. The setup occurs at the beginning of the period. The integration of the capacity constraint is defined in the Discrete LSP with scheduling is the “all-or-nothing-assumption”, where that is one item produced over a full period or not at all [Salomon, 1991]. In this case, the number of items of the consecutive batches must be the same. If the number of items by batch in two consecutive separated periods is changed, a new setup state must be taken into consideration. which mean that, in the DLSP the continuity of the setup time is not considered. The latter drawback is handled in the Continuous Setup LSP (CSLSP) which allows partial capacity usage. Still, only a single item can be produced per period. [Brahimi et al., 2006] defined the CSLSP as a sequence independent kind of a small bucket problem, what still means that only one item can be produced per period, but the “all-or-nothing” assumption is given up. The problems with multi-item is classified as a coordinated lot-sizing and batch-scheduling problems. The first problem studied in this domain is the Economic Lot-sizing and batch-Scheduling Problem (ELSP) in [Salomon, 1991]. [Salomon, 1991] defined the model as a minimization of sum of setup and inventory holding costs, while determining the production schedule. The ELSP studied the

model of multi-item produced separately. The twins of the CSLP with one item is the Proportional Lot-sizing and batch-Scheduling Problem (PLSP). In the PLSP two different items can be produced per period. We can use the capacity of a full period at most for two different items with only one changeover allowed per period. The development of this model is the General Lot-sizing and batch-Scheduling Problem (GLSP). In the GLSP, the planning horizon is finite, the demand is dynamic and no backlogging is allowed. In this model, the periods are divided into sub-periods, which allowed more than one setup per period with a restricted number of lots per period [Drexl and Kimms, 1997]. The extension of the GLSP is the GLSP with Setup Times (GLSPST). The GLSPST is a single-level multi-item capacitated model with dynamic demand and a finite planning horizon [Tischer, 2007]. The objective of the GLSPST is to minimize the inventory holding costs and the sequence dependent setup costs [Meyr, 2000].

Recent applications of lot sizing and batch scheduling models are described in [Relvas et al., 2013], [Khan et al., 2014] among others. [Relvas et al., 2013] studied the lot sizing and batch scheduling with inventory management of an oil products distribution system, they have proposed two different models to the sizing decision of batch volume. The first one is the fixed batch size model (FBS) and uses a set of prefixed volumes of batches that vary with product, where a decision has to be taken to select which one to pump. On the other hand, the variable batch size model (VBS) provides for each product a valid interval for batch volume, being any intermediate value allowed. [Relvas et al., 2013] proved that the (VBS) model offers a better final values than the (FBS). Next, [Khan et al., 2014] studied an integrated supply chain model with errors in quality inspection and learning in production, concerning the lot sizing problem they supposed that the supplier replenishes the order in a number of equal-sized shipments, there aim is to determine an optimal supplier–customer inventory policy for a single product.

### 2.3.2 Multi-Level lot-sizing and batch-scheduling problems

The DLSP, ELSP, PLSP, GLSP and GLSPST are single level models for simultaneous Lot-sizing and Batch-scheduling problems. The extension of the single level models mentioned above leads to the multi-level LSP (MLLSP) model which is studied in different complex real world industrial cases. The feature of the MLLSP is that the production of each item generates dependent demand for its constituent components, if any [Brahimi et al., 2006]. [Tischer, 2007] classified the MLLSP as NP-hard problem and [Salomon, 1991] sorted it in four structures (Serial, Assembly convergent, Arborescent divergent and General network) as follows:

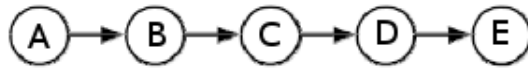
The serial product structure

The assembly product convergent structure

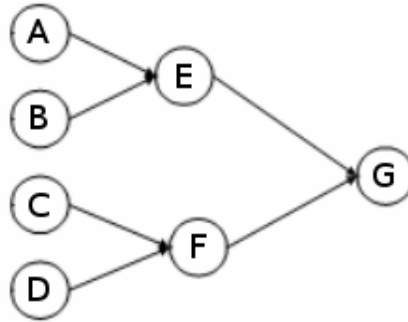
The arborescent product divergent structure

The general product network structure

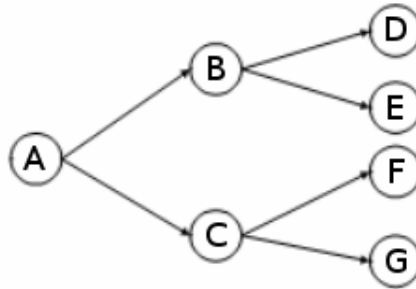
**Figure 2.3:** Serial product structure



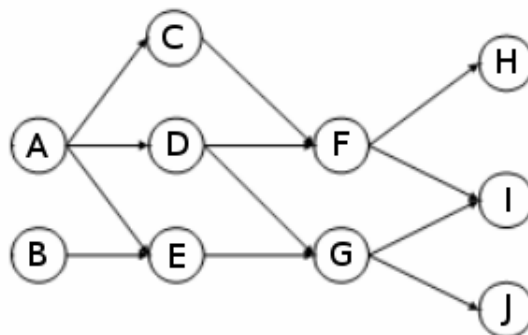
**Figure 2.4:** Assembly product convergent structure



**Figure 2.5:** Arborescent product divergent structure



**Figure 2.6:** General product network structure



**Figure 2.7:** Different product structures

The four structures are presented in Figure 2.7, where Figure 2.3 represents the

serial product structure, Figure 2.4 represents the assembly product convergent structure, Figure 2.5 represents the arborescent product divergent structure and Figure 2.6 represents the general product network structure. As seen in Figure 2.3, in the serial structure each item has a single predecessor and a single successor in the network. For example the item B has the item A as predecessor and the item C as successor. The serial product structure is studied by [Billington et al., 1994], where they assumed a linear product structure and uniform processing times without setup times. They applied sequentially a multi-item single-level specialized heuristic to each level of the problem to solve the problem. This type of problem frequently occurs in "process" industries, furniture making, and apparel manufacturing. In furniture manufacturing, for instance, one product line can consist of several products that must be processed through an identical sequence of machines with a setup between products due to the different materials being used [Billington et al., 1994]. For the assembly convergent structure represented in Figure 2.4, each item can be made from several predecessors and one successor. For example, the item E has two predecessors (A and B) and it has one successor which is the item G. [Blackburn and Millen, 1984] and [Simpson, 1999] and others, studied a different lot-sizing assembly systems and proposed several heuristics methods, where most of them based on the level-by-level approach found in Material Requirements Planning (MRP) system. [Blackburn and Millen, 1984] considered an assembly convergent system to study an alternative approach to the capacity planning problem, by integrating it with the MRP lot-sizing process. [Simpson, 1999] studied nine multiple level production planning heuristics under rolling horizon planning conditions. Four of nine evaluated heuristics in his study represents the sequential (level-by-level) application of single item lot-sizing technique. Four others represent these same algorithms applied in conjunction with the multiple level cost modification technique of [Blackburn and Millen, 1982]. The ninth lot-sizing rule is an iterative technique which begins with a lot-for-lot solution proposed by him in [Simpson and Erenguc, 1998]. Then, he proposed a tight lower bound to evaluate these heuristics. His lower bound is similar to much of some mathematical programming developed for multiple level production planning proposed in ([Afentakis et al., 1984], [Crowston and Wagner, 1973], [Zangwill, 1969]). In contrast, the arborescent divergent structure represented in Figure 2.5 has at most one predecessor, but may have any number of successors. For example, the item B has one predecessor which is A and two successors are (D and E). The arborescent network is the prototype for a distribution system, whereby each node is represented as a warehouse may have at most one immediate predecessor as a distribution center, but possibly may have more than one successors seen as customers. The arborescent system was studied by ([Maxwell and Muckstadt, 1985] and [Muckstadt and Roundy, 1987]), in which the end-item (the customer demand) is constant. They have focused on the relationship of the reorder intervals at intermediate stages to the end-item reorder intervals. These analytic treatments lend insight to the overall problem, but do not appear easily adaptable to the more common situation of time-varying demand. [Bookbinder and Heath, 1988] have em-

ployed a simulation model to study time-varying demands in a pure arborescent system. In a recent work focused on a part of it on the pure arborescent system, [Bookbinder and Koch, 1990] studied a complex manufacturing system often has arborescent or non-assembly portions in its network. They compared the pure arborescent system with the pure assembly systems, then they proposed a mixed assembly/arborescent systems. They concluded that the pure arborescent system could give an optimal total cost, if cost revision is not applied. The arborescent system could be used in a sub-assembly line in automotive company and in the distribution center. Finally with the general network structure represented in Figure 2.6, here each item can be made from several predecessors and can have several successors. For example, item F has two predecessors are (C and D) and two successors are (H and I). The general assembly structure is studied by [Wang et al., 2011], where they proposed an automotive general assembly plant and developed a data driven simulation method to integrate assembly, arborescent and serial structures with Just-in-Time (JIT) and Material Handling System (MHS). They developed a data driven approach to automatically generate simulation model based on manufacturing process parameters and configurations, and also real-time update the model based on dynamic production information during the simulation running phase. They applied their work to a typical automotive general assembly plant to quickly and automatically generate simulation model for production analysis. The objective of the MLLSP is to minimize the relevant costs which are the setup costs, the inventory holding cost and the production costs[Salomon, 1991].

In this thesis, we study the coordination of scheduling activities, which includes the coordinated lot-sizing and delivery scheduling problems. At the most general level, coordination can be seen in terms of integrating decisions of different functions (for example, inventory planning, production planning, distribution planning, etc.). [Bhatnagar et al., 1993] refer to this level of coordination as “general coordination”. At another level, the problem of coordination may be addressed by linking decisions within the same function at different echelons in the organization. They classified the research on general coordination into three categories as follow:

Supply and production planning

Inventory and distribution planning

Production and distribution planning

These categories represent integration of decision-making related to studies on coordination between supplier and customer(s) focused on determining the order quantity that is jointly optimal for both. Our work contributes to the second and third research categories. We consider complex supply chain with decisions on integrated production-delivery-inventory problems.



## 2.4 The integrated Production-Delivery-Inventory problems

An integrated production and distribution planning problem is the problem of simultaneously finding the decision variables from different functions that have traditionally been optimized independently [Seyedhosseini and Ghoreyshi, 2014]). Production planning tackles decision of how to transform raw materials into finished products respecting to meet demands on time with minimum cost. Determining lot sizes, that is, calculating the quantity to be produced for each item at each time, is an important decision in tactical production planning. On the other hand, distribution planning tackles decision of how to deliver the finished products to the customers respecting to meet their demands on time with minimum cost. Integrated production and distribution system often includes facilities producing the products and number of distribution centers warehousing products. Due to the number of decision variables to be determined, the integrated production and distribution planning problem is so complex that optimal values are very hard to obtain. In addition, considerations such as complex structure of the network, geographical span of the supply chain, and involvement of different entities with conflicting objectives can further complicate the problem [Pandey et al., 2007]. Indeed, simplification of a real-life scenario in developing a supply chain model has become unavoidable as most of the complex production delivery planning problems are classified under the category of NP-hard problems [Thomas and Griffin, 1996] and [Ajian and Lair, 2008].

### 2.4.1 Basic studied models

The integrated production and delivery with inventory costs planning was studied by several researchers. [Blumenfeld et al., 1985] determined optimal shipping policies (direct shipping, shipping via a consolidation terminal and a combination of terminal and direct shipping) by analyzing trade-offs between transportation, inventory and production setup costs. In another work, [Blumenfeld et al., 1986] considered the problem of scheduling production and distribution for a producer supplying parts to a final assembly manufacturer. They considered a very specific scenario that featured one destination per part type, identical production cycles with each production cycle including a production run of every part type, and fixed transportation costs per shipment. They showed that coordination can reduce costs by up to 42% and that maximum savings occur when the demand, item value and variable costs are the same for all items. [Blumenfeld et al., 1987] reported a successful implementation of this research at GM's Delco electronics division that resulted in a 26% (\$2.9 million per year) reduction in logistics costs. More recently, [Chandra and Fisher, 1994] studied a plant that produces a number of products over time and maintains an inventory of finished goods at the plant. The products are distributed by a fleet of trucks to a number of retail outlets at which the demand for each product is known

for every period of a planning horizon. They compared two approaches to managing this operation, one in which the production scheduling and vehicle routing problems are solved separately, and another in which they are coordinated within a single model. Their results showed that the reduction in total operating cost from coordination ranged from 3% to 20%, which indicates that the conditions under which companies should consider the organizational changes necessary to support coordination of production and distribution. [Fumero and Vercellis, 1999] proposed an integrated optimization model for production, distribution and inventory planning, with the aim of optimally coordinating important and interrelated logistic decisions such as capacity management, inventory allocation, and vehicle routing. They solved the integrated production distribution model via Lagrangian relaxation. [Bertazzi et al., 2005] considered a complex production-distribution inventory system, where a facility produces several items which are distributed to a set of retailers by a fleet of vehicles with a fixed transportation costs. They proposed two different policies: The order-up-to level policy, in which the order-up-to level quantity is shipped to each retailer whenever served (i.e. the quantity delivered to each retailer is such that the maximum level of the inventory at the retailer is reached) and the fill-fill-dump policy, in which the order-up-to level quantity is shipped to all but the last retailer on each delivery route, while the quantity delivered to the last retailer is the minimum between the order-up-to level quantity and the residual transportation capacity of the vehicle. Two different decomposition of the problem are proposed together with optimal or heuristic procedures for the solution of the sub-problems. They concluded that the fill-fill-dump policy reduces the average cost with respect to the order-up-to level policy and that one of the decomposition is more effective.

In the recent years, the integrated production-delivery-inventory problem has received a lot of attention. [Sarmiento and Nagi, 1999], [Fahimnia et al., 2013] and [Chen, 2010] provided comprehensive reviews on the general subject. [Chen, 2010] reviewed the production and distribution scheduling models and classified these problems in five classes: (1) models with individual and immediate delivery; (2) models with batch delivery to a single customer by direct shipping method; (3) models with batch delivery to multiple customers by direct shipping method; (4) models with batch delivery to multiple customers by routing method (5) models with fixed delivery departure date. In the first model, jobs have delivery windows, and thus production windows can be incurred, however, due to the immediate and individual delivery requirement, the problems under this model can be reduced to fixed-interval scheduling problems (without the delivery). For all other models, no production windows have been specially considered in the survey. Problems addressing an objective function that combines machine scheduling with the delivery costs are rather complex. However, they are more practical than those involving just one of the two factors, since these combined-optimization problems are often encountered when real-world supply chain management is considered. [Bard and Nananukul, 2009a] solved the production and delivery problem in three different works. In the first one, they formulated an integrated lot sizing and inventory routing problem as

a mixed integer program with the objective of maximizing the net, and they developed a two-step algorithm to estimate daily delivery quantities and then solved the problem for each day of the planning horizon [Bard and Nananukul, 2009a]. After this work, they proposed an algorithm centered on reactive tabu search in [Bard and Nananukul, 2009b] for solving a model with a single production facility, a set of customers with time varying demand, a finite planning horizon, and a fleet of vehicles for making the deliveries. In [Bard and Nananukul, 2010], they proposed a hybrid methodology that combines exact and heuristic procedures within a branch-and-price framework for an integrated production and inventory routing problem. These results showed that, the proposed hybrid scheme can solve medium and large instances with up to 50 customers and 8 time periods within 1 hour. This level of performance could not be matched by standard branch and price alone. [Fu et al., 2010] studied the problem of coordinated scheduling of production and delivery subject to the production window constraint and the delivery capacity constraint. They considered a single delivery time case and multiple delivery time case. On the first one the problem became a LSP and no delivery schedule is necessary. On the case of the multiple delivery time, the goal is to find a feasible coordinated production and delivery schedule whose total profit is close to optimal which means that it is not optimal.

### 2.4.2 Extension of the basic models

Determining the optimum batch size is related to the determining of the ordering policy for procurement of raw materials, the production schedule, the level of inventory, distribution batch size etc. In the classical ordering policy, it is often assumed that the shortages are either completely backlogged or completely lost. In reality, often some customers are willing to wait until replenishment, especially if the wait will be short, while others are more impatient and go elsewhere. The relation between the order-size and the batch-size is very complex, whereas the batch-size is a solution of the order-size. The batch-size could be represented in the different stage of the supply chain, but the order-size is represented mostly in the first stages under study. [Chuang et al., 2013] studied an integrated supplier-customer inventory model with an order-size-dependent trade credit and defective items. Their model is related to the customer's order quantity; the customer can fully delay its payment if its order reaches a threshold quantity, but otherwise may only partially delay its payment. They proposed an iterative algorithm to solve the problem. They found that the supplier should set the proportion of partial delay payment and the threshold quantity more careful, therefore the supplier can avoid the more loss in profit and to attract the sales more effective. Most of authors assumed that the equal batches sizes in their works to prevent the complexity of the batch-size problematic. The equal batches sizes is presented by [Szendrovits, 1975]. His work is extended by [Goyal, 1976], where they fixed the transportation cost with a regardless of the batch-size. After that, [Szendrovits and Drezner, 1980] developed

the equal-sized batch and showed that this assumption reduces the total costs in a multi-stage production-inventory system to reduce the work-in-process. In the early nineties, [Sarker and Parija, 1994] developed an ordering policy, for raw materials to meet the requirements of a production facility under a fixed quantity, periodic delivery policy. They considered that the manufacturer is allowed to place only one order for raw material per cycle. They fixed the delivered batch-size and the delivery time in a fixed interval. An exact algorithm is proposed to solve the problem. In another work, [Sarker and Parija, 1996] developed a model with multiple lots of raw material for one lot of the product. The objective is to minimize the total cost for meeting equal shipments of the finished products, at fixed intervals, to the customers. In a recent work, [Sarker and Khan, 1999] considered a similar model to ([Sarker and Parija, 1994] and [Sarker and Parija, 1996]), but without the assumption of delivery of final product to the customer. They assumed that the production order-size equal to the delivery batch-size, and a product cannot be delivered until the whole lot is finished and quality certification is ready. In a recent work, [Bogaschewsky et al., 2001] solved the problem of multi-stage production/inventory system where subsequent batch shipments are of unequal size. Due to the complexity of the model, it required developing algorithms for finding the optimal solution. [Hill and Omar, 2006] relaxed the assumption said that the first  $k$  shipments increase in size, and that the last  $n - k$  shipments are of equal sizes, studied by [Hill, 1999]. [Hill and Omar, 2006] relaxed the assumption that the customer's inventory carrying charges exceed those at the supplier. The multi-product lot-sizing problem with sequence-dependent setup times is studied by [Clark and Clark, 2000]. They formulated a mixed-integer programming model to solve the problem. The sequence-dependent setup times and cost is examined by [Transchel et al., 2011], where they studied the problem of multi-product production planning and scheduling. [Sajadieh et al., 2010] investigated an integrated production-inventory model for two-stage supply chain (supplier–customer) where the demand of customers at the customer is stock-dependent. [Grunder, 2010] considered a single-product batch scheduling problem with the objective of minimizing the sum of production, transportation and holding cost. Particularly, he assumed that the delivery time depend on the batch sizes and proposes a dynamic programming approach based on a dominance relation property. [Wang et al., 2013] have extended this study to an integrated scheduling problem for single-item supply chain involving due date considerations with the objective of minimizing the total logistics cost.

[Khanra et al., 2011] developed an EOQ (Economic Order Quantity) model for a deteriorating item having time dependent demand when delay in payment is permissible. The deterioration rate is assumed to be constant and the time varying demand rate is taken to be a quadratic function of time. [Mishra and Sahab Singh, 2011] developed the inventory model for deteriorating items with time dependent linear demand and holding cost. Most recently, [Glock, 2012] studied a single-supplier–single-customer integrated inventory model with lot size-dependent lead time. They as-

sumed that lead time consists of production and setup and transportation time. They showed that lead time reduction is especially beneficial in case of high demand uncertainty and that may either be reduced by shortening setup and transportation time or by increasing the production rate, which results in a reduced production time. [Pal et al., 2015] worked on price and stock dependent demand for deteriorating item under inflation and delay in payment. [Swami et al., ] considered an economic ordered quantity model for deteriorating item with stock dependent demand and holding cost per unit time.

## 2.5 The integrated Delivery-Inventory problems

The distribution planning tackles decision of how to deliver the finished products to the customers respecting to meet their demands. The delivery and inventories management have a decisive influence on the effectiveness of the distribution process. Previously, the delivery and the inventory management in the distribution process optimization was usually considered independently with a negligible mutual impact. However, the integrated delivery-inventory problem has recently motivated some authors to model these two activities simultaneously, in the objective to minimize the total delivery-inventory costs. The delivery-inventory problem is denoted as Vendor-Managed Inventory (VMI) problem. [Borade and Sweeney, 2014] says that, the VMI problem is a widely used collaborative inventory management policy in which manufacturers manages the inventory of retailer and takes responsibility for making decisions related to the timing and extent of inventory replenishment. VMI partnerships help organizations to reduce demand variability, inventory holding and distribution costs.

A pioneering paper is due to [Bertazzi et al., 1997], where a given set of shipping frequencies is allowed and different products may be shipped at different frequencies. Several heuristic algorithms, based upon the idea of first solving single link problems and then locally improving the solution, are proposed. The effectiveness of these proposed heuristics has been evaluated according to [Burns et al., 1985], where [Bertazzi et al., 1997] showed that the efficiency of these proposed heuristics, in terms of the required computational resources, is certainly lower than the efficiency of different approaches, such as the one based on the method by [Burns et al., 1985]. [Herer and Levy, 1997] considered a system of a central warehouse, a fleet of trucks with a finite capacity, and a set of customers, for each of whom there is an estimated consumption rate, and a known storage capacity. The objective is to determine when to service each customer, as well as the way to be performed by each truck, in order to minimize the total discounted costs. To solve the problem, they proposed a rolling horizon approach that takes into consideration holding, transportation, fixed ordering and stock out costs. [Viswanathan and Mathur, 1997] considered a distribution systems with a central warehouse and many retailers that stock a number of different products, where the products are delivered from the warehouse to

the retailers by vehicles that combine the deliveries to several retailers into efficient vehicle routes. they proposed a heuristic that develops a stationary nested joint replenishment policy. These results showed that, the proposed heuristic is capable of solving problems involving distribution systems with multiple products. [Sindhuchao et al., 2005] considered a system that consists of a set of geographically dispersed suppliers that manufacture one or more non-identical items, and a central warehouse that stocks these items. The warehouse faces a constant and deterministic demand for the items from outside retailers. The items are collected by a fleet of vehicles that are dispatched from the central warehouse. The vehicles are capacitated, and must also satisfy a frequency constraint. They studied the case where each vehicle always collects the same set of items. They formulated and solved the problem by using a branch-and-price algorithm, then they proposed a greedy constructive heuristics and a very large-scale neighborhood search algorithm. These results indicates that the constructive heuristics used in conjunction with one of the proposed very large-scale neighborhood algorithms can find near-optimal solutions very efficiently. [Archetti et al., 2007] considered a distribution problem in which a product has to be shipped from a supplier to several retailers over a given time horizon. Each retailer defines a maximum inventory level. The supplier monitors the inventory of each retailer and determines its replenishment policy, guaranteeing that no stock out occurs at the retailer (supplier-managed inventory policy). Every time a retailer is visited, the quantity delivered by the supplier is such that the maximum inventory level is reached (deterministic order-up-to level policy). Shipments from the supplier to the retailers are performed by a vehicle of given capacity. They presented a mixed-integer linear programming model and they derived new additional valid inequalities used to strengthen the linear relaxation of the model. They implemented a branch-and-cut algorithm to solve the model optimally. Then, they studied two different types of replenishment policies in [Archetti et al., 2011]. The first one is the well-known order-up to level (OU) policy, where the quantity shipped to each retailer is such that the level of its inventory reaches the maximum level. The second one is the maximum level (ML) policy, where the quantity shipped to each retailer is such that the inventory is not greater than the maximum level. They showed that, when the transportation is outsourced, the problem with OU policy is NP-hard, whereas there exists a class of instances where the problem with ML policy can be solved in polynomial time. They also showed the worst-case performance of the OU policy with respect to the more flexible ML policy. In this study, [Archetti et al., 2011] focused on the ML policy and the design of a hybrid heuristic, they implemented an exact algorithm for the solution of the problem with one vehicle and designed a hybrid heuristic for the multi-vehicle case. These results showed that the proposed heuristic can produce high quality solutions in a very short amount of time. In [Archetti et al., 2012], they studied the previous problem with a single vehicle which has a given capacity. The transportation cost is proportional to the distance traveled, whereas the inventory holding cost is proportional to the level of the inventory at the customers and at the supplier. They proposed a heuristic that combines a tabu search scheme with ad hoc designed mixed-integer

programming models. The effectiveness of the heuristic was proved over a set of benchmark instances for which the optimal solution was known.

### 2.5.1 Basic studied models

In the supply chain management literature, the single-supplier single-customer problem has received a lot of attention in recent years as it is the building block for the wider supply chain. The global supply chain can be very complex and link-by-link understanding of joint policies can be very useful [Ben-Daya et al., 2008]. The integrated single-supplier single-customer problem is called the joint economic lot sizing problem (JELP). This term is coined by [Banerjee and Banerjee, 1994]. The basic policies considered in the literature for JELP were based on the combination of the equal-shipment sizes and the increasing shipment sizes. The reason for considering these two policies is that the requirement of equal-shipment sizes is good from the customer's perspective. However, having a small first shipment followed by larger ones leads to a small minimum system stock, which means a reduced average system inventory.

One of the early works related to JELP was due to [Goyal, 1977], he proposed a simple model with infinite production rate and lot-for-lot assumption. In his research, the supplier produces in lots and sends the entire lot to the customer. [Banerjee and Banerjee, 1994] kept that lot-for-lot policy, but relaxed the assumption of infinite production rate. [Goyal, 1988] contributed to the efforts of generalizing the problem by relaxing the lot-for-lot policy. He assumed that the production lot is shipped in a number of equal-size shipments, but only after the entire lot has been produced and he showed that his joint total relevant cost is less than or equal to that of the JELP model. In [Goyal, 1995], a different approach than equally-sized shipment to come up with an idea of geometric shipment size is used. This means that the successive shipment size is the product of the prior shipment size in relation to the ratio of production and demand rate. In this paper, the author formulated the problem and gave the optimal expression for the first shipment size as a function of the number of shipments. [Viswanathan, 1998] called the model of [Goyal, 1995] as the "deliver what is produced" (DWP) policy, and showed the model studied by [Lu, 1995], where he called it as "Identical delivery quantity" (IDE). [Viswanathan, 1998] concluded that neither policy is better than the other for all type of problems, and the best policy depends on the problem's parameters. [Hill, 1997] generalized the model of [Goyal, 1995] by taking the geometric growth factor as a decision variable. He showed numerically that his policy outperforms both the equal-shipment-size policy and the policy adopted by [Goyal, 1995]. [Goyal, 2000] extended the work of [Hill, 1997], by proposed that the following shipment sizes will be determined by first shipment size. The following shipment sizes are depended of the production rate. The resulting improvement was demonstrated with a small number of experiments. It was unclear whether the improvement was in general significant, or for what kind of problems it is so.

In a recent work, [Goyal and Nebebe, 2000] suggested that among “ $n$ ” shipments, the first shipment is smaller and followed by  $(n - 1)$  equal-sized, which is equal to the product of the first shipment size as well as its rate of production over rate of demand. [Pan and Yang, 2002] extended the work of [Goyal, 1988], where they argued that better customer satisfaction levels and reduced safety stock levels can be achieved through improving lead-time. However, these changes occur at the expense of lead-time crashing cost. In a recent work, [Zanoni et al., 2012] compared different policies (equal or nearly equal in size). They considered two issues, which are the “VMI with consignment” inventory policy and the “Learning Curve”. They showed how learning in production can give flexibility to the supply chain stakeholders in assigning the size and the time of each shipment. In [Bylka, 2013] the assumption on equal in size deliveries has been relaxed. He presented a continuous deterministic model, and to satisfy the customer’s demands, he supposed that the product is delivered in discrete batches from the supplier’s stock to the customer’s stock and all shipments are realized instantaneously.

### 2.5.1.1 Problems with different shipment policies

The papers listed above incorporated the problem under an assumption on shipment policy, in which the sizes of successive shipments either are increased by a factor or are equal in size. [Hill, 1999] combining these two type of policies, derived the structure of a globally optimal production distribution cycle. He showed that the structure of the optimal policy includes shipments increasing in size according to a geometric series followed by equal-sized shipments. He also suggested an exact iterative algorithm for solving the problem. [Hill, 1999] was revisited by [Hill and Omar, 2006], who relaxed the assumption that the customer’s inventory carrying charges exceed those at the supplier. This is the case especially if the supplier is a small specialist manufacturer with little storage space available and the customer a large company, which possesses low-cost storage facilities. They showed how the optimal batch shipment policy may be derived when unit stock-holding costs increase as the product moves down to the customer under non-required equal shipping size. An alternative solution technique for the problems studied by [Hill, 1999] and [Hill and Omar, 2006], which guarantees that the optimal solution is always found, can be found in [Hoque, 2009]. The model studied in [Hill, 1999] and revisited by [Hill and Omar, 2006], has been modified by [Zhou and Wang, 2007], where they refuted the assumptions that customer’s unit holding cost is greater than the supplier’s or not, and they also allowed shortages but only for customers. They showed that their model performs better in reducing the average total cost regardless of the supplier’s or customer’s stock-holding costs, which are never equal to each other. In this study they observed that it is more beneficial for the integrated system to make the supplier’s holding cost higher than the customer’s than to make the supplier’s holding cost lower than the customer’s if shortages are not permitted to occur, otherwise it just reverses. In a most recent work, [Bylka and Górný, 2015]



developed a continuous deterministic model with centralized decision process, and proposed two types of shipment strategies, neither required equal in size of all deliveries nor assumed the delivery cost per shipment is the same. Further, they considered that the stock level at the customer as a decision with respect to a possible choice of shipment policy used. Their developed model is realistic and general in the sense that the well know results such as [Braglia and Zavanella, 2003] and [Zanoni and Grubbstrom, 2004] are special case of their proposed model.

In all models above, it has been implicitly assumed that the transportation cost is part of the ordering cost. This assumption is not in general valid because transportation cost is affected by the routing decisions and the selected shipment size. [Baumol and Vinod, 1970] introduced freight cost into an inventory model. They proposed two models for inventory cost minimization and profit maximization. In those models, they include freight rates, speed, variance in speed, and the en-route losses in an order-sizing model. They assume that unit shipping cost is fixed and is not dependent on the shipment size. [Abad and Aggarwal, 2005] studied the problem of determining the re-seller's lot size and pricing assuming that the re-seller is responsible for paying for freight. They assumed that the final demand for the product is sensitive to the selling price that the re-seller sets. Another paper that considers transportation cost is [Ertogral et al., 2007]. They developed two models that integrate the transportation cost explicitly in the single-supplier/single-customer problem. For the first model, the transportation cost is considered to be in an all-unit-discount format. In the second model, they explored the option of over declaring a shipment to exploit the transportation unit cost structure. They concluded that the explicitly integrating transportation cost could be realize a savings, and could affected on the inventory decisions.

Few research works have been conducted in the context of the integrated delivery-inventory problems with multi-transporters, assuming generally that a transporter is always available to deliver the products. Consequently, such problems can be classified into two classes based on the vehicle availability. The first class assumes that there are an infinite number of vehicles available at each departure time, whereas the second class assumes that there are a limited number of vehicles available at each departure time. [Wang and Lee, 2005] and [Stecke and Zhao, 2007] have considered problems with different types of vehicles. In these both papers, vehicles are heterogeneous due to different characteristics, like their delivery speed, cost or capacity. Mainly, it is assumed that the fastest vehicle is more costly. [Omar, 2009] considered an integrated production-delivery system to determine the number and sizes of shipments which minimizes the total cost assuming the supplier and customer collaborate and find a way of sharing the consequent benefits. In this case, the optimal policy may be derived when the shipments size are identical. [Merzouk et al., 2008] studied the multiple transporters in a simple supply chain with a so-called width-first search Branch and Bound. The manufacturer cost is considered to be the same for all jobs and linear with respect to the holding time. However the proposed exact method could not solve the medium to large size instances. In all models above, it is

assumed that the transportation of the products has processed without the routing of the transporters between the customers. Another major research direction considered the case of one-supplier/multi-customer supply chains with inventory routing problem.

### 2.5.1.2 Problems with due date objectives

In this subsection, we reviewed the integrated problems with due date constraints. The due-date assignment problem has already been considered as a critical factor during the negotiation process in supply chains. When the demand is high, these lead times will be understated leading to missed due dates and disappointed customers, or to higher costs due to expediting. When the demand is low, they will be overstated and some customers may choose to go elsewhere. In scheduling theory, one of the most important criteria for measuring the performance of schedules is related to due dates or deadlines. Performance measures involving due dates model the informal criteria that are applied by practitioners. This makes these objective functions a very attractive and widely explored subject of research [Sterna, 2011].

In practice, the scheduler may need an efficient method for quoting the due dates and job scheduling. Thus, an increasing number of studies have considered the due date assignment as a part of the scheduling problem, where they showed that the determination of due dates can be a major factor in improving system performance. Several methods have been proposed to establish and specifying an optimal due dates or to solve scheduling problems with due-date constraints. The due-date assignment procedure is concerned with the performance of some scheduling rules. The problems with due date determination have received considerable attention in the last three decades due to the introduction of new methods of inventory management such as just-in-time concepts [Hsu, 2013]. Early research on due date assignment in single-machine scheduling is conducted by [Seidmann et al., 1981] and [Panwalkar et al., 1982], where they considered the constrained version where the scheduler must decide on a common due date for all the jobs (this due date assignment method is usually referred to as common due date assignment method). Many studies have focused on the common due date assignment problem. For reviews of scheduling models that consider common due date assignment with practical applications, the reader can refer to ([Lauff and Werner, 2004], [Cheng and Gupta, 1989] and [Gordon et al., 2002]). While [Seidmann and Smith, 1981] study the unrestricted case where each job can have a different due date. They proposed a method to identify due date for incoming order. They presented an analytical formulation of a dynamic single-machine scheduling problem with an objective of minimizing expected aggregate cost per job. These studies have inspired extensive research on due date assignment in scheduling as [Cheng and Gupta, 1989], [Gordon et al., 2002] and [Gordon et al., 2012]. Most of researchers in the eighties and early nineties have dealt with due date decisions under a single product and single stage production system. In the early years of the current century [Van Ooijen and Bertrand, 2001]

studied the investigation of the setting of optimal cost internal due dates for determining the priorities on the floor shop.

The due-date management or orders with availability intervals and lead time sensitive revenues in a single-server setting is studied by [Keskinocak et al., 2001]. In their model, revenues obtained from the customers are sensitive to the lead-time, there is a threshold of lead-time above which the customer does not place an order and the quoted lead times are 100% reliable. They showed that different strategies for quotation are needed for different categories of customers. The benefits of lead time flexibility to the manufacturer in an offline setting with 100% service guarantees have been studied by [Charnsirisakskul et al., 2006]. They presented an optimization model for due date and price sensitive customers under deterministic demand function. The pricing and lead-time decisions from a supplier-retailer perspective considered by [Liu et al., 2007]. In their model, the demand is sensitive to price and lead-time decisions and it is considered as deterministic. [Kaminsky and Kaya, 2008] analyzed a make-to-order supply chains and design effective scheduling for the centralized and decentralized versions of those systems. They developed due date quotation model in the supply chain environment where it is affected by the performance of supplier. A problem with known average lead-time is studied by [Zorzini et al., 2008]. They investigated a current practice supporting capacity and delivery lead-time management in the capital goods sector, and they proposed a model to formalize the decision process for setting due dates in the selected cases. In two recent works, [Steiner and Zhang, 2007] addressed the minimization of the sum of the total weighted number of tardy jobs and delivery costs on a single machine based on the batch setup time for one customer, where they presented the optimal properties and a pseudo-polynomial time DP algorithm for obtaining the optimal solution. [Steiner and Zhang, 2011] presented a pseudo-polynomial algorithm for minimizing the sum of the common due date assignment costs, the weighted number of tardy jobs, and the batch delivery costs for a single customer. Most recently, [Rasti-Barzoki et al., 2013] proposed heuristic and branch-and-bound algorithms for solving the problem of studied by [Steiner and Zhang, 2011]. [Rasti-Barzoki et al., 2013] proposed studied multiple customers problem and proposed an integer programming method, a heuristic algorithm, and a branch-and-bound method. [Piya, 2015] proposed a method to quote the due date and the price of incoming orders to multiple customers simultaneously when the contingent orders exist. Their proposed method is based on two steps method to counter the uncertainty caused by contingent orders while preparing quotes for multiple new customers simultaneously.

The complexity of the problem of minimizing weighted number of early jobs on a simple supply chain is studied in different similar and convergent ways. [Karp, 1972] proved that the problem of minimizing weighted number of tardy jobs on a single machine is NP-hard in the ordinary sense even if all the jobs are subject to the same due dates. A survey of due-date-based research reveals that due dates are usually treated as given information and taken as input to a scheduling problem.

However, actually to respond to customers needs, the due date can be a decision variable within the domain of the scheduling problem and according to different related variables could be affected to the total cost as the delivery time, the delivery cost and the inventory cost.

## 2.5.2 Extended studied models

Although researchers have given a considerable attention on synchronization of the single-supplier single-customer integrated inventory system. Another major research direction extended the previous basic models in many different directions. The global supply chain can be very complex and link-by-link understanding of joint policies can be very useful [Ben-Daya et al., 2008].

A several authors have investigated the effect of the number of customers on the integrated problems. [Lu, 1995] developed a one-supplier multi-customer integrated inventory model, while [Parija and Sarker, 1999] extended their published work on single-supplier, single-customer, integrated production-inventory problems with lumpy delivery systems under perfect and imperfect production cycle situations. [Lu, 1995] argued that all the previous studies assumed that the supplier must know the customer's holding and ordering costs, which are quite difficult to estimate unless the customer is willing to reveal the true values. Therefore, the author considered another circumstance, in which the objective is to minimize the supplier's total cost per year, subject to the maximum cost that the customer may be prepared to incur. [Parija and Sarker, 1999] introduced the problem of determining the production start-time and proposed a method that determines the cycle length and raw material ordering frequency for a long-range planning horizon. The cycle-length is restricted to be an integer-multiple of all shipment intervals to the customers as an ideal situation, the solution to which may be sub-optimal. [Viswanathan and Piplani, 2001] proposed a model to study and analyze the benefit of coordinating supply chain inventories by means of common replenishment epochs or time periods. A one-supplier multi-customer supply chain is considered for a single product. Under their strategies, the supplier specifies common replenishment periods and requires all customers to replenish only at pre-determined time periods. However, the authors did not include any inventory cost of the supplier in the model. In most papers dealing with integrated inventory models, the transportation cost is considered only as a part of fixed setup or replenishment cost. [Ertogral et al., 2007] have studied how the results of incorporation of transportation cost into the model influence on better decision making under equal size shipment policies. A fundamental advance in the two-side cost structure is in recognizing how delivery-transportation costs apply to both sides. [Hoque, 2008] proposed three models for supplying a single-item from a single-supplier to multiple customers under deterministic demand by synchronizing the production flow with equal-sized batch transfer in the first two and unequal-sized batches transfer in the third. In the first two models, all batches

forwarded are of exactly the same size but the timing of their shipment is different. In the first assumption, the manufacturer transfers a batch to a customer as soon as its processing is finished, whereas in the second a batch is transferred to a customer as soon as the previously sent batch to the customer is finished. In the third model, the subsequent shipment lot sizes increase by the ratio of production rate and sum of demand rates on all the customers. [Zavanella and Zanoni, 2009] proposed a model for a single-supplier multi-customer system, integrated in a shared management of the customer's inventory, so as to pursue a reduction or the stability of the holding costs while descending the chain. [Hoque, 2011a] transferred the lot from a supplier to multiple customers with  $l$  number of unequal sized batches first; where the next one is a multiple of the previous one by the ratio ( $k > 1$ ) of the production and the total demand rates, followed by  $(n - l)$  number of equal sized batches. The equal sized batches are restricted to be less than or equal to the  $l$ th batch (the largest unequal sized batch) multiplied by  $k$ . The models developed were solved by applying Lagrangian Multiplier method. However, in cases of single-supplier single-customer or single-supplier multi-customer or multi-stage production, synchronization of the production flow by transferring the lot with equal and/or unequal sized batches was found to lead to the least total cost for some numerical problems. Although [Hoque, 2011a] served that purpose, it did not cope with the relaxation of the discussed impractical assumptions. Following this trend of synchronization, [Hoque, 2011b] developed two generalized single-supplier multi-customer integrated inventory models by accumulating the inventory at the supplier and customers independently, but with the traditional trend of ignoring the cost of benefit sharing. Transportation of each of the batches incurs a transportation cost. In order to implement the models by taking into account the industry reality, he also incorporate them with the relaxation of the discussed impractical assumptions. [Battini et al., 2010] developed a single supplier and multi-customer consignment stock inventory model in which many clients can establish a consignment stock inventory policy with the same supplier.

Recently, [Jha and Shanker, 2013] studied an integrated single-supplier multi-customer model with lead time reduction under independent normally distributed demand on the customers. They assumed that non-identical lead time of the customers and the inventory at the customers is reviewed using continuous review policy. [Hariga et al., 2014] analyzed Hoque's models *I* and *II* studied in [Hoque, 2011a], then they modified some of Hoque's models. [Hariga et al., 2014] compared the cost between the results of the models in [Hoque, 2011a] and [Zavanella and Zanoni, 2009], then they concluded that both models are not appropriate as they are using different functional forms of the total setup and ordering costs. Moreover, it is shown that Hoque's model yields impractical solutions for zero transportation costs. When the total setup and ordering cost was adjusted to be similar the one in Zavanella and Zanoni's model, Hoque's model resulted in a larger total cost per unit time. [Chiu et al., 2013] studied a single-producer multi-retailer integrated inventory system with scrap in production. They considered all random defective items produced

as scrap items and a multi-shipment policy was used to synchronously deliver finished items to multiple retailers in order to satisfy customer demands. An optimal production lot-size and shipment policy that minimized the expected system costs was derived with the help of a mathematical model.

### 2.5.2.1 Problems with incompatible job families

Generally, jobs having identical processing times belong to the same family, and jobs in different families cannot be processed together in the same batch. A synchronization of production, delivery and inventory schedules for the incompatible jobs is therefore considerably more involved. The first models proposed for this problems examined each stage separately by neglected the others stages. The production set-up costs are supposed negligible and shipments are supposed always made in full loads in the work of [Maxwell and Muckstadt, 1979]. The only costs affected by the production and transportation schedules in this case are inventory costs. They developed a method for minimizing total inventory costs. Since the early 1990s, an abundance of research has been done on the integrated problems with incompatible job families by taking into consideration more than one stage. [Uzsoy, 1995] studied a model with constant job sizes and proposed two different algorithm to solve the problem according to the release dates of the jobs. If the jobs are released at the same time, he provided exact polynomial time algorithms for the problems of minimizing make-span, total weighted completion time and maximum lateness on a single batch processing machine and heuristic algorithms for minimizing the above three objectives on parallel machines. When the jobs are subject to different release dates, he developed an exact polynomial time algorithm for the problem of minimizing make-span and a heuristic algorithm for the problem of minimizing maximum lateness. Their system could discover rules that are structurally similar to and competitive with the batch-apparent-tardiness-cost rule proposed by [Mehta and Uzsoy, 1998]. In another work, [Blumenfeld et al., 1991] studied a model with one supplier and multiple customers where the supplier produces multiple products, one for each customer. Each product is allowed to be produced multiple times within a production cycle. The authors studied the case of homogeneous products, where the production cycle time is an integer multiple of delivery cycle time. The delivery of multiple products to multiple customers was the motivation for research by (Hahm and Yano) in more than three works. First, they considered the one-supplier-one-customer model in [Hahm and Yano, 1992]. In this work, they assumed that the unit inventory holding cost at the supplier is the same as that at the customer. They formulated the problem as a nonlinear mixed integer program which is solved by a heuristic approach. In [Hahm and Yano a, 1995], they considered the multiple items case and a single shipment to deliver the products from the supplier to the assembler. Then, they relaxed the assumption set by [Hahm and Yano a, 1995] in [Hahm and Yano b, 1995], by allowed the multiple shipments through a production cycle which requires the multiple items to be partitioned into groups. Another policy

of delivery is proposed in [Hahm and Yano c, 1995] differs from that proposed by [Hahm and Yano a, 1995], in which multiple equally spaced shipments take place at each cycle. The work of [Hahm and Yano a, 1995] was extended by [Khouja, 2000]. He considered the problems of determining the production scheduling and distribution intervals for different types of components when a supplier provides different kinds of components. He used a volume flexible where component quality depends on both lot sizes and unit production times of each items. Recently, [Khouja, 2003] examined production sequencing and distribution scheduling in a single-product and multi-product supply chain when production intervals equal distribution intervals. Most recently, [Stecke and Zhao, 2007] described a storage constrained, inbound inventory routing problem for incompatible jobs families. They explained the importance of the storage space on the routing and the inventory decisions. A variation of the multi-product version which also considers a system called (many-to-one structure) was analyzed by [Moin et al., 2011]. They considered a finite horizon, multi-periods, multi-suppliers and multi-products where a fleet of capacitated homogeneous vehicles, housed at a depot, transport products from the suppliers to meet the demand specified by the assembly plant in each period. Similarly but with a many-to-many structure, [Ramkumar et al., 2012] studied a multi-item multi-depot IRP, and they proposed a mixed integer linear program to solve this problem. They showed that the MILP is limited on time, since small instances with only two vehicles, two products, two suppliers, three customers and three periods could not be solved to optimal within eight hours of computing time. [Coelho et al., 2013] reviewed the problem of the incompatibles jobs families in an IRP with more focus given to the methodological aspects. Then in [Coelho and Laporte, 2013], they considered a multi-product multi-vehicle, where it deals with share inventory capacity and shared vehicle capacity for all products. They solved the problem using branch-and-cut and the implementation is able to solve instances with up to five products, five vehicles, three periods and 30 customers. In the most recent work, [Coelho and Laporte, 2015] studied a wide range of routing problems with several compartments used for the delivery of several products spanning several period. They developed two models, and they could solve instances containing up to 50 customers, four compartments and 14 vehicles.

### 2.5.2.2 Models with inventory routing problems

Some papers are devoted to the case with constant deterministic, customer specific, demand over an infinite time horizon. In their papers ([Anily and Federgruen, 1990b], [Anily and Federgruen, 1990a] and [Anily and Federgruen, 1991a]), they analyzed fixed partition policies for the inventory routing problem with constant deterministic demand rates and an unlimited number of vehicles. In [Anily and Federgruen, 1990a], they considered a class of general routing problems where the cost of driving a route depends both on its length and the number of points visited on the route via some general cost function having two arguments. The paper described a class of simple

heuristics of complexity  $O(n \log n)$  which are shown to be asymptotically accurate if the cost function satisfies certain conditions. Then in [Anily and Federgruen, 1990b], they considered a distribution system with a depot and many geographically dispersed retailers each of which faces external demands occurring at constant, deterministic but retailer specific rates. In the objective of determining the feasible replenishment strategies (i.e., inventory rules and routing patterns) minimizing (infinite horizon) long-run average transportation and inventory costs. They restricted the problem to a class of strategies in which a collection of regions (sets of retailers) is specified. When one of the retailers in a given region receives a delivery, this delivery is made by a vehicle who visits all other outlets in the region as well (in an efficient route). They proposed a class of low complexity heuristics and a lower and upper bounds to solve the problem. Experimental studies showed that the proposed heuristics come close to the optimal solution even for problems of moderate size. For further information, see also the comment by [Hall, 1991] and the rejoinder by [Anily and Federgruen, 1991a]. The analyses work in [Anily and Federgruen, 1990b] was extended in [Anily and Federgruen, 1993], where the central depot also keeps an inventory. Here, the depot faces fixed ordering costs as well as inventory costs, and its replenishment strategy must be coordinated with the strategies of the customers. For the same class of replenishment strategies as in [Anily and Federgruen, 1990b] restricted to power-of-two structures, the proposed heuristic is almost asymptotically optimal. In a power-of-two-structure, the replenishment periods for all customers and the warehouse are power-of-two multiples of a base planning period, see [Roundy, 1985] for details on the power-of-two structure. In a later paper by [Anily and Federgruen, 1991b], details about calculating the lower bounds are described. Then in [Anily, 1994], she showed that the optimal solution can be bounded from below by a special partitioning problem whose solution can be given in closed form, and she proposed a simple heuristic in [Anily, 1994]. Recently, [Anily and Bramel, 2004] derived lower bounds for the special case when a customer can only belong to one region. A probabilistic analysis of the performance of this bound demonstrates that it is asymptotically 98.5 % effective. [Zhao et al., 2007] focused on the integrated system with a warehouse responsible for the replenishment of a single item to the retailers with demands occurring at a specific constant rate, combining deliveries into efficient routes. They proposed a fixed partition policy for this type of problem, in which the replenishment interval of each of the retailer's partition region is accorded the power of two principles. To solve the problem, they proposed a lower bound and they compared it to a tabu search algorithm. These results revealed the effectiveness of the policy as well as of the proposed algorithm. In another work, [Raa and Aghezzaf, 2009] assumed a deterministic constant customer demand rates and therefore, they adopted a long-term cyclical approach. Further, [Raa and Aghezzaf, 2009] took into consideration the realistic side-constraints such as limited storage capacities, driving time restrictions and constant replenishment intervals. They proposed heuristic algorithm to solve the problem, which is analyzed and evaluated against a comparable state-of-the-art heuristic as [Viswanathan and Mathur, 1997].



The Inventory-Routing Problem (IRP) is among one of the model types in the literature that are applied to this kind of problems. In such problems, distribution of a product(s) from a central location to multiple geographically dispersed customers using limited/unlimited number of capacitated vehicles is considered. The decisions to be made are (1) when to deliver to each customer, (2) how much to deliver to each customer at each time, and (3) how to serve customers using the vehicles [Bertazzi et al., 2008]. [Bertazzi et al., 2013] introduced the inventory routing problems with multiple customers in the case the demand of multiple customers has to be satisfied. They presented and discussed the basic problem of the IRP, where one vehicle only is available. Then, they considered the extension of this problem to the multi-vehicle case. Recently, an interesting work has been published in [Coelho et al., 2012a], [Coelho et al., 2012b] and [Koc et al., 2013], where in the first one, [Coelho et al., 2012a] introduced the concept of consistency in the Inventory-Routing Problem with Transshipment (IRPT) solutions. The multi-vehicle IRP, with and without consistency requirements, is formulated as a mixed integer linear program. A meta-heuristic, that applies an adaptive large neighborhood search scheme where some sub-problems are solved exactly, is implemented. These results showed that the problem is very difficult to solve exactly by the MIP model and the proposed heuristic can produce high quality solutions within reasonable computing times. They showed also the importance of the used of the transshipment, which allows to reduce the solution cost significantly on these instances, depending on the ratio between the unit transshipment cost and the cost of using the supplier's vehicle. The case with transshipment is analyzed in [Coelho et al., 2012b] where an adaptive large neighborhood search heuristic is proposed. [Koc et al., 2013] analyzed the manufacturer's planning problem under different delivery policies and different types of orders. The orders can be split or consolidated, and two types of vehicles are considered. The first type of vehicle is expensive and available in unlimited numbers. The second type, which is less expensive, has limited and time-varying availability. In this work, the issue of multi-transporters and its importance are mentioned, however no resolution method has been proposed. [Leung and Chen, 2013] fixed the delivery departure dates in the case of a single machine and a set of orders. The completed orders are then delivered to manufacturers by a finite number of vehicles. Three objectives are used. The first one is to minimize the worst-case delivery performance relative to the due dates. The second and third objectives go one step further by considering both the worst case delivery performance and the total transportation cost.

## 2.6 Conclusion

The state of the art briefly recalls the development of the supply chain management systems, starting from the representation of the lot-sizing and batch-scheduling problems with an overview of the single and multi-level models. Then, we reviewed the

integrated production-delivery-inventory problem and we cited the basic models and moving toward more advanced models. The rest of this state of the art focuses on the integrated delivery-inventory problem, where we reviewed the basic integrated models as the models with different shipment policies and the problems with due date objectives, moreover we cited numerous extensions of the integrated delivery-inventory models as the problems with incompatible job families and the models with inventory-routing-problems. We showed that each study has its own weakness. Realistically, most of the problems are constrained, however our models are different from many existing models and it can be applied to a variety of real-world industrial problems.

There are very few researchers considered constraints mentioned above in the integrated production-delivery-inventory problems or in the integrated delivery-inventory problems one at a time. In our work, we present two different models. In the first one, a single-supplier/single-customer integrated supply chain model is presented as a production-delivery-inventory considerations, with multiple transporters and unequal due dates constraints; this model considers unequal batch size dependent on times and costs, transporter time and costs dependent by each transporter, a capacity constraint and a customer holding cost to each product arrive before its due date. In the second model, a single-supplier/multi-customer integrated model with unequal due dates constraints; this model considers a holding cost dependent by each customer, a transporter time and cost dependent also by customer, a capacity constraint, and the model is considered with unequal batch size. In our proposed models, the holding cost at the supplier could be higher or lower than at the customer; the transporter time and cost are considered different from a transporter to another; the transporter time and cost at the customers are different; the holding cost at the customers could be different from than other customers; the due dates related to each jobs could be equal or unequal from a job to another.

This work emphasizes the modeling and algorithmic development of optimization models for coordinated scheduling. Since most of the problems that we are dealing with high computational complexity. We focus on the development of efficient and effective heuristic algorithms for those problems, in the objective is to minimize the total cost in the supply chain, within the different constraints under assumption to guarantee a certain customer level. In each of the problems studied, we also conduct computational experiments to test the performance of our algorithms and discuss the insights obtained from the experimental results.

# 3 Multi-Customer with Single-Transporter Integrated Lot Sizing and Delivery Scheduling Problem

## 3.1 Introduction

The expansion of supplier to accommodate the maximum number of customers is considered as a key factor in the evolution of companies, in order to increase their profits. Classical scheduling problems did not consider delivery costs, so considering the coordination between the delivery costs and scheduling objective to minimize the total cost is an important point that researchers have paid attention to recently. One important benefit of this coordination is a more efficient management of inventories across the entire supply chain. In traditional inventory management, the optimal production and shipment policies for vendor and customers in a two-echelon supply chain are managed independently. As a result, the optimal lot size for the customer may not result in an optimal policy for the vendor, and vice versa. To overcome this difficulty, the integrated vendor–customers model is developed, where the joint total relevant cost for the customer as well as the vendors is minimized. Consequently, determining the production and shipment policies based on integrated total cost function, rather than customer’s or vendor’s individual cost functions results in reduction of the total inventory cost of the system. The system under study in this chapter, is inspired from a distribution system of a local pharmaceutical industry. In the pharmaceutical industry, contractors and generic manufacturers have been increasing their market share for the last decade and their competition advance relies largely on providing a fast response to their customers and being flexible in order to react quickly to changes in demand.

The increased competition has also forced other manufacturers in the same direction [Stefansson et al., 2011]. [Aptel and Pourjalali, 2001] stated that management of pharmaceutical supplies is one of the most important managerial issues in health care industries. However, many health care industries experience difficulty in managing their distribution-inventory pharmaceutical products. It is thus important to improve production schedules in order to strive for minimizing the total delivery-inventory costs between the vendor and the customers in a pharmaceutical industry

case.

Existing inventory models for multi-customers are not applicable to pharmaceutical products for several reasons. Pharmaceutical products can be more expensive than other products to purchase and distribute, and shortages and improper use of essential medicines can have a high cost in terms of wasted resources and preventable diseases and death. Therefore, special care should be taken in pharmaceutical inventory decisions to ensure 100% product availability at the right time, at the right cost, and in good condition to the right customers. The quality of health care industries strongly depends on the availability of pharmaceuticals on time. If a shortage occurs at a customer, an emergency delivery is necessary, which is very costly and can be implicated for patient health. Inventory management strategies that are unsuitable for health care industries may lead to large financial losses and a significant impact on patients. Hence, inventory strategies for pharmaceutical products are more critical than those for other products. Thus, a specific inventory model is necessary for control of pharmaceutical products to save patient lives and reduce unnecessary inventory costs.

In this chapter, we investigate a delivery-inventory supply-chain for multiple customers products in a customer. The studied system is composed of a central supplier which has to deliver products to different customers sites with a single transporter at given due dates. The objective is to reduce the overall cost which includes the delivery costs and an earliness penalty that is incurred for products which are delivered before their due dates. The problem studied in this chapter inspired from a real-world scheduling problem of a Central medical stores in the health systems.

The contributions in this chapter include enhancements to the batching and the delivery of batches of the central supplier in this context, which takes into account the different parameters related to the problem under study environment. Moreover, we propose an efficient genetic algorithm which is compared with a developed Branch and Bound algorithm and a Mixed Integer Programming model as an exact solutions. Two important advantages of this chapter should be emphasized. First, the resolution of real world scheduling problems helped to focus the research in the development of optimization models that can effectively be implemented. Second, although the scheduling models and solution approaches developed were motivated by a case-study, they can be applied to other types of industries, as long as the scheduling problem has a similar structure.

The rest of the chapter is organized as follows. In Section 2, the problem definition and the mathematical formulation is introduced, the notation used along this chapter are defined, then the problem is formulated as a mixed integer programming (MIP) model. In Section 3, we describe the proposed B&B algorithm as an exact method of resolution. In section 4 and 5, we developed an heuristic algorithm and a genetic algorithm for solving the problem, respectively. At last, in Sections 6 and 7, we provide the experimental results and draw some conclusions of this chapter.

## 3.2 Problem Definition and Mathematical Formulation

We consider a supply chain scheduling problem where there are  $m$  customers and one manufacturer in which each customer  $h$  orders a finite number of heterogeneous jobs to the manufacturer,  $\{J_1, J_2, \dots, J_n\}$ . The jobs have to be ready for delivery in batches by their common contracted due date for each job. In (Figure 3.1) a single supplier and multi-customer system has been depicted as an example. Each customer  $h$  orders a finite number of jobs to the supplier.

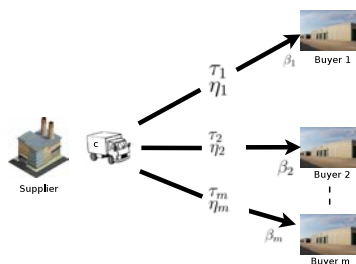


Figure 3.1

**Figure 3.1:** Single-supplier and multi-customer model

There is a single transporter considered to deliver the products and each round trip between the supplier and a customer  $h$  requires a delivery cost  $\eta_h$  as well as a delivery time  $\tau_h$ . The batches delivered from the supplier to the customers can be of different sizes. It is assumed that directing delivery method is used for sending the batches to the customers, which means that the jobs are transmitted to each customer separately [Chen, 2010]. The total number of jobs belonging to the same batch cannot exceed the capacity  $c$  of the transporter. Each job  $j$  has a due date  $d_j$  specified by the customers and each job has to arrive to the customer site before its due date. If job  $j$  of customer  $h$  arrives before its due date  $d_j$ , it will incur as an earliness penalty  $\beta_h$ . Batching and sending several jobs in the batches will reduce the transportation costs. The objective is to determine the sequence of batches that has to be processed, such that the expected total cost of both supplier and customers sites is minimized.

### 3.2.1 Notations

The following notations are used in developing the mathematical model:

*Parameters*

- $J = 1, 2, \dots, n$ : set of all jobs, where  $n$  is the total number of jobs,
- $H = 1, 2, \dots, m$ : set of all customers,
- $j$ : index for jobs,  $j \in J$ ,

- $k$ : index for batches,
- $h$ : index for customers,  $h \in H$ ,
- $d_j$ : due date of the job  $j$ ,
- $cl_j$ : destination of the job  $j$ ,
- $c$ : capacity of the transporter,
- $\tau_h$ : time for the vehicle to deliver a batch to customer  $h$  and to return to the supplier location,
- $\eta_h$ : delivery cost to deliver a batch to customer  $h$  and to return to the supplier location,
- $\beta_h$ : customer earliness penalty function for customer  $h$ ,

*Primary variables*

- $\delta_{jk}^1 = 1$  if the job  $j$  belongs to the  $k^{th}$  batch, 0 otherwise,
- $\delta_{kh}^2 = 1$  if the batch  $k$  belongs to the customer  $h$ , 0 otherwise,

*Secondary variables*

- $y_k = 1$  if the batch  $k$  exists and is not empty, 0 otherwise,
- $C_j$ : the arrival time of job  $j$  at the customer,
- $B_k$ : the arrival time of the batch  $k$  at the customer,
- $B_{jk}^\delta \geq 0$ : the result of  $B_k \times \delta_{jk}^1$ ,
- $k$ : index for batches,  $k \in K$ ,
- $K = 1, 2, \dots, u$ : set of all batches, where  $u$  is the total number of batches,
- $u_h$ : number of delivered batches for customer  $h$ ,

### 3.2.2 Mixed Integer Programming Model

In this research, the integrated model is developed for delivery and inventory policies. The first term aims to calculate the total delivery cost for  $n$  delivered products through  $\eta_h u_h$ . The second term calculates the inventory costs for all jobs arrived before their due dates through  $\beta_{cl_j}(d_j - C_j)$ . The objective function in equation (Equation 3.1) attempts to minimize the total cost of the system under study.

The supply chain has many aspects that need to be considered in a supply chain model. However, by taking all concerned factors into account, the model would be of so high complexity that it would be extremely hard for analysis. In this section, the mathematical programming model of the above mentioned problem is presented.

Using the structural properties, we develop a MIP model for the mentioned problem as follows:

$$\text{Min } Z = \sum_{h=1}^m \eta_h u_h + \sum_{j=1}^n \beta_{cl_j} (d_j - C_j) \quad (3.1)$$

Subject to :

$$\sum_{k=1}^n \delta_{jk}^1 = 1, \quad j = 1, \dots, n \quad (3.2)$$

$$\sum_{h=1}^m \delta_{kh}^2 \leq 1, \quad k = 1, \dots, n \quad (3.3)$$

$$\delta_{k,cl_j}^2 \geq \delta_{jk}^1, \quad j, k = 1, \dots, n \text{ and } k \leq j \quad (3.4)$$

$$u_h = \sum_{k=1}^n \delta_{kh}^2, \quad h = 1, \dots, m \quad (3.5)$$

$$y_k \geq y_{k+1}, \quad k = 1, \dots, n - 1 \quad (3.6)$$

$$\sum_{j=k}^n \delta_{jk}^1 \leq c, \quad k = 1, \dots, n \quad (3.7)$$

$$C_j \leq d_j, \quad j = 1, \dots, n \quad (3.8)$$

$$B_{k+1} \geq B_k, \quad k_1, k_2 \in J^2 \text{ and } k_1 < k_2 \quad (3.9)$$

$$B_{k+1} - B_k \geq \sum_{h=1}^m \tau_h (\delta_{k+1,h}^2 + \delta_{kh}^2), \quad k = 1 \dots n - 1 \text{ and } h \in H \quad (3.10)$$

$$C_j = \sum_{k=1}^n B_k \times \delta_{jk}^1, \quad j = 1, \dots, n \quad (3.11)$$

$$C_j \geq 0, \quad j = 1, 2, \dots, n \quad (3.12)$$

$$\delta_{jk}^1, \delta_{kh}^2 \in \{0, 1\}, \quad j, k = 1, \dots, n \text{ and } h = 1, \dots, m \quad (3.13)$$

The objective function (Equation 3.1) minimizes the sum of delivery and customers inventory cost. Constraint (Equation 3.2) guarantees that, each job must be scheduled exactly in one batch, in this constraint the jobs will be batched only in the batch which it belongs. Constraints (Equation 3.3 and Equation 3.4) indicate that, each batch must be scheduled exactly to one customer, in this constraint the batches will be delivered only to the customer which it belongs. Constraint (Equation 3.5) calculates the number of batches delivered to each customer. Constraint (Equation 3.6) guarantees that no empty batch is allowed. Constraint (Equation 3.7) prevents the number of jobs scheduled in one delivery batch to exceed the capacity of the vehicle. Constraint (Equation 3.8) indicates that arrival time of each job is at least equal to the contracted due date for each customer. Constraint (Equation 3.9) arranges the arrival time of batches by order. Constraint (Equation 3.10) shows the relation between the arrival time of two consecutive batches and the delivery time which their belong, to prevents the overlap between two consecutive batches in the case of two consecutive jobs delivered separately to two different customers. Constraint (Equation 3.11) represents the conversion of the completion time of the products that belong to the same batch  $C_j$  to  $B_k$  the arrival time of each batch to the customers. This constraint is represented in a non-linear way in this mathematical representation to facilitate the understanding of the problem. Constraints (Equation 3.12) and (Equation 3.13) define the range of the variables.

For ease of reference, we denote this problem: Multiple Customers Lot Sizing Delivery Scheduling Problem (MCLSDSP).

The complexity of the MCLSDSP is still an open question. To the best of our knowledge, no polynomial algorithm can solve this problem. However from simulation experiments, we observe that the problem is still intractable on an empirical basis. In the next section, a Branch-and-Bound (B&B) algorithm with a lower bound is described to solve the problem as an exact method.

### 3.3 Branch and Bound Algorithm

In this section, we describe the B&B algorithm that we have developed to solve the MCLSDSP. The objective of this B&B is to solve small and medium instances,



and to be a reference for validating the efficiency of the proposed heuristic algorithms. This B&B algorithm maintains a list of sub-problems (nodes) whose union of feasible solutions contains all feasible solutions of the original problem. The list is initialized with the original problem itself. In each major iteration, the algorithm selects a current sub-problem from the list of unevaluated nodes. This branching seems to be natural, however the number of branches will be very large for large problems. Consequently, if this method is used in the B&B algorithm, it may take too much time to find optimal solutions, as redundant schedules would be checked repeatedly. Yet, several of the sub-problems would already have been eliminated upon the generation of nodes, since the B&B tree includes redundant solutions.

At each node of the search tree, the number of products that still needs to be delivered to each customer has to be updated. Iterations are performed until the list of sub-problems to be processed is empty. The crucial part of a successful B&B algorithm is the computation of the lower bounds. Therefore, we have developed a lower bound described in the next part.

Efficient lower bound would significantly reduce the time and efforts needed for the B&B method. Based on the main feature of the problem, the lower bound value for the problem is the summation of lower bounds on the total earliness cost and the transportation cost. We assume that  $w$  is a partial batch sequence solution,  $z(w)$  is the evaluation of  $w$ , and  $r_h(w)$  is the number of products remaining at the customer  $h$  for partial solution  $w$ . This notation will be used throughout this part.

In each node, the solutions are built from the last batch to the first one and the evaluation of the partial or complete solution is processed with backward equations. The research of a solution starts by constructing a partial solution  $w$ . Then, the remainder of products is added in order to get a complete solution, with the objective of achieving a minimum delivery cost. Therefore, the more the transporter will be loaded, the more this lower bound will be efficient.

**Proposition 1** *For a partial solution  $w$ , a lower bound for the delivery cost of the remaining products is given by:*

$$\sum_{h=1}^m \left\lceil \frac{r_h}{c} \right\rceil \times \eta_h \quad (3.14)$$

*Proof.* For each customer  $h$ , if  $r_h(w)$  is the number of products remaining to be delivered, the number of round trips will be equal to  $\left\lceil \frac{r_h}{c} \right\rceil$ , and the delivery cost of the remaining products is as denoted in equation (Equation 3.14).  $\square$

We add the partial solution  $w$  to the solution found in equation (Equation 3.14) to get the lower bound of the current node under study.

**Corollary 1** *The lower bound  $LB(w)$  of the partial solution  $w$  is given as follows:*

$$LB(w) = z(w) + \sum_{h=1}^m \left\lceil \frac{r_h}{c} \right\rceil \times \eta_h \quad (3.15)$$

*Proof.* Straightforward. □

The mathematical model and the B&B algorithm developed in the previous sections could solve the small and medium size instances, however the time of resolution to solve the large size instances grows exponentially (demonstrated later, in section 6.3). Therefore, developing fast heuristics to yield near-optimal solutions in a reasonable running time is still of great importance. In the next section, a solving method is proposed to solve the problem and to study its performances.

### 3.4 Heuristic Algorithm

In this section, we proposed an efficient heuristic algorithm to solve the multi-customer lot-sizing and delivery-scheduling problem. This procedure is a heuristic, and therefore does not capture all solutions, possibly including the optimal solution to the problem. The proposed algorithm is divided on two i defines the lot sizes and then schedules these batches according to the due dates of the jobs. This heuristic will be denoted Lot Sizing and Scheduling algorithm (*LSS*) and will have the same operational characteristics assumed in the development of the proposed exact methods of the MCLSDSP.

The *LSS* algorithm starts by generating an initial solution through the means of a progressive constructive procedure. Then, it applies the above-mentioned two-steps process until a predefined stop condition is satisfied. At first, some elements of the current solution are constructed. Then, a local improvement phase based on a swap operator is applied to the reconstructed solution in order to improve its quality. Finally, *LSS* chooses the optimum solution between the current solution and the solution obtained from the improvement procedure.

Based on the prune rule, the following heuristic is proposed: lets denote that  $(x; cl_j; C_j)$  the notation which will be used for a solution of one batch, where the first term  $x$  describes the number of jobs in this batch, the second term  $cl_j$  describes the customers destination and the third term  $C_j$  is the arrival time of the product at the customer to which it belongs. For level 0, there are no jobs. For the first level (includes only one job), there is only one possible joint solution which is  $(1; cl_n; C_n)$ . For level  $k$  (includes  $k$  jobs), all the "good solutions" for a number of  $k$  jobs will be kept. The process to build the "good solutions" for level  $k$  is described as follows: (1) build delivery-inventory solutions of level  $k$  by considering all the delivery-inventory solutions in the retained "good solutions" of all the previous levels from 1 to  $(k - 1)$ . For each retained solution of level  $k' \leq k$ , a new solution of level  $k$  is built by simply adding a batch of  $(k - k')$  jobs, if this is possible. Then, repeat this procedure until the level  $n$  is considered.

The details of the Algorithm (Algorithm 3.1) are presented as follows: The generation of the initial solution and the construction procedure is represented from lines 1

to 4. Then, the size of batches procedure is represented from lines 5 to 17 according a scattering/gathering procedure. The improvement procedure is named in line 14, then it is described in Algorithm Algorithm 3.2.

---

**Algorithm 3.1** The Lot Sizing Scheduling Algorithm

---

```
for  $j \leftarrow 0$  to  $n$  do
   $currentJob \leftarrow n - j + 1$  ;
   $jobDest \leftarrow cl[CurrentJob]$  ;
   $nbrPartsDelivered [jobDest]++$  ;
  for  $level \leftarrow 0$  to  $j$  do
    if ( $TestOfCapacity()$ ) then
       $currentSol \leftarrow GetBestSol(level)$  ;
      for  $h \leftarrow 1$  to  $m$  do
        if ( $TestQuantityDeliv()$ ) then
           $AddBatch(currentSol)$  ;
        end if
      end for
       $Evaluate(currentSol)$  ;
       $Improvement(currentSol)$  ;
       $CompareAndUpdate(currentSol)$  ;
    end if
  end for
end for
 $bestSolution \leftarrow GetBestSol n$  ;
```

---

The lot sizing procedure, performed in an iterative way, extends a partial solution by adding one job from a set  $J$  of all jobs. The construction of the good solution advances progressively and in a hierarchical manner. The process starts from the last job and arrives recursively to the first one. The jobs are distributed to the customers to whom they belong, and the batches sizes are defined according to a scattering/gathering procedure described in Algorithm Algorithm 3.1.

In this algorithm, the number of delivered jobs  $j$  varies from 0 to  $n$  (line 1). For each level of  $j$  delivered jobs, the different partial solutions are built from the solution of previous levels ( $<j$ ). Moreover, the necessary number of batches to these solutions is added, to complete the partial solution of level  $j$ . After every product addition to level  $j$ , the partial solution of this level is completed by adding the necessary delivery scheme to the considered solution in the list of all kept solutions from 0 to  $(j - 1)$ , to obtain the new list of solutions of level  $j$ . A test of verification of the capacity of the transporter used is done directly after each advancement in  $level$  (See line 6 in Algorithm Algorithm 3.1). This destruction procedure leads to a list of solutions for level  $j$ .

**Algorithm 3.2** Improvement phase of the Lot Sizing Scheduling Algorithm

---

```

Input a solution currentSol
bestEval  $\leftarrow$  Evaluate(CurrentSol) ;
indexBatch  $\leftarrow$  NumberOfBatches - 1 ;
while (indexBatch  $\geq$  0 ) do
    Swap(currentSol, indexBatch , indexBatch + 1) ;
    if (Evaluate (currentSol) < bestEval) then
        bestEval  $\leftarrow$  Evaluate(currentSol) ;
        improve  $\leftarrow$  true ;
    else
        Swap(currentSol , indexBatch, indexBatch + 1) ;
        Evaluate(currentSol) ;
    end if
indexBatch  $\leftarrow$  indexBatch - 1 ; end while
    
```

---

The final step of each level  $j$  is denoted in line 15, which is mentioned in Algorithm Algorithm 3.1. In this phase the good solution is memorized and inscribed to level  $j$ . After that, a new level ( $j + 1, j + 2\dots$ ) is started till reaching level  $n$ . Each solution in level  $j$  is improved according to a swap operator that is described by Algorithm Algorithm 3.2.

In this operator, all consecutive batches are swapped, by starting from the last batch recursively to the first one, while the index of batch is positive (See line 6 in Algorithm Algorithm 3.2). After every swap operation, the new solution is kept, if it is better than the current. If not, a new swap operation is generated. The improvement operation stops, when the index of batches equals 0.

Let's take an example to explain the application of the *LSS* algorithm, to illustrate the MCLSDSP. We consider a problem of three jobs and two customers. The due dates associated with these jobs equal to 1000, 1100, 1150, where jobs 1 and 3 belong to customer 1 and job 2 belongs to customer 2. The transporter delivery cost and time depend upon the customer's location with ( $\eta_1 = 20, \tau_1 = 60 u.t$  and  $\eta_2 = 15, \tau_2 = 40 u.t$ ) belonging to customer 1 and 2 respectively. The storage costs at the customers are set to be ( $\beta_{cl_1} = \beta_{cl_3} = 30$  and  $\beta_{cl_2} = 15$ ), belonging to customer 1 and 2 respectively. The *LSS* process is described in detail as follows: The process starts by the last job recursively to arrive to the first one.

1. Set  $n = 1$ . For  $j = \{3\}$ , there is only one possible joint solution which it  $(1, cl_1 - 1150)$ .
2. Set  $n = 2$ . For  $j = \{3, 2\}$ , there are different possible solutions. Firstly a complementary solution is built by simply adding a batch of  $(2-1)$  job to the previous delivery solution. The potential delivery scheme is equal to  $[(1; cl_1; 1150), (1; cl_2; 1100)]$ . Due to the improvement phase, two solutions could be obtained according to the swap operation which are  $\{[(1; cl_2; 1100), (1; cl_1; 1150)]$  and

$[(1; cl_1; 1150), (1; cl_2; 1100)]$ }, then the two potential joint solutions are compared and the good solution is kept, which is  $[(1; cl_2; 1100), (1; cl_1; 1150)]$ .

3. Set  $n = 3$ . For  $j = \{3, 2, 1\}$ , based on the delivery scheme of the first step, the new delivery schemes are  $\{[(1; cl_1; 1150), (1; cl_2; 1100), (1; cl_1; 1000)]$  and  $[(1; cl_1; 1150), (1; cl_1; 1000), (1; cl_2; 1100)]$ . Then, two solutions could be obtained according to the swap operation which are  $\{[(1; cl_2; 1100), (1; cl_1; 1000), (1; cl_1; 1150)]$  and  $[(1; cl_1; 1000), (1; cl_2; 1100), (1; cl_1; 1150)]$ }, then the four potential joint solutions are compared and the good solution is kept, which is  $[(1; cl_1; 1000), (1; cl_2; 1100), (1; cl_1; 1150)]$ . after that, based on the delivery scheme of the second step, the delivery scheme on this step is  $[(1; cl_2; 1100), (1; cl_1; 1150), (1; cl_1; 1000)]$ . In the improvement phase, a new solution could be obtained which is  $[(1; cl_1; 1000), (1; cl_2; 1100), (1; cl_1; 1150)]$ . The two potential joint solutions are compared and the good solution is kept, which is  $[(1; cl_1; 1000), (1; cl_2; 1100), (1; cl_1; 1150)]$ . Finally, the two potential joint solutions kept in each level are compared and the best final solution is recovered, which is  $[(1; cl_1; 1000), (1; cl_2; 1100), (1; cl_1; 1150)]$ .

We provide in the next section a genetic algorithm started by the solutions found in the heuristic algorithm proposed above, in the objective is to improve the results of the approach method of resolution which could help us to solve large instances.

## 3.5 Genetic Algorithm

A genetic algorithm (GA) is a heuristic search algorithm that resembles natural selection. This algorithm was invented by [Holland, 1975] and developed and applied to different areas [Goldberg, 1989]. As the name suggested, GA is an evolutionary algorithm inspired by reproductive biology. As the name suggested, GA is an evolutionary algorithm inspired by reproductive biology. Genetic algorithms, when applied to scheduling, view sequences or schedules as individuals or members of a population. Each individual is characterized by its fitness. The fitness of an individual is measured by the associated value of the objective function. The procedure works iteratively, and each iteration is referred to as a generation. The population of one generation consists of survivors from the previous generation plus the new schedules, i.e., the offspring (children) of the previous generation. The population size usually remains constant from one generation to the next. The offspring is generated through reproduction and mutation of individuals that were part of the previous generation (the parents). Individuals are sometimes also referred to as chromosomes. The use of genetic algorithms has its advantages and disadvantages. One advantage is that they can be applied to a problem without having to know much about the structural properties of the problem. They can be very easily coded and they often give fairly good solutions.

### 3.5.1 Gene Representation

A genetic algorithm, as a search process, differs in one important aspect from other algorithms. At each iterative step a number of different schedules are generated and carried over to the next step. In other algorithms only a single schedule is carried over from one iteration to the next. Hence those algorithms may be regarded as special cases of genetic algorithms with a population size that is equal to 1. This diversification scheme is an important characteristic of genetic algorithms. In genetic algorithms the neighborhood concept is also not based on a single schedule, but rather on multiple schedules. The design of the neighborhood of the current population of schedules is therefore based on more general techniques than those used in other algorithms. A new schedule can be generated by combining parts of different schedules from the current population. A mechanism that creates such a new schedule is often referred to as a crossover operator.

There is numerous coding methods to represent the solutions of a genetic algorithm. In our case, a solution is a list of batches (and jobs composing batches) the transporter will deliver to precise customers in a determine order. For example the chromosome  $((1,c1),(3,c3),(2,c1),(3,c2))$  represent the solution of 4 batches : first 1 jobs are delivered to customer 1 then 3 jobs are delivered to customers 2 and so on. To code this form of solution, we use the number coding method; which is the most commonly used coding method. Each chromosome is a list of numbers which formed groups (called genes) of identical strictly positive integers and represent the number of jobs containing in a batch and the customers which will be delivered. Every batch is separate by a 0 to easily identifies the different genes. For instance, the previous example may be represent by the chromosome of Table 3.1.

1	0	3	3	3	0	1	1	0	2	2	2
---	---	---	---	---	---	---	---	---	---	---	---

**Table 3.1:** A chromosome representing a solution

### 3.5.2 Detailed algorithm

The algorithm we use is detailed in Figure 3.2. We start by creating a population of random individuals then use the crossover operator on some of them then mutating them depending of a dynamic mutation probability which is randomly chosen every twenty cycles. The best solution of the current population are determine then use to begin another cycle and create a new population until a chosen time limit or maximum cycles number is reached.

After deciding on an encoding, the second decision to make in using a genetic algorithm is how to perform selection-that is, how to choose the individuals in the population that will create offspring for the next generation, and how many offspring

---

**Algorithm 1.3** Find best solution with genetic algorithm

---

**Require:**  $lengthPop > 0 \vee maxIteration > 0 \vee timeLimit > 0$

$population \leftarrow bestSolution\ of\ LSS\ algorithm$

```

for  $j \leftarrow 0$  to  $maxIteration$  do
  if  $j \equiv 0 \pmod{20}$  then
     $mutationProbability \leftarrow computeDynamicMutationProbability(j)$ 
  else
     $parent1 \leftarrow chooseRandomIndividual(population)$ 
     $parent2 \leftarrow chooseRandomIndividual(population)$ 
     $offspring \leftarrow crossover(parent1, parent2)$ 
    for  $l \leftarrow 0$  to  $size(offspring)$  do
       $off \leftarrow getIndividualFrom(offspring, l)$ 
       $notok \leftarrow true$ 
      while  $notok$  do
         $mutate(off)$ 
        if  $evaluate(off) \geq 0$  then
           $add\ off\ to\ newPopulation$ 
           $notok \leftarrow false$ 
        end if
      end while
    end for
  end if
if  $timeElapse() < timeLimit$  then
  for  $k \leftarrow 0$  to  $lengthPop/2$  do
    if  $i = 0$  then
       $add\ bestIndividualForCurrentGeneration\ to\ newPopulation$ 
       $add\ bestIndividualForCurrentGeneration\ to\ newPopulation$ 
    end if
  end for
   $population \leftarrow newPopulation$ 
   $newPopulation \leftarrow \emptyset$ 
end if
end for

```

---

**Figure 3.2:** Genetic algorithm

each will create. The purpose of selection is, of course, to emphasize the fitter individuals in the population in hopes that their offspring will in turn have even higher fitness.

### 3.5.2.1 Selection

This operator selects chromosomes in the population for reproduction. The fitter the chromosome, the more times it is likely to be selected to reproduce. The selection process is based on the fitness value  $fi$  of the chromosomes. [Holland, 1975] original GA used fitness-proportionate selection, in which the "expected value" of an individual (*i.e.*, the expected number of times an individual will be selected to reproduce) is that individual's fitness divided by the average fitness of the population. The most common method for implementing this is "roulette wheel" sampling: each individual is assigned a slice of a circular "roulette wheel," the size of the slice being proportional to the individual's fitness. Roulette wheel selection scheme is used to

randomly choose, with replacement, two chromosomes as “parents”.

### 3.5.2.2 Crossover

This operator randomly chooses a locus and exchanges the sub-sequences before and after that locus between two chromosomes to create two offspring. It could be said that the main distinguishing feature of a GA is the use of crossover. Single-point crossover is the simplest form: a single crossover position is chosen at random and the parts of two parents after the crossover position are exchanged to form two offspring. The idea here is, of course, to recombine building blocks (schema) on different strings. Single-point crossover has some shortcomings, though. For one thing, it cannot combine all possible schema. Likewise, schema with long defining lengths are likely to be destroyed under single-point crossover. In this work, The crossover is a linear order that intends to preserve relative positions of the genes, and work on a sub-sequence level of each chromosome to preserve a good batch sub-sequence. We decide to switch some parts of two solutions (parents) to obtain two new solutions (children). Two sub-sequences of the same length are randomly chosen in each chromosome and then switched. This method cannot guarantee the legality of the new born solutions which may be discard if necessary.

Table 3.2 shows the details of a crossover between two solutions then the resulting children. After selecting two parents, a random value is generated to determine whether the crossover process has to be performed.

parent 1											
1	0	3	3	3	0	1	1	0	2	2	2
parent 2											
2	0	3	3	0	2	2	2	0	1	1	1
After the crossover, the two offsprings are obtained as follows:											
child 1											
1	0	3	3	3	0	3	3	0	2	2	2
child 2											
2	0	1	1	0	2	2	2	0	1	1	1

**Table 3.2:** The two parents crossover operator

### 3.5.2.3 Mutation

Here, we defined a mutation operators, the purpose of this kind of operators is to ensure the non-stagnation of the solutions in a local extreme by randomly modifying a chromosome without considering another one. In case of every chromosomes convergence through a unique solution, two randomly chosen chromosomes may be quite identical and the crossover operator will create two children similar to their parents.



the mutation operator is needed to explore other solutions and ensure the solutions legitimacy. A common view in the GA community, dating back to [Holland, 1975] book *Adaptation in Natural and Artificial Systems*, is that crossover is the major instrument of variation and innovation in GAs, with mutation insuring the population against permanent fixation at any particular locus and thus playing more of a background role. This differs from the traditional positions of other evolutionary computation methods, such as evolutionary programming and early versions of evolution strategies, in which random mutation is the only source of variation. However, the appreciation of the role of mutation is growing as the GA community attempts to understand how GAs solve complex problems. Some comparative studies have been performed on the power of mutation versus crossover; for example, [Spears et al., 1993] formally verified the intuitive idea that, while mutation and crossover have the same ability for "disruption" of existing schema, crossover is a more robust "constructor" of new schema.

In this study, we use three mutation operators which are defined as follows.

- A swap operator which works to swap two randomly selected couples of batches and their corresponding customers of one solution.
- A split mutation operator, which aims to divide a randomly selected batch of one solution into two separate batches with the same customer.
- A fusion operator, which works to merge two successive batches of one solution of a supply link into one new batch. The corresponding customer is chosen among the two customers of the initial batches.

### 3.5.2.4 New Population and stopping criterion

A complete cycle of the selection, crossover and mutation operations forms an iteration. Once the offspring is reproduced, it is stored in a temporary population. Until all the offspring have been reproduced, the "offspring population" is then combined with the latest population to form two new populations. The new chromosomes are compared according to their fitness values. The chromosomes with high total cost are eliminated and the remaining chromosomes form the next generation. The GA is then performed again until the stopping criterion is reached. The stopping criterion for the problem is either when number of iterations are completed.

## 3.6 Experimental results

In this section, a set of problems taken from the supplier data with different sizes are used for this study. The computational experiments are carried out to test the performance of the B&B, *LSS* algorithm, Genetic algorithm and the results of the MIP model.

The MIP model was solved with CPLEX solver and the proposed algorithms is coded in JAVA language. The computations were carried out on a desktop Intel core 2 processor operating at 2.67 GHz clock speed and 4 GB RAM. The model where run for at most one hour by instance or until the solver found the models optimal solutions or ran out of the memory, in this case the best optimal solution founded is collected as near optimal solution. The reference of time limit of resolution is based on the real time of the preparation of a scheduling in the actual case. As a comparison, CPLEX solver is used to exactly solve the model with small scale random instances. Some adjustments are done on the parameters of research in CPLEX, in the objective to accelerate the research of solutions.

### 3.6.1 Numerical example

To clarify the problem, we consider a simple numerical example in (Table 3.3) as follow. Two customers ordered 5 jobs at the same time (Monday at 8:00 am) and they would receive their products at the same time (Thursday at 8:00 am), that means all the products have the same due date equal to *72 hours*. The supplier and its customers open *24 hours per day*. Three jobs ( $j = 1, 3$  and  $5$ ) for customer 1 and two jobs ( $j = 2$  and  $4$ ) for customer 2. The vehicle capacity  $c = 2$ . The transporter delivery cost and time depend of the customer's positions with ( $\eta_1 = 1000$  Euro,  $\tau_1 = 6$  hours and  $\eta_2 = 750$  Euro,  $\tau_2 = 4$  hours) belong to customer 1 and 2 respectively, ( $\beta_1 = \beta_3 = \beta_5 = 30$  Euro/hour and  $\beta_2 = \beta_4 = 20$  Euro/hour) belong to customer 1 and 2 respectively.

Jobs	1	2	3	4
Customer's	1	2	1	2
Due date	Thursday 8:00 am	Thursday 8:00 am	Thursday 8:00 am	Thursday 8:00 am
Departure Time	Wednesday 06:00 am	Wednesday 06:00 pm	Thursday 2:00 am	Wednesday 06:00 pm
Arrival Time	Wednesday 12:00 pm	Wednesday 10:00 pm	Thursday 8:00 am	Wednesday 10:00 pm
Inventory Cost	600	200	0	200

**Table 3.3:** A numerical example

The solution is shown in (Table 3.3) for this problem. As it is shown, the vehicle makes three round trips among them two to customer 1 and one to customer 2. Three batches  $k_1 = 1$ ,  $k_2 = 2$  and  $k_3 = 2$  are denoted. The products arrive at the customer in the batch to which they belong in the completion time cited in (Table 3.3). The total delivery cost equals to  $\eta_1 \times 2 + \eta_2 \times 1 = 2750$  Euro and the total storage cost at the customers equals to  $\beta_1 \times [(d_1 - C_1) + (d_3 - C_3) + (d_5 - C_5)] + \beta_2 \times [(d_2 - C_2) + (d_4 - C_4)] = 30 \times [20 + 0 + 0] + 20 \times [10 + 10] = 1000$  Euro. The amount of the objective function is *3750 Euro*.

According to the confidentiality of the data base of the real-life data of the system under study, a several classes of problem have been considered for which several instances have been generated randomly.

### 3.6.2 Test cases

The characteristics of orders to schedule differ by customers, transporter capacity, quantity delivered, due date, transporter time, transporter cost and the storage cost at each customer. Three cases are considered to test the proposed methods. The characteristics of the case are listed in Table 3.6.2. For each case {A, B and C}, the number of products  $n$ , number of customers  $h$ , the transporter time, the transporter cost and the storage cost at each customer are displayed.

**Table 3.4:** Main characteristics of the test cases

Case	$n$	$h$	$\beta_h(\mathbf{Euro})$	$\eta_h(\mathbf{Euro})$	$\tau_h(\mathbf{Hours})$
A	10	2, 3, 4	[1; 5]	[1000; 1500]	[3; 5]
	20	2, 3, 4			
	30	2, 3, 4			
	40	2, 3, 4			
B	10	2, 3, 4	[10; 50]	[1000; 1500]	[3; 5]
	20	2, 3, 4			
	30	2, 3, 4			
	40	2, 3, 4			
C	10	2, 3, 4	[100; 500]	[1000; 1500]	[3; 5]
	20	2, 3, 4			
	30	2, 3, 4			
	40	2, 3, 4			

In the first class,  $\eta_h$  is higher than  $\beta_h$ , where  $\eta_h$  and  $\beta_h$  are randomly generated from the uniform distribution with ranges [1000, 1500] and [1, 5], respectively. In the second class,  $\eta_h$  are generated in the same way and with the same distribution with ranges [1000, 1500] and  $\beta_h$  of the first class are multiplied by 10, where  $\beta_h$  are randomly generated from the uniform distribution with ranges [10, 50]. In the third class,  $\beta_h$  is calculated by multiplying the ranges of the first class by 100, where  $\beta_h$  are randomly generated from the uniform distribution with ranges [100, 500] *Euro*. The results can be seen in (Table 3.5, Table 3.6, Table 3.7 and Table 3.8).

### 3.6.3 Comparison of the computational time of solutions

Table 3.5, Table 3.6 and Table 3.7 show the solution time obtained for each method. In the computational study, the following parameters are used: the vehicle's capacity is randomly generated from the uniform distribution with range  $[n/5, 2n/5]$ , further, its round-trip delivery time for each customer are randomly generated from the uniform distribution with range [3, 5] *hours*. The due dates  $(d_j)_{j=1..n}$  are uniformly separated with values randomly generated.

Moreover, for each case, the number of jobs set as 10, 20, 30 and 40, and the number of customers as 2, 3 and 4 for each case.

The parameters are generated with a magnitude order which is consistent with those of the supplier. For each combination, 25 problem instances are randomly generated and the average cpu time for each method of resolution are collected.

Class of Problem→		Class A			
		(h)→	2	3	4
		(n)↓	CpuT (s)	CpuT (s)	CpuT (s)
CPLEX	10	0.644	0.505	1.660	
B&B		0.006	0.005	0.019	
LSS		0.015	0.013	0.012	
GA		7.1	7.2	7.3	
CPLEX	20	53.67	45.31	43.25	
B&B		0.046	0.385	0.891	
LSS		0.026	0.031	0.041	
GA		7.5	7.6	8.1	
CPLEX	30	144.842	197.274	288.585	
B&B		0.221	1.749	20.934	
LSS		0.044	0.064	0.103	
GA		8.2	8.8	8.2	
CPLEX	40	420.495	429.103	467.098	
B&B		1.785	40.361	308.997	
LSS		0.069	0.121	0.193	
GA		9.4	10.3	8.6	

**Table 3.5:** Computational results for  $\beta_h \in [1, 5]$  *Euro*

The results show that heuristic algorithm runs much faster than the exact methods of resolution. For the exact methods, the B&B algorithm which is supported by the lower bound to accelerate the proposed algorithm runs faster than the CPLEX solver. The CPLEX solver finds the optimal solution, however its computational time grows exponentially as the instance size increases in the same class. Conversely, the resolution of the B&B algorithm is acceptable, which explains the efficiency of the lower bound used in the B&B algorithm to give the optimal solution for small to medium size instances. The computational time of the proposed *LSS* will never exceed 0.3 second. Moreover, the *LSS* can obtain optimal or near optimal solutions for all of the situations.

In Table 3.5, the results show that it was possible to solve all the instances with the three proposed methods. In this case, the total storage cost at the customer  $\sum_{j=1}^n \beta_{cl_j}(d_j - C_j)$ , which constitutes the second part from the objective function (Equation 3.1), will be less than those of the total transporter cost  $\sum_{h=1}^m \eta_h u_h$ , which

Class of Problem →	Class B			
	$(h) \rightarrow$	2	3	4
	$(n) \downarrow$	CpuT (s)	CpuT (s)	CpuT (s)
CPLEX	10	3.439	2.106	4.753
B&B		0.006	0.006	0.023
LSS		0.015	0.013	0.012
GA		7.2	7.5	8.0
CPLEX	20	462.669	515.522	474.654
B&B		0.149	1.133	6.161
LSS		0.025	0.031	0.042
GA		7.9	8.1	8.7
CPLEX	30	> 3600	>3600	>3600
B&B		1.471	63.433	>3600
LSS		0.045	0.061	0.101
GA		9.1	9.5	8.9
CPLEX	40	>3600	>3600	>3600
B&B		24.229	>3600	>3600
LSS		0.068	0.110	0.186
GA		9.3	9.8	10.7

**Table 3.6:** Computational results for  $\beta_h \in [10, 50]$  *Euro*

constitutes the first part from the objective function (Equation 3.1). In this case, the problem is very easy to solve, because the vehicle is fully loaded according to the cheapness of the storage cost at the customer.

In Table 3.6, the problem becomes harder to solve with CPLEX onset from 30 products on regardless of the number of customers. However, the time of resolution of the B&B algorithm exceeds the proposed time limit onset from 4 customers if the number of products equals to 30, and onset from 3 customers when the number of products equals to 40. In this case, the number of batches is increased and the number of products by batch is decreased.

In the third class of problem, in Table 3.7, the efficiency of CPLEX decreases, where the MIP model solves only the instances of 10 products with 2, 3 and 4 customers. However, the B&B solves the same instances as in Table 3.6, but its processing time grows progressively when the number of customers and products increase. In this case, the vehicle is very lightly loaded. Clearly, more the instances will be larger, more they will be harder to solve.

These results show that the proposed B&B algorithm is more efficient than the MIP model, and the *LSS* proposed algorithm gives an optimal or near optimal solutions for small to large size instances.

Class of Problem→	(h)→	Class C		
		2	3	4
	(n)↓	CpuT (s)	CpuT (s)	CpuT (s)
CPLEX	10	15.208	18.749	29.442
B&B		0.008	0.008	0.020
LSS		0.014	0.013	0.013
GA		7.0	6.9	7.6
CPLEX	20	>3600	>3600	>3600
B&B		0.301	2.882	32.497
LSS		0.026	0.031	0.036
GA		7.8	7.8	7.8
CPLEX	30	>3600	>3600	>3600
B&B		6.602	241.455	>3600
LSS		0.046	0.069	0.110
GA		8.2	8.5	8.9
CPLEX	40	>3600	>3600	>3600
B&B		78.240	>3600	>3600
LSS		0.074	0.135	0.213
GA		8.4	8.9	9.6

**Table 3.7:** Computational results for  $\beta_h \in [100, 500]$  Euro

### 3.6.4 Comparison of the quality of solutions

In this section, the performance of the proposed *LSS* heuristic algorithm and Genetic algorithm are analyzed thoroughly, by comparing these results with the performance of the proposed exact methods. In the first time, we will compare the results of the proposed heuristic algorithm with the results of the fast exact method found above which is the *B&B* algorithm. Then, we will study the performance of our GA which starts by *LSS* heuristic algorithm as an initial algorithm.

The three considered cases are found in (Table 3.8). For each case, three scenarios are considered beginning with two customers in use, then three and four. Moreover, for each case, the number of products sets as 10, 20, 30 and 40 respectively. In each case, the customer storage cost  $\beta_h$  is generated from a discrete uniform distribution in the interval  $[1, 5]$ ,  $[10, 50]$  and  $[100, 500]$  Euro for the three classes, respectively.

In the computational study, the following parameters are used: the vehicle's capacity is randomly generated from the uniform distribution with range  $[n/5, 2n/5]$ , further, its round-trip delivery time for each customer are randomly generated from the uniform distribution with range  $[3, 5]$  hours. The due dates  $(d_j)_{j=1..n}$  are uniformly separated with values randomly generated.

### 3.6.4.1 Heuristic algorithm performance

Considering the different parameters, 36 situations of the problem are tested. For each situation, 25 problem instances are generated to study the performance of the *LSS*. Based on the results of the exact methods, the error ratio is defined as  $ER(LSS/Exact) = (E_{LSS} - E_{Exact}) / E_{Exact}$ , where  $E_{LSS}$  denotes the average of the evaluation of the solution generated by the proposed *LSS*, and  $E_{Exact}$  denotes the average of the evaluation of the solution generated by exact methods. The results are displayed in Table 3.8.

Class of		The Error Ratio		
Problem↓		ER=( $E_{LSS} - E_{B\&B}$ )/ $E_{B\&B}$		
	( $h$ )→	3	4	5
	( $n$ )↓			
Case A	10	6.62 %	9.11 %	2.17%
	20	4.23%	7.13%	7.67%
	30	6.79%	9.52%	2.32%
	40	6.41%	8.80%	5.27%
Case B	10	8.10%	0.00%	1.15%
	20	11.05 %	8.62%	3.68%
	30	8.77%	8.19%	4.03%
	40	6.84%	12.43%	12.28%
Case C	10	7.70%	5.90%	2.18%
	20	6.18%	5.10%	2.89%
	30	3.57%	9.00%	12.19%
	40	8.59%	12.12%	10.19%
		Overall Average = 5.82%		

**Table 3.8:** The performance results for  $\beta_h \in [1, 5], [10, 50], [100, 500]$  *Euro* for three classes respectively

Table 3.8 shows clearly that the overall average equals to 5.82% which demonstrates that the proposed *LSS* is capable of generating near-optimal solutions within a reasonable amount of CPU time. One of the reasons may be the improvement phase which is presented in the approach method presented in section 4. As seen in each case, the average error ratios appear in an increasing trend as the value of  $n$  increases.

### 3.6.4.2 Genetic algorithm performance

In order to evaluate the performance of the proposed GA, we compare these results with the performance of the B&B algorithm. We used the same instances as above with three scenarios beginning with two customers in use, then three and four.

Considering the different parameters depicted in Table 3.6.2, we tested 36 situations of the problem. For each situation, we randomly generated 25 problem instances for the performance test of the GA.

Class of		The Error Ratio		
Problem↓		ER=(E <sub>GA</sub> -E <sub>B&amp;B</sub> )/E <sub>B&amp;B</sub>		
	(h)→	3	4	5
	(n)↓			
Case A	10	0.00%	0.00%	0.00%
	20	0.00%	0.00%	0.00%
	30	0.00%	0.00%	0.00%
	40	0.00%	0.00%	0.00%
Case B	10	0.00%	0.00%	0.00%
	20	0.00%	0.00%	0.00%
	30	0.00%	0.00%	0.00%
	40	0.00%	0.00%	0.00%
Case C	10	0.00%	0.00%	0.00%
	20	0.00%	0.00%	0.00%
	30	0.00%	0.00%	0.00%
	40	0.00%	0.00%	0.00%

**Table 3.9:** The performance results with  $\beta_h \in [1, 5], [10, 50], [100, 500]$  Euro for three classes respectively

After the comparison between the results of the proposed *LSS* algorithm as initial solution of the GA, we have compared the results found by the GA with the results of the *B&B* algorithm. We used the same method of evaluation to evaluate our developed GA, where we calculate the error ratio between the GA and the *B&B* algorithm. The error ratio between the solution found by *LSS* algorithm and the *B&B* algorithm is defined as  $ER=(E_{GA}-E_{B\&B})/E_{B\&B}$ , where  $E_{GA}$  denotes the average of the evaluation of the initial solution generated by the developed GA, and  $E_{B\&B}$  denotes the average of the evaluation of the proposed Branch and Bound.

According to Table 3.9 for the ER of the different cases (A, B and C), the results confirms the optimal solution obtained by the GA heuristic, within a reasonable amount of CPU time. Compared with the results found by *LSS* in Table 3.8, the GA still performs better than the *LSS*. GA is capable of obtaining an optimal solution compared to the *LSS* for the tested instances, and requires far less computational time compared to the proposed *B&B* algorithm. Therefore, the GA heuristic developed in this chapter is capable of validating the quality of the solutions obtained from other methods of resolution that attempt to solve this kind of supply chain optimization problems.



## 3.7 Conclusion

In this chapter we have studied the case of a single supplier sourcing a products at multiple heterogeneous customers. To reduce storage and transportation costs, the transporter serves every customer separately. Further, it is assumed that jobs are first processed in the production stage, and then gathered in batches and delivered to the customer to which it belongs. The resource of products is a number of available machines that produce according to a multi-customer command. Each job has a due date specified by the customers. Moreover, it was supposed that a job that arrives at the customer before its due date will incur a customer inventory cost. We proposed two methods of resolution and two heuristic algorithms. We have first proposed a MIP model and developed a *B&B* algorithm as exact methods. Then, we have developed a simple heuristic algorithm and showed that the relative error ratio of the heuristic solution is important and it needs to improvement. We have further developed a more sophisticated genetic algorithm with a better solutions more the first heuristic algorithm. Our computational study not only shows that the improved genetic algorithm is effective in practice but also shows that it can provide a substantial optimal results in a few time of resolution.



# 4 Single-Customer with Multi-Transporter Integrated Lot Sizing and Delivery Scheduling Problem

## 4.1 Introduction

In many make-to-order businesses, the production only takes place after customers place their orders. Under such a circumstance, coordinating production with transportation and inventory can become challenging, especially when the related facilities possess very limited resources. The coordination between the supplier and the customer for improving the performance of inventory control has received a great deal of attention and the integrated approach has been studied for years. Typically, the integrated approach focuses on the production-delivery-inventory decisions of supply chain partners while minimizing the total relevant cost of the system. The delivery method used plays a role in resolving the problem, such as the number of transporters used, the capacity of each transporter and the size of each batch. Due to the number of decision variables to be determined, the integrated production and distribution planning problem is so complex that optimal values are very hard to obtain.

In this chapter we investigate a variant of the lot sizing and batch scheduling problem with delivery time and cost dependent on batch size. This delivery and inventory scheduling problem based on finite number of heterogeneous or homogeneous vehicles. In particular, we address a model with more than one transporter, in which a number  $n$  of jobs are produced and delivered in un-equal sized batches. In the first time, the supplier received an order of  $n$  jobs. When the production of one batch at the supplier's end is finished, the products are loaded in a chosen transporter capacity. In order to do so, we have to assume that the supplier has enough production resources represented by a multi-machine, and that the delivery sequence of batches is the same as the production one. The round trip cost increases with the augmentation of the size of the batch and the capacity of the transporter. The setup time between any two batches is not considered because all the jobs are identical. The customer will demand a specific date for the arrival of each product. Therefore to satisfy this requirement and to minimize the related costs, the different

products have to be gathered in batches before being transported. The aggregation of products will lead to the need for storage, which would mean storage cost for both the supplier and the customer. The computation of the production/delivery date is processed by backward equation [Elmahi et al., 2006]. It is a method where lot-sizing and scheduling is done chronologically backwards. The objectives of the study in this chapter are:

- To study the behavior of each parameters of the problem under study.
- To find optimal cost.
- To find optimal batch size with the respect of the different constraints.
- To find optimal batch number related to each transporter.
- To find best policy to reduce total system cost.

As far as we know, the assumption of a batch-size-dependent delivery times and costs in order to take into account the loading and the unloading times of the transporters, in a single-supplier/single-customer with multiple vehicles configuration has not been studied till now. This assumption is one of the distinguishing features of the work presented in this chapter. The second distinguishing feature is to propose an exact resolution method based on a dominance property, which is very efficient in the identical transporters case and to compare the corresponding results to a mixed-integer-programming model solved by CPLEX and an approach method consists of an efficient genetic algorithm.

For ease of reference, we denote this problem: the Multi-Transporter Lot Sizing and Delivery Scheduling Problem (MT-LSDSP). The complexity of the (MT-LSDSP) is still an open question. To the best of our knowledge, no polynomial algorithm can solve this problem. However from simulation experiments, we observe that the problem is still intractable on an empirical basis. The given mathematical model is non linear as the delivery time of a batch depends on its size.

The rest of the chapter is organized as follows. In Section 2, the problem description is introduced, the notation used along this chapter are defined, then a definition of the studied problem is given and an illustrative example is given. The mathematical model is formulated and analyzed. In Section 3, mathematical analyses are established where the first one is a dominance relation in general case and the second one is a specific dominance relation for the identical transporters case. We then introduce in section 4 a dynamic programming approach and we formulate the problem as a mixed integer programming (MIP) model as an exact methods for solving the problem, and a genetic algorithm is developed as an approach method. Finally, in Sections 5 and 6, we provide the experimental results and draw some conclusions of this chapter.

## 4.2 Problem description

In this section the formulation of the operational model is presented to illustrate the assumptions production, delivery and inventory policies. This work is based on some notations are first defined, then the problem definition is represented, and an illustrative example is represented and a mathematical formulation is proposed.

### 4.2.1 Notations

#### *Parameters*

- $J = 1, 2, \dots, n$ : set of all jobs, where  $n$  is the total number of jobs,
- $H = 1, 2, \dots, m$ : set of all transporters, where  $m$  is the total number of transporters,
- $j$ : index for jobs,  $j \in J$ ,
- $h$ : index for transporters,  $h \in H$ ,
- $d_j$  : due date of the job  $j$ ,
- $c_h$  : capacity of the vehicle  $h$ ,
- $\tau_{h,b}$ : delivery time of the vehicle  $h$  to deliver a batch of size  $b$ . Denote that,  $\tau_{h,0}$  is the return time of the empty transporter  $h$  from the manufacturer to the subcontractor,
- $\eta_h$  : round-trip delivery cost of the transporter  $h$ ,
- $\alpha_h$  : sum of the subcontractor holding and delivery costs when using the transporter  $h$ . The  $\alpha_h$  functions are assumed to be non-decreasing functions,
- $\beta_j$  : manufacturer earliness penalty function for job  $j$ . The  $\beta_j$  functions are assumed to be non-increasing functions,

#### *Primary variables*

- $\sigma_k$ : size of production and delivery batch  $k$ ,
- $h_k$  : the transporter which is assigned to deliver the batch  $k$ ,

#### *Secondary variables*

- $K = 1, 2, \dots, u$ : set of all batches, where  $u$  is the total number of batches,
- $k$ : index for batches,  $k \in K$
- $u_h$ : number of delivered batches for transporter  $h$ ,  $u = \sum_{h=1}^m u_h$
- $C_j$  : the arrival time of job  $j$  at the manufacturer,
- $B_k$ : the arrival time of the batch  $k$  at the manufacturer,
- $\lceil k \rceil_\sigma = \sum_{k=1}^k \sigma_k$  ,  $\lfloor k \rfloor_\sigma = \lceil k \rceil_\sigma - \sigma_k + 1$  : index of the last and first jobs of batch  $k$ , respectively.

### 4.2.2 Problem definition

The studied model in this chapter aims to study an integrated problem for job scheduling and batching wherein different inventory holding, costs between the supplier and the customer are allowed. There are  $n$  identical jobs  $j$  to be processed on a facility of multi-machines available at the subcontractor. Each job  $j$  (numbered  $j = 1, 2, \dots, n$ ) has a due date  $d_j$  specified by the manufacturer. It is not allowed to any job to arrive after its due date, however if the job  $j$  arrives before its due date, it will incur an earliness penalty cost. The jobs are batch processed at the subcontractor with no setup time between any two batches. Moreover, each job needs a constant processing time, however we assume that the total production time of any batch is less than the minimum round-trip delivery time for the transporters. After the processing of a batch is completed, the products are delivered to the manufacturer in different batch sizes by an available vehicle. We define a delivery-batch as a group of jobs that are delivered together in one shipment. Each transporter  $h$  has a given capacity  $c_h$ , a round-trip delivery cost  $\eta_h$  and for each batch size  $b$ , a round trip delivery time  $\tau_{h,b}$  which includes the loading and unloading phase of the products, where the delivery cost and time dependent on batch sizes. It is supposed that each job  $j$  that arrives before its due date will incur an earliness penalty (or storage cost)  $\beta_j$ , which is a non-decreasing function, according to the assumption of [Baptiste, 2000].

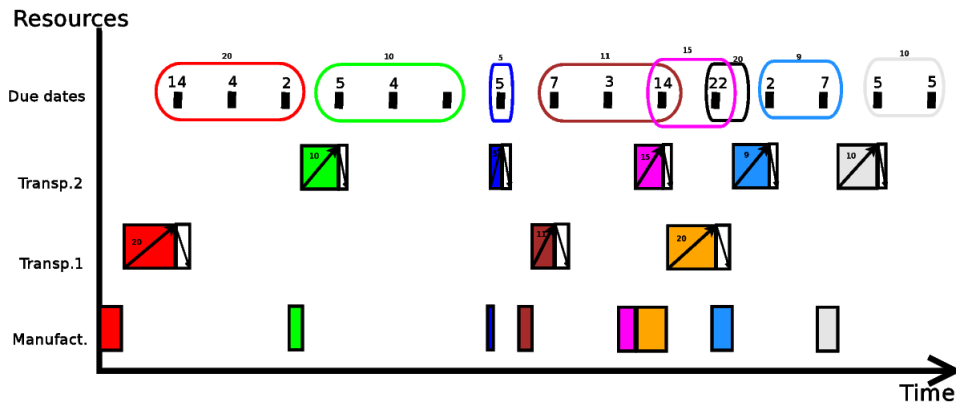
The sequence of the delivery batches that has to be processed and the schedule of the transporters that have to be used, have to satisfy the total manufacturer demand and need to minimize the global cost of the system. In the section an illustrative example is shown to explain in more details the considered system, then a mathematical formulation of the problem is proposed.

### 4.2.3 Illustrative Example

Here, an example is given to explain the considered problem with 100 jobs and two different transporters whose main characteristics are as follows. For the first transporter (resp. 2nd transporter), the capacity  $c_h$  is 20 (resp. 15) ; the delivery time  $\tau_{h,b}$  is given between 40 when the transporter is empty and 110 when it is fully loaded (resp. between 35 and 89) and the delivery cost  $\eta_h$  is 80 (resp. 60). We remark that the first transporter is more , but slower, than the second one.

In Figure 4.1, the gantt chart draws the activity of the different processes involved in the system, that is to say the manufacturing and delivery processes for the 2 transporters, depending on the time parameter. Moreover the due dates of the products have been represented at the top of the axe of the resources, in order to understand the relationship between the delivery dates of the different batches and their corresponding due dates. For the solution represented in Figure 4.1, 8 batches have been scheduled, 3 of which are delivered by the first transporter and the 5

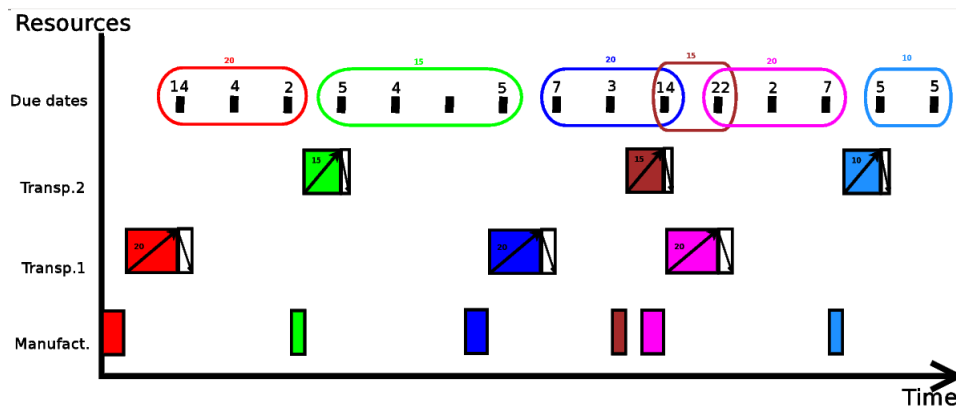
## 4.2 Problem description



**Figure 4.1:** The case of optimal solution (optimal cost = 1750.24)

others are transported with the second one. The size of the batches is respectively 20 (transp. 1), 10 and 5 (transp.2), 11 (transp.1), 15 (transp.2), 20 (transp.1), 9 and 10 (transp.2) with an optimal cost equal to 17520.24.

The second solution is given in Figure 4.2 and corresponds to a fully loaded solution. In this case, the transporters are fully loaded from the beginning of the horizon schedule until this end. The size of the batches is respectively 20 (transp. 1), 15 (transp.2), 20 (transp.1), 15 (transp.2), 20 (transp.1) and 10 (transp.2). We observe that the number of deliveries is reduced to 6, however the total cost of this solution is worst than those of the optimal solution. This can be explained by the fact that the earliness penalty of the jobs has increased drastically due to the fact that the batches have more jobs and have to be scheduled earlier in the planning.



**Figure 4.2:** The case of fully loaded solution ( Cost = 2158)

### 4.3 Mathematical Formulation and Dominance property

The considered problem consists of determining the sequence of batches, as well as specifying the transporter that has to deliver the parts from the subcontractor to the manufacturer.

The elements of the model's profit are the subcontractor production storage cost, the delivery cost of each batch and the earliness penalty at the manufacturer. Each element has its special internal constraints which are affected by the due dates imposed by the manufacturer. We assume that there are no waiting time between the end of the production of a batch and the starting date of its delivery, which is possible in a just-in-time context with enough production machines. Consequently, the subcontractor WIP holding cost will depend only on the batch size and can thus be included with the delivery costs of the corresponding delivery batch. These both costs are represented by a non-decreasing function  $\alpha_h$  that depends on the size of the batch. However products need to be delivered at the latest possible dates in order to reduce the manufacturer earliness penalties, which rely on the arrival time  $C_j$ , the due date  $d_j$ , and the earliness penalty function  $\beta_j$ .

The WIP holding time incurs an inventory cost that will depends on the batch size. The total WIP holding cost and delivery cost is represented in equation (Equation 4.1) as follows:

$$\sum_{h=1}^m \sum_{k=1}^u \alpha_h(\sigma_k) \tag{4.1}$$

On the other hand, the products have to arrive at the manufacturer at the latest possible dates with respect to the due date initially imposed. The holding cost of the customer will be formulated as follows:

$$\sum_{j=1}^n \beta_j(C_j) \tag{4.2}$$

Consequently, the objective function is the sum of the delivery cost (Equation 4.1) and the holding cost (Equation 4.2) defined in the following expression:

$$f(\sigma) = \sum_{h=1}^m \sum_{k=1}^u \alpha_h(\sigma_k) + \sum_{j=1}^n \beta_j(C_j) \tag{4.3}$$

subject to :



$$1 \leq \sigma_k \leq c_{h_k}, \quad k = 1, \dots, u \quad (4.4)$$

$$\sum_{k=1}^u \sigma_k = n, \quad k = 1, \dots, u \quad (4.5)$$

$$[k]_{\sigma} = \sum_{k=1}^k \sigma_k, \quad k = 1, \dots, u \quad (4.6)$$

$$C_j \leq d_j, \quad j = 1, \dots, n \quad (4.7)$$

$$C_j \leq C_{j+1}, \quad j = 1, \dots, n - 1 \quad (4.8)$$

$$C_j = C_{[k]} \quad k = 1, \dots, u, \quad j = [k], \dots, [k] \quad (4.9)$$

$$C_{[k_2]} - C_{[k_1]} \geq (\tau_{h_{k_2}, 0} + \tau_{h_{k_2}, \sigma_{k_2}}), \quad k_1, k_2 = 1, \dots, n; \quad k_1 < k_2 \text{ and } h_{k_1} = h_{k_2} \quad (4.10)$$

The objective function (Equation 4.3) minimizes the sum of delivery costs between the manufacturer and the subcontractor, and the storage cost on each side. Constraint (Equation 4.4) guarantees that the number of jobs scheduled in one delivery batch cannot exceed the capacity of the vehicle. Constraint (Equation 4.5) defines the total number of production jobs. Constraint (Equation 4.6) indicates the accumulation of jobs by each batch. Constraints (Equation 4.7) and (Equation 4.8) guarantee that each job must arrive at the manufacturer on time and by order. Constraint (Equation 4.9) defines that the arrival time of jobs for a given batch are identical. Constraint (Equation 4.10) maintains the time intervals between the arrival of two batches delivered by the same transporter to be greater or equal to the round-trip delivery time of the transporter for the second batch.

In the next subsection, we propose a general dominance relationship which will be used to accelerate the resolution of the problem. Then, the specific case when all the transporters share the same characteristics (capacity, delivery time and cost) is studied and an improved dominance relationship is highlighted between the solutions of the new problem.

### 4.3.1 General case

We define a partial solution where the number of delivered jobs is less than  $n$  jobs. A partial solution is not a complete solution, simply because its number of delivered

jobs is less than the total number of jobs requested by the manufacturer. The number of transporters used to deliver a sequence of batches for a partial solution could be less or more than that of a complete solution. In the same way, the number of the round trips done on a partial solution could be less or more than that of a complete solution.

We introduce now a general dominance relation among the partial solutions. We denote  $u^\sigma$  (resp.  $u^\omega$ ) the final number of batches that will be delivered for a partial solution  $\sigma$  (resp.  $\omega$ ).

**Theorem 1.** *Let  $\sigma$  and  $\omega$  be two partial solutions with  $\sigma \neq \omega$ .  $\sigma$  dominates  $\omega$  if the two sequences satisfy the following properties :*

$$\exists k_1, Q = \sum_{k=k_1}^{u^\sigma} \sigma_k = \sum_{k=k_1}^{u^\omega} \omega_k \quad (4.11)$$

$$\sum_{h=1}^m \sum_{k=k_1}^{u^\sigma} \alpha_h(\sigma_k) + \sum_{j=j_1}^n \beta_j(d_j - C_j^\sigma) \leq \sum_{h=1}^m \sum_{k=k_1}^{u^\omega} \alpha_h(\omega_k) + \sum_{j=j_1}^n \beta_j(d_j - C_j^\omega) \quad (4.12)$$

$$\forall k \geq k_1, B_k^\sigma - \tau_{h_k, \sigma_k} \geq \min \left( \min_{k' \geq k_1 / h_k^\sigma = h_{k'}^\omega} (B_{k'}^\omega - \tau_{h_{k'}, \omega_{k'}}), +\infty \right) \quad (4.13)$$

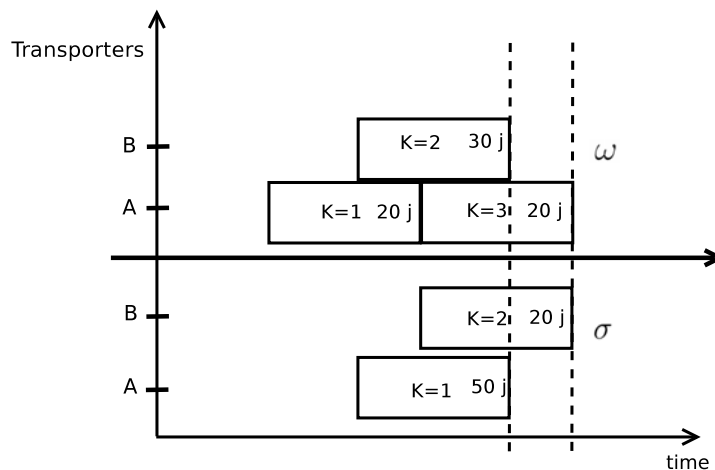
with  $j_1$  first job in the partial solution:  $j_1 = n - Q + 1$ .

*Proof.* Suppose that the complete solution  $\omega^*$  satisfies (Equation 4.11), (Equation 4.12) and (Equation 4.13), and that is not dominated by any solution that derives from  $\sigma$ . Now consider the complete solution  $\sigma'$  that derives from  $\sigma$  as follows: □

$$\begin{cases} \sigma'_k = \omega_k^*, & 1 \leq k < k_1 \\ \sigma'_k = \sigma_k, & k_1 \leq k \leq u^\sigma \end{cases} \quad (4.14)$$

For the new solution  $\sigma'$ , the number of new batches equals  $k_1 - 1$  which is the same as the batches that have been added to  $\omega^*$ . Moreover, the production holding cost for these batches will be the same for  $\sigma'$  and  $\omega^*$ . From this, the equation (Equation 4.12) can be rewritten as:

$$\sum_{h=1}^m \sum_{k=1}^{j_1-1} \alpha_h(\sigma'_k) \leq \sum_{h=1}^m \sum_{k=1}^{j_1-1} \alpha_h(\omega_k^*) \quad (4.15)$$



**Figure 4.3:** An example of the dominance relation

From the manufacturer point of view, we consider a batch  $\omega_k^*$  with  $k < k_1$ . This batch is the same as the batch  $\sigma'_k$ . The completion time of the job inside this batch will depend on the corresponding due dates, which are the same, and will depend on the next departure time of the same transporter  $h_k$ . The equation (Equation 4.13) shows that for a given transporter  $h$ , the departure date of this transporter for the partial solution  $\sigma$ , and consequently  $\sigma'$ , is always delayed than the earliest departure date of the same transporter for the partial solution  $\omega$ . Consequently, the arrival time of the jobs from the first batch to the batch  $k_1 - 1$  for the solution  $\sigma'$  are delayed than those of the solution  $\omega^*$ . It follows that :

$$\sum_{j=1}^{j_1-1} \beta_j (d_j - C_j^{\sigma'}) \leq \sum_{j=1}^{j_1-1} \beta_j (d_j - C_j^{\omega^*}) \quad (4.16)$$

The sum of the three expressions (Equation 4.12, Equation 4.15 and Equation 4.16) leads to:  $F(\sigma') \leq F(\omega^*)$ . Thus, the initial assumption is false, which concludes the proof.

This proposition means that the best batch sequence solution obtained from the partial solution  $\sigma$  is better than the best batch sequence solution obtained from the partial solution  $\omega$ .

To illustrate this proposition, let's consider the 2 partial solutions  $\sigma$  and  $\omega$  in the Figure 4.3 with 2 transporters  $A$  and  $B$ . The number of delivered jobs is 70 for the two partial solutions. If the total partial cost of  $\sigma$  is less than that of  $\omega$ , then  $\sigma$  dominates  $\omega$ , because for all batches, the minimum starting delivery time for transporters  $A$  and  $B$  is later for  $\sigma$  than for  $\omega$ .

**Lemma 2.** Consider 2 partial solutions  $\sigma$  and  $\omega$  with the same number of delivered parts, and the set of transporters that is affected to  $\sigma$  is a subset of the set of transporters that is affected to  $\omega$ .

$\sigma$  dominates  $\omega$  if the partial evaluation of  $\sigma$  is better than those of  $\omega$ , and for each transporter  $h$  used for  $\sigma$ , the first delivery starting date of  $h$  for  $\sigma$  is scheduled after the first delivery starting date of the same transporter  $h$  for  $\omega$ .

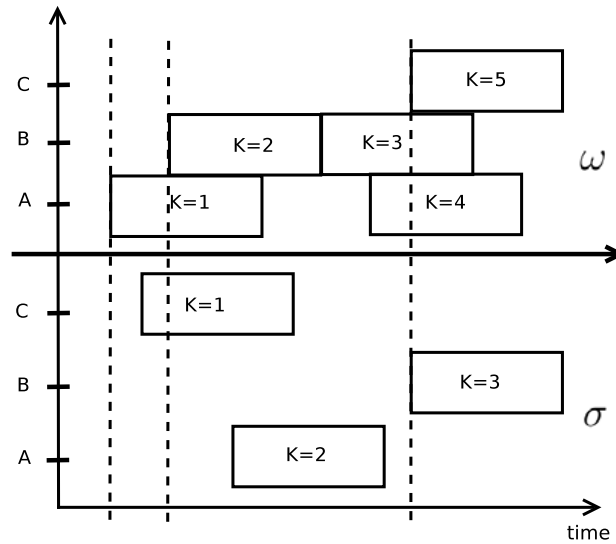
*Proof.* This is a direct consequence of theorem 1. □

In the theorem 1, we establish a dominance relationship for single supply link situation between two partial solutions with the same number of delivered jobs.

### 4.3.2 Identical transporters case

In the case of homogeneous transporters, every transporter can be exchanged with another one. Consequently, the dominance relationship of theorem 1 may be adapted to the identical transporters case, at the first glance, when two solutions have the same number of transporters used. More precisely, if a solution  $\omega$  has less transporter used than another solution  $\sigma$ , then no dominance can exist because the unused transporter of solution  $\omega$  could be planned as late as possible and it is not guaranteed that a transporter already used for solution  $\sigma$  could be planned in the same way. Consequently, the manufacturer holding cost for solution  $\sigma$  could be higher than those for solution  $\omega$ .

To introduce the dominance relationship in the case of a homogeneous fleet of transporters, we introduce  $\nu_i$  the index for the  $i^{th}$  transporter used in the considered solution, without taking into account the case where the same transporter is used more than once.



**Figure 4.4:** An example of the dominance relation in the identical transporters case

Let's take the example to explain the dominance relation with identical transporters. Consider a problem with 3 identical transporters A, B and C ( Figure 4.4), and two different solutions  $\sigma$  and  $\omega$ . For solution  $\sigma$ , the first transporter used is C, then A is used for the second delivery and finally B is used. In this case,  $\nu_1(\sigma) = C$ ,  $\nu_2(\sigma) = A$  and  $\nu_3(\sigma) = B$ . For the solution  $\omega$ , the first batch is delivered by A and the second batch is delivered by B, so we have  $\nu_1(\omega) = A$  and  $\nu_2(\omega) = B$ . The third batch is delivered by B but B has already been used for the second batch. We consider then the 4<sup>th</sup> batch that is delivered by transporter A which has already been used ; so the final batch is treated, which is delivered by C, and this gives :  $\nu_3(\omega) = C$ .

The delivery dates of the transporters for solution  $\sigma$  and  $\omega$  can be compared in the order given by the  $\nu$  function. We observe that the transporter  $\nu_1(\sigma)$  for  $\sigma$  starts its delivery after the first delivery of the transporter  $\nu_1(\omega)$  for  $\omega$ . In the same way, the earliest starting time of the transporter  $\nu_2(\sigma)$  (resp.  $\nu_3(\sigma)$ ) for  $\sigma$  is later than the earliest starting time of the transporter  $\nu_2(\omega)$  (resp.  $\nu_3(\omega)$ ) for  $\omega$ . If moreover, the current evaluation of solution  $\sigma$  is better than the current evaluation of solution  $\omega$ , then solution  $\sigma$  dominates solution  $\omega$ . The creation of the same new batch in the beginning of both solutions will incur a greater manufacturer holding cost for solution  $\omega$  than for solution  $\sigma$ , which mean the new solution of  $\sigma$  dominates that of  $\omega$ .

**Theorem 3.** *Let consider the homogeneous transporter case and let  $\sigma$  and  $\omega$  be two partial solutions with  $\sigma \neq \omega$ .  $\sigma$  dominates  $\omega$  if the two sequences satisfy the following constraint :*

$$\forall i \in [1, m] , \min_{k/h_k^\sigma = \nu_i^\sigma} (B_k^\sigma - \tau_{\nu_i^\sigma, \sigma_k}) \geq \min_{k/h_k^\omega = \nu_i^\omega} (B_k^\omega - \tau_{\nu_i^\omega, \omega_k}) \quad (4.17)$$

with the constraints (Equation 4.11) and (Equation 4.12) of theorem ( 1).

*Proof.* Based on the same principle used on the general dominance relation section 3.4, we will prove the dominance relation of the identical transporter case. If a sequence of batches is added for a given transporter  $h_1$  for the solution  $\omega$ , this sequence can be added to the transporter  $h_2$  for  $\sigma$  such as  $h_1$  (resp.  $h_2$ ) is the  $i^{th}$  transporter used for  $\omega$  (resp.  $\sigma$ ). This is equivalent to say, that:  $\exists i, \nu_i^\omega = h_1$  and  $\nu_i^\sigma = h_2$ . The overhead of this sequence of batches for  $\omega$  will be greater than that for  $\sigma$  because the corresponding transporter start later for solution  $\sigma$  than for solution  $\omega$ . This means that the finishing time  $C_j^\sigma$  for the jobs inside that sequence of batch will be greater than the corresponding  $C_j^\omega$  for  $\omega$ . Therefore, the storage cost of the sequence solution  $\sigma$  for the manufacturer will be less than that of the sequence solution  $\omega$ , which concludes the proof.  $\square$

The case where all the transporters share the same parameters is a restriction of the MT-LSDSP in the general case. However, the resolution of this particular problem

has an important application, as it can provide both a lower and an upper bound for the optimal value of a general problem. In fact, let consider  $P$  a MT-LSDSP with  $m$  transporters, if we consider the new problem  $P_{min}$  (resp.  $P_{max}$ ), which derives from  $P$ , and for which the capacities of the transporters is the maximum (resp. minimum) of the capacities of the transporters of  $P$ ; the delivery times of the transporters is the minimum (resp. maximum) of the delivery times of the transporters of  $P$  and the delivery costs of the transporters is the minimum (resp. maximum) of the delivery costs of the transporters of  $P$ , then the optimal solution obtained for  $P_{min}$  (resp.  $P_{max}$ ) is a lower bound (resp. upper bound) of the optimal solution of  $P$ .

In the next section, two solving methods that will be proposed to resolve the problem.

## 4.4 Solving methods

To solve the problem, we propose in this section a dynamic programming algorithm based on the dominance procedure to optimally solve the problem. In order, to validate the proposed model the general model will be linearized to built a mixed integer programming model.

### 4.4.1 Dynamic programming approach

Dynamic programming is a powerful optimization technique applicable to many complex problems requiring a sequence of interrelated decisions. Currently, dynamic optimization techniques have become an essential part of modern control theory, systems designs and operations, such as the design of the aircraft, the operation of a reservoir system and multi objectives routing problems see [Li and Haimes, 1989]. The main advantages of the dynamic programming formulation are, first, an optimal solutions found rather than near optimal solutions, second, the formulation permits general cost functions for each stage in the network.

**Algorithm 4.1** Description of the *GDP* algorithm

---

```

dom[0] ← ∅
for j ← 1 to n
  dom[j] ← ∅
  foreach transp
    foreach batch ≤ j and batch ≤ ctransp
      foreach sol ∈ dom[j − batch]
        insertBatchFirst (sol, batch, transp);
        dominant ← true
        foreach soldom ∈ dom[j]
          if (soldom dominates sol)
            dominant ← false;
          elseif (sol dominates soldom)
            remove soldom from dom[j];
          end foreach
        if (dominant) add sol to dom[j];
      end foreach
    end foreach
  end foreach
minTotalCost ← minsol ∈ som[n](cost(sol));

```

---

As a consequence, we propose a generalized dynamic programming scheme, named *GDP*, to build the optimal solutions starting from a single delivered part and ending with all the delivered parts (see algorithm Algorithm 4.1). Each level  $k$  will be composed of the dominant solutions for a number  $k$  of delivered parts. The dominated solutions will not be retained as they will lead to worse evaluations than the dominant ones. The process to build all the dominant solutions for level  $k$  consists on considering the retained solutions of all previous levels from level 1, to level  $k - 1$ . For each retained solution of level  $k' < k$ , a new solution of level  $k$  is built by simply adding a batch of  $k - k'$  parts for each transporter  $h$ . With this procedure, the number of possible solutions for level  $k$  will grow very quickly, as this is the sum of all the dominant solutions of the previous levels. To reduce the size of this set of solutions, the dominance criteria is applied among the population, in such a way that the dominated solutions of that level are removed. After reaching level  $n$ , the solution with the minimal evaluation is the optimal solution of the considered problem.

This procedure allows to find the optimal solution in the general case with heterogeneous transporters and a delivery time that is batch-size dependent. However, the experimental results show that the dominance relationship does not reduce efficiently the number of dominant solutions, and consequently, the processing time of the *GDP* grows very quickly when the number of jobs or the number of transporters increases.

### 4.4.2 The Mixed Integer Programming Model

We have assumed for the MT-LSDSP general model, that the delivery time of the transporters were batch-size dependent. Therefore, the corresponding mathematical model is not linear. In order to build a linear MIP model, this assumption has to be relaxed and the delivery times for a given transporter has to be considered as fixed, which of course produces a problem less complex to solve. By using a MIP model and by solving it with CPLEX, we will measure the performances of the proposed algorithms in the case where the delivery times of the transporters are constant.

In the following, we formulate our problem as a mixed-integer programming model. This model is used to determine resource deployment, given a logistics structure. In practice, such a tool is useful for evaluating and supporting strategic decision making. Furthermore, firms using such a model would most likely have an existing logistics structure, making variable-fixing an appropriate action. Before the model is presented, new parameters and variables, which are used in the MIP model, are described below.

#### *Parameters*

- $\tau_h$ : round trip delivery time of the vehicle  $h$  to deliver a batch,
- $\beta_j$  : the unit holding cost for job  $j$  in the manufacturer area
- $c_{max} = \max_{h=1..m} (c_h)$  : the maximum capacity of the transporters
- $\xi_j = E(\frac{j-1}{c_{max}}) + 1$  : the minimum batch number for job  $j$  taking into account the maximum capacity of the transporters.

#### *Primary variables*

- $y_k = 1$  if the batch  $k$  exists and is not empty, 0 otherwise,
- $\delta_{jk}^1 = 1$  if the job  $j$  belongs to the  $k^{th}$  batch, 0 otherwise,
- $\delta_{kh}^2 = 1$  if the batch  $k$  is delivered by the transporter  $h$ , 0 otherwise,

With the notations mentioned above, the mixed integer programming model is proposed as follows:

$$Min Z = \sum_{h=1}^m \eta_h u_h + \sum_{j=1}^n \beta_j (d_j - C_j) \tag{4.18}$$

Subject to:



$$\sum_{k=\xi_j}^j \delta_{jk}^1 = 1 \quad j = 1, \dots, n \quad (4.19)$$

$$\sum_{h=1}^m \delta_{kh}^2 = y_k \quad k = 1, \dots, n \quad (4.20)$$

$$y_k \geq y_{k+1} \quad k = 1, \dots, n-1 \quad (4.21)$$

$$u_h = \sum_{k=1}^n \delta_{kh}^2 \quad h = 1, \dots, m \quad (4.22)$$

$$\sum_{j=k}^n \delta_{jk}^1 \leq \sum_{h=1}^m c_h \times \delta_{kh}^2 \quad k = 1, \dots, n \quad (4.23)$$

$$C_j \leq d_j \quad j = 1, \dots, n \quad (4.24)$$

$$B_{k_1} \leq B_{k_2} \quad k_1, k_2 \in J^2; k_1 < k_2 \quad (4.25)$$

$$C_j = \sum_{k=\xi_j}^j B_k \times \delta_{jk}^1 \quad j = 1, \dots, n; \quad (4.26)$$

$$B_{k_2} - B_{k_1} \geq \sum_{h'=1}^m 2\tau_h \delta_{k_1 h'}^2 - (2 - (\delta_{k_1 h}^2 + \delta_{k_2 h}^2))M \quad k_1, k_2 \in J^2; k_1 < k_2; h \in H \quad (4.27)$$

$$\delta_{jk}^1 \leq \delta_{j+1,k}^1 + \delta_{j+1,k+1}^1 \quad j, k = 1, \dots, n-1; k \leq j \quad (4.28)$$

$$\delta_{jk}^1, \delta_{kh}^2 \in \{0, 1\} \quad j, k = 1, \dots, n; h = 1, \dots, m \quad (4.29)$$

The objective function (Equation 4.18) minimizes the sum of delivery and manufacturer inventory cost. Constraint (Equation 4.19) guarantees that, each job must be scheduled exactly in one batch, in this constraint the jobs will be batched only in the batch which it belongs. Constraint (Equation 4.21) assures that no empty batch allowed during the resolution of the problem. Constraint (Equation 4.20) forces every batch to be delivered using one transporter only. Constraint (Equation 4.22) calculates the number of batches delivered by each transporter. Constraint (Equation 4.23) guarantees that the number of jobs scheduled in one delivery batch cannot exceed the capacity of the vehicle to which it belongs. Constraint (Equation 4.24) and

(Equation 4.25) guarantees that each job must arrive at the manufacturer by order and on time. Constraint (Equation 4.26) represents the conversion of the completion time of the products belong to the same batch  $C_j$  to  $B_k$  the arrival time of each batch to the manufacturer. This constraint represented in a non-linear way in this mathematical representation to facilitate the understanding of the problem. Constraints (Equation 4.27) prevent the overlap between two consecutive batches in the case of two consecutive jobs delivered separately by the same transporter. Constraint (Equation 4.28) indicates that two consecutive jobs  $j$  and  $j + 1$  will either be in the same batch or be in consecutive batches. Constraints (Equation 4.29) define the range of the variables.

The performance of the both exact methods will be compared to a developed genetic algorithm, introduced and represented in the next section.

### 4.4.3 Genetic Algorithm

#### 4.4.3.1 Introduction

The exact methods are a methods these examine the search space completely with an add proposed by the developer as the dominance relation proposed in our work. In addition, at each search step only the solution concerned with the dominance relation are allowed to minimize the search space. This may cause some concerns for the time of resolution of the large instances tested. Genetic algorithm (GA) is a general, robust, and well developed optimization method. As a global optimization method, genetic algorithm has been used successfully to find optimal or near-optimal solutions for a wide variety of optimization problems, Genetic Algorithms (GAs) was invented by [Holland, 1975] who developed this idea in his book "Adaptation in natural and artificial systems". Holland proposed GA as a heuristic method based on "Survival of the fittest". A flowchart of a simple GA is illustrated in Figure 4.5. In a genetic algorithm, a population of strings (called chromosomes or the genotype of the genome), which encode candidate solutions (called individuals, creatures, or phenotypes) to an optimization problem, evolve; towards better solutions. The evolution usually starts from a population of randomly generated individuals and happens in generations. In each generation, the fitness of every individual in the population is evaluated, multiple individuals are stochastically selected from the current population (based on their fitness), and modified (recombined and possibly randomly mutated) to form a new population. The new population is then used in the next iteration of the algorithm. Commonly, the algorithm terminates when either a maximum number of generations have been produced, or a satisfactory fitness level has been reached for the population. If the algorithm has been terminated due to a maximum number of generations, a satisfactory solution may or may not have been reached [Jat and Yang, 2009]. [Wang, 2012] used the (GA) to reduce the problem of the Integrated Scheduling of Production Distribution-Inventory problems (ISPDI).

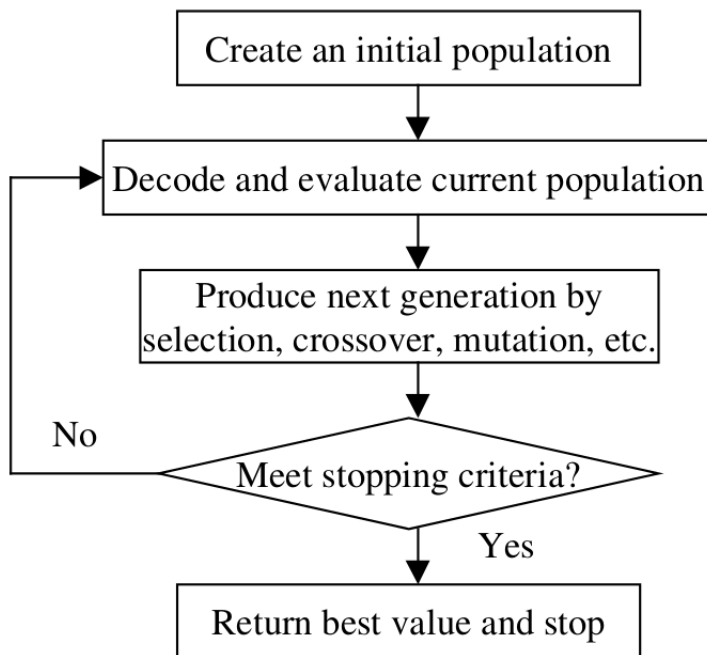


Figure 4.5: A simple GA flowchart

#### 4.4.3.2 Gene Representation

A GA is applied to the well connection optimization problem in the following manner. In GA, a set of decision variables is usually, though not necessarily, encoded as a binary string, with each single variable corresponding to a segment of the binary string. In GA vocabulary, the binary string is called a chromosome, and the segment of the binary string representing a variable is named a gene. With such an encoding scheme, decision variables take discrete values. Based on the characteristics of the scheduling problem under study, we adopt the real number coding method; which is the most commonly used coding method. The chromosome is composed of different genes representing the batches of the solution, every gene that is associated with a batch model contains two important information: the batch size and the corresponding transporter. For example, the chromosome  $((2, t_1), (3, t_2), (4, t_1), (1, t_1))$  corresponds to a solution of 4 batches: first, 2 jobs are delivered using the transporter  $t_1$ , then 3 jobs are delivered with the transporter  $t_2$ , and so on. For instance, the previous example may be represent by the chromosome of Table 4.1.

1	1	0	2	2	2	0	1	1	1	1	0	1
---	---	---	---	---	---	---	---	---	---	---	---	---

Table 4.1: A chromosome representing a solution

### 4.4.3.3 Evolving the population

A GA evolves a population by means of selection, crossover, and mutation, etc. When the evolution terminates, the chromosome with the best fitness represents the optimal solution of the optimization problem. This is the major part of a GA, and is an iterative procedure with each iteration containing the following steps:

- **Evaluation the population**

Decode each chromosome and evaluate its fitness (function value). For each chromosome, a rate allocation problem is solved and the optimal objective function value is assigned as the fitness of the corresponding chromosome.

- **Genetic Operators**

The current generation of the population is modified through selection, crossover, mutation, and other operations to produce the next generation of population.

1. **Selection**

In the selection steps, we replicate chromosomes for mating. This study implements a tournament selection, in which a mating chromosome randomly picked two chromosomes from the current population, and the one with higher fitness value is selected as the mating chromosome. Repeat this procedure until  $n$  mating chromosomes are selected.

2. **Crossover**

In the crossover step, we introduce two-points crossover operator based on the characteristics of the scheduling problem. This crossover is a linear order crossover that intends to preserve relative positions of the genes, and works on a sub-sequence level of each chromosome. The function of this crossover is to preserve a good batch sub-sequence. For this two-point crossover operator, we have introduced an appropriate linear coding in order to represent every batch delivered by a transporter  $t$  with a sequence of genes whose length is the corresponding batch size and of value  $t$ . Moreover two consecutive batches are separated with a gene whose value is '0'. For example in (Table 4.2), we consider a problem with 10 products and 3 transporters. Parent1 equals  $(2, t_1), (3, t_2), (4, t_1), (1, t_1)$ , therefore its coding begins with 2 (the size of batch 1) genes whose value is 1 (transporter  $t_1$ ), then continue with a '0' and a sequence of 3 genes whose value is 2 (3 jobs delivered by  $t_2$  for batch 2), and so on.

For this crossover operator, two sub-sequences of same length are determined randomly for both parents. The children are then created by exchanging these two sub-sequences except the '0' values; these are not exchanged in order to keep the legality of the solutions. For example, the two-point crossover of *parent1* and *parent2* will generate two children (Table 4.2).

3. **Mutation**

parent 1													
1	1	0	2	2	2	0	1	1	1	1	0	1	
parent 2													
3	3	3	0	2	2	0	3	3	0	1	1	0	2
child 1													
1	1	0	3	3	2	0	1	1	1	1	0	1	
child 2													
3	3	3	0	2	2	0	2	2	0	1	1	0	2

**Table 4.2:** The crossover operator

Mutation is used to produce small perturbations on chromosomes to promote diversity of the population. There are several mutation operators such as swapping, inversion, insertion and shift mutation (see [Cheng and Gen, 1997]). In this study, three mutation operators are defined as follows :

- A swap operator which works to swap two randomly selected couples of batches and their corresponding transporters of one solution.
- A split mutation operator, which aims to divide a randomly selected batch of one solution into two separate batches with the same transporter.
- A fusion operator, which works to merge two successive batches of one solution of a supply link into one new batch. The corresponding transporter is chosen among the two transporters of the initial batches.

#### 4. Elitism

In this study, the "roulette wheel of reserving elites" method is adopted. This selection scheme allows chromosomes to be selected for the next generation through a probability that is directly proportionate to their fitness. A maximum of two chromosomes are chosen randomly based on these scheme [Goldberg, 1989].

We provide in the next section the experimental results obtained with the dynamic programming, the mixed integer programming model and the genetic algorithm and we analyze them.

## 4.5 Experimental and Computational results

In this section, we evaluate the computational performance of the three techniques of resolution used to solve the problem under study: the proposed genetic algorithm, the proposed dynamic programming procedures and the mixed integer programming model solved by CPLEX, through 1650 test cases randomly generated, with respect to various parameter settings. Two different cases proposed for the evaluation of the problem, where the first one is the general case with heterogeneous transporter and the second one is the case with identical transporters. The performance of

GA was measured by the average error gap compared to the fast exact method (which is the developed *GDP* in this study), and was defined as  $ER(GA/GDP) = 100 * (GA - GDP) / GDP$  where *GA* denotes the best evaluation found by the genetic algorithm and *GDP* stands for the optimal value found by the dynamic programming method. The performance of the proposed *GDP* procedure was measured by its CPU time needed to find the optimal solutions and was compared with that by the CPLEX solver that solves the MIP model directly. Both the *GDP* procedure and the GA were programmed in JAVA language and implemented through a desktop Intel core 2 processor operating at 2.67 GHz clock speed and 4 GB RAM. The maximum solving time allowed for this instances is one hour. The MIP model is solved by CPLEX on the same machine.

### 4.5.1 Instances with Heterogeneous Transporters

Three classes of problems are considered so that in the first class the total transporter cost is bigger than the total manufacturer storage cost, this case is denoted as A in (Table 4.3). In the second class which is denoted as B in (Table 4.4), the total transporter cost is almost equal to the total customer storage cost. In the third class which is denoted as C in (Table 4.5), the storage cost is less than that of the total transporter cost. For each class, three scenarios are considered, beginning with two transporters in use, then three transporters and finally four transporters. Moreover, for each case, the number of jobs equals 10, 20 and 30, respectively. For each scenario, the vehicles's capacity are randomly generated from the uniform distribution with range  $[n/3, 2n/3]$ , further, its round-trip delivery time of each transporter is randomly generated from the uniform distribution with range  $[20, 100]$ . The due dates  $(d_j)_{j=1..n}$  are uniformly distributed over an horizon  $[0, 50.n]$ . The transporters cost  $\eta_h$  are randomly generated from the uniform distribution with ranges  $[100, 250]$ . The storage cost at the customer's varies in every case, where  $\beta$  is randomly generated from the uniform distribution with ranges  $[0.01, 0.05]$ ,  $[1, 5]$  and  $[10, 50]$  in the three different cases A, B and C, respectively.

The results of the CPLEX solver and the *GDP* procedure are shown in (Table 4.3, Table 4.4 and Table 4.5). The results show that the *GDP* runs much faster than the solution proposed by the MIP model. The CPLEX solver which used to solve the MIP model finds the optimal solution, however its computational time grows exponentially as the instance size increases, regardless of the parameters of the studied problem. In contrast, our proposed *GDP* is influenced by the value of the parameters used and the increase of the complexity of the problem. With a small and medium instances, the computational time of the proposed *GDP* will never exceed one hour. Moreover, the *GDP* can obtain the optimal solutions for all the situations.

For the problem of the class A in Table 4.3, we observe that *GDP* which is supported by the dominance relation runs faster than the CPLEX solver. The CPLEX solver

Class of		CPLEX	<i>GDP</i>	CPLEX	<i>GDP</i>	CPLEX	<i>GDP</i>
Problem↓		CpuT (s)	CpuT (s)	CpuT (s)	CpuT (s)	CpuT (s)	CpuT (s)
	Size( $n$ )→	10		20		30	
	vehicles( $m$ )↓						
Case A	2	0.105	0.026	2.009	0.042	9.427	0.075
	3	0.219	0.053	4.064	0.830	54.642	0.206
	4	0.131	0.089	5.173	0.382	95.208	1.740

**Table 4.3:** Results of instances with heterogeneous transporters with  $\beta = [0.01, 0.05]$

finds the optimal solution but its computational time grows rapidly as the instance size and the number of vehicles increases. Conversely, the computational time of the *GDP* is very short, which explains the efficiency of the dominance relation used in the *GDP* method to give the optimal solution from small to medium-sized instances. In this case, the two methods solve the problem rapidly for the parameters used. In these experiments, the optimal solution corresponds to fully loaded vehicles. In this case, the total holding cost at the customer is less than those of the total transporters cost. Consequently, this configuration is the least complex to solve, because the vehicles have to be fully loaded in order to minimize the delivery cost.

Class of		CPLEX	<i>GDP</i>	CPLEX	<i>GDP</i>	CPLEX	<i>GDP</i>
Problem↓		CpuT (s)	CpuT (s)	CpuT (s)	CpuT (s)	CpuT (s)	CpuT (s)
	Size( $n$ )→	10		20		30	
	vehicles( $m$ )↓						
Case B	2	0.756	0.027	106.745	0.086	>3600	0.115
	3	1.098	0.035	402.299	0.11	>3600	8.341
	4	2.417	0.114	>3600	7.204	>3600	100.047

**Table 4.4:** Results of instances with heterogeneous transporters with  $\beta = [1, 5]$

In the second class of the problem B in Table 4.4, the problem becomes harder to solve with CPLEX onset from 30 products regardless of the number of transporters. Our *GDP* runs faster than the CPLEX solver but the gap between the two methods becomes significantly prominent as the number of products and transporters increases. In this case, the time of resolution of the CPLEX solver starts to increase rapidly according to the variation of  $\beta$  in  $[1, 5]$ . In the optimal solution, we noticed that the number of batches is increased and the number of products by batch is decreased gradually.

In the third class of the problem C in Table 4.5, we observe that the *GDP* runs much faster than the CPLEX solver when the number of products is more than 10 products. Interestingly, The efficiency of CPLEX decreases enormously, where the MIP model solves only the instances of 10 products with 2, 3 and 4 vehicles. In this

Class of		CPLEX	<i>GDP</i>	CPLEX	<i>GDP</i>	CPLEX	<i>GDP</i>
Problem↓		CpuT (s)	CpuT (s)	CpuT (s)	CpuT (s)	CpuT (s)	CpuT (s)
	Size( $n$ )→	10		20		30	
	vehicles( $m$ )↓						
Case C	2	2.511	0.036	>3600	0.133	>3600	0.310
	3	12.080	0.138	>3600	6.210	>3600	8.517
	4	27.466	2.083	>3600	95.570	>3600	1300.24

**Table 4.5:** Results of instances with heterogeneous transporters with  $\beta = [10, 50]$

case, the computational time of the two methods becomes very large so that the variation of  $\beta$  equals to  $[10, 50]$ , where in the optimal solution the vehicles are very lightly loaded, but our proposed *GDP* is still more efficient than the results of the MIP model. These results show the efficiency of the proposed *GDP* method to give the optimal solution from small to medium-sized instances.

Now, we analyzed the performances of the dynamic programming and the genetic algorithm for the heterogeneous transporters instances. The aim of these experiments is to show the efficiency of the GA compared to the *GDP*. Several classes of problem have been considered for which several instances have been generated with different number of jobs and transporters. Three classes (1, 2 and 3) of problems are considered and the results are denoted in Table 4.6. For each class, three scenarios are considered, beginning with two transporters in use, then three transporters and finally four transporters. Moreover, for each case, the number of jobs equals 10, 20 and 30, respectively. For each scenario, the vehicles's capacity are randomly generated from the uniform distribution with range  $[n/3, 2n/3]$ , further, its round-trip delivery time of each transporter is randomly generated from the uniform distribution with range  $[20, 100]$ . The due dates  $(d_j)_{j=1..n}$  are uniformly distributed over an horizon  $[0, 50.n]$ . The transporters cost  $\eta_h$  are randomly generated from the uniform distribution with ranges  $[100, 250]$ . The storage cost at the customer's varies in every case, where  $\beta$  is randomly generated from the uniform distribution with ranges  $[0.01, 0.05]$ ,  $[1, 5]$  and  $[10, 50]$  in the three different cases 1, 2 and 3, respectively.

The parameters of the GA are summarized as follows:

- Population size : 500
- Termination condition: 500 iterations or fitness of the best individual did not change for a certain number of generations
- Mutation probability : 0.1
- Crossover probability : 0.6

Table 4.6 report on the performance of solution approaches of both method of resolution, the proposed *GDP* algorithm and the GA, under the impact of three parameters:  $n, m, \beta$ . As we can see, the GA demonstrated a promising performance with the largest average error ratio less than 5%, in the first and second cases,



Class of		GA	<i>GDP</i>	ER	GA	<i>GDP</i>	ER	GA	<i>GDP</i>	ER
Problem↓		avg. value			avg. value			avg. value		
	( <i>n</i> )→	10			20			30		
	( <i>m</i> )↓									
Case 1	2	710.1	710.1	0%	843.5	843.5	0%	726.8	726.8	0%
	3	627.4	627.3	0%	733.0	733.1	0%	784.5	784.5	0%
	4	630.4	630.3	0%	653.0	653.1	0%	612.6	612.6	0%
	( <i>n</i> )→	10			20			30		
	( <i>m</i> )↓									
Case 2	2	1686.6	1686.5	0%	2432.9	2408.6	1.0%	3241.1	3176.9	2.0%
	3	1402.0	1402.0	0%	2557.8	2513.4	1.7%	3060.4	2977.2	2.8%
	4	1120.9	1120.9	0%	2088.6	2062.5	1.3%	3365.5	3219.7	4.5%
	( <i>n</i> )→	10			20			30		
	( <i>m</i> )↓									
Case 3	2	6448.8	6428.3	0.3%	11077.6	10721.6	3.3%	17679.1	16691.3	5.9%
	3	3298.6	3217.1	2.5%	7011.2	5995.1	16.9%	15796.4	13354.3	18.3%
	4	2478.2	2289.6	8.2%	6737.8	5650.6	19.2%	14235.6	12367.4	15.1%

**Table 4.6:** Comparison of the quality of the solutions found by the GA and the *GDP*

where  $\beta$  equal to  $[0.01, 0.05]$  and  $[1, 5]$ , respectively. In the third case, the average error ration increase significantly and consistently as the number of products and the number of transporters increase. This is because the problem becomes more complicated to solve, if the number of products by batch is more restrictive.

## 4.5.2 Instances with Homogeneous Transporters

The performance of the dominance relation in the identical transporters case has been evaluated thoroughly. Here all the transporters are identical in capacity, round trip time and cost.

Three cases are considered, similar to what was done in the heterogeneous transporters, these three cases (A, B and C) are found in (Table 4.7), respectively. For each case, three scenarios are considered beginning with two transporters in use, then three and four. Moreover, for each case, the number of jobs is 10, 20 and 30 respectively.

For each scenario, the vehicles's capacity are randomly generated from the uniform distribution with range  $[n/3, 2n/3]$ , further, its round-trip delivery time of each transporter is randomly generated from the uniform distribution with range  $[20, 100]$ . The due dates  $(d_j)_{j=1..n}$  are uniformly separated with values randomly generated

from the uniform distribution with range  $[0, 50]$ . The transporters costs  $\eta_h$  are randomly generated from the uniform distribution with range  $[100, 250]$ . The storage cost at the manufacturer varies in every case, where  $\beta$  is randomly generated from the uniform distribution with ranges  $[0.01, 0.05]$ ,  $[1, 5]$  and  $[10, 50]$  in the three different cases A, B and C, respectively. For each combination, 25 problem instances are randomly generated and the average Cpu time for both methods of resolution CPLEX solver and the *GDPid* is calculated and denoted in (Table 4.7).

The results of the MIP model is compared with that of the dynamic programming (*GDPid*) with the dominance relation of the identical transporters (theorem 3). The results are shown in (Table 4.7).

Class of Problem↓		CPLEX	<i>GDPid</i>	CPLEX	<i>GDPid</i>	CPLEX	<i>GDPid</i>
		CpuT (s)	CpuT (s)	CpuT (s)	CpuT (s)	CpuT (s)	CpuT (s)
Case A	Size( $n$ )→	10		20		30	
	vehicles( $m$ )↓						
	2	0.093	0.021	3.219	0.041	46.209	0.095
	3	0.198	0.017	5.264	0.06	48.295	0.134
	4	0.182	0.017	11.807	0.056	49.231	0.105
Case B	Size( $n$ )→	10		20		30	
	vehicles( $m$ )↓						
	2	0.146	0.019	80.932	0.047	122.657	0.097
	3	0.807	0.016	135.759	0.051	92.750	0.228
	4	0.977	0.015	>3600	0.078	61.618	0.468
Case C	Size( $n$ )→	10		20		30	
	vehicles( $m$ )↓						
	2	2.050	0.018	>3600	0.061	>3600	0.188
	3	23.643	0.025	>3600	0.224	>3600	3.153
	4	73.338	0.029	>3600	0.794	>3600	35.203

**Table 4.7:** Results of small-medium instances with identical transporters

Concerning the computational time of resolution, the proposed *GDPid* is often much more efficient than the proposed MIP model. Especially, in the complex case when the number of transporters are more than three and the number of products become large. For instance in the third case of problem in (Table 4.7) when  $\beta = [10, 50]$  (case C) and the number of products equal to 30 with 4 transporters, the time of resolution of our *GDPid* is still efficient and it does not exceed one minute, while CPLEX needs more than one hour.

Compared with the results found in (Table 4.3, Table 4.4 and Table 4.5), the proposed *GDPid* is faster than that of the *GDP* and its time of resolution is not expelled in the same pace according to the increase of the number of products or transporters.

After the comparison of the Computational Results of the *GDPid* algorithm and the MIP model, we analyzed the performances of the dynamic programming with the identical transporters case and the genetic algorithm. The aim of these experiments is to show the efficiency of the GA compared to the *GDPid*. Several classes of problem have been considered for which several instances have been generated with different number of jobs and transporters. Three classes (1, 2 and 3) of problems are considered and the results are denoted in Table 4.8. For each class, three scenarios are considered, beginning with two transporters in use, then three transporters and finally four transporters. Moreover, for each case, the number of jobs equals 10, 20 and 30, respectively. For each instance, the vehicles's capacity are the same and it is randomly generated from the uniform distribution with range  $[n/3, 2n/3]$ . Further, the round-trip delivery time of each transporter is the same in each instance and it is randomly generated from the uniform distribution with range  $[20, 100]$ . The due dates  $(d_j)_{j=1..n}$  are uniformly distributed over an horizon  $[0, 50.n]$ . The transporters cost  $\eta_h$  is the same for all the transporters in the same instances, and it is randomly generated from the uniform distribution with ranges  $[100, 250]$ . The storage cost at the customer's  $\beta$  is randomly generated from the uniform distribution with ranges  $[0.01, 0.05]$ ,  $[1, 5]$  and  $[10, 50]$  in the three different cases 1, 2 and 3, respectively.

The parameters of the GA are summarized as follows:

- Population size : 500
- Termination condition: 500 iterations or fitness of the best individual did not change for a certain number of generations
- Mutation probability : 0.1
- Crossover probability : 0.6

Table 4.8 report on the performance of solution approaches of both method of resolution, the proposed *GDPid* algorithm and the GA. As we can see, the GA demonstrated a promising performance with the largest average error ratio less than 1.5%, in the first and second cases, where  $\beta$  equal to  $[0.01, 0.05]$  and  $[1, 5]$ , respectively. In the third case, the average error ratio increase significantly and consistently as the number of products and the number of transporters increase, but it does not exceed 15.11%. In the third case, the problem becomes more complicated to solve, where the number of products by batch is more restrictive. Compared with the quality of results in the heterogeneous transporters cases, the GA demonstrate an efficient performance with a maximum average error ratio less than to 15.11%.

### 4.5.3 Large instances with Homogeneous Transporters

To test the performance of the *GDPid* with a large instance, we conducted experiments using random instances with large problem sizes. We decided to study the case B considered in (Table 4.7), with 25, 50, 75 and 100 products, it is called case  $B'$ . Three scenarios are considered, beginning with two transporters in use, then

Class of		GA	GDPid	ER	GA	GDPid	ER	GA	GDPid	ER
Problem↓		avg. value			avg. value			avg. value		
	$(n) \rightarrow$	10			20			30		
	$(m) \downarrow$									
Case 1	2	1019.1	1019.1	0%	1030.2	1030.2	0%	1260.8	1260.8	0%
	3	1163.0	1163.0	0%	864.8	864.8	0%	1083.2	1083.2	0%
	4	773.6	773.6	0%	758.3	758.3	0%	795.2	795.2	0%
	$(n) \rightarrow$	10			20			30		
	$(m) \downarrow$									
Case 2	2	1427.1	1427.1	0%	3216.6	3186.9	0.93%	4781.3	4737.2	0.93%
	3	1545.7	1545.7	0%	2378.0	2361.9	0.68%	3644.0	3617.4	0.73%
	4	1327.6	1327.6	0%	2984.5	2963.8	0.70%	3209.5	3163.0	1.47%
	$(n) \rightarrow$	10			20			30		
	$(m) \downarrow$									
Case 3	2	9946.3	9946.3	0.00%	17830.2	17521.9	1.76%	26939.1	24980.6	7.84%
	3	3320.9	3320.9	0.00%	8367.7	7951.1	5.24%	16951.7	14806.2	14.49%
	4	3332.5	3299.9	0.99%	7081.5	6421.8	10.27%	14235.6	12367.4	15.11%

**Table 4.8:** Comparison of the quality of the solutions found by the GA and the *GDPid*

three and finally four. In (Table 4.9), only the performance of the *GDPid* is studied. the results show that, *GDPid* is able to solve the problem of this large instance with a reasonable time despite the increase in the number of products or transporters. However, the resolution time of the problem increases significantly according to the increase in the number of products, and the number of transporters.

Class of		<i>GDPid</i>	<i>GDPid</i>	<i>GDPid</i>	<i>GDPid</i>
Problem↓		CpuT (s)	CpuT (s)	CpuT (s)	CpuT (s)
	Size( $n$ )→	25	50	75	100
	vehicles( $m$ )↓				
Case $B'$	2	0.212	1.157	14.992	24.733
	3	0.514	28.727	435.240	1152.733
	4	0.738	207.290	>3600	>3600

**Table 4.9:** Results of large instances with identical transporters

## 4.6 Conclusion

In this chapter we investigate an Integrated Batch Production and Distribution Problem. In particular, we address a Multi-Transporter Lot Sizing and Delivery Scheduling Problem, where in a supply chain environment involving a fleet of heterogeneous or homogeneous transporters with batch-size-dependent delivery time and cost. The objective is to find a joint schedule in order to minimize the total delivery and inventory cost.

The problem is formulated as a non-linear model in a general way. We analyze several properties of the problem, which lead to dominance property for the general case when transporters are heterogeneous. This dominance property is enhanced in the identical transporters case. Based on these dominant properties, a dynamic programming algorithm has been proposed and compared with CPLEX in the case where the delivery time is not batch-size dependent, which allows to have a linear model. In the objective of solving large instances, we have developed an Improved Genetic Algorithm as an approach method to solve the problem. Each method of resolution have been evaluated computationally and the performance of the fast exact method of resolution has been compared with the developed GA.

The computational results show that the dynamic programming approach with heterogeneous transporters is much more efficient than CPLEX for solving small and medium-sized instances. The GA has been evaluated with the *GDP*, where the error ratio between the GA and the *GDP* demonstrate that the efficiency of the GA decreases significantly when the storage cost at the customer  $\beta$  increase, where the problem becomes more complicated to solve.

In the case of identical transporters, the results show that the proposed dominance relation reduces more drastically the time of resolution compared with the proposed MIP model solved by Cplex. The time of resolution of this method does not exceed 36.00 seconds in the most complicated case where  $n = 30$ ,  $m = 4$  and  $\beta$  is randomly generated from the uniform distribution with ranges  $[10, 50]$ .

The study presented in this chapter may be extended in several directions. First, improving the efficiency of the proposed dynamic programming method in the general case, although it is already an efficient method, in order to solve larger problem instances. For that purpose, it is possible to use a lower bound by solving a similar problem with identical transporters case. Secondly, from both theoretical and practical point of view, there exist other possible objectives of interests to be considered, to this initial model in order to be closer to the industrial reality of the considered problem. The first extension could be to consider a different schedule between the production stage and the delivery stage, which means that products may wait between the end of the production stage and the start of the delivery. The second one is to consider the multi-products case when products have different volumes. The methods presented in this chapter form a foundation for modeling and solving these extensions.



# 5 Multi-Customers/Transporters Integrated Lot Sizing and Delivery Scheduling Problem

## 5.1 Introduction

The supply chain optimization problem we discuss in this chapter considers delivery-inventory decisions. In particular, we study a multi-customer multi-transporter supply chain model, coordinating delivery-inventory decisions. The coordination between supplier and customer for improving the performance of inventory control has received a great deal of attention and the integrated approach has been studied for years. Typically, integrated approach focuses on the production-inventory decisions of supply chain partners while minimizing the total relevant cost of the system. During processing and supply of the product, inventory accumulates with the manufacturer and also with each of the buyers in this system. Delivery of the product in small lots reduces the inventory cost but increases set-up, ordering and transportation cost. On the other hand, delivery in larger lots increases inventory cost but reduces the other costs, and scheduling interference results because of scarce storage capacity at both the manufacturer and the buyers. Synchronization of the production flow is essential for the control of inventory and hence for minimizing the total cost of the system. The integration approach has been researched for years. Much research has focused on this area under various assumptions and objective measures that differ from the problem proposed in this chapter.

This chapter considers integrated delivery-inventory for the supply of multi-items to more than one customer, after its production by a manufacturer. The manufacturer transfers the jobs to a customers by available transporters (Figure 5.1), in different delivery scenarios. The aim of this chapter is to determine the best way to construct an optimal scheduling for the delivery of customers orders, specifying the arrival time of each product and to develop an algorithmic approach capable of establishing solution feasibility for instances of various types and difficulty levels. We focus on single plant production (supplier) scheduling with multi-product, multi-customer delivery-storage problem. The jobs are operated in batch mode and the sequence dependent setup and cost time are considerable in the general model of this study, in the objective to study the delivery-inventory stage, with different delivery structures. In the first scenario the problem is studied in a general way,

where a fleet of heterogeneous transporters are available at the supplier to serve the customers. Here, each transporter could serve different customers without any pre-allotment to any customer. Then in the second scenario the problem is studied with multiple homogeneous transporters without allotment also. In the third model we studied the case of multiple homogeneous transporters with pre-allotment delivery structure, where the customer will be served according to the pre-allotted proposition. In the fourth model, we studied the case of one transporter available to serve one customer. The objective is to determine an optimal scheduling for the delivery of customer orders, specifying the arrival time of each product in the different scenarios. The general objective of this chapter is to analyze the efficiency of the results of the optimal delivery and inventory scheduling in the studied scenarios. A comparative study of the solutions of several numerical problems is also carried out, to validate the analytical findings and to examine the limitations of the methods used.

The organization of the remainder of this chapter is structured as follows. In Section 2, the problem definition and formulation is introduced, the notation used along this chapter are defined, then some assumptions are given. In Section 3, the problem is studied as a general model, then we proposed B&B algorithm as an exact method of resolution to solve the general model, and different extensions of the general model are formulated as Mixed-Integer-Programming Models. In section 4 and 5, we provide the experimental results and draw some conclusions of this chapter.

## 5.2 Problem definition and formulation

In the supplier side, a facility of multi-machines is available to produce a finite number of heterogeneous jobs (products), which prevents the supplier's storage cost assumption. Each job needs a constant processing time, however we assume that the total production time of any batch is less than the minimum round-trip delivery time for the transporters. Each round trip between the factory and each customer  $h$  requires a delivery cost  $\eta_h$  as well as a delivery time  $\tau_h$ . The batches delivered from the vendor to the customers can be equal or unequal-sized batches. It is assumed that directing delivery method is used for sending the batches to the customers. Multiple transporters are considered to deliver the product from the supplier to the customers. Each transporter has a specific  $c_t$ , where the total number of job belong to the same batch cannot exceed the capacity of the vehicle used. If job  $j$  of customer  $h$  is delivered before its due date  $d_j$ , it is known as a early job and a cost equal to  $\beta_h$  is incurred, where  $\beta_h$  equal to the earliness penalty of the job  $j$  belongs to the customer  $h$ . Batching and sending several jobs in the batches will reduce the transportation costs. The cost of system includes delivery and holding costs. The objective is to determine the sequence of batches that has to be processed and scheduled of the distributions that have to be used by customers, such that the expected total cost of both vendor and customers be minimized.



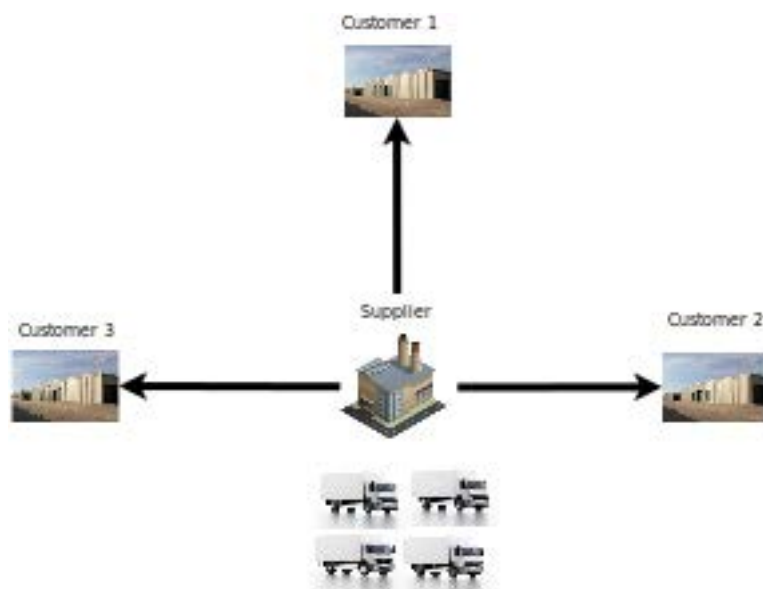
In Figure 5.1 a general multi-customer multi-transporter model has been depicted as an example.

This chapter studies the case where a single supplier delivers a homogeneous product to a multiple customers. The customers order the product at the vendor and receive the products in batch shipments. Delivery of the product in small lots reduces the inventory cost but increases transportation cost. On the other hand, delivery in larger lots increases inventory cost but reduces the other costs, and scheduling interference results because of scarce storage capacity at both the manufacturer and the customers.

Each round trip between the factory and a customer  $h$  requires a delivery cost  $\eta_{bht}$  as well as a delivery time  $\tau_{bht}$  dependent of the size of the batch  $b$ , the transporter  $t$  and the customer  $h$ . Both scenarios of the delivery structure will be discussed later. Each considered transporter has a capacity  $c_t$ , where the total number of jobs belonging to the same batch does not exceed the capacity of the vehicle used. If job  $j$  of customer  $h$  is delivered before its due date  $d_j$ , it is known as an early job and it incurs a cost equal to  $\beta_h$  dependent of customer, where  $\beta_h$  equal to the earliness penalty at the customer  $h$ .

Figure 5.1 illustrates the general model for the delivery consideration. It represents the case where a fleet of transporters could serve all the customers without allotment. Whereas the other models will be represented along the chapter.

The objective is to determine the sequence of batches that has to be processed and scheduled of the distributions that have to be used by customers, such that the expected total cost will be minimized. The cost of the system includes delivery and storage costs.



**Figure 5.1:** Multi-customer Multi-transporter general model

### 5.2.1 General notations

The following terminology is used in this chapter:

*Parameters*

- $m$ : number of customers,
- $n$ : number of jobs,
- $p$ : number of transporters,
- $J = \{1, 2, \dots, n\}$ : set of all jobs,
- $H = \{1, 2, \dots, m\}$ : set of all customers,
- $T = \{1, 2, \dots, p\}$ : set of all transporters,
- $j$ : index for jobs,  $j \in J$ ,
- $h$ : index for customers,  $h \in H$ ,
- $t$ : index for customers,  $t \in T$ ,
- $k$ : index for batches,
- $n_h$ : number of products by customer,
- $d_j$ : due date of the job  $j$ ,
- $cl_j$ : customer destination of the job  $j$ ,
- $c_t$ : capacity of the transporter  $t$ ,
- $\tau_{b_k ht}$ : time for the vehicle  $t$  to deliver a batch of size  $b_k$  to customer  $h$  and to return to the supplier location,
- $\eta_{b_k ht}$ : delivery cost for vehicle  $t$  to deliver a batch of size  $b_k$  to customer  $h$  and to return to the supplier location,
- $\alpha(k)$ : delivery cost of batch  $k$ ,
- $\beta_j$ : customer earliness penalty function for job  $j$ , which may also depend from the customer.

*Primary variables*

- $\delta_{jk}^1 = 1$  if the job  $j$  belongs to the  $k^{th}$  batch, 0 otherwise,
- $\delta_{kh}^2 = 1$  if the batch  $k$  is delivered to the customer  $h$ , 0 otherwise,
- $\delta_{kht}^3 = 1$  if the batch  $k$  is delivered to the customer  $h$  by the transporter  $t$ , 0 otherwise,

*Secondary variables*

- $C_j$ : the arrival time of job  $j$  at the customer,
- $B_k$ : the arrival time of the batch  $k$  at the customer,
- $b_k$ : size of batch  $k$
- $y_k = 1$  if the batch  $k$  exists and is not empty, 0 otherwise,
- $u_h$ : number of delivered batches for customer  $h$ ,

## 5.2.2 Assumptions

Apart from the assumptions already stated and in developing the models we assume the general following hereafter:

1. The system consists of  $m$  customers who are supplied by a single-supplier, multiple customers and multiple transporters available to deliver the products to the customer  $h$  to which it belongs.
2. Each customer  $h$  orders a finite number of jobs to the manufacture,  $\{J_1, J_2, \dots, J_n\}$
3. The demands among the customers are independent in time.
4. The sequence and machine dependent setup time are negligible.
5. The jobs have to be ready for delivery in batches by their common contracted due date for each job.
6. Each customer estimates individual demand equal to  $n_h$ , inventory costs are denoted  $\beta_j$ .
7. The round trip time of transporters ( $\tau_h$ ) from the supplier to the customers depend of the size of the batch.
8. The storage cost at the customers  $\beta_j$  is dependent of the customer and the considered job in the context of multi-product.

## 5.3 The General studied Model

In this section, the mathematical programming models is presented. Using the structural properties of each model, we developed different MIP models for the mentioned problems as follows:

The objective function of the studied problem is the same for the both models, and it is represented as follow:

$$\text{Min } Z = \sum_{h \in H} \sum_{k=1}^{u_h} \alpha(b_k) + \sum_{j=1}^n \beta_j (d_j - C_j) \quad (5.1)$$

The two terms of the objective function in equation (Equation 5.1) represent on the one hand, the delivery costs  $\sum_{h \in H} \sum_{k=1}^{u_h} \alpha(b_k)$ , and, on the other hand, the inventory costs of the jobs through  $\beta_j (d_j - C_j)$ . For ease of reference, we denote this problem: Multiple Customers-Transporters Lot Sizing Delivery Scheduling Problem (MCTLSDSP).

### 5.3.1 The general model with batch size dependent time and cost

In the general model we studied the case of single-supplier/multi-customer with multiple heterogeneous transporters available to serve the customers. Here, the capacity of transporters is different and the batch size dependent proposition is taken into consideration, which means that the delivery time and cost is dependent of the batch size and of the considered transporter. In the first time we developed a Branch and Bound method to study this problem, then a linear Mixed-integer-programming model is proposed to study this problem. Due to the linearisation status of the MIP model, we supposed that the delivery batch size and cost of all batches is the same.

### 5.3.2 Branch and Bound Algorithm

In this section, we describe the B&B algorithm that we have developed to solve the MCTLSDSP. The objective of this B&B is to solve small and medium instances, and to be a reference for validating the efficiency of the proposed heuristic algorithms. This B&B algorithm maintains a list of subproblems (nodes) whose union of feasible solutions contains all feasible solutions of the original problem. The list is initialized with the original problem itself. In each major iteration, the algorithm selects a current subproblem from the list of unevaluated nodes. This branching seems to be natural, however the number of branches will be very large for large problems. Consequently, if this method is used in the B&B algorithm, it may take too much time to find optimal solutions, as redundant schedules would be checked repeatedly. Yet, several of the subproblems would already have been eliminated upon the generation of nodes, since the B&B tree includes redundant solutions.

At each node of the search tree, the number of products that still needs to be delivered with each transporter to each customer has to be updated. Iterations are performed until the list of subproblems to be processed is empty. The crucial part of a successful B&B algorithm is the computation of the lower bounds. Therefore, we have developed a lower bound described in the next part.

Efficient lower bound would significantly reduce the time and efforts needed for the B&B method. Based on the main feature of the problem, the lower bound value for the problem is the summation of lower bounds on the total earliness cost and the transportation cost. We assume that  $w$  is a partial batch sequence solution,  $z(w)$  is the evaluation of  $w$ , and  $r_h(w)$  is the number of products remaining at the customer  $h$  for partial solution  $w$ . This notation will be used throughout this part.

In each node, the solutions are built from the last batch to the first one for each transporter and each customer. The evaluation of the partial or complete solution is processed with backward equations. The research of a solution starts by constructing a partial solution  $w$ . Then, the remainder of products is added in order to

get a complete solution, with the objective of achieving a minimum delivery cost. Therefore, the more the transporter will be loaded, the more this lower bound will be efficient.

**Proposition 1** *For a partial solution  $w$ , a lower bound for the delivery cost of the remaining products is given by:*

$$\sum_{h=1}^m \left\lceil \frac{r_h(w)}{\max_{t \in T} (c_t)} \right\rceil \times \min_{t \in T} (\eta_{0ht}) \quad (5.2)$$

*Proof.* For each customer  $h$ , if  $r_h(w)$  is the number of products remaining to be delivered, the minimum number of round trips will be equal to  $\left\lceil \frac{r_h(w)}{\max_{t \in T} (c_t)} \right\rceil$ , and the delivery cost of the remaining products is as denoted in equation (Equation 5.2).  $\square$

We add the partial solution  $w$  to the solution found in equation (Equation 5.2) to get the lower bound of the current node under study.

**Corollary 1** *The lower bound  $LB(w)$  of the partial solution  $w$  is given as follows:*

$$LB(w) = z(w) + \sum_{h=1}^m \left\lceil \frac{r_h(w)}{\max_{t \in T} (c_t)} \right\rceil \times \min_{t \in T} (\eta_{0ht}) \quad (5.3)$$

*Proof.* Straightforward.  $\square$

In the next section, we proposed a Mixed-integer-programming model to solve the general case of the single-supplier/multi-customer with multiple heterogeneous transporters.

### 5.3.2.1 The general Mixed-Integer-Programming model

The general model considered previously is non-linear, due to the assumption that the delivery costs and times are dependent of the batch size. In order to model this problem with a MIP approach, we have to linearize it and we assume that the delivery costs and times will only be dependent of the considered transporter and customer. In the remaining of this chapter, we denote :

- $\tau_{ht}$ : time for the vehicle  $t$  to deliver a batch to customer  $h$  and to return to the supplier location,
- $\eta_{ht}$ : delivery cost for vehicle  $t$  to deliver a batch to customer  $h$  and to return to the supplier location,

Here we studied the general case of the single-supplier/multi-customer with multiple heterogeneous transporters with the assumption of identical batch size time and cost. A principal important variable is used in this model with is  $\delta_{kht}^3$  where the batch  $k$  is deliver to customer  $h$  with the transporter  $t$ . The transporter have different capacity and available to serve all the customers in the same time or progressively.

Due to the linearisation status of the MIP model, the objective fuction of the MCTLSDSP is given by:

$$Min Z = \sum_{t \in T} \sum_{h=1}^m \sum_{k=1}^{u_h} \eta_{ht} \times \delta_{kht}^3 + \sum_{j=1}^n \beta_j (d_j - C_j) \quad (5.4)$$

Subject to:

$$\sum_{k=1}^n \delta_{jk}^1 = 1, \quad j = 1, \dots, n \quad (5.5)$$

$$\sum_{h=1}^m \sum_{t=1}^p \delta_{kht}^3 \leq 1, \quad k = 1, \dots, n \quad (5.6)$$

$$\sum_{t=1}^p \delta_{k,cl_j,t}^3 \geq \delta_{jk}^1, \quad j, k = 1, \dots, n \quad (5.7)$$

$$u_h = \sum_{k=1}^n \sum_{t=1}^p \delta_{kht}^3, \quad h = 1, \dots, m \quad (5.8)$$

$$y_k \geq y_{k+1}, \quad k = 1, \dots, n - 1 \quad (5.9)$$

$$\sum_{j=1}^n \delta_{jk}^1 \leq \sum_{h=1}^m \sum_{t=1}^p c_t \times \delta_{kht}^3, \quad k = 1, \dots, n \quad (5.10)$$

$$C_j \leq d_j, \quad j = 1, \dots, n \quad (5.11)$$

$$C_j = \sum_{k=1}^n B_k \times \delta_{jk}^1, \quad j = 1, \dots, n \quad (5.12)$$

$$B_{k_2} - B_{k_1} - \sum_{h=1}^m (\tau_{ht} + M) \times (\delta_{k_2,h,t}^3 + \delta_{k_1,h,t}^3) \geq -2 \times M, \quad (5.13)$$

$$k_1, k_2 = 1 \dots n, \quad k_2 > k_1, \quad t = 1 \dots p$$

$$C_j \geq 0, \quad j = 1, 2, \dots, n \quad (5.14)$$

$$\delta_{jk}^1, \delta_{kht}^3 \in \{0, 1\}, \quad j, k = 1, \dots, n; \quad h = 1, \dots, m; \quad t = 1 \dots p \quad (5.15)$$

Constraint (Equation 5.5) guarantees that, each job must be scheduled exactly in one batch, in this constraint the jobs will be batched only in the batch which it belongs. Constraints (Equation 5.6) and (Equation 5.7) indicate that, each batch must be scheduled exactly to one customer by a specific transporter  $t$ , in this constraint the batches will be delivered only to the customer which it belongs with the transporter  $t$ . Constraint (Equation 5.8) calculates the number of batches delivered to each customer and the transporter used to transport this batch. Constraint (Equation 5.9) guarantees that no empty batch is allowed. Constraint (Equation 5.10) prevents the number of jobs scheduled in one delivery batch to exceed the capacity of the transporter used. Constraint (Equation 5.11) indicates that arrival time of each job is at least equal to the contracted due date for each customer. Constraint (Equation 5.12) represents the conversion of the completion time of the products belong to the same batch  $C_j$  to  $B_k$  the arrival time of each batch to the customers. This constraint represented in a non-linear way in this mathematical representation to facilitate the understanding of the problem. Constraints (Equation 5.13) shows the relation between the arrival time of two consecutive batches and the delivery time which their belong and the availability of a transporter  $t$  at the supplier, to prevents the overlap between two consecutive batches in the case of two consecutive jobs delivered separately in two different batches. This constraint take into consideration the cases of: two or more consecutive batches belong to same customer and delivered with the same transporter  $t$ ; two or more consecutive batches belongs to two different customers by the same transporter  $t$ ; two or more consecutive customers served by two or more different transporters. Constraints (Equation 5.14) and (Equation 5.15) define the range of the variables.

The general proposed MIP model and the B&B algorithm developed in the previous sections could solve the small and some medium size instances, however the time of resolution to solve the large size instances grows exponentially. In the next section, a specific models are proposed as a particular case to solve the problem rapidly and to study its performances.

### 5.3.2.2 Specific model with homogeneous delivery structure

In the first specific model, we suppose that the transporters are homogeneous and we keep the consideration of the availability of the transporters to serve the customers without allotment. Here the parameters of the capacity of the transporters  $c_t$  becomes the same to all the transporter used and it equals to  $c$ . The variable  $\delta_{kht}^3$  equals to 1 when the batch  $k$  will be delivered to the customer  $h$  with the transporter  $t$  will be replaced here by  $\delta_{kh}^2$ , which means that each batch  $k$  belongs to customer  $h$  without any consideration of the transporter used to serve this batch due the assumption of the homogeneous transporters specified in this section. For this model the constraints are defined as follows:

subject to :

$$\sum_{k=1}^n \delta_{jk}^1 = 1, \quad j = 1, \dots, n \quad (5.16)$$

$$\sum_{h=1}^m \delta_{kh}^2 \leq 1, \quad k = 1, \dots, n \quad (5.17)$$

$$\delta_{k,cl_j}^2 \geq \delta_{jk}^1, \quad j, k = 1, \dots, n; \quad k \leq j \quad (5.18)$$

$$u_h = \sum_{k=1}^n \delta_{kh}^2, \quad h = 1, \dots, m \quad (5.19)$$

$$y_k \geq y_{k+1}, \quad k = 1, \dots, n - 1 \quad (5.20)$$

$$\sum_{j=k}^n \delta_{jk}^1 \leq c, \quad k = 1, \dots, n \quad (5.21)$$

$$C_j \leq d_j, \quad j = 1, \dots, n \quad (5.22)$$

$$C_j = \sum_{k=1}^n B_k \times \delta_{jk}^1, \quad j = 1, \dots, n \quad (5.23)$$



$$B_{k+nt} - B_k - \sum_{h=1}^m \tau_{ht} \times (\delta_{k+nt,h}^2 + \delta_{k,h}^2) \geq 0, \quad (5.24)$$

$$k = 1 \dots n, \quad k \leq n - nt$$

$$C_j \geq 0, \quad j = 1, 2, \dots, n \quad (5.25)$$

$$\delta_{jk}^1, \delta_{kh}^2 \in \{0, 1\}, \quad j, k = 1, \dots, n; \quad h = 1, \dots, m \quad (5.26)$$

Constraint (Equation 5.16) guarantees that, each job must be scheduled exactly in one batch, in this constraint the jobs will be batched only in the batch which it belongs. Constraints (Equation 5.17) and (Equation 5.18) indicate that, each batch must be scheduled exactly to one customer, in this constraint the batches will be delivered only to the customer which it belongs. Constraint (Equation 5.19) calculates the number of batches delivered to each customer. Constraint (Equation 5.20) guarantees that no empty batch is allowed. Constraint (Equation 5.21) prevents the number of jobs scheduled in one delivery batch to exceed the capacity of the vehicle. Constraint (Equation 5.22) indicates that arrival time of each job is at least equal to the contracted due date for each customer. Constraint (Equation 5.23) represents the conversion of the completion time of the products belong to the same batch  $C_j$  to  $B_k$  the arrival time of each batch to the customers. This constraint represented in a non-linear way in this mathematical representation to facilitate the understanding of the problem. Constraints (Equation 5.24) shows the relation between the arrival time of two consecutive batches and the delivery time which their belong and the availability of a vehicle at the supplier, to prevents the overlap between two consecutive batches in the case of two consecutive jobs delivered separately in two different batches. Constraints (Equation 5.25) and (Equation 5.26) define the range of the variables.

### 5.3.2.3 Specific model with homogeneous transporters allotted to each customer

In this model, we assume that each customer  $h$  has a fleet of transporters allotted to each customer to deliver the products to its customer. We introduced the new notation called  $nt_h$ , which is the number of transporters allotted to customer  $h$ , where  $nt_h = \frac{(nt \times n_h)}{n}$  with  $h = 1, \dots, m$ . Here we supposed that all the transporter used have identical capacity equals to  $c$ . The set of batches for each customer  $h$  is

$$k_h \in \{n_1, \dots, n_h\}, \quad h \in H$$

Since they are different customers, the cumulative sum of products by customer per demand is

$$x_h = \sum_{h=0}^m n_h, \quad h \in H \text{ and } x_0 = 0$$

subject to :

$$\sum_{k=1}^{k_{[cl_j]}} \delta_{j,k}^1 = 1, \quad j = 1, \dots, n \quad (5.27)$$

$$\sum_{h=1}^m \delta_{k,h}^2 \leq 1, \quad k = 1, \dots, n \ \& \ h = cl_k \quad (5.28)$$

$$\delta_{k,cl_j}^2 \geq \delta_{j,k}^1, \quad j = 1, \dots, n \ \& \ k = 1, \dots, k_{[cl_j]} \quad (5.29)$$

$$u_h = \sum_{k=1}^n \delta_{kh}^2, \quad h = 1, \dots, m \ \& \ h = cl_k \quad (5.30)$$

$$\sum_{j=1}^{k_{[cl_k]}} \delta_{j,k}^1 \leq c, \quad k = 1, \dots, n \quad (5.31)$$

$$C_j \leq d_j, \quad j = 1, \dots, n \quad (5.32)$$

$$C_j = \sum_{k=1}^{k_{[cl_j]}} B_k \times \delta_{j,k}^1, \quad j = 1, \dots, n \quad (5.33)$$

$$B_{k_2} - B_{k_1} - \sum_{h=1}^m [(\tau_{ht}/nt[h]) + M] \times (\delta_{k_2,h}^2 + \delta_{k_1,h}^2) \geq -2 \times M, \quad k_1 < k_2 \ \& \ h \in H \quad (5.34)$$

$$C_j \geq 0, \quad j = 1, 2, \dots, n \quad (5.35)$$

$$\delta_{jk}^1, \delta_{kh}^2 \in \{0, 1\}, \text{ with } j, k = 1, \dots, n, \text{ \& } h \in H \quad (5.36)$$

Constraint (Equation 5.27) guarantees that, each job must be scheduled exactly in one batch, in this constraint the bounds  $k_{cl_j}$ ,  $x_{[cl_j-1]} + k$  and  $(k \times c) - x_{[cl_j-1]} \geq j \geq k - x_{[cl_j-1]}$  limit the combination of the number of bearable batches of job  $j$ . Constraints (Equation 5.28 and Equation 5.29) indicate that, each batch must be scheduled exactly to one customer, where each job could be scheduled in the batch  $k$  according to the proposed bounds  $(k \times c) - x_{[cl_j-1]} \geq j \geq k - x_{[cl_j-1]}$ . Constraint (Equation 5.30) calculates the number of batches delivered to each customer, with  $h = cl_k$ . Constraint (Equation 5.31) prevents the number of jobs scheduled in one delivery batch to exceed the capacity of the vehicle. Here, the jobs belong to the same customer is limited between  $(k \times c) - x_{[cl_j-1]}$  and  $k - x_{[cl_j-1]}$  according to the proposed bounds in equation (Equation 5.31). Constraint (Equation 5.32) indicates that the arrival time of each job is at least equal to the contracted due date for each customer. Constraint (Equation 5.33) represents the conversion of the completion time  $C_j$  of the jobs between  $(k \times c) - x_{[cl_j-1]}$  and  $k - x_{[cl_j-1]}$  belong to the same batch  $k$  to  $B_k$  the arrival time of batches. This constraint represented in a non-linear way in this mathematical representation to facilitate the understanding of the problem. Constraints (Equation 5.34) shows the relation between the arrival time of two batches belong to the same customer ( $x_{[cl_{j_1-1}]} + k_1$  and  $x_{[cl_{j_2-1}]} + k_2$ ), the delivery time which their belong and the availability of a vehicle belongs to this customer at the supplier, to prevent the overlap between the batches in the case of jobs delivered separately in two different batches. The conditions used in this constraint set Constraints (Equation 5.35) and (Equation 5.36) define the range of the variables.

#### 5.3.2.4 Specific model with homogeneous one transporter allotted to one customer ( $h$ equals to $p$ )

Here we supposed that each customer has one allotted transporter used to serve each customer. The capacity of all the transporters used is the same and equals to  $c$ .

subject to :

$$\sum_{k=1}^n \delta_{j,k}^1 = 1, \quad j = 1, \dots, n \quad (5.37)$$

$$\sum_{h=1}^m \delta_{k,h}^2 \leq 1, \quad k = 1, \dots, n \quad (5.38)$$

$$\delta_{k,cl_j}^2 \geq \delta_{j,k}^1, \quad j \& k = 1, \dots, n \quad (5.39)$$

$$u_h = \sum_{k=1}^n \delta_{kh}^2, \quad h = 1, \dots, m \quad (5.40)$$

$$y_k \geq y_{k+1}, \quad k = 1, \dots, n - 1 \quad (5.41)$$

$$\sum_{j=1}^n \delta_{j,k}^1 \leq c, \quad k = 1, \dots, n \quad (5.42)$$

$$C_j \leq d_j, \quad j = 1, \dots, n \quad (5.43)$$

$$C_j = \sum_{k=1}^n B_k \times \delta_{j,k}^1, \quad j = 1, \dots, n \quad (5.44)$$

$$B_{k_2} - B_{k_1} - \sum_{h=1}^m (\tau_{ht} + M) \times (\delta_{k_2,h}^2 + \delta_{k_1,h}^2) \geq -2 \times M, \quad k_1, k_2 = 1 \dots n, \& k_2 > k_1 \quad (5.45)$$

$$C_j \geq 0, \quad j = 1, 2, \dots, n \quad (5.46)$$

$$\delta_{jk}^1, \delta_{kh}^2 \in \{0, 1\}, \quad \text{with } j, k = 1, \dots, n, \& h \in H \quad (5.47)$$

Constraint (Equation 5.37) guarantees that, each job must be scheduled exactly in one batch, in this constraint the jobs will be batched only in the batch which it belongs. Constraints (Equation 5.38 and Equation 5.39) indicate that, each batch must be scheduled exactly to one customer, in this constraint the batches will be delivered only to the customer which it belongs. Constraint (Equation 5.40) calculates the number of batches delivered to each customer. Constraint (Equation 5.41) guarantees that no empty batch is allowed. Constraint (Equation 5.42) prevents the

number of jobs scheduled in one delivery batch to exceed the capacity of the vehicle. Constraint (Equation 5.43) indicates that the arrival time of each job is at least equal to the contracted due date for each customer. Constraint (Equation 5.44) represents the conversion of the completion time  $C_j$  belong to the same batch  $k$  to  $B_k$  the arrival time of batches. This constraint represented in a non-linear way in this mathematical representation to facilitate the understanding of the problem. Constraints (Equation 5.45) shows the relation between the arrival time of two consecutive batches and the delivery time which their belong, to prevents the overlap in the case of two consecutive batches delivered separately to the same customer. In this constraint we remark that the relationship is confined only between the batches and the customer to which it belongs  $\delta_{k,h}^2$ . Constraints (Equation 5.46) and (Equation 5.47) define the range of the variables.

## 5.4 Experimental Results

The MIP models was solved with CPLEX solver to exactly solve the models. The computations were carried out on a desktop Intel core 2 processor operating at 2.67 GHz clock speed and 4 GB RAM. The models where run for at most one hour by instance or until the solver found the models optimal solutions or ran out of the memory, in this case we collect the best optimal solutions found. The solver was terminated when the time of resolution exceeds 3600 seconds.

The following sections describe the main results obtained from the comparison of the mathematical programming formulation proposed models with a random instance. The efficiency is the most important criterion for evaluating the results obtained from the different models. Inside it, we study the quality of solutions and the computational time in each model.

### 5.4.1 Evaluation of the quality of different models

In order to evaluate the coordination mechanisms developed in this chapter, let us solve a small problem of supplying 8 jobs to 2 different customers. The jobs are delivered by two transporters depend of the delivery structure used. The capacity of the transporters equal to 2 in all the specific proposed models, and it equals to (2 and 3) for the transporter 1 and 2 respectively, for the general model. In the customers side, the transporter delivery time and cost depend of the the distance between the supplier and each customer with ( $\eta_{1t} = 8$ ,  $\tau_1 = 11$ ,  $\eta_{2t} = 10$ ,  $\tau_{2t} = 15$ ) belongs to customer 1 and 2 respectively, and the customer cost in each side equals to ( $\beta_{j1} = 1$ ,  $\beta_{j2} = 2$ ) for all jobs  $j1$  (resp.  $j2$ ) requested by customer 1 (resp. 2). The first customer ordered 5 jobs and the second customer ordered 3 jobs as follows; ( $j_1, j_2, j_5, j_6, j_8$ ) belongs to customer 1 and ( $j_3, j_4, j_7$ ) belongs to customer 2, with a due dates  $\{10010, 10020, 10050, 10060, 10080\}$  and  $\{10030, 10040, 10070\}$  belongs

to customer 1 and 2 respectively. The data of different parameters are given in Table 5.4.1.

**Table 5.1:** Data for a single-supplier and 2-customers problem

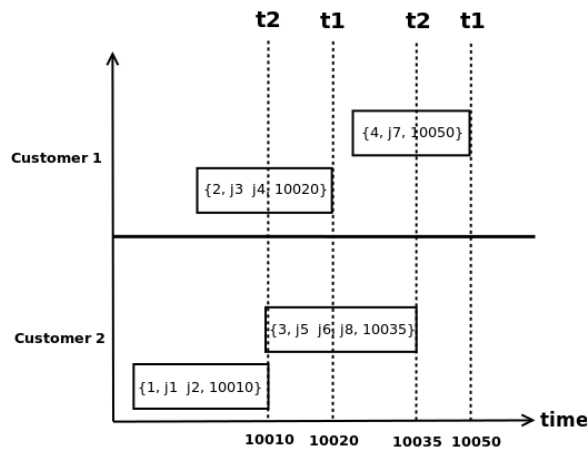
Parameters	Customer 1	Customer 2
$\tau_{ht}$	11	15
$\eta_{ht}$	16	20
$\beta_j$	1	2
$n_h$	5	3
$j = 1, \dots, n$	$j_1, j_2, j_5, j_6, j_8$	$j_3, j_4, j_7$
$d_j$	$\{d_1, d_2, d_5, d_6, d_8\}$ $\{10010, 10015, 10035, 10040, 10050\}$	$\{d_3, d_4, d_7\}$ $\{10020, 10030, 10050\}$

A comparative evaluation of the results of each method is given in Figure 5.2, Figure 5.3, Figure 5.4 and Figure 5.5, where an illustration of the models and the result for each proposed delivery structures is presented.

Lets denote that  $\{k; j; C_j\}$ , the notation which is used in Table 5.4.1 for a solution of one batch, where the first term  $k$  describes the batch number, the second term  $j$  describes the jobs which will be delivered in this batch and the third term  $C_j$  is the arrival time of the products which are in batch  $k$  at the customer's to which it belongs.

### 5.4.1.1 Evaluation of the general model

For the first model which is the general model of the single-supplier multi-customer with multi-transporter. The result in Figure 5.2 shows that there is 4 batches with 2 batches for each customer with a delivery cost equals to  $2 \times \eta_{1t} + 2 \times \eta_{2t} = 32 + 40 = 72$ .

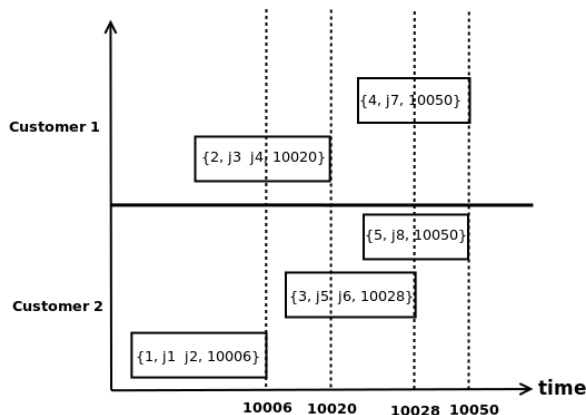


**Figure 5.2:** Illustration of models for different delivery structures

The jobs  $j_1$  and  $j_2$  arrive in the first batch at 10010 by the second transporter  $t_2$ , which incurs an inventory cost for job  $j_2$  at the second customer equals to  $(5 \times 1 = 5)$ . Then, in the second batch the jobs 3 and 4 arrive at 10020 by the transporter  $t_1$ , which incurs an inventory cost for job  $j_4$  at the first customer equals to  $(10 \times 2 = 20)$ . For the third batch, the transporter  $t_2$  delivers the jobs  $(j_5, j_6, j_8)$  at 10035, which incurs an inventory cost for job  $j_5$  and  $j_8$  at the customer 2 equals to  $(5 \times 1 + 15 \times 1 = 20)$ . In the last batch belongs to customer 1, the job  $j_7$  arrives on time and it is not incur any inventory cost. The total cost for general model becomes equal to  $72 + 5 + 20 + 20 = 117$ .

### 5.4.1.2 Evaluation of the specific model with homogeneous transporters without allotment

For the specific model which is the multi-homogeneous delivery structure of the single-supplier multi-customer with multi-transporter. The result is depicted in Figure 5.3 shows that there is 5 batches with 2 batches belong each customer 1 and three batches belong to customer 2, which incur a delivery cost equals to  $3 \times \eta_{1t} + 2 \times \eta_{2t} = 48 + 40 = 88$ .

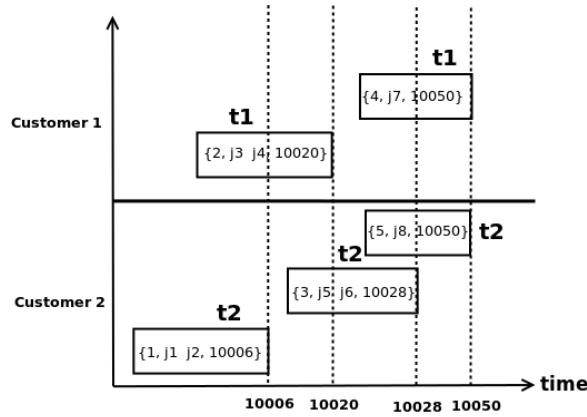


**Figure 5.3:** Illustration of models for different delivery structures

The jobs  $j_1$  and  $j_2$  arrive in the first batch at 10006, which incurs an inventory cost for job  $j_1$  and  $j_2$  at the second customer equals to  $(4 \times 1 + 9 \times 1 = 13)$ . Then, in the second batch the jobs 3 and 4 arrive at 10020, which incurs an inventory cost for job  $j_4$  at the first customer equals to  $(10 \times 2 = 20)$ . For the third batch, the jobs  $(j_5, j_6)$  arrive at 10028, which incurs an inventory cost at the customer 2 equals to  $(7 \times 1 + 12 \times 1 = 19)$ . In the two last batches belong to customer 1 and 2 respectively, the job  $j_7$  and  $j_8$  arrive on time and they not incur any inventory cost. The total cost for general model becomes equal to  $88 + 13 + 20 + 19 = 140$ .

### 5.4.1.3 Evaluation of the specific model with homogeneous transporters allotted to each customer

For the specific model which is the homogeneous transporters allotted to each customer. here the transporters  $t_1$  and  $t_2$  are allotted to customers 1 and 2 respectively. The result is depicted in Figure 5.4 and it shows that there is 5 batches with 2 batches belong each customer 1 and three batches belong to customer 2, which incur a delivery cost equals to  $3 \times \eta_{1t} + 2 \times \eta_{2t} = 48 + 40 = 88$ .



**Figure 5.4:** Illustration of models for different delivery structures

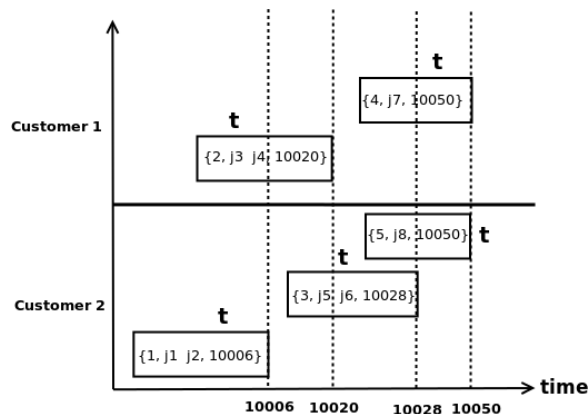
The jobs  $j_1$  and  $j_2$  arrive in the first batch at 10006 by the transporter allotted to this customer which is  $t_2$ . The jobs  $j_1$  and  $j_2$  incur an inventory cost for at the second customer equals to  $(4 \times 1 + 9 \times 1 = 13)$ . Then, in the second batch the jobs 3 and 4 arrive at 10020 by the transporter  $t_1$ , which incurs an inventory cost for job  $j_4$  at the first customer equals to  $(10 \times 2 = 20)$ . For the third batch, the jobs  $(j_5, j_6)$  arrive at 10028 by the transporter  $t_2$ , which incurs an inventory cost at the customer 2 equals to  $(7 \times 1 + 12 \times 1 = 19)$ . In the two last batches belong to customer 1 and 2 respectively, the job  $j_7$  and  $j_8$  arrive in parallel on time by the transporters  $t_1$  and  $t_2$  respectively and they not incur any inventory cost. The total cost for general model becomes equal to  $88 + 13 + 20 + 19 = 140$ .

### 5.4.1.4 Evaluation of the specific model with homogeneous one transporter allotted to one customer ( $h$ equals to $p$ )

For the specific model which is the homogeneous one transporter allotted to one customer ( $h$  equals to  $p$ ) scenario, the number of batches equal to 5 with three batches for customer 2 and two batches for customer 1, which mean that the delivery cost for this model equals to  $3 \times \eta_{2t} + 2 \times \eta_{1t} = 48 + 40 = 88$ .

We start by the second customer where we have one transporter allotted to this customer, in the first batch belongs to this customer we have two jobs ( $j_1$  and  $j_2$ ) arrive at 10006 which incurs an inventory cost equals to  $(4 \times 1 + 9 \times 1 = 13)$ . In





**Figure 5.5:** Illustration of models for different delivery structures

the second batch belongs to the same customer (batch 3) jobs ( $j_5$  and  $j_6$ ) arrive at 10028 with 7 units of time for job  $j_5$  and 12 units of time for job  $j_6$ , which incurs an inventory cost at the customer 2 equals to  $(7 \times 1 + 12 \times 1 = 19)$ . In the last batch belongs to customer 2, the job  $j_8$  arrives on time without any inventory cost. Now, for the first customer, the first batch belongs to this customer arrives at 10020 and it includes the jobs 3 and 4, which incurs an inventory cost for job  $j_4$  at the first customer equals to  $(10 \times 2 = 20)$ . The second batch belongs to this customer arrives on time at 10050, which does not incur any inventory cost for the job  $j_7$ . The total cost for general model becomes equal to  $88 + 13 + 19 + 20 = 140$ .

#### 5.4.1.5 Summary of the evaluation of the quality of solutions

We observe that the inventory cost at the customers obtained in in the general model which equals to 45 is better than that obtained in the specific models which equals to 52, consistent with the delivery structure used in each model and the storage cost at the customers. The delivery cost in the general model which equals to 72 is better than that of the specific models which equals to 88, this difference is due to the efficiency of the heterogeneous transporters proposed in the general model. The total cost in the specific models is higher than the corresponding cost in the general model, and this is mainly due to the reduction in the inventory cost at the customers due to the efficient delivery structure proposed in the general model with the heterogeneous transporters.

### 5.4.2 Evaluation of the Computational Results

To test the performance of the computational time of the models discussed in this chapter we randomly generated a set of test problems and compared the computation times to the general purpose solver CPLEX. The characteristics of the case are listed in Table 5.4.2. Characteristics of orders to schedule differ by customers, transporter

capacity, quantity delivered, due date, transporter time, transporter cost and the storage cost at each customer. Three cases {A, B and C} are considered to test the proposed models. For each case, the number of products  $n$ , number of customers  $h$ , number of transporters  $nt$ , the round trip time  $\tau_{ht}$ , the transporter cost  $\eta_{ht}$  and the storage cost  $\beta_j$  at each customer are displayed in Table 5.4.2.

**Table 5.2:** Main characteristics of the test cases

Case	$n$	$h$	$nt$	$\beta_j$	$\eta_{ht}$	$\tau_{ht}$
A	5	3	3 or 4	[0.1; 0.5]	[10; 60]	[5; 50]
	10	3	3 or 4			
	15	3	3 or 4			
B	5	3	3 or 4	[1; 5]	[10; 60]	[5; 50]
	10	3	3 or 4			
	15	3	3 or 4			
C	5	3	3 or 4	[10; 50]	[10; 60]	[5; 50]
	10	3	3 or 4			
	15	3	3 or 4			

In the first class (A),  $\eta_{ht}$  is bigger than  $\beta_j$ , where  $\eta_{ht}$  and  $\beta_j$  are randomly generated from the uniform distribution with ranges [10, 60] and [0.1, 0.5], respectively. In the second class (B),  $\eta_{ht}$  are generated in the same way and with the same distribution with ranges [10, 60] and  $\beta_j$  of the first class are multiplied by 10, where  $\beta_j$  are randomly generated from the uniform distribution with ranges [1, 5]. In the third class (C), we kept the same value of  $\eta_{ht}$  and we changed  $\beta_j$  by multiplying the ranges of the first class by 100, where  $\beta_j$  are randomly generated from the uniform distribution with ranges [10.0, 50.0]. For each class under resolution the number of customers equal to 3, and each case is tested with 3 or 4 transporters. The results of the solution time obtained for each model can be seen in Table 5.3.

**Table 5.3:** Total computational time for the three classes

		Class A		
		10	15	20
(n)→		CpuT(s)	CpuT(s)	CpuT(s)
3	general model	5,6	30,6	250,4
	homo. Trans. without allotment	2,4	8,2	56,3
	homo. Trans. with allotment	1,2	4,9	25,7
	one trans. to each cust. ( $p$ equals to $h$ )	2,7	12,3	27,0
4	general model	50,8	650,2	2367,3
	homo. Trans. without allotment	33,7	236,4	995,8
	homo. Trans. with allotment	4,4	89,6	665,7
	one trans. to each cust. ( $p$ equals to $h$ )	6,9	112,3	407,1
		Class B		
		10	15	20
(p)↓		CpuT(s)	CpuT(s)	CpuT(s)
3	general model	339,4	>3600	>3600
	homo. Trans. without allotment	178,4	1568,4	>3600
	homo. Trans. with allotment	74,3	802,4	3301,3
	one trans. to each cust. ( $p$ equals to $h$ )	129,5	2174,2	>3600
4	general model	3190,5	>3600	>3600
	homo. Trans. without allotment	2675,3	>3600	>3600
	homo. Trans. with allotment	542,7	2390,7	>3600
	one trans. to each cust. ( $p$ equals to $h$ )	2550,6	>3600	>3600
		Class C		
		10	15	20
(p)↓		CpuT(s)	CpuT(s)	CpuT(s)
3	general model	727,4	>3600	>3600
	homo. Trans. without allotment	472,3	>3600	>3600
	homo. Trans. with allotment	257,4	>3600	>3600
	one trans. to each cust. ( $p$ equals to $h$ )	349,5	>3600	>3600
4	general model	3540,2	>3600	>3600
	homo. Trans. without allotment	3120,4	>3600	>3600
	homo. Trans. with allotment	1376,4	>3600	>3600
	one trans. to each cust. ( $p$ equals to $h$ )	3259,3	>3600	>3600

Table 5.3 shows the solution time obtained for each model. illustrate the computational time solution for each model with 3 customers, respectively. As can be seen, increasing the number of jobs in different models leads to increase the time of resolution. During the resolution, CPLEX runs out of memory and could not solve the instances of the general model more than 10 jobs in Case B, and the instances with 20 jobs in models (one trans. to each cust. ( $p$  equals to  $h$ )) and (homo. Trans. without allotment) in Case B with 3 Transporters. In the Case B with 4

Transporters, CPLEX could solve the instances with 10 and 15 jobs for the model of homo. Trans. with allotment. Then, in Case C CPLEX runs out of memory with  $n$  more than 10 jobs for all the models.

It is however interesting to see that an exact solution are obtained for the CPLEX solver to solve all the instances of different models in class A. In this class, the total storage cost at the customers  $\sum_{j=1}^n \beta_j(d_j - C_j)$ , which constitute the second part from the objective function (Equation 3.1), will be less than that of the total transporter cost  $\sum_{h=1}^m \eta_{ht}u_h$ , which constitute the first part from the objective function (Equation 3.1). In this case, the problem is the least complex to solve, because the vehicles are fully loaded according to the cheapness of the storage cost at the customers.

When the MIP model was tested it became evident that it suffered from a poor linear programming (LP) relaxation (Class B and C). The poor LP relaxation leads to an increased solution time since the CPLEX solver uses the LP relaxed solution as an internal lower bound for solving the problem. If the internal lower bound is poor, it can take long time to solve the proposed model. In the solution of the instances tested in Class B we noticed that, the increase in number of batches and the decrease of the size of products by batch in the optimal solution. In this case the solver solves only the instances with 10 and 15 jobs in the most instances but with a large time of resolution very near to one hour. Then in the third cases (Class C), we observe that CPLEX could solve instance with 10 jobs, then it runs out of memory with time of resolution more than one hour. In this case, we noticed that the vehicle is very lightly loaded.

The development of the time of resolution as the number of jobs increases is studied, for different Models in each Case (A, B and C) in (Table 5.3). The results show the fastness of the (homo. Trans. with allotment) model to give an optimal solutions for small size instances. As can be seen, if the number of jobs and transporters increase, the time of resolution will increase in each model.

## 5.5 Conclusion

This chapter describes four models to solve an integrated batching and scheduling delivery-inventory problem where a single supplier supplies a products to multiple customers with multi-transporter considerations. In the first model, a general model with heterogeneous transporters with different capacity transfer items to the customers without allotment. A Branch and Bound algorithm is developed to solve the general model and a Mixed-Integer-Programming is proposed to solve this model. In the second model, we studied the case of multi-homogeneous-transporter without allotment, where all the transporters used have the same capacity, and a Mixed-Integer-Programming is proposed to solve this model. Third we studied the model with homogeneous transporters with allotment, where the number of transporters

dependent of the demand of each customers, and a Mixed-Integer-Programming is proposed to solve this model.. Forth the last model, one transporter allotted to each customer with the consideration of the homogeneous transporters also, and a Mixed-Integer-Programming is proposed to solve this model.

Each transfer of batches incurs transportation cost and each product arrives at the customer before its due date incurs an inventory cost. Optimum solution techniques for all four models are presented while sensitivity analysis has also been performed to study the quality of solution of each method. In this chapter, we do not restrict the vendor to have either greater or less holding cost than everyone else in the system. The studied models in this chapter is also flexible in accommodating unequal transportation time and cost for each customer in the system, this costs dependent of the distance between the supplier and the customers. A numerical study performed on the four models presented here reveals that under similar circumstances, the first model is always more efficient in solving the integrated delivery-inventory system.



# 6 Conclusions and Future Extensions

## 6.1 Conclusions

This thesis presents some new models and resolution approaches for the batching and scheduling problem in multi-scenario supply chain management, as part of the development of a broader scheduling methodology. The primary objective of this work was to develop a general and integrated methodology for these complex, highly combinatorial problems. A real case-study from the central medical stores was used as test-bed in this research. Emphasis was given to the specific features of this industrial sector, involving a significant work to contextualize and determine how the batching and scheduling functions are performed. The key contributions of this thesis and recommendations for future research are summarized in what follows.

## 6.2 Summary of Contributions

Batching and Scheduling problems in supply chain have received considerable attention in the past decades due to their importance for the efficiency of operations. A variety of modeling approaches has appeared in the literature, introducing different types of formulations and involving multiple decisions and objectives. In general, there has been an effort to take into account the computational efficiency of the formulations, particularly when addressing large instances problems. Nevertheless, modeling, computational performance, and the integration of optimization methods in the real decision-making processes of companies, are still open issues that have been addressed in this study.

In general, the major contributions of this study can be summarized as follows:

In the first studied problem with single-supplier/multi-customer and one transporter used to serve the costumers separately, we have proposed some enhancements to the batching and the delivery of batches of the real-life studied in the central medical stores, which takes into account the different parameters related to the health care environment. Moreover, we have proposed an efficient genetic algorithm which is compared with a developed Branch and Bound algorithm and a Mixed Integer Programming model as an exact solutions. This study leads to two important advantages. First, the resolution of real world scheduling problems helped to focus

the research in the development of optimization models that can effectively be implemented. Thus, practical scheduling requirements have been discussed with the central medical stores and considered in the models whenever possible. Second, although the scheduling models and solution approaches developed were motivated by a case-study, they can be applied to other types of industries, as long as the scheduling problem has a similar structure.

In the second studied problem with single-supplier/single-customer and multiple homogeneous and heterogeneous transporters cases, we have assumed the batch-size-dependent delivery times and costs in order to take into account the loading and the unloading times of the transporters, this configuration has not been studied till now, as far as we know. In a second distinguishing feature for the same studied problem, we have developed a dynamic programming algorithm based on a dominance property, which is very efficient in the identical transporters case and compared it to a mixed-integer-programming model solved by CPLEX. Then an approach method consists of an efficient genetic algorithm is developed and compared to the proposed exact methods.

In the third studied problem with single-supplier/multi-customer with multiple transporters consideration, we have proposed a general model with multiple heterogeneous transporters available to serve the customer without any allotment, then we have proposed a model with allocated transporters to each customers. For this work, we have developed branch and bound algorithm in the general case, which is compared to a MIP model solved by CPLEX.

The study presented in this chapter may be extended in several directions. First, improving the efficiency of the proposed dynamic programming method in the general case, although it is already an efficient method, in order to solve larger problem instances. For that purpose, it is possible to use a lower bound by solving a similar problem with identical transporters case. Secondly, from both theoretical and practical point of view, there exist other possible objectives of interests to be considered, to this initial model in order to be closer to the industrial reality of the considered problem. The first extension could be to consider a different schedule between the production stage and the delivery stage, which means that products may wait between the end of the production stage and the start of the delivery. The second one is to consider the multi-products case when products have different volumes. The methods presented in this chapter form a foundation for modeling and solving these extensions.

The outputs of this thesis are significant contributions for better modeling scheduling problems and for solving real industrial problems. They can also be viewed as a sound basis for the development of improved and more sophisticated decision support tools for dealing with those problems.



## 6.3 Possible Future Extensions

The integrated models presented here are subject to certain restrictions, which can be considered in future research. By relaxing some of those restrictions, the problem will become more complex but more realistic. However, in order to enhance the applicability and model performance, the following extensions are suggested:

The first research opportunity is related to the single-supplier/multi-customer with one transporter and could be advanced by allowing the vehicle routing with integrated delivery and storage cost. This work could be integrated into the setup time and cost into the production stage, and the volume of products into the delivery stage.

The second research opportunity is concerned with the development of single-supplier single-customer with multiple transporters case by improving the efficiency of the proposed dynamic programming method in the general case, although it is already an efficient method, in order to solve larger problem instances. For that purpose, it is possible to use a lower bound by solving a similar problem with identical transporters case. Another possible extension for this work could be related to the theoretical and practical point of view, where there exist other possible objectives of interests to be considered. The first consideration could be related to a different schedule between the production stage and the delivery stage, which means that products may wait between the end of the production stage and the start of the delivery. The second one is to consider the multi-products case when products have different volumes. The methods presented in this chapter form a foundation for modeling and solving these extensions.

The third research opportunity is associated to the single-supplier/multi-customer with multiple transporters, where we aim to develop the general case which is without allocated transporters according to an industrial study by added some new constraints of the general model. Moreover, we aim to develop an efficient approach algorithm to solve a medium and large instances due to the large time of resolution found by the proposed exact methods.



# Bibliography

- [Abad and Aggarwal, 2005] Abad, P. and Aggarwal, V. (2005). Incorporating transport cost in the lot size and pricing decisions with downward sloping demand. *International Journal of Production Economics*, 95(3):297–305.
- [Afentakis et al., 1984] Afentakis, P., Gavish, B., and Karmarkar, U. (1984). Computationally efficient optimal solutions to the lot-sizing problem in multistage assembly systems. *Management Science*, 30(2):222–239.
- [Ajian and Lair, 2008] Ajian, N. and Lair, M. (2008). *An integrated model for optimising manufacturing and distribution network scheduling*. PhD thesis, University of South Australia.
- [Anily, 1994] Anily, S. (1994). The general multi-retailer eoq problem with vehicle routing costs. *European Journal of Operational Research*, 79(3):451–473.
- [Anily and Bramel, 2004] Anily, S. and Bramel, J. (2004). An asymptotic 98.5% effective lower bound on fixed partition policies for the inventory-routing problem. *Discrete Applied Mathematics*, 145(1):22–39.
- [Anily and Federgruen, 1990a] Anily, S. and Federgruen, A. (1990a). A class of euclidean routing problems with general route cost functions. *Mathematics of Operations Research*, 15(2):268–285.
- [Anily and Federgruen, 1990b] Anily, S. and Federgruen, A. (1990b). One warehouse multiple retailer systems with vehicle routing costs. *Management Science*, 36(1):92–114.
- [Anily and Federgruen, 1991a] Anily, S. and Federgruen, A. (1991a). Rejoinder to comments on one-warehouse multiple retailer systems with vehicle routing costs. *Management Science*, 37(11):1497–1499.
- [Anily and Federgruen, 1991b] Anily, S. and Federgruen, A. (1991b). Structured partitioning problems. *Operations Research*, 39(1):130–149.
- [Anily and Federgruen, 1993] Anily, S. and Federgruen, A. (1993). Two-echelon distribution systems with vehicle routing costs and central inventories. *Operations Research*, 41(1):37–47.
- [Aptel and Pourjalali, 2001] Aptel, O. and Pourjalali, H. (2001). Improving activities and decreasing costs of logistics in hospitals: a comparison of us and french hospitals. *The International Journal of Accounting*, 36(1):65–90.

- [Archetti et al., 2012] Archetti, C., Bertazzi, L., Hertz, A., and Speranza, M. G. (2012). A hybrid heuristic for an inventory routing problem. *INFORMS Journal on Computing*, 24(1):101–116.
- [Archetti et al., 2007] Archetti, C., Bertazzi, L., Laporte, G., and Speranza, M. G. (2007). A branch-and-cut algorithm for a vendor-managed inventory-routing problem. *Transportation Science*, 41(3):382–391.
- [Archetti et al., 2011] Archetti, C., Bertazzi, L., Paletta, G., and Speranza, M. G. (2011). Analysis of the maximum level policy in a production-distribution system. *Computers & Operations Research*, 38(12):1731–1746.
- [Banerjee and Banerjee, 1994] Banerjee, A. and Banerjee, S. (1994). A coordinated order-up-to inventory control policy for a single supplier and multiple buyers using electronic data interchange. *International Journal of Production Economics*, 35:85–91.
- [Baptiste, 2000] Baptiste, P. (2000). Batching identical jobs. *Mathematical Methods of Operations Research*, 52(3):355–367.
- [Bard and Nananukul, 2009a] Bard, J. F. and Nananukul, N. (2009a). Heuristics for a multiperiod inventory routing problem with production decisions. *Computers & Industrial Engineering*, 57(3):713–723.
- [Bard and Nananukul, 2009b] Bard, J. F. and Nananukul, N. (2009b). The integrated production–inventory–distribution–routing problem. *Journal of Scheduling*, 12(3):257–280.
- [Bard and Nananukul, 2010] Bard, J. F. and Nananukul, N. (2010). A branch-and-price algorithm for an integrated production and inventory routing problem. *Computers & Operations Research*, 37(12):2202–2217.
- [Battini et al., 2010] Battini, D., Gunasekaran, A., Faccio, M., Persona, A., and Sgarbossa, F. (2010). Consignment stock inventory model in an integrated supply chain. *International Journal of Production Research*, 48(2):477–500.
- [Baumol and Vinod, 1970] Baumol, W. J. and Vinod, H. D. (1970). An inventory theoretic model of freight transport demand. *Management science*, 16(7):413–421.
- [Ben-Daya et al., 2008] Ben-Daya, M., Darwish, M., and Ertogral, K. (2008). The joint economic lot sizing problem: Review and extensions. *European Journal of Operational Research*, 185(2):726–742.
- [Bertazzi et al., 2013] Bertazzi, L., Bosco, A., Guerriero, F., and Lagana, D. (2013). A stochastic inventory routing problem with stock-out. *Transportation Research Part C: Emerging Technologies*, 27:89–107.
- [Bertazzi et al., 2005] Bertazzi, L., Paletta, G., and Speranza, M. G. (2005). Minimizing the total cost in an integrated vendor-managed inventory system. *Journal of heuristics*, 11(5-6):393–419.

- [Bertazzi et al., 2008] Bertazzi, L., Savelsbergh, M., and Speranza, M. G. (2008). Inventory routing. In *The vehicle routing problem: latest advances and new challenges*, pages 49–72. Springer.
- [Bertazzi et al., 1997] Bertazzi, L., Speranza, M. G., and Ukovich, W. (1997). Minimization of logistic costs with given frequencies. *Transportation Research Part B: Methodological*, 31(4):327–340.
- [Bhatnagar et al., 1993] Bhatnagar, R., Chandra, P., and Goyal, S. K. (1993). Models for multi-plant coordination. *European Journal of Operational Research*, 67(2):141–160.
- [Billington et al., 1994] Billington, P., Blackburn, J., Maes, J., Millen, R., and Van Wassenhove, L. N. (1994). Multi-item lotsizing in capacitated multi-stage serial systems. *IIE transactions*, 26(2):12–18.
- [Blackburn and Millen, 1982] Blackburn, J. D. and Millen, R. A. (1982). Improved heuristics for multi-stage requirements planning systems. *Management Science*, 28(1):44–56.
- [Blackburn and Millen, 1984] Blackburn, J. D. and Millen, R. A. (1984). Simultaneous lot-sizing and capacity planning in multi-stage assembly processes. *European Journal of Operational Research*, 16(1):84–93.
- [Blumenfeld et al., 1986] Blumenfeld, D. E., Burns, L. D., and Daganzo, C. F. (1986). Synchronizing production and transportation schedules. *Transportation Research Part B: Methodological*, 25(1):23–37.
- [Blumenfeld et al., 1991] Blumenfeld, D. E., Burns, L. D., and Daganzo, C. F. (1991). Synchronizing production and transportation schedules. *Transportation Research Part B: Methodological*, 25(1):23–37.
- [Blumenfeld et al., 1987] Blumenfeld, D. E., Burns, L. D., Daganzo, C. F., Frick, M. C., and Hall, R. W. (1987). Reducing logistics costs at general motors. *Interfaces*, 17(1):26–47.
- [Blumenfeld et al., 1985] Blumenfeld, D. E., Burns, L. D., Diltz, J. D., and Daganzo, C. F. (1985). Analyzing trade-offs between transportation, inventory and production costs on freight networks. *Transportation Research Part B: Methodological*, 19(5):361–380.
- [Bogaschewsky et al., 2001] Bogaschewsky, R. W., Buscher, U. D., and Lindner, G. (2001). Optimizing multi-stage production with constant lot size and varying number of unequal sized batches. *Omega*, 29(2):183–191.
- [Bookbinder and Heath, 1988] Bookbinder, J. H. and Heath, D. B. (1988). Replenishment analysis in distribution requirements planning\*. *Decision Sciences*, 19(3):477–489.
- [Bookbinder and Koch, 1990] Bookbinder, J. H. and Koch, L. A. (1990). Production planning for mixed assembly/arborescent systems. *Journal of Operations Management*, 9(1):7–23.

- [Borade and Sweeney, 2014] Borade, A. B. and Sweeney, E. (2014). Decision support system for vendor managed inventory supply chain: a case study. *International Journal of Production Research*, (ahead-of-print):1–30.
- [Braglia and Zavanella, 2003] Braglia, M. and Zavanella, L. (2003). Modelling an industrial strategy for inventory management in supply chains: the 'consignment stock' case. *International Journal of Production Research*, 41(16):3793–3808.
- [Brahimi et al., 2006] Brahimi, N., Dauzere-Peres, S., Najid, N. M., and Nordli, A. (2006). Single item lot sizing problems. *European Journal of Operational Research*, 168(1):1–16.
- [Burns et al., 1985] Burns, L. D., Hall, R. W., Blumenfeld, D. E., and Daganzo, C. F. (1985). Distribution strategies that minimize transportation and inventory costs. *Operations Research*, 33(3):469–490.
- [Bylka, 2013] Bylka, S. (2013). Non-cooperative consignment stock strategies for management in supply chain. *International Journal of Production Economics*, 143(2):424–433.
- [Bylka and Górný, 2015] Bylka, S. and Górný, P. (2015). The consignment stock of inventories in coordinated model with generalized policy. *Computers & Industrial Engineering*.
- [Chandra and Fisher, 1994] Chandra, P. and Fisher, M. L. (1994). Coordination of production and distribution planning. *European Journal of Operational Research*, 72(3):503–517.
- [Charnsirisakskul et al., 2006] Charnsirisakskul, K., Griffin, P. M., and Keskinocak, P. (2006). Pricing and scheduling decisions with leadtime flexibility. *European Journal of Operational Research*, 171(1):153–169.
- [Chen, 2010] Chen, Z.-L. (2010). Integrated production and outbound distribution scheduling: Review and extensions. *Operations Research*, 58(1):130–148.
- [Cheng and Gen, 1997] Cheng, R. and Gen, M. (1997). Genetic algorithms and engineering design. *New York*.
- [Cheng and Gupta, 1989] Cheng, T. and Gupta, M. (1989). Survey of scheduling research involving due date determination decisions. *European Journal of Operational Research*, 38(2):156–166.
- [Chiu et al., 2013] Chiu, S., Liu, K., Lee, C., and Chiu, Y. (2013). A single-producer multi-retailer integrated inventory system with scrap in production. *Res. J. Appl. Sci. Eng. Technol*, 5(4):1154–1159.
- [Chuang et al., 2013] Chuang, C.-J., Ho, C.-H., Ouyang, L.-Y., and Wu, C.-W. (2013). An integrated inventory model with order-size-dependent trade credit and quality improvement. *Procedia Computer Science*, 17:365–372.
- [Clark and Clark, 2000] Clark, A. R. and Clark, S. J. (2000). Rolling-horizon lot-sizing when set-up times are sequence-dependent. *International Journal of Production Research*, 38(10):2287–2307.

- [Coelho et al., 2012a] Coelho, L. C., Cordeau, J.-F., and Laporte, G. (2012a). Consistency in multi-vehicle inventory-routing. *Transportation Research Part C: Emerging Technologies*, 24:270–287.
- [Coelho et al., 2012b] Coelho, L. C., Cordeau, J.-F., and Laporte, G. (2012b). The inventory-routing problem with transshipment. *Computers & Operations Research*, 39(11):2537–2548.
- [Coelho et al., 2013] Coelho, L. C., Cordeau, J.-F., and Laporte, G. (2013). Thirty years of inventory routing. *Transportation Science*, 48(1):1–19.
- [Coelho and Laporte, 2013] Coelho, L. C. and Laporte, G. (2013). A branch-and-cut algorithm for the multi-product multi-vehicle inventory-routing problem. *International Journal of Production Research*, 51(23-24):7156–7169.
- [Coelho and Laporte, 2015] Coelho, L. C. and Laporte, G. (2015). Classification, models and exact algorithms for multi-compartment delivery problems. *European Journal of Operational Research*, 242(3):854–864.
- [Crowston and Wagner, 1973] Crowston, W. B. and Wagner, M. H. (1973). Dynamic lot size models for multi-stage assembly systems. *Management Science*, 20(1):14–21.
- [Drexl and Kimms, 1997] Drexl, A. and Kimms, A. (1997). Lot sizing and scheduling survey and extensions. *European Journal of operational research*, 99(2):221–235.
- [Elmahi et al., 2006] Elmahi, I., Grunder, O., and Elmoudni, A. (2006). A modelling-optimization approach for discrete event systems using the (max,+) algebra and genetic algorithms. *International Journal of Innovative Computing, Information and Control*, 2(4):771–788.
- [Ertogral et al., 2007] Ertogral, K., Darwish, M., and Ben-Daya, M. (2007). Production and shipment lot sizing in a vendor–buyer supply chain with transportation cost. *European Journal of Operational Research*, 176(3):1592–1606.
- [Fahimnia et al., 2013] Fahimnia, B., Farahani, R. Z., Marian, R., and Luong, L. (2013). A review and critique on integrated production–distribution planning models and techniques. *Journal of Manufacturing Systems*, 32(1):1–19.
- [Fleischmann, 1990] Fleischmann, B. (1990). The discrete lot-sizing and scheduling problem. *European Journal of Operational Research*, 44(3):337–348.
- [Fu et al., 2010] Fu, B., Huo, Y., and Zhao, H. (2010). Coordinated scheduling of production and delivery with production window and delivery capacity constraints. In *Algorithmic Aspects in Information and Management*, pages 141–149. Springer.
- [Fumero and Vercellis, 1999] Fumero, F. and Vercellis, C. (1999). Synchronized development of production, inventory, and distribution schedules. *Transportation science*, 33(3):330–340.

- [Glock, 2012] Glock, C. H. (2012). Coordination of a production network with a single buyer and multiple vendors. *International Journal of Production Economics*, 135(2):771–780.
- [Goldberg, 1989] Goldberg, D. (1989). Genetic algorithms in search, optimization, and machine learning.
- [Gordon et al., 2002] Gordon, V., Proth, J.-M., and Chu, C. (2002). A survey of the state-of-the-art of common due date assignment and scheduling research. *European Journal of Operational Research*, 139(1):1–25.
- [Gordon et al., 2012] Gordon, V., Strusevich, V., and Dolgui, A. (2012). Scheduling with due date assignment under special conditions on job processing. *Journal of Scheduling*, 15(4):447–456.
- [Goyal, 1976] Goyal, S. (1976). Note-note on "manufacturing cycle time determination for a multi-stage economic production quantity model". *Management Science*, 23(3):332–333.
- [Goyal, 1977] Goyal, S. (1977). An integrated inventory model for a single supplier-single customer problem. *The International Journal of Production Research*, 15(1):107–111.
- [Goyal, 1995] Goyal, S. (1995). A one-vendor multi-buyer integrated inventory model: A comment. *European journal of operational research*, 82(1):209–210.
- [Goyal, 2000] Goyal, S. (2000). On improving the single-vendor single-buyer integrated production inventory model with a generalized policy. *European Journal of Operational Research*, 125(2):429–430.
- [Goyal, 1988] Goyal, S. K. (1988). A joint economic-lot-size model for purchaser and vendor: A comment. *Decision sciences*, 19(1):236–241.
- [Goyal and Nebebe, 2000] Goyal, S. K. and Nebebe, F. (2000). Determination of economic production–shipment policy for a single-vendor–single-buyer system. *European Journal of Operational Research*, 121(1):175–178.
- [Grunder, 2010] Grunder, O. (2010). Lot sizing, delivery and scheduling of identical jobs in a single-stage supply chain. *International Journal of Innovative Computing, Information and Control*, 6(8):3657–3668.
- [Hahm and Yano, 1992] Hahm, J. and Yano, C. A. (1992). The economic lot and delivery scheduling problem: the single item case. *International Journal of Production Economics*, 28(2):235–252.
- [Hahm and Yano a, 1995] Hahm, J. and Yano a, C. A. (1995). The economic lot and delivery scheduling problem: the common cycle case. *IIE transactions*, 27(2):113–125.
- [Hahm and Yano b, 1995] Hahm, J. and Yano b, C. A. (1995). The economic lot and delivery scheduling problem: Models for nested schedules. *IIE transactions*, 27(2):126–139.



- [Hahm and Yano c, 1995] Hahm, J. and Yano c, C. A. (1995). The economic lot and delivery scheduling problem: Powers of two policies. *Transportation Science*, 29(3):222–241.
- [Hall, 1991] Hall, R. W. (1991). Comments on one-warehouse multiple retailer systems with vehicle routing costs. *Management Science*, 37(11):1496–1497.
- [Hariga et al., 2014] Hariga, M., Hassini, E., and Ben-Daya, M. (2014). A note on generalized single-vendor multi-buyer integrated inventory supply chain models with better synchronization. *International Journal of Production Economics*, 154:313–316.
- [Harris, 1913] Harris, F. W. (1913). How many parts to make at once. *Operations Research*, 38(6):947–950.
- [Herer and Levy, 1997] Herer, Y. T. and Levy, R. (1997). The metered inventory routing problem, an integrative heuristic algorithm. *International Journal of Production Economics*, 51(1):69–81.
- [Hill and Omar, 2006] Hill, R. and Omar, M. (2006). Another look at the single-vendor single-buyer integrated production-inventory problem. *International Journal of Production Research*, 44(4):791–800.
- [Hill, 1997] Hill, R. M. (1997). The single-vendor single-buyer integrated production-inventory model with a generalised policy. *European Journal of Operational Research*, 97(3):493–499.
- [Hill, 1999] Hill, R. M. (1999). The optimal production and shipment policy for the single-vendor single-buyer integrated production-inventory problem. *International Journal of Production Research*, 37(11):2463–2475.
- [Holland, 1975] Holland, J. H. (1975). Adaptation in natural and artificial systems. *Ann Arbor*.
- [Hoque, 2008] Hoque, M. (2008). Synchronization in the single-manufacturer multi-buyer integrated inventory supply chain. *European Journal of Operational Research*, 188(3):811–825.
- [Hoque, 2009] Hoque, M. (2009). An alternative optimal solution technique for a single-vendor single-buyer integrated production inventory model. *International Journal of Production Research*, 47(15):4063–4076.
- [Hoque, 2011a] Hoque, M. (2011a). Generalized single-vendor multi-buyer integrated inventory supply chain models with a better synchronization. *International Journal of Production Economics*, 131(2):463–472.
- [Hoque, 2011b] Hoque, M. (2011b). An optimal solution technique to the single-vendor multi-buyer integrated inventory supply chain by incorporating some realistic factors. *European Journal of Operational Research*, 215(1):80–88.
- [Hsu, 2013] Hsu, C. J. (2013). Scheduling to a common due date on unrelated parallel-machine with deteriorating jobs. In *Advanced Materials Research*, volume 605, pages 521–527. Trans Tech Publ.

- [Jat and Yang, 2009] Jat, S. and Yang, S. (2009). A guided search genetic algorithm for the university course timetabling problem. In *The 4th Multidisciplinary International Scheduling Conference: Theory and Applications (MISTA 2009), Dublin, Ireland: 180-191, 10-12 Aug 2009*.
- [Jha and Shanker, 2013] Jha, J. and Shanker, K. (2013). Single-vendor multi-buyer integrated production-inventory model with controllable lead time and service level constraints. *Applied Mathematical Modelling*, 37(4):1753–1767.
- [Kaminsky and Kaya, 2008] Kaminsky, P. and Kaya, O. (2008). Scheduling and due-date quotation in a make-to-order supply chain. *Naval Research Logistics (NRL)*, 55(5):444–458.
- [Karp, 1972] Karp, R. M. (1972). *Reducibility among combinatorial problems*. Springer.
- [Keskinocak et al., 2001] Keskinocak, P., Ravi, R., and Tayur, S. (2001). Scheduling and reliable lead-time quotation for orders with availability intervals and lead-time sensitive revenues. *Management Science*, 47(2):264–279.
- [Khan et al., 2014] Khan, M., Jaber, M. Y., and Ahmad, A.-R. (2014). An integrated supply chain model with errors in quality inspection and learning in production. *Omega*, 42(1):16 – 24.
- [Khanra et al., 2011] Khanra, S., Ghosh, S., and Chaudhuri, K. (2011). An eqo model for a deteriorating item with time dependent quadratic demand under permissible delay in payment. *Applied Mathematics and Computation*, 218(1):1–9.
- [Khouja, 2000] Khouja, M. (2000). The economic lot and delivery scheduling problem: common cycle, rework, and variable production rate. *IIE Transactions*, 32(8):715–725.
- [Khouja, 2003] Khouja, M. (2003). Synchronization in supply chains: implications for design and management. *Journal of the Operational Research Society*, 54(9):984–994.
- [Koc et al., 2013] Koc, U., Toptal, A., and Sabuncuoglu, I. (2013). A class of joint production and transportation planning problems under different delivery policies. *Operations Research Letters*, 41(1):54–60.
- [Lauff and Werner, 2004] Lauff, V. and Werner, F. (2004). Scheduling with common due date, earliness and tardiness penalties for multimachine problems: A survey. *Mathematical and Computer Modelling*, 40(5):637–655.
- [Leung and Chen, 2013] Leung, J. Y.-T. and Chen, Z.-L. (2013). Integrated production and distribution with fixed delivery departure dates. *Operations Research Letters*, 41(3):290–293.
- [Li and Haimes, 1989] Li, D. and Haimes, Y. (1989). Multiobjective dynamic-programming-the state-of-the-art. *Control-Theory and Advanced Technology*, 5(4):471–483.

- [Liu et al., 2007] Liu, L., Parlar, M., and Zhu, S. X. (2007). Pricing and lead time decisions in decentralized supply chains. *Management Science*, 53(5):713–725.
- [Lu, 1995] Lu, L. (1995). A one-vendor multi-buyer integrated inventory model. *European Journal of Operational Research*, 81(2):312–323.
- [Maxwell and Muckstadt, 1979] Maxwell, W. L. and Muckstadt, J. A. (1979). Coordination of production schedules with shipping schedules. revision. Technical report, DTIC Document.
- [Maxwell and Muckstadt, 1985] Maxwell, W. L. and Muckstadt, J. A. (1985). Establishing consistent and realistic reorder intervals in production-distribution systems. *Operations Research*, 33(6):1316–1341.
- [Mehta and Uzsoy, 1998] Mehta, S. V. and Uzsoy, R. (1998). Minimizing total tardiness on a batch processing machine with incompatible job families. *IIE transactions*, 30(2):165–178.
- [Merzouk et al., 2008] Merzouk, S., Grunder, O., and El Bagdouri, M. (2008). The holding and the transportation costs optimization in a simple supply chain: The multiple transporters case.
- [Meyr, 2000] Meyr, H. (2000). Simultaneous lotsizing and scheduling by combining local search with dual reoptimization. *European Journal of Operational Research*, 120(2):311–326.
- [Mingers and Brocklesby, 1997] Mingers, J. and Brocklesby, J. (1997). Multi-methodology: towards a framework for mixing methodologies. *Omega*, 25(5):489–509.
- [Mishra and Sahab Singh, 2011] Mishra, V. K. and Sahab Singh, L. (2011). Deteriorating inventory model for time dependent demand and holding cost with partial backlogging. *International Journal of Management Science and Engineering Management*, 6(4):267–271.
- [Moin et al., 2011] Moin, N. H., Salhi, S., and Aziz, N. (2011). An efficient hybrid genetic algorithm for the multi-product multi-period inventory routing problem. *International Journal of Production Economics*, 133(1):334–343.
- [Muckstadt and Roundy, 1987] Muckstadt, J. A. and Roundy, R. O. (1987). Multi-item, one-warehouse, multi-retailer distribution systems. *Management Science*, 33(12):1613–1621.
- [Omar, 2009] Omar, M. (2009). An integrated equal-lots policy for shipping a vendors final production batch to a single buyer under linearly decreasing demand. *International Journal of Production Economics*, 118(1):185–188.
- [Pal et al., 2015] Pal, S., Mahapatra, G., and Samanta, G. (2015). A production inventory model for deteriorating item with ramp type demand allowing inflation and shortages under fuzziness. *Economic Modelling*, 46:334–345.

- [Pan and Yang, 2002] Pan, J. C.-H. and Yang, J.-S. (2002). A study of an integrated inventory with controllable lead time. *International Journal of Production Research*, 40(5):1263–1273.
- [Pandey et al., 2007] Pandey, A., Masin, M., and Prabhu, V. (2007). Adaptive logistic controller for integrated design of distributed supply chains. *Journal of Manufacturing Systems*, 26(2):108–115.
- [Panwalkar et al., 1982] Panwalkar, S., Smith, M., and Seidmann, A. (1982). Common due date assignment to minimize total penalty for the one machine scheduling problem. *Operations research*, 30(2):391–399.
- [Parija and Sarker, 1999] Parija, G. R. and Sarker, B. R. (1999). Operations planning in a supply chain system with fixed-interval deliveries of finished goods to multiple customers. *IIE transactions*, 31(11):1075–1082.
- [Piya, 2015] Piya, S. (2015). Dealing with customers enquiries simultaneously under contingent situation. *International Journal of Industrial Engineering Computations*, 6(3):391–404.
- [Quadt, 2004] Quadt, D. (2004). *Lot-sizing and scheduling for flexible flow lines*, volume 546. Springer Science & Business Media.
- [Raa and Aghezzaf, 2009] Raa, B. and Aghezzaf, E.-H. (2009). A practical solution approach for the cyclic inventory routing problem. *European Journal of Operational Research*, 192(2):429–441.
- [Ramkumar et al., 2012] Ramkumar, N., Subramanian, P., Narendran, T., and Ganesh, K. (2012). Mixed integer linear programming model for multi-commodity multi-depot inventory routing problem. *OPSEARCH*, 49(4):413–429.
- [Rasti-Barzoki et al., 2013] Rasti-Barzoki, M., Hejazi, S. R., and Mazdeh, M. M. (2013). A branch and bound algorithm to minimize the total weighed number of tardy jobs and delivery costs. *Applied Mathematical Modelling*, 37(7):4924–4937.
- [Relvas et al., 2013] Relvas, S., Magatao, S. N. B., Barbosa-Povoa, A. P. F., and Jr., F. N. (2013). Integrated scheduling and inventory management of an oil products distribution system. *Omega*, 41(6):955 – 968.
- [Roundy, 1985] Roundy, R. (1985). 98%-effective integer-ratio lot-sizing for one-warehouse multi-retailer systems. *Management science*, 31(11):1416–1430.
- [Sajadieh et al., 2010] Sajadieh, M. S., Thorstenson, A., and Jokar, M. R. A. (2010). An integrated vendor–buyer model with stock-dependent demand. *Transportation Research Part E: Logistics and Transportation Review*, 46(6):963–974.
- [Salomon, 1991] Salomon, M. (1991). *Deterministic lotsizing models for production planning*, volume 355. Springer.
- [Sarker and Parija, 1994] Sarker, B. R. and Parija, G. R. (1994). An optimal batch size for a production system operating under a fixed-quantity, periodic delivery policy. *Journal of the Operational Research Society*, pages 891–900.

- [Sarker and Parija, 1996] Sarker, B. R. and Parija, G. R. (1996). Optimal batch size and raw material ordering policy for a production system with a fixed-interval, lumpy demand delivery system. *European Journal of Operational Research*, 89(3):593–608.
- [Sarker and Khan, 1999] Sarker, R. A. and Khan, L. R. (1999). An optimal batch size for a production system operating under periodic delivery policy. *Computers & Industrial Engineering*, 37(4):711–730.
- [Sarmiento and Nagi, 1999] Sarmiento, A. M. and Nagi, R. (1999). A review of integrated analysis of production–distribution systems. *IIE transactions*, 31(11):1061–1074.
- [Seidmann et al., 1981] Seidmann, A., Panwalkar, S., and Smith, M. (1981). Optimal assignment of due-dates for a single processor scheduling problem. *The International Journal Of Production Research*, 19(4):393–399.
- [Seidmann and Smith, 1981] Seidmann, A. and Smith, M. L. (1981). Due date assignment for production systems. *Management Science*, 27(5):571–581.
- [Seyedhosseini and Ghoreyshi, 2014] Seyedhosseini, S. and Ghoreyshi, S. (2014). An integrated model for production and distribution planning of perishable products with inventory and routing considerations. *Mathematical Problems in Engineering*, 2014.
- [Simpson, 1999] Simpson, N. (1999). Multiple level production planning in rolling horizon assembly environments. *European Journal of Operational Research*, 114(1):15–28.
- [Simpson and Erenguc, 1998] Simpson, N. C. and Erenguc, S. S. (1998). Improved heuristic methods for multiple stage production planning. *Computers & operations research*, 25(7):611–623.
- [Sindhuchao et al., 2005] Sindhuchao, S., Romeijn, H. E., Akçali, E., and Boondiskulchok, R. (2005). An integrated inventory-routing system for multi-item joint replenishment with limited vehicle capacity. *Journal of Global Optimization*, 32(1):93–118.
- [Spears et al., 1993] Spears, W. M., De Jong, K. A., Bäck, T., Fogel, D. B., and De Garis, H. (1993). An overview of evolutionary computation. In *Machine Learning: ECML-93*, pages 442–459. Springer.
- [Stecke and Zhao, 2007] Stecke, K. E. and Zhao, X. (2007). Production and transportation integration for a make-to-order manufacturing company with a commit-to-delivery business mode. *Manufacturing & Service Operations Management*, 9(2):206–224.
- [Stefansson et al., 2011] Stefansson, H., Sigmarsdottir, S., Jensson, P., and Shah, N. (2011). Discrete and continuous time representations and mathematical models for large production scheduling problems: A case study from the pharmaceutical industry. *European Journal of Operational Research*, 215(2):383–392.

- [Steiner and Zhang, 2007] Steiner, G. and Zhang, R. (2007). Minimizing the weighted number of late jobs with batch setup times and delivery costs on a single machine. *Multiprocessor Scheduling*, 85.
- [Steiner and Zhang, 2011] Steiner, G. and Zhang, R. (2011). Minimizing the weighted number of tardy jobs with due date assignment and capacity-constrained deliveries. *Annals of Operations Research*, 191(1):171–181.
- [Sterna, 2011] Sterna, M. (2011). A survey of scheduling problems with late work criteria. *Omega*, 39(2):120–129.
- [Swami et al., ] Swami, M. A., Singh, S., Pareek, S., and Yadav, A. S. Inventory policies for deteriorating item with stock dependent demand and variable holding costs under permissible delay in payment.
- [Szendrovits, 1975] Szendrovits, A. Z. (1975). Manufacturing cycle time determination for a multi-stage economic production quantity model. *Management Science*, 22(3):298–308.
- [Szendrovits and Drezner, 1980] Szendrovits, A. Z. and Drezner, Z. (1980). Optimizing multi-stage production with constant lot size and varying numbers of batches. *Omega*, 8(6):623–629.
- [Thomas and Griffin, 1996] Thomas, D. J. and Griffin, P. M. (1996). Coordinated supply chain management. *European journal of operational research*, 94(1):1–15.
- [Tischer, 2007] Tischer, S. (2007). *Lotsizing and scheduling problems*. PhD thesis, uniwiien.
- [Transchel et al., 2011] Transchel, S., Minner, S., Kallrath, J., Löhndorf, N., and Eberhard, U. (2011). A hybrid general lot-sizing and scheduling formulation for a production process with a two-stage product structure. *International Journal of Production Research*, 49(9):2463–2480.
- [Uzsoy, 1995] Uzsoy, R. (1995). Scheduling batch processing machines with incompatible job families. *International Journal of Production Research*, 33(10):2685–2708.
- [Van Ooijen and Bertrand, 2001] Van Ooijen, H. and Bertrand, J. (2001). Economic due-date setting in job-shops based on routing and workload dependent flow time distribution functions. *International Journal of Production Economics*, 74(1):261–268.
- [Viswanathan, 1998] Viswanathan, S. (1998). Optimal strategy for the integrated vendor-buyer inventory model. *European journal of operational research*, 105(1):38–42.
- [Viswanathan and Mathur, 1997] Viswanathan, S. and Mathur, K. (1997). Integrating routing and inventory decisions in one-warehouse multiretailer multiproduct distribution systems. *Management Science*, 43(3):294–312.

- [Viswanathan and Piplani, 2001] Viswanathan, S. and Piplani, R. (2001). Coordinating supply chain inventories through common replenishment epochs. *European Journal of Operational Research*, 129(2):277–286.
- [Wagner and Whitin, 1958] Wagner, H. M. and Whitin, T. M. (1958). Dynamic version of the economic lot size model. *Management science*, 5(1):89–96.
- [Wang, 2012] Wang, D. (2012). *Integrated Scheduling of Production and Transportation Operations with Stage-dependent Inventory Costs and Due Dates Considerations*. PhD thesis, Université de Technologie de Belfort-Montbéliard.
- [Wang et al., 2013] Wang, D., Grunder, O., and El Moudni, A. (2013). Single-item production–delivery scheduling problem with stage-dependent inventory costs and due-date considerations. *International Journal of Production Research*, 51(3):828–846.
- [Wang and Lee, 2005] Wang, H. and Lee, C.-Y. (2005). Production and transport logistics scheduling with two transport mode choices. *Naval Research Logistics (NRL)*, 52(8):796–809.
- [Wang et al., 2011] Wang, J., Chang, Q., Xiao, G., Wang, N., and Li, S. (2011). Data driven production modeling and simulation of complex automobile general assembly plant. *Computers in Industry*, 62(7):765–775.
- [Zangwill, 1969] Zangwill, W. I. (1969). A backlogging model and a multi-echelon model of a dynamic economic lot size production system—a network approach. *Management Science*, 15(9):506–527.
- [Zanoni and Grubbstrom, 2004] Zanoni, S. and Grubbstrom, R. W. (2004). A note on an industrial strategy for stock management in supply chains: modelling and performance evaluation. *International Journal of Production Research*, 42(20):4421–4426.
- [Zanoni et al., 2012] Zanoni, S., Jaber, M. Y., and Zavanella, L. E. (2012). Vendor managed inventory (vmi) with consignment considering learning and forgetting effects. *International Journal of Production Economics*, 140(2):721–730.
- [Zavanella and Zanoni, 2009] Zavanella, L. and Zanoni, S. (2009). A one-vendor multi-buyer integrated production-inventory model: The ‘consignment stock’ case. *International Journal of Production Economics*, 118(1):225–232.
- [Zhao et al., 2007] Zhao, Q.-H., Wang, S.-Y., and Lai, K. K. (2007). A partition approach to the inventory/routing problem. *European Journal of Operational Research*, 177(2):786–802.
- [Zhou and Wang, 2007] Zhou, Y.-W. and Wang, S.-D. (2007). Optimal production and shipment models for a single-vendor–single-buyer integrated system. *European Journal of Operational Research*, 180(1):309–328.
- [Zorzini et al., 2008] Zorzini, M., Corti, D., and Pozzetti, A. (2008). Due date (dd) quotation and capacity planning in make-to-order companies: Results from an

empirical analysis. *International Journal of Production Economics*, 112(2):919–933.