OPERATIONS SCHEDULING

WITH RENEWABLE AND NON-RENEWABLE RESOURCES

By

SHENGBIN WANG

A Dissertation submitted to the

Graduate School - Newark

Rutgers, The State University of New Jersey

in partial fulfillment of the requirements

for the degree of

Doctor of Philosophy

Graduate Program in Management

Supply Chain Management Major

written under the direction of

Dr. Lei Lei

and approved by

Newark, New Jersey

January, 2014

© Copyright 2013

Shengbin Wang

All Rights Reserved

ABSTRACT OF THE DISSERTATION

Operations Scheduling with Renewable and Non-Renewable Resources by Shengbin Wang Dissertation Director: Dr. Lei Lei

We study a supply chain operations scheduling problem subject to both renewable and non-renewable resources. After an operation has been completed, the non-renewable resource is consumed whereas the renewable resource may be resumed for the next operation. Of both the renewable and the non-renewable resources, limited amounts are available and they need to be delivered to the locations where they are needed. The operations have deadlines, and the availability of the renewable resources depends on the sequence of the operations. Our problem is to find a coordinated operations schedule for the non-renewable and renewable resources so that the total tardiness across all the customers in the given network is minimized.

Part 1 of this dissertation presents an overview of existing solution methodologies for integrated supply chain operations scheduling/planning problems involving production, inventory, distribution, and routing. We take into account problems dealing with operational decisions and classify them according to their characteristics, such as time constraints and routing decisions that are directly related to our research problem. Various methodologies are presented and discussed, and their possible integrations, combinations, and extensions are discussed.

In Part 2 of the dissertation, we build a mixed integer-programming model, present a complexity classification for our problem, and show where the borderline lies between NP-Hardness and polynomial time solvability. We analyze the structural properties of our problem, provide strongly polynomial-time solutions for several special cases that have practical applications, and identify the cases that are computationally intractable. Finally, we propose a framework of heuristic procedures for solving more general versions of this problem.

In Part 3 of this study, we introduce a mathematical programming based rolling horizon heuristic that is able to locate near optimal solutions within ten-minute of CPU time for networks up to 80 customer service operations. This heuristic solves the problem through an iterative process. In each iteration, a subset of customers and a subset of batches of non-renewable resources, together with the travel teams (renewable resources), are scheduled by solving a respective optimization problem of a much smaller size. Through an extensive empirical study with over 5,000 randomly generated test cases under various parameters, the empirical error gaps of this proposed solution approach, when compared to the best solution obtained by a commercial optimizer within one-hour of CPU time, are constantly within 5%.

This work can be extended in several directions. One of them is to conduct a thorough simulation study to assess the impact of management policies on the effectiveness of emergency logistics involving bottleneck renewable and non-renewable resources. Another one is to design and evaluate meta-heuristics for solving a more general version of our problem.

Preface

This Ph.D. thesis entitled "Operations Scheduling with Renewable and Non-Renewable Resources" has been prepared by Shengbin Wang during the period from September 2009 to November 2013, at the department of Supply Chain Management and Marketing Sciences at the Rutgers University.

The Ph.D. project has been completed under the supervision of my advisor Professor Lei Lei. The subjects of the thesis are establishing mathematical models, analyzing properties, and proposing solution methodologies for solving supply chain operations scheduling problems subject to both renewable and non-renewable resources, and reporting the effectiveness of these methodologies. The thesis is submitted as a partial fulfillment of the requirement for obtaining the Ph.D. degree at the Rutgers University. The project was supported by the Teaching Assistantship and Dean's Summer Research Fund of Rutgers Business School, Rutgers University.

Acknowledgements

This dissertation encompasses the research contributions and efforts that have made up my four-year Ph.D. life. First of all, I would like to express my most sincere gratitude to my advisor Dr. Lei Lei. My thanks to her will never be enough. Without her continuous and generous support and encouragement, I would not have completed this work or my doctoral degree. Her passion and dedication will definitely keep impacting me in my future academic career. It has been the greatest honor for me to be one of her students and to learn so much in the process of her guiding my dissertation.

I would also like to give my special thanks to my dissertation committee members: Dr. Yao Zhao, Dr. Lian Qi, and Dr. Tan Miller. My gratitude must also be given to Dr. Mike Pinedo. I appreciate their valuable time and critical comments toward my research. Their help on improving this dissertation was invaluable.

I must also thank the professors who taught me at Rutgers Business School: Dr. Michael Katehakis, Dr. Jian Yang, Dr. Xiaowei Xu, Dr. Lei Wang, Dr. Endre Boros, Dr. Douglas Jones, and Dr. Kangbok Lee. Their knowledge has profoundly impacted and stimulated me. I feel honored to have the opportunity of working with them both in and out of the classroom. Finally, I want to thank my parents, my cousin, Xin Li, and my friends, Jing Ren, Ziyue Huang, Yuan Hong, Quan Deng, Yifeng Liu, Shaofeng Zheng, Yijun Wang, Xin Xu, Ken Chen, Feng Liu, Truman Hong, Mingyuan Song, and Youqing Zhou. They have given me a lot of support and help during my four years of study at Rutgers University.

North Carolina, December 2013

Shengbin Wang

Table of Contents

ABSTRACT OF THE DISSERTATION	II
PREFACE	IV
ACKNOWLEDGEMENTS	V
TABLE OF CONTENTS	VII
LISTS OF TABLES	IX
LIST OF ILLUSTRATIONS	X
CHAPTER 1. INTRODUCTION	1
CHAPTER 2. LITERATURE REVIEW	7
2.1 Production and Distribution Problem (PDP)	14
2.1.1 Heuristic and Metaheuristic Algorithms	16
2.1.2 Constraints Relaxation-Based Approaches	20
2.1.3 Variables Relaxation-Based Approaches	25
2.1.4 Remarks on PDP	28
2.2 PDP with Time Constraint (PDPT)	29
2.2.1 Metaheuristic and Iterative Approach	31
2.2.2 Mathematical Modeling and the Use of Optimization Solver	34
2.2.3 Remarks on PDPT	37
2.3 DDP with Pouting (DDPP)	29
2.3.1 Mathematical Programming Approach	40
2.3.2 Metaheuristic Approach	43
2.3.3 Incorporating Routing Cost Approximation for Solving PDPR	45
2.3.4 Remarks on PDPR	47
2.4 PDP with Routing and Time Constraints (PDPRT)	48
2.4.1 Decomposition Methods	50
2.4.2 Integrated Methods	52
2.4.3 Remarks on PDPRT	54
2.5 PDP in Emergency Logistics (PDPEL)	55
2.6 Discussion	61
2.6.1 Structure of Solution Approach	61

2.6.2 Problem Structure and Solution Approaches 2.6.3 Trends and Applications	67 69
CHAPTER 3. PROBLEM DEFINITION AND THE MIP MODEL	71
CHAPTER 4. A STRUCTURAL ANALYSIS OF PROBLEM P	77
4.1 The Single DC Case	79
 4.2 Multiple DCs in a Fixed Assignment Environment 4.2.1 A Variable Number of DCs and a Single Medical Team 4.2.2 Fixed Numbers of DCs and Teams in a Fixed Assignment Environment 	88 88 93
4.3 Multiple DCs in an Open Assignment Environment	99
4.4 Overview of Complexity Results	106
4.5 Conclusion	107
CHAPTER 5. A ROLLING-HORIZON BASED HEURISTIC FOR SOLVI	NG P 111
5.1 The Single Batch Problem	111
5.2 A Rolling-Horizon (RH) Heuristic for Solving P	119
5.3 Empirical study	128
CHAPTER 6. FUTURE EXTENSIONS	137
6.1 A Simulation Study for Assessing the Impact of Management Policies	137
6.2. Designing Meta-heuristics for Solving General Versions of Problem	145
BIBLIOGRAPHY	147
APPENDIX I. MATLAB MIP CODE FOR PROBLEM P	158
APPENDIX II. MATLAB RH CODE FOR PROBLEM P	162
APPENDIX III. MATLAB MANAGERIAL INSIGHT CODE FOR PROB	LEM P 173
APPENDIX IV. MATLAB SIMULATION CODE	175
VITA	179

Lists of Tables

Labol		Occurrence	
Laber			
Table 2.1	Categories of the integrated operations planning	10	
	problems		
Table 2.2	Summary of Literature Review of Emergency	50	
	Logistics	58	
Table 2.3	Summary of Solution Approaches	65	
Table 4.1	Parameters for DCs and Batches (Greedy Algorithm)	91	
Table 4.2	Parameters for Hospitals (Greedy Algorithm)	91	
Table 4.3	Results Generated with the Greedy Algorithm	92	
Table 4.4	Parameters for DCs and Batches (Dynamic	0.0	
	Programming)	98	
Table 4.5	Parameters for Hospitals (Dynamic Programming)	98	
Table 4.6	Complexity Framework of problem $P(F, n_K, n_M)$	106	
Table 4.7	Complexity Framework of problem $P(O, n_K, n_M)$	107	
Table 4.8	Heuristic Approaches for a General Problem	110	
Table 5.1	Parameters Used in the Empirical Study for the RH	129	
	Algorithm		
Table 6.1	Parameters Used in the Empirical Study for the RH	1/1	
	Algorithm	1+1	
Table 6.2	The Average Tardiness Savings for Four Different	144	
1 4010 0.2	Cases		

List of Illustrations

Lahol	abel Title		
Label			
Figure 2.1	Network Structure and Material Flow	11	
Ei	An Overview of Existing Procedures for Solving the	65	
	Integrated Problem		
Figure 3.1	Supply Network with Both Renewable and Non-renewable	73	
	Resources	75	
Figure 4.1	Examples with a Single DC Supply Process	83	
Figure 1 2	A Numerical Example for the Proposed Dynamic	99	
Figure 4.2	Programming Algorithm		
Figure 5.1	Examples of the Multiple-DC Single-Batch Process	112	
Figure 5.2	Flow Chart of an Example Min-Cost Max-Flow Problem	117	
Figure 5.3	An Illustrative Example of Lemma 5.3	119	
	A Graphical Illustration of a Two-Batch N-Hospital (N=7)		
Figure 5.4	Sub-Problem Solved by the RH Algorithm during the	120	
	Search Process		
Figure 5.5	A Flowchart of the Search Process by the Proposed RH	123	
Figure 5.5	Algorithm	125	
Figure 5.6	Special Case P ₉	125	
Figure 5.7	Tōhoku earthquake and Tsunami, background map: Miyagi	130	
Eigene 5 9	Error Gap Distribution under High Level of Variability in	121	
Figure 5.8	Customer Demand $(n_K = 2)$	131	
E	Error Gap Distribution under High Level of Variability in	122	
Figure 5.9	Customer Demand $(n_K = 3)$	132	
Eigene 5 10	Average Empirical Error Gaps under Different Levels of	122	
Figure 5.10	Demand Variability $(n_K = 2)$	155	
		1	

Figure 5.11	Average Empirical Error Gaps under Different Levels of Demand Variability $(n_K = 3)$	133
Figure 5.12	Computational Effort Required by the RH Algorithm and that by GUROBI	134
Figure 5.13	Average Error Gap (%) against the Number of Travel Teams	134
Figure 5.14	The Impact of Pre-positioning Inventory Before the Arrival of a Disaster	137
Figure 6.1	A Flowchart of the Simulation Process of Evaluation of Mobile DC	143
Figure 6.2	Average Tardiness Comparison for Demand with Poisson Distribution with $n_K = n_M = 3$ and $n_H = 20$	145
Figure 6.3	Average Tardiness Comparison for Demand with Poisson Distribution with $n_K = n_M = 4$ and $n_H = 30$	145
Figure 6.4	Average Tardiness Comparison for Demand with Triangle Distribution with $n_K = n_M = 3$ and $n_H = 20$	146
Figure 6.5	Average Tardiness Comparison for Demand with Triangle Distribution with $n_K = n_M = 4$ and $n_H = 30$	146

Chapter 1. Introduction

A supply chain is defined as an integrated business process with bidirectional flows of products, information, cash, and services, between tiers of suppliers, manufacturers, logistics partners, distributors, retailers, and customers. Due to fast changes in the marketplace and the rapid expansion of supply chains (Eksioglu et al., 2007), ensuring highly coordinated production, inventory, and distribution over a multi-echelon supply chain network is vital, and has an immediate impact on customer services and profit margins. This importance will continue to increase along with the following trends:

Globalization: All functions in a supply chain network, such as procurement, production, distribution and consumptions, have now become more globalized. Most multi-national firms have business facilities located over multiple continents, with many local markets to serve; face the need for emerging market penetration and the challenge of capacity shortages and rising shipping costs; and are constantly confronting environmental/sustainability concerns. At the same time, the promises and flexibility of third-party logistics and subcontracting opportunities offer a great incentive to expand supply chains globally. As supply chains expand, the need to ensure a more precise match between demand and supply increases the importance of integrated operations planning.

Pressure on lead time reduction and profit margin improvement: Since customer demand for both products and services typically changes over time, time-to-market is more important than ever in order to meet the expectations of demanding customers. For most supply chains, production is not the only major process to be considered; there are many other stages, such as sourcing, distribution, inventory, packaging, and order processing that together could account for a significant portion of the lead time. A less-coordinated supply chain process could easily diminish or eliminate the profit margin and lead to poor customer service.

Advances in information technology: Advances in information technology during the past two decades have significantly improved data visibility (e.g., inventory visibility and shipping status) and information accessibility along the supply chain. Data can be automatically collected, retrieved, and manipulated in various ways and shared by many supply chain partners (e.g., through RFID). Furthermore, today's computing power allows us to solve larger-scale integrated operations planning problems relatively easily and more rapidly, which were difficult, if not impossible, only a few decades ago when optimization problems of a combinatorial nature were considered computationally intractable.

Serving the needs of emerging non-commercial supply chains: A network for disaster relief operations is a typical illustration of a non-commercial supply chain. Disaster relief and emergency logistics (e.g., in response to Hurricane Katrina in Louisiana in 2005, the tsunami in Japan in 2011, and Hurricane Sandy in New Jersey and New York in 2012) usually cannot be effectively handled by a single state or a single local government. Today's Internet allows the need for disaster relief to be communicated cross-country and internationally within minutes of an event, and the rapid formation of disaster relief supply chains for quick response to people in the affected areas. A highly effective and fully integrated production and distribution operation that pulls supplies from different industries and states to ensure delivery of these resources to the people in an affected area is critical to human well-being.

Among these trends in the application of integrated production and distribution operations, emergency logistics has received least attention in academia. This dissertation focuses on a particular topic in this area: emergency operations scheduling subject to both renewable and non-renewable resources. When operations scheduling becomes subject to both renewable and non-renewable resources and when the services of the customers depend on the availability of both types of resources at the same time, the resulting scheduling problems become very difficult (Haghani and Oh 1996). One reason is that the execution of an operation depends on the availability of both resources, each of which being subject to different constraints, making the problem much harder to solve relative to its counterpart that is subject to just one type of resource. In the literature, results for such scheduling problems involving both renewable and non-renewable resources are limited (e.g. Ait-Kadia, Menye and Kane 2011; Bottcher et al. 1999; Can and Ulusoy 2010; Lee and Lei 2001; Nudtasomboon and Randhawa 1997).

This operations scheduling problem is commonly encountered in disaster relief processes. During such a process, various medical supplies, such as syringes, antibiotics, surgical blades, vaccines, and bandages (i.e., non-renewable resources) as well as medical teams consisting of nurses, doctors, and/or first aid social workers (i.e., renewable resources) have to be available at the same time in order to take care of the patients' needs. One well-known case of such a challenge was the situation immediately following Hurricane Katrina, which had a devastating effect on New Orleans in 2005. Both federal and local governments launched their emergency response systems in order to help affected people. However, the most critical bottleneck in the entire process was the capacity of nurses. Even though medical kits and vaccines were delivered on time, patients could not be treated without the availability of nurses. As a more recent example, Hurricane Sandy, the largest Atlantic hurricane on record (http://en.wikipedia.org/ wiki/Hurricane sandy), ruined the entire stocks of medical supplies at many local pharmacies while the demand for supplies (e.g., cardiac medicines, anti-clotting medicines, and statins) increased significantly, which led to many tough challenges in relief operations. The massive tornado that hit Oklahoma in late May 2013 is another example of resource-constrained emergency operations scheduling, where the tornado caused severe blood shortages in hospitals and shelters (http://www.thetimesnews.com /news/top-news/the-alamance-scene-blood-needed-in-oklahoma-tornado-aftermath-1.148 471) and delayed the medical treatments despite the availability of travelling medical teams.

Such operations scheduling problems are also fairly common in the practice of project management, where the usage of non-renewable resources (e.g. construction materials and lumber supplies) and renewable resources (e.g. cement mixers, engineers, and trucks) have to be coordinated and synchronized. According to Assaf and Al-Hejji

(2006), poor communication and coordination of labor and construction supplies is one of the main causes of delays in large construction projects. Besides, labor shortage has been one of the most frequent causes of project delays.

The rest of this dissertation is organized as follows. We survey the existing solution methodologies for integrated supply chain operations scheduling/planning problems involving production, inventory, distribution, and routing in Chapter 2. We take into account problems dealing with operational decisions and classify them according to their characteristics, such as time constraints and routing decisions that are directly related to our research problem. Various methodologies are presented and discussed, and their possible integrations, combinations, and extensions are discussed. In Chapter 3, we formally define our operations scheduling problem with renewable and non-renewable resources, and build a solid mixed integer-programming model towards the problem. Moreover, we present a complexity classification for our problem, and show where the borderline lies between NP-Hardness and polynomial time solvability in Chapter 4. In the same chapter, we also analyze the structural properties of our problem, provide strongly polynomial-time solutions for four special cases that have practical applications, and identify the cases that are computationally intractable. In Chapter 5, we introduce a rolling horizon based heuristic that is able to locate near optimal solutions within ten-minute of CPU time for networks up to 80 customer service operations. This heuristic solves the problem through an iterative process. In each iteration, a subset of customers and a subset of batches of non-renewable resources, together with the travel teams (renewable resources), are scheduled by solving a respective optimization problem of a much smaller size. Through an extensive empirical study with over 5,000 randomly generated test cases under various parameters, the empirical error gaps of this proposed solution approach, when compared to the best solution obtained by a commercial optimizer within one-hour of CPU time, are constantly within 5%. Finally in Chapter 6, we discuss several extensions in various directions. One of them is to conduct a thorough simulation study to assess the impact of management policies on the effectiveness of emergency logistics involving bottleneck renewable and non-renewable resources. Another one is to design and evaluate meta-heuristics for solving a more general version of our problem.

Chapter 2. Literature Review

In this chapter, we focus on the solution methodologies for solving various *integrated/coordinated* production and distribution operations planning problems reported in the current literature. This literature review does not focus on results related to decisions for supply chain designs (e.g., facility location and/or facility capacity), or on those results that only deal with a single operation such as inventory, or routing, or production scheduling, but rather addresses issues unique to process integration.

There have been several survey papers dealing with integrated operations problems, each with its own focus. Among these, the pioneer review by Thomas and Griffin (1996) defines a generic structure for a supply chain network, and classifies published results at both the strategic planning level and the operational planning level, where the latter falls into our scope. The models related to operational planning are classified into buyer and vendor coordination, production-distribution coordination, and inventory-distribution coordination; up through the time of this study, most researchers, because of limitations on computational capability, have decomposed such multi-stage problems into several two-stage problems which are then solved separately. Erenguc et al. (1999) review the studies on managing supply chain networks with three distinct stages consisting of suppliers, plants, and distribution centers, and focus on the results for joint operational decision-making across the three stages. Decisions that need to be jointly made regarding optimizing production/distribution planning are discussed. Sarmiento and Nagi (1999) consider integrated production/distribution planning systems at both the strategic and tactical levels with an explicit consideration of transportation. They classify the problems based on the type of decisions being modeled (e.g. decisions on production, distribution, or inventory management) and on the number of locations per echelon in the model. Three categories of two-echelon models are identified, and the differences between such models and those in classical Inventory Routing studies are discussed. Fahimnia et al. (2008b) review existing production /distribution planning models and provide a table summarizing 19 papers according to problem attributes (e.g. numbers of plants, distribution centers, and customers, multi-periods, multi-products, routing), types of modeling approaches (e.g. mathematical programming, optimization, simulation and combinations of these), and the solution methods applied.

There are also two recent survey papers on integrated operations planning: Mula et al. (2010) and Fahimnia et al. (2013). Mula et al. (2010) cite 44 papers published since 1985 among the 54 references, and classify these works based on the decision levels (e.g. strategic, tactical, and operational), modeling approach (e.g., linear programming, and multi-objective integer linear programming), objective (e.g., total cost, and customer satisfaction), level of information sharing (e.g., production cost, lead time, inventory level, and demand), and solution methodologies. Fahimnia et al. (2013) cite 139 papers related to integrated operations planning, and classify these papers by two criteria: complexity of the network structure and solution methodologies. Interestingly, in spite of the large number of references listed in these surveys, only 19 papers were common to both surveys. However, there is no analysis in either survey on the relationship between problem structures and the methodologies reported in these works.

Unlike the existing surveys, we focus here on the relationships between the problem structures and solution methodologies. Such a survey provides information to the researchers on the solution approaches, developed for solving problems defined over different types of network structures, and their effectiveness. We classify the integrated operations planning problems into four categories. For each category, we present a basic mathematical model and, based upon the properties of the respective network structure, analyze the existing solution methodologies. To define these categories, two attributes are used: time constraint and routing. Most integrated operations planning problems involve multiple time periods. For each period, the ending inventory level, production quantity, and distribution amount must be determined. Since a continuous time scale within a period has to be considered in some studies to describe time constraints like arbitrary delivery deadlines or travel times, there is a need to model the time constraints explicitly. Note that without such explicit modeling of time constraints, as many studies in the past have done, we often have to assume that any quantity produced in one period is delivered to customers in the same period, which leads to a gap between the models and the real-world practice. For those studies involving direct shipment between suppliers and customers, we allow the shipping capacity to be defined as either the maximum outgoing flow amount or the fleet size and/or capacity of vehicles. For the studies in which one vehicle may visit several customers in one trip, we allow vehicle routing issues to be explicitly included in the model. We categorize the problems into four categories in Table 2.1.

Issues in the Literature	Production	Distribution	Time	Routing
Problem Categories	issues	issues	constrains	issues
Production and Distribution Problem (PDP)	X	Х		
PDP with Time Constraints (PDPT)	Х	Х	Х	
PDP with Routing (PDPR)	X	Х		Х
PDP with Routing and Time Constraints (PDPRT)	Х	Х	X	X

 Table 2.1. Categories of the integrated operations planning problems

We also refer readers to another survey by Yossiri et al. (2012), in which the authors categorize the studies according to their inclusion of decision variables related to the flow quantity of production, inventory, distribution and routing.

Before we give the details of 5 categories, we introduce the common assumptions and notation used to define the four categories of problems (PDP, PDPT, PDPR, and PDPRT). For each assumption, we then discuss its extensions or variations that are found in literature.

Product and Time Dimension:

• We consider the multi-product problem (i.e., with multiple commodities) over a given planning horizon of multiple time periods.

Network Structure and Material Flow:

• The supply chain network has three stages: manufacturers, distribution centers (DCs) and customers, as shown in Figure 2.1. Each customer has a certain demand to be fulfilled in each period. Both manufacturers and DCs hold inventories of products. Manufacturers produce and fill their own inventories, and send products to DCs, which in turn send the products to customers.



Figure 2.1 Network Structure and Material Flow

• *Extensions or Variations in the literature:*

o There exist suppliers to provide manufacturers with raw material.

• There exist third parties that serve as contract manufacturers or DCs. The third parties usually charge higher prices than regular players.

• In some cases, manufacturers may deliver the product directly to customers.

Production and Transportation Capacity:

- Each manufacturer has a maximum production capacity (e.g., the maximum quantity that it is able to produce) in each period. Both manufacturers and DCs have a maximum transportation capacity (e.g., the maximum outgoing flow quantity) in each period.
- Extensions or Variations in the literature:
 - Manufacturer's production capacity can be increased at an additional fixed and/or variable cost (e.g., overtime work).
 - Transportation capacity can be defined by the vehicle attributes (e.g., the fleet size, the vehicle loading capacity, the maximum number of trips, and the total working hours in one period, etc.).

Customer Demand Fulfillment and On-time Delivery:

- All customer orders must be fulfilled on time, and no customer carries inventory.
- Extensions or Variations in the literature:
 - If an order is not fulfilled on-time, it is lost (called a *lost-sale*).
 - If an order is not fulfilled on-time, it can be fulfilled later with a penalty cost
 (either as a backorder delivered in a subsequent period, or as a late
 shipment within the same period).

Cost Components:

• Each manufacturer has a fixed, and variable, cost of production, and each DC has

a fixed, and variable, cost for handling the product. Both manufacturers and DCs incur inventory holding costs. The shipments from manufacturers to DCs, and from DCs to customers, result in a shipping cost.

• *Extensions or Variations in the literature:*

• When raw materials are required, the purchase cost is considered.

- When a third party is involved, the respective costs (e.g., contract fees) are included.
- If a late delivery (backorder) is allowed, the relevant penalty cost is included.
- \circ If a lost-sale is allowed, the shortage penalty is included.

While a representative mathematical model for each of the following sections is built upon these basic assumptions, its variations are introduced as we discuss individual papers.

Throughout this survey, we will use the following notation: let $M=\{m\}$, $B=\{i\}$, $J=\{j\}$ and $K=\{k\}$ denote the set of manufacturing facilities, the set of distribution/transshipment centers (DCs), the set of customers, and the set of products ordered by customers, respectively. When routing decisions are involved, let V(m) denote the set of vehicles of manufacturer m. Let $T = \{t\}$ denote the set of periods. For simplicity, $\forall m$, $\forall i$, $\forall j$, $\forall k$, $\forall v$ and $\forall t$ may be used instead of $\forall m \in M$, $\forall i \in B$, $\forall j \in J$, $\forall k \in K$, $\forall v \in V(m)$ and $\forall t \in T$.

2.1 Production and Distribution Problem (PDP)

The **production and distribution problem**, or **PDP**, is primarily concerned with coordinating production and outbound distributions to minimize the total costs associated with production, inventory, and transportation over a discrete multi-period planning horizon. Since PDP does not explicitly include the routing and shipping times, the models for PDP involve only inventory flow balance, facility capacity and transportation capacity constraints (e.g., see Thomas and Griffin, 1996).

To formally define the mathematical model for the PDP, we introduce the following notation: For any given period t, let $C_{m,t}^{k}$ be the production capacity of manufacturer mfor product k, $C_{a,b,t}$ be the transportation capacity from location a to location b for $(a,b) \in M \times B \cup B \times J$, and $d_{j,t}^{k}$ be the demand for product k by customer j. Let $I_{a,0}^{k}$ be the initial inventory of product k at location a for $a \in M \cup B \cup J$. For decision variables, let $W_{a,b,t}$ and $Z_{m,t}^{k}$, respectively, be the binary variables denoting the decision for a flow from location a to location b for $(a,b) \in M \times B \cup B \times J$ in period t, and the decision for a production batch for product k by manufacturer m in period t. Let S, Q, P, and I, each with proper superscript and subscript indices, be continuous variables denoting the shortage amount, flow quantity, production quantity, and inventory level, respectively. For example, $Q_{m,j}^{k}$ denotes the flow quantity of product k from manufacturer m to DC i in period t. In addition, we use M||J, and B||J, to denote a network involving only manufacturers and customers, and distribution centers and

Minimize:
$$G(W_{m,i,l}, W_{i,j,l}, Z_{m,l}^k, S_{j,l}^k, Q_{m,l,l}^k, Q_{m,l}^k, P_{m,l}^k, I_{m,l}^k, I_{j,l}^k, I_{j,l}^k)$$
 (2.1)

s.t.

$$I_{m,t-1}^{k} + P_{m,t}^{k} - \sum_{\forall i} Q_{m,i,t}^{k} = I_{m,t}^{k}, \qquad \forall m, k, t$$
(2.2)

$$I_{i,t-1}^{k} + \sum_{\forall m} Q_{m,i,t}^{k} - \sum_{\forall j} Q_{i,j,t}^{k} = I_{i,t}^{k}, \qquad \forall i,k,t$$
(2.3)

$$I_{j,t-1}^{k} + \sum_{\forall i} Q_{i,j,t}^{k} - (d_{j,t}^{k} - S_{j,t}^{k}) = I_{j,t}^{k}, \qquad \forall j, k, t$$
(2.4)

$$P_{m,t}^k \le C_{m,t}^k \cdot Z_{m,t}^k, \qquad \forall m,k,t \qquad (2.5)$$

$$\sum_{\forall k} \mathcal{Q}_{m,i,t}^{k} \leq C_{m,i,t} \cdot W_{m,i,t}, \qquad \forall m, i, t \qquad (2.6)$$

$$\sum_{\forall k} Q_{i,j,t}^k \le C_{i,j,t} \cdot W_{i,j,t}, \qquad \forall i, j, t \qquad (2.7)$$

$$W_{m,i,t}, W_{i,j,t}, Z_{m,t}^{k} \in \{0,1\}, \quad S_{j,t}^{k}, Q_{m,i,t}^{k}, Q_{i,j,t}^{k}, P_{m,t}^{k}, I_{m,t}^{k}, I_{i,t}^{k}, I_{j,t}^{k} \ge 0 \qquad \forall m, i, j, k, t \quad (2.8)$$

The objective function (2.1) minimizes the total operations cost, consisting of raw materials, facility setup, production, inventory, and transportation costs. Constraints (2.2) - (2.4) ensure the flow balances at the manufacturing facilities, DCs and customer sites, respectively, while constraints (2.5) - (2.7) are network capacity constraints.

While special cases of PDP, such as the classical transportation problem and the transshipment problem, can be solved in strongly polynomial time, the general version of the PDP is difficult to solve. More precisely, the multi-product PDP defined by (2.1) - (2.8) is strongly NP-hard, because a special case of this PDP is a multi-product

multi-period lot-sizing problem which has been proved to be strongly NP-hard by Chen and Thizy (1990). Therefore, a general version of PDP could require an excessive amount of computational time to verify the solution optimality when the network size becomes large.

In this section, we focus on the existing solution methodologies for variations of the PDP defined by (2.1) - (2.8), and classify them into three categories. The first one is *heuristic and metaheuristic algorithms*, in which a solution (or a set of solutions) is constructed by relatively simple rules and then improved through an iterative process. The other two are both mathematical programming-based solution approaches, and differ in the way that a problem is relaxed: *constraints relaxation approaches* and *variables relaxation approaches*. Note that while the routing decision is not considered in this section, we do include those problems that assume fixed routing.

2.1.1 Heuristic and Metaheuristic Algorithms

Because of the intractability of the general PDP, feasible solutions with acceptable quality and minimal solution time have been commonly discussed in the literature. Representative solution approaches in this category are *greedy heuristics* and *genetic algorithms*.

Park (2005) proposes a two-phase heuristic for solving a multi-product PDP defined upon an M|J network to maximize the total profit. The phase I problem is formed by aggregating the demand of all customers in each period, defined by $D_t^k = \sum_{\forall j} d_{j,t}^k$ and then replacing constraint (2.4) by $I_{t-1}^k + Q_t^k - D_t^k = I_t^k$, $\forall k, t$, in the model, which reduces the problem to a single-customer multi-period model and allows one to quickly determine the values of $P_{m,t}^k$ by solving a production lot-sizing problem (Fumero and Vercellis, 1999) with constant production capacity. All unsatisfied demand is penalized as shortage and no backorder is considered. Given $P_{m,t}^k$, the author then solves a distribution problem in phase II to determine the values of $Q_{m,j,t}^k$, by applying a bin-packing heuristic together with local improvement procedures which consolidate partial loads by shifting shipping periods and reducing the level of stock-out using leftover production Through computational experiments on 21 test problems of three sizes, this capacity. heuristic achieves an error gap, or a difference between the optimal and heuristic solutions, of $5.6 \sim 6.8\%$ for small-size cases and no more than 9.2% for all the test cases. The computation time is less than 3 sec for small cases and no more than 1200 sec for large cases.

Ahuja et al. (2007) study a two-echelon M||J single product PDP with *single* sourcing constraint, which means that each customer receives shipment from at most one supplier in each period. In addition to constraints (2.2) - (2.7), the authors also include a constraint on inventory perishability, so that the maximum inventory time for the product is bounded by a given constant N. Thus, at any period t, the ending inventory at DC i, $i \in I$, cannot exceed its future demand from all customers in the next N periods, or $I_{i,t} \leq \sum_{n=1}^{N} \sum_{\forall i} Q_{i,j,t+n}$. The resulting PDP is decomposed into two sub-problems. One includes only binary facility-customer assignment variables, and the other includes variables for transportation flow and inventory levels. A proposed greedy heuristic is used to assign the facility-customer pairs, upon which a very-large-scale-neighborhood (VLSN) search heuristic is applied to improve the quality of the solution. Extensive tests on randomly generated problem sets are conducted, and the error gap obtained by comparing the heuristic with the best lower bound obtained by CPLEX within 15 minutes of CPU time is less than 3% in all cases. The authors also report that their error gaps have a decreasing tendency as the number of customers is increased, and it is less than 0.1 % in the largest size case. The computation time is less than 40 seconds in all cases.

Some researchers consider PDP with extensions such as fixed routes for transportation or direct shipment. Lei et al. (2006) investigate an integrated production, inventory and distribution routing problem encountered from the practices of after-merge operations of a chemical company. A two-phase approach is proposed, where the Phase I problem is defined by assuming direct shipment between manufacturing plants and customers. The assumptions on direct shipments allow the authors to solve an optimization problem with a significantly reduced complexity, which yields a feasible solution to the original problem. The problem in Phase II is to improve the solution from Phase I, and is modeled as a shortest path problem on a directed acyclic graph. An empirical study that evaluates the computational performance of this solution approach is also reported. Liu et al. (2008) study a multi-product packing and delivery problem with a single capacitated truck and a fixed sequence of customer locations. The authors first apply a network flow-based polynomial time algorithm to solve the problem assuming no

split deliveries, and then allow the split delivery to improve the truck efficiency by using a greedy heuristic with a time complexity of $O(|J|^3 \log |J|)$. In both papers, optimal solutions of the special cases (with restriction) are modified to obtain feasible solutions to the original problems.

During the past two decades, the genetic algorithm (GA), inspired by the process of natural evolution, has been quickly gaining in popularity. In Jang et al. (2002), the problem of production and distribution planning over a three-echelon M|B||J network is considered. Constraints similar to (2.1) - (2.7) are included and a material transform factor Γ is used to define the rate of raw materials consumption: $I_{m,t-1} + P_{m,t} - \sum_{\forall i} \Gamma_{mi} \cdot Q_{m,i,t} = I_{m,t}, \forall m,t.$ The proposed GA algorithm is compared with that obtained by CPLEX. Among randomly generated test problems, the solution time of GA is quite stable, averaging from 334 to 546 seconds, while that required by the CPLEX solver exhibits exponential growth with respect to problem size, from 32 to 67,854 seconds to obtain the optimal solutions. The proposed GA also demonstrates strong performance, with an average error gap of 0.2%. Gen and Syarif (2005) propose a GA-based approach for their M|J network. A new solution approach called the spanning-tree-based genetic algorithm is presented together with the fuzzy logic controller concept for auto-tuning the GA parameters. The proposed method is also compared with a traditional spanning-tree-based approach. This comparison shows that the proposed approach achieves a better result in every experiment, with an average improvement from 0.05% to 0.65% for six different settings. Kannan et al. (2010) develop an M|B||J network model for battery recycling. Besides production, inventory

and transportation cost, the objective function contains additional cost factors for recycling such as collection, disposal and reclaiming cost. The authors introduce a heuristic-based genetic algorithm to solve the problem and compare the result with that obtained by GAMS, a commercial solver. In experiments with different problem sizes and heuristic parameters (population and iteration), the maximum error observed is 7.4% compared with the results from GAMS. Moreover, the average computation time of the GA-based approach is less than 315 seconds for the largest problem while that by GAMS is at least 2800 seconds for the smallest problem.

2.1.2 Constraints Relaxation-Based Approaches

Another popular solution approach to PDP in the current literature is to relax a subset of constraints in order to make the relaxed problem easier to solve. The major approach in this regard is the well-known *Lagrangean relaxation*, by which difficult constraints are placed into the objective function with coefficients called Lagrangean multipliers so that the resulting problem is "easily solvable." One example of such an easily solvable problem is a network flow problem (Ahuja et al., 1993). Another important approach is based upon problem decomposition, by which a subset of constraints is temporarily simplified or removed from the original model to make the remaining problem decomposable. When a Lagrangean relaxation is adapted to achieve the decomposition, the resulting process is called *Lagrangean decomposition*. In constraints relaxation – based approaches, identifying the constraints to be relaxed and ensuring that the search converges to the optimal or near-optimal solution quickly are two critical steps for achieving the quality and effectiveness of such solution approaches. For example, in the basic model defined by (2.1) - (2.8), when we relax constraint (2.3) and incorporate it in the objective function with penalty factors, the problem is decomposed into two problems as follows:

- *Minimize*: $G^{I}(W_{m,i,t}, Z_{m,t}^{k}, Q_{m,i,t}^{k}, P_{m,t}^{k}, I_{m,t}^{k})$ s.t. (2.2), (2.5), (2.6)
- *Minimize*: $G^{2}(W_{i,j,t}, S_{j,t}^{k}, Q_{i,j,t}^{k}, I_{j,t}^{k})$ s.t. (2.4), (2.7), (2.8)

where both G^{I} and G^{2} include the penalty terms for violating constraint (2.3).

Yung et al. (2006) use constraints relaxation to solve a multi-product single-period PDP, and thus the time index *t* is dropped from all the notations, defined upon an M||J network. Their study involves decisions on production and transportation, as well as on lot-sizing and order quantity. The average inventory level is used to define the inventory cost, and variables z_m^k and x_{mj}^k are added to denote production lot size and shipping quantity for product *k*. The model contains flow balance constraints similar to (2.2) - (2.4), and capacity constraints similar to (2.5) - (2.7). However, the objective function includes terms P_m^k / z_m^k as the number of production lots for product *k* at manufacturer *m* and terms Q_{mj}^k / x_{mj}^k as the number of shipments of product *k* from *m* to *j*, which lead to a non-linear objective function that is neither convex nor concave. In order to apply Lagrangean relaxation, an artificial variable R_{mj} , where:

$$\sum_{k} Q_{mj}^{k} = R_{mj} \tag{2.9}$$

is utilized, and redundant constraints $\sum_{k} P_m^k = \sum_{j} R_{mj}$, $\sum_{k} d_j^k = \sum_{m} R_{mj}$, and $0 \le R_{mj} \le \sum_{k} d_j^k$ are added to the model. By relaxing constraint (2.9), the original

model is decomposed into two independent sub-models. The first one deals with joint decisions on production and lot-sizing and thus contains variables P_m^k , z_m^k and the aggregated transportation flow, R_{mj} . In the second model, the constraints for transportation planning involving Q_{mj}^k and ordering quantity x_{mj}^k are included. By continuously updating the Lagrangean multipliers and the artificial variables, two sub-problems are iteratively solved. The test result is compared with that obtained by Fmincon, a non-linear programming technique in MATLAB 6.1. Among seven problem settings, Fmincon cannot terminate for three cases while the proposed algorithm is able to solve all the cases. In terms of the solution performance, the proposed algorithm saves 1.5% to 8% in cost, with less CPU time, over what Fmincon achieves for all the cases solved.

Eksioglu et al. (2007) consider a variation of multi-product multi-period PDP on an M||J network where only the production facility carries an inventory and there are no capacity limits for inventory and transportation. The model contains flow balance constraints:

$$I_{m,t-1}^{k} + P_{m,t}^{k} - \sum_{\forall j} Q_{m,j,t}^{k} = I_{m,t}^{k}$$
(2.10)

instead of (2.1) and (2.2). Since the model does not allow shortages, it has:

$$\sum_{\forall m} \mathcal{Q}_{m,j,t}^k = d_{j,t}^k \tag{2.11}$$

instead of (2.4), and capacity constraint (2.5) with binary indicator variables for production. Unlike the previous solution approach, which uses redundant aggregated variables, this approach introduces redundant disaggregated variables. The authors

reformulate the original model by introducing a new variable $Q_{mjt\tau}^k$, which defines the amount of product k from manufacturer m to customer j to satisfy demand in period τ using the quantity produced in period t, where $t \le \tau$. Thus, the original variables can be expressed by new variables as follows:

$$P_{mt}^{k} = \sum_{j=1}^{J} \sum_{\tau=t}^{T} Q_{mjt\tau}^{k}, \quad \forall m, k, t$$
(2.12)

$$Q_{mjt}^{k} = \sum_{s=1}^{t} Q_{mjst}^{k} , \quad \forall m, j, k, t$$
 (2.13)

$$I_{mt}^{k} = \sum_{j=1}^{J} \sum_{s=1}^{t} \sum_{\tau=t+1}^{T} Q_{mjs\tau}^{k}, \quad \forall m, k, t$$
(2.14)

By using constraints (2.12) - (2.13), the original model becomes a facility location problem. The authors then show that the linear programming (LP) relaxation of the location model provides a tighter lower bound than the LP relaxation of the original model. Lagrangean decomposition is applied to the resulting location problem by introducing z_{mjtr}^{k} , clone or copy of Q_{mjtr}^{k} :

$$Q_{mjt\tau}^{k} = z_{mjt\tau}^{k}$$
(2.15)

Accordingly, redundant constraints for $z_{mjt\tau}^k$:

$$\sum_{m=1}^{M} \sum_{t=1}^{\tau} z_{mjt\tau}^{k} = d_{j\tau}^{k}$$
(2.16)

$$\sum_{j=1}^{J} \sum_{k=1}^{K} \sum_{\tau=1}^{T} z_{mj\tau\tau}^{k} \le C_{mt}$$
(2.17)

$$z_{mjt\tau}^k \ge 0 \tag{2.18}$$

are then added into the model. By relaxing (2.15) using a Lagrangean multiplier, the model is decomposed into two sub-problems. The first one containing $Q_{mjt\tau}^k$ is an

uncapacitated multi-product problem and is further decomposed into |K| single product sub-sub-problems which are NP-hard but solvable by dynamic programming. On the other hand, the second one containing z_{mjtr}^{k} can be modeled as an LP problem. For test problems of large sizes, the sub-problems are solved by using the primal-dual algorithm and the total running times vary from 4 to 87 CPU seconds with empirical error gaps no more than 5%.

Karakitsiou and Migdalas (2008) consider a single product PDP defined on an M||J network. The model has flow balance constraints similar to (2.2) - (2.4), and capacity constraints similar to (2.5) - (2.7). Defining a new variable:

$$r_{m,t} = \sum_{j} Q_{m,j,t}$$
(2.19)

the inventory flow balance constraint at *m* is replaced by:

$$I_{m,t-1} + P_{m,t} - r_{m,t} = I_{i,t}$$
(2.20)

and the transportation capacity constraint is replaced by:

$$0 \le r_{m,t} \le C_{m,t}^{S} \tag{2.21}$$

where $C_{m,t}^{S}$ is the maximum outbound shipping quantity. Moreover, a redundant constraint:

$$\sum_{m} r_{m,t} = \sum_{j} d_{j,t} \tag{2.22}$$

is added. In order to apply Lagrangean decomposition, a clone variable of $r_{m,t}$, denoted as $z_{m,t}$, is introduced:

$$r_{m,t} = z_{m,t}$$
 (2.23)

- - -

so that constraint (2.20) can be replaced by:
$$I_{m,t-1} + P_{m,t} - z_{m,t} = I_{i,t}$$
(2.24)

$$0 \le z_{m,t} \le C_{m,t}^{\delta} \tag{2.25}$$

By relaxing (2.23) and using Lagrangean multipliers, the original model is decomposed into two independent parts: the first one deals with variables $P_{m,t}$, $I_{i,t}$ and $z_{m,t}$ along with constraints (2.5), (2.24), and (2.25), while the second one deals with $Q_{m,j,t}$ and $r_{m,t}$ along with constraints (2.4), (2.19), (2.21) and (2.22). The first sub-problem can be further decomposed, over the manufacturing facilities, into |M| sub-sub-problems that can each be modeled as a linear programming problem. The second sub-problem can also be further decomposed, over the time horizon, into |T| sub-sub-problems, each as a network flow problem. In order to check the quality of the solutions produced by the Lagrangean relaxation, the results are compared with the optimal solution obtained by GLPK solver, a free and open source software. For six randomly generated problems involving 30 to 1200 nodes, the empirical error gaps are no more than 6% and the required computation time is no more than 350 seconds.

2.1.3 Variables Relaxation-Based Approaches

During the past decade, the *variables relaxation*-based approaches, in which a selected subset of integer variables is relaxed so that the reduced problem can be relatively easy to solve, have gained a significant amount of attention from researchers. While the Lagrangean relaxation procedures aim at reducing the duality gaps, most

variables relaxation based approaches focus on reducing the sub-optimality due to rounding linear values to integers.

Dogan and Goetschalckx (1999) introduce a multi-product multi-period PDP model involving strategic decisions on the network and detailed production planning on the machine level along with deterministic seasonal customer demands. The network under consideration includes candidates for suppliers, potential manufacturing facilities, and DCs with multiple possible configurations and customers. The manufacturing facilities have alternative facility types, which introduce binary variables for the facility selections, and integer variables are used to define the number of machines used in each facility during each period. In addition to the ending inventory, the authors also consider the work-in-process inventory which defines part of the inventory holding cost. Replenishment of raw material may happen more than once during each period. Transportation flow quantities and production quantities on each machine at each facility are also decision variables. Benders decomposition is used as the solution methodology. In the mixed integer master problem, the status of the facilities, the production lines, and the production and inventory quantities are determined. The reduced problem becomes a minimum-cost transportation flow problem, and its optimal cost is added to the mixed integer master problem to find a feasible schedule satisfying the obtained flow cost. The search terminates when the master problem can find no lower cost solutions. For the real life problem that motivated this study, the proposed approach saves the company an additional 2% over the hierarchical approach, where optimal strategic and tactical decisions are made sequentially. The Benders decomposition solution method with

acceleration techniques utilizing disaggregated cuts, dual variables and the LP relaxation in the initial iterations reduces the running time by a factor of 480, versus a standard Benders decomposition algorithm.

Yilmaz and Catay (2006) consider a variation of PDP involving a single product, multiple suppliers, multiple producers, and multiple distributors, with an option of capacity expansion at additional fixed and variable costs. New continuous variables representing increased capacity, and binary variables indicating capacity expansion decisions for transportation and facility, are introduced. Only manufacturers are allowed to carry inventory, and thus the inventory balance is only considered at the manufacturers' sites. Three different LP relaxation-based heuristics are used to solve the problem, and the relaxed variables are then adjusted to 0 or 1 according to different search mechanisms. The results are then compared with CPLEX solutions obtained with a 300-second time limit.

Another representative study on variables relaxation-based approaches is performed by Lei et al. (2009). The authors consider a single product multi-period PDP defined upon a M||B||J network with both forward and reverse flows. Because of the need to model the reverse flow in the supply chain network, new constraints such as

$$H_{i,t-1} + \sum_{\forall j} R_{i,j,t} - \sum_{\forall m} R_{m,i,t} = H_{i,t}, \quad \forall i,t$$

are added, where variable R refers to the reverse flow quantity, and H refers to the reverse product inventory levels. A partial LP relaxation-based rolling horizon method is proposed. With this approach, a given multi-period planning horizon is partitioned into

three intervals: the current period, the immediate next period, and a consolidated period covering all future time periods. In the first interval, all the original constraints and the integer requirements remain unchanged. For the second and the third intervals, only the integer requirements on the number of truck trips between the DC and customers are relaxed. To reduce the computational effort of each iteration, the forward and backward demands in the third interval are equal to the sum of the forward and backward demands of all the time periods in that interval, respectively. The ending inventories obtained from the solution to the first interval are then fixed as the beginning inventories for the second interval, and this process repeats by redefining intervals until all the time periods achieve integer solutions. Randomly generated test cases are used to benchmark the computational performance of the proposed algorithm against that obtained by the CPLEX within one-hour CPU time. Over 70 test cases are randomly generated, and the largest error gap observed is 0.16%, and the required computation time is less than 5 seconds; the average computation time required by CPLEX for solving these cases far exceeds 700 CPU seconds.

2.1.4 Remarks on PDP

In general, if the particular PDP problem being studied has a relatively simple structure, the well-known solution methodologies from the literature can often be effectively adapted. For example, when a PDP problem is defined on a two-stage supply chain network and the constraints are limited to those defined by (2.2) - (2.8), the original problem can be decomposed by either a sequential decomposition or Lagrangean

decomposition, which allows the decomposed problem to be modeled as an easy-to-solve problem such as the lot-sizing problem, or a linear programming or network flow problem.

While not included in this survey, it should be pointed out that in the literature, there has also been a significant amount of work focusing on production and distribution involving uncertainty in demand, processes, and/or supplies, for which stochastic and fuzzy models have been applied extensively. The difference between stochastic and fuzzy models is that a stochastic model usually follows a known probabilistic distribution, while a fuzzy model is described by a simple distribution, such as a triangular distribution, based on expert knowledge. Representative work in stochastic PDP can be found in studies by Park (2005), Aliev et al. (2007), Lejeune and Ruszczynski (2007), and Liang and Cheng (2009). Also note that while the exact methods have rarely been discussed in the literature for solving PDP problems, they could be appropriate if the problem has a special structure, such as that given by Wang et al. (2010).

2.2 PDP with Time Constraint (PDPT)

PDP with time constraints (PDPT) is a natural extension of the PDP model, which explicitly takes into account production and transportation time and usually assumes a deadline for the shipment arrival to the customer. To define the shipment arrival times, additional notation must be introduced. Let r_m^k be the production rate for product k at manufacturer m. Let $\tau_{m,i}$ and $\tau_{i,j}$ be the transportation times from manufacturer m to DC *i*, and from DC *i* to customer *j*, respectively. Let $L_{j,t}$ be the deadline at customer site *j* in period *t*, by which time the shipment of commodities should have arrived at *j*; otherwise a shortage or tardiness cost would be incurred. Let *MM* be a very large positive number. The deadline constraints are defined as follows.

$$\frac{P_{m,t}^{k}}{r_{m}^{k}} + \tau_{m,i} + \tau_{i,j} - (3 - Z_{m,t}^{k} - W_{m,i,t} - W_{i,j,t})MM \le L_{j,t} \quad \forall m, i, j, k, t$$
(2.26)

The basic PDPT model is defined by (2.1) - (2.8) and (2.26).

Some papers study PDPT problems involving production lead times and delivery lead times over a multi-period planning horizon. Let $l_{m,i}$ and $l_{i,j}$ represent lead times from manufacturer *m* to DC *i*, and from DC *i* to customer *j*, respectively. In this case, (2.2) - (2.4) should be replaced by the following constraints.

$$I_{m,t-1}^{k} + P_{m,t}^{k} - \sum_{\forall i} Q_{m,i,t}^{k} = I_{m,t}^{k}, \qquad \forall m,k,t \ (2.27)$$

$$I_{i,t-1}^{k} + \sum_{\forall i} Q_{m,i,t-l_{m,i}}^{k} - \sum_{\forall i} Q_{i,j,t}^{k} = I_{i,t}^{k}, \qquad \forall i, k, t \quad (2.28)$$

$$I_{j,t-1}^{k} + \sum_{\forall i} Q_{i,j,t-l_{i,j}}^{k} - (d_{j,t}^{k} - S_{j,t}^{k}) = I_{j,t}^{k}, \qquad \forall j, k, t \quad (2.29)$$

Due to the complexity of PDPT, using a single methodology, such as a Lagrangean relaxation, or a simple heuristic algorithm, may not be effective enough to solve the problem. In the literature, two major approaches have been discussed. One is iteration-based, and starts with an initial solution (or a group of solutions), and then continuously improves the solution (or a set of solutions) iteratively by a relatively

simple procedure; most metaheuristic-based algorithms belong to this category. The other is to formulate the original problem into a mathematical model and then use optimization software to derive the optimal or near-optimal solutions. The latter approach has typically been used for solving some case-specific problems.

There are also several papers using simulations to deal with PDPT involving uncertainty. Most such studies (e.g., Lee et al., 2002; Lee and Kim, 2002; and Safaei et al., 2010) start with a deterministic version of the problem and solve it to find an initial solution. Through simulation, the solution is evaluated and the parameters of the respective deterministic problem are modified until the solution stabilizes. In this survey, we only include such simulation studies that report on the approaches to solve respective deterministic versions of the PDPT problem.

In this section, we focus on the existing solution methodologies for solving PDPT. Two categories of solution approaches are reviewed: 1) *metaheuristic and iterative approach*, and 2) *mathematical modeling and the use of an optimization solver*. Again, we do not consider detailed routing decisions in this section, and hence we treat all transportation operations as direct shipping or fixed routing.

2.2.1 Metaheuristic and Iterative Approach

Naso et al. (2007) consider the integrated problem of finding an optimal schedule for the just-in-time (JIT) production and delivery of ready-mixed concrete with manufacturers and customers. The study involves a single product in a single period with no inventory permitted. Times required for the loading, unloading and shipping operations of each truck must be explicitly modeled. In addition, outsourcing options of production and third-party (or overtime) trucks are permitted at an additional cost. All decision variables are binary, where $x_{jvr} = 1$ if job *j* is assigned to truck *v* as the *r*-th task: $y_{mj} = 1$ if job *j* is produced at manufacturer *m*, and $y_{oj} = 1$ if job *j* is outsourced. The scheduling algorithm combines a GA and a set of constructive heuristics, which are guaranteed to terminate in a feasible schedule for any given set of jobs.

Gebennini et al. (2009) consider a multi-period strategic and operational planning problem for a single manufacturer that offers a single product with uncertain demand on an M||B||J network. Production lead times and delivery lead times are considered, where lead time may be an integer multiple of one time period, and inventory and stockout costs are considered with safety stock (SS) determination. Thus, the problem to minimize the total cost is modeled as a mixed-integer non-linear programming problem in which the objective function includes a non-linear term representing the SS cost, $\sum_{i\in B} c_i^s \hat{k} \sqrt{\sum_{j\in J} \hat{\sigma}_{ij}^2 \theta_{ij}}$ where c_i^s is the inventory cost for DC *i*, \hat{k} is safety factor to control the customer service level, $\hat{\sigma}_{ij}^2$ is the combined variance at DC *i* serving customer *j*, and θ_{ij} is a 0-1 decision variable equal to 1 if DC *i* supplies customer *j* in any time period. This non-linear term is linearized to $\sum_{i\in B} \sum_{j\in J} c_i^s \frac{1}{SS_i} \hat{k}^2 \hat{\sigma}_{ij}^2 \theta_{ij}$ where SS_i is a lower bound on the optimal amount of SS carried at DC *i*, because the closer SS_i is to the optimal SS level at DC *i*, the closer the formula is to the optimal SS cost. A recursive procedure based on the modified linear model is developed in order to find an admissible solution to the non-linear model and quantify the minimized global logistic cost, while also taking the effect of safety-stock management into consideration. Since the optimal safety-stock level is unknown, the value is initially set to a lower bound on the effective safety-stock quantity for each DC. It is claimed that the proposed recursive procedure converges on the global optimal solution of the original non-linear problem in a finite number of iterations.

Yimer and Demirli (2010) address a multi-period, multi-product scheduling problem in a multi-stage build-to-order supply chain manufacturing system with consideration of lead times for production and delivery. For the sake of efficient modeling performance, the entire problem is first decomposed into two sub-problems: 1) a downstream part: from manufacturers through distributors and retailers to customers, and 2) an upstream part: from suppliers through fabricators to manufacturers. Both sub-problems are then formulated as MIP models with the objective of minimizing the associated aggregate costs while improving customer satisfaction. A GA-based heuristic is proposed with a chromosome of three parts: 1) product ID, total production quantity at each plant, and inventory level at each DC in the period; 2) flow proportion floating values; and 3) status values for feasibility. If a candidate solution is infeasible, it is revised by a proposed repair heuristic. The fitness value is measured by the original objective function value and the degree of infeasibility. Using some test instances, the best solutions obtained from GA are of high quality compared with the lower bounds obtained from LINGO, a non-linear programming solver.

Sabri and Beamon (2000) develop an integrated multi-objective supply chain model that facilitates simultaneous strategic and operational planning using an iterative method in a four-tier network. They consider stochastic demand and capacity constraints in all layers of the supply chain, and shortages are allowed, but penalized, while a fixed setup production cost is incurred. Total production lead time at manufacturer *m* for product *k* is $g_m^k + \frac{Q_m^k}{r_m^k} + l_m^k + \theta_m^k$ where $g_m^k, Q_m^k, r_m^k, l_m^k$ and θ_m^k are production setup time, production quantity, production rate, waiting time, and material delay time, respectively. θ_m^k is determined by the bill of material of product *k* and customer service level. They first find a solution for the strategic model and then use the solution as an input to solve the operational model. New parameters determined in solving the operational model are used to solve the strategic model, and this iteration terminates when all binary variables no longer change. LINGO is used in solving each sub-problem.

2.2.2 Mathematical Modeling and the Use of Optimization Solver

While some researchers try to develop effective solution methodologies to solve the PDPT, others put more effort into the modeling process. In this subsection, we summarize research in which the models are solved by mathematical optimization software such as CPLEX. The common feature of the following papers is that the authors concentrate on the models rather than the design of methodologies. The size of the computational testing

instances is small enough for the solver to handle, or the problem comes from real world practice so that the solution by a solver is applicable.

Rizk et al. (2006) examine a multiple-product production-distribution planning problem on a single manufacturer and a single destination. The manufacturer operates a serial production process with a bottleneck stage, subject to a predetermined production sequence. The manufacturing cost consists of the changeover cost of intermediate products and the inventory holding cost of final products. The transportation cost is characterized by a general piecewise linear function of transportation quantity with break points of Λ_h with $\Lambda_0 = 0$. In the *h*-th interval $(\Lambda_{h-1}, \Lambda_h]$, let v_h be the slope of its straight line, A_h be the discontinuity gap at the beginning of the interval and E_h be the ending value. Thus, the transportation cost is $z(\Lambda) = (E_{h-1} + A_h) + v_h \lambda_h$, $\lambda_h = \Lambda - \Lambda_{h-1}$. Valid inequalities to strengthen these formulations are proposed and the strategy of adding extra 0-1 variables to improve the branching process is examined.

Chen and Lee (2004) investigate a multi-period simultaneous optimization of multiple conflict objectives with market demand uncertainties and uncertain product prices in a supply chain network consisting of manufacturers, DCs, retailers and customers. The scenario-based approach is adopted for modeling the uncertain market demands, and the product prices are taken as fuzzy variables where the incompatible preference on prices for different participants are handled simultaneously. The whole model becomes a mixed-integer non-linear programming problem to compromise fair profit distribution, safe inventory levels, maximum customer service levels, and decision robustness to uncertain product demands. Considering incompatible preference of product prices for all participants will be determined by applying the fuzzy multi-objective optimization method, non-linear MIP solvers, DICOPT and CONOPT, are used for the numerical example.

Dhaenens-Flipo and Finke (2001) provide a multiple period model on an M||B||Jnetwork which comes from a practical case at the European industrial division of the manufacturer. Since switching from one product to another on a production line may take a long time, it is assumed that at most one switching per period and per production line is allowed. There are three aggregated products and three line types according to capability to handle these products. All possible sequences in each manufacturing line are enumerated, and they are used in a mixed integer programming model. The set of available product sequences of the line *m* is denoted by S(m) and these sequences are indexed by *s*. At this stage, the data involved concerns the total production time (B_m) available on line *m*, the production time (TP_m^k) and cost (CP_m^k) of product *k* on line *m*, the changeover time (TC_{sm}) and the cost (CC_{sm}) associated with the products of sequence *s* on line *m*. Let p_m^k be a quantity of product *k* manufactured on line *m*, and let y_{sm} be 1 if sequence *s* is chosen for the line *m*. Thus, we need to add following constraints:

$$\sum_{s \in S(m)} y_{sm} = 1 \text{ for } \forall m \tag{2.30}$$

$$p_m^k - \sum_{s \in S(m):k \in s} y_{sm} \times B_m / TP_m^k \le 0 \quad \text{for} \quad \forall m, \forall k$$
(2.31)

$$\sum_{k} p_{m}^{k} \times TP_{m}^{k} + \sum_{s \in S(m)} y_{sm} \times TC_{sm} \le B_{m} \text{ for } \forall m$$
(2.32)

The proposed MIP has constraints (2.30) - (2.32), flow balance equations similar to (2.2)

- (2.4), and domain constraints. For problems of industrial sizes, the model is able to provide a sub-optimal solution in less than 2 hours (23 minutes on the average) by CPLEX.

Fahimnia et al. (2008a) survey 20 papers and define a representative mixed integer program formulation for the integration of an aggregate production and distribution plan on an M|B||J network. Three production alternatives are considered: regular-time production, overtime production, or outsourcing. They illustrate with an example to show that considering production alternatives can give a more accurate and better schedule than considering average production cost.

2.2.3 Remarks on PDPT

Lagrangean relaxations and decomposition-based techniques are not effective for solving the general PDPT problems because newly added time constraints often change the model structure significantly. The production and transportation time as well as the incurred deadline constraints all add more complexities to the original PDP, since a feasible solution for a PDP may violate the deadline constraint in PDPT. Even after a PDPT is decomposed, the resulting sub-problems may still be NP-hard and therefore make Lagrangean relaxation and decomposition-based solution approaches fail to function effectively. Therefore, most literature results reported are either customized solution approaches for specific PDPTs or efficient algorithms for solving some special cases of PDPT.

2.3 PDP with Routing (PDPR)

PDP with routing (PDPR) will be discussed in this section. Because of its complicated structure, most papers assume a two-stage network and thus those problems can be considered as a combination of the capacitated lot-sizing problem and the inventory routing problem. The aim of the problem is to minimize the total cost composed of inventory holding, production and transportation costs.

We consider a basic model defined upon a two-echelon supply chain consisting of a set of manufacturers and a set of customers, where customer *j* has demand d_{jt} in period *t*. For simplicity, a single product is considered and thus the superscript for product type (*k*) is dropped. We assume that there is a fleet of homogenous vehicles belonging to manufacturer *m*, denoted by *V*(*m*). Since the PDPR model contains routing decisions in it, the quantity being carried by a vehicle is different from the quantity delivered to a customer by a vehicle in a period. Thus, the following parameters and decision variables are added to PDP:

 f_{njlt}^{v} = fixed cost of vehicle v of manufacturer m along (j, l) in period t

 g_{mjlt}^{v} = unit shipping cost for vehicle *v* of manufacturer *m* along (*j*, *l*) in period *t* ξ_{mjlt}^{v} = equals 1 if vehicle *v* of manufacturer *m* serves *l* immediately after *j* in period *t* Q_{mjlt}^{v} = quantity carried by vehicle *v* of manufacturer *m* along (*j*, *l*) in period *t* q_{mjt}^{v} = quantity delivered by vehicle *v* of manufacturer *m* to customer *j* in period *t* for $m \in M, j \in \{m\} \cup J, l \in \{m\} \cup J, t \in T$. The objective function of the model consists of production, inventory and transportation (routing) costs. The transportation cost is changed as follows:

$$\sum_{t} \sum_{m} \sum_{v \in V(m)} \sum_{j,l \in \{m\} \cup J, j \neq l} f_{mjlt}^{v} \xi_{mjlt}^{v} + g_{mjlt}^{v} Q_{mjlt}^{v}$$
(2.33)

Moreover, routing constraints should be included in the model. The flow conservation constraints are:

$$\sum_{l \in \{m\} \cup J, l \neq j} \mathcal{Q}_{mjlt}^{\nu} - \sum_{l \in \{m\} \cup J, l \neq j} \mathcal{Q}_{mljt}^{\nu} = -q_{mjt}^{\nu} \qquad \forall m, \nu, j, t$$
(2.34)

$$\sum_{j\in J} \mathcal{Q}_{mjmt}^{\nu} - \sum_{j\in J} \mathcal{Q}_{mmjt}^{\nu} = -\sum_{j\in J} q_{mjt}^{\nu} \qquad \forall m, \nu, t$$
(2.35)

We need an inventory balance constraint for each customer.

$$I_{j,t-1} + \sum_{m} \sum_{v \in V(m)} \sum_{j \in J} q_{mjt}^{v} - d_{jt} = I_{j,t} \qquad \forall j,t$$
(2.36)

Since ξ_{mjlt}^{v} represents the existence of flow on (j, l) and each customer can be served by at most one manufacturer, we have the following constraints:

$$Q_{mjlt}^{\nu} \leq \xi_{mjlt}^{\nu} MM \qquad \qquad \forall m, \nu, t, \ j, l \in J \qquad (2.37)$$

$$\sum_{m} \sum_{\nu \in V(m)} \sum_{l \in \{m\} \cup J, l \neq j} \xi_{mljt}^{\nu} \le 1 \qquad \forall j, t$$
(2.38)

We classify the relevant papers, according to their solution methodologies, into three classes. Since the problem includes the routing decisions, all methods use decomposition. However, each decomposed problem is solved by a different solution approach. One approach is to use mathematical programming or simple heuristic algorithms. The other two are to use a metaheuristic, such as a tabu search, and the approximation approach, respectively.

2.3.1 Mathematical Programming Approach

Fumero and Vercellis (1999) study a multiple period and multiple product problem with a single manufacturer. They assume that there are fixed setup costs and vehicle usage costs which occur independently from the amount of produced or carried product. In the model, partial order serving is allowed. They decompose the problem into production (capacitated lot-sizing) and distribution (multi-period vehicle routing) problems by using Lagrangean relaxation, relaxing the constraints which ensure the balance at the central plant among production, inventory and deliveries. Furthermore, the vehicle capacity constraints are relaxed in order to simplify the solution of the routing sub-problem. The Lagrangean dual problem is solved by using a variable target subgradient optimization algorithm which is described in Fumero (1997). Additionally, they employ an alternative decomposition method in which the production plan is developed without considering the distribution plan, and then used as an input for the distribution model. They show that the Lagrangean decomposition method outperforms the alternative decomposition method.

Bard and Nananukul (2010) propose a hybrid methodology which is a combination of an exact method and heuristic procedures within a branch-and-price (B&P) framework for the problem with a single manufacturer and a single product type. The master problem (MP) is defined by the production and inventory decisions, and the remaining routing problem can be decomposed by period, yielding |T| sub-problems. In the reformulated model, each column in the MP corresponds to a feasible schedule for all customers. They use a novel column generation heuristic and a rounding heuristic in order to improve the algorithmic efficiency. They show that the B&P heuristic is efficient and can derive high-quality solutions for large problems within a reasonable amount of time.

Ruokokoski et al. (2010) consider the problem of determining a production schedule for an uncapacitated plant, replenishment schedules for multiple customers, and a set of routes for a single uncapacitated vehicle. The aim of the problem is to fulfill customer demand over a finite horizon at a minimum total cost of distribution, setups, and inventories. This paper introduces a basic mixed integer linear programming formulation and provides exact methods through several strong reformulations of the problem. Moreover, two families of valid inequalities, 2-matching and generalized comb inequalities, are introduced to strengthen these formulations, and they are used within a branch-and-cut framework. Comb inequalities are known to be facets for the traveling salesman problem (Grötschel and Padberg, 1979) and 2-matching inequalities are generalized comb inequalities under certain conditions. An a priori tour-based heuristic is also provided and, with available solvers and strong formulations, excellent solutions can be obtained within a short time, even for the largest problems.

Archetti et al. (2011) consider a production-routing system, where a manufacturer with unlimited capacity produces one product, which is distributed to a set of retailers by a fleet of vehicles. The objective is to determine the production policy, the customer replenishment policy and the transportation policy from among two different types of policies: maximum level (ML) and order-up to level (OU), with minimum total cost. A three-step sequential heuristic is proposed on the ML policy. In the first step, unlimited production quantity is assumed, and the distribution part of the problem concerning inventory at customers and delivery routes is optimized by solving a customer problem with branch-and-cut and iteratively adding it to the solution. In the second step, the production plan is determined by solving the classical uncapacitated lot-sizing problem, which can be optimally solved in polynomial time. In the third step, the improvement procedure, removing and inserting two retailers, is repeated until there is no further improvement.

Cetinkaya et al. (2009) consider a three-layer practical supply chain problem and develop a multi-product and multi-period model to improve the outbound supply chain of Frito Lay North America (FLNA), consisting of a factory warehouse, multiple DCs, and a set of customers. Some customers can receive supplies directly from the factory warehouse, which is called direct delivery (DD). They do not consider the production costs but the production capacities. The objective function contains the inventory holding cost, the truck loading and dispatch cost, mileage costs, and handling costs. The proposed solution methodology decomposes the integrated problem into two sub-problems - inventory and routing problems - and they are iteratively solved until either no further improvement is found or the maximum number of iterations is reached. The routing sub-problem is solved period by period. As a preprocessing, they use full-truck load (FTL) shipments with a route having a single destination for customers

with large order quantities, and use less-than-truck load (LTL) shipments with truck routes for other customers. They then use a savings algorithm proposed by Clark and Wright (1964) and utilized by Chopra and Mendl (2001) and add an improvement step, called the cheapest insertion heuristic, a well-known travelling salesman problem heuristic. For inventory sub-problem, the objective function includes the corresponding route-based setup costs and all cost terms of the overall model, except the loading and routing parameters considered in the routing sub-problem. The CPLEX 9.0 solver is used to solve the inventory sub-problem.

2.3.2 Metaheuristic Approach

Bard and Nananukul (2009a) consider the problem of a B||J network where inventory handling at both the customer and manufacturer sites is permitted, but the inventory level must be zero at the end of the each period, with no shortages allowed. They solve the problem by using a two-phase approach, which is similar to the method developed by Lei et al. (2006). In the first phase, they formulate the model as a mixed integer program without taking into account the routing constraints. They find a feasible solution which determines the sufficient delivery amounts for all customers using the proposed model. The solutions derived in the first phase are used as an initial solution for the tabu search algorithm, which is used in the second phase to solve the integrated problem. The path relinking method is used to obtain better solutions. They show that the lower bounds obtained from the relaxed version in the first phase are not very effective for evaluating the proposed algorithm. However, according to the computational results, the proposed method can derive 10-20% better solutions, but requires more computational effort than the GRASP (greedy randomized adaptive search procedure) proposed by Boudia et al. (2007).

Bard and Nananukul (2009b) propose three algorithms with a B&P framework for the Inventory Routing Problem (IRP) as a sub-problem of the integrated production–inventory–distribution–routing problem. For less computation, a two-step procedure is proposed: the first step involves developing a model for determining delivery quantities for each customer in each period. The second step involves finding actual routes in light of the current set of branching constraints with a vehicle routing problem (VRP) tabu search method. According to computational experiments, while the B&P algorithm generates better results than the tabu search approaches (3.6% on average), the tabu search outperforms the B&P algorithm in terms of the computation time (more than ten times faster on average).

Yossiri et al. (2012) develop a decomposition heuristic based on an adaptive large neighborhood search (ALNS) for the problem defined on a network consisting of a plant and multiple customers to minimize the total production, setup, inventory and routing costs. In the first stage, a set of initial solutions are generated with different setup schedules by solving two sub-problems: 1) production and distribution problem with approximate transportation costs, and 2) routing problem; both are solved heuristically. In the second stage, the initial solutions are improved by ALNS. When a solution is modified by removing a customer from a route and inserting it in a different period, one

has to identify the new delivery quantity for the customer, which may also affect the production, inventory, and other delivery quantity decisions. It is not always necessary to reinsert the removed nodes, because the demands can be satisfied from available inventory and, furthermore, the removed nodes can be inserted in multiple periods. To deal with these issues, binary variables are defined accordingly. During the transformation process, the binary decisions concerning routing are modified according to the cheapest insertion rule and then, with fixed binary variables concerning production setup, the continuous variables are adjusted by solving the minimum cost flow problem.

2.3.3 Incorporating Routing Cost Approximation for Solving PDPR

When the decomposition method is applied, a PDPR problem is usually solved through two phases. During the first phase, a reduced version of PDPR is solved, where many studies assumed direct shipments to customers (e.g., Lei et al., 2006); and then during the second phase, vehicle routing decisions are made to improve the solutions obtained in the first phase. The advantage of such a phased approach is to reduce the search complexity in each phase. However, using direct shipment to replace vehicle routing in the first phase can sometimes also lead to a solution that is feasible but deviates significantly from the optimal solution to the original problem.

Another alternative solution approach to PDPR is based on continuous approximation models for the vehicle routing problems. Such an approach uses a continuous approximation of the optimal routing cost in the phase-one problem instead of assuming direct shipments. Note that this provides an estimation of the actual routing cost without explicitly solving the vehicle routing problem. Once the phase-one problem is solved and the assignments of vehicles to customers are determined, the exact routing decisions under the given vehicle assignments are made during the second phase.

Shen and Qi (2007) incorporate a continuous approximation function in their integrated supply chain design model to estimate the optimal vehicle routing cost. Specifically, the approximate function that they propose is

$$V_{mvt} = \frac{2}{C^{v}} \sum_{j \in J} q_{mjt}^{v} \mu_{mjt}^{v} + \left(1 - \frac{2}{C^{v}}\right) \Phi \left|v_{t}\right| \sqrt{\frac{A}{|J|}}$$

where

 V_{mvt} = the approximate routing cost of vehicle v of manufacturer m in period t

 C^{v} = the capacity of vehicle v of manufacturer m

 q_{mjt}^{v} = the quantity delivered by vehicle v of manufacturer m to customer j in period t

 μ_{mjt}^{v} = the unit cost of a direct shipment by vehicle v of manufacturer m to customer j in period t

 $|v_t|$ = the number of customers served by vehicle v of manufacturer m in period t

- A = the area where customers are scattered
- Φ = parameter, and Φ = 0.75 for Euclidean metrics

Shen and Qi (2007) numerically demonstrate the effectiveness of this approximation function using a data set with 150 points from Christofides, Mingozzi, and Toth (1979),

and show that this approximation function performs especially well when the number of customers is sufficiently large. In particular, when the number of customers is more than 80, the approximation error is typically less than 5%.

When the above continuous approximation function is incorporated in the phased approach, parameters q_{mjt}^{v} and $|v_t|$ vary with the decision of assignments of vehicles to customers, while all the remaining parameters are given constants. Compared to the direct shipment assumption that is often made in the literature, this approximation function provides a more accurate estimation of the routing cost without increasing the problem complexity. This approach may be used as an alternative to further enhance the performance of phased approaches.

2.3.4 Remarks on PDPR

In this section, the total cost of the PDP with routing is minimized, where the total cost is composed of inventory holding, production and routing cost. Since the problem includes the vehicle routing problem, it is very difficult to find the optimal solution or an approximate solution close to the optimum. Thus, most algorithms use a decomposition approach and metaheuristic algorithms, such as a tabu search, to solve routing sub-problems. When there is a single manufacturer, the decomposition approach is frequently used because the upstream problem can be regarded as a capacitated lot-sizing problem. Moreover, after obtaining a solution, various improvement heuristics are also often used as post-processing procedures. Since the optimal value is most likely

unavailable, the performance of an algorithm is presented by comparing its solution with a lower bound, or with a solution obtained by either previous approaches or an optimization solver.

2.4 PDP with Routing and Time Constraints (PDPRT)

PDP with routing and time constraints (PDPRT) will be discussed in this section. Time constraints appear in different forms, such as time window, due date, and exact arrival time predetermined by customers. It can be considered as a combination of an inventory routing problem (IRP) with time constraints and a capacitated lot-sizing problem.

In most of the existing literature, a two-echelon supply chain which contains a single plant and a set of geographically scattered customers is considered. Due to the complexity of the problem, multiple manufacturers are rarely considered (see Lei et al., 2006; and Bilgen and Gunther, 2009). Generally, the objective function contains the production cost, the transportation cost (routing cost) and the inventory holding cost. On the other hand, minimizing the makespan consisting of production time and transportation time, and maximizing the satisfied demand are considered as objectives in Geismar et al. (2008) and Armstrong et al. (2008), respectively. Although third party vehicles are rarely considered, Lei et al. (2006) take third party transshipments into account. We consider a representative model with a two-echelon supply chain network consisting of a set of manufacturers and a set of customers. For the sake of simplicity, we assume that customer j has demand d_{jt} with due date L_{jt} in period t.

We assume that there is a fleet of homogenous vehicles belonging to each manufacturer. In order to deal with time constraints we need to define additional parameters and variables. The objective function and constraints other than time constraints are equivalent to those in the model in section 5. Thus, we focus only on time constraints:

$$\tau_{mjlt}^{v}$$
 = travel time of vehicle v of manufacturer m on arc (j, l) in period t
 T_{mit}^{v} = arrival time of vehicle v of manufacturer m at customer j in period t

In order to guarantee due date constraints, we have the following additional constraints:

$$T_{mjt}^{\nu} + \tau_{mjlt}^{\nu} \le T_{mlt}^{\nu} + MM(1 - \xi_{mjlt}^{\nu}) \qquad \forall m, \nu, t, \ j, l \in J, j \neq l$$
(2.39)

$$T_{mlt}^{\nu} \le L_{lt} \qquad \qquad \forall m, \nu, t , \quad l \in J \qquad (2.40)$$

The solution methodologies used to solve this problem in the literature fall into two different groups, according to their structures; the first one solves the problem in an integrated manner, while the second one partitions the problem into small pieces which are easier to solve. In these decomposition methods the solution from the first phase is used as an input to the second phase. Using integrated methods, the solution may be improved by an iterative process.

Chandra and Fisher (1994) solve the production and transportation scheduling

problems in separate and integrated manners and compare those results. In their model, the plant can produce several products in a limited time and transporters are allowed to partially deliver to a set of customers with unlimited capacity in a period. The plant has an unlimited production capacity and the inventory holding costs are not involved in the total costs. First, they implement their integrated approach in small examples and show that firms can reduce their operation costs about 3-20 % by coordinating their production and distribution activities. Second, in the decomposed part, they assume that the production scheduling problem can be modeled as a capacitated lot-sizing problem and the distribution problem can be modeled as a standard multi-period local delivery routing problem. The interface of GAMS, ZOOM/XMP, a solver, is used to solve the production scheduling problem. They use three well-known vehicle routing heuristics - sweep (Gilette and Miller, 1974), nearest neighbor rule (Rosencrantz et al., 1974) and feasible insertion rule (Chandra, 1989) - in order to find an initial solution to the distribution problem. A local improvement heuristic is used to combine the production and distribution problems. Since the work of Chandra and Fisher (1994), many extended studies have been conducted with various approaches including decomposition and compounded methods.

2.4.1 Decomposition Methods

Using decomposition methods, the problem is usually partitioned into two sub-problems - production planning and routing problems - which are solved sequentially.

Lei et al. (2006) investigate an integrated production, inventory and distribution routing problem where there is no fixed cost of using a vehicle, and each transporter can make multiple trips during each period. They use a two-phased approach that solves the problem in two separate stages but in an integrated manner. In the first phase, they assume that the distribution of the products from plants to customers is carried out by direct shipment. The problem is formulated as a mixed integer programming problem, neglecting the vehicle routing constraints, and solved by the CPLEX MIP solver. In the second phase, they propose a heuristic transporter routing algorithm, called the Load Consolidation (LC) algorithm, to consolidate the loads into routing decisions. The LC algorithm determines the sequence of transporter trips and allocates the transporters to the trips without violating the transporter capacity and available time constraints. The EOP (Extended Optimal Partitioning) procedure is used in order to find the shortest path among the feasible trips which are identified in the first phase. They compare the LC algorithm and CPLEX MIP solver with 56 test problems. According to their test results, the LC algorithm can solve the problem in less than 0.2 seconds while the CPLEX MIP solver needs more than 2 hours to solve the overall problem.

Geismar et al. (2008) develop a two-phase heuristic to solve a single period integrated production and transportation scheduling problem for a product with a short life span. The first phase uses either a genetic algorithm (GA) or a memetic algorithm (MA) to select a locally optimal permutation of a given set of customers. MAs have a local search parameter and a relatively small population size as a result of different population management. In the second phase, for a given permutation of customers, Beasley's (1983) "first route-cluster second" method is used to simultaneously determine the customers to be served and the vehicle routes to be used, and a linear program formulation is used to minimize the makespan for a given set of trips. The Gilmore-Gomory (1964) algorithm for two machine no-wait flowshops is then used to order the subsequences of customers to form the integrated schedule.

2.4.2 Integrated Methods

Among the papers dealing with integrated methods, some papers propose problem-specific methodologies for problem-solving, while others focus on new modeling techniques.

Boudia and Prins (2009) examine a multi-period production distribution problem in a two-echelon supply chain which is very close to the model proposed by Chandra and Fisher (1994), but differs in that the limited vehicle capacity and a single product are considered. They use a memetic algorithm with population management (MA/PM) to handle production and distribution problems simultaneously. The proposed algorithm is evaluated in three sets of 30 instances with 50, 100 and 200 customers over 20 periods. They compare the proposed algorithm with two previous algorithms: the two-phase algorithm (H1) proposed by Boudia et al. (2005) and the three-phase algorithm (H2) based on GRASP developed by Boudia et al. (2007). They show that the memetic algorithm can generate better solutions than GRASP, which also solves the related

problem from an integrated perspective.

Armstrong et al. (2008) solve a similar problem with a branch-and-bound search procedure in order to maximize the total satisfied demand by choosing a subset of customers from the given sequence who will be served by a single vehicle. The constraints of the problem refer to the product lifespan, the production/distribution capacity, and the delivery time window. Since there is no inventory handling at the supply chain members, it is important to synchronize the production and distribution planning decisions successfully. Empirical studies on the computational effort required by the proposed search procedure comparing to that required by CPLEX on randomly generated test cases are summarized. A branch-and-bound search algorithm is also proposed and is shown to outperform CPLEX with limited running time.

Bilgen and Gunther (2009) consider an integrated production and distribution planning problem in the fast-moving consumer goods industry, with a so-called block-planning approach, which establishes cyclical production patterns defined by setup families. The aim is to minimize the total cost, consisting of production costs, inventory holding costs at distribution centers, and transportation costs for FTL and LTL transportation modes. Unlike the other related studies, they consider two types of production setup cost - major setup costs for each block started on one of the lines (e.g. for clean-out in the food industry), and minor setup costs for the production lots of the individual products. They trace the time in terms of the block and lot production completion times. Two different periods are used in this study: macro periods (e.g. weeks) are used for the block assignments and micro periods (e.g., days) are used for the distribution schedule and external demand elements. They compare two different block-planning approaches: the flexible and the rigid block which differ by their degree of flexibility in the scheduling of the production lots. A mixed-integer linear programming model is proposed to solve the problem and CPLEX is used as a solver. The numerical results reveal that the flexible block-planning approach can provide considerable cost savings compared to the rigid block-planning approach.

Bolduc et al. (2010) consider the split delivery vehicle routing problem with production and demand calendars. They propose a simple decomposition procedure to provide a starting solution and use a tabu search with new neighbor reduction strategies. After the tabu iterations are completed, an improvement heuristic is applied. They implement their procedure on a randomly generated 100 instances with 50 customers and 10 periods. The results show that the developed model is effective in terms of both solution quality and computation time.

2.4.3 Remarks on PDPRT

In the decomposition method, there are two general approaches: the first one considers the production problem and the routing problem separately, while the second solves the problem including production and simplified distribution and then solves the routing problem. For the integrated method, there are two approaches. The first is to solve the problem simultaneously using mathematical programming with an optimization

package, while the second is to use an iterative method in which the solution is improved over iterations through a metaheuristic such as GA and tabu search. Even though there is no clear dominance between the decomposed method and the composed method, the decomposed method is always useful to find an initial feasible solution. For example, Bolduc et al. (2010) use a decomposed method to find an initial solution and then improve it by a tabu search algorithm in an integrated manner.

Since the problem is already complicated by including the vehicle routing problem in it, researchers have focused on a two-echelon problem with static demand. Thus, natural generalizations are required, such as two echelons to multiple echelons, static demand to stochastic demand, and excluding third party to including third party.

2.5 PDP in Emergency Logistics (PDPEL)

Since the first application of PDP to emergency logistics in the 1970s, many publications have appeared focusing on emergency operations management (Caunhye, Nie, and Pokharel 2012). Among those that are closely related to our work, Haghani and Oh (1996) studied the operations scheduling problem of a large-scale multicommodity, multi-modal distribution network with time window constraints. The authors proposed two heuristics to solve the resulting mixed integer programming (MIP) model, which differs from the one we study here in that only non-renewable resources were considered. Ozdamar, Ekinci, and Kucukyazici (2004) studied a similar emergency logistic planning problem encountered during natural disasters. Since the supply-demand relationships at

the different locations shift during the emergency logistics process, the decision plan including both vehicle routing problem (VRP) and network flow problem needs to be generated frequently. A Lagrangian relaxation-based approach was used to find approximate solutions. Yi and Kumar (2007) studied the distribution problem of a large-scale multi-commodity, multimodal network flow model in which medical supplies are transported to distribution centres (DCs) in disaster areas. An iterative two-phased solution approach was proposed, where the algorithm constructs stochastic routes in the first phase, and develops a network flow-based algorithm for the multi-commodity dispatch in the second phase under the given vehicle routes. Nolz, Semet, and Doerner (2011) discussed different risk measures and provided a memetic algorithm based approach to solve a multi-objective mathematical model for the distribution of emergency supplies after disasters. Yuan and Wang (2009) considered two models of a path selection problem in emergency settings, and proposed a Dijkstra algorithm-based approach to solve the single-objective model and an ant colony algorithm to solve the multi-objective model. Afshar and Haghani (2012) studied a mathematical model based on FEMA's three-layer logistics structure. Their study incorporated both detailed routing for emergency supply deliveries and location selection for operating facilities. The authors also pointed out that a good heuristic for solving this complex problem is needed at the next research step. Comprehensive reviews in this area can be found in the work by Wright et al. (2006), Balcik et al. (2010), Hartmann and Briskorn (2010) and more recently Huang, Smilowitz, and Balcik (2012) and Caunhye, Nie, and Pokharel (2012). The survey by Caunhye, Nie, and Pokharel (2012) reviewed more than 70 papers on the management of disaster relief operations that are performed either before or after the

impact of a disaster. However, as indicated in Hartmann and Briskorn (2010), there still remains a lot of work to be done on the joint allocation of both renewable and nonrenewable resources.

It should also be noted that the distinction between renewable and non-renewable resources is quite common in resource allocation and assignment problems (e.g. Ait-Kadia, Menye and Kane 2011) and in resource-constrained project scheduling (e.g. Brucker et al. 1999). In the resource assignment and project scheduling literature, however, very few papers have dealt with both renewable and non-renewable resources, and most of these have used heuristic approaches as solution methods. Representative studies in this area can be categorised as follows. Resource assignment results can be found in Ait-Kadia, Menye and Kane (2011), Bachlaus, Tiwarib, and Chan (2009), Celano, Costa, and Fichera (2008), Eckstein and Rohleder (1998), Hwang and Kogan (2003) and Karsu and Azizoglu (2012). Resource-constrained project scheduling with a single type of resource can be found in Brucker and Kramer (1996), Chan, Wong, and Chan (2006), Debels and Vanhoucke (2007), Deblaere et al. (2007), Deblaere, Demeulemeester, and Herroelen (2011a, 2011b), Depuy and Whitehouse (2001), Klein (2000), Ranjbar, Reyck, and Kianfar (2009), Robinson and Moses (2006), Schirmer (2001), Tormos and Lova (2003), Vanhoucke, Demeulemeester, and Herroelen (2001b) and Van de Vonder et al. (2006). Resource-constrained project scheduling with both renewable and non-renewable resources can be found in Bottcher et al. (1999), Vanhoucke, Demeulemeester, and Herroelen (2001a), Lee and Lei (2001) and Nudtasomboon and Randhawa (1997). Some papers have dealt with multi-mode resource-constrained project scheduling with renewable as well as non-renewable resources such as those by Gagnon, D'Avignona, and Boctor (2009), Can and Ulusoy (2010) and Wong, Chan, and Chung (2012). To our knowledge, most of the methodologies proposed in project scheduling with resource constraints are heuristic-based approaches.

A comprehensive classification of existing results in PDPEL based on model constraints, resource types considered, and solution methodologies is described in the following table.

	Constraints			Resource		Methodology				
Consideration Paper	Sequence Dependence	Time Constraint	Single Sourcing	Renewable	Non-Renewable	Genetic Algorithm	Branch and Bound	Meta Heuristics	Simulation	Problem Specific Heuristic and
Afshar and Haghani (2012)	\checkmark	\checkmark			\checkmark					
Ahuja et al. (1990)	\checkmark	\checkmark			V				V	
Ait-Kadia et al. (2011)	\checkmark	\checkmark		\checkmark	V				\checkmark	
Alfieri and Brandimarte (1997)		\checkmark	\checkmark	\checkmark				\checkmark		
Bachlaus et al. (2009)		\checkmark			V				V	
Balcik and Beamon	\checkmark								$\overline{\mathbf{v}}$	

(2008)								
Balcik et al. (2008)	V	V						
Ballestin and Trautmann		1						
(2008)		,						
Bottcher et al. (1999)	V	V	\checkmark	\checkmark		\checkmark		
Brucker and Kramer								
(1996)								
Can and Ulusoy (2010)		V	\checkmark	\checkmark	\checkmark			
Celano et al. (2008)	V	\checkmark	\checkmark					
Chan et al. (2006)	V	V		\checkmark	\checkmark			
Changfeng et al. (2006)	V			\checkmark	\checkmark			
Debels and Vanhoucke								
(2007)								
Deblaere et al. (2007)	V	V		\checkmark				
Deblaere et al. (2011a)	V	\checkmark		\checkmark				
Deblaere et al. (2011b)	V	V		\checkmark				
Depuy and Whitehouse								
(2001)								
Eckstein and Rohleder								
(1998)								
Fenton (2003)	V			\checkmark				
Gagnon et al. (2009)	V	V	\checkmark	\checkmark			\checkmark	
Haghani and Oh (1996)	V	\checkmark		\checkmark				
Hwang and Kogan								
(2003)								
Karsu and Azizoglu	\checkmark	\checkmark		\checkmark		\checkmark	\checkmark	

(2012)									
Klein (2000)		\checkmark			\checkmark			\checkmark	\checkmark
Lee and Lei (2001)	\checkmark			\checkmark	\checkmark				
Leus and Herroelen (2004)	\checkmark	\checkmark					\checkmark		
Mazzola and Neebe (1986)							\checkmark		
Mingozzi et al. (1999)		V		\checkmark					
Mokhtari et al. (2011)	\checkmark	V			\checkmark				
Moon et al. (2004)	\checkmark					\checkmark			
Nudtasomboon and Randhawa (1997)				\checkmark					\checkmark
Nolz et al. (2011)		V			\checkmark				
Onwubolu and Mutingi (2001)									
Rajkumar et al. (2011)	\checkmark	V	\checkmark		\checkmark				
Ranjbar et al. (2009)		V		\checkmark					
Robinson and Moses (2006)	\checkmark	\checkmark							\checkmark
Schirmer (2001)	\checkmark	V			\checkmark	\checkmark	\checkmark	\checkmark	
Sheu et al. (2012)									
Sheu (2007)					\checkmark				
Simons et al. (1999)					\checkmark				
Tormos and Lova (2003)	\checkmark	\checkmark			\checkmark				
Van de Vonder et al. (2006)	\checkmark	\checkmark							\checkmark
Vanhoucke et al. (2001a)	V	\checkmark		\checkmark	\checkmark		\checkmark		
--------------------------	--------------	--------------	--------------	--------------	--------------	---	--------------	--	
Vanhoucke et al. (2001b)	\checkmark	\checkmark		\checkmark			\checkmark		
Wong et al. (2012)				\checkmark	V	V			
Yi and Kumar (2007)	V	V			V				
Yuan and Wang (2009)	\checkmark	\checkmark			\checkmark				
Our research	V	V	\checkmark	\checkmark	V				

 Table 2.2
 Summary of Literature Review of Emergency Logistics

2.6 Discussion

In a realistic situation, such as multi-product, multi-echelon, distribution routing, the problem under consideration has a complicated structure with a huge size. Moreover, each problem in the literature has its unique assumptions and definitions. Various approaches are considered and analyzed for different problems, and therefore it is very difficult to propose an integrated view of the entire set of methodologies. In this section, we provide three different perspectives. The first one is to classify the solution approaches with a perspective on the decomposition framework, and solution methodologies applied to the decomposed sub-problems. The second one is to relate the problem structure to the utilized solution approaches. The last one is to address the importance of applications of various PDP methodologies to emergency logistics.

2.6.1 Structure of Solution Approach

Most problems in the literature are computationally difficult to solve optimally, and thus different decomposition approaches are utilized. When the problem is decomposed, the optimality of the problem may not be guaranteed, but each decomposed problem is much easier to solve and sometimes can be solved effectively (e.g., optimally or near-optimally) and efficiently (e.g., in polynomial time or in pseudo-polynomial time). Moreover, after the original problem is decomposed into sub-problems, each sub-problem can be further decomposed according to the structure of the sub-problem. The overall framework of the solution methodology in terms of decomposition has the following three categories.

- 1) *No Decomposition*: The entire problem is solved at once.
- 2) Mathematical Decomposition: The original problem is decomposed according to mathematical properties. Two representative decompositions are Lagrangean decomposition and Benders decomposition. In Lagrangean decomposition, some of constraints are relaxed by Lagrangean relaxation and the problem under consideration can be decomposed into independent sub-problems. In Benders decomposition, some of the variables are fixed and the problem can be decomposed.
- 3) *Heuristic Decomposition*: The original problem is decomposed according to problem-specific properties. A common way is to decompose the problem with respect to layers. Thus, the upstream problem and the downstream problem are separately defined. Another method is to decompose into a strategic problem and an operational problem.

When the problem (or decomposed sub-problem) cannot be further decomposed, or is going to be solved directly, several approaches are utilized. The major solution approaches in the literature can be summarized:

- Exact Algorithm Development: When the problem (or sub-problem) can be formulated as a problem which has a known optimal algorithm in polynomial time (or pseudo-polynomial time), it can be solved optimally. Typical examples are Network Flow Problems, Linear Programming (LP), and Dynamic Programming.
- 2) Modeling with an Optimization Solver: Some papers describe the problem with an exact mathematical formulation, such as Linear Programming (LP), Non-linear Programming (NLP), and Mixed Integer Programming (MIP), and solve it with an optimization solver. When the problem size is small enough or the problem has unique properties, optimal solutions can be obtained in a reasonable time frame. Various optimization solvers are found in the literature, such as CPLEX, GAMS, AMPL, LINGO, and GLPK. In order to strengthen the formulation, additional constraints, such as valid inequalities, can be inserted. In most cases, an approximate solution by an optimization solver is acceptable, given the error limit or running time limit.
- 3) *Mathematical Programming Approach*: When the sub-problem is still too hard to optimally solve, there are several approaches utilizing mathematical programming techniques. Representative methods are Lagrangean relaxation and LP relaxation.
- 4) *Metaheuristic*: Metaheuristics iteratively improve a candidate solution with regard to a given measure of quality. A metaheuristic makes few or no assumptions about

the problem being optimized and can search very large spaces of candidate solutions. However, it does not guarantee that an optimal solution is ever found. The solution quality and running times are highly dependent on the setup parameters for metaheuristic approach. Examples are Local Search (e.g., Tabu Search, Simulated Annealing), Evolutionary Algorithms (e.g., Genetic Algorithm), and Swarm Intelligence (e.g., Particle Swarm Optimization, Ant Colony Optimization).

5) *Problem-Specific Algorithms*: According to the problem-specific property, an algorithm can be developed only for the particular problem. In many cases, values of variables are sequentially decided. A representative one is a greedy algorithm, which makes a locally optimal choice at each stage with the hope of finding a global optimum. After obtaining a solution, a local improvement procedure may be applied.

Figure 2.2 gives an overview of the existing procedures for solving the integrated problem. If a problem is directly solvable, it can be solved using an exact method. Otherwise, we may try to decompose it into multiple sub-problems with minor changes from the original problem, or try to use other solution approaches. If the problem is decomposed, sub-problems can be solved separately and each of them is considered as an independent problem. Then, we can iteratively check whether the sub-problems are directly solvable or further decomposable. If the problem (or sub-problem) is not decomposable or we do not attempt to further decompose it, several solution approaches are applicable.



Figure 2.2. An Overview of Existing Procedures for Solving the Integrated Problem

Based on the above classification, the solution approaches used in the literature surveyed in this chapter can be classified in Table 2.3. We make the following observations:

- When the problem is solved without decomposition, the two major methodologies are modeling with an optimization solver, and a meta-heuristic, in which the structural property is not well-utilized.
- When a mathematical decomposition is utilized as an overall framework, the sub-problem is always solved by mathematical programming methods for optimal

or approximate solutions. In other words, if one would like to apply mathematical decomposition, sub-problems should be able to be well-handled by mathematical programming methods.

• When the problem is heuristically decomposed, metaheuristic and problem-specific heuristics are frequently used.

Overall	No	Mathematical	Heuristic			
framework	Decomposition	Decomposition	Decomposition			
Sub-problem						
Methodology						
Modeling with	Rizk et al. (2006)		Sabri and Beamon (2000)			
Optimization Solver	Chen and Lee (2004)		Cetinkaya et al. (2009)			
	Dhaenens-Flipo and Finke		Chandra and Fisher (1994)			
	(2001)					
	Fahimnia et al. (2008a)					
	Bilgen and Gunther (2009)					
Exact Algorithm	Armstrong et al. (2008)	Yung et al. (2006)	Bard and Nananukul			
Development	Ruokokoski et al. (2010)	Eksioglu et al. (2007)	(2009b)			
		Karakitsiou and Migdalas	Archetti et al. (2011)			
		(2008)				
		Dogan and Goetschalckx	(
		(1999)				
Mathematical	Yilmaz and Catay (2006)	Fumero and Vercellis (1999)	Bard and Nananukul			
Programming	Lei et al. (2009)		(2010)			
Approach			Bard and Nananukul			
			(2009b)			
			Archetti et al. (2011)			
Metaheuristic	Jang et al. (2002)		Ahuja et al. (2007)			

	Gen and Syarif (2005)	Yimer and Demirli (2010)		
	Kannan et al. (2010)	Bard and Nananukul		
	Naso et al. (2007)	(2009a)		
	Boudia and Prins (2009)	Geismar et al. (2008)		
	Bolduc et al. (2010)	Yossiri et al. (2012)		
Problem-Specific	Lei et al. (2006)	Park (2005)		
Algorithm	Liu et al. (2008)	Cetinkaya et al. (2009)		
	Gebennini et al. (2009)	Chandra and Fisher (1994)		
	Shen and Qi (2007)	Lei et al. (2006)		
		Geismar et al. (2008)		
		Archetti et al. (2011)		

Table 2.3 Summary of Solution Approaches

2.6.2 Problem Structure and Solution Approaches

In the reviewed papers, along with their problem structure and methodologies used, when routing is involved as a part of the decision, the problem includes a vehicle routing problem (VRP), which is one of the well-known difficult combinatorial optimization problems. Thus, we separately discuss the problems where routing is considered, and those where it is not.

For the problems without routing decisions (PDP and PDPT), the methodologies for PDP and PDPT are different.

 The major solution methodology for PDP is to use Lagrangean decomposition as a framework and mathematical programming for the decomposed problems.
 Especially when the PDP is defined on a supply chain network with two stages, Lagrangean decomposition works very well, because the sub-problems can be solved optimally. However, when PDP is defined on a network with three or more stages, Lagrangean decomposition is rarely used.

• The major methodology of PDPT is to establish a mathematical model without decomposition and use an optimization solver. Half of the papers dealing with PDPT use an optimization solver, even though some mathematical models are non-linear, while no papers use mathematical programming for overall or decomposed problems. It may imply that the problem with time constraints can be clearly defined in a mathematical model, but the time constraints make it difficult to utilize the mathematical structure for mathematical programming-type algorithm development.

For the problems with routing decisions (PDPR and PDPRT), mathematical decomposition is rarely used, while heuristic decomposition is frequently used. When the problem is decomposed heuristically, the upstream problem deals with production lot-sizing and the downstream problem is defined for routing decisions. Decomposed sub-problems are solved by various methods.

• In PDPR, one sub-problem may be modeled and solved by an optimization solver, and the other sub-problem solved by a problem-specific heuristic. In another case, one sub-problem is solved by mathematical programming for an approximate solution, while the other sub-problem is solved by an exact algorithm for the optimal solution. • In PDPRT, a mathematical programming approach is rarely used as the solution methodology for decomposed problems because of the complexity of the decomposed problems. Instead, metaheuristic and problem-specific heuristic approaches are widely used. In both PDPR and PDPRT, the solution approaches cannot directly give a solution close to the optimum and, thus, local improvement heuristics are frequently used as a post-processing procedure.

In addition, we observe the following relationships between problem structure and methodologies used:

- The mathematical programming approach works better for problems without time constraints.
- When the problem structure is complicated, problem-specific algorithms and local improvement heuristics are frequently used.
- Metaheuristics can be applied for most problem structures.

2.6.3 Trends and Applications

The trend in solution approaches for modern supply chain operations is to use a hybrid methodology, by combining the aforementioned methods and the use of a simulation as a framework, especially for practical and large-scale problems. When a simulation is used as a framework for solving the problem, a mathematical model is first established by relaxing some uncertain factors and solved with a variety of approaches. Its solution is then used as the input to the simulation model, then incorporated with different uncertainty sources such as demand, facility failure, delivery time, etc., and the output of the simulation model gives feedback for the parameters of the mathematical model to be revised accordingly. This procedure can be repeated until the obtained solution is efficient and robust.

One of the most important applications is in emergency logistics. Today's Internet allows the need for disaster relief to be communicated cross-country and internationally within minutes of an event, and the rapid formation of disaster relief supply chains for quick response to people in the affected areas. A highly effective and fully integrated production and distribution operation that pulls supplies from different industries and states to ensure delivery of these resources to the people in an affected area is critical to human well-being. Many solution methodologies can be extended in this area. For example, during the post-disaster period, the time in the rescue process becomes the most important issue for severely injured patients. The problem of producing and allocating different types of resources and service operations to customers in the affected areas is a classical example of PDPT. Therefore, all the modeling and solution methodologies can be directly or indirectly utilized to solve the emergency logistics problem. If the routing issue is considered (e.g., trucking routes in delivery of medical kits), methodologies for solving PDPRT can be used. Thus, the focus of this dissertation is to apply the existing results reviewed in this chapter to solve the emergency operations scheduling problem that will be defined in the next chapter.

Chapter 3. Problem Definition and the MIP Model

In this chapter, we formally define the research problem in this reseach and build up the Mixed Integer Programming (MIP) model.

Our problem is defined upon a two-stage supply chain network (see Figure 3.1) consisting of:

(i) A set of customers, **H**. The service to each customer h, $h \in H$, requires a simultaneous availability of both renewable and non-renewable resources, and has an expected service completion time d_h . The service starting time at customer h, S_h , is determined by the latest arrival time of the two types of resources. A tardiness penalty incurs whenever the actual completion time passes d_h . There is an expected (i.e., predetermined) service duration p_h at the site of customer h. To deliver the service at customer h, a total of D_h units of non-renewable resources are needed.

(ii) A set of distribution centers (DCs), **K**. Each DC k, $k \in K$, receives a sequence of n_{B_k} batches of non-renewable resources from its upstream suppliers, and each batch is defined by batch size Q_{jk} , and batch arrival (release) time A_{jk} , $1 \le j \le n_{B_k}$, $k \in K$. That is, additional Q_{jk} units of non-renewable resources become available at time A_{jk} . The shipping time from DC k to a certain customer h is given by $\overline{\tau}_{kh}$.

(iii) A set of home-bases of the renewable resources (e.g., medical teams), M. Each home-base *i*, $i \in M$, dispatches a renewable resource (i.e., a team) that travels to the customer locations to perform service operations. The route of each renewable resource (i.e., the sequence of customer sites to be visited/served), H_i , is assumed to be given in this study. Each renewable resource departs from its base at a given time point (i.e., the travel team release time) r_i , and travels along a fixed route $H_i = \{h_i(0), h_i(1), h_i(2), ..., h_i(n_i)\}$, where elements $h_i(k)$, $k=1, 2, ..., n_i$, represent hospitals along the route assigned to this renewable resource. The travel time of team *i* between two consecutive customer sites is defined as $\tau_{h_i(l)h_i(l+1)}$, for all $i \in M$, $0 \le l \le n_i - 1$. In addition, we assume that the triangle inequality holds for the travel time of both renewable and non-renewable resources. The problem is to allocate non-renewable resources from DCs to customers located along different routes of medical teams to support the service operations so that the total tardiness is minimized. Let us introduce the following notations.



Figure 3.1 Supply Network with Both Renewable and Non-renewable Resources

Model Parameters

- *H* : Set of hospitals (i.e., customers);
- K : Set of DCs;
- M: Set of medical teams, or, equivalently, the set of their home-bases;
- B_k : Set of batches (of the non-renewable resource) that arrive at DC k, for all $k \in K$;
- n_{H} : The total number of hospitals;
- n_K : The total number of DCs;
- n_M : The total number of medical teams;
- n_{B_k} : The total number of batches that arrive at DC k, for all $k \in K$;
- n_i : The total number of hospitals on the given route of medical team *i*, for all $i \in M$;

- $h_i(l)$: The *l*-th hospital on the route of team *i*, for all $i \in M$; $l=1, 2, ..., n_i$;
- H_i : The set of hospitals visited by team *i*, including the home-base of team *i*;

$$H_i = \{h_i(0), h_i(1), h_i(2), \dots, h_i(n_i)\}$$
, for all $i \in M$, and $h_i(0) = i$;

- D_h : The quantity of the non-renewable resource ordered by hospital h, for all $h \in H$;
- p_h : The given service time duration at hospital h, for all $h \in H$;
- d_h : Due date for completing service at hospital h, for all $h \in H$;
- A_{jk} : Release (arrival) time of the *j*-th batch of non-renewable resource at DC k, $k \in K, j \in B_k$;
- Q_{jk} : Quantity of the *j*-th batch of non-renewable resource arriving at DC *k*, for all $k \in K, j \in B_k$;
- $\overline{\tau}_{_{kh}}$: Travel time from DC k to hospital h, for all $k \in K, h \in H$;
- $\tau_{h_i(l)h_i(l+1)}$: Travel time of team *i* from $h_i(l)$ to $h_i(l+1)$, for all $i \in M$, $0 \le l \le n_i 1$;
- r_i : Release time of team *i* at its home-base, for all $i \in M$;

Decision variables

- q_{jkh} : Quantity of non-renewable resources shipped from the *j*-th batch at DC *k* to hospital *h*, for all $k \in K, j \in B_k, h \in H$;
- : Binary variable, $z_{jkh} = 1$ if the *j*-th batch of DC *k* supplies hospital *h* and $z_{jkh} = 0$ otherwise, for all $k \in K, j \in B_k, h \in H$;

Auxiliary variables

- S_h : Starting time of service at hospital h, for all $h \in H$;
- T_h : Tardiness in delivering service to hospital h, for all $h \in H$;

With these notations, our problem, **P**, can be defined as the following mixed integer-programming model.

$$\mathbf{P}: \quad Minimize \ G = \sum_{\forall h \in H} T_h \tag{3.1}$$

Subject to

$$\sum_{\forall k \in K} \sum_{\forall j \in B_{k}} q_{jkh} = D_{h} \qquad \text{for all } h \in H \qquad (3,2)$$

 $\sum_{\forall h \in H} q_{jkh} \le Q_{kj} \qquad \text{for all } k \in K, j \in B_k$ (3.3)

$$q_{jkh} \le z_{jkh} Q_{jk} \qquad \qquad \text{for all } k \in K, j \in B_k, h \in H \qquad (3.4)$$

$$(A_{jk} + \overline{\tau}_{kh}) z_{jkh} \le S_h \qquad \text{for all } k \in K, j \in B_k, h \in H \qquad (3.5)$$

$$S_{h_{i}(l)} + p_{h_{i}(l)} + \tau_{h_{i}(l)h_{i}(l+1)} \leq S_{h_{i}(l+1)},$$

$$p_{h_{i}(0)} = 0, S_{h_{i}(0)} = r_{i}$$
for all $i \in M, l = 0, 1, ..., n_{i} - 1$
(3.6)

$$T_h \ge S_h + p_h - d_h \qquad \text{for all } h \in H \qquad (3.7)$$

$$z_{jkh} \in \{0,1\}, \ q_{jkh} \ge 0, S_h \ge 0, \ T_h \ge 0 \quad \text{for all} \ k \in K, j \in B_k, h \in H$$
 (3.8)

In this model, the objective function (3.1) is to minimize the total tardiness across all hospitals in the network. Constraint sets (3.2) to (3.4) are related to the non-renewable

resource distribution. Constraint set (3.2) ensures that the demand for the non-renewable resource at each hospital will be completely fulfilled. Constraint set (3.3) ensures that the total shipping quantity from a given batch does not exceed the batch size. Constraint set (3.4) establishes the relationship between variables q_{ikh} and z_{ikh} . Constraint sets (3.5) to (3.7) are time-related constraints. Constraint set (3.5) ensures that the starting time of a service at a hospital will not be earlier than the latest arrival time of the non-renewable resource. Constraint set (3.6) ensures that the starting time of the service at any hospital will not be earlier than the earliest arrival time of the medical team. Constraint set (3.7)defines the tardiness of the services, and constraint set (3.8) defines the domains of decision variables. The design of objective function (3.1) is justified by the criticality of achieving a fast response to serve the needs for disaster relief (http://www.ifrc.org/PageFiles/53419/MAA0000410p.pdf,

http://www.gps.gov/applications/safety/, and Han, et al. (2011)).

In the next two chapters, we present a structrural analysis of Problem P and design a rolling horizon based heuristic to solve P efficiently.

Chapter 4. A Structural Analysis of Problem P

In this chapter, we discuss the computational complexity of the problem with different parameter settings by presenting either NP-hard proofs or polynomial time algorithms. In Section 4.1, we consider the case where there is a single DC and present a computational complexity result. In Section 4.2, we assume that the assignment between DCs and customers are given and fixed, and present polynomial time algorithms for two special cases. A greedy-type algorithm is presented for the case with a variable number of DCs and a single team, and a polynomial-time dynamic programming algorithm is developed for the case with fixed numbers of DCs and teams. In Section 4.3, when the assignments of DCs to customers have to be optimized, we present an NP-hard proof for the general case and provide polynomial time algorithms for two special cases. In Section 4.4, we summarize the results obtained in terms of computational complexity. Finally, in Section 4.5, we propose a framework of heuristic procedures for solving the more general problems in practice and discuss future research directions.

In order to analyze the structural properties of problem P, we make a distinction between two different environments with regard to the assignments of DCs to hospitals:

The Fixed Assignment Environment (F): This case refers to settings where the DC-hospital assignments are fixed in advance. That is, each customer in the network has its own designated DC for its supply of the non-renewable resource (e.g., which occurs in real life when each hospital or shelter is supplied by a local DC during the disaster relief);

The Open Assignment Environment (O): The assignments of DCs to hospitals are

For simplicity, the cardinalities of sets |B|, |D|, |M|, and |H| are denoted as n_B , n_D , n_M , and n_H , respectively. Furthermore, let us introduce the notation $P(\varphi, n_D, n_M)$, where $\varphi \in \{F, O\}$ stands for the DC-hospital assignment environment, $n_D \ge 1$ stands for the total number of DCs, and $n_M \ge 1$ stands for the total number of medical teams (i.e., the renewable resources). With this notation, for example, the problem with a given and fixed DC-Hospital assignment ($\varphi = F$), two DCs ($n_D = 2$), and a single travel team ($n_M = 1$) is denoted as P(F, 2, 1). The computational complexity of problem **P** depends on the instances defined by parameters φ , n_D and n_M , which will be discussed in subsequent sections. We will make a distinction between cases with a *fixed number* and a *variable number* of n_D or n_M , since they may result in a different computational complexity (i.e., one may be polynomial time solvable, while the other may be NP-hard) A fixed number is considered here as a bounded constant. When a fixed number appears as an exponent in the time complexity of an algorithm, the algorithm is still considered as a polynomial time algorithm. When a variable number appears as an exponent in the time complexity of an algorithm, the algorithm becomes an exponential time algorithm.

We note that if the triangle inequality does not hold for the travel times of the medical teams and of the non-renewable resources, then even the simplest case with a single DC and a single team is already NP-hard.

Theorem 4.1. If the triangle inequality does not hold for the travel times of the medical teams, then problem $\mathbf{P}(F, 1, 1)$ is NP-hard.

Proof. We start with the definition of the PARTITION Problem, which is a well-known NP-complete problem and will be used for the reduction in the proof. The input of PARTITION is set $S = \{a_1, ..., a_n\}$ where a_j is a positive integer. The output of

PARTITION is set
$$S_1 \subset S$$
 such that $\sum_{a_j \in S \setminus S_1} a_j = \sum_{a_j \in S_1} a_j = \frac{A}{2}$ where $A = \sum_{j=1}^n a_j$.

Let *H* be $\{1, 2, ..., n\}$ with $(p_h, D_h, d_h, w_h) = \left(0, a_h, (A - a_1) + \sum_{l=1}^{h-1} a_l, 1\right)$. We have one DC,

denoted by DC 1, which has two batch arrivals with $A_{11} = 0$, $A_{12} = A$ and $Q_{11} = Q_{12} = \frac{A}{2}$. We have one team that is available at time $r_1 = A - a_1$. Transportation times are

$$\tau_{hh'} = 0$$
 and $\overline{\tau}_{1h} = \sum_{l=2}^{h} a_l$. Then, if hospital *h* is served by the first batch, its tardiness is zero while if by the second batch its tardiness is $T_h = \max\left\{0, \left(A + \sum_{l=2}^{h} a_l\right) - \left(A - a_1 + \sum_{l=2}^{h-1} a_l\right)\right\} = a_h$. Therefore, there is a partition if and

only if there is a schedule with the total tardiness less than or equal to $\frac{A}{2}$.

4.1 The Single DC Case

When there is a single DC, there is clearly no difference between the fixed assignment environment and the open assignment environment. Thus, in this section, we determine the computational complexity of the case with a single DC and a variable number of teams.

Theorem 4.2. The problem with a single DC and a variable number of medical teams, i.e., $P(F, 1, n_M)$, is strongly NP-hard. "F" denotes the fixed assignment between DC and customers, "1" means single DC and " n_M " is the number of medical teams.

Proof. We start with the definition of 3-PARTITION Problem, which is a well-known strongly NP-complete problem and will be used for the reduction in the proof. The input of the 3-PARTITION problem is set $S = \{a_1, ..., a_{3_t}\}$ where a_j is a positive

integer and
$$\frac{A}{4} < a_j < \frac{A}{2}$$
 where $A = \sum_{j=1}^{3t} a_j$. The output of 3-PARTITION is a

partition of set S into disjoint sets $S_1, S_2, ..., S_t$ such that $\sum_{a_j \in S_t} a_j = A$.

We consider a problem instance of $\mathbf{P}(F, 1, n_M)$. We have 3t teams and team *i* has two demand points, $h_i(1)$ and $h_i(2)$ with the following information for i = 1, ..., 3t;

$$-(p_{h_i(1)}, D_{h_i(1)}, d_{h_i(1)}, w_{h_i(1)}) = (A, a_i, tA^3, 1),$$

$$-(p_{h_i(2)}, D_{h_i(2)}, d_{h_i(2)}, w_{h_i(2)}) = (0, A^2 - 2a_i, A, 1).$$

We have a single DC, denoted by DC 1, with 2*t* batches and batches 2j-1 and 2j referred to as group *j* for j = 1, ..., t. The batch information, for j = 1, ..., t, is as follows: $-(Q_{2j-1}, A_{2j-1,1}) = (A, (j-1)A^3),$ $-(Q_{j2}, A_{2j,1}) = (3A^2 - 2A, (j-1)A^3 + A).$

All transportation times are zero. Note that in any feasible schedule the tardiness of demand point $h_i(1)$ is always zero.

Suppose that there exists a 3-partition, $S_1, S_2, ..., S_t$. Thus, S_j has exactly three elements for j = 1, ..., t. Then, we can construct a schedule where batches in Group j serve all the demand points of the teams in S_j . More precisely, batch 2j - 1 serves demand points $h_i(1)$ for $i \in S_j$ and batch 2j serves demand points $h_i(2)$ for $i \in S_j$. The total tardiness of the demand points of team i for $i \in S_j$ is exactly, $3(j-1)A^3$. Thus, the total tardiness of all the demand points is $\sum_{j=1}^{t} 3(j-1)A^3 = \frac{3}{2}t(t-1)A^3$. Therefore, if there exists a 3-partition for the 3-PARTITION problem, then there exists a schedule for $\mathbf{P}(F, 1, n_M)$ with the total tardiness being less than or equal to $\frac{3}{2}t(t-1)A^3$.

Suppose that there exists a schedule such that the total tardiness is less than or equal to $\frac{3}{2}t(t-1)A^3$. Let S'_j be the set of teams whose hospitals are fully served by batch group j but not fully served by batch group j-1. Note that the total quantity of batch group j is $Q_{j-1,2} + Q_{j,2} = A + (3A^2 - 2A) = 3A^2 - A$ and the total required quantity of hospitals belonging to team i is $A^2 - a_i$. This implies that $\left| \bigcup_{b=1}^j S'_b \right| \le 3j$ for j = 1, ..., t.

Thus, if hospital $h_i(2)$ is fully served by batch group j, its tardiness is at least

$$(j-1)A^3 - A$$
. Then, if $\sum_{i=1}^{3t} T_{h_i(2)} \le \frac{3}{2}t(t-1)A^3$, then $|S'_j| = 3$ for $j = 1, ..., t$.

Suppose there exists a team j such that $\sum_{i \in S'_j} D_{h_i(1)} < A$, then

 $\sum_{i \in S'_j} \left(D_{h_i(1)} + D_{h_i(2)} \right) = \sum_{i \in S'_j} \left(A^2 - D_{h_i(1)} \right) > 3A^2 - A.$ It implies that at least one of the hospitals

in $\{h_i(2) | i \in S'_j\}$ is not fully served by group *i*, leading to a contradiction. Suppose that there exists team *j* such that $\sum_{i \in S'_j} D_{h_i(1)} > A$ then one of hospitals in $\{h_i(1) | i \in S'_j\}$ is not

fully served by the first batch in group *j* and leads to a delay of at least one of hospitals $in\{h_i(2) | i \in S'_j\}$, $implying \sum_{h \in S'_j} T_h > 3(j-1)A^3$. Therefore, the total tardiness is strictly

greater than
$$\frac{3}{2}t(t-1)A^3$$
. Therefore, if $\sum_{i=1}^{3t} T_{h_i(2)} \le \frac{3}{2}t(t-1)A^3$, then
 $\sum_{i\in S'_j} D_{h_i(1)} = \sum_{i\in S'_j} a_{h_i(1)} = A$ for $j = 1, ..., t$.

Therefore, if there exists a schedule of $\mathbf{P}(F, 1, n_M)$ with the total tardiness being less than or equal to $\frac{3}{2}t(t-1)A^3$, then there exists a 3-partition for the 3-PARTITION problem. Therefore, there is a 3-partition if and only if there is a schedule for $\mathbf{P}(F, 1, n_M)$ with the total tardiness less than or equal to $\frac{3}{2}t(t-1)A^3$. \Box

We now consider two solvable cases in which there is a single DC but the DC may receive multiple batches. For the first case, we assume each traveling team has a given tour of length one (i.e., there is only a single hospital on each tour). Furthermore, we assume that all renewable resources arrive at the corresponding hospitals at time zero, and that all hospitals have an identical demand size, i.e., $D_h = D_{h'}, \forall h, h' \in H$. Let A_j be *j*-th batch arrival time, Q_j be the *j*-th batch supply capacity, $\overline{\tau}_h$ be the shipping time from the single DC to hospital h, q_{jh} be the quantity shipped from the *j*-th batch to hospital h, and z_{jh} be the binary variable that is equal to 1 if hospital *h* is served by the *j*-th batch. In this case, problem P is reduced to the following one (see Figure 4.1(a)):



(a) P₁: Multiple Batches and Unit Tour Length
 (b) P₂: Two-Batch and Arbitrary Tour Lengths
 Figure 4.1 Examples with a Single DC Supply Process

Consider the following algorithm that solve problem P_1 .

Algorithm 4.1: Let $w_h = \overline{\tau}_h + p_h - d_h$. Sequence all hospitals in H in a non-increasing order of w_h , and let the new sequence be $H^A = \{h_1, h_2, ..., h_N\}$. Deliver all batches from this single DC to hospitals according to this order, i.e., use the first batch to serve h_1 , and if there is a remainder then use that to serve h_2 , so on so forth until all hospitals' demands are fulfilled.

Lemma 4.1. Algorithm 4.1 finds the optimal solution to P_1 in $O(n_H)$.

Proof. We first show that Algorithm 4.1 brings the optimal solution with the minimum total tardiness. If this is not true, then there must exist an optimal solution, Σ , that yields a smaller total tardiness than that derived by Algorithm 1, and hence some hospital(s)'s tardiness calculated by Algorithm 4.1 must be larger than that by Σ . Let hospital h be the first such kind of hospital in sequence H^{A} . We use batches j and j*, respectively, to denote the last batch of the DC that fulfills the demand of hospital h in the solution associated with Algorithm 4.1 and Σ , and derive $T_h(j) = \max\{0, A_j + w_h\}$ and $T_h(j^*) = \max\{0, A_{j^*} + w_h\}$ as the respective tardiness's for hospital h. Our discussions $T_h(j) = \max\{0, A_j + w_h\} > T_h(j^*) = \max\{0, A_{j^*} + w_h\}$ above imply that Since $w_h = \overline{\tau}_h + p_h - d_h$ is a given number regardless the position of h in any sequence, $A_j > A_{j^*}$ must hold to make the above inequality happen, and hence $j^* < j$. Also, we can find at least one hospital h' that precedes h in H^A (i.e., $w_{h'} > w_h$) and has the last batch *j*'received from the DC in the optimal schedule Σ satisfy $j^* < j'$ (otherwise, $j \le j^*$).

Since *h* and *h*' have the same demand, if we switch them in the service sequence Σ , this will not affect the tardiness's of any other hospitals. Thus, the consequent change of the total tardiness is $\Delta T = (T_h^*(j^*) + T_{h'}^*(j')) - (T_h^s(j') + T_{h'}^s(j^*))$, where $T_h^s(j')$ and $T_{h'}^s(j^*)$ are the tardiness's associated with hospitals *h* and *h'* after the switch. $\Delta T = (T_h^*(j^*) + T_{h'}^*(j')) - (T_h^s(j') + T_{h'}^s(j^*))$.

Considering $A_{j^*} < A_{j^*}$ and $w_{h'} > w_h$, we have $A_{j'} + w_{h'} > A_{j'} + w_h$, $A_{j^*} + w_{h'} > A_{j^*} + w_h$. Therefore, the above formula implies that $\Delta T = 0$ when $A_{j'} + w_{h'} < 0$ or $A_{j^*} + w_h > 0$. When $A_{j'} + w_{h'} > 0 > A_{j^*} + w_h$, $T_h^*(j^*) + T_{h'}^*(j') = A_{j'} + w_{h'}$. Consider the following cases.

i. When
$$A_{j'} + w_h < 0, A_{j^*} + w_{h'} < 0$$
, $T_h^s(j') + T_{h'}^s(j^*) = 0$, and $\Delta T = A_{j'} + w_{h'} > 0$;
ii. When $A_{j'} + w_h < 0, A_{j^*} + w_{h'} > 0$, $T_h^s(j') + T_{h'}^s(j^*) = A_{j^*} + w_{h'}$, and $\Delta T = A_{j'} - A_{j^*} > 0$;
iii. When $A_{j'} + w_h > 0, A_{j^*} + w_{h'} < 0$, $T_h^s(j') + T_{h'}^s(j^*) = A_{j'} + w_h$, and $\Delta T = w_{h'} - w_h > 0$;

iv. When
$$A_{j'} + w_h > 0, A_{j^*} + w_{h'} > 0$$
, $T_h^s(j') + T_{h'}^s(j^*) = A_{j'} + w_h + A_{j^*} + w_{h'}$, and
 $\Delta T = -(A_{j^*} + w_h) > 0;$

Summarizing the above analysis, we conclude $\Delta T \ge 0$. In other words, by switching *h* and *h*' in the sequence of optimal solution Σ , we always reach another optimal solution with $j^*>j'$, which results in a contradiction with the earlier discussions. Hence Algorithm 4.1 can solve this case optimally. Clearly Algorithm 4.1 has a complexity of $O(n_H)$. \Box

The next special case includes a single DC with two batches, multiple teams, and each team has multiple hospitals along its route. The hospitals may have different demand sizes, but for those in the same sequence their demand sizes are non-decreasing as the sequence order. For simplicity, we assume that the due dates of all hospitals are equal to zero. Moreover, we assume that team traveling time between the hospitals in the same team sequence is very small so that the renewable resource arrival time is negligible. Under this assumption, we can see that all hospitals in the same sequence are clustered as a group, and thus we can also assume that the non-renewable resource arrival times at all hospitals in the same sequence are identical. Therefore, the tardiness of each hospital is completely determined by the non-renewable resource arrival time (see Figure 4.1(b)). In this case we can simplify the original MIP model by introducing the following notations. Suppose that there are n_M teams ($n_M \ge 2$) and n_H hospitals, and that team I visits a number of n_I hospitals (in set H_I), team 2 visits a number of n_2 hospitals (in set H_2),..., and team i visits a number of n_i hospitals (in set H_i), where $\sum_{1 \le i \le |M|} n_i = n_H$. Let A_j

be *j*-th batch arrival time and Q_j be the *j*-th batch supply capacity, for j=1,2. Let $\overline{\tau}_i$ be the shipping time from the single DC to all hospitals in team *i*, for $i=1,2...,n_M$. Let q_{jh} be the quantity shipped from the *j*-th batch to hospital *h*, and z_{jh} be the binary variable that is equal to 1 if customer *h* is <u>fully</u> served by the *j*-th batch. Based on the notations, we have the following model for special case P₂.

$$\begin{split} P_{2}: Minimize & \sum_{\forall i \in M} \sum_{\forall h \in H_{i}} \left[(A_{1} + \tau_{i}) z_{1h} + (A_{2} + \tau_{i})(1 - z_{1h}) \right] \\ s.t. \\ & \sum_{\forall h \in H} D_{h} \leq Q_{1} + Q_{2}; \\ & \sum_{j=1,2} q_{jh} = D_{h}; & \forall h \in H \\ & \sum_{\forall h \in H} q_{jh} \leq Q_{j}; & \forall h \in H, \ j = 1, 2 \\ & q_{jh} \leq z_{jh} Q_{j}; & \forall h \in H, \ j = 1, 2 \\ & z_{1h_{i}(l)} \geq z_{1h_{i}(l+1)}; & \forall i \in M, \ l = 1, 2, ..., n_{i} - q_{jh} \geq 0; \ z_{jh} \in \{0,1\}; & \forall h \in H, \ j = 1, 2 \end{split}$$

Note that constraint set $z_{1h_i(l)} \ge z_{1h_i(l+1)}$ guarantees that non-renewable service order must follow the order in the team sequence, and hence this constraint set can replace constraint set (3.6) in problem P. Note also that the objective function in P₂ is equivalent to min $A_2n_H + \sum_{\forall i \in M} n_i\tau_i - (A_2 - A_1)\sum_{\forall i \in M} \sum_{\forall h \in H_i} z_{1h}$. The first two terms are fixed

1

positive numbers, and thus the objective function here is further equivalent to max $\sum_{\forall i \in M} \sum_{\forall h \in H_i} z_{1h}$, i.e., we are trying to allocate as many hospitals as possible to the first

batch. Since only z_{1h} is involved in the objective function, P_2 can be further simplified as the following program.

$$\max \sum_{\forall i \in M} \sum_{\forall h \in H_i} z_{1h}$$

s.t. $\sum_{\forall h \in H} D_h z_{1h} \le Q_1; z_{1h_i(l)} \ge z_{1h_i(l+1)}, \forall i \in M, l = 1, 2, ..., n_i - 1; z_{1h} \in \{0, 1\}, \forall h \in H$

This is a special case of Knapsack problem, and can be solved by the following algorithm.

Algorithm 4.2: Each time allocate one hospital by the following rule. For all hospitals sequences (visited by a given team), pick the first available hospital that has not been served by batch 1, and among these hospitals picked up from each sequence, we choose the one that has the smallest demand size. Let batch 1 serve this hospital. If more than one hospitals assume the smallest demand size, randomly choose one. Keep allocating hospitals to batch 1 by this rule until either no more hospital can be fully served or all hospitals' demands are fulfilled.

Since algorithm 4.2 always choose the current available hospitals that have the smallest size and the demand sizes of hospitals are increasing in a team sequence, the greedy type algorithm can find the optimal solution for P_2 .

Lemma 4.2. Algorithm 4.2 finds the optimal solution to P_2 in $O(n_{H^2}^{n_{H^2}})$.

4.2 Multiple DCs in a Fixed Assignment Environment

In this section, we deal with multiple DCs in an environment with fixed assignments. In particular, we consider the following two cases: (i) a variable number of DCs with a single medical team and (ii) a fixed number of DCs with a fixed number of medical teams. We present polynomial time algorithms for both cases.

4.2.1 A Variable Number of DCs and a Single Medical Team

We consider problem P₃ with a variable number of *DCs*, n_K , and a single medical team ($n_M = 1$). We shall show that this problem can be solved optimally in linear time by a greedy algorithm. Without loss of generality, we assume that the route along which the single team visits the hospitals is specified by the indices of hospitals or $\langle 1, 2, ..., n_H \rangle$.

<u>Algorithm 4.3 (Greedy Algorithm</u>): (Given route $\langle 1, 2, ..., n_H \rangle$) For the next hospital in the given route of the travel team, allocate from the batches received so far at the assigned DC an amount that is sufficient to meet the demand for the non-renewable resource.

Lemma 4.3. In the schedule generated by the Algorithm 4.3, each hospital starts its service at its earliest possible time.

Proof. Suppose the claim is not true. If so, there must exist a problem instance S such

that in the schedule generated by the Greedy Algorithm some hospitals do not start the services at earliest possible time. Let σ^G be the schedule of instance *S* generated by the Greedy Algorithm. Let hospital *h* be the first hospital in schedule σ^G that does not start the service at its earliest possible time and let DC *k* be the DC assigned to hospital *h*. Let σ be the schedule in which hospital *h* has the earliest possible start time of the service. Let $S_h(\sigma^G)$ and $S_h(\sigma)$ be service start times at hospital *h* in schedules σ^G and σ , respectively. Then, by definition, $S_h(\sigma^G) > S_h(\sigma)$ and $S_{h'}(\sigma^G) \le S_{h'}(\sigma)$ for $h' = 1, \dots, h-1$.

Recall that $s_h(\sigma^G)$ is either the arrival time of the non-renewable resource or the arrival time of the renewable resource at hospital *h*. If $S_h(\sigma^G)$ is determined by the arrival time of the non-renewable resource, i.e., $S_h(\sigma^G) = S_{h-1}(\sigma^G) + p_{h-1} + \tau_{h-1,h}$, then $S_h(\sigma^G)$ is the earliest possible start time since $S_{h'}(\sigma^G)$ is the earliest possible start time at hospital *h'* for $h' = 1, \dots, h-1$, which leads to a contradiction. Thus, $S_h(\sigma^G) > S_{h-1}(\sigma^G) + p_{h-1} + \tau_{h-1,h}$ and $s_h(\sigma^G)$ is determined by the arrival time of the renewable resource at hospital *h*.

Let α and β be the last batches of DC k that serve hospital h in schedules σ^{G} and σ , respectively.

Thus, $S_h(\sigma^G) = A_{\alpha k} + \overline{\tau}_{kh}$. If $\beta \ge \alpha$, then $S_h(\sigma) \ge A_{\beta k} + \overline{\tau}_{kh} \ge A_{\alpha k} + \overline{\tau}_{kh} = S_h(\sigma^G)$ would be a contradiction. Hence, $\beta < \alpha$.

Consider in schedule σ the set of all hospitals served by DC k and precede hospital h and denote the set as H'. Then, there exists at least one hospital in H' that is provided by batch γ of DC k for some $\gamma \ge \alpha$. Otherwise, in schedule σ , all hospitals in H' are provided by batches preceding batch α and hospital h is also provided by batches preceding batch α since β is the last batch for hospital h and $\beta < \alpha$. However, it is impossible to provide hospitals in $H' \cup \{h\}$ with batches preceding batch α . Thus, we can say that there exists at least one hospital in H' that is provided by batch γ of DC k for some $\gamma \ge \alpha$ and let h' be the last hospital in hospitals provided by batch γ for some $\gamma \ge \alpha$. Therefore, by the triangular inequality,

$$S_{h}(\sigma) \geq S_{h'}(\sigma) + \sum_{a=h'}^{h-1} \tau_{a,a+1} \geq A_{\gamma k} + \overline{\tau}_{kh'} + \sum_{a=h'}^{h-1} \tau_{a,a+1} \geq A_{\gamma k} + \overline{\tau}_{kh} \geq A_{\alpha k} + \overline{\tau}_{kh} = S_{h}(\sigma^{G}),$$

which is a contradiction. This completes the proof. \Box

Theorem 4.3. Algorithm 4.3 solves problem P_3 with a variable n_D in linear time.

Proof. By Lemma 4.3, in the schedule by the Greedy Algorithm, all the hospitals start the service at their earliest possible times. Thus, the Greedy Algorithm is optimal. It takes $O(n_B + n_H)$ time to assign batch quantities to hospitals and it takes $O(n_H)$ time to calculate start times of all hospitals. Therefore, the overall time complexity is $O(n_B + n_H)$. This completes the proof. \Box

We now report a numerical example that is solved by the proposed algorithm 3 to optimality. In this example, we are given ten hospitals ($n_H = 10$) and two distribution centers ($n_K = 2$) for the non-renewable resource. DC 1 has three batches and is assigned to serve hospitals 1, 4, 5, 7, 8, and 10, while DC 2 has two batches and is assigned to serve hospitals 2, 3, 6, and 9. There is only one medical team that starts out at time 0 from its

home-base location (0, 0). The parameters are described in Tables 4.1 - 4.2, where coordinates are used to calculate the required shipping time between two locations with speed 1.

DC k		DC 1		D	OC 2
Coordinates in the xy axis		(0, 0)		(1	1, 0)
Batch j	1	2	3	1	2
Batch Arrival time A_{kj}	3	7	10	2	40
Batch Capacity Q_{kj}	15	8	12	10	8

Table 4.1	Parameters	for DCs a	nd Batches	(Greedy A	Algorithm)
-----------	------------	-----------	------------	-----------	------------

I	Hospital	Service DC	Demand	Penalty Cost	Due Date	Duration	Coordinate in
	h	k	D_h	W_h	d_{h}	p_h	the x-y axis
	1	1	5	2	10	3	(2, 0)
	2	2	3	4	30	3	(0, 1)
	3	2	7	10	15	5	(2, 2)
	4	1	8	3	55	4	(1, 4)
	5	1	9	12	20	5	(2, 5)
	6	2	4	5	60	3	(0, 8)
	7	1	4	5	12	6	(4, 0)
	8	1	3	2	18	4	(1, 7)
	9	2	2	3	50	5	(3, 3)
	10	1	5	1	95	4	(5, 1)

Table 4.3 summarizes the results obtained by the proposed greedy algorithm.

Hospital	Service	q_{kjh}		Last	Last Batch	Team	Service	Due	Tardiness	Penalt	
	DC			Batc	Arrival	Arrival	Starting	Date		y Cost	
			j		h	Time	Time	Time			
h	k	1	2	3	j′	$A_{kj'} + \tau_{kh}$	$\tau_{_{h}}$	S _h	d_h	T_{h}	$w_h T_h$
1	1	5			1	5.00	2.00	5.00	10.00	0.00	0.00
2	2	3		-	1	3.41	10.24	10.24	30.00	0.00	0.00
3	2	7		-	1	4.24	15.47	15.47	15.00	5.47	54.70
4	1	8			1	4.12	22.71	22.71	55.00	0.00	0.00
5	1	2	7		2	12.39	28.12	28.12	20.00	13.12	157.44
6	2		4	-	2	48.06	36.73	48.06	60.00	0.00	0.00
7	1		1	3	3	14.00	60.00	60.00	12.00	54.00	270.00
8	1			3	3	17.07	73.62	73.62	18.00	59.62	119.24
9	2		2	-	2	43.61	82.09	82.09	50.00	37.09	111.27
10	1			5	3	15.10	89.92	89.92	95.00	0.00	0.00

Table 4.3 Results Generated with the Greedy Algorithm

Finally, we obtain the optimal objective function value of $G^* = 712.65$. By tracking the solution, we have,

$$q_{1,1,1}^* = 5$$
, $q_{1,1,4}^* = 8$, $q_{1,1,5}^* = 2$, $q_{2,1,5}^* = 7$, $q_{2,1,7}^* = 1$, $q_{3,1,7}^* = 3$, $q_{3,1,8}^* = 3$, $q_{3,1,10}^* = 5$ and

 $q_{1,1,2}^* = 3$, $q_{1,2,3}^* = 7$, $q_{2,2,6}^* = 4$, $q_{2,2,9}^* = 2$. All the other variables are equal to zero.

4.2.2 Fixed Numbers of DCs and Teams in a Fixed Assignment Environment

We consider the problem P₄ with a fixed number of DCs, n_K , and a fixed number of teams, n_M , (i.e., multiple, but fixed number of teams, each one having a predetermined route). We shall prove that this problem can be solved via dynamic programming in polynomial time. Recall that $H_i = \{h_i(0), h_i(1), h_i(2), ..., h_i(n_i)\}$ is defined to be the sequence of hospitals, or the given route, served by team $i, i=1, ..., n_M$ and $h_i(0) = i$.

Before presenting the polynomial time algorithm, we need to introduce a useful lemma that is critical for proving the time complexity of the proposed dynamic programming algorithm.

Lemma 4.4. The starting time of service at hospital $h_i(l)$ is always of the form

$$S_{h_i(l)} = A_{v(h_i(u)),k(h_i(u))} + \overline{\tau}_{k(h_i(u)),h_i(u)} + \sum_{s=u}^{l-1} \left(p_{h_i(s)} + \tau_{h_i(s),h_i(s+1)} \right) \text{ for some } u = 0, 1, \dots, l,$$

where $k(h_i(u))$ is the DC that serves $h_i(u)$, $v(h_i(u))$ is the last batch that serves $h_i(u)$, $A_{v(h_i(u)),k(h_i(u))}$ is the arrival time of batch $v(h_i(u))$ at DC $k(h_i(u))$ with $A_{v(h_i(0)),k(h_i(0))} = r_i$ and $p_{h_i(0)} = 0$, $\overline{\tau}_{k(h_i(u)),h_i(u)}$ is the travel time from DC $k(h_i(u))$ to hospital $h_i(u)$, and $\tau_{h_i(0),h_i(1)}$ is the travel time from the base of team *i* to $h_i(1)$.

Proof. The starting time of the service at hospital $h_i(l)$ is

- either the arrival time of non-renewable resource at the hospital, which is some batch arrival time at DC $k(h_i(l))$ plus the travel time between DC $k(h_i(l))$ and hospital
- or the service starting time at the hospital visited right before $h_i(l)$ plus the service duration at the previous hospital and the travel time between the previous hospital and the current hospital.

In fact, we have
$$S_{h_i(l)} = \max \{A_{\nu(h_i(l)),k(h_i(l))} + \overline{\tau}_{k(h_i(l)),h_i(l)}, s_{h_i(l-1)} + p_{h_i(l-1)} + \tau_{h_i(l-1),h_i(l)}\}$$
.
Then, $S_{h_i(l-1)}$ can be considered recursively. Thus, we consider the first hospital

 $h_i(u)$ among hospitals served by the same team (team *i*) such that there is no idle time of team *i* between hospital $h_i(u)$ and hospital $h_i(l)$.

Now we consider three cases in terms of u.

 $h_i(l)$,

- If u = l, then the service starting time at hospital $h_i(l)$ is determined by the arrival time of the non-renewable resource. So $S_{h_i(l)} = A_{\nu(h_i(l)),k(h_i(l))} + \overline{\tau}_{k(h_i(l)),h_i(l)}$.
- If $1 \le u < l$, then the service starting time at hospital $h_i(l)$ is the service starting time at hospital $h_i(u)$ plus the total service duration and the outgoing travel times from hospitals $h_i(u)$, $h_i(u+1)$, ..., $h_i(l-1)$. The service starting time at hospital $h_i(u)$ is determined by the arrival time of the non-renewable resource.

Thus,
$$S_{h_i(l)} = A_{v(h_i(u)), k(h_i(u))} + \overline{\tau}_{k(h_i(u)), h_i(u)} + \sum_{s=u}^{l-1} \left(p_{h_i(s)} + \tau_{h_i(s), h_i(s+1)} \right).$$

- If u = 0, then the service starting time at hospital $h_i(u)$ is determined by the arrival time of team *i* at $h_i(1)$.

Thus,
$$S_{h_i(l)} = r_i + \tau_{h_i(0), h_i(1)} + \sum_{s=1}^{i-1} \left(p_{h_i(s)} + \tau_{h_i(s), h_i(s+1)} \right)$$

The formula in the lemma contains all three cases. This completes the proof. \Box

In order to solve this problem, we construct an acyclic graph where a node represents a partial schedule. Let node $(j^1, j^2, ..., j^{n_D} | u^1, u^2, ..., u^{n_M}, S^1, S^2, ..., S^{n_M})$ denote a partial schedule with the first u^i hospitals in H_i being fully served by the first j^k batches from DC k for all $k \in K$ and the starting time of the service at $h_i(u^i)$ being S^i for all $i \in M$. For simplicity, the node is denoted as $\langle j^k | u^i, S^i \rangle$. Let $\kappa(i, l)$ be the index of the DC that $h_i(l)$ is served by.

Node $\langle j^k | u^i, S^i \rangle$ for all $j^k \in B_k$, $u^i_j \in \{1, 2, ..., n_i\}$, $i \in M$, is referred to as valid only when $\sum_{i=1}^{n_M} \sum_{l=1}^{u^i} \sum_{\kappa(i,l)=k} D_{h_i(l)} \leq \sum_{\nu=1}^{j^k} Q_{\nu k}$ for all $k \in K$, implying that the total demand

quantity of the hospitals covered does not exceed the total batch quantity under consideration. From now on, we only consider valid nodes.

By Lemma 4.4, when u^i is given, the set of possible values of s^i has cardinality $O(|H_i| \times |B_{\kappa(i,u^i)}|) = O(|H_i| \times n_B)$. Thus, the total number of nodes is bounded by

$$O\left(\prod_{k=1}^{n_{D}}|B_{k}|\times\prod_{i=1}^{n_{M}}|H_{i}|\times\prod_{i=1}^{n_{M}}(|H_{i}|n_{B})\right)=O(n_{B}^{n_{D}+n_{M}}n_{H}^{2n_{M}}).$$

 $\langle j^k | u^i, S^i \rangle$ has outgoing arcs connecting other nodes and they can be classified into ease of notation, For the node two cases. connected is denoted as $\left\langle \left(j^{k}\right)' | \left(u^{i}\right)', \left(S^{i}\right)' \right\rangle$ which stands for $\left(\left(j^{1}\right)',\left(j^{2}\right)',...,\left(j^{n_{D}}\right)'\Big|\left(u^{1}
ight)',\left(u^{2}
ight)',...,\left(u^{n_{M}}
ight)',\left(S^{1}
ight)',\left(S^{2}
ight)',...,\left(S^{n_{M}}
ight)'
ight)$ Case (i): for $\overline{i} \in M$, let \overline{k} be $\kappa(\overline{i}, u^{\overline{i}} + 1)$. Then, the outgoing arc to $\left\langle \left(j^{k}\right)' | \left(u^{i}\right)', \left(S^{i}\right)' \right\rangle$ is defined when $- Q_{h_{\overline{i}}(u^{\overline{i}}+1)} + \sum_{i=1}^{n_{M}} \sum_{l=1}^{u^{i}} \sum_{v \in [i]} D_{h_{i}(l)} \leq \sum_{v \in [v]}^{j^{k}} Q_{v\overline{k}};$ - $(j^k)' = j^k$ for all $k \in D$; $-\begin{cases} \left(u^{i}\right)'=u^{i} & \text{for } i\neq\overline{i} \\ \left(u^{i}\right)'=u^{i}+1 & \text{for } i=\overline{i} \end{cases};$ $-\begin{cases} \left(S^{i}\right)' = S^{i} & \text{for } i \neq \overline{i} \\ \left(S^{i}\right)' = \max\left\{A_{j^{\overline{k}}, \overline{k}} + \overline{\tau}_{\overline{k}, h_{i}(u^{i}+1)}, S^{i} + p_{h_{i}(u^{i})} + \tau_{h_{i}(u^{i}), h_{i}(u^{i}+1)}\right\} & \text{for } i = \overline{i} \end{cases}$ and its length is $w_{h_{\overline{i}}(u^{\overline{i}}+1)} \cdot \max\left\{0, (S^{\overline{i}})' + p_{h_{\overline{i}}(u^{\overline{i}}+1)} - d_{h_{\overline{i}}(u^{\overline{i}}+1)}\right\}$.

Case (ii): for $\overline{k} \in D$, the outgoing arc to $\left\langle \left(j^{k}\right)' | \left(u^{i}\right)', \left(S^{i}\right)' \right\rangle$ is defined when
$$-\begin{cases} \left(j^{k}\right)' = j^{k} & \text{for } k \neq \overline{k} \\ \left(j^{k}\right)' = j^{k} + 1 & \text{for } k = \overline{k} \end{cases};$$
$$- \left(u^{i}\right)' = u^{i} & \text{for all } i \in M ;$$
$$- \left(S^{i}\right)' = S^{i} & \text{for all } i \in M ,$$

and its length is zero.

Since each node has at most $O(n_K + n_M)$ outgoing arcs, the total number of arcs is bounded by $O((n_K + n_M)n_B^{n_K + n_M}n_H^{2n_M})$, which turns out $O(n_B^{n_K + n_M}n_H^{2n_M})$ when n_D and n_M are considered fixed constants.

The origin is (0, 0, ..., 0 | 0, 0, ..., 0, 0, 0, ..., 0). Since we have to cover all hospitals we define a dummy destination node and connect all nodes such that $u^i = n_i$ with zero length for all $i \in M$. Then the shortest path from the origin to the destination identifies the schedule for dispatching from batches to hospitals.

Since the shortest path problem defined on an acyclic graph G = (N, A) from the origin to all the nodes can be solved in O(|A|) time (Ahuja et al., 1990) where N and A are the node set and the arc set, respectively, the problem can be solved in $O(n_B^{n_K+n_M} n_H^{2n_M})$ time.

Theorem 4.4. Problem P₄ with n_k and n_M fixed can be solved in $O(n_B^{n_k+n_M} n_H^{2n_M})$.

We present a numerical example that is solved by our proposed Dynamic

Programming algorithm to optimality. In this example, we are given four hospitals ($n_H = 4$), two distribution centers ($n_K = 2$), and two travel teams ($n_M = 2$). Each DC has two batches ($|B_1| = |B_2| = 2$). DC 1 is located at (0, 0) and serves hospitals 1 and 3, while DC 2 is located at (0, 5), and serves hospitals 2 and 4. Team 1 starts from base (0, 0) and serves hospitals 1 and 2, while Team 2 starts from base (5, 5) and serves hospitals 3 and 4. Both teams start at time 0. The parameters are given in Tables 4.4-4.5, where coordinates are used to calculate the transportation time between locations assuming a speed of 1.

DC k	DC	21	D	C 2
Coordinates in the x-y axis	(0,	0)	(0	0, 5)
Batch j	1	2	1	2
Batch Arrival time A_{kj}	3	7	2	8
Batch Capacity Q_{kj}	15	20	25	10

Table 4.4 Parameters for DCs and Batches (Dynamic Programming)

Hospita	Demand	Penalty Cost	Due Date	Duration	Coordinate in
1	\mathcal{Q}_h	W_h	d_h	P_h	the x-y axis
h					
1	10	5	10	3	(2, 0)
2	20	5	10	3	(0, 1)
3	15	10	10	5	(2, 5)
4	10	5	15	3	(0, 8)

Table 4.5 Parameters for Hospitals (Dynamic Programming)

According to Tables 4.4 and 4.5, the transportation times are as follows:

$$\overline{\tau}_{1,1} = 2, \ \overline{\tau}_{1,3} = 5.4, \ \overline{\tau}_{2,2} = 4, \ \overline{\tau}_{2,4} = 3 \text{ and } \tau_{h_1(0),1} = 2, \ \tau_{1,2} = 2.2, \ \tau_{h_2(0),3} = 3, \ \tau_{3,4} = 3.6$$

The optimal solution is $z_{1,1,3} = z_{2,1,2} = z_{1,2,1} = z_{2,2,4} = 1$, (See Figure 4.2) and the optimal objective function value is 105.



Figure 4.2 A Numerical Example for the Proposed Dynamic Programming Algorithm 4.3 Multiple DCs in an Open Assignment Environment

In this section, we consider the case with multiple DCs in an open assignment environment. We first show that even with a single team and any fixed number of DCs, the problem is already NP-hard. Then we consider two special cases with variable numbers of DCs and medical teams, and show that both of these two special cases can be solved in polynomial time.

Theorem 4.5 The problem with two DCs and a single medical team in an open assignment environment, i.e., P(O, 2, 1), is NP-hard.

Proof. In this proof, we will use a reduction to the PARTITION Problem (which had been defined earlier in the proof of Theorem 4.1). Let H be $\{1, 2, ..., n\}$ with $(p_h, D_h, d_h, w_h) = (0, a_h, 0, 1)$. We have two DCs, each having two batch arrivals with the following information: $A_{k1} = 0, A_{k2} = A$ and $Q_{1k} = Q_{2k} = \frac{A}{2}$. All transportation times are zero. If the demand at hospital h is satisfied at time 0, then its tardiness is zero, while if the demand is satisfied at time A, then its tardiness is A. Therefore, there is a partition if and only if there is a schedule with a total tardiness of zero. \Box

Based on the theorem above, we can state that the problem with a single team and a fixed number of DCs is NP-hard.

Theorem 4.6 The problem with a single team and a variable number of DCs in an open assignment environment, i.e., $P(O, n_{\kappa}, 1)$, is strongly NP-hard.

Proof. In this proof, we will use a reduction to 3-PARTITION (which had been defined

earlier proof Theorem 4.1). Let in the of Hbe $\{1, 2, \dots, 3t\}$ with $(p_h, D_h, d_h, w_h) = (0, a_h, 0, 1)$. We have *t* DCs, each having two batch arrivals with the following information: $A_{k1} = 0, A_{k2} = A$ and $Q_{1k} = Q_{2k} = A$. All transportation times and all processing times are zero. If the demand at hospital h is satisfied at time 0, then its tardiness value is zero, while if it is satisfied at time A then its tardiness value is A. Because of the single sourcing constraint, each demand point can be served by only one DC and thus the first batch of each DC can serve at most 3 demand points. Thus, there is a 3-partition if and only if there is a schedule with the total tardiness being zero. \Box

Now, we consider two special cases with variable n_K and n_M that are strongly polynomial time solvable.

The first case can be denoted as $P(O, n_K, n_M)$ with $D_h = D$, $n_H \le n_M$, and with n_D and n_M being variable. That is, each team may visit at most one hospital, and the hospital order sizes for non-renewable resources are identical. In this case, the respective problem, $P(O, n_K, n_M)$ with $D_h = D$ and $n_H \le n_M$, can be solved in strongly polynomial time. While different DCs may receive their batches from upstream suppliers at different points in time, we do assume that $\sum_{\forall k} \sum_{\forall j} Q_{jk} \ge \sum_{\forall h} D_h$. Let \mathbf{P}_5 denote this problem.

Let *D* be the common order size of all hospital and let τ_h be the team arrival time at hospital *h*. Since the sizes of hospitals' demands for non-renewable resources are all identical, the outgoing quantity from a DC at a certain time moment is always a multiple

of *D*. Now consider the following algorithm.

Algorithm 4.4 (Transportation Algorithm)

Step 1. Let

$$\Delta = 0$$

For $j = 1$ to $|B_k|$
$$b_{kj} = \left\lfloor \frac{Q_{jk} + \Delta}{D} \right\rfloor, \quad Q'_{jk} = D \cdot b_{kj}, \quad \Delta = Q_{jk} + \Delta - Q'_{jk}$$

where b_{kj} denotes the maximum number of orders that the *j*-th batch at DC *k* (plus residuals from previous batches) may fulfill.

Step 2. Formulate the following optimization problem:

- Supplier (k, j) refers to the j-th batch of DC k that has a capacity of b_{kj}, for all k ∈ K, j ∈ B_k;
- Customer (h) refers to hospital h that has a unit demand size, for all $h \in H$;
- The edge cost between supplier (k, j) and customer (h) is the resulting tardiness if a unit of non-renewable resource is delivered from supplier (k, j) to customer (h), which is defined as

$$C_{jkh} = w_h \cdot \max\left\{\max\left\{A_{jk} + \overline{\tau}_{kh}, \tau_h\right\} + p_h - d_h, 0\right\};$$

• Let x_{jkh} be a binary variable, and $x_{jkh} = 1$ if supplier (k, j) is assigned to serve customer (h), and 0 otherwise, which leads to the following model:

Minimize
$$\sum_{k \in K} \sum_{j \in B_k} \sum_{h \in H} C_{jkh} x_{jkh}$$
(4.1)

Subject to $\sum_{h \in H} x_{jkh} \le b_{kj} \qquad \text{for all } k \in K, j \in B_k \qquad (4.2)$ $\sum_{k \in K} \sum_{j \in B_k} x_{jkh} = 1 \qquad \text{for all } h \in H \qquad (4.3)$ $x_{jkh} \in \{0,1\} \qquad \text{for all } k \in K, j \in B_k, h \in H \qquad (4.4)$

The constraint matrix (4.2) - (4.4) is totally unimodular. Therefore, the problem defined by (4.1) - (14.4) can be solved to optimality by relaxing binary variables x_{jkh} to continuous variables in [0, 1] as a transportation problem, which has the computational complexity of $O(UV^2(\log U + V \log V))$ where U and V are the cardinalities of the supplier set and the demand set, respectively (Brenner, 2008). Since $U = \sum_{k \in D} |B_k| = n_B$ and $|V| = n_H$, the following theorem holds.

Theorem 4.7 Problem P₅ can be solved in $O(n_B n_H^2 (\log n_B + n_H \log n_H))$ time.

Special case defined by P_5 refers to the situation where the orders for the non-renewable resource are all identical and where each medical team visits only one hospital. Such a situation occurs in practice when the hospitals' orders for non-renewable resources are fulfilled, for example, by the truckload and when there are sufficient

medical teams on the ground, each being assigned to serve a designated hospital with flooding relief from patients. During the recent hurricane Isaac many (http://www.memphis-umc.net/news/detail/1663), full truckloads were dispatched to various sites of Louisiana (http://www.therepublic.com/view/story/c5070625abc 14340a3065b5ede228ef6/OK--Isaac-Relief-Supplies). Campbell et al. (2008) also reported unit-sized demand in the vehicle routing problem encountered during relief efforts.

Let us now consider another polynomial time solvable case where each DC serves at most one team (and thus all the demand points that team visits) for $n_M < n_K$. The resulting problem is then to find the best matching between the set of DCs and the set of teams. Let **P**₆ denote this problem. For the sake of simplicity, we may assume that $n_M = n_K$ by adding $n_K - n_M$ dummy teams with zero demand. Let c_{ki} be the cost of assigning DC k to serve the given route of team i, and x_{ki} be the binary decision variable which equals to 1 if DC k is assigned to serve the given route of team i and 0 otherwise. Then c_{ki} can be calculated as follows.

Step 1. If $\sum_{j \in B_k} Q_{jk} < \sum_{h \in \sigma_i} D_h$, then $c_{ki} = \infty$. DONE

Step 2. $c_{ki} = 0$

 $j \leftarrow 1$, tempQ $\leftarrow Q_{k1}$, tempT $\leftarrow \max\left\{r_i, A_{1k} + \overline{\tau}_{k,\sigma_i(1)}\right\}$ For h = 1 to $|\sigma_i|$ 104

While (tempQ $< Q_{\sigma_i(h)}$)

$$j \leftarrow j + 1$$

$$\operatorname{tempQ} \leftarrow \operatorname{tempQ} + \mathcal{Q}_{jk}$$

$$\operatorname{tempT} \leftarrow \max\left\{\operatorname{tempT}, A_{jk} + \overline{\tau}_{k,\sigma_i(h)}\right\}$$

$$c_{ki} \leftarrow c_{ki} + w_{\sigma_i(h)} \max\left\{\operatorname{tempT} + p_{\sigma_i(h)} - d_{\sigma_i(h)}, 0\right\}$$

$$\operatorname{tempT} \leftarrow \operatorname{tempT} + p_{\sigma_i(h)} + \tau_{\sigma_i(h),\sigma_i(h+1)}$$

$$\operatorname{tempQ} \leftarrow \operatorname{tempQ} - \mathcal{Q}_{\sigma_i(h)}$$

Given $\{c_{ki} | \text{ for all } i \in M, k \in K\}$, we formulate and solve the following problem.

$$\text{Minimize } \sum_{k \in K} \sum_{i \in M} c_{ki} x_{ki} \tag{4.5}$$

Subject to
$$\sum_{k \in K} x_{ki} = 1$$
 for all $i \in M$ (4.6)

$$\sum_{i \in M} x_{ki} = 1 \qquad \text{for all } k \in K \tag{4.7}$$

 $x_{ki} \in \{0,1\}$ for all $i \in M, k \in K$ (4.8)

This is the well known assignment problem and the time complexity of solving an assignment problem is known to be $O(|V|^3)$ where V is the vertex set by Munkres (1957). Therefore, our problem can be solved in $O(n_K^{3})$. **Theorem 4.8** Problem P₆ can be solved in $O(n_{\kappa}^{3})$ time.

Note that problem P_6 can be found in practices where each medical team has its own dedicated supplier (i.e., DC) that provides non-renewable resources and transportation vehicles to support the travel team with relief operations. For example, during the disaster relief of Haiti earthquake, AmeriCares dispatched over 200 medical teams together with various medical supplies, representing the US support to the Haiti survivors (http://www.americares.org /whatwedo/mop/). When each disaster area is served by one U.S. medical team together with the medical supplies of AmeriCares, the assumptions of P_6 hold.

4.4 Overview of Complexity Results

We have studied the operations scheduling problem involving both renewable and non-renewable resources. To gain an understanding regarding the structural properties of the problem, we made a distinction between two cases: the DC-hospital assignments are either given and fixed (F), or open (O) and to be optimized. Under the various conditions, we identify NP-hard cases along with complexity proofs and polynomial time solvable cases along with time complexity analysis. We summarize the results in the following tables.

n _K	n _M	1	Fixed	Variable
1		$\mathbb{P}(O(n_B + n_H))$	$P\left(O\left(n_B^{1+n_M}n_H^{2n_M}\right)\right)$	SNP

	\uparrow	$P(O(n_H))$	Theorem 4.2
		Lemma 4.1	'
		$\mathbf{P}(O(n_{H}^{2}))$	
		Lemma 4.2	
	$\mathbf{P}(O(n_B + n_H))$	$\mathbf{P}\left(O\left(n_{B}^{n_{D}+n_{M}}n_{H}^{2n_{M}}\right)\right)$	SNP
Fixed	\uparrow	Theorem 4.4	↓
Variable	$\mathbb{P}(O(n_B + n_H))$	Onen	SNP
	Theorem 4.3	Open	\downarrow

P: Polynomial time solvable (complexity), NP: NP-hard. SNP: Strongly NP-hard

Table 4.6	Complexity	Framework of	f Problem	$\mathbf{P}(F, n_K, n_M)$
-----------	-------------------	--------------	-----------	---------------------------

n _M	1	Fixed	Variable
		$\mathbf{P}\left(O\left(n_B^{1+n_M}n_H^{2n_M}\right)\right)$	
		Theorem 4.4	
1	$\mathbf{P}(O(n_B + n_H))$	$P(O(n_H))$	SNP
1	Theorem 4.3	Lemma 4.1	Theorem 4.2
		$P(O(n_H^2))$	
		Lemma 4.2	
Fixed	NP	NP	SNP
FIXEU	Theorem 4.5	\rightarrow	\downarrow
Variable	SNP	SNP	SNP
variaure	Theorem 4.6	\rightarrow	\downarrow

P: Polynomial time solvable (complexity), NP: NP-hard. SNP: Strongly NP-hard

Table 4.7 Complexity Framework of Problem $P(O, n_K, n_M)$

4.5 Conclusion

The results obtained in this chapter, especially those that are proven to be polynomial time solvable, can be used to design heuristics for solving more general versions of this problem. While the results throughout this paper assumed predetermined routes of the medical teams (i.e., the renewable resources), routing decisions are allowed in practice. If we relax the non-renewable resource requirement constraints, then the remaining problem is similar to a Vehicle Routing Problem (VRP) with tardiness costs; this implies that well known VRP heuristic algorithms can be used to find reasonable solutions for the routes of the medical teams. The next step is to determine the DC-hospital assignment. Due to the combinatorial nature of the single sourcing constraint, this step also needs a heuristic approach. For each given route, we regard the team's arrival time as the due date for the non-renewable resource's arrival at each hospital. We now define and solve a transportation problem in which supply nodes are batches in DCs and demand nodes are hospitals. Even though the optimal solution of the transportation problem does not satisfy the single sourcing constraints, we may derive a feasible DC-hospital assignment by adjusting such optimal solutions. After the routing and the DC-hospital assignments have been determined, the results in this study can be utilized. When the numbers of DCs and teams are relatively small, we can apply the dynamic programming algorithm in Theorem 4.4. Otherwise, we can make a sequence of teams and repeatedly apply the results of Theorem 4.3 and obtain a greedy solution.

A second heuristic approach for a general version of this problem can be described as follows. We can make the decisions with regard to the DC assignments and the batch assignments simultaneously when the problem is more restricted. For example, when a medical team visits only one hospital and/or when a DC is dedicated to supply a set of hospitals served by the same medical team, we can apply Theorems 4.7 and 4.8, respectively. Even if the problem does not satisfy such a restricted property, we may simplify the problem first, solve the simplified problem optimally and adjust the solution obtained in order to ensure feasibility of the original problem. We may assume that a part of the problem satisfies the restricted property and produce an optimal solution for that part. By applying this procedure repeatedly, we can obtain a heuristic solution.

We may also consider a third heuristic procedure by integrating the routing decisions with the allocation of resources. We already mentioned in the introduction that when the amount of non-renewable resource available at time zero is sufficient, then the problem turns out to be equivalent to a parallel machine scheduling problem with sequence dependent setup times and release dates and with the total weighted tardiness as objective. Lee and Pinedo (1997) proposed an algorithm that computes the priority levels of all yet to be scheduled jobs, assigns the job with the highest priority and repeats this procedure until all jobs are scheduled. The main difference between the parallel machine scheduling problem and our problem is that we have to consider also the non-renewable resources. According to Lee and Pinedo (1997), when evaluating the priority levels, we have to take all the parameters such as the weights, due dates, and release dates into account. Then we can modify the algorithm by approximating the release date of a job with the maximum value of the original release date and the earliest possible non-renewable resource arrival time. After a job has been scheduled and all parameters have been updated, we can repeat the procedure.

The following table summarizes the three heuristic approaches for the general version of our multi-resource operations scheduling problem.

Sub-Problems	Approach 1	Approach 2	Approach 3
Routing			
Desision			Revised version of a
Decision	VRP heuristic	VRP heuristic	Houristic for Parallal
\downarrow			
			Machine Scheduling
DC Assignment	Transportation		with Common
\downarrow	Problem heuristic	Theorems 4.7 and	with Sequence
·		Theorems 1.7 and	Dependent Setup
Batch	Theorems 4.3 and	4.8	T '
Assignment	4.4		limes
Assignment	4.4		

Table 4.8 Heuristic Approaches for a General Problem

Chapter 5. A Rolling-Horizon Based Heuristic for Solving P

In this chapter, we first lay out the theoretical background for Rolling-Horizon (RH) heuristic for solving Problem P in section 5.1. The RH heuristic is formally stated in details in section 5.2, followed by section 5.3 where empirical study is carried out to evaluate the effectiveness of RH heuristic.

5.1 The Single Batch Problem

If each DC receives only a single batch from its upper stream supplier, problem **P** can be reduced to a variation of a network flow problem (Ahuja, et al., 1993). To see this, first assume that each traveling team is assigned to visit only a single hospital (i.e., the case with unit tour length). Let A_k be the arrival (or release) time of the single batch at DC k, let Q_k be the respective batch size, let r_h be the release time of team h (since each team is uniquely assigned to each hospital, we may use h to denote both hospital h and the team assigned to serve h), let τ_h be the travel time from the team base to hospital h, let $\overline{\tau}_{kh}$ be the shipping time from DC k to hospital h, let q_{kh} be the quantity shipped from DC k to hospital h, and let z_{kh} be the binary variable that is equal to 1 if hospital h is served by DC k. Under these assumptions, problem **P** is reduced to the following (also see Figure 5.1(a)):

$$P_{\gamma}: Minimize \sum_{\forall h \in H} \max \{0, \max_{\forall k \in K} \{(A_k + \overline{\tau}_{kh})z_{kh}\} + p_h - d_h, r_h + \tau_h + p_h - d_h\}$$

s.t.
$$\sum_{\forall k \in K} q_{kh} = Q_h; \sum_{\forall h \in H} q_{kh} \le Q_k; q_{kh} \le z_{kh}Q_k; q_{kh} \ge 0; z_{kh} \in \{0,1\}, \forall h \in H, k \in K$$

Lemma 5.1. \mathbf{P}_7 can be solved in $O((\sum_{k \in K} n_{B_k})n_H^2(\log(\sum_{k \in K} n_{B_k}) + n_H \log n_H)).$

Proof. Problem \mathbf{P}_7 is equivalent to the known Time Minimizing Transportation Problem (TMTP) if we define $t_h = \max \{0, \max_{k \in K} \{(A_k + \overline{\tau}_{kh}) z_{kh}\} + p_h - d_h, r_h + \tau_h + p_h - d_h\}$. It is known that TMTP is solvable in strongly polynomial time (Sonia and Puri, 2004). Brenner (2008) proposed an algorithm for solving such problems at $O((\sum_{k \in K} n_{B_k}) n_H^2 (\log(\sum_{k \in K} n_{B_K}) + n_H \log n_H))$.



(a) Single-Batch with Unit-Tour Length. (b) Single-Batch with Tour Length ≤ 2

Figure 5.1 Examples of the Multiple-DC Single-Batch Process

We now consider the case where each DC receives only a single batch of non-renewable resources, and the tour length of each team is no more than two. Problem P is then reduced to the following one, or P_8 (see Figure 5.1(b)), in which $\langle h', h \rangle$ denotes the two consecutive locations visited by a team. If *h* stands for the first hospital (or the only

hospital) on a given medical team's route, then *h*' represents the respective team base or a dummy hospital, and we have $s_{h'} = r_i$, $p_{h'} = 0$ and $\tau_{h'h} = \tau_{ih}$.

$$\begin{array}{ll} P_8: \textit{Minimize} & \sum_{\forall h \in H} T_h \\ s.t. \\ \sum_{\forall k \in K} q_{kh} = Q_h; & \forall h \in H \\ \sum_{\forall h \in H} q_{kh} \leq Q_k; & \forall k \in K \\ q_{kh} \leq z_{kh}Q_k; & \forall h \in H, k \in K \\ (A_k + \overline{\tau}_{kh})z_{kh} \leq S_h; & \forall h \in H, k \in K \\ s_{h'} + p_{h'} + \tau_{h'h} \leq S_h, & \forall h \in H, h' \in M \cup H \\ T_h \geq s_h + p_h - d_h; & \forall h \in H, k \in K \\ q_{kh} \geq 0; z_{kh} \in \{0,1\}; & \forall h \in H, k \in K \end{array}$$

The following process transforms a given P_8 into the Min-Cost Max-Flow Problem. Step 1: Define Nodes

- 1. Establish the source and sink nodes of the network.
- 2. Each node in the first layer of the network represents the single batch received at each DC. Since a total of n_K DCs are involved in the problem, we have n_K nodes in this layer.
- 3. The nodes in the second layer of the network represent the decomposed batches from DCs to different hospitals. Each DC may serve all hospitals, with different service quantity that could be even zero. Considering the problem has n_K DCs and n_H hospitals, $n_K n_H$ nodes exist in this layer.
- 4. Each hospital corresponds to multiple nodes in the third layer of the network, each of which represents a demand fulfillment scenario in order to serve this hospital:

- a. The first hospital along the route of each medical team corresponds to n_K nodes (each of which represents a scenario that the latest batch this hospital receives is from the DC with the k-th latest arrival time among all DCs, $1 \le k \le n_k$ Specifically, the first node among the n_k nodes represents the scenario that the supplies come from all DCs; the second node among the n_{K} nodes represents the scenario that the supplies come from all DCs except the one that has the latest arrival time; the third node corresponds to the scenario that the supplies come from all except the two DCs with the latest arrival time, ..., and the last $(n_{\kappa} - th)$ node represents the scenario that the supplies only come from the DC that has the earliest arrival time. Note that the flow into any of these nodes can be zero. Since a total of n_M medical teams are serving hospitals, we have a total of n_M such hospitals that are scheduled at the beginning of their routes. Therefore, there exist $n_K n_M$ this kind of nodes in the third layer of this network.
- b. If a medical team visits two hospitals in its route, then the second hospital corresponds to n_K^2 nodes. This is because for this type of hospitals, the service scenario relies on not only the flow patterns (n_K) of non-renewable resources as we introduced above in part i, but also that of the medical teams (The team may arrive at this hospital after it fulfills the demand at the first hospital in its route with any of the n_K service scenarios we introduced in part i above). Therefore, a total of $n_K^2(n_H n_M)$ nodes exist for such hospitals.

Step 2: Define Edges, and the corresponding capacities and costs

- The capacity of the edge from the source node to any node in the first layer (Batches of DCs) is just the single batch size, and the cost is zero because no tardiness is involved here.
- The capacity of the edge from any node in the first layer to the sink node is infinite, but the cost is very large, such that only unused supply will be allowed to go this way.
- 3. The capacity and cost of the edge from any node in the first layer to any node in the second layer are infinite and zero, respectively.
- 4. The capacity of the edge from any node in the second layer to any node in the third layer is infinite, but the costs are different. For any hospital *h*, suppose it receives the batch from DC k_1 first, that from DC k_2 second, ..., and that from DC k_{n_k} the last.
 - a. If hospital h is the first hospital served by a medical team in its route, based on our discussions at Step 1.d part i, it has n_K associated nodes in the third layer. In addition, all inbound flows to the first node associated with hospital h have cost max. $\{0, r_i + \tau_{ih} + p_h - d_h, A_{k_{n_k}} + \overline{\tau}_{k_{n_k}h} + p_h - d_h\}$ (the second and third elements in this maximum function, respectively, represent the tardiness at hospital h caused by the medical team and the non-renewable resources), those to its second node have $\cos t \max \left\{0, r_i + \tau_{ih} + p_h - d_h, A_{k_{n_{K-1}}} + \overline{\tau}_{k_{n_{K-1}}h} + p_h - d_h\right\}, \dots, \text{ and those to the}$ last node have cost max. $\{0, r_i + \tau_{ih} + p_h - d_h, A_{k_1} + \overline{\tau}_{k_1h} + p_h - d_h\}$.

- b. If hospital *h* is the second hospital served by a medical team in its route, based on our discussions at Step 1.d part ii, it has n_{K}^{2} associated nodes in the third layer, which we use $\{n_{st}\}$, $1 \le s, t \le n_{K}$, to denote. Then the inbound flows to node n_{st} have $\max . \{0, \max\{r_{i} + \tau_{ih'}, A_{k_{n_{K}-t+1}} + \overline{\tau}_{k_{n_{K}-t+1}h'}\} + p_{h'} + \tau_{h'h} + p_{h} - d_{h},$ $\cot A_{k_{n_{K}-s+1}} + \overline{\tau}_{k_{n_{K}-s+1}h} + p_{h} - d_{h}\}$.
- 5. For each hospital, outbound flows from its corresponding nodes in the third layer will first be collected at a temporary node, before arriving at the sink node. All the related edges have zero cost and infinite capacity except the edge between the temporary node and the sink, whose capacity is equal to the demand, D_h , of the hospital.

By the above transformation process, the following result holds.

Lemma 5.2. Problem P₈ can be solved in $O(n_K^6 n_H^4 (\log n_K + \log n_H))$.

Proof. Since an equivalent min-cost max-flow problem can be constructed based on P_8 and the min-cost max-flow problem is known to be solvable in $O(m \log n \ (m + n \log n))$, according to Orlin (1993), where $n = n_K + n_K n_H + n_K n_M + n_K^2 (n_H - n_M)$ and

$$m = n_{K} + n_{K}n_{M} + n_{K}^{2}n_{H} + n_{K}^{2}n_{H}n_{M} + 2n_{K}^{2}(n_{H} - n_{M}) + n_{K}^{3}n_{H}(n_{H} - n_{M})$$

The claim holds^[1]. \diamond

Figure 5.2 illustrates an example of solving P_8 as a min-cost max-flow problem. In this

example, we are given two DCs (each having a single batch), two medical teams, and three hospitals (where hospitals 1 and 3 are assigned to the team with tour length two, while hospital 2 is assigned to the other team with a unit tour length). Note that hospitals 1 and 2 are the first hospitals on the routes of medical teams 1 and 2, respectively. Layer 1 has two nodes, each of which represents a DC. Layer 2 has six nodes, each of which represents a unique batch-hospital assignment pair. Layer 3 has eight nodes, each of which defines a possible scenario of allocating non-renewable resources from the two DCs to an individual hospital. The total number of scenarios associated with hospital 3 is squared because it is the second hospital in the service route, and how it is served depends on the service scenarios of hospital 1 as well.



Figure 5.2 Flow Chart of an Example Min-Cost Max-Flow Problem¹

While Lemmas 5.1 and 5.2 assume each DC receives only a single batch from its

¹ Note that, the case with multiple batches assigned to each DC can also be solved as a min-cost max-flow problem, however, at a much higher level of computational complexity.

upper stream supplier, we can show that, in a more general situation with multiple batches to each DC, as long as the minimum batch size is no less than the maximum of order sizes, i.e., $\min_{k \in K, j \in B_k} Q_{kj} \ge \max_{h \in H} D_h$, the following result holds.

Lemma 5.3. If $\min_{k \in K, j \in B_k} Q_{jk} \ge \max_{h \in H} D_h$ then there exists an optimal solution to problem $\mathbf{P}, \Sigma^* = \{z_{jkh}^*, q_{jkh}^*, S_h^*, T_h^*\}$, such that each hospital *h* receives non-renewable resources from at most two *consecutive* batches that originate from the same DC. –

Proof. Suppose in an optimal schedule S', there is at least one hospital, say h, that receives supplies from more than two batches from the same DC, k. Let j^a and j^b denote the first and last batches, respectively, that serve h from this DC, and $j^a < j^b - 1$. Let q_{j^akh} and q_{j^bkh} denote the quantities shipped to h from batches j^a and j^b , respectively. We can then reassign the shipping quantity q_{j^akh} from batch j^a , originally serving hospital h, to serve other hospital(s) originally assigned to batch $j^b - 1$, and then allocate an equivalent quantity $q = q_{j^akh}$ from batch $j^b - 1$ to serve h (note that the assumption $\min_{k \in K, j \in B_k} Q_{jk} \ge \max_{h \in H} D_h$ ensures $Q_{j^b - 1,k} \ge q_{j^akh}$ and therefore such a reassignment is always feasible). This adjustment is illustrated in Figure 3. Doing so does not change the tardiness of hospital h while making the tardiness of other hospital(s) originally served by $j^b - 1$ no more than that in the original optimal solution S'. This implies that the new solution is still optimal. Repeating this process will reach an improved solution such that each hospital receives supplies from at most two consecutive batches from the same DC.



Figure 5.3. An Illustrative Example of Lemma 5.3

5.2 A Rolling-Horizon (RH) Heuristic for Solving P

We propose a rolling-horizon based greedy heuristic algorithm for solving \mathbf{P} . The design of this solution approach is motivated by the results of Lemmas 5.1 and 5.2; if each DC has only one or two batches of non-renewable resources to allocate and if the tour of each traveling team covers very few hospitals, then the reduced problem becomes easier to solve.

Similar to the approach used in most existing rolling horizon based heuristics (e.g., de Araujo et al. (2007), Beraldi et al. (2008), Lei et al. (2009), and Li et al. (2010)), our algorithm also follows an iterative process to search for a feasible solution to the original problem **P**. Each iteration starts with a given (heuristically constructed) sequence of hospitals that have not yet been served (i.e., the first iteration starts will all n_H

hospitals). During each iteration, we construct a sub-problem (see Figure 4 below) that focuses on the <u>next two batches</u> to be allocated from each DC and <u>the next N hospitals</u> in the given sequence, $N \le n_H$, being a search parameter. Let Ω_1 be the collection of the first available batches across all DCs, and Ω_2 be the collection of the second available batches across all DCs, where $n_K \ge |\Omega_1| \ge |\Omega_2|$. Let $\Omega = \Omega_1 \cup \Omega_2$. We then solve this two-batch *N*-hospital sub-problem with objective function (3.1) optimally.



Figure 5.4 A Graphical Illustration of a Two-Batch N-Hospital (N=7) Sub-Problem Solved by the RH Algorithm during the Search Process

Such a sub-problem contains a significant less number of integer variables, and can be solved quickly using GUROBI (a commercial optimizer) typically within few CPU seconds. After a sub-problem is solved, we permanently fix the resource allocation of the batches in set Ω_1 , while release the batches in set Ω_2 together with the hospitals supplied by the batches in Ω_2 to the next iteration. For those hospitals whose orders are only *partially supplied* from the batches in set Ω_1 , we update their order sizes by subtracting the partial shipment received in the current iteration, and then release such hospitals with revised order sizes to the next iteration. We also update the travel team service completion time at the respective hospitals, and then go to the next iteration. The iteration process repeats itself until all the hospitals are served, and yields a new feasible solution to the original problem **P**. The sequence (i.e., a permutation of n_H hospitals) at which hospitals receive the supplies from DCs in this new feasible solution to **P** is then used as the initial hospital sequence for the next round to generate another feasible solution. The search terminates when the newly obtained feasible solution no longer improves the previous one. Let *RH* denote this rolling horizon based heuristic. The details of the algorithm are described below.

Algorithm RH {*Input: An initial sequence* σ_0 *and an initial objective function value* G_0 }

<u>Step 1</u>. Construct a sub-problem, consisting of the next two batches from each DC and the next N customers in sequence σ_0 , where parameter N is determined by

$$\sum_{1 \leq h \leq N} D_h \leq \sum_{k \in K} Q_{1,k} + Q_{2,k} < \sum_{1 \leq h \leq N+1} D_h$$

so that the total demand of the next *N* hospitals in σ_0 does not exceed the total supplies of the next two batches of the n_K DCs. Let Ω_1 be the collection of the first available batches cross all the DCs, and Ω_2 be the collection of the second available batches cross all the DCs. Solve the

two-batch N-hospital sub-problem optimally against objective function (1). Note that the hospital service starting times, S_h , for those hospitals included in the sub-problem, may be different from those obtained previously.

Step 2. Permanently fix the non-renewable resource allocation of the batches in set Ω_1 , delete such batches from further consideration, and release the batches in set Ω_2 to the next iteration. For hospitals whose orders have been fully supplied by the batches in set Ω_1 , update their service starting times and team departure times, and then permanently remove such hospitals from sequence σ_0 . For those hospitals whose orders are partially supplied by the batches in set Ω_1 , update their order sizes by subtracting the partial shipment received in the current iteration and then release such hospitals, together with their revised order sizes, to the next iteration (i.e., keep such hospitals in σ_0).

Step 3. If $\sigma_0 \neq \emptyset$, return to Step 1. Otherwise, a new feasible solution is obtained which defines a new sequence σ and a new objective function *G*. If $G \ge G_0$, which means the newly obtained feasible solution does not improve the previous solution, terminate the search with the best feasible solution obtained so far. Otherwise, replace σ_0 by σ , and G_0 by *G*, return to Step 1 for the next loop.

The flowchart of algorithm *RH* is given in Figure 5.5. The search starts with a given hospital sequence constructed upon the optimal solution of a linear programming (LP) relaxation of **P** with binary variables z_{jkh} , $\forall k \in K$, $j \in B_k$, $h \in H$ relaxed from $\{0,1\}$ to [0,1]. Let σ_0 denote this initial hospital sequence, and $|\sigma_0| = n_H$. Note that by rounding up the values of relaxed binary variables, $\{z_{kjh}\}$, in this LP solution, we can obtain an initial feasible solution to **P**, which will be improved through the remaining search process.



Figure 5.5 A Flowchart of the Search Process by the Proposed RH Algorithm

It should be pointed out that the proposed algorithm *RH* does not aim at solving the emergency operations scheduling problem for the entire disaster relief process. Instead, it

aims at driving a quick solution for a given group of hospitals, each with a specific demand on the non-renewable resources, at a particular time point of the relief process.

While an analytical error bound of the RH algorithm to the original problem **P** is difficult to derive, it can be constructed under certain assumptions on a simplified network. One example of this is given in Lemma 5.4, where we show the error gap between the optimal solution and the solution obtained by the *RH* algorithm is bounded from above by the maximum inter-arrival time of two consecutive batches at the DC. We derive the error bound for a simplified network (see Figure 5.6) with a single DC, four batches of unit (identical) batch sizes, two traveling teams, each visits two hospitals, and a total of four hospitals with unit demand quantity, which equals to the batch size, on non-renewable resource. Let's denote this special case as P_9 . Without loss of generality, we assume $A_1 < A_2 < A_3 < A_4 \qquad , \qquad w_1 > w_2, w_3 > w_4,$ that and $w_1 \ge w_3$ (where $w_h = \overline{\tau}_h + p_h - d_h$, h = 1, 2, 3, 4). Let $v_h = \overline{\tau}_{h-1} + p_{h-1} + \tau_{h-1,h} + p_h - d_h$ for h = 2, 4• By triangular inequality, we have $v_h = \overline{\tau}_{h-1} + p_{h-1} + \tau_{h-1,h} + p_h - d_h > p_{h-1} + \overline{\tau}_h + p_h - d_h = p_{h-1} + w_h$.



Figure 5.6 Special Case P₉

Lemma 5.4. For any given instance of P₉, the gap between the optimal objective function value T^* and the objective function value achieved by RH, T^{RH} , is bounded by the maximum inter-arrival time between two consecutive batches, i.e., $|T^{RH} - T^*| \le \max_{1 \le i \le 3} (A_{i+1} - A_i)$.

Proof. By the assumption of case **P**₉, $Q_j = D_h, \forall j, \forall h$. Therefore, the DC only delivers its supplies to a hospital after the shipment to the immediate predecessor of that hospital is fulfilled (see Lemma 4.3). The optimal assignment of batches to hospitals is thus a permutation of hospital set $H=\{1, 2, 3, 4\}$ with each hospital served by exactly one batch from the DC. Let π be a permutation of H.

The optimal objective value can now be expressed as

$$\begin{split} T^* &= Min. \{T(\pi) | \forall \pi\} = \min \{ \\ \max \{0, A_1 + w_1\} + \max \{0, A_2 + w_2, A_1 + v_2\} + \max \{0, A_3 + w_3\} + \max \{0, A_4 + w_4, A_3 + v_4\}, \\ \max \{0, A_1 + w_1\} + \max \{0, A_3 + w_2, A_1 + v_2\} + \max \{0, A_2 + w_3\} + \max \{0, A_4 + w_4, A_2 + v_4\}, \\ \max \{0, A_1 + w_1\} + \max \{0, A_4 + w_2, A_1 + v_2\} + \max \{0, A_2 + w_3\} + \max \{0, A_3 + w_4, A_2 + v_4\}, \\ \max \{0, A_2 + w_1\} + \max \{0, A_4 + w_2, A_2 + v_2\} + \max \{0, A_1 + w_3\} + \max \{0, A_3 + w_4, A_1 + v_4\}, \\ \max \{0, A_3 + w_1\} + \max \{0, A_4 + w_2, A_2 + v_2\} + \max \{0, A_1 + w_3\} + \max \{0, A_2 + w_4, A_1 + v_4\}, \\ \max \{0, A_2 + w_1\} + \max \{0, A_3 + w_2, A_2 + v_2\} + \max \{0, A_1 + w_3\} + \max \{0, A_2 + w_4, A_1 + v_4\}, \\ \max \{0, A_2 + w_1\} + \max \{0, A_3 + w_2, A_2 + v_2\} + \max \{0, A_1 + w_3\} + \max \{0, A_4 + w_4, A_1 + v_4\}, \\ \max \{0, A_2 + w_1\} + \max \{0, A_3 + w_2, A_2 + v_2\} + \max \{0, A_1 + w_3\} + \max \{0, A_4 + w_4, A_1 + v_4\}, \\ \max \{0, A_2 + w_1\} + \max \{0, A_3 + w_2, A_2 + v_2\} + \max \{0, A_1 + w_3\} + \max \{0, A_4 + w_4, A_1 + v_4\}, \\ \max \{0, A_2 + w_1\} + \max \{0, A_3 + w_2, A_2 + v_2\} + \max \{0, A_1 + w_3\} + \max \{0, A_4 + w_4, A_1 + v_4\}, \\ \max \{0, A_2 + w_1\} + \max \{0, A_3 + w_2, A_2 + v_2\} + \max \{0, A_1 + w_3\} + \max \{0, A_4 + w_4, A_1 + v_4\}, \\ \max \{0, A_2 + w_1\} + \max \{0, A_3 + w_2, A_2 + v_2\} + \max \{0, A_1 + w_3\} + \max \{0, A_4 + w_4, A_1 + v_4\}, \\ \max \{0, A_2 + w_1\} + \max \{0, A_3 + w_2, A_2 + v_2\} + \max \{0, A_1 + w_3\} + \max \{0, A_4 + w_4, A_1 + v_4\}, \\ \max \{0, A_2 + w_1\} + \max \{0, A_3 + w_2, A_2 + v_2\} + \max \{0, A_1 + w_3\} + \max \{0, A_4 + w_4, A_1 + v_4\}, \\ \max \{0, A_2 + w_1\} + \max \{0, A_3 + w_2, A_2 + v_2\} + \max \{0, A_1 + w_3\} + \max \{0, A_4 + w_4, A_1 + v_4\}, \\ \max \{0, A_3 + w_4, A_4 + w_4\} + \max \{0, A_4 + w_4, A_4 + v_4\}, \\ \max \{0, A_4 + w_4, A_4 + w_4\}, \\ \max \{0, A_4 + w_4, A_4 + w_4\} + \max \{0, A_4 + w_4, A_4 + w_4\}, \\ \max \{0, A_4 + w_4, A_4 + w_4\}, \\ \max \{0, A_4 + w_4, A_4 + w_4\}, \\ \max \{0, A_4 + w_4, A_4 + w_4\}, \\ \max \{0, A_4 + w_4, A_4 + w_4\}, \\ \max \{0, A_4 + w_4, A_4 + w_4\}, \\ \max \{0, A_4 + w_4, A_4 + w_4\}, \\ \max \{0, A_4 + w_4, A_4 + w_4\}, \\ \max \{0, A_4 + w_4, A_4 + w_4\}, \\ \max \{0, A_4 + w_4, A_4 + w_4\}, \\ \max \{0, A_4 + w_4, A_4 + w_4\}, \\ \max \{0, A_4 + w_4, A_4 + w_4\}, \\ \max \{0, A_4 + w_4, A_4 + w_4\}, \\$$

To observe the error gaps by Algorithm RH, we have three exclusive but complete cases. Case 1. $w_1 > w_2 \ge w_3 > w_4$.

Let the initial solution be the non-increasing order of w_h 's, i.e., $\{1, 2, 3, 4\}$. Since our algorithm considers two batches each time, the first subproblem involves batches 1 and 2, and hospitals 1 and 2. By Lemma 4.1, the optimal schedule for this subproblem is $\{1 \rightarrow 1, 2 \rightarrow 2\}$. Next we fix the assignment of batch 1 with hospital 1, and relax batch 2. The second subproblem involves batch 2 and 3, and hospitals 2 and 3. Since hospitals 2 and 3 are not in the same route, and by Lemma 4.1 and the assumption, the optimal schedule is $\{2 \rightarrow 2, 3 \rightarrow 3\}$. Next we fix the assignment of batch 2 with hospital 2, and relax batch 3. The third subproblem involves batch 3 and 4, and hospitals 3 and 4. Since hospitals 3 and 4 are in the same route, and by Lemma 4.3, the optimal schedule is $\{3 \rightarrow 3, 4 \rightarrow 4\}$. The algorithm terminates here because the current solution is exactly the same as the initial solution. So the delivery schedule determined by our algorithm is $\{1 \rightarrow 1, 2 \rightarrow 2, 3 \rightarrow 3, 4 \rightarrow 4\}$. Hence, the objective function value is as follows. $T^{RH} = \max\{0, A_1 + w_1\} + \max\{0, A_2 + w_2, A_1 + v_2\} + \max\{0, A_3 + w_3\} + \max\{0, A_4 + w_4, A_3 + v_4\}$

$T^{*} = \min\{T^{*} = \max\{0, A_{1} + w_{1}\} + \max\{0, A_{2} + w_{2}, A_{1} + v_{2}\} + \max\{0, A_{3} + w_{3}\} + \max\{0, A_{4} + w_{4}, A_{3} + v_{4}\}, T^{2} = \max\{0, A_{1} + w_{1}\} + \max\{0, A_{3} + w_{2}, A_{1} + v_{2}\} + \max\{0, A_{2} + w_{3}\} + \max\{0, A_{4} + w_{4}, A_{2} + v_{4}\}, T^{3} = \max\{0, A_{1} + w_{1}\} + \max\{0, A_{4} + w_{2}, A_{1} + v_{2}\} + \max\{0, A_{2} + w_{3}\} + \max\{0, A_{3} + w_{4}, A_{2} + v_{4}\}, R^{3} = \max\{0, A_{1} + w_{1}\} + \max\{0, A_{4} + w_{2}, A_{1} + v_{2}\} + \max\{0, A_{2} + w_{3}\} + \max\{0, A_{3} + w_{4}, A_{2} + v_{4}\}, R^{3} + \max\{0, A_{1} + w_{1}\} + \max\{0, A_{4} + w_{2}, A_{1} + v_{2}\} + \max\{0, A_{2} + w_{3}\} + \max\{0, A_{3} + w_{4}, A_{2} + v_{4}\}, R^{3} + \max\{0, A_{1} + w_{1}\} + \max\{0, A_{2} + w_{2}, A_{1} + v_{2}\} + \max\{0, A_{2} + w_{3}\} + \max\{0, A_{3} + w_{4}, A_{2} + v_{4}\}, R^{3} + \max\{0, A_{1} + w_{1}\} + \max\{0, A_{2} + w_{2}, A_{1} + v_{2}\} + \max\{0, A_{2} + w_{3}\} + \max\{0, A_{3} + w_{4}, A_{2} + v_{4}\}, R^{3} + \max\{0, A_{1} + w_{1}\} + \max\{0, A_{2} + w_{2}, A_{1} + v_{2}\} + \max\{0, A_{2} + w_{3}\} + \max\{0, A_{3} + w_{4}, A_{2} + v_{4}\}, R^{3} + \max\{0, A_{3} + w_{4}, A_{3} + v_{4}\}, R^{3} + \max\{0, A_{3} + w_{4}, A_{3} + v_{4}\}, R^{3} + \max\{0, A_{3} + w_{4}, A_{3} + v_{4}\}, R^{3} + \max\{0, A_{3} + w_{4}, A_{3} + v_{4}\}, R^{3} + \max\{0, A_{3} + w_{4}, A_{3} + v_{4}\}, R^{3} + \max\{0, A_{3} + w_{4}, A_{3} + v_{4}\}, R^{3} + \max\{0, A_{3} + w_{4}, A_{3} + v_{4}\}, R^{3} + \max\{0, A_{3} + w_{4}, A_{3} + v_{4}\}, R^{3} + \max\{0, A_{3} + w_{4}, A_{3} + v_{4}\}, R^{3} + \max\{0, A_{3} + w_{4}, A_{3} + v_{4}\}, R^{3} + \max\{0, A_{3} + w_{4}, A_{4} + v_{4}\}, R^{3} + \max\{0, A_{4} + w_{4}, A_{4} + w_{4}\}, R^{3} + \max\{0, A_{4} + w_{4}, A_{4} + w_{4}\}, R^{3} + \max\{0, A_{4} + w_{4}, A_{4} + w_{4}\}, R^{3} + \max\{0, A_{4} + w_{4}, A_{4} + w_{4}\}, R^{3} + \max\{0, A_{4} + w_{4}, A_{4} + w_{4}\}, R^{3} + \max\{0, A_{4} + w_{4}, A_{4} + w_{4}\}, R^{3} + \max\{0, A_{4} + w_{4}, A_{4} + w_{4}\}, R^{3} + \max\{0, A_{4} + w_{4}, A_{4} + w_{4}\}, R^{3} + \max\{0, A_{4} + w_{4}, A_{4} + w_{4}\}, R^{3} + \max\{0, A_{4} + w_{4}, A_{4} + w_{4}\}, R^{3} + \max\{0, A_{4} + w_{4}, A_{4} + w_{4}\}, R^{3} + \max\{0, A_{4} + w_{4}, A_{4} + w_{4}\}, R^{3} + \max\{0, A_{4} + w$

optimal

objective

function

Note $T^{RH} = T^1$. By the proof of Lemma 4.1, we have $\max\{0, A_2 + w_2, A_1 + v_2\} + \max\{0, A_3 + w_3\} \le \max\{0, A_3 + w_2, A_1 + v_2\} + \max\{0, A_2 + w_3\}$, and

$$\max\{0, A_2 + w_2, A_1 + v_2\} + \max\{0, A_3 + w_3\} \le \max\{0, A_4 + w_2, A_1 + v_2\} + \max\{0, A_2 + w_3\}$$

So the gap between T^{RH} and T^2 is

this

case,

the

in

Note

$$|T^{RH} - T^{2}| \leq |\max\{0, A_{4} + w_{4}, A_{3} + v_{4}\} - \max\{0, A_{4} + w_{4}, A_{2} + v_{4}\}|$$

$$\leq A_{3} - A_{2} \leq \max_{1 \leq i \leq 3} (A_{i+1} - A_{i})$$

and the gap between T^{RH} and T^3 is

$$|T^{RH} - T^{3}| \leq |\max\{0, A_{4} + w_{4}, A_{3} + v_{4}\} - \max\{0, A_{3} + w_{4}, A_{2} + v_{4}\}|$$

$$\leq \max\{A_{3} - A_{2}, A_{4} - A_{3}\} \leq \max_{1 \leq i \leq 3}(A_{i+1} - A_{i})$$

Therefore, the gap between
$$T^{RH}$$
 and
 T^* satisfies $|T^{RH} - T^*| \le \max\{|T^{RH} - T^2|, |T^{RH} - T^3|\} \le \max_{1 \le i \le 3}(A_{i+1} - A_i)$.
Case 2. $w_1 \ge w_3 \ge w_2 \ge w_4$.

Use the same argument as in case I, the delivery schedule determined by our algorithm is $\{1, 3, 2, 4\}$. Hence, the objective function value is as follows.

$$T^{RH} = \max\{0, A_1 + w_1\} + \max\{0, A_3 + w_2, A_1 + v_2\} + \max\{0, A_2 + w_3\} + \max\{0, A_4 + w_4, A_2 + v_4\}$$

By similar approach as in case 1, we also have $|T^{RH} - T^*| \le \max_{1 \le i \le 3} (A_{i+1} - A_i)$.

Case 3. $w_1 \ge w_3 > w_4 \ge w_2$.

is

value

Use the same argument as in case I, the delivery schedule determined by our algorithm is $\{1, 3, 4, 2\}$. Hence, the objective function value is as follows. $T^{RH} = \max\{0, A_1 + w_1\} + \max\{0, A_4 + w_2, A_1 + v_2\} + \max\{0, A_2 + w_3\} + \max\{0, A_3 + w_4, A_2 + v_4\}$ By similar approach as in case 1, we also have $|T^{RH} - T^*| \le \max_{1 \le i \le 3} (A_{i+1} - A_i)$.

Thus, in all three cases, our bound is valid. This concludes the proof. \Box

5.3 Empirical study

To observe the empirical performance of the proposed *RH* approach, we randomly generated 5,420 test cases under various parameter values, which are summarized in Table 1 below.

Parameters	Range of the Parameter Value
The total number of hospitals/customers (n_{H})	[10,80]
The total number of DCs (n_K)	2, 3
The average number of batches at each DC (n_{B_k})	$\left\lceil n_{H} / n_{K} \right\rceil$
The total number of travel teams (n_M)	3, 4, 5
Hospital order size (D_h)	Uniform (20, 30), Uniform (20,
	70), Uniform (20, 100)
Hospital service time duration (p_h)	D _h /20units/hour

<i>Team release time from the base</i> (r_i)	Uniform (0, 2)
Hospital specified service completion time (d_h)	Uniform (5, 100)

Table 5.1 Parameters Used in the Empirical Study for the RH Algorithm

In the experiments, we considered two types of networks: those with 2 DCs ($n_{K} = 2$) and those with 3 DCs ($n_K = 3$). For each type of network, the network size was defined by the total number of hospitals (or customer demand points). We first generated hospitals' order quantities for non-renewable resources, and then generated the number of batches at DCs, with each batch size randomly sampled from a uniform distribution between 100 units and 150 units. For each given set of parameter values, we randomly generated 30 test cases. The total number of medical teams varied from $n_M = 3$ to $n_M = 5$, with $n_M = 4$ for most test cases. The hospitals in the network generated for each test case were then randomly assigned to the routes of travel teams. The time intervals between every two consecutive batches at a DC were randomly sampled from a uniform distribution between 6 and 12 time units. Finally, the arrival time of the first batch of non-renewable resources to a DC was randomly sampled from a uniform distribution between 0 and 2 time units. The nodes in the supply chain network (i.e., DCs, home bases of travel teams, and hospitals) were randomly scattered over an area of 2,500 square miles, proportional to the scale of 2011 Tohoku earthquake and Tsunami (see Figure 5.7 below). The estimated travel/shipping time between each pair of locations in the network is calculated by Euclidean distance divided by a speed of 25 miles per time unit.



Figure 5.7 Tōhoku earthquake and Tsunami, background map: Miyagi Prefecture, Japan (Source: martinjapan.blogspot.com/2011/03/extreme-flooding-along-coast-due-to.html).

For most test cases, the commercial GUROBI solver failed to find the optimal solution within one-hour CPU time limit (on a Dell desktop, Intel Core TM 2 Duo CPU, E8400 with 3 GB RAM). For such cases, we used the best feasible GUROBI solutions obtained within the time limit as a surrogate for the optimal solution. In contrast, the proposed search algorithm, *RH*, terminated with the best feasible solutions obtained within 1 or 2 minutes of CPU time for networks with less than 40 hospitals, and within 8 to 12 minutes for larger networks covering up to 80 hospitals. For each test case, we collected two performance measures: the required CPU time to terminate the search, and the *empirical error gap* defined as

$$Gap = \frac{|G^{RH} - G^*|}{G^*}$$

where G^* stands for the minimum total tardiness obtained using the commercial solver,

GUROBI, to solve problem **P** defined by (1) - (8), and G^{RH} stands for the total tardiness of the operation plan obtained by the proposed search algorithm *RH*.

Figures 5.8 and 5.9 present the empirical error gap distributions against the number of hospitals when $n_K = 2$, and $n_K = 3$, respectively, and $n_M = 4$, under a high level of variability in hospital demand (defined by Uniform (20, 100)). As the results show, the proposed *RH* algorithm was able to find a feasible solution (i.e., an operation plan) that was within 5% from the optimal or the surrogate of optimal solution for all the 4,200 randomly generated test cases. We can also observe that the error gap increased as the network size increased. This is because of the myopic nature of the rolling-horizon based greedy heuristic (i.e., in each iteration - we found and then fixed the optimal solution to a sub-problem, which may not however



Figure 5.8 Error Gap Distribution under High Level of Variability in Customer Demand

 $(n_{K} = 2)$



Figure 5.9 Error Gap Distribution under High Level of Variability in Customer Demand

$$(n_{K} = 3)$$

be global optimal). We also note that when the network sizes were relatively small (e.g. $n_H < 40$), the empirical error gap was fairly reasonable (e.g., within 3% cross all the test cases) as we can see from Figures 5.8 and 5.9. Similar error gap distributions were also observed when the variability levels of hospital demand on non-renewable resources were low or moderate (see Figures 5.10 and 5.11).


Figure 5.10 Average Empirical Error Gaps under Different Levels of Demand Variability

 $(n_{K} = 2)$



Figure 5.11 Average Empirical Error Gaps under Different Levels of Demand Variability

In Figure 5.12, we report the required solution time (in CPU seconds) by the proposed *RH* algorithm and that by the commercial solver GUROBI when the total number of DCs in a network equals to $n_K = 2$, and $n_K = 3$, respectively. As the results show, the required computational effort by the *RH* algorithm was significantly less than that required by GUROBI.

 $(n_{K} = 3)$



Figure 5.12 Computational Effort Required by the RH Algorithm and that by GUROBI



Figure 5.13 Average Error Gap (%) against the Number of Travel Teams

We can see that GUROBI failed to find the optimal solution within 1 CPU hour when the network size goes beyond $n_H = 60$. In contrast, the proposed RH algorithm was able to terminate with a quality feasible solution within few minutes of CPU time in all the experiments even when $n_{H} = 80$. It should be pointed out that, when the network contains $n_K = 3$ distribution centers, the proposed algorithm *RH* does take a longer time to terminate relative to the $n_K = 2$ case. This is because, when $n_K = 3$, each randomly generated test case has more batches of non-renewable resources to be scheduled during the search process. When the number of batches increases, the number of iterations in each round also increases. In Figure 5.13, we report error gap comparisons under different numbers of travel teams involved in the emergency operations. While we can observe a trend of increasing error gaps when network size goes up, we also see that, when $n_K = 2$ for the same network size (i.e., the same value of n_H), the involvement of more travel teams (which requires us to allocate non-renewable resources among more routes) tends to lead to large error gaps. This is caused by a higher probability of introducing more scheduling errors by heuristically assigning non-renewable resources among more teams. However, when supply increases or becomes more sufficient, i.e., $n_{K} = 3$, the errors are mitigated and offset, which results in a relatively balanced errors among different teams.

We also studied the potential impact of pre-positioning inventory at DCs before the arrival of a disaster. In particular, we examined the relationship between the total tardiness and the sizes of the batches pre-positioned at DCs (as a percentage of the total demand) at time zero. The amount of non-renewable resources pre-positioned at each DC at time zero can be interpreted as safety stock used to mitigate the potential risks caused

by disasters (see Tomlin (2006) and Qi (2013)), and the more received by DCs at time zero, the stronger the magnitude of inventory mitigation. To study the impact of inventory pre-positioning, we experimented with networks determined by parameters $n_H = 50$, $n_{K} = 2$ or 3, $n_{M} = 4$, and $D_{h} = Uniform(20,100)$ for all h in H. The total amount of inventory pre-positioned at time zero across all the DCs was set at 5%, 10%, 15%, 20%, 25%, 30%, 35%, 40%, 45%, 50%, 55%, and 60% of the total demand to the non-renewable resources, respectively. For each set of parameter values, we randomly generated 10 test cases, and report the resulting average total tardiness in Figure 5.14. As we can see, there is a tremendous drop in the resulting tardiness as the amount of pre-positioned inventory increases from 0% to 40% of the total customer demand to the non-renewable resources. This indicates that a sufficient inventory of non-renewable resources to be pre-positioned before the arrival of a disaster could make a fundamental difference in the resulting tardiness since the service to each hospital is now only controlled by a single resource (i.e., the travel team), instead of both resources. We also observe a faster decrease in total tardiness when $n_{\kappa} = 3$. This implies a natural trade-off between the number of DCs and the total response/transportation time.



Figure 5.14 The Impact of Pre-positioning Inventory Before the Arrival of a Disaster

Chapter 6. Future Extensions

There are two major extensions of this research. The first one is to conduct a thorough simulation study to assess the impact of management policies on the effectiveness of emergency logistics involving bottleneck renewable and non-renewable resources. The second one is to design and evaluate meta-heuristics for solving a more general version of problem P.

6.1 A Simulation Study for Assessing the Impact of Management Policies

Effective emergency logistics requires both resources and management. One important topic in this regard, from both academic research and practices point of view is the development of effective disaster relief management policies. One of such policies is the deployment of mobile distribution centers that allow truck trailers to carry the non-renewable resources and distribute the supplies among the demand points (e.g., hospitals, shelters, schools, etc.) on real time to minimize the patient waiting time. We have performed some preliminary simulation studies in this regard, which can be summarized as follows.

We extend problem P to a multi-period one (the set of period is defined as T with the total number of all periods is given as |T|). This multi-period problem is defined as follows. (i). In each period t, $t \in T$, the expected service completion time for each customer h, $h \in H$, is $d_{h,t}$. The service processing at customer h, is p_h , which is linear function of the customer's non-renewable resource demand, $Q_{h,t} \cdot Q_{h,t}$ is determined in the following way. At the end of each period, each customer h observes a forecasting demand for the next period, $Q_{h,t}^+$. The actual demand $Q_{h,t}$ is a stochastic function of $Q_{h,t-1}^+$. Note in this setting, $Q_{h,t}$ can be zero, i.e., there is no demand for some hospitals in some periods.

(ii). We have two types of DCs, regular and mobile. Each regular DC k, $k \in K$, has a fixed location, and receives at most one batch of non-renewable resources from its upstream suppliers in each period t. Each batch is defined by a batch size, $Q_{k,t}$, and a forecasting batch arrival time, $A_{k,t}^+$, $k \in K$. The actual demand $A_{k,t}$ is a stochastic function of $A_{k,t}^+$. Starting from time $A_{k,t}$, the total quantity of non-renewable resources that are available for delivery to customers equal to $Q_{k,t}$ plus those leftover from previous batches at DC k. The mobile DC k_0 has an initial location in one period, and moves to the location of the most urgent customer with smallest $d_{h,t}$ among all that have demand in the next period, at the end of the current period. The distribution time from a certain DC to a certain customer h, is given by $\overline{\tau}_{kh}$ (same value for all periods). The allocation rule from DC to customer is a greedy type heuristic: starting from the customers which has the earliest due date in the current period, for each customer h,

sequence all the DCs including the mobile one in an increasing order based on $\overline{\tau}_{kh}$. Let the nearest DC serve it, if that DC doesn't have enough batch supply, let the second nearest DC serve it, if both DCs couldn't fulfill it, let the third nearest one serve it, so on so forth. After one customer is fully served, we move to the next customer with the earliest due date. This process terminates when either all customers are served or all DCs have no stock left.

(iii). Each team $i, i \in M$, departs from its base at time 0 in each period, and visits and serves the nearest customer among all that have demand. After serving one customer, the team should visit the next customer that has demand and is nearest to its current location (the location of the incumbent). Each team continues to visit customers until the expected arrival time at a certain customer exceeds the maximal length of a period. At the end of each period, the team should come back to its base. The travel time between any two customer sites in the same team is given by $\tau_{h,h'}$ (same value for all periods).

We aim to compare the scenario where there exists a mobile DC that travels on site freely and the scenario where all DCs are regular DCs. The second scenario is served as comparison base that can be used to evaluate the potential cost savings and effectiveness of our first scenario. The numbers of DCs in the two scenarios are the same. The measure for the comparison is the total tardiness across all hospitals over all periods. Our simulation procedure is described as follows.

Simulation Procedure:

Input:

The initial locations of team bases, regular and mobile DCs, and customers were randomly generated in a 10*10 square area (adjusted by traveling time). The length of each period is 24 hours, and the total number of periods in one simulation run is |T|=100. Note $\overline{\tau}_{kh}$ and $\tau_{h,h'}$ can be computed by the distance between two corresponding locations. For example, $\tau_{h,h'} = \sqrt{(x_h - x_{h'})^2 + (y_h - y_{h'})^2}$ where x and y represent x-axis and y-axis coordinates of customers *h* and *h'*. The forecasting demands are generated by two ways. One way is a hybrid of a uniform distribution and a Poisson distribution. The other way is a uniform distribution multiplied by a triangle function centered at *t=50*. The other parameter distributions and value ranges are summarized in Table 6.1.

Parameters	Range and Distribution of the Parameter
	Value
Coordinates of bases, DCs, and	Uniform (0, 10)
customers	
The total number of customers (n_{H})	20, 30
The total number of Regular DCs (n_{K})	2, 3
Number of Mobile DC	1

The total number of travel teams (n_M)	3, 4
Forecasting demand size $(Q_{h,t}^+)$	Uniform (0, 30)* Poisson(t, 50),
	Uniform (0, 30)*(1- 1-2*t/ T)
Actual demand size $(Q_{h,t})$	$Q_{h,t-1}^+ + Normal(0,5)$
Hospital service time duration (p_h)	$Q_{h,t}/5$
Batch size (Q_{kl})	Uniform (0, 200)
<i>Forecasting batch arrival time</i> $(A_{k,t}^{+})$	Uniform (0, 12)
Actual batch arrival time $(A_{k,t})$	$A_{k,t} + Normal(0,3)$
Customer service completion time $(d_{h,t})$	Uniform (6, 18)

Table 6.1 Parameters Used in the Empirical Study for the RH Algorithm

The flow chart that illustrates this simulation process is as



follows.

Figure 6.1 A Flowchart of the Simulation Process of Evaluation of Mobile DC

In the experiments, we considered two types of networks: the smaller networks have 2 regular DCs, 1 mobile DC, 3 travel teams $(n_M=3)$, and 20 hospitals $(n_H=20)$, and the larger networks have 3 regular DCs, 1 mobile DC, 4 teams $(n_M=4)$, and 30 hospitals $(n_H=30)$. In each setting, we randomly run the simulation 100 times for each of two different customer forecasting demand distributions. Figures 2 to 5 present the simulation results based on different networks. We used the average tardiness per customer per period as the comparison measure between the scenario with mobile DC and the one with only regular DCs. We can observe that in all network settings, the scenario with mobile DC consistently provided lower level of average tardiness than that with only regular DCs. The average tardiness savings for each of the four cases are summarized in Table 6.2 below.

Cases	Tardiness Savings
Possion Distribution with $n_{K} = n_{M} = 3$	16.62%
and $n_{H} = 20$	
Possion Distribution with $n_K = n_M = 4$	14.28%
and $n_{H} = 30$	
Triangle Distribution with	18.35%
$n_K = n_M = 3$ and $n_H = 20$	
Triangle Distribution with	17.07%
$n_{K} = n_{M} = 4 \text{ and } n_{H} = 30$	

Table 6.2 The Average Tardiness Savings for Four Different Cases



Figure 6.2 Average Tardiness Comparison for Demand with Poisson Distribution with

$$n_{K} = n_{M} = 3$$
 and $n_{H} = 20$





$$n_{K} = n_{M} = 4$$
 and $n_{H} = 30$



Figure 6.4 Average Tardiness Comparison for Demand with Triangle Distribution with

Average Tardiness Comparison for Demand with Triangle Distribution Without Mobile DO |K| = |M| = 4 and |H| = 30With Mobile DC Average Tardines Simulation Run





There are other management policies can be included in such simulation study. The other two policies to be evaluated are a). the non-renewable resources inventory-positioning before the hit of an natural disaster, and b). Using selected hospitals as transshipment depots for the non-renewable resources. Managerial insights generated from such simulation studies will have great values to support the practices of disaster relief operations.

6.2. Designing Meta-heuristics for Solving General Versions of Problem P

 $n_{K} = n_{M} = 3$ and $n_{H} = 20$

Real life emergency logistics are usually more complicated than those being studied in academic literature. To make a solution approach to become a valuable tool to support the day-to-day operations in practice, meta-heuristics have a significant role in this regard. It will be interesting to design such meta-heuristics that are able to derive operations schedules for the problems beyond the one we studied in this dissertation, such as those that including last minute change in batch arrival times, cancellation of hospital orders, random travel times, deviation in travel team release times, and stochastic hospital service times, etc.

When such more general scenarios are taking into the problem definition, mathematical programming based approaches are no longer sufficient enough, and meta-heuristics are the resort. In the current literature, there is a significant lack of studies on design meta-heuristics for complex emergency logistics operations scheduling. Examples of such meta-heuristics including *hierarchical heuristics* that allocate the non-renewable resources after the operations schedule of travel teams has been developed and *cluster heuristics* that partition the customer demand points into clusters and then schedule the operations for serving each cluster to minimize the tardiness of serving the customer group. Designing such meta-heuristics is certainly another interesting extension of this dissertation research.

Bibliography

- Afshar, A., and A. Haghani. 2012. "Modeling Integrated Supply Chain Logistics in Real-time Large-scale Disaster Relief Operations."Socio-Economic Planning Sciences 46 (4): 327–338.
- [2]. Ahuja, R.K., Huang, W., Romeijn, H.E. and Morales, D.R. (2007) A heuristic approach to the multi-period single-sourcing problem with production and inventory capacities and perishability constraints. INFORMS Journal on Computing 19(1) 14–26.
- [3]. Ahuja, R.K, Magnanti, T.L. and Orlin, J.B. (1993) Network Flows: Theory, Algorithms, and Applications, Prentice Hall, Englewood Cliffs, N.J.
- [4]. Ahuja, R. K., K. Mehlhorn, J. Orlin, and R. E. Tarjan. 1990. "Faster Algorithms for the Shortest Path Problem." Journal of the ACM 37 (2): 213–223.
- [5]. Ait-Kadia, D., J.-B. Menye, and H. Kane. 2011. "Resources Assignment Model in Maintenance Activities Scheduling." International Journal of Production Research 49 (22): 6677–6689.
- [6]. Aliev, R.A., Fazlollahi, B., Guirimov, B.G. and Aliev, R.R. (2007) Fuzzy-genetic approach to aggregate production–distribution planning in supply chain management. Information Sciences 177(20) 4241–4255.
- [7]. Alfieri, A., and P. Brandimarte. 1997. "Job Shop Scheduling with Delay-Renewable Resources: A Comparison of Decomposition Strategies." International Journal of Production Research 35 (7): 1807–1823.
- [8]. Archetti, C., Bertazzi, L., Paletta, G. and Speranza, M.G. (2011) Analysis of the maximum level policy in a production-distribution system. Computers and Operations Research 38(12) 1731-1746.
- [9]. Armstrong, R., Gao, S. and Lei, L. (2008) A zero-inventory production and distribution problem with a fixed customer sequence. Annals of Operations Research 159(1) 395–414.
- [10]. Assaf, S. A., and S. Al-Hejji. 2006. "Causes of Delay in Large Construction Projects." International Journal of Project Management 24 (4): 349–357.
- [11]. Bachlaus, M., M. K. Tiwarib, and F. T. S. Chan. 2009. "Multi-objective Resource Assignment Problem in a Product-Driven Supply Chain Using a Taguchi-based DNA Algorithm." International Journal of Production Research 47 (9): 2345–2371.
- [12]. Balcik, B., and B. M. Beamon. 2008. "Facility Location in Humanitarian Relief."

International Journal of Logistics: Research and Applications 11 (2): 101–121.

- [13]. Balcik, B., B. M. Beamon, and K. Smilowitz. 2008. "Last Mile Distribution in Humanitarian Relief." Journal of Intelligent Transportation Systems 12 (2): 51–63.
- [14]. Balcik, B., B. M. Beamon, C. C. Krejci, K. M. Muramatsu, and M. Ramirez. 2010. "Coordination in Humanitarian Relief Chains: Practices, Challenges and Opportunities." International Journal of Production Economics 126 (1): 22–34.
- [15]. Ballestin, F., and N. Trautmann. 2008. "An Iterated-local-search Heuristic for the Resource-constrained Weighted Earliness-Tardiness Project Scheduling Problem." International Journal of Production Research 46 (22): 6231–6249.
- [16]. Bard, J.F. and Nananukul, N. (2009a) The integrated production-inventory-distribution -routing problem. Journal of Scheduling 12(3) 257–280.
- [17]. Bard, J.F. and Nananukul, N. (2009b) Heuristics for a multiperiod inventory routing problem with production decisions. Computers and Industrial Engineering 57(3) 713–723.
- [18]. Bard, J.F. and Nananukul, N. (2010) A branch-and-price algorithm for an integrated production and inventory routing problem. Computers and Industrial Engineering 37(12) 2202–2217.
- [19]. Beasley, J.E. (1983) Route-first cluster-second methods for vehicle routing. Omega 11(4) 403–408.
- [20]. Beraldi, P., Ghiani, G., Grieco, A., and Guerriero, E., 2008. Rolling-horizon and fix-and-relax heuristics for the parallel machine lot-sizing and scheduling problem with sequence-dependent set-up costs. Computers & Operations Research, 35(11), 3644-3656.
- [21]. Berman, O. and Wang, Q. (2006) Inbound logistic planning: minimizing transportation and inventory cost. Transportation Science 40(3) 287–299.
- [22]. Bilgen, B. and Günther, H.-O. (2010) Integrated production and distribution planning in the fast moving consumer goods industry: a block planning application. OR Spectrum 32(4) 927–955.
- [23]. Bolduc, M.-C., Laporte, G., Renaud, J. and Boctor, J.J. (2010) A tabu search heuristic for the split delivery vehicle routing problem with production and demand calendars. European Journal of Operational Research 202(1) 122–130.
- [24]. Bottcher, J., A. Drexl, R. Kolisch, and F. Salewski. 1999. "Project Scheduling under Partially Renewable Resource Constraints." Management Science 45 (4): 543–559.
- [25]. Boudia M., Louly, M.A.O. and Prins, C. (2005) Combined optimization of production and distribution. In: International Conference on Industrial Engineering and Systems Management (IESM'05, Marrakech, Morocco) Mons, Belgium, Extended version to appear

in Production Planning and Control.

- [26]. Boudia, M., Louly, M.A.O. and Prins, C. (2007) A reactive GRASP and path relinking for a combined production-distribution problem. Computers and Operations Research 34(11) 3402–3419.
- [27]. Boudia, M. and Prins, C. (2009) A memetic algorithm with dynamic population management for an integrated production-distribution problem. European Journal of Operational Research 195(3) 703-715.
- [28]. Brenner, U. 2008. "A Faster Polynomial Algorithm for the Unbalanced Hitchcock Transportation Problem." Operations Research Letters 36 (4): 408–413.
- [29]. Brucker, P., and A. Kramer. 1996. "Polynomial Algorithms for Resource-constrained and Multiprocessor Task Scheduling Problems."European Journal of Operational Research 90 (2): 214–226.
- [30]. Brucker, P., A. Drexl, R. Möhring, K. Neumann, and E. Pesch. 1999.
 "Resource-Constrained Project Scheduling: Notation, Classification, Models, and Methods." European Journal of Operational Research 112 (1): 3–41.
- [31]. Campbell, A. M., D. Vandenbussche, and W. Hermann. 2008. "Routing for Relief Efforts." Transportation Science 42 (2): 127–145.
- [32]. Can, A., and G. Ulusoy. 2010. Multi-Project Scheduling with 2-Stage Decomposition. Technical Report. Sabanci University. ID: SU_FENS_2011/0010.
- [33]. Caunhye, A., X. Nie, and S. Pokharel. 2012. "Optimization Models in Emergency Logistics: A Literature Review." Socio-Economic Planning Sciences 46 (1): 4–13.
- [34]. Celano, G., A. Costa, and S. Fichera. 2008. "Scheduling of Unrelated Parallel Manufacturing Cells with Limited Human Resources." International Journal of Production Research 46 (2): 405–427.
- [35]. Cetinkaya, S., Uster, H., Easwaran, G. and Keskin, B.B. (2009) An integrated outbound logistics model for Frito-Lay: Coordinating aggregate-level production and distribution decisions. Interfaces 39(5) 460–475.
- [36]. Chan, F. T. S., T. C. Wong, and L. Y. Chan. 2006. "Flexible Job-shop Scheduling Problem under Resource Constraints." International Journal of Production Research 44 (11): 2071–2089.
- [37]. Chandra, P. (1989) On coordination of production and distribution decisions. Ph.D. Dissertation, Department of Decision Sciences, The Wharton School, University of Pennsylvania, Philadelphia, PA.
- [38]. Chandra, P. and Fisher, M.L. (1994) Coordination of production and distribution planning.

European Journal of Operational Research 72(3) 503-517.

- [39]. Changfeng, Y., Z. Dinghua, P. Wenli, and B. Kun. 2006. "Research on Resources Optimization Deployment Model and Algorithm for Collaborative Manufacturing Process." International Journal of Production Research 44 (16): 3279–3301.
- [40]. Chen, C.L. and Lee, W.C. (2004) Multi-objective optimization of multi-echelon supply chain networks with uncertain product demands and prices. Computers and Chemical Engineering 28(6-7) 1131–1144.
- [41]. Chen, Z.L. (2010) Integrated production and outbound distribution scheduling: Review and extensions. Operations Research 58(1) 130–148.
- [42]. Chen, W.-H. and Thizy J.-M. (1990) Analysis of relaxations for the multi-item capacitated lot-sizing problem. Annals of Operations Research 26(1) 29–72.
- [43]. Chopra, S. and Meindl, P. (2001) Supply chain management: strategy, planning, and operations. Prentice Hall, Upper Saddle River, NJ.
- [44]. Christofides, N., Mingozzi, A. and Toth, P. (1979) The vehicle routing problem. In Christofides, N., Mingozzi, A., Toth, P. and Sandi, C. (eds.) Combinatorial Optimization. Wiley, Chichester.
- [45]. de Araujo, S., Arenales, M., and Clark, A., 2007. Joint rolling-horizon scheduling of materials processing and lot-sizing with sequence-dependent setups. Journal of Heuristics, 13 (4), 337-358.
- [46]. Debels, D., and M. Vanhoucke. 2007. "A Decomposition-based Genetic Algorithm for the Resource-Constrained Project-Scheduling Problem." Operations Research 55 (3): 457–469.
- [47]. Deblaere, F., E. Demeulemeester, W. Herroelen, and S. Van de Vonder. 2007. "Robust Resource Allocation Decisions in Resource-Constrained Projects." Decision Sciences 38 (1): 5–37.
- [48]. Deblaere, F., E. Demeulemeester, and W. Herroelen. 2011a. "Reactive Scheduling in the Multi-Mode RCPSP." Computers and Operations Research 38 (1): 63–74.
- [49]. Deblaere, F., E. Demeulemeester, and W. Herroelen. 2011b. "Proactive Policies for the Stochastic Resource-Constrained Project Scheduling Problem." European Journal of Operational Research 214 (2): 308–316.
- [50]. Demirkol, E.and Uzsoy, R., 2000. Decomposition methods for reentrant flow shops with sequence-dependent setup times. Journal of Scheduling, 3 (3), 155-177.
- [51]. Depuy, G. W., and G. E. Whitehouse. 2001. "A Simple and Effective Heuristic for the Resource Constrained Project-Scheduling Problem." International Journal of Production Research 39 (14): 3275–3287.

- [52]. Dhaenens-Flipo, C. and Finke, G. (2001) An integrated model for an industrial production distribution problem. IIE Transactions 33(9) 705–715.
- [53]. Dogan, K and Goetschalckx, M. (1999) A primal decomposition method for the integrated design of multi-period production-distribution systems. IIE Transactions 31(11) 1027–1036.
- [54]. Eckstein, A. L. H., and T. R. Rohleder. 1998. "Incorporating Human Resources in Group Technology/Cellular Manufacturing." International Journal of Production Research 36 (5): 1199–1222.
- [55]. Eksioglu, S.D., Eksioglu, B. and Romeijn, H.E. (2007) A Lagrangean heuristic for integrated production and transportation planning problems in a dynamic, multi-item two-layer supply chain. IIE Transactions 39(2) 191–201.
- [56]. Erenguc, S.S., Simpson, N.C. and Vakharia, A.J. (1999) Integrated production /distribution planning in supply chains: An invited review. European Journal of Operational Research 115(2) 219–236.
- [57]. Fahimnia, B., Luong, L. and Marian, R. (2008a) An integrated model for the optimisation of a two-echelon supply network. Journal of Achievements in Materials and Manufacturing Engineering 31(2) 477–484.
- [58]. Fahimnia, B., Luong, L. and Marian, R. (2008b) Optimization/simulation modeling of the integrated production-distribution plan: an innovative survey. WSEAS Transactions on Business and Economics 5(3) 52–65.
- [59]. Fahimnia, B., Farahani, R.Z., Marian, R. and Luong, L. (2013) A review and critique on integrated production-distribution models and techniques. Journal of Manufacturing Systems 32(1) 1–19.
- [60]. Fenton, G. 2003. "Coordination in the Great Lakes." Forced Migration Review, September, 23–24.
- [61]. Fumero, F. and Vercellis, C. (1997) Integrating distribution, machine assignment and lot-sizing via Lagrangean Relaxation. International Journal of Production Economics 49(1) 45–54.
- [62]. Fumero, F. and Vercellis, C. (1999) Synchronized development of production, inventory, and distribution schedules. Transportation Science 33(3) 330–340.
- [63]. Gagnon, M., G. D'Avignona, and F. Boctor. 2009. An Algorithm for the MRCPS Problem with Renewal and Non-Renewal Resource Types. Quebec: Faculty of Business Administration, Laval University.
- [64]. Gebennini, E., Gamberini, R. and Manzini, R. (2009) An integrated

production–distribution model for the dynamic location and allocation problem with safety stock optimization. International Journal of Production Economics 122(1) 286–304.

- [65]. Geismar, H.N., Laporte, G., Lei, L. and Sriskandarajah, C. (2008) The integrated production and transportation scheduling problem for a product with a short lifespan. INFORMS Journal on Computing 20(1) 21–33.
- [66]. Gen, M.S. and Syarif, A. (2005) Hybrid genetic algorithm for multi-time period production/distribution planning. Computers and Industrial Engineering 48(4) 799–809.
- [67]. Gillet, B. and Miller, L. (1974) A heuristic algorithm for the vehicle dispatch problem. Operations Research 22(2) 340–349.
- [68]. Gilmore, P. and Gomory, R. (1964) Sequencing a one-state variable machine: A solvable case of the traveling salesman problem. Operations Research 12(5) 675–679.
- [69]. Grötschel, M. and Padberg, M.W. (1979) On the symmetric travelling salesman problem I: Inequalities. Mathematical Programming 16(1) 265-280.
- [70]. Haghani, A., and S. C. Oh. 1996. "Formulation and Solution of a Multi-commodity, Multi-modal Network Flow Model for Disaster Relief Operations." Transportation Research Part A: Policy and Practice 30 (3): 231–250.
- [71]. Hartmann, S., and D. Briskorn. 2010. "A Survey of Variants and Extensions of the Resource-Constrained Project Scheduling Problem." European Journal of Operational Research 207 (1): 1–14.
- [72]. Huang, M., K. Smilowitz, and B. Balcik. 2012. "Models for Relief Routing: Equity, Efficiency and Efficacy." Transportation Research Part E 48 (1): 2–18.
- [73]. Hwang, N.-C., and K. Kogan. 2003. "Dynamic Approach to Human Resources Planning for Major Professional Companies with a Peak-Wise Demand." International Journal of Production Research 41 (6): 1255–1271.
- [74]. Inman, R. and Jones, P., 1993. Decomposition for Scheduling Flexible Manufacturing Systems. Operations Research, 41 (3), 608-617.
- [75]. Jang, Y.-J., Jang, S.-Y., Chang, B.-M. and Park, J. (2002) A combined model of network design and production/distribution planning for a supply network. Computers and Industrial Engineering 43(1-2) 263–281.
- [76]. Kannan, G., Sasikumar, P. and Devika, K. (2010) A genetic algorithm approach for solving a closed loop supply chain model: A case of battery recycling. Applied Mathematical Modeling 34(3) 655–670.
- [77]. Karakitsiou, A. and Migdalas, A. (2008) A decentralized coordination mechanism for integrated production-transportation-inventory problem in the supply chain using

Lagrangian relaxation. Operational Research 8(3) 257-278.

- [78]. Karsu, O., and M. Azizoglu. 2012. "The Multi-Resource Agent Bottleneck Generalized Assignment Problem." International Journal of Production Research 50 (2): 309–324.
- [79]. Klein, R. 2000. "Project Scheduling with Time-varying Resource Constraints." International Journal of Production Research 38 (16): 3937–3952.
- [80]. Lee, C.-Y., and L. Lei. 2001. "Multiple-Project Scheduling with Controllable Project Duration and Hard Resource Constraint: Some Solvable Cases." Annals of Operations Research 102 (1–4): 287–307.
- [81]. Lee, K., Lei, L., Pinedo, M. and Wang, S., 2013. (2013): Operations scheduling with multiple resources and transportation considerations, International Journal of Production Research, DOI: 10.1080/00207543.2013.781283
- [82]. Lee, Y.H., Kim, S.H. and Moon, C. (2002) Production-distribution planning in supply chain using a hybrid approach. Production Planning and Control 13(1) 35–46.
- [83]. Lee, Y.H. and Kim, S.H. (2002) Production-distribution planning in supply chain considering capacity constraints. Computers and Industrial Engineering 43(1-2) 169–190.
- [84]. Lee, Y. H., and M. Pinedo. 1997. "Scheduling Jobs on Parallel Machines with Sequence Dependent Setup Times." European Journal of Operational Research 100 (3): 464–474.
- [85]. Lei, L., Liu, S., Ruszczynski, A. and Park, S. (2006) On the integrated production, inventory, and distribution routing problem. IIE Transactions 38(11) 955–970.
- [86]. Lei, L., Zhong, H. and Chaovalitwongse, W.A. (2009) On the integrated production and distribution problem with bidirectional flows. INFORMS Journal on Computing 21(4) 585–598.
- [87]. Lejeune, M.A. and A. Ruszczynski. (2007) An efficient trajectory method for probabilistic production–inventory–distribution problems. Operations Research 55(2) 378–394.
- [88]. Leus, R., and W. Herroelen. 2004. "Stability and Resource Allocation in Project Planning." IIE Transactions 36 (7): 667–682.
- [89]. Li, Z. and Ierapetritou, M., 2010. Rolling horizon based planning and scheduling integration with production capacity consideration. Chemical Engineering Science, 65 (22), 5887-5900.
- [90]. Liang, T.F. and Cheng, H.W. (2009) Application of fuzzy sets to manufacturing /distribution planning decisions with multi-product and multi-time period in supply chains. Expert Systems with Applications 36(2) 3367–3377.
- [91]. Liu, S., Lei, L. and Park, S. (2008) On the multi-product packing-delivery problem with fixed route. Transportation Research Part E-Logistics and Transportation Review 44(3)

350-360.

- [92]. Mausser, H. and Lawrence, S., 1997. Exploiting block structure to improve resource-constrained project schedules. Glover, F., Osman, I., and Kelley, J., eds. Metaheuristics: State of the Art. Kluwer.
- [93]. Mazzola, J., and A. Neebe. 1986. "Resource-Constrained Assignment Scheduling." Operations Research 34 (4): 560–572.
- [94]. Mingozzi, A., M. A. Boschetti, S. Ricciardelli, and L. Bianco. 1999. "A Set Partitioning Approach to the Crew-Scheduling Problem." Operations Research 47 (6): 873–888.
- [95]. Mokhtari, H., I. N. K. Abadia, and A. Cheraghalikhani. 2011. "A Multi-Objective Flow Shop Scheduling with Resource-Dependent Processing Times: Trade-Off between Makespan and Cost of Resources." International Journal of Production Research 49 (19): 5851–5875.
- [96]. Moon, C., J. S. Kim, and M. Gen. 2004. "Advanced Planning and Scheduling Based on Precedence and Resource Constraints for E-Plant Chains." International Journal of Production Research 42 (15): 2941–2955.
- [97]. Mula, J., Peidro, D., Diaz-Madronero, M. and Vicens, E. (2010) Mathematical programming models for supply chain production and transport planning. European Journal of Operational Research 204(3) 377–390.
- [98]. Munkres, J. 1957. "Algorithms for the Assignment and Transportation Problems." Journal of the Society of Industrial and Applied Mathematics 5 (1): 32–38.
- [99]. Naso, D., Surico, M., Turchiano, B. and Kaymak, U. (2007) Genetic algorithms for supply-chain scheduling: A case study in the distribution of ready-mixed concrete. European Journal of Operational Research 177(3) 2069–2099.
- [100]. Nolz, P. C., F. Semet, and K. F. Doerner. 2011. "Risk Approaches for Delivering Disaster Relief Supplies." OR Spectrum 33 (3): 543–569.
- [101]. Nudtasomboon, N., and S. Randhawa. 1997. "Resource-Constrained Project Scheduling with Renewable and Non-Renewable Resources and Time-Resource Tradeoffs." Computers & Industrial Engineering 32 (1): 227–242.
- [102]. Orlin, J., 1993. A Faster Strongly Polynomial Minimum Cost Flow Algorithm. Operations Research,41 (2), 338-350.
- [103]. Onwubolu, G. C., and M. Mutingi. 2001. "Optimizing the Multiple Constrained Resources Product Mix Problem Using Genetic Algorithms." International Journal of Production Research 39 (9): 1897–1910.
- [104]. Ozdamar, L., E. Ekinci, and B. Kucukyazici. 2004. "Emergency Logistics Planning in

Natural Disasters." Annals of Operations Research 129 (1-4): 217-245.

- [105]. Palpant, M., Artigues, C. and Michelon, P., 2004. LSSPER: Solving the resource-constrained project scheduling problem with large neighborhood search. Annals of Operations Research, 131 (1-4), 237-257.
- [106]. Park, Y.B. (2005) An integrated approach for production and distribution planning in supply chain management. International Journal of Production Research 43(6) 1205–1224.
- [107]. Rajkumar, M., P. Asokan, N. Anilkumar, and T. Page. 2011. "A GRASP Algorithm for Flexible Job-Shop Scheduling Problem with Limited Resource Constraints." International Journal of Production Research 49 (8): 2409–2423.
- [108]. Ranjbar, M., B. D. Reyck, and F. Kianfar. 2009. "A Hybrid Scatter Search for the Discrete Time/Resource Trade-Off Problem in Project Scheduling." European Journal of Operational Research 193 (1): 35–48.
- [109]. Rizk, N., Martel, A. and D'Amours, S. (2006) Multi-item dynamic production –distribution planning in process industries with divergent finishing stages. Computers and Operations Research 33(12) 3600–3623.
- [110]. Robinson, K. R., and S. A. Moses. 2006. "Effect of Granularity of Resource Availability on the Accuracy of Due Date Assignment." International Journal of Production Research 44 (24): 5391–5414.
- [111]. Rosencrantz, D.J., Steams, R.E. and Lewis, P.M. (1974) Approximate algorithms for the TSP. Proceedings 15th IEEE Symposium on Switching and Automata Theory 33–42.
- [112]. Ruokokoski, M., Solyali, O., Cordeau, J.-F., Jans, R. and Sural, H. (2010) Efficient formulations and a branch-and-cut algorithm for a production-routing problem. GERAD Tech. Rep. G-2010-66, HEC Montreal, Canada.
- [113]. Sabri, E.H. and Beamon, B.M. (2000) A multi-objective approach to simultaneous strategic and operational planning in supply chain design. Omega 28(5) 581–598.
- [114]. Safaei, A.S., Husseini, S.M.M., Farahani, R.Z., Jolai, F. and Ghodsypour, S.H. (2010) Integrated multi-site production-distribution planning in supply chain by hybrid modeling. International Journal of Production Research 48(14) 4043–4069.
- [115]. Sarmiento, A. M. and Nagi, R. (1999) A review of integrated analysis of production distribution systems. IIE Transactions 31(11) 1061–1074.
- [116]. Schirmer, A. 2001. "Resource-Constrained Project Scheduling: An Evaluation of Adaptive Control Schemes for Parameterized Sampling Heuristics." International Journal of Production Research 39 (7): 1343–1365.
- [117]. Shen, Z.M. and Qi, L. (2007) Incorporating inventory and routing costs in strategic

Location models. European Journal of Operational Research 179(2) 372-389.

- [118]. Sheu, J.-B. 2007. "A Coordinated Reverse Logistics System for Regional Management of Multi-Source Hazardous Wastes." Computers & Operations Research 34 (5): 1442–1462.
- [119]. Sheu, C., L. J. Krajewski, and G. Cai. 2012. "A Heuristic for Designing Manufacturing Focus Units with Resource Considerations." International Journal of Production Research 50 (6): 1608–1623.
- [120]. Simons, J. V., Jr., M. D. Stephens, and W. P. Simpson, III. 1999. "Simultaneous versus Sequential Scheduling of Multiple Resources Which Constrain System Throughput." International Journal of Production Research 37 (1): 21–33.
- [121]. Sonia and Puri, M., 2004. Two Level Hierarchical Time Minimizing Transportation Problem. Sociedad de Estadística e InvestigaciónOperativa, 12 (2), 301-330.
- [122]. Sprecher, A., 2002. Network decomposition techniques for resource constrained project scheduling. Journal of the Operational Research Society, 53 (4), 405-414.
- [123]. Thomas, D.J. and Griffin, P.M. (1996) Coordinated supply chain management. European Journal of Operational Research 94(1) 1–15.
- [124]. Tormos, P., and A. Lova. 2003. "An Efficient Multi-Pass Heuristic for Project Scheduling with Constrained Resources." International Journal of Production Research 41 (5): 1071–1086.
- [125]. Van de Vonder, S., E. Demeulemeester, W. Herroelen, and R. Leus. 2006. "The Trade-Off between Stability and Makespan in Resource-Constrained Project Scheduling." International Journal of Production Research 44 (2): 215–236.
- [126]. Vanhoucke, M., E. Demeulemeester, and W. Herroelen. 2001a. "An Exact Procedure for the Resource-Constrained Weighted Earliness–Tardiness Project Scheduling Problem." Annals of Operations Research 102 (1–4): 179–196.
- [127]. Vanhoucke, M., E. Demeulemeester, and W. Herroelen. 2001b. "On Maximizing the Net Present Value of a Project under Renewable Resource Constraints." Management Science 47 (8): 1113–1121.
- [128]. Wang, H.Y., Liu, D.-C., Xing, T. and Zheng, L. (2010) A dynamic model for serial supply chain with periodic delivery policy. International Journal of Production Research 48(3) 821–834.
- [129]. Wong, C. S., F. T. S. Chan, and S. H. Chung. 2012. "A Joint Production Scheduling Approach Considering Multiple Resources and Preventive Maintenance Tasks." International Journal of Production Research 51 (3): 883–896.
- [130]. Wright, P., M. Liberatore, and R. Nydick. 2006. "A Survey of Operations Research Models

and Applications in Homeland Security." Interfaces 36 (6): 514-529.

- [131]. Yi, W., and A. Kumar. 2007. "Ant Colony Optimization for Disaster Relief Operations." Transportation Research Part E: Logistics and Transportation Review 43 (6): 660–672.
- [132]. Yilmaz, P. and Catay, B. (2006) Strategic level three-stage production distribution planning with capacity expansion. Computers and Industrial Engineering 51(4) 609–620.
- [133]. Yimer, A.D. and Demirli, K. (2010) A genetic approach to two-phase optimization of dynamic supply chain scheduling. Computers and Industrial Engineering 58(3) 411–422.
- [134]. Yossiri, A., Cordeau, J.-F. and Jans, R. (2012) Optimization-based adaptive large neighborhood search for the production routing problem. Transportation Science DOI:10.1287/trsc.1120.0443.
- [135]. Yuan, Y., and D. Wang. 2009. "Path Selection Model and Algorithm for Emergency Logistics Management." Computers & Industrial Engineering 56 (3): 1081–1094.
- [136]. Yung, K.-L., Tang, J., Ip, A.W.H. and Wang, D. (2006) Heuristics for joint decisions in production, transportation, and order quantity. Transportation Science 40(1) 99–116.

Appendix I. Matlab MIP Code for Problem P

function Y= bigurobi(H,K,M,Nb,Q_h,n_i,p_h,Q_kj,A_kj,d_h,tau_kh,tau_h)

var z,q,s,T

nz=K*Nb*H;

```
A1= zeros(H, 2*nz+2*H);
```

b1=Q_h';

for h=1:H

X = zeros(1,H); X(h)=1;

A1(h,:)= [zeros(1,nz),repmat(X,1,Nb*K),zeros(1,2*H)];

end

```
A2= zeros(Nb*K,2*nz+2*H);
```

```
b2=reshape(Q_kj,K*Nb,1);
```

for j = 1:Nb

for k=1:K

```
\begin{split} X &= zeros(1,nz); \\ X((H*Nb*(k-1)+(j-1)*H+1):(H*Nb*(k-1)+j*H)) &= ones(1,H); \\ A2((j-1)*K+k,:) &= [zeros(1,nz),X,zeros(1,2*H)]; \end{split}
```

end

end

```
A3= zeros(nz,2*nz+2*H);
```

b3= zeros(nz,1);

for k=1:K

for j= 1:Nb

for h=1:H

```
m=(k-1)*Nb*H+(j-1)*H+h;
```

A3(m,m)= -Q_kj(k,j);

A3(m,m+nz)= 1;

end

end

end

```
A4= zeros(nz,2*nz+2*H);
```

```
b4= zeros(nz,1);
```

for k=1:K

```
for j=1:Nb
```

```
for h= 1:H
```

```
m=(k-1)*Nb*H+(j-1)*H+h;
```

```
A4(m,m)=A_kj(k,j)+tau_kh(k,h);
```

```
A4(m,2*nz+h)= -1;
```

end

end

end

```
A5= zeros(H, 2*nz+2*H);
```

```
b5= zeros(H,1);
```

if H>1

```
for h=1:(H-1)
```

```
A5(h,(2*nz+h):(2*nz+h+1))=[1,-1];
```

```
b5(h) = -p_h(h)-tau_h(h+1);
```

end

end

if M>1

```
for m=1:(M-1)

A5(n_i(m),:)= zeros(1,2*nz+2*H);

A5(n_i(m),2*nz+n_i(m)+1)=-1;

b5(n_i(m))=-tau_h(n_i(m)+1);
```

end

end

A5(H,2*nz+1)=-1; % tau_01<= s_1

b5(H)= -tau_h(1);

A6= zeros(H, 2*nz+2*H);

```
b6=(p_h-d_h)';
```

for h=1:H

A6(h,2*nz+h) = -1;

A6(h,2*nz+H+h)=1;

end

```
A7= [zeros(nz),diag(ones(1,nz)),zeros(nz,2*H)];
```

```
A8= [zeros(H,2*nz),diag(ones(1,H)),zeros(H)];
```

```
A9= [zeros(H,2*nz+H),diag(ones(1,H))];
```

```
b7 = zeros(nz+2*H,1);
```

try

```
clear model;
```

```
model.A= sparse([A1;A2;A3;A4;A5;A6;A7;A8;A9]);
```

```
model.obj = [zeros(1,2*nz+H),ones(1,H)];
```

```
model.rhs= [b1;b2;b3;b4;b5;b6;b7];
```

```
model.sense= [repmat('=',1,H),repmat('<',1,Nb*K+2*nz+H),repmat('>',1,3*H+nz)];
```

```
model.vtype= [repmat('B',1,nz),repmat('C',1,nz+2*H)];
```

model.modelsense= 'min';

clear params;

params.outputflag = 0;

result = gurobi(model, params);

%disp(result.objval)

%disp(result.x)

catch gurobiError

fprintf('Error reported\n');

end

Y= [result.objval,result.x'];

End

Appendix II. Matlab RH Code for Problem P

tic;

Z= zeros(51,H0);

```
Y= cogurobi(H0,K0,M0,Nb0,Q_h0,n_i0,p_h0,Q_kj0,A_kj0,d_h0,tau_kh0,tau_h0);
```

Z(1,:)= Y((2*K0*Nb0*H0+2):(2*K0*Nb0*H0+H0+1));

 $%Z(1,:)=d_h0;$

T = zeros(1,50);

for N= 1:50

H= H0; K= K0; M= M0; Nb= Nb0;

 $Q_h = Q_h0; n_i = n_i0; p_h = p_h0;$

 $Q_kj=Q_kj0; A_kj=A_kj0; d_h=d_h0;$

tau_kh= tau_kh0; tau_h= tau_h0;

A_kj1=A_kj(:,1:2);

 $Q_kj1 = Q_kj(:,1:2);$

```
K1=K; Nb1=2;
```

n=1;

 $Z_h = Z(N,:);$

while H>1

 $[Z_hs,index] = sort(Z_h);$

x=0; i=1;

while x< sum(sum(Q_kj1)) && i<=length(index)

x=x+Q h(index(i));

i=i+1;

end

H1=i-2; M1=M;

```
if i>length(index) && x< sum(sum(Q_kj1))
```

H1=i-1;

end

```
X=1:H; index 1= X(Z_h <= Z_hs(H1));
```

```
Q_h1 = Q_h(index1);
```

```
n_i1= zeros(1,M);
```

for i=1:M

 $n_i(i) = sum(index 1 \le n_i(i));$

end

```
p_h1= p_h(index1);
```

 $d_h1 = d_h(index1);$

tau_kh1= tau_kh(:,index1);

```
tau_h1= tau_h(index1);
```

nz1=K1*Nb1*H1;

n_i10= n_i1;

```
if length(unique(n_i10)) < length(n_i10)
```

```
M1= M1-(length(n_i10)-length(unique(n_i10)));
```

n_i10= unique(n_i10);

end

 $if n_i 0(1) = 0$

M1=M1-1;

n_i10= n_i10(2:end);

end

Y= bigurobi(H1,K1,M1,Nb1,Q_h1,n_i10,p_h1,Q_kj1,A_kj1,d_h1,tau_kh1,tau_h1); z_kjh1=Y(2:(nz1+1));

```
q_kjh1=Y((nz1+2):(2*nz1+1));
```

$$s_h1 = Y((2*nz1+2):(2*nz1+H1+1));$$

$$T_h1 = Y((2*nz1+H1+2):end);$$

Q_kj1=Q_kj(:,(n+1):(n+2));

A_kj1=A_kj(:,(n+1):(n+2));

for h=1:H1

for k=1:K1

$$Q_h1(h) = Q_h1(h)-q_kjh1((k-1)*Nb1*H1+h);$$

end

end

index2= sort([index1(Q_h1>0.001),X(Z_h>Z_hs(H1))]);

H= length(index2);

if H~=0

 $Q_h(index1) = Q_h1;$

 $Q_h = Q_h(index2);$

index1= index1(Q_h1<0.001);

d_h=d_h(index2);

 $Z_h = Z_h(index2);$

tau_kh= tau_kh(:,index2);

tau_h2= tau_h(index2);

 $n_i2=zeros(1,M);$

for i= 1:M

$$n_i2(i) = sum(index2 \le n_i(i));$$

end

Z1=Z(N+1,Z(N+1,:)==0);

if length(unique(n_i2)) == length(n_i2) & n_i2(1) \sim = 0

```
if sum(index1<=n_i(1))~= 0 && index1(1)==1
```

 $tau_h2(1)= tau_h2(1)+s_h1(max(index1(index1<index2(1))))+p_h1(max(index1(index1<index2(1))));$

 $Z1(index1(index1<index2(1))) = s_h1(index1(index1<index2(1)));$

end

if sum(index1<= $n_i(1)$)~= 0 && max(index1>index2(1) & index1<= $n_i(1)$)==1

index3= index1(index1>index2(1) & index1<=n_i(1));

index4= index2(index2<= $n_i(1)$);

Z1(index3) = -1;

for ii= 1: (length(index4)-1)

if index4(ii+1)-index4(ii)>1

 $tau_h2(ii+1) = tau_h2(ii+1) + sum(tau_h(index3(index3<index4(ii+1)) & index3>index4(ii))) + sum(p_h(index3(index3<index4(ii+1))) & index3>index4(ii))));$

end

end

end

if M>1

for i= 1:(M-1)

if $sum(index1 \le n_i(i+1)) \ge sum(index1 \le n_i(i))$

if $max(index1(index1<index2(n_i2(i)+1))>n_i(i))==1$

 $\begin{array}{c} tau_h2(n_i2(i)+1) = \\ tau_h2(n_i2(i)+1) + s_h1(max(index1(index1 < index2(n_i2(i)+1))) - (n_i(i) - n_i1(i))) + p_h1(max(index1(index2(n_i2(i)+1))) - (n_i(i) - n_i1(i))); \end{array}$

 $Z1(index1(index1<index2(n_i2(i)+1) & index1>n_i(i))) = s_h1(index1(index1(index2(n_i2(i)+1)) & index1>n_i(i))(n_i(i)-n_i1(i)));$

end

if max(index1>index2(n_i2(i)+1) & index1<=n_i(i+1))==1

&

index3=

index1<=n i(i+1));

index4= index2(index2>n_i(i) & index2<= n_i(i+1));

index1(index1>index2(n i2(i)+1))

Z1(index3) = -1;

for ii= 1: (length(index4)-1)

```
tau_h2(n_i2(i)+ii+1)= tau_h2(n_i2(i)+ii+1)+sum(tau_h(index3(index3<index4(ii+1))))+sum(p_h(index3(index3<index4(ii+1))))+sum(p_h(index3(index3<index4(ii+1)))));
```

end

end

```
end
```

end

end

end

```
n_i= n_i2;
```

```
M1= M;
```

elseif length(unique(n_i2)) == length(n_i2) & n_i2(1) == 0

M1= M-1;

if sum(index1<=n i(1))~= 0

 $Z1(index1(index1 \le n_i(1))) = s_h1(index1(index1 \le n_i(1)));$

end

for i=1:M1

if $sum(index1 \le n_i(i+1)) \ge sum(index1 \le n_i(i))$

if max(index1(index1<index2(n_i2(i)+1))>n_i(i))==1

 $tau_h2(n_i2(i)+1) = tau_h2(n_i2(i)+1)+s_h1(max(index1<index2(n_i2(i)+1)))-(n_i(i)-n_i1(i)))+p_h1(max(index1<index1<index2(n_i2(i)+1)))-(n_i(i)-n_i1(i)));$

&

```
\label{eq:21} \begin{split} Z1(index1(index1<index2(n_i2(i)+1) & index1>n_i(i))) = \\ s_h1(index1(index1<index2(n_i2(i)+1) & index1>n_i(i)) - (n_i(i)-n_i1(i))); \end{split}
```

end

if max(index1>index2(n_i2(i)+1) & index1<=n_i(i+1))==1
index3= index1(index1>index2(n_i2(i)+1) & index1<=n_i(i+1));
index4= index2(index2>n_i(i) & index2<= n_i(i+1));
Z1(index3)= -1;
for ii= 1: (length(index4)-1)
if index4(ii+1)-index4(ii)>1

 $\label{eq:linear_state} \begin{array}{l} tau_h2(n_i2(i)+ii+1)=\\ tau_h2(n_i2(i)+ii+1)+sum(tau_h(index3(index3<index4(ii+1) \& index3>index4(ii))))+sum(p_h(index3(index3<index4(ii+1) \& index3>index4(ii)))); \end{array}$

end

end

end

end

end

 $n_i = n_i 2(2:end);$

elseif length(unique(n_i2))~= length(n_i2) && n_i2(1)~= 0

M1= M-(length(n_i2)-length(unique(n_i2)));

if $sum(index1 \le n_i(1)) \ge 0$ && index1(1) = 1

 $tau_h2(1)=$

 $tau_h2(1)+s_h1(max(index1(index1<index2(1))))+p_h1(max(index1(index1<index2(1))));$

 $Z1(index1(index1<index2(1))) = s_h1(index1(index1<index2(1)));$

end

if sum(index1<=n_i(1))~= 0 && max(index1>index2(1) & index1<=n_i(1))==1

index3= index1(index1>index2(1) & index1<=n_i(1));

Z1(index3) = -1;

for ii= 1: (length(index4)-1)

if index4(ii+1)-index4(ii)>1

 $\label{eq:tau_h2(ii+1)=tau_h2(ii+1)+sum(tau_h(index3(index3<index4(ii+1))))+sum(p_h(index3(index3<index4(ii+1))));$

end

end

end

[n_i2,tt]= unique(n_i2);

```
if M1==1
```

if $sum(index1 \le n_i(2)) \ge sum(index1 \le n_i(1))$

```
Z1(index1(index1>n_i(1)))=
s_h1(index1(index1>n_i(1))-(n_i(1)-n_i1(1)));
```

end

else

 $n_{t}=n_{i}(tt);$ $n_{1}t=n_{i}(tt);$

for i= 1:(M1-1)

if $sum(index1 \le n_it(i+1)) \ge sum(index1 \le n_it(i))$

if max(index1(index1<index2(n_i2(i)+1))>n_it(i))==1

 $tau_h2(n_i2(i)+1)=tau_h2(n_i2(i)+1)+s_h1(max(index1<index2(n_i2(i)+1)))-(n_it(i)-n_i1t(i)))+p_h1(max(index1(index1<index2(n_i2(i)+1)))-(n_it(i)-n_i1t(i)));$

$$\label{eq:lindex1} \begin{split} &Z1(index1 < index2(n_i2(i)+1) & index1 > n_it(i))) = \\ s_h1(index1 < index2(n_i2(i)+1) & index1 > n_it(i)) - (n_it(i)-n_i1t(i))); \end{split}$$

end

a=1:length(n_i);

 $a1=n_i(a(n_it(i)==n_i)+1);$
&

if max(index1>index2(n i2(i)+1) & index1<=a1)==1

index3= index1(index1>index2(n_i2(i)+1) & index1<=a1);

index4= index2(index2>n it(i) & index2<= a1);

Z1(index3) = -1;

for ii= 1: (length(index4)-1)

if index4(ii+1)-index4(ii)>1

```
tau_h2(n_i2(i)+ii+1)= tau_h2(n_i2(i)+ii+1)+sum(tau_h(index3(index3<index4(ii+1))))+sum(p_h(index3(index3<index4(ii+1))))+sum(p_h(index3(index3<index4(ii+1))));
```

end

end

end

end

end

if tt(1)>1

 $\begin{array}{ccc} Z1(index1(index1>n_i(1) & index1<=n_i(tt(1))))= \\ s_h1(index1(index1>n_i(1) & index1<=n_i(tt(1)))-(n_i(1)-n_i(1(1)))); \end{array} \\ \end{array}$

end

for j=1:(length(tt)-1)

```
if tt(j+1)-tt(j)>1
```

 $Z1(index1(index1>n_i(tt(j)+1) & index1<=n_i(tt(j+1)))=s_h1(index1(index1>n_i(tt(j)+1) & index1<=n_i(tt(j+1)))-(n_i(tt(j)+1)-n_i1(tt(j)+1)));$

end

end

end

else

M1=M-1-(length(n i2)-length(unique(n i2)));

 $[n_i2,tt] = unique(n_i2);$

if sum(index1<=n i(1))~= 0

 $Z1(index1(index1 \le n_i(1))) = s_h1(index1(index1 \le n_i(1)));$

end

 $n_{it}=n_{i(tt)};$ $n_{i1}t=n_{i1(tt)};$

for i=1:M1

if $sum(index1 \le n_it(i+1)) \ge sum(index1 \le n_it(i))$

if $max(index1(index1 < index2(n_i2(i)+1)) > n_it(i)) = 1$

 $\label{eq:linear_tau_h2(n_i2(i)+1)=tau_h2(n_i2(i)+1)+s_h1(max(index1<index2(n_i2(i)+1)))-(n_it(i)-n_i1t(i)))+p_h1(max(index1(index1<index2(n_i2(i)+1)))-(n_it(i)-n_i1t(i)));$

 $Z1(index1 < index2(n_i2(i)+1) & index1 > n_it(i)) = s_h1(index1 < index2(n_i2(i)+1) & index1 > n_it(i)) = (n_it(i)-n_i1t(i)));$

end

 $a= 1:length(n_i);$ $a1= n_i(a(n_it(i)==n_i)+1);$

if max(index1>index2(n_i2(i)+1) & index1<=a1)==1

index3= index1(index1>index2(n_i2(i)+1) & index1<=a1);

index4= index2(index2>n_it(i) & index2<= a1);

Z1(index3) = -1;

for ii= 1: (length(index4)-1)

if index4(ii+1)-index4(ii)>1

 $tau_h2(n_i2(i)+ii+1)=$ tau_h2(n_i2(i)+ii+1)+sum(tau_h(index3(index3<index4(ii+1))))+sum(p_h(index3(index3<index4(ii+1))))+sum(p_h(index3(index3<index4(ii+1)))));

nd

end

end

end

170

end

end

if tt(1)>1

$$\label{eq:constraint} \begin{split} Z1(index1 < n_i(1) \& index1 < =n_i(tt(1))) = s_h1(index1(index1 > n_i(1) \& index1 < =n_i(tt(1))) - (n_i(1) - n_i(1))); \end{split}$$

end

for j=1:(length(tt)-1)

if tt(j+1)-tt(j)>1

 $\begin{array}{c} Z1(index1(index1>n_i(tt(j)+1)) & index1<=n_i(tt(j+1)))=0 \\ s_h1(index1(index1>n_i(tt(j)+1)) & index1<=n_i(tt(j+1)))-(n_i(tt(j)+1)-n_i1(tt(j)+1))); \end{array}$

end

end

 $n_i = n_i 2(2:end);$

end

tau_h= tau_h2; p_h= p_h(index2); Z(N+1,Z(N+1,:)==0)= Z1; M=M1; n=n+1;

else

Z1=Z(N+1,Z(N+1,:)==0); $Z1(index1)=s_h1;$ Z(N+1,Z(N+1,:)==0)=Z1;

end

end

if H== 1

```
Y=bigurobi(H,K,M,Nb1,Q_h,n_i,p_h,Q_kj1,A_kj1,d_h,tau_kh,tau_h);
```

```
Z(N+1,Z(N+1,:)==0)=Y(2*K*Nb1+2);
```

end

```
if max(Z(N+1,:)==-1)==1
```

for i= 2:H0

```
if Z(N+1,i) = -1
```

```
Z(N+1,i) = Z(N+1,i-1)+p_h0(i-1)+tau_h0(i);
```

end

end

end

 $T(N) = sum(max(Z(N+1,:)+p_h0-d_h0,0));$

end

Time= toc;

[a,b]= min(T);

minsumT= a;

minZ = Z(b+1,:);

Time

minsumT

minZ

Appendix III. Matlab Managerial Insight Code for Problem P

H0= 50 ; M0= 4; Nb0= 30; K0= 3;

n i0= [12,24,37,50];

x= 0:0.1:0.6;

y= zeros(10,length(x));

for ii=1:10

Q_h0= rand(1,H0)*80+20;

p_h0= Q_h0/20;

Q_kj0= rand(K0,Nb0)*50+50;

A kj0= cumsum(rand(K0,Nb0)+5,2)-5;

d_h0= rand(1,H0)*H0;

d_h0(1:n_i0(1))=sort(d_h0(1:n_i0(1)));

for i=2:M0

$$d_h0((n_i0(i-1)+1):n_i0(i)) = sort(d_h0((n_i0(i-1)+1):n_i0(i)));$$

end

```
xh= sort(rand(1,H0))*100; yh= rand(1,H0)*100;
```

xk= sort(rand(1,K0))*100; yk= rand(1,K0)*100;

xm= sort(rand(1,M0))*100; ym= rand(1,M0)*100;

tau_kh0= zeros(K0,H0);

for k=1:K0

for h=1:H0

```
tau_kh0(k,h) = sqrt((xh(h)-xk(k))^2+(yh(h)-yk(k))^2)/50;
```

end

end

tau_h0= zeros(1,H0);

```
tau_h0(1) = sqrt((xm(1)-xh(1))^2+(ym(1)-yh(1))^2)/50;
    for h=2:H0
         tau_h0(h) = sqrt((xh(h)-xh(h-1))^2+(yh(h)-yh(h-1))^2)/50;
    end
    for m=1:(M0-1)
         tau_h0(n_i0(m)+1)=
sqrt((xm(m+1)-xh(n_i0(m)+1))^2+(ym(m+1)-yh(n_i0(m)+1))^2)/50;
    end
    y(ii,1)= fiter2(H0,K0,M0,Nb0,Q_h0,n_i0,p_h0,Q_kj0,A_kj0,d_h0,tau_kh0,tau_h0);
    for i=2:length(x)
         per=x(i);
         q= per*sum(Q_h0);
         r=rand(1,K0);
         Q_kj0(:,1)= q*r/sum(r);
         y(ii,i)= fiter2(H0,K0,M0,Nb0,Q_h0,n_i0,p_h0,Q_kj0,A_kj0,d_h0,tau_kh0,tau_h0);
    end
```

```
y(ii,:)= y(ii,:)/y(ii,1);
```

end

```
ybar= sum(y,1)/ii;
```

plot(x,ybar)

xlabel('x')

ylabel('ybar')

title('graph')

Appendix IV. Matlab Simulation Code

```
T = 100;
```

```
hos.number = 20;
team.number = 3;
dc.number = 3;
dcMidx = 3;
hos.loc = 10*rand(2,hos.number);
dc.loc = 10*rand(2,dc.number);
team.loc = 10*rand(2,team.number);
dc.Q=zeros(dc.number,T);
dc.A=zeros(dc.number,T);
dc.Acon=zeros(dc.number,T);
for k=1:dc.number
    for t=1:T % number of periods
         dc.Q(k,t)=fix(200*rand);
         dc.A(k,t)=fix(12*rand);
         dc.Acon(k,t) = max(0,dc.A(k,t)+normrnd(0,3*rand)); %confirmed arrival time
    end
end
hos.Q = zeros(hos.number,T);
hos.p = zeros(hos.number,T);
hos.d = zeros(hos.number,T);
hos.Qcon = zeros(hos.number,T);
for h=1:hos.number
    for t=1:T
         %hos.Q(h,t)=fix(rand*30)*(1 - 2*abs(0.5-t/T));% demand of hospital h at period t,
integer
         hos.Q(h,t)=fix(rand*30)*poisspdf(t,50);
         hos.d(h,t)=fix(6+12*rand); % due date of hospital h at period t, integer
         hos.p(h,t)=hos.Qcon(h,t)/5; % processing time of hospital h at period t
    end
end
tau mh=zeros(team.number,hos.number);
tau kh=zeros(dc.number,hos.number);
team0 = team;
dc0 = dc:
hos0 = hos;
for runtype = 0:1
```

+

```
team = team0;
    dc = dc0;
    hos = hos0;
    hos.stock = zeros(hos.number,1);
    dc.stock = zeros(dc.number,1);
    tardiness = ones(hos.number,T)*24;
    if runtype == 0
         disp('With mobile DC, Average tardiness is');
    end
     if runtype == 1
         disp('Without mobile DC, Average tardiness is');
    end
     for t = 1:T
         if runtype == 0
              if t>1
                    [\sim,hidx] = max(hos.Q(:,t));
                   dc.loc(:,dcMidx) = hos.loc(:,hidx);
                    dc.Acon(dcMidx,t) = fix(6*rand);
              end
         end
         dc.stock = dc.stock + dc.Q(:,t);
         for h=1:hos.number
              hos.Qcon(h,t) = max(1,hos.Q(h,t)+normrnd(0,5));
         end
         hos.t = zeros(hos.number,1);%latest batch arrival time
         hos.isServed = zeros(hos.number,1);
         hos.isServed(hos.Qcon(:,t)==0) = 1;
         tardiness(hos.isServed == 1,t) = 0;
         team.t = zeros(team.number,1);
         for k=1:dc.number
              for h=1:hos.number
                   tau kh(k,h)=sqrt(
                                                    (dc.loc(1,k)-hos.loc(1,k))^2
(dc.loc(2,k)-hos.loc(2,k))^2);
              end
         end
          [~,hosOrder] = sort(hos.d(:,t));
          for h = hosOrder'
              [\sim, dcOrder] = sort(tau kh(:,h));
              for k = dcOrder'
                   if hos.stock(h) >= hos.Qcon(h,t)
                         break;
                   end
```

```
trans = min(hos.Qcon(h,t) - hos.stock(h),dc.stock(k));
                   if trans > 0
                         dc.stock(k) = dc.stock(k) - trans;
                        hos.stock(h) = hos.stock(h) + trans;
                        hos.t(h) = max(hos.t(h), dc.Acon(k,t)+tau kh(k,h));
                   end
              end
              if hos.stock(h) < hos.Qcon(h,t)
                   if t<T
                         hos.Q(h,t+1) = hos.Q(h,t+1) + hos.Qcon(h,t);
                   end
              end
         end
         hos.isServed(hos.Qcon(:,t)>hos.stock) = -1;
         flag = 1;
         while flag == 1
              for m=1:team.number
                   for h=1:hos.number
                         tau mh(m,h)=sqrt((team.loc(1,m)-hos.loc(1,h))^2+
((team.loc(2,m)-hos.loc(2,h)))^2);
                   end
              end
              for m = 1:team.number
                   if sum(hos.isServed==0) == 0
                         flag = 0;
                        break;
                   end
                   serveIdx = find(hos.isServed==0);
                   [\sim, idx] = min(tau mh(m, serveIdx));
                   h = serveIdx(idx);
                   team.loc(:,m) = hos.loc(:,h);
                    team.t(m) = team.t(m) + tau mh(m,h);
                   team.t(m) = max(hos.t(h), team.t(m));
                   if hos.p(h,t) + team.t(m) < 24
                        team.t(m) = team.t(m) + hos.p(h,t);
                        hos.stock(h) = hos.stock(h) - hos.Qcon(h,t);
                        hos.isServed(h) = 1;
                         tardiness(h,t) = max(0,team.t(m)-hos.d(h,t));
                   else
```

```
team.t(m) = 24; end
```

end

```
if flag == 0
break;%all the hospital are served
end
%other wise, check team
flag = 0;
for m = 1:team.number
if team.t(m) < 24
flag = 1;
end
end
end
```

end

```
tardiness_t = sum(tardiness,1);
tardiness_h = sum(tardiness,2);
tardiness_m(runtype +1) = mean(tardiness(:));
disp(tardiness_m(runtype+1));
hos.Q;
hos.Qcon;
```

end

VITA

Shengbin Wang

1984	Born in Harbin, China
2003-2007	B.S., Zhejiang University, China
2005	First Prize of Chinese Undergraduate Mathematical Contest in Modeling
2006	Honorable Mention of Mathematical Contest in Modeling (MCM)
2007-2009	M.S., New Jersey Institute of Technology, NJ, USA
2007-2009	Teaching Assistantship, New Jersey Institute of Technology
2009-2013	Ph.D. in Management (Supply Chain Management Major), Rutgers University
2009-2013	Teaching Assistantship, Rutgers University
2010-2013	Dean's Fund Research Fellowship, Rutgers Business School

Publications

2012	"Urgent Epidemic Control Mechanism for Aviation Networks", with Peng, C., Shi, M., and Jin, X., Practical Applications of Intelligent Systems Advances in Intelligent and Soft Computing, 2012, Vol. 124, pp. 355- 360.
2012	"Structure Properties of an Operations Scheduling Problem with Renewable and Non-Renewable Resources", with Lei, L. and Pinedo, M., Refereed Proceedings of 2012 MSOM Annual Conference, Columbia University, New York, NY.
2013	"Operations Scheduling with Multiple Resources and Transportation Considerations", with Lee, K., Lei, L., and Pinedo, M., International Journal of Production Research, Vol. 51, Nos. 23-24, pp. 7071–7090.