

SUPPLY CHAIN SCHEDULING FOR MULTI-MACHINES AND
MULTI-CUSTOMERS

A Thesis Submitted to the College of
Graduate Studies and Research
In Partial Fulfillment of the Requirements
For the Degree of Doctor of Philosophy
In the Division of Biomedical Engineering
University of Saskatchewan
Saskatoon

By

BIN HAN

© Copyright Bin Han, September 2015. All rights reserved.

PERMISSION TO USE

In presenting this thesis in partial fulfilment of the requirements for a Postgraduate degree from the University of Saskatchewan, I agree that the Libraries of this University may make it freely available for inspection. I further agree that permission for copying of this thesis in any manner, in whole or in part, for scholarly purposes may be granted by the professor or professors who supervised my thesis work or, in their absence, by the Head of the Department or the Dean of the College in which my thesis work was done. It is understood that any copying or publication or use of this thesis or parts thereof for financial gain shall not be allowed without my written permission. It is also understood that due recognition shall be given to me and to the University of Saskatchewan in any scholarly use which may be made of any material in my thesis.

Requests for permission to copy or to make other use of material in this thesis in whole or part should be addressed to:

Head of the Division of Biomedical Engineering

University of Saskatchewan

57 Campus Drive, Saskatoon, Saskatchewan S7N 5A9

ABSTRACT

Manufacturing today is no longer a single point of production activity but a chain of activities from the acquisition of raw materials to the delivery of products to customers. This chain is called supply chain. In this chain of activities, a generic pattern is: processing of goods (by manufacturers) and delivery of goods (to customers). This thesis concerns the scheduling operation for this generic supply chain. Two performance measures considered for evaluation of a particular schedule are: time and cost. Time refers to a span of the time that the manufacturer receives the request of goods from the customer to the time that the delivery tool (e.g. vehicle) is back to the manufacturer. Cost refers to the delivery cost only (as the production cost is considered as fixed). A good schedule is thus with short time and low cost; yet the two may be in conflict. This thesis studies the algorithm for the supply chain scheduling problem to achieve a balanced short time and low cost.

Three situations of the supply chain scheduling problem are considered in this thesis: (1) a single machine and multiple customers, (2) multiple machines and a single customer and (3) multiple machines and multiple customers. For each situation, different vehicles characteristics and delivery patterns are considered. Properties of each problem are explored and algorithms are developed, analyzed and tested (via simulation).

Further, the robustness of the scheduling algorithms under uncertainty and the resilience of the scheduling algorithms under disruptions are also studied. At last a case study, about medical resources supply in an emergency situation, is conducted to illustrate how the developed algorithms can be applied to solve the practical problem.

There are both technical merits and broader impacts with this thesis study. First, the problems studied are all new problems with the particular new attributes such as on-line, multiple-customers and multiple-machines, individual customer oriented, and limited capacity of delivery tools. Second, the notion of robustness and resilience to evaluate a scheduling algorithm are to the best of the author's knowledge new and may be open to a new avenue for the evaluation of any scheduling algorithm. In the domain of manufacturing and service provision in general, this thesis has provided an effective and efficient tool for managing the operation of production and delivery in a situation where the demand is released without any prior knowledge (i.e., on-line demand). This situation appears in many manufacturing and service applications.

ACKNOWLEDGMENTS

I would like to express my gratitude to my supervisor, Professor W.J. Zhang, whose expertise, understanding, and patience, added considerably to my graduate experience. I am also thankful to my supervisor for his enthusiasm and faith in my capabilities. I would like to thank the other members of my advisory committee, Professor F. X. Wu, Professor R. Deters, Professor M. Keil, and Professor M. Gupta, for the assistance they provided during the whole process of my graduate research. I would like to thank the external examiner, Professor M. Chen, for the suggestions he provided for this thesis and the future research.

I also would like to thank my previous supervisor Professor X.W. Lu in the period of my study in East China University of Science and Technology of China for his guidance.

I also would like to thank the support of CSC scholarship, which guarantees to bear my study and living expenses in the four years.

Many thanks go to my family and friends, who support my research during the study.

TABLE OF CONTENTS

PERMISSION TO USE	i
ABSTRACT	ii
ACKNOWLEDGMENTS	iv
TABLE OF CONTENTS	v
LIST OF TABLES	x
LIST OF FIGURES	xii
ACRONYMS	xiv
NOMENCLATURE	xv
1 INTRODUCTION	1
1.1 Supply Chain Scheduling	1
1.2 The Problem Statement	2
1.3 Objectives and Scope of the Thesis	5
1.4 Organization of the Thesis	7
2 BACKGROUND AND LITERATURE REVIEW	8
2.1 Introduction	8
2.2 Preliminaries	9
2.2.1 Evaluation of Algorithm	9
2.2.2 Intelligent Algorithm	16

2.2.3	On-line Problem and On-line Algorithm	20
2.2.4	Traveling Salesman Problem	21
2.2.5	Concepts in Classical Scheduling	22
2.2.6	Multi-agent Scheduling	25
2.2.7	Robustness and Resilience	26
2.3	Supply Chain Scheduling	27
2.3.1	Classification of the Supply Chain Scheduling Problem	29
2.3.2	Algorithms for Class 1 - Distribution as an Extension of Production	32
2.3.3	Algorithms for Class 2 - Production and Distribution as Two Stages	35
2.3.4	Algorithms for Class 3 - More than Two Stages	40
2.3.5	Conclusion with Further Discussion	42
3	PROBLEM ASSUMPTIONS AND NOTATIONS	44
3.1	Problem Descriptions	44
3.2	Problem Assumptions	47
3.3	Problem Formulation and Notations	48
4	ALGORITHMS FOR SINGLE-MACHINE AND MULTI-CUSTOMERS	
	PROBLEMS	59
4.1	Algorithm for Problem SMP1	59
4.2	Algorithm for Problem SMP2	65
4.3	Algorithm for Problem SMP3	101
4.4	Algorithm for Problem SMP4	109
4.5	Algorithm for SMP Problems with Capacity Limited Vehicles	116

4.6	Simulated Experiment for SMH without Routing	119
4.7	Algorithm for Problem SMP5	132
4.8	Algorithm for Problem SMP6	138
4.9	Algorithm for Problem SMP7	144
4.10	Algorithm for Problem SMP8	147
4.11	Simulated Experiment for SMH with Routing	158
4.12	Concluding Remarks	172
5	ALGORITHMS FOR MULTI-MACHINES AND SINGLE-CUSTOMER	
	PROBLEMS	174
5.1	Algorithm for Problem MSP1	174
5.2	Algorithm for Problem MSP2	177
5.3	Algorithm for Problem MSP3	178
5.4	Algorithm for Problem MSP4	180
5.5	Algorithm for Problem MSP5	182
5.6	Simulated Experiment for MSH5	184
5.7	Concluding Remarks	186
6	ALGORITHMS FOR MULTI-MACHINES AND MULTI-CUSTOMERS	
	PROBLEMS	187
6.1	Algorithm for Problem MMP1	187
6.2	Algorithm for Problem MMP2	188
6.3	Algorithm for Problem MMP3	189
6.4	Algorithm for Problem MMP4	190

6.5	Simulated Experiment for MMH without Routing	191
6.6	Algorithm for Problem MMP5	197
6.7	Algorithm for Problem MMP6	198
6.8	Algorithm for Problem MMP7	199
6.9	Algorithm for Problem MMP8	200
6.10	Simulated Experiment for MMH with Routing	200
6.11	Concluding Remarks	206
7	ROBUSTNESS AND RESILIENCE OF ALGORITHMS	207
7.1	Introduction	207
7.2	Definition of Robustness	208
7.3	Definition of Resilience	208
7.4	Robustness of Algorithms	209
7.5	Resilience of Algorithms	212
7.6	Concluding Remarks	216
8	CASE STUDY: MEDICAL RESOURCES ALLOCATION	217
8.1	Emergency Management	217
8.2	Evacuation Problems	221
8.3	Problem Descriptions for The Medical Resources Allocation	222
8.4	Policy to Assign Aid Sites	226
8.5	Simulated Experiment for Case Study	227
8.6	Concluding Remarks	249
9	CONCLUSIONS	251

9.1 Overview and Contributions	251
9.2 Future Work	255
LIST OF REFERENCES	258
APPENDIX A LOWER BOUND FOR SMP2	273
APPENDIX B COPYRIGHT PERMISSIONS	275

LIST OF TABLES

Table 2.1: Orders of Common Functions	10
Table 2.2: An Overview of The Supply Chain Scheduling Problems	31
Table 4.1: Results of Algorithms SMH1-SMH4 with $k = 2$	122
Table 4.2: Results of Algorithms SMH1-SMH4 with $k = 3$	124
Table 4.3: Results of Algorithms SMH1-SMH4 with $k = 8$	125
Table 4.4: Results of Algorithms SMH1-SMH4 with $k = 10$	126
Table 4.5: Results of Algorithms SMH1-SMH4 with $k = 20$	127
Table 4.6: Results of Algorithm CSMH4 with $C = 2$	129
Table 4.7: Results of Algorithm CSMH4 with $C = 5$	130
Table 4.8: Results of Algorithm CSMH4 with $C = 8$	131
Table 4.9: Results of Algorithms SMH5, SMH6 and SMH8	171
Table 5.1: Results of Algorithm MSH5	185
Table 6.1: Results of Algorithms MMH1-MMH4 for $k = 2$	193
Table 6.2: Results of Algorithms MMH1-MMH4 for $k = 5$	194
Table 6.3: Results of Algorithms MMH1-MMH4 for $k = 10$	195
Table 6.4: Results of Algorithms MMH1-MMH4 for $k = 20$	196
Table 6.5: Results of Algorithms MMH5-MMH8 for $k = 2$	203

Table 6.6: Results of Algorithms MMH5-MMH8 for $k = 5$ 204

Table 6.7: Results of Algorithms MMH5-MMH8 for $k = 10$ 205

Table 6.8: Results of Algorithms MMH5-MMH8 for $k = 20$ 206

Table 7.1: The Results of SMH4 under Uncertainty 211

Table 7.2: The Results of SMH8 under Uncertainty 212

Table 8.1: Evacuation Problems for Different Network Models 223

Table 9.1: Results of Algorithms for All Problems 253

LIST OF FIGURES

Figure 1.1: The Layout of Supply Chain Scheduling	3
Figure 2.1: Comparison between Polynomial Function and Exponential Function	11
Figure 2.2: Comparison between Polynomial Function and Factorial Function .	12
Figure 2.3: Venn Diagram of Complexity Concepts for $P \neq NP$	16
Figure 2.4: Flow Chart for SA	18
Figure 2.5: Flow Chart for GA	19
Figure 3.1: Network of The Manufacturer and Customers	45
Figure 4.1: The Jobs Processing in η (1)	69
Figure 4.2: The Jobs Preparation in η (2)	72
Figure 4.3: The Modified Jobs Processing in The Interval $(l_1 D, C_{max}^{(2)}(\eta)]$	74
Figure 4.4: The Modified Jobs Processing in The Interval $(\lfloor \frac{C_{max}^{(1)}(opt)}{D} \rfloor D, C_{max}^{(2)}(\eta)]$	79
Figure 4.5: The Jobs Processing in η (3)	89
Figure 4.6: The Jobs Processing in η (4)	97
Figure 4.7: The Jobs Processing for The Worst Instance	101
Figure 4.8: The Jobs Processing in The Interval $(C_{max}(\mathcal{J}^{(1)}), C_{max}^{(1)}(\eta)]$ (1) . . .	111
Figure 4.9: The Jobs Processing in The Interval $(C_{max}(\mathcal{J}^{(1)}), C_{max}^{(1)}(\eta)]$ (2) . . .	149
Figure 4.10: The Results of SMH7 and Original GA for Case 1 with $k = 5$	161

Figure 4.11: The Results of SMH7 and Original GA for Case 1 with $k = 10$. . .	162
Figure 4.12: The Results of SMH7 and Original GA for Case 1 with $k = 20$. . .	163
Figure 4.13: The Results of SMH7 and Original GA for Case 2 with $k = 5$	164
Figure 4.14: The Results of SMH7 and Original GA for Case 2 with $k = 10$. . .	165
Figure 4.15: The Results of SMH7 and Original GA for Case 2 with $k = 20$. . .	166
Figure 4.16: The Results of SMH7 and Original GA for Case 3 with $k = 5$	167
Figure 4.17: The Results of SMH7 and Original GA for Case 3 with $k = 10$. . .	168
Figure 4.18: The Results of SMH7 and Original GA for Case 3 with $k = 20$. . .	169
Figure 7.1: The results of SMH7 under normal and disruption case with $k = 5$.	213
Figure 7.2: The results of SMH7 under normal and disruption case with $k = 10$	214
Figure 7.3: The results of SMH7 under normal and disruption case with $k = 20$	215
Figure 8.1: Great Disasters in Recent Years	218
Figure 8.2: Infectious Diseases in Recent Years	219
Figure 8.3: The Locations of 30 Aid Sites and 5 Medical Centers	230
Figure 8.4: The Locations of 30 Aid Sites and The Virtual Medical Center . . .	246
Figure 8.5: The Case Study Result for $D = 5$	247
Figure 8.6: The Case Study Result for $D = 25$	248
Figure 8.7: The Case Study Result for $D = 50$	249
Figure 10.1: The Construction of Lower Bound for SMP2	274

ACRONYMS

ACO:	ant colony optimization (pp. 17)
EM:	emergency management (pp. 7)
FPTAS:	fully polynomial-time approximation scheme (pp. 38)
GA:	genetic algorithm (pp. 17)
IPD:	integrated production-distribution (pp. 2)
MMP:	multi-machines multi-customers problem (pp. 56)
MSP:	multi-machines single-customer problem (pp. 54)
NN:	neural network (pp. 17)
NOLTSP:	nomadic OLTSP (pp. 22)
NP:	non-deterministic polynomial-time (pp. 13)
NPC:	NP-complete (pp. 14)
OLTSP:	on-line TSP (pp. 22)
P:	polynomial-time (pp. 13)
SA:	simulated annealing (pp. 17)
SMP:	single-machine multi-customers problem (pp. 53)
SNPC:	strongly NP-complete (pp. 14)
TSP:	traveling salesman problem (pp. 20)

NOMENCLATURE

I :	an instance
η :	a feasible schedule.
opt :	an optimal off-line schedule.
k :	the number of customers (aid sites).
s :	the number of medical centers.
K :	the set of all customers (aid sites), $\{1, 2, \dots, k\}$
m :	the number of machines (processors).
$J_j^{(i)}$:	the j th job for the i th customer (aid site), where $i = 1, 2, \dots, k$.
$\mathcal{J}^{(i)}$:	the set of all jobs for the i th customer (aid site).
$\mathcal{J}_{\leq t}^{(i)}$:	the set of all jobs for the i th customer (aid site) released before t .
n_i :	the number of all jobs for the i th customer (aid site).
n :	the number of all jobs, $n = \sum_{i=1}^k n_i$.
$r_j^{(i)}$:	the release time of job $J_j^{(i)}$.
<i>on – line</i> :	the jobs are released in the on-line environment.
$p_j^{(i)}$:	the processing (preparation) time for job $J_j^{(i)}$
$P^{(i)}$:	the sum of the processing (preparation) time of all the jobs for the i th customer.

- $P_{[t_1, t_2]}^{(i)}(\eta)$: the sum of the processing (preparation) time of the jobs for the i th customer which are processed in the interval $[t_1, t_2]$ in the schedule η .
- P : the sum of the processing (preparation) time of all the jobs.
- $pmtn$: the processing (preparation) of jobs can be interrupted and restarted later.
- $C_j^{(i)}(\eta)$: the completion time of job $J_j^{(i)}$ in the schedule η .
- $C_{max}^{(i)}(\eta)$: $\max_{J_j^{(i)} \in \mathcal{J}^{(i)}} C_j^{(i)}(\eta)$.
- idle time: when there are free machines (processors) and there are no uncompleted jobs.
- waiting time: when there are free machines (processors) but there are uncompleted jobs.
- delay: there is waiting time in the schedule.
- block: a time interval that a machine (processor) is not free.
- $C_{max}(U, m, \eta)$: the completion time for jobs set U being processed on m machines (processors) in schedule η .
- $C_{max}(U)$: the optimal maximum completion time for jobs set U being processed on a single machine (processor), which can be found by schedule all the jobs on the machine (processor) without delay.
- $\rho_j^{(i)}(\eta)$: the departure time of job $J_j^{(i)}$ in the schedule η .
- $\rho_{max}^{(i)}(\eta)$: $\max_{J_j^{(i)} \in \mathcal{J}^{(i)}} \rho_j^{(i)}(\eta)$, the latest time of delivery of jobs in η .
- T_{cd} : the transportation time between place c and place d , where $c, d = 0, 1, 2, \dots, k$ ('0' represents the manufacturer, '1, 2, \dots , k ' represent the customers).
- direct*: the jobs of different customers (aid sites) cannot share a batch, which means that all jobs need to be delivered to the corresponding customer (aid site) directly.
- routing*: the jobs of different customers (aid sites) can share a batch, which means that

a routing path is needed to deliver a batch.

$V(x, y)$: there are x vehicles available, each with a capacity y , where $x \in \{1, \infty\}$ and $y \in \{C, \infty\}$ (the symbol " ∞ " means "enough" in the engineering sense).

$D_j^{(i)}(\eta)$: the return time of the vehicle which delivers the job $J_j^{(i)}$ in the schedule η .

$D_{max}^{(i)}(\eta)$: $\max_{J_j^{(i)} \in \mathcal{J}^{(i)}} D_j^{(i)}(\eta)$, makespan of the i th customer (aid site) in the schedule η .

$\sum_{i=1}^k D_{max}^{(i)}(\eta)$: the total makespans in the schedule η .

D : the cost of one delivery which is a constant.

$TC(\eta)$: the total cost of all the deliveries in the schedule η , which is the number of deliveries timing by D .

$Z(\eta)$: weighted sum of two objectives in the schedule η , $w_1 \sum_{i=1}^k D_{max}^{(i)}(\eta) + w_2 TC(\eta)$, where w_1 and w_2 are two weights.

$U[a, b]$: the Uniform distribution on interval $[a, b]$.

$P(\lambda)$: the Poisson distribution with expectation λ .

$N(\mu, \sigma^2)$: the Normal distribution with expectation μ and standard deviation σ .

inf : infimum, greatest lower bound.

CHAPTER 1 INTRODUCTION

1.1 Supply Chain Scheduling

Supply chain is a network of autonomous and semi-autonomous business, which include supplier, manufacturer, delivery, warehouse, distributor, retailer and customers. By autonomous it is meant that all businesses in a supply chain are under independent managerial framework or have independent decision making powers. By semi-autonomous it is means that businesses in a supply chain may not be completely under different managerial frameworks. The operations of supply chain include order taking, material supply, production plan, job scheduling, cargo transportation, product storage and customer service. One assumption behind a supply chain is that different businesses are located differently. Therefore, transportation makes sense for the supply chain. A generic pattern in a supply chain is: production-transportation pair. The goal of supply chain management is to optimize the effectiveness of the whole supply chain system and operation by the reduction of cost, increase of quality, and reduction of supply time.

Supply chain scheduling is to make decisions on job flows over the production infrastructure and transportation infrastructure such that from a manufacturer's perspective, both the cost and time are minimized. The major difference between the supply chain

scheduling problem and classic production scheduling problem lies in that the former has to consider both production and delivery, namely integrated production and distribution (IPD).

Two different kinds of the supply chain scheduling are concerned. If the information of all jobs is known beforehand, it is called off-line supply chain scheduling. If the information of future jobs is not known beforehand of scheduling, it is called on-line supply chain scheduling. The off-line supply chain scheduling problem is similar to the traditional production planning problem. The on-line scheduling problem is to the situations where scheduling is carried out while jobs are arising.

1.2 The Problem Statement

This thesis considers the supply chain scheduling problem with a single manufacturer and one or multiple customers. The problem can be described as follows: The customers place orders of jobs to the manufacturer. The manufacturer processes the jobs on the machines and then delivers the completed jobs to the customers by the vehicles through a transportation network (see Figure 1.1).

Such a supply chain scheduling problem has many applications. The laptop assembly is an example where the customers order their specific laptops to a manufacturer. The manufacturer assembles the laptop computers on the assembly machines and then delivers computers to the customers. Another example is catering service, where customers place their orders through phone call. The restaurant cook the food on the hearth or in the oven and then deliver the dishes to the customers. Applications are also found in the

health service sector particularly in some emergent situation. When an epidemic disease suddenly arises, e.g., SARS, lack of proper drugs often occurs in some areas. Preparation of drugs and delivery of them to the areas in need falls into this problem.

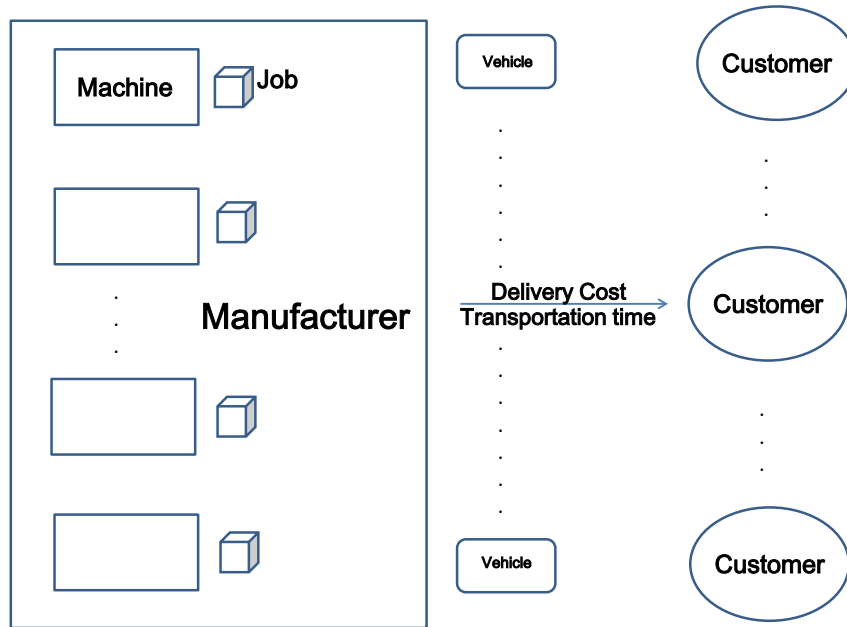


Figure 1.1. The Layout of Supply Chain Scheduling

Two performance measurements are concerned in supply chain scheduling: time and cost. It is always desired that the whole process has a short time and low cost. However, the two objectives may conflict with each other. For instance, reduction of the cost by having fewer vehicles for delivery of goods may certainly prolong the time that the customers receive the goods. Therefore, the supply chain scheduling problem is a multi-objective optimization problem in nature.

This thesis studied the supply chain scheduling problem which is particularly characterized by the following attributes: (1) there is a single machine and there are multiple customers

(mentioned before), (2) there are multiple machines and there is a single customer, and (3) there are multiple machines and there are multiple customers. For each type of problem characterized by these attributes, further information is specified, which includes the job release situation, job processing situation, job delivery situation, and characteristics of vehicles for job delivery. The time-based objective is the total makespan, which is the time interval from the point of time a job is released to the point of time a vehicle is returned to the manufacturing site. The cost-based objective is the total delivery cost (which is assumed to be the number of deliveries multiplied by the unit cost).

On a general note, the existing supply chain scheduling problem does not optimize the completion time for individual customers but the total completion time (i.e., the sum of the completion time of each customer) [Averbakh, 2010] or the maximum completion time among the completion time of each individual customer [Chen and Vairaktarakis, 2005]. It is however much desired in practice to schedule jobs such that the completion time of jobs associated with each individual customer is directly concerned and optimized. In this thesis, scheduling jobs to optimize the completion time for each individual customer is considered, and such a problem is called customer-oriented scheduling for short in this thesis. This looks like a paradigm shift in scheduling in the manufacturing environment. Besides this, the existing work on supply chain scheduling has not systematically studied the problem with multiple manufacturers and customers with constraints on the delivery tool (e.g., vehicle). Further, this thesis attempts to explore two new measures for the scheduling algorithm or even any general algorithm for the operations management: (1) the robustness of a supply chain scheduling algorithm from a system's perspective and

(2) the resilience of a supply chain scheduling algorithm. Finally, the problem of how the theory developed in this thesis is also addressed in the case of medical resources distribution in emergency situations.

1.3 Objectives and Scope of the Thesis

The overall objective of the thesis was to study the supply chain scheduling problem with the new paradigm that is the customer-orientation rather than the job-orientation. The problem is characterized by the three attributes as discussed before. It is noted that there could be more attributes to characterize the supply chain scheduling problem to a more realistic situation and the three attributes should then be viewed as general assumptions of the thesis. This thesis sought for the solution to the problem on the avenue of analytical algorithms rather than iterative algorithms. Note that the analytical algorithm directly generates the job actions and their orders, while the iterative algorithm generates the job actions and orders through a particular iteration scheme of searching in a feasible region of job actions and their orders.

Specific objectives were then defined in light of the overall objective.

Objective 1: To define the supply chain scheduling problem with the characteristics of one machine and multiple customers and to develop analytical algorithms, including their performance analysis, for the problem.

Objective 2: To define the supply chain scheduling problem with the characteristics

of multiple machines and single customer and to develop analytical algorithms, including their performance analysis, for the problem.

Objective 3: To define the supply chain scheduling problem with the characteristics of multiple machines and multiple customers and to develop analytical algorithms, including their performance analysis, for the problem.

Objective 4: To explore new measures to evaluate a supply chain scheduling algorithm, which may go beyond the existing measure (i.e. the worst scenario performance of an algorithm) but are focused on the robustness and resilience of a system along with its process.

Objective 5: To develop a test-bed in the area of medical resources allocation in emergency situations to give some idea of the effectiveness and efficiency of the analytical algorithms as developed in the first three objectives. A secondary objective with the test-bed development was to give guidelines to applications of the analytical algorithms for various scheduling problems in both manufacturing and service industries.

There are further specific assumptions for the development in this thesis, which are more problem-specific and algorithm-specific, and therefore, they appear in the occasion of specific problems and their algorithms are discussed.

1.4 Organization of the Thesis

Chapter 2 gives a background and review of the literature on supply chain scheduling, which also includes the preliminaries for algorithms and how they are evaluated. Chapter 3 will discuss the notation for problem descriptions and the definition of the problems that were tackled in this thesis. Chapter 4 will describe the algorithm for the problem of one machine and multiple customers, Chapter 5 will describe the algorithm for the problem of multiple machines and one customer, and Chapter 6 will describe the problem of multiple machines and multiple customers. Clearly, the foregoing three chapters correspond to the first three objectives (see the discussion in Section 1.3). Chapter 7 discusses potential new measures for algorithms, i.e., robustness and resilience. This chapter corresponds to the fourth objective. Chapter 8 will describe a case study for medical resources allocation in emergency management (EM), which corresponds to the fifth objective. Finally, Chapter 9 will summarize the contribution of this thesis, give conclusions drawn from the study presented in this thesis, and discuss future work.

CHAPTER 2 BACKGROUND AND LITERATURE REVIEW

2.1 Introduction

The purpose of this chapter is to provide the background for the problems studied in this thesis and to present a literature review of the state of arts in solving the problems studied in this thesis. In particular, Section 2.2 discusses basic concepts and notations related to the work of this thesis, such as complexity analysis, intelligent algorithm, on-line problem, traveling salesman problem, scheduling problem, multi-agent scheduling problem, robustness and resilience. Section 2.3 presents a review of the development on the topic of supply chain scheduling. Along with the review, a classification scheme is proposed for all supply chain scheduling problems based on how the delivery of completed goods is integrated with the production of goods. Particularly, this section proposes three classes of supply chain scheduling problems. In every class, its origin is briefly discussed and the latest literature is then commented and analyzed. Issues of the future research are also identified.

2.2 Preliminaries

2.2.1 Evaluation of Algorithm

The most common approaches to evaluate algorithms are "accuracy" and "run-time". For the problem that all information is known prior to the beginning of the problem solving (which is also called off-line problem), the accurate solution to the problem can always be achieved through enumeration. Therefore, the run-time performance becomes an important evaluation method (for off-line problems), that is given a number of accurate solutions, which one runs fastest.

Let \mathcal{P} denote a set of problems and I a particular problem in \mathcal{P} (I is also called an instance of \mathcal{P}). \mathcal{P} is characterized by a set of parameters. I is then defined by a set of values of the parameters of \mathcal{P} . Let \mathcal{A} denote an algorithm to \mathcal{P} and I . Particularly, the run-time performance of \mathcal{A} makes sense for I and when the run-time performances of \mathcal{A} for all I s of \mathcal{P} are known, the run-time performance of \mathcal{A} for \mathcal{P} makes sense. \mathcal{A} for I can be viewed as a set of operations on I , and the operations can further be decomposed into a series of basic operations, such as addition, subtraction, multiplication, division, comparison, and assignment. Assumes that all the basic operations take the same unit time. The run-time of \mathcal{A} for I can be defined as the number of basic operations on the set of specific values of the parameters. The value of the parameter can be represented by the binary format in the computer world. Let $n = |I|$ denote the sum of the lengths of the binary string of all the parameters of I . The run-time performance of \mathcal{A} for I is thus a function of n : $T(n)$. However, the analytical expression of $T(n)$ is difficult to be obtained. In practice,

the asymptotic bound of $T(n)$ is found, which measures the performance of \mathcal{A} for I .

Definition 2.1 [Knuth, 1976]. Let $f(n)$ and $g(n)$ be two functions defined on \mathbb{N} . $T(n) = O(f(n))$ if and only if there exists a positive real number c_1 and an integer n_1 such that $T(n) \leq c_1 f(n)$ for all $n > n_1$. $T(n) = \Omega(g(n))$ if and only if there exists a positive real number c_2 and an integer n_2 such that $T(n) \geq c_2 g(n)$ for all $n > n_2$. $T(n) = \Theta(f(n))$ if and only if $T(n) = O(f(n))$ and $T(n) = \Omega(f(n))$.

A short run-time is always desired, and the function $f(n)$ satisfying $T(n) = O(f(n))$ is concerned. Several common functions $f(n)$ for $O(f(n))$ are summarized in Table 2.1.

Table 2.1. Orders of Common Functions

Notation	Name
$O(1)$	constant
$O(\log n)$	logarithmic
$O(n)$	linear
$O(n \log n) = O(\log n!)$	loglinear
$O(n^2)$	quadratic
$O(n^a), a \geq 1$	polynomial
$O(a^n), a > 1$	exponential
$O(n!)$	factorial

Among them in Table 2.1, the polynomial function is of more interest as its computation rate is acceptable compared with the exponential function and the factorial function (Figure 2.1 and Figure 2.2). Therefore, development of the algorithms is very important,

the run-time of which is bounded by the polynomial function, that is, $T(|I|) = O(p(|I|))$ ($p(\cdot)$ is polynomial function). Such algorithms are also called "polynomial algorithm". However, finding polynomial algorithms is not guaranteed for problems, and this raises the question of whether a polynomial algorithm exists for every problem. The answer is still not known for sure but almost all researchers believe it is not true. In fact, a classification of problems in terms of polynomial algorithms is important to the development of algorithms. The following is a brief introduction of the theory for this classification.

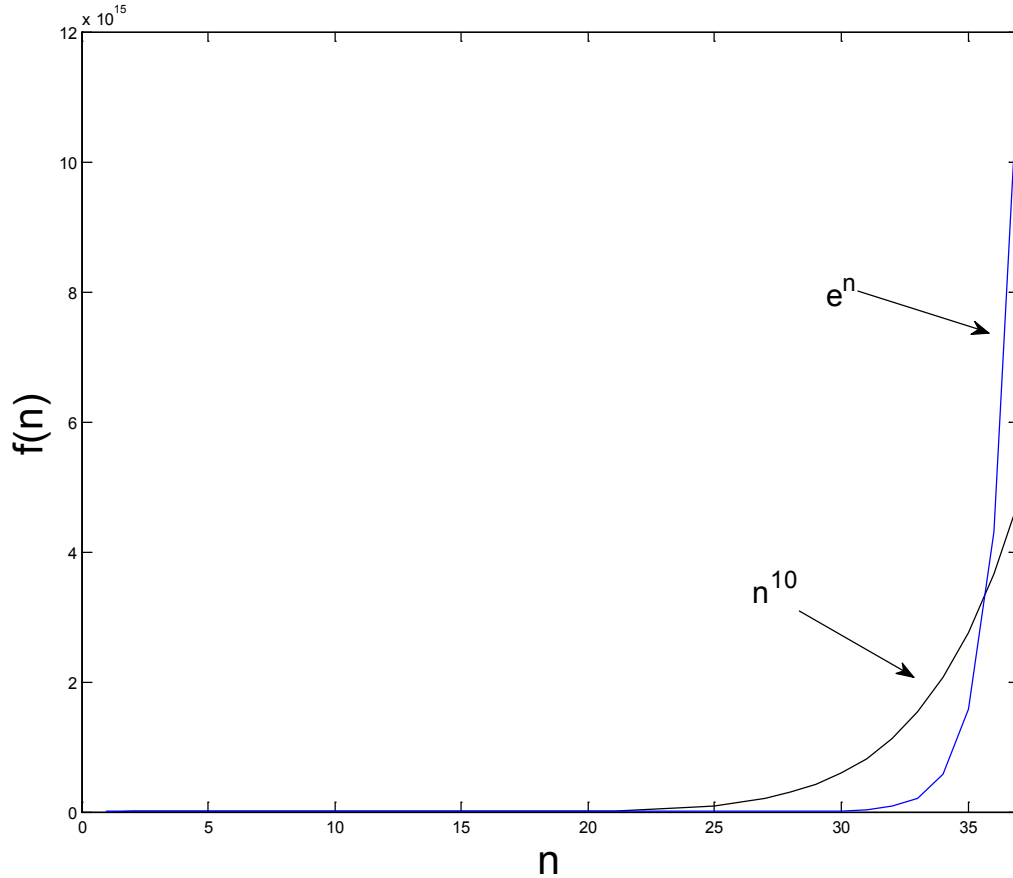


Figure 2.1. Comparison between Polynomial Function and Exponential Function

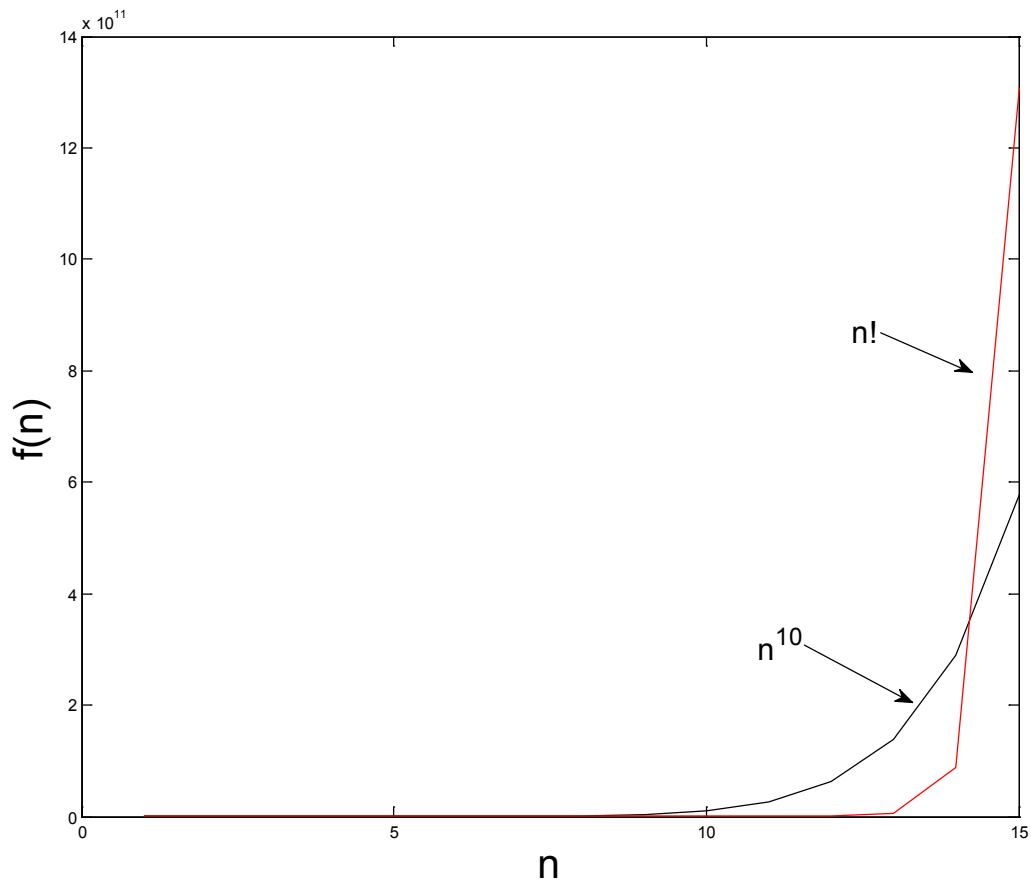


Figure 2.2. Comparison between Polynomial Function and Factorial Function

Definition 2.2. *A decision problem is a question with a yes-or-no answer and yes answer or no answer depends on the problem..*

In general, every problem can be viewed as a decision problem. For instance, for a minimization problem, the corresponding decision problem is whether the optimal value is less than a certain value. In particular, the decision problem cannot be "harder" than the original one, which implies that the intractability of the (original) problem is determined by the decision problem. In the following, the difficulty of a decision problem is discussed.

Definition 2.3. Let I be an arbitrary instance of decision problem \mathcal{DP} . If the answer of I is "yes" and the yes answer can be proved in $O(p(|I|))$ time, \mathcal{DP} is a non-deterministic polynomial-time (NP) problem.

Among the problems characterized as the NP problem, there is a class of problems that can be solved in the polynomial time.

Definition 2.4. Decision problem \mathcal{DP} is polynomial time problem (P), if for every instance I of \mathcal{DP} , the answer can be determined in $O(p(|I|))$ time.

It is noted that the above definition does not even require that \mathcal{DP} is in NP. This is because the proof for Definition 2.4 is exactly the proof in Definition 2.3. Therefore, $\mathcal{DP} \in P$ implies $\mathcal{DP} \in NP$, or $P \subseteq NP$.

The decision problems in P can be viewed as the easiest problems in NP. Now, the next step is to find the hardest ones in NP. Therefore, a method to compare the intractability of a decision problem is needed. The following concept is defined first.

Definition 2.5 [Karp, 1972]. Decision problem \mathcal{DP}_1 is polynomially transformed to decision problem \mathcal{DP}_2 , if for an arbitrary instance I_1 of \mathcal{DP}_1 , an instance I_2 of \mathcal{DP}_2 can be constructed in $O(p(|I_1|))$ time such that the answer of I_1 is yes if and only if the answer of I_2 is yes.

From Definition 2.5, if \mathcal{DP}_1 can be polynomially transformed to \mathcal{DP}_2 , \mathcal{DP}_2 having a polynomial algorithm implies that \mathcal{DP}_1 also having a polynomial algorithm. In other words, \mathcal{DP}_2 is harder than \mathcal{DP}_1 . Based on this comparison, the hardest problem in NP can be found.

Definition 2.6 [Garey and Johnson, 1979]. *Decision problem \mathcal{DP} is called NP-complete if:*

1. \mathcal{DP} is NP problem,
2. Every NP problem can be polynomially transformed to \mathcal{DP} .

Although NP-complete (NPC) problems are the hardest ones in NP, there are still different levels of intractability among these problems.

Definition 2.7. *Decision problem \mathcal{DP} is strongly NP-complete (NP-complete in the strong sense, SNPC), if it remains NP-complete even all parameters of I are bounded by $p(|I|)$.*

The above definitions have constructed an intractability framework for NP problems. A natural question is whether this framework can be implemented to general problems which may not be decision problems. The definition NPC and the method of polynomial transformation cannot be applied directly as they are only for decision problems. In the following, a method to compare the intractability of the general problems is given.

Definition 2.8 [Goldreich, 2008]. *Suppose that \mathcal{GP}_1 and \mathcal{GP}_2 are two problems, if \mathcal{GP}_1 's algorithm \mathcal{A}_1 calls \mathcal{GP}_2 's algorithm \mathcal{A}_2 a polynomial number of times, then \mathcal{GP}_1 is polynomially reducible to \mathcal{GP}_2 .*

It is noted that \mathcal{GP}_2 is harder than \mathcal{GP}_1 if \mathcal{GP}_1 is polynomially reducible to \mathcal{GP}_2 . In that sense, the intractability framework for NP problems can be implemented to the general problem.

Definition 2.9. *Problem \mathcal{GP} is called NP-hard, if a NPC problem is polynomially reducible to it. Problem \mathcal{GP} is called strongly NP-hard (SNP-hard), if a SNPC problem is polynomially reducible to it.*

Back to the original question: Does a polynomial algorithm exist for every problem? If the answer is 'yes', this implies that $P=NP$. However, this is still unknown and most researchers even believe that this is not true. Actually, a famous conjecture states that $P \neq NP$, which means there are decision problems which do not have polynomial algorithms [Gasarch, 2002; Rosenberger, 2012]. As NPC is a class of the hardest decision problems, the conjecture concludes that NPC problems do not have polynomial algorithms and nor do NP-hard problems. Figure 2.3 shows a venn diagram to illustrate the relationship among P, NP, NPC, SNPC, NP-hard and SNP-hard under the condition $P \neq NP$.

From the current point of view, it is almost impossible to develop polynomial algorithms for NP-hard problems, which frequently occurs in practice. Therefore, it makes sense to develop algorithms that are not accurate or that are approximate but with a good runtime performance, which leads to the notion of approximate algorithms. The definition of approximate algorithms for minimization (maximum) problem is as follows.

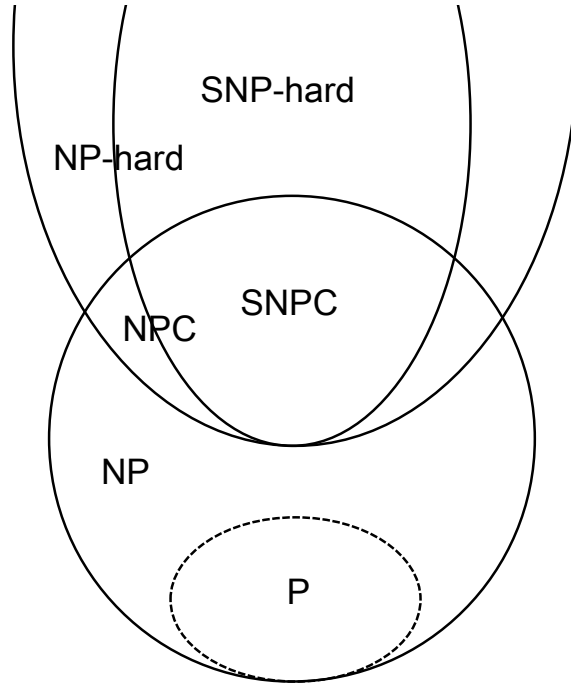


Figure 2.3. Venn Diagram of Complexity Concepts for $P \neq NP$

Definition 2.10. Let \mathcal{MP} be a minimization problem and I be an instance of \mathcal{MP} , $\mathcal{A}(I)$ be the value of objective function of algorithm \mathcal{A} for I and $OPT(I)$ be the optimal value for I . If $\frac{\mathcal{A}(I)}{OPT(I)} \leq r$ for all I and $r \geq 1$, then the algorithm \mathcal{A} is called a r -approximate algorithm. Furthermore, if $R_{\mathcal{A}} = \inf\{r \geq 1, \frac{\mathcal{A}(I)}{OPT(I)} \leq r, \text{ for all } I\}$, the algorithm \mathcal{A} has approximate ratio $R_{\mathcal{A}}$. A similar definition can be made for maximum problem by replacing $\frac{\mathcal{A}(I)}{OPT(I)}$ with $\frac{OPT(I)}{\mathcal{A}(I)}$.

2.2.2 Intelligent Algorithm

For the NP-hard problems that the accuracy is of the first importance, the traditional analytical algorithms cannot satisfy the requirement and then the notion of intelligent algorithms emerges. Different from the deterministic analytical algorithms, the intelligent

algorithms are stochastic, which empowers them to escape from local optimality and search for better results. Most common intelligent algorithms imitate the pattern of physical system, natural selection, pheromone communication, learning mechanism to construct the models, such as simulated annealing (SA), genetic algorithm (GA), ant colony optimization (ACO), neural network (NN). In the following, the SA and GA are introduced.

Simulated annealing (SA) is inspired by an annealing process in metallurgy and proposed in the early 80s [Kirkpatrick et al., 1983; Černý, 1985]. In the process of annealing, the cooling implies the decrease of molecular energy (temperature) and the crystallization of metal. Simulating this process, SA sets a temperature parameter and accepts worse solutions with a certain probability which falls with the decrease of the temperature parameter. One practically acceptable probability is $e^{-\frac{\Delta}{T}}$, where T is the temperature parameter and Δ is the difference of objective values between the current solution and the new solution. Figure 2.4 presents the flow chart for SA.

Genetic algorithm (GA) mimics the process of natural selection and dates back to the work of Holland [1975]. First, every solution is encoded as a string of chromosomes, which is also called individual. Next, GA initializes a population of solutions and evolves better solutions. The main process contains three operators: selection operator, crossover operator and mutation operator, which are illustrated as follows.

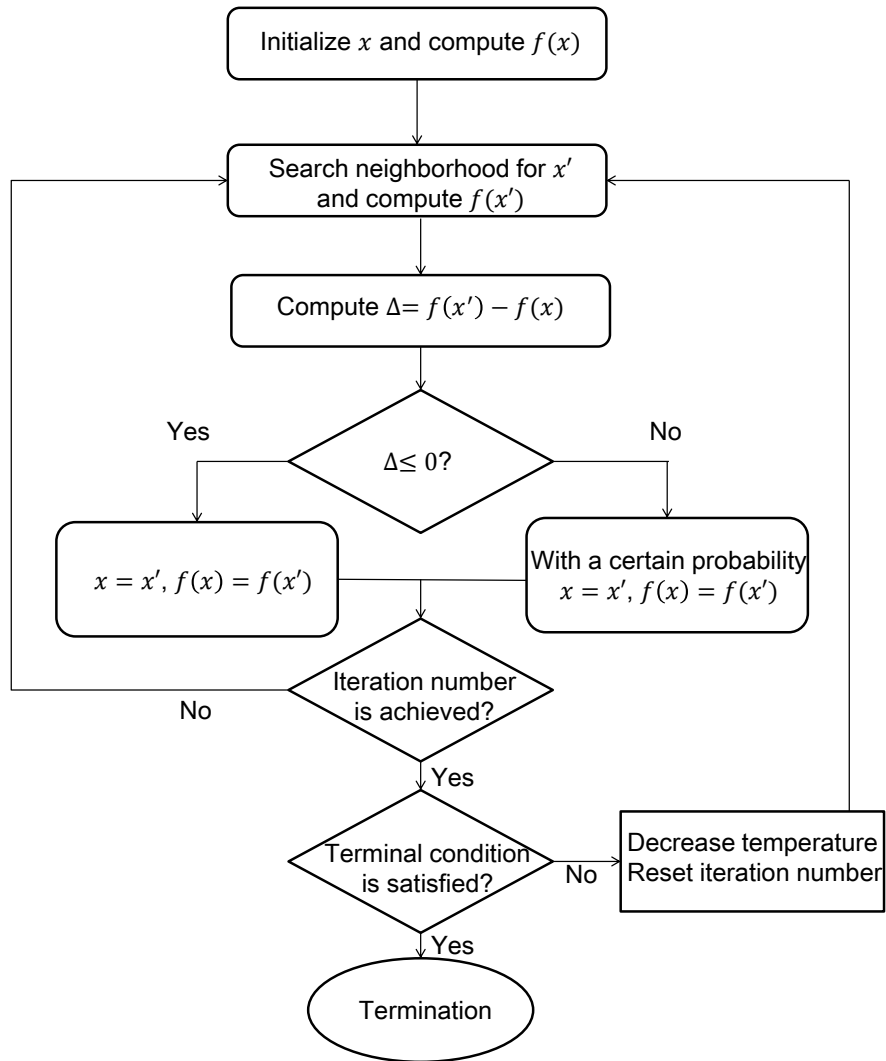


Figure 2.4. Flow Chart for SA

- A portion of the existing solutions in the population are selected for breeding the new generation. The selecting process bases on a fitness function which is related to the objective function. The solution with a higher fitness function is more likely to be selected.
- The selected solutions are pairwise coupled to crossover and generate new child solutions under a certain probability.

- Several chromosomes of a child solution are mutated to generate a new solution under a certain probability.

Figure 2.5 presents the flow chart for GA.

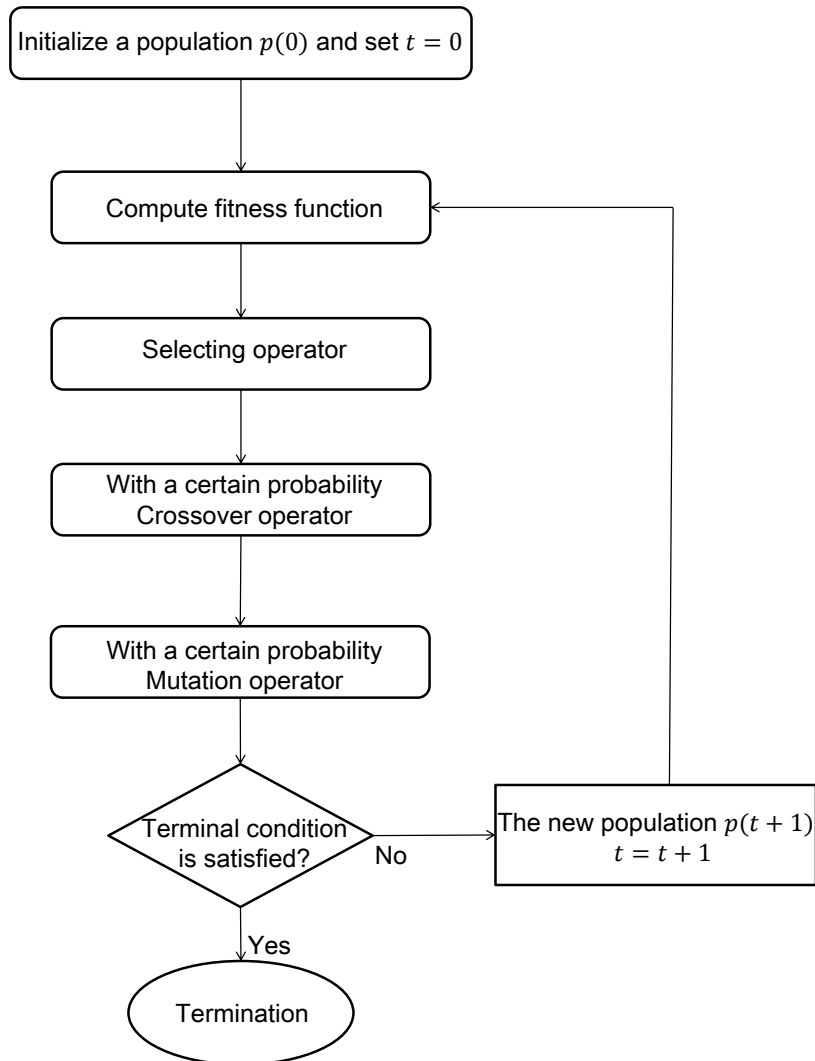


Figure 2.5. Flow Chart for GA

The probability that SA and GA can achieve a global optimal solution approaches 1 [Granville et al., 1994; Schmitt, 2004]. However, this convergency theory is asymptotical based on an infinity number of iterations, which is not practically applicable. In reality,

if a enough number of iterations are implemented, SA and GA can find a solution, which can be viewed as an important reference of the global optimal solution.

2.2.3 On-line Problem and On-line Algorithm

In the modern era of information explosion, on-line problems apply to many practical cases and arise in many areas [Albers, 2003]. A formal definition of the on-line problem can be described as follows.

Definition 2.11 [Albers, 2003]. *There is a request sequence of services $s = s(t_1), s(t_2), \dots, s(t_n)$, which must be served by a server. At time t , no knowledge of any request $s(t')$ with $t' > t$ is known. There will be a cost to serve for these requests and the goal of making service decisions is to minimize the total cost for the entire request sequence.*

From Definition 2.11, at time t , on-line algorithms must decide the services for requests $s(t')$ with $t' \leq t$ without the knowledge of requests $s(t'')$ with $t'' > t$. The evaluation of an algorithm is a step in the algorithm development process. Different from off-line algorithms, on-line algorithms can be measured through the so-called competitive ratio analysis [Borodin and El-Yaniv, 1998; Prush et al., 2004].

Definition 2.12. *Let \mathcal{OMP} be an on-line minimization problem and I be an instance of \mathcal{OMP} , $\mathcal{A}(I)$ be the value of the objective function of on-line algorithm \mathcal{A} for I and $OPT(I)$ be the off-line optimal value for I . If $\frac{\mathcal{A}(I)}{OPT(I)} \leq r$ for all I and $r \geq 1$, then the on-line algorithm \mathcal{A} is called a r -competitive algorithm. Furthermore, if $R_{\mathcal{A}} = \inf\{r \geq 1, \frac{\mathcal{A}(I)}{OPT(I)} \leq r, \text{ for all } I\}$, the on-line algorithm \mathcal{A} has competitive ratio $R_{\mathcal{A}}$. A similar definition can be made for on-line maximum problem by replacing $\frac{\mathcal{A}(I)}{OPT(I)}$ with $\frac{OPT(I)}{\mathcal{A}(I)}$.*

Based on the competitive ratio analysis, the measurement of the intractability of an on-line problem is developed, which is called the lower bound of the on-line problem.

Definition 2.13. *For an on-line problem, if no on-line algorithm can achieve a competitive ratio less than L , L is the lower bound of this on-line problem.*

To obtain such a lower bound, a series of 'bad' instances of the on-line problem need to be constructed and then to prove that no on-line algorithm can satisfy $\frac{A(I)}{OPT(I)} \leq L$ for these instances. The construction of these instances is sophisticated and is related to exploration of the structure properties. In particular, it is always desired to make L as large as possible.

From the developers' perspective, the competitive ratio of a certain on-line algorithm can be viewed as an upper bound of the on-line problem. Therefore, when the two bounds are identical, the on-line algorithm is considered to achieve the on-line optimality.

Definition 2.14. *The on-line algorithm for an on-line problem is called on-line optimal, if the competitive ratio of this on-line algorithm equals the lower bound of the on-line problem.*

2.2.4 Traveling Salesman Problem

The traveling salesman problem (TSP) aims to find the shortest route for a traveling salesman to visit each node of a given transportation network exactly once and return the origin. An instance of the TSP is given by a weighted graph and an initial vertex. The goal is to find a tour, i.e., a Hamiltonian circuit, which has a minimum length. The decision problem version of TSP is NP-complete because the Hamiltonian Circuit

problem can be polynomial transformed to it [Karp, 1972]. Therefore, TSP is a NP-hard problem. It has also been proved that the general TSP cannot be approximated within any constant unless $P=NP$ [Orponen and Mannila, 1990]. In the metric case, however, there is an approximation algorithm according to [Christofides, 1976]. His algorithm gives an approximation ratio of $\frac{3}{2}$. If the edge weights are restricted to 1 and 2, there is a $\frac{8}{7}$ -approximation algorithm [Berman and Karpinsk, 2006]. The situation is even more favorable in the Euclidean plane, for which Arora [1997] gives a $(1 + \varepsilon)$ approximation scheme.

In the on-line version of the problem (OLTSP), the salesman can communicate with the nodes to visit (also called the server) while he is traveling. Every request has a release time that represents the time when the request of the node is available to the salesman. The objective function is given by the maximum completion time (makespan). This variation is also called the nomadic OLTSP (NOLTSP). Ausiello et al. [2001] give a 2.5-competitive algorithm for general metric spaces and prove that no on-line algorithm can be better than 2-comprtitive. They also give a $\frac{7}{3}$ -competitive algorithm for the special case of the real line. Lipmann [2003] gives algorithm RETURN HOME which attains an improved upper bound of $(1 + \varepsilon)$ on general metric spaces. He also gives a 2.06-competitive algorithm for the real line, together with a lower bound for this case of approximately 2.03.

2.2.5 Concepts in Classical Scheduling

Scheduling theory governs the decision process for the rational use of resources to accomplish multiple tasks. The work of scheduling is rooted in manufacturing industry, so

the terms "job", "machine" and so on are wildly used in this field. Usually, notations J_1, J_2, \dots, J_n are used to represent jobs and M_1, M_2, \dots, M_m to represent machines, where n and m are the number of jobs and machines, respectively. A classic scheduling problem is to assign jobs to machines timely.

In scheduling, there are three machine configurations: single machine, parallel machines and dedicated machines. For the single machine case, there is only one machine to process jobs. For the parallel machines case, each machine has the same function and every job only needs to be processed on one machine. For the dedicated machines case, the machines have different functions and every job includes different operations which need to be processed in different machines. Flow shop is a special case of the dedicated machine where every job has one operation at one machine and all the jobs have the same order of operations.

Given a schedule η , the completion time $C_j(\eta)$ of job J_j in η can be determined, which is the time that J_j completes the processing (the last operation of J_j completes the processing for the dedicated machines case). When the schedule is not specified, the completion time is represented as C_j for short. Thus, the objectives of the scheduling are the function of jobs' completion time. The common objectives in scheduling problems are listed as follows.

- (1) C_{max} : the maximum completion time, $C_{max} = \max_{1 \leq j \leq n} C_j$.
- (2) L_{max} : the maximum lateness time, $L_{max} = \max_{1 \leq j \leq n} L_j$, where $L_j = C_j - d_j$ is the lateness of J_j and d_j is deadline.

- (3) T_{max} : the maximum tardiness time, $T_{max} = \max_{1 \leq j \leq n} T_j$, where $T_j = \max\{0, C_j - d_j\}$.
- (4) F_{max} : the maximum flow time, $F_{max} = \max_{1 \leq j \leq n} F_j$, where $F_j = C_j - r_j$ is flow time of J_j and r_j is release time.
- (5) $\sum_{j=1}^n (w_j)C_j$: the total (weighted) completion time, where w_j is the weight for J_j .
- (6) $\sum_{j=1}^n (w_j)T_j$: the total (weighted) tardiness.
- (7) $\sum_{j=1}^n (w_j)F_j$: the total (weighted) flow time.
- (8) $\sum_{j=1}^n (w_j)U_j$: the total (weighted) number of tardy jobs, where $U_j = \begin{cases} 0, & C_j \leq d_j, \\ 1, & \text{otherwise.} \end{cases}$

As this thesis aims to consider the makespan as the time objective, the related work of the classic scheduling problem is also reviewed in the following.

When the machine configuration is a case of parallel machines, there are m identical machines, and a particular machine and a particular job are however exclusively related to each other at any time. Several results have been obtained for the on-line problem of the classical parallel-machine scheduling to minimize the makespan. If the preemption of job processing is allowed, the 1-competitive on-line algorithm (Re-schedule Algorithm) is available according to Hong and Leung [1992], whereby the McNaughton algorithm is applied whenever there is a new job. The McNaughton algorithm finds the shortest preemption schedule on parallel machines [McNaughton, 1959]. If the preemption of job processing is not allowed, the longest processing time (LPT) rule can generate a $\frac{3}{2}$ -competitive schedule. According to the LPT rule, the job with the longest processing time is processed whenever

there are idle machines [Chen and Vestjens, 1997].

In the context of classic scheduling, the makespan refers to C_{max} . In the context of supply chain scheduling, completion of the task of a job includes not only the processing or production but also the delivery. Therefore, the completion of a job extends to the completion of the delivery of a job (particularly the event that the job delivery vehicle is back to the production or manufacturing site). D_j refers to the time that the delivery vehicle of job J_j is back to the production site. As such, for supply chain scheduling, the makespan is $D_{max} = \max_{1 \leq j \leq n} D_j$. Likewise, the corresponding lateness, tardiness, flow time, and number of tardy jobs can also be defined. The objective in classic scheduling is then extended to the objective in supply chain scheduling.

2.2.6 Multi-agent Scheduling

In the scheduling problem, when there is more than one customer, the competition for production resources among the customers needs to be considered. Each customer desires to achieve an optimality, so the problem is a multiple objectives problem. In literature, the customer is also viewed as agents, and the scheduling of multiple customers is also called multi-agents scheduling. There are different kinds of criteria for the multiple objectives optimization: minimize the primary objective while the others are bounded [Agnetais et al., 2004]; combine all the objectives into a single one [Baker and Smith, 2003]; formulate the Pareto-solutions [Agnetais et al., 2000]. However, most of the studies in literature are focused on the cases that there are two agents or customers [Ng et al., 2006; Mor and Mosheiov, 2010; Wang et al., 2010].

For the on-line problem for single machine and two customers, to minimize the total makespan, lower bound analysis was presented first, and then the on-line algorithms were given for both the preemption and non-preemption cases [Ding and Sun, 2010]. The algorithm for the preemption case achieved on-line optimal but the other one did not. For the problem with the batch-processing, an on-line algorithm with the competitive ratio of 2 was developed, which achieved an on-line optimum for the case that the capacity of batches were unbounded [Nong et al., 2008].

2.2.7 Robustness and Resilience

Robustness is a property that allows the system to be strong and health against the internal and external disturbances. This definition of robustness has been applied to many fields, including biology [Kitano, 2004; Félix and Wagner, 2006], control [Ray and Stengel, 1991; Bhattacharyya et al., 2000] and computer science [Baker et al., 2008; Sørensen, 2011]. Essentially, robustness in these fields reflects the ability of the system being insensitive to uncertainty. In engineering, particularly from an engineering system's perspective, strong and health become the performing functions. Robustness is thus related to the function of a system; particularly, a system still performs its required function under disturbances [Zhang and Lin, 2010]. In the field of algorithm for decision making (e.g., scheduling), uncertainty may be represented by the deterministic variability in the parameters and the robustness is particularly measured by the worst case performance of algorithm [Bertsimas and Sim, 2004; Ben-Tal et al., 2009], e.g., approximative analysis and competitive analysis mentioned before. In this thesis, both kinds of robustness, as aforementioned, are discussed for algorithms.

Resilience is a property that allows the system to be persistent against the changes, which is first proposed from ecology [Holling, 1973] and then applied to management and engineering [Zhang, 2007, 2008; Zhang and Lin, 2010]. The resilience of algorithms is also discussed in this thesis.

2.3 Supply Chain Scheduling

Scheduling refers to the timely allocation of resources to complete a task or job. The rise of the importance of scheduling is congruent with the age of mass production since the mid 50s [Johnson, 1954; Jackson, 1955; Smith, 1956; McNaughton, 1959]. The method for the best scheduling practice has been gradually improved in the second half of the last century, especially after the proposal of the three field notation [Graham, 1979]. A large number of research results were obtained in this period of time [Hu, 1961; Graham, 1969; Garey and Johnson, 1976; Gonzalez et al., 1977; Frederickson, 1983; Friesen, 1987]. However, these research results are limited in applications due to the rapid change of how the business world operates in the last decade. The traditional scheduling approach met challenges, as more and more new situations have appeared in the business world, demanding higher quality, cheaper price, and faster supply time.

The major change in the business world is that the business organization tends to be more dividing into small units, each of which keeps its core competence, and this makes the business organization more agile [Zhang et al., 1997]. Further, this change also leads that the business world is more like a network with nodes representing the business entities and edges representing their connections. Consequently, traditional scheduling, which focuses

on a single unit in the context of this network, faces a big challenge that is local optimal result. It is clear that scheduling must be conducted over an entire network (or chain in a bit narrow sense). Another generalization out of this network characteristic of the business world is that every activity is called supply.

The above has been pushing to the emergency of the notion of supply chain scheduling. In supply chain scheduling, it is particularly assumed that the manufacturer and the customer are not at the same place and thus the distribution of products from the manufacturer to the customer becomes an indispensable element to be considered. The scheduling in production with the coverage of distribution is a generic problem studied by many researchers in the last decade [Potts, 1980; Hall and Shmoys, 1992; Cheng et al., 1996]. In general, the integrated production-distribution (IPD) scheduling can significantly reduce the cost and improve the performance [Hall and Potts, 2003; Chen and Vairaktarakis, 2005].

Further, as the supply chain scheduling aims to describe a more realistic mechanism of industry, different situations have raised great attention in the scheduling community. In the first situation, all information of future jobs is known beforehand and thus the corresponding schedule can be decided beforehand. In the second situation, none of information of future jobs is known beforehand and the decisions of schedule are taken as jobs arrive at manufacturers. This second situation is also called on-line supply chain scheduling while the first situation is called off-line supply chain scheduling [Averbakh and Xue, 2007; Han, 2012]. As a schedule is determined by an algorithm, there are thus off-line algorithm and on-line algorithm, respectively. The evaluation of an algorithm is crucial to the efficiency

of a schedule. It is noted that the off-line algorithm and on-line algorithm can be measured by the so-called complexity analysis and competitive analysis, respectively (see the previous discussion in Section 8.2 and Section 2.2.3).

Several review articles of supply chain scheduling appear such as Sarmiento and Nagi [1999], Goetschalckx et al. [2002], Chen [2010]. In particular, Chen [2010] extended the three field notation for the classical scheduling problem to the five field notation for the supply chain scheduling problem. However, a systematic classification of the supply chain scheduling problem is still missing, and remedy of this deficiency is the motivation of the following discussion. .

Hereafter, a classification scheme for the problems of supply chain scheduling is first built, three classes in particular. Later, the existing articles are reviewed, which propose solutions to the supply chain scheduling problems against the three classes, especially the articles in the recent five years, summarize their results, and list them into the three class framework. For each class, the papers in literature are introduced first and their contributions are presented. The literature review is concluded and the directions of future research is discussed at the end of this chapter.

2.3.1 Classification of the Supply Chain Scheduling Problem

The supply chain scheduling problems are classified in terms of how distribution is integrated with production, or integration of production and distribution (IPD for short).

There are three ways of IPD, and they are:

- (1) The distribution is considered as a part of production and only the production cost is concerned [Hall and Shmoys, 1992; Lee and Chen, 2001; Wang and Cheng, 2007];
- (2) The production and the distribution are two sequential activities and both the production cost and the transportation cost are concerned [Chen and Vairaktarakis, 2005; Hall and Potts, 2005; Averbakh and Xue, 2007];
- (3) More than two activities in a total supply chain, such as supply, production, inventory, loading, setup, transportation and so on, are concerned [Hall and Potts, 2003; Lee et al., 2003; Delavar et al., 2010].

Three classes of IPD are practically meaningful. Class (1) refers to traditional manufacturers, e.g., steel mill [Cowling and Johansson, 2002]. Class (2) refers to emerging businesses, e.g., apparel business [Pundoor and Chen, 2005] and catering service [Chen and Vairaktarakis, 2005], which pay an equal attention to the production and distribution. Class (3) refers to a set of units that are globally distributed [Zhang et al., 1997; Viswanadham, 2002].

Although many solutions to the supply chain scheduling problem were proposed in the 1980s and 1990s, the rise of research in this field happened in the last decade, especially in the recent five years. Further, most of the new studies still belong to the above three classes. In Table 2.2, an overview of the existing solutions to supply chain scheduling problems is given based on the above classification. Then, in the subsequent sections, details of these solutions will be discussed.

Table 2.2. An Overview of The Supply Chain Scheduling Problems Based on The Classification

Classification	Objective	References
Class 1	makespan	[Potts, 1980], [Zdrzalka, 1991, 1994, 1995], [Hall and Shmoys, 1992], [Hall et al., 2001], [Lee and Chen, 2001], [Wang and Cheng, 2007], [Wang and Cheng, 2009a], [Li et al., 2011], [Dong et al., 2013], [Ng and Lu, 2012], [Liu et al., 2014], [Pei et al., 2014a,b], [Zhong and Lv, 2014], [Gao et al., 2015]
	total (weighted) delivery/flow time	[Lee and Chen, 2001], [Hall et al., 2001], [Li et al., 2005], [Fan, 2010], [Wan and Zhang, 2014]
Class 2	maximum lateness	[Hall et al., 2001], [Condotta et al., 2013]
	total (weighted) tardiness	[Hall et al., 2001], [Ullrich, 2013]
Class 3	total (weighted) number of tardy jobs	[Hall et al., 2001], [Li and Li, 2014]
	makespan and total delivery cost	[Chen and Vairaktarakis, 2005], [Li and Ou, 2005], [Tang et al., 2014], [Han, 2012]
Class 4	total (weighted) delivery/flow time	[Hall and Potts, 2003], [Hall and Potts, 2005], [Chen and Vairaktarakis, 2005], [Chen and Pundoor, 2006], [Li and Ou, 2007], [Averbakh and Xue, 2007], [Ji et al., 2007], [Mazdeh et al., 2007], [Mazdeh et al., 2011], [Averbakh and Baysan, 2012, 2013a,b], [Feng and Zheng, 2013], [Selvarajah and Zhang, 2014], Fan et al. [2015], [Hall and Potts, 2003],
	and total delivery cost	[Hall and Potts, 2003], [Kim and Oron, 2013a], [Gao, 2011]
Class 5	maximum lateness and total delivery cost	[Hall and Potts, 2005], [Steiner and Zhang, 2009], [Steiner and Zhang, 2011], [Kim and Oron, 2013b], [Rasti-Barzoki et al., 2013], [Rasti-Barzoki and Hejazi, 2013]
	total (weighted) earliness and total delivery cost	[Cheng et al., 1996], [Yang, 2000], [Gao, 2011]
Class 6	total (weighted) tardiness and total delivery cost	[Hall and Potts, 2003]
	total (weighted) number of tardy jobs and total delivery cost	[Hall and Potts, 2003]
Class 7	total (weighted) earliness and total delivery cost	[Ullrich, 2012], [Lee et al., 2012], [Lee et al., 2013], [Meinecke and Scholz-Reiter, 2014]
	total (weighted) delivery time	[Wang and Cheng, 2009b], [Qi, 2005], [Yeung et al., 2011]
Class 8	inventory cost and delivery cost	[Lee et al., 2003], [Bertazzi et al., 2005], [Sawik, 2009], [Lee and Yoon, 2010], [Wang et al., 2014]
	production cost, inventory cost and delivery cost	[Fan et al., 2010, 2013]
Class 9	shortage penalty, inventory cost and delivery cost	[Hajiaghaci-Keshteli and Aminmayeri, 2014]
	production cost, loading cost and delivery cost	[Baghalian et al., 2013]
Class 10	current investment cost, expected shortage cost, production cost and delivery cost	[Hajiaghaci-Keshteli et al., 2014]
	departure time earliness and tardiness costs, and delivery tardiness cost	[Alonso-Ayuso et al., 2013], [Celikbilek, 2014]
Class 11	total loading time of vehicles	[Alonso-Ayuso et al., 2013], [Celikbilek, 2014]
	total profit	[Celikbilek, 2014]

2.3.2 Algorithms for Class 1 - Distribution as an Extension of Production

The earliest paper may refer to Potts [1980], who considered that there was a transportation time for every completed job and the time objective included the delivery time of jobs. The machine configuration was a single-machine case and the objective was the makespan. For this problem, a $\frac{3}{2}$ -approximated algorithm was designed and analyzed. Later, the result was extended to the cases that there were different constraints on jobs. For instance, there was a fixed delivery date for jobs [Hall et al., 2001], there was a priority order of jobs [Hall and Shmoys, 1992], there was a setup time for the machine to start [Zdrzalka, 1991, 1995], and preemption was allowed [Zdrzalka, 1994].

The above studies only considered the situations where (1) there was one job for every customer, (2) there were a sufficient number of vehicles, and (3) a vehicle may contain one job only. Later, the problems of one customer but with a limited number of vehicles and with a limited capacity of vehicles were also studied [Lee and Chen, 2001; Li et al., 2005; Wang and Cheng, 2009a]. In addition, the problems with different machine configurations were studied, such as parallel machines [Wang and Cheng, 2007] and flow shop machines [Lee and Chen, 2001]. Chen [2010] made a survey for the supply chain scheduling problems as well as their algorithms for this class of problems with a focus on the constraints. In the following, the latest results of this class are reviewed in terms of different objectives of the problems.

The objective of the makespan was discussed by many researchers in the recent years. Li et al. [2011] investigated the batch processing. They assumed that the jobs of different

customers cannot be processed and delivered in the same batch. They showed that for one vehicle the problem was NP-hard. An algorithm with the approximate ratio of $\frac{3}{2}$ was developed. The configuration of two-machine open shop was studied by Dong et al. [2013]. They solved the problem of one vehicle with the vehicle capacity c . An algorithm with the approximate ratio of 2 was proposed for general c while a $\frac{3}{2}$ -approximation algorithm was proposed for $c = 1$ especially. Zhong and Lv [2014] studied the supply chain scheduling problem with the flow-shop configuration of machines, where stage one was single-machine and stage two was two parallel-machine. The jobs need to be transported between two stages and the vehicle can only take one job every shipment. They stated that the problem was strongly NP-hard and applied a heuristic with approximation ratio $\frac{3}{2}$. Numerical simulation was conducted to show the normal performance. Pei et al. [2014a] explored the supply chain scheduling problem with non-identical job sizes. They showed that the problem was strongly NP-hard and derived a lower bound. A two-phase heuristic algorithm was applied to solve the problem, which was $\frac{7}{2}$ -approximating. Later, they extended the work to the case that there were multiple manufacturers [Pei et al., 2014b]. A modified gravitational search algorithm was designed and a simulated experiment was conducted to demonstrate the performance. Gao et al. [2015] studied the problem that the production was batch processing where batch processing time is the summary of all jobs and the distribution was no-wait. They proved that the problem was strongly NP-hard and provided polynomial exact solutions for some special cases. For the general case, they designed a 2-approximate algorithm and numerically demonstrated the performance of the algorithms.

Ng and Lu [2012] studied the problem for single-machine and single-customer in the on-line environment. There was one vehicle with capacity c and the objective was also to minimize the makespan. When the preemption was allowed, an on-line algorithm with the best competitive ratio $\frac{\sqrt{5}+1}{2}$ was designed for $c \leq 2$. When the preemption was not allowed, an on-line algorithm with the best competitive ratio $\frac{\sqrt{5}+1}{2}$ was designed for $c = 1$ and an on-line algorithm with asymptotic competitive ratio $\frac{\sqrt{5}+1}{2}$ was designed for $c \leq 2$. Liu et al. [2014] also investigated the on-line problem of supply chain scheduling with the processing time being constrained in an interval $[p, \frac{\sqrt{5}+1}{2}p]$, where $p > 0$. The on-line algorithms with the competitive ratio of $\frac{\sqrt{5}+1}{2}$ were proposed for both cases that the capacity of vehicles was limited and unlimited, respectively. In particular, The result was on-line optimal for the case the capacity of vehicles was unlimited.

There were studies with the objectives other than the makespan. Fan [2010] considered the objective as a total flow time for the supply chain scheduling problem for single-machine and two customers. The total flow time differs from the makespan in that the former refers to the level of the whole procedure while the latter refers to the level of all jobs. The preemption of jobs processing was allowed and there was one vehicle with the unlimited capacity. The author analyzed the complexity of the problem and applied dynamic programming to solve it. Condotta et al. [2013] considered the supply chain scheduling problem to minimize the maximum lateness. The problem involved single machine, one customer and multiple vehicles of limited capacity. A mixed integer linear program was first formulated and a tabu search algorithm for production part was proposed. The coordinated solutions were generated by complementing every production scheduling with

an optimal distribution scheme. The performance of the solutions was shown in a computational simulation. Ullrich [2013] studied the problem to minimize the total tardiness. There were parallel-machine and multiple vehicles. In particular, the vehicles may be with a different capacity. In addition, the delivery of every job should be implemented in a time window. He applied a genetic algorithm for this problem and conducted a numerical study. It was shown that the algorithm had a great performance for small-size instances. Wan and Zhang [2014] investigated the case with m parallel-machine for single-customer to minimize the total delivery times, where v vehicles with limited capacity served the delivery. They proved that the problem was strongly NP-hard for arbitrary m and provided a $2 - \frac{1}{m}$ -approximation algorithm. The problem with the objective of the total number of tardy jobs was discussed by Li and Li [2014], where the departure date of delivery was prescribed. They showed that the problem can be solved by a polynomial algorithm.

2.3.3 Algorithms for Class 2 - Production and Distribution as Two Stages

In this situation, the distribution becomes an independent business entity. This means that the transportation or delivery may work for multiple units. Therefore, both the production cost and the transportation cost are considered. Research on supply chain scheduling for this situation may refer to the studies of [Cheng et al., 1996; Yang, 2000]. The first systematic work for this situation was from Hall and Potts [2003], who integrated the transportation cost with classical scheduling problems and considered the case of single-machine and single-customer. There were a sufficient number of vehicles and the capacity of all vehicles was unlimited in their problem. The production costs in their paper were the common objective of classical scheduling problems, such as the total (weight-

ed) completion times, the total (weighted) flow times, the maximum lateness, the total (weighted) number of tardy jobs. For each problem, they either proved the NP-hardness or proposed a polynomial algorithm. They extended their work to the case that there was constraint on vehicles [Hall and Potts, 2005]. The basic work for multiple customers was proposed by Chen and Vairaktarakis [2005], who aimed to minimize the makespan or the total completion times with both single-machine and parallel-machine. There were a sufficient number of vehicles but the capacity of all vehicles was limited in their problem. They assumed that the number of customers was prescribed and applied enumeration for the routing. After that, the number of studies in this field with the different characteristics of real life problems increased dramatically. For instance, there were multiple manufacturers [Chen and Pundoor, 2006; Li and Ou, 2007], there were both time-based objective and cost-based objective [Chen and Pundoor, 2006], vehicle characteristics were different [Li and Ou, 2005], the delivery batches were constrained [Ji et al., 2007], etc. In particular, as the combination into a single objective was one method for the problem with multiple objectives, another method was to minimize the primary objective with the others bounded. As such, there were results to minimize the transportation cost with the production cost being bounded [Chen and Pundoor, 2006, 2009]. The survey of Chen [2010] also reviewed the work in these problems.

There were a lot of studies for the objective of the total (weighted) flow/delivery times and the transportation cost. Mazdeh et al. [2007] addressed the scheduling with single-machine for single-customer to minimize the total flow times and the transportation cost. They devised a branch-and-bound algorithm and showed a significant improvement by

simulated experiments. They later extended their result to the case with the batch processing machine to minimize the total weighted flow times and the transportation cost [Mazdeh et al., 2011]. Another work of the batch processing machines was from Feng and Zheng [2013], which aimed to minimize the total delivery times and the transportation cost. For both cases with the unbounded and bounded batch, dynamic programming algorithms were developed. Chang et al. [2014] considered the case with the parallel-machines for multi-customers to minimize the total weighted delivery times and the transportation cost. There were a sufficient number of vehicles with the limited capacity in their problem. An algorithm using ant colony techniques was applied to search near-optimal solutions. Selvarajah and Zhang [2014] considered the special case that the jobs could be outsourced. When the outsourcing budgets were limited, they showed that it was NP-hard to minimize the total delivery times and the transportation cost and proposed a pseudo-polynomial algorithm and a polynomial approximation algorithm. When the outsourcing budgets were unlimited, they stated the equivalence of the problem and the shortest path problem. Fan et al. [2015] explored the single-machine configuration with an availability constraint to minimize the total delivery times and the transportation cost. For the resumable case that the jobs processing can be continued after interrupting, an optimal algorithm was developed; for the non-resumable case that jobs processing must be re-started after interruption, a $\frac{3}{2}$ -approximate algorithm was developed.

Another main concern of the objective was the total weighted number of tardy jobs and the total transportation cost. Steiner and Zhang [2009] studied the supply chain scheduling problem for single-machine and multi-customers. Jobs of the same customer could

be processed and delivered in batches and there was a batch setup time. Due to the NP-hardness of the problem, a pseudo-polynomial algorithm was designed for a restricted case and then a fully polynomial-time approximation scheme (FPTAS) was proposed. They also considered the case that the due dates could be relaxed with a penalty cost [Steiner and Zhang, 2011]. Kim and Oron [2013b] explored the problem with the multi-location production for single-customer. A vehicle with the limited capacity was available and all jobs had the same processing time. They solved the problem by reducing it to a shortest-path problem but the algorithm was exponential for a general number of machine locations. The numerical results were presented for the single-machine case. The work of Rasti-Barzoki et al. [2013] addressed the problem with two stages of the processing of single-machine and two flow-shop machines. The number and the capacity of vehicles were sufficient and single-customer was considered. They analyzed the structural properties of both cases and derived a branch and bound algorithm, which outperformed the dynamic programming algorithm. The result was extended to the case of multi-customers and that the total weighted tardiness was added into the objective [Rasti-Barzoki and Hejazi, 2013]. A heuristic algorithm and a branch and bound algorithm were provided.

The studies with other objectives were studied in literature as well. Gao [2011] considered the objective of the total weighted tardiness, the total weighted earliness, and the transportation cost, where there was single-machine for multi-customers and there were a fleet of vehicles with limited capacity. A modified greedy algorithm was applied for this problem. Kim and Oron [2013a] studied the total weighted tardiness and the transportation cost with multi-location production for single-customer. They designed an algorithm which

was exponential to the number of machines. Special cases with reduced computational complexity were further discussed. With a single batching machine for one customer, Tang et al. [2014] applied different optimization treatments to bi-criteria (the makespan and the transportation cost) supply chain scheduling. Four variations of the problem are defined: to minimize the makespan and the transportation cost (P1), to minimize the makespan with the transportation cost bounded (P2), to minimize the transportation cost with the makespan bounded (P3), and to find the Pareto set (P4). After proving the strongly NP-hardness of these problems, heuristic algorithms with worst case analysis were proposed for P1, P2 and P3, while an exact algorithm was designed for P4.

The on-line version of this algorithm was first considered by Averbakh and Xue [2007]. There was a single-machine and there were a sufficient number of vehicles with an unlimited capacity. The objective was to minimize the total flow time and the delivery cost. Due to lack of future information, there was a lower bound 2 for the competitive ratio of all on-line algorithms even for the case of one customer. They designed a best possible on-line algorithm for the case of one customer, which achieved the competitive ratio 2. The algorithm was modified to the multi-customers case (jobs of different customers did not share a batch and thus routing was not allowed) but the result was not good. Averbakh [2010] later extended the work to the case that the capacity of vehicles was limited. He considered several special cases and designed the corresponding on-line algorithms. For the one customer case the algorithms were on-line optimal but for multi-customers case the algorithm was not good. Another improved on-line algorithm for the multi-customers case was presented. The competitive ratio of this algorithm was $3 + \alpha$, where α is the ratio of

the largest processing time to the smallest processing time [Averbakh and Baysan, 2013a]. They also studied the problem in the semi-online environment, where partial information of future was known [Averbakh and Baysan, 2012, 2013b].

2.3.4 Algorithms for Class 3 - More than Two Stages

The globalization of production industry in the new century implied the integration of supply, production, transportation, inventory, loading, and so on. As such, more than two stages were involved and different kinds of costs and revenues were considered. Hall and Potts [2003] considered the combined problem of supply, production and transportation. Meanwhile, they stated that the integration could significantly reduce the cost comparing with the optimization of three single stages. The inventory cost was included into the objectives of supply chain scheduling model [Lee et al., 2003; Bertazzi et al., 2005; Sawik, 2009; Wang and Cheng, 2009b]. As the integration of different stages, the decisions of scheduling and transportation could be made simultaneously, which would reduce the inventory cost and improve the efficiency [Qi, 2005].

In the new decade, more and more studies have been conducted in this field. Yeung et al. [2011] considered the supply chain scheduling problem with dual delivery modes, which was modeled by a flow shop with time windows. As the transportation cost and the inventory cost were both involved, the problem was proved to be NP-hard. By exploring several structural properties, they developed optimal pseudo-polynomial algorithms. Later, the flow shop configuration was modified by assuming parallel machines at every stage and nonzero transportation times [Ullrich, 2012]. As the problem was strongly NP-hard

even for special case with zero transportation times, only a numerical study for small-size instances was conducted to analyze the performance of the work. Further, the objectives for classical scheduling were also explored for the new problems, such as the weighted flow time [Lee and Yoon, 2010], the total delivery time [Lee et al., 2012], and the total tardiness [Lee et al., 2013].

Furthermore, new problem structures and more complex objectives were introduced. Alonso-Ayuso et al. [2013] considered warehousing as the crucial role of the supply chain and wished to optimally organize the involved operations. The objective was to minimize the total loading time of vehicles. They used data from a real mattress warehouse to conduct a computational experiment to demonstrate the performance of the solution. Fan et al. [2010, 2013] studied the supply chain scheduling problem with heterogeneous vessels to minimize the total shortage, inventory and transportation cost. Celikbilek [2014] investigated the manufacturing scheduling and transportation mode in a cellular manufacturing where production should be completed in cells and transportation methods were limited by mode and capacity. A mixed integer mathematical model was constructed to maximize the total profit and a small size instance experiment was conducted to show the results. However, analytical algorithms could only be applied to very few problems in these models because of the complexity. Therefore, different techniques were adopted, such as intelligent algorithm [Hajiaghahi-Keshteli and Aminnayeri, 2014; Meinecke and Scholz-Reiter, 2014; Wang et al., 2014], Taguchi's method [Hajiaghahi-Keshteli et al., 2014], dynamic property with control theory [Li et al., 2001; Zhang, 2010; Ivanov et al., 2012; Inanov and Sokolov, 2012], and piecewise linear model [Baghalian et al., 2013].

2.3.5 Conclusion with Further Discussion

The supply chain scheduling is getting its popularity with the globalization of manufacturing industry and market, especially the advancement of transportation tools and communication techniques. This chapter focused on the strategies as well methods to model the supply chain scheduling problem. Three classes of problems were established and the related work, in particular in the recent five years on these problems, were reviewed.

There are a couple of future works in the area of supply chain scheduling. First, more attention should be paid to the supply chain scheduling problems in the on-line environment as they are in line with the real application situation and in a natural manner. In this thesis, more than half of the work is about the on-line supply chain scheduling problems, which is expected to significantly advance the research status of this field. Second, in the previous work of the supply chain scheduling problem in Class 2, the objective includes the time function involved with either all the jobs [Averbakh and Xue, 2007; Averbakh, 2010] or the whole procedure [Chen and Vairaktarakis, 2005] but never all the customers. While in this thesis, the objective of the total makespan and the total delivery cost will be considered, which is indeed a new problem or new feature of the problem in this area. Third, more different configurations of machines and customers and more different characteristics of constraints for supply chain scheduling should be studied. Three configurations are studied in this thesis (see the discussion in Section 1.3): (1) single-machine multi-customers, (2) multi-machines single-customer and (3) multi-machines multi-customers.

For every configurations, new features of the problems will be further defined in terms of the different release environments, processing patterns, vehicle characteristics and delivery patterns. Forth, the robustness and the resilience of algorithms for supply chain scheduling should be discussed as disturbances and damages always exist in real world problems. In this thesis, the robustness and the resilience of algorithms for the above new problems are explored, which is new, to the best of the author's knowledge, in algorithm development.

CHAPTER 3
PROBLEM ASSUMPTIONS AND NOTATIONS

3.1 Problem Descriptions

In a large-scale manufacturing and/or service operation, the customers demand products (jobs) and place orders to the manufacturer. The order placing of a job is also called job release. There are two different situations for jobs release: (1) all the information of the jobs is known beforehand and (2) the information of jobs release is not known until they are released. Scheduling algorithms for the two situations are completely different.

After knowing the jobs, or the jobs being released, the manufacturer need time to process them on machines. Therefore, when a job is released, the decision maker should decide when to process it and which machine to process it. The first two constraints on the processing of a job are: (1) the job is released and (2) there are free machines.

The manufacturer and the customers are at different locations which form a transportation network. Therefore, when a job is processed and completed, they should be delivered to the corresponding customers. For each completed job, the decision maker should decide when to deliver it, which vehicle to load it, and which path of the transportation network to travel through if there is more than one customer (Figure 3.1 shows that the manufacturer

and customers formulate a transportation network where the vehicles are delivering). The constraint on the delivery a job is that (1) the job has been completed and (2) there are available vehicles at that moment. In particular, the delivery is implemented on the transportation network, at which the road situation will be one constraint.

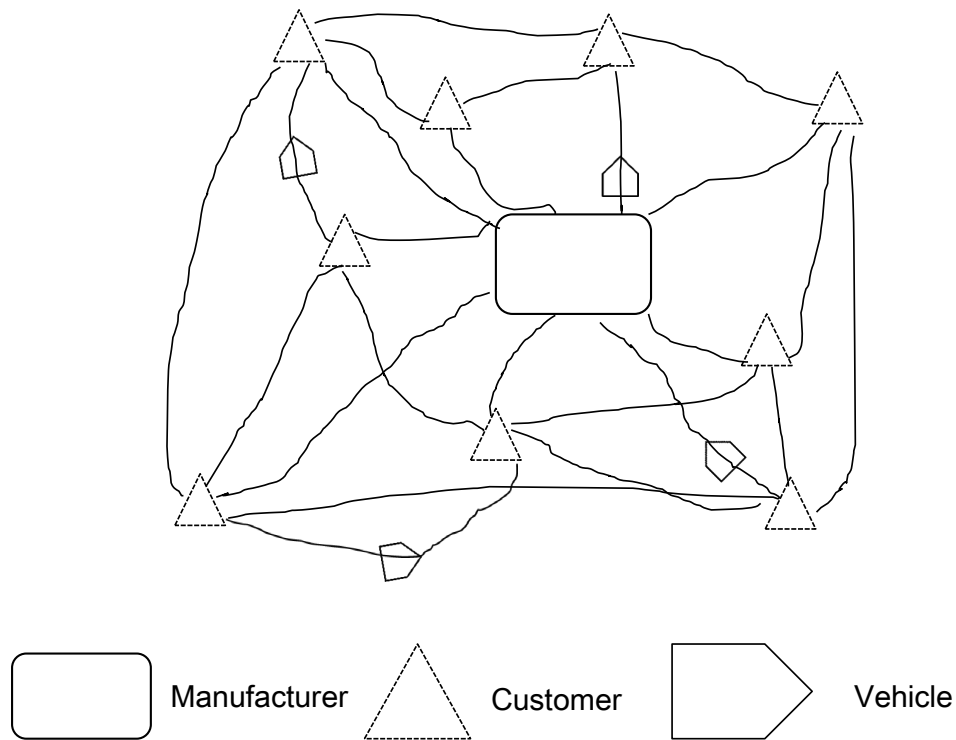


Figure 3.1. Network of The Manufacturer and Customers

The time when a job is released is called "release time" of the job. The time period that takes to process a job is called "processing time" of the job. The time when a job starts is called "starting time" of the job. The time when a job is completed is called "completion time" of the job. The time when a job leaves the manufacturer is called "departure time"

of the job. The time when a shipment returns to the manufacturer is called "return time". Thus, the time-based objective is defined as a function of release time, processing time, starting time, completion time, departure time and return time. Obviously, this objective should be minimized to make the whole process efficient.

When a vehicle delivers a job to its destination, there is a transportation cost for the shipment. Therefore, the total transportation cost arises for the whole delivery process, which is the cost-based objective. It is noted that the transportation cost gets larger as the increase of deliveries. For the efficiency of the whole schedule, this objective also should be minimized.

In this thesis, the above two kinds of objectives are considered, which conflict with each other. Minimization of the time-based objective implies a high frequency of vehicles transportation, which definitely causes additional transportation cost. On the other hand, minimization of the cost-based objective requires a high economize the utilization of vehicles, which will delay the delivery of jobs and result in a poor time performance. Therefore, a trade-off between the two objectives needs to be conducted.

In its very nature, the problem is a multi-objective problem. Different approaches can be used for the multi-objective problem. Taking two objectives as an example, there are approaches: (1) to minimize one first and then minimize the other; (2) to minimize one by taking the other as a constraint and vice versa; (3) to combine the objectives into a single objective and minimize their weighted sum. The three approaches share the common trait of subjectivity. For instance, in method (3), the user's choice of weights is subjective. In

method (2), the threshold that defines the constraint is subjective. Several problems considered in this thesis are in an on-line environment, which implies that future information of jobs is unknown. Therefore, it is difficult to set a threshold for the time-based function or the cost-based function. Therefore, this study explores method (3), i.e., to minimize the weighted sum of the objectives. From another point of view, the weights are the prices for the two groups of values with different unit. In this sense, the combination of two objectives into a single one is meaningful.

3.2 Problem Assumptions

As this thesis studies the problem of supply chain scheduling and develops algorithms to solve them, several assumptions need to be made such that the problems are tractable. It is noted that the assumptions are valid in the procedure of algorithms development but will be relaxed in the algorithms assessment and case study. In this way, the results of this thesis take into account of both theory and practice.

In this study, the following assumptions are made for algorithms development.

1. *There is only one manufacturer*

In practice, one manufacturer situation is often valid in the context of manufacturing. Multiple manufacturers should be the case in emergency situations.

2. *There is no constraint on the transportation network*

No constraint on the transportation network means that the capacity of the transportation network is assumed unlimited. This means that no competition on the transportation network happens among the customers and manufacturer. This assumption is not unrealistic in the modern city according to [Rainey and Andreas, 2015; Wang et al., 2013a]. In Chapter 7, to evaluate the resilience of the algorithms, the case that transportation network is disrupted is considered.

3. All parameters are deterministic

It is reasonable to assume that all the parameters of jobs are deterministic. Any uncertainty on parameters can be considered the so-called noise. The effect of the noise on the algorithm will be examined under the notion of robustness of the algorithm.

3.3 Problem Formulation and Notations

Suppose there are n_i jobs $J_1^{(i)}, \dots, J_{n_i}^{(i)}$ with the processing time $p_1^{(i)}, \dots, p_{n_i}^{(i)}$, released at the time $r_1^{(i)}, \dots, r_{n_i}^{(i)}$ from the i th customer ($i = 1, 2, \dots, k$), respectively, to the manufacturer which has machines to process them. After jobs are completed, they are loaded into batches or shipments and then transported to the customers by vehicles. There is a delivery cost for a batch.

Both off-line and on-line environments of jobs release are considered. In the off-line environment, information (release time, processing time and the number) of jobs is known beforehand. In the on-line environment, the information of future jobs is unknown beforehand until but their release time.

Both single-machine configuration and multi-machines configuration are considered. In the single-machine configuration, the manufacturer only has one machine. In the multi-machines configuration, the manufacturer has multiple parallel machines which have the same function.

Two different delivery patterns are considered. In the first delivery pattern, the jobs of different customers do not share a batch which means that all jobs need to be delivered to the corresponding customer directly. In the second delivery pattern, the jobs of different customers share a batch, which means that a routing path is needed to deliver a batch. The routing path in this case means that there must be several options of delivery with respect to different customers (e.g., customer 1 goes first and then customer 2, etc.) .

The goal of the problem is to minimize both time-based objective and cost-based objective. For every customer, its own time-based objective is the time that the manufacturer operates for it, which is the time that the delivery vehicle for its last job is back to the manufacturer, i.e., makespan. Therefore, the time-based objective for all customers is the total makespans. The cost-based objective is the total delivery cost, which is the number of batches multiplied by the cost of one batch delivery (noticing: the cost of one batch delivery is constant but the number of the batches is a variable). As mentioned before, to minimize the weighted sum of the two objectives is the method taken in this thesis to deal with the two objectives problem.

The solution of the problem is a schedule which should specify when a job is processed, which machine a job is processed on, which batch a job is loaded in, when a batch is

transported, and which path a batch is transported through.

The following notations are listed to describe the problem:

- η : a feasible schedule.
- opt : an optimal off-line schedule.
- k : the number of customers.
- K : the set of all customers, $\{1, 2, \dots, k\}$
- m : the number of machines.
- $J_j^{(i)}$: the j th job for the i th customer, where $i = 1, 2, \dots, k$.
- $\mathcal{J}^{(i)}$: the set of all jobs for the i th customer.
- $\mathcal{J}_{\leq t}^{(i)}$: the set of all jobs for the i th customer released before t .
- n_i : the number of all jobs for the i th customer.
- n : the number of all jobs, $n = \sum_{i=1}^k n_i$.
- $r_j^{(i)}$: the release time of job $J_j^{(i)}$.
- *on – line*: the jobs are released in the on-line environment.
- $p_j^{(i)}$: the processing time for job $J_j^{(i)}$
- $P^{(i)}$: the sum of the processing time of all the jobs for the i th customer.
- $P_{[t_1, t_2]}^{(i)}(\eta)$: the sum of the processing time of the jobs for the i th customer which are processed in the interval $[t_1, t_2]$ in the schedule η .
- P : the sum of the processing time of all the jobs.
- *pmtn*: the processing of jobs can be interrupted and restarted later.
- $C_j^{(i)}(\eta)$: the completion time of job $J_j^{(i)}$ in the schedule η .
- $C_{max}^{(i)}(\eta)$: $\max_{J_j^{(i)} \in \mathcal{J}^{(i)}} C_j^{(i)}(\eta)$.

- idle time: when there are free machines and there are no uncompleted jobs.
- waiting time: when there are free machines but there are uncompleted jobs.
- delay: there is waiting time in the schedule.
- block: a time interval that a machine is not free.
- $C_{max}(U, m, \eta)$: the completion time for jobs set U being processed on m machines in schedule η .
- $C_{max}(U)$: the optimal maximum completion time for jobs set U being processed on a single machine, which can be found by scheduling all the jobs on the machine without delay.
- $\rho_j^{(i)}(\eta)$: the departure time of job $J_j^{(i)}$ in the schedule η .
- $\rho_{max}^{(i)}(\eta)$: $\max_{J_j^{(i)} \in \mathcal{J}^{(i)}} \rho_j^{(i)}(\eta)$, the latest time of delivery of jobs in η .
- T_{cd} : the transportation time between place c and place d , where $c, d = 0, 1, 2, \dots, k$ ('0' represents the manufacturer, '1, 2, \dots , k ' represent the customers).
- *direct*: the jobs of different customers do not share a batch, which means that all jobs need to be delivered to the corresponding customer directly.
- *routing*: the jobs of different customers share a batch, which means that a routing path is needed to deliver a batch.

$V(x, y)$: there are x vehicles available, each with a capacity y , where $x \in \{1, \infty\}$ and $y \in \{C, \infty\}$ (the symbol " ∞ " means "enough" in the engineering sense).

- $D_j^{(i)}(\eta)$: the return time of the vehicle which delivers the job $J_j^{(i)}$ in the schedule η .
- $D_{max}^{(i)}(\eta)$: $\max_{J_j^{(i)} \in \mathcal{J}^{(i)}} D_j^{(i)}(\eta)$, makespan of the i th customer in the schedule η .
- $\sum_{i=1}^k D_{max}^{(i)}(\eta)$: the total makespans in the schedule η .
- D : the cost of one delivery, which is a constant.

- $TC(\eta)$: the total cost of all the deliveries in the schedule η , which is the number of deliveries timing by D .
- $Z(\eta)$: the weighted sum of two objectives in the schedule η , $w_1 \sum_{i=1}^k D_{max}^{(i)}(\eta) + w_2 TC(\eta)$, where w_1 and w_2 are two weights.

This thesis assumes that the cost for each batch is the same. Therefore,

$$w_1 \sum_{i=1}^k D_{max}^{(i)} + w_2 TC = w_1 \sum_{i=1}^k D_{max}^{(i)} + w_2 D \times z = w_1 \left(\sum_{i=1}^k D_{max}^{(i)} + \frac{w_2}{w_1} D \times z \right), \quad (3.1)$$

where w_1 and w_2 are the weights and z is the number of batches. Thus, minimizing $w_1 \sum_{i=1}^k D_{max}^{(i)} + w_2 TC$ is equivalent to minimizing $\sum_{i=1}^k D_{max}^{(i)} + \frac{w_2}{w_1} D \times z$. The unit delivery cost is D . Let $D' = \frac{w_2}{w_1} D$ and let TC' be the total delivery cost. Then the problem is equivalent to minimizing $\sum_{i=1}^k D_{max}^{(i)} + TC'$. Without loss of generality and without confusion, in the remainder of this thesis the objective function is $\sum_{i=1}^k D_{max}^{(i)} + TC$ (TC replaces TC'). The final objective in the schedule η is $Z(\eta) = \sum_{i=1}^k D_{max}^{(i)}(\eta) + TC(\eta)$.

In this thesis, problems are considered in terms of different configurations of machines and customers. Three types of problems are considered: (1) there is one machine in the manufacturer and there are multiple customers, (2) there are multiple (parallel) machines in the manufacturer and there is one customer, and (3) there are multiple parallel machines in the manufacturer and there multiple customers. For each type, specific problems with different characteristics that describe for example the vehicle capacity are defined. The following are all the specific problems derived from the three general problems along with their five field notation representation [Chen, 2010].

(1) Single-machine Multi-customers Problem (SMP)

Eight specific problems are defined:

$$\text{SMP1: } 1|r_j, \textit{pmtn}|V(\infty, \infty), \textit{direct}|k| \sum D_{max}^{(i)} + TC$$

Jobs are released off-line, processed in "pmtn" pattern and delivered in "direct" pattern. The capacity of vehicles and the number of vehicles are both enough.

$$\text{SMP2: } 1|r_j, \textit{pmtn}, \textit{on - line}|V(\infty, \infty), \textit{direct}|k| \sum D_{max}^{(i)} + TC$$

Jobs are released on-line, processed in "pmtn" pattern and delivered in "direct" pattern. The capacity of vehicles and the number of vehicles are both enough.

$$\text{SMP3: } 1|r_j|V(\infty, \infty), \textit{direct}|k| \sum D_{max}^{(i)} + TC$$

Jobs are released off-line and delivered in "direct" pattern. The capacity of vehicles and the number of vehicles are both enough.

$$\text{SMP4: } 1|r_j, \textit{on - line}|V(\infty, \infty), \textit{direct}|k| \sum D_{max}^{(i)} + TC$$

Jobs are released on-line and delivered in "direct" pattern. The capacity of vehicles and the number of vehicles are both enough.

$$\text{SMP5: } 1|r_j, \textit{pmtn}|V(\infty, \infty), \textit{routing}|k| \sum D_{max}^{(i)} + TC$$

Jobs are released off-line, processed in "pmtn" pattern and delivered in "routing" pattern. The capacity of vehicles and the number of vehicles are both enough.

$$\text{SMP6: } 1|r_j, \textit{pmtn}, \textit{on - line}|V(\infty, \infty), \textit{routing}|k| \sum D_{max}^{(i)} + TC$$

Jobs are released on-line, processed in "pmtn" pattern and delivered in "routing" pattern. The capacity of vehicles and the number of vehicles are both enough.

$$\text{SMP7: } 1|r_j|V(\infty, \infty), \textit{routing}|k| \sum D_{max}^{(i)} + TC$$

Jobs are released off-line and delivered in "routing" pattern. The capacity of vehicles and the number of vehicles are both enough.

$$\text{SMP8: } 1|r_j, \textit{on - line}|V(\infty, \infty), \textit{routing}|k| \sum D_{max}^{(i)} + TC$$

Jobs are released on-line and delivered in "routing" pattern. The capacity of vehicles and the number of vehicles are both enough.

(2) Multi-machines Single-customer Problem (MSP)

Five specific problems are defined (as there is one customer, $D_{max}^{(1)}$ is written as D_{max} for short):

$$\text{MSP1: } Pm|r_j, on - line|V(1, \infty), direct|1|D_{max} + TC$$

Jobs are released on-line and delivered in "direct" pattern. The capacity of vehicles is enough but the number of vehicles is one.

$$\text{MSP2: } Pm|r_j, on - line|V(\infty, \infty), direct|1|D_{max} + TC$$

Jobs are released on-line and delivered in "direct" pattern. The capacity of vehicles and the number of vehicles are both enough.

$$\text{MSP3: } Pm|r_j, pmtn, on - line|V(1, C), direct|1|D_{max} + TC$$

Jobs are released on-line, processed in "pmtn" pattern and delivered in "direct" pattern. The capacity of vehicles is C and the number of vehicles is one.

$$\text{MSP4: } Pm|r_j, on - line|V(1, C), direct|1|D_{max} + TC$$

Jobs are released on-line and delivered in "direct" pattern. The capacity of vehicles is

C and the number of vehicles is one.

$$\text{MSP5: } Pm|r_j, on - line|V(\infty, C), direct|1|D_{max} + TC$$

Jobs are released on-line and delivered in "direct" pattern. The capacity of vehicles is C and the number of vehicles is enough.

(3) Multi-machines Multi-customers Problem (MMP)

Eight sepecific problems are defined:

$$\text{MMP1: } Pm|r_j, pmtn|V(\infty, \infty), direct|k|\sum D_{max}^{(i)} + TC$$

Jobs are released off-line, processed in "pmtn" pattern and delivered in "direct" pattern. The capacity of vehicles and the number of vehicles are both enough.

$$\text{MMP2: } Pm|r_j, pmtn, on - line|V(\infty, \infty), direct|k|\sum D_{max}^{(i)} + TC$$

Jobs are released on-line, processed in "pmtn" pattern and delivered in "direct" pattern. The capacity of vehicles and the number of vehicles are both enough.

$$\text{MMP3: } Pm|r_j|V(\infty, \infty), direct|k|\sum D_{max}^{(i)} + TC$$

Jobs are released off-line and delivered in "direct" pattern. The capacity of vehicles and the number of vehicles are both enough.

$$\text{MMP4: } Pm|r_j, on - line|V(\infty, \infty), direct|k| \sum D_{max}^{(i)} + TC$$

Jobs are released on-line and delivered in "direct" pattern. The capacity of vehicles and the number of vehicles are both enough.

$$\text{MMP5: } Pm|r_j, pmtn|V(\infty, \infty), routing|k| \sum D_{max}^{(i)} + TC$$

Jobs are released off-line, processed in "pmtn" pattern and delivered in "routing" pattern. The capacity of vehicles and the number of vehicles are both enough.

$$\text{MMP6: } Pm|r_j, pmtn, on - line|V(\infty, \infty), routing|k| \sum D_{max}^{(i)} + TC$$

Jobs are released on-line, processed in "pmtn" pattern and delivered in "routing" pattern. The capacity of vehicles and the number of vehicles are both enough.

$$\text{MMP7: } Pm|r_j|V(\infty, \infty), routing|k| \sum D_{max}^{(i)} + TC$$

Jobs are released off-line and delivered in "routing" pattern. The capacity of vehicles and the number of vehicles are both enough.

MMP8: $Pm|r_j, on - line|V(\infty, \infty), routing|k| \sum D_{max}^{(i)} + TC$

Jobs are released on-line and delivered in "routing" pattern. The capacity of vehicles and the number of vehicles are both enough.

CHAPTER 4
ALGORITHMS FOR SINGLE-MACHINE AND MULTI-CUSTOMERS PROBLEMS

In this chapter, the problems for single-machine and multi-customers are considered and corresponding algorithms are developed. As described in Section 3.3, eight problems are defined in terms of different release environments, preparation patterns and delivery patterns. For all these problems, the corresponding algorithms are developed, their analysis is proposed, and the simulation is also conducted to give an idea of the effectiveness of the algorithms.

4.1 Algorithm for Problem SMP1

SMP1 has the following features: Jobs are released off-line, processed in "pmtn" pattern and delivered in "direct" pattern. The capacity of vehicles and the number of vehicles are both enough.

As the information of jobs is known beforehand, all the jobs of the same customer should be delivered in one batch when they are all completed. Therefore, there are k batches in the optimal schedule and $D_{max}^{(i)}(opt) = \rho_{max}^{(i)}(opt) + 2T_{0i} = C_{max}^{(i)}(opt) + 2T_{0i}$ for $i = 1, 2, \dots, k$, which implies that $Z(opt) = \sum_{i=1}^k D_{max}^{(i)}(opt) + TC(opt) = \sum_{i=1}^k C_{max}^{(i)}(opt) + 2\sum_{i=1}^k T_{0i} + kD$. Then, this problem is equivalent with the agent scheduling problem

$1|r_j, pmtn| \sum C_{max}^{(i)}$. This problem to two customers has been discussed in [Ding and Sun, 2010]. However, the result for the case there are more than two customers is not known. This section will solve the problem for general case. Therefore, in the following discussion, it is assumed that $T_{0i} = 0$ for all i and $D = 0$.

Meanwhile, because job processing can be interrupted and resumed later, any uncompleted job can be chosen to process. In this situation, it is only needed to determine which customer's jobs needs to be processed. Therefore, the mechanism to choose a customer need to be developed. Such a mechanism can be based on the concept of the priority of customers, which is defined as follows.

Definition 4.1. *A priority of customers is a permutation (i_1, i_2, \dots, i_k) of the customer set $K = \{1, 2, \dots, k\}$.*

Next, a schedule can be constructed based on the priority of customers in the following way: at every time, the customer with the highest priority can occupy the machine. Such a schedule called priority schedule.

Lemma 4.1. *There exists an optimal schedule for SMP1 which is a priority schedule and the priority of the customers is consistent with the order of the customers' completion time in the schedule.*

Proof of Lemma 4.1: Without loss of generality, suppose the order of the customers' completion time is $1, 2, \dots, k$ in an optimal schedule opt , that is, $C_{max}^{(1)}(opt) \leq C_{max}^{(2)}(opt) \leq \dots \leq C_{max}^{(k)}(opt)$. The priority of customers is set as $1, 2, \dots, k$, and at every time the machine processes the job of the customer with the highest priority, which generates the

priority schedule opt' .

$$\begin{aligned}
C_{max}^{(1)}(opt') &= C_{max}(\mathcal{J}^{(1)}) \leq C_{max}^{(1)}(opt) \\
C_{max}^{(2)}(opt') &= C_{max}\left(\bigcup_{i=1}^2 \mathcal{J}^{(i)}\right) \leq C_{max}^{(2)}(opt) \\
&\dots \\
C_{max}^{(k)}(opt') &= C_{max}\left(\bigcup_{i=1}^k \mathcal{J}^{(i)}\right) \leq C_{max}^{(k)}(opt)
\end{aligned} \tag{4.1}$$

As the optimality of opt , it should be satisfied that $C_{max}^{(i)}(opt') = C_{max}^{(i)}(opt)$ for $i = 1, 2, \dots, k$ and thus opt' is also an optimal schedule, which completes the proof. \square

When $k = 2$, the priority can be found by comparing $C_{max}(\mathcal{J}^{(1)})$ and $C_{max}(\mathcal{J}^{(2)})$, which then solves the problem [Ding and Sun, 2010]. However, the method cannot be extended to a larger k . For the following instance with $k = 3$: Customer 1 has two jobs $J_1^{(1)} = (0, 1)$ and $J_2^{(1)} = (2, 1)$, Customer 2 has two jobs $J_1^{(2)} = (0, 1)$ and $J_2^{(2)} = (2, 1)$, and Customer 3 has one job $J_1^{(3)} = (0, 2.6)$. The optimal schedule will process the jobs with the priority of the customers $(1, 2, 3)$ while $C_{max}(\mathcal{J}^{(1)}) = C_{max}(\mathcal{J}^{(2)}) > C_{max}(\mathcal{J}^{(3)})$. Therefore, more properties need to be explored.

Lemma 4.2. *The optimal schedule for SMP1 that satisfies Lemma 4.1 has sub-optimality, that is, if (i_1, i_2, \dots, i_k) is the optimal priority for the customer set $\{1, 2, \dots, k\}$, then (i_1, i_2, \dots, i_h) is an optimal priority for the customer set $\{i_1, i_2, \dots, i_h\}$ for all $h \leq k$.*

Proof of Lemma 4.2: Suppose there exists a h_0 such that $(i_1, i_2, \dots, i_{h_0})$ is not the optimal priority of the customer set $\{i_1, i_2, \dots, i_{h_0}\}$ but $(i'_1, i'_2, \dots, i'_{h_0})$ is. Let the schedule generated by the priority $(i_1, i_2, \dots, i_{h_0}, i_{h_0+1}, \dots, i_k)$ be η while the schedule generated

by $(i'_1, i'_2, \dots, i'_{h_0}, i_{h_0+1}, \dots, i_k)$ be η' .

$$\sum_{l=1}^k C_{max}^{(i_l)}(\eta) = \sum_{l=1}^{h_0} C_{max}^{(i_l)}(\eta) + \sum_{l=h_0+1}^k C_{max}^{(i_l)}(\eta) \quad (4.2)$$

For the customers i_l ($l = h_0 + 1, \dots, k$), their jobs processing are the same in η and η' , which implies $C_{max}^{(i_l)}(\eta) = C_{max}^{(i_l)}(\eta')$ for $l = h_0 + 1, \dots, k$. In addition, as $(i_1, i_2, \dots, i_{h_0})$ is not the optimal priority of the customer set $\{i_1, i_2, \dots, i_{h_0}\}$ but $(i'_1, i'_2, \dots, i'_{h_0})$, there is $\sum_{l=1}^{h_0} C_{max}^{(i_l)}(\eta) > \sum_{l=1}^{h_0} C_{max}^{(i'_l)}(\eta')$.

$$\sum_{l=1}^k C_{max}^{(i_l)}(\eta) > \sum_{l=1}^{h_0} C_{max}^{(i'_l)}(\eta') + \sum_{l=h_0+1}^k C_{max}^{(i_l)}(\eta') = \sum_{l=1}^k C_{max}^{(i'_l)}(\eta') \quad (4.3)$$

This contradicts with the assumption that η is optimal. Therefore, no such h_0 exists, which proves the lemma. \square

Based on the above two lemmas, a dynamic programming for this problem is proposed as follows (dynamic programming is a recursive method to solve a complex problem by dividing it into simpler subproblems with an optimal substructure [Sniedovich, 2010]).

Algorithm SMH1

Value function:

$F(A)$ = the minimum total cost for the jobs of the customers in set A .

$f(A)$ = the customer with the least priority in a schedule achieving $F(A)$.

Initial conditions:

$F(\emptyset) = 0$.

Recursive relation:

For $A \subseteq \{1, 2, \dots, k\}$, $F(A) = \min\{F(A \setminus \{i\}) + C_{max}(\cup_{l \in A} \mathcal{J}^{(l)}) | i \in A\}$; $f(A) =$

$\operatorname{argmin}\{F(A \setminus \{i\}) + C_{\max}(\cup_{l \in A} \mathcal{J}^{(l)}) \mid i \in A\}$. Ties can be broken by choosing the largest index.

Optimal solution value:

$$F(\{1, 2, \dots, k\}).$$

Theorem 4.1. *The problem SMP1 can be solved by the algorithm SMH1 in the time $O(nk2^k)$.*

Proof of Theorem 4.1: It needs to prove that the function $F(A)$ can find the optimal value for the customer set A . Induction for $|A|$ is applied. The statement is obviously true when $|A| = 0$ and suppose that it is also true for the case that $|A| = h$ ($h \geq 1$). Then when $|A| = h + 1$, for any $i \in A$, the following two cases are discussed.

Case 1: There exists an optimal solution opt such that Customer i has the least priority.

From the hypothesis, $F(A \setminus \{i\})$ can find the optimal value for the customer set $A \setminus \{i\}$.

By Lemma 4.2,

$$F(A \setminus \{i\}) + C_{\max}(\cup_{l \in A} \mathcal{J}^{(l)}) = \sum_{l \in A \setminus \{i\}} C_{\max}^{(l)}(opt) + C_{\max}^{(i)}(opt) = \sum_{l \in A} C_{\max}^{(l)}(opt) \quad (4.4)$$

Case 2: There is no optimal solution such that Customer i has the least priority.

Let i_1, i_2, \dots, i_h be the optimal priority for $A \setminus \{i\}$. Then the schedule η generated by the priority $(i_1, i_2, \dots, i_h, i)$ cannot be optimal for A . Therefore,

$$\begin{aligned} F(A \setminus \{i\}) + C_{\max}(\cup_{l \in A} \mathcal{J}^{(l)}) &\geq \sum_{l \in A \setminus \{i\}} C_{\max}^{(l)}(\eta) + C_{\max}^{(i)}(\eta) \\ &= \sum_{l \in A} C_{\max}^{(l)}(\eta) > \sum_{l \in A} C_{\max}^{(l)}(opt). \end{aligned} \quad (4.5)$$

Combining along with their discussion, it holds: $F(A) = \min\{F(A \setminus \{i\}) + C_{max}(\cup_{l \in A} \mathcal{J}^{(l)}) \mid i \in A\} = \sum_{i \in A} C_{max}^{(i)}(opt)$. Meanwhile, $f(A)$ can find the customer with the least priority in the optimal solution.

$F(K)$ can find the optimal value, and the optimal priority (i_1, i_2, \dots, i_k) can be determined as follows: $i_k = f(K)$ and for $l = k - 1, \dots, 1$, $i_l = f(K \setminus \{i_k, \dots, i_{l+1}\})$, where $K = \{1, 2, \dots, k\}$.

For the set A with i elements, the computation time of $F(A)$ is $O(ni)$. As the number of such sets is C_k^i for $i = 1, 2, \dots, k$, the total computation time is $O(\sum_{i=1}^k niC_k^i) = O(nk2^k)$.

□

Note that when k is a parameter, this algorithm is exponential. For a large k , a simulated annealing algorithm SA_SMH1 based on Lemma 4.1 is proposed.

Algorithm SA_SMH1

Initialize a priority list: (i_1, i_2, \dots, i_k) . Let η be the schedule generated by this priority.

Initialize the temperature loop parameter T and the internal loop parameter TT .

Temperature loop: T exponentially decreases to 1

Internal loop: from 1 to TT

Randomly choose two sequential customers from the list: i_a and i_{a+1} . Let η' be the schedule generated by $(i_1, i_2, \dots, i_{a-1}, i_{a+1}, i_a, \dots, i_k)$. Let Δ be the difference between the objective values of η and η' . If $\Delta < 0$, accept the new schedule and update the schedule and the priority list; otherwise, accept the new schedule with the probability $e^{-\frac{\Delta}{T}}$.

In the application, the parameter T is set to be P while the parameter TT is set to be n^2 , where P is the sum of the processing time of all the jobs and n is the number of all the jobs. The performance of SA_SMH1 is shown in a simulated experiment (or simulation for short) which will be presented later.

4.2 Algorithm for Problem SMP2

SMP2 has the following features: Jobs are released on-line, processed in "pmtn" pattern and delivered in "direct" pattern. The capacity of vehicles and the number of vehicles are both enough. Actually, SMP2 is the on-line version of SMP1. When there is only one customer, the lower bound of the problem is 2 [Han, 2012]. The lower bound can be constructed similarly (see Appendix). Therefore, the lower bound of SMP2 is at least 2.

Corollary 4.1. *No on-line algorithm for SMP2 can have competitive ratio less than 2, even all processing times being 0.*

Note that the lower bound construction releases a job for Customer i every D period (see Appendix). The main idea to develop an on-line algorithm is to deal with such a situation. Meanwhile, the completion time of jobs for each customer is an important index to solve the competition among customers. In the following, an on-line algorithm is proposed for SMP2.

Algorithm SMH2

At the time t that a new job arrives, the customers are re-indexed in an increasing order of $C_{max}(\mathcal{J}_{\leq t}^{(i)})$ (If there is more than one customer with the same $C_{max}(\mathcal{J}_{\leq t}^{(i)})$, their order is the original index order). When a new job arrives or the machine is free, process

available jobs of the customer with the highest on-line priority.

At the time of lD where $l \geq 1$ and l is integer, if there is no uncompleted job for Customer i , then there must be a batch to deliver all the completed jobs of Customer i ; otherwise, there is no operation for these jobs.

On-line algorithm SMH2 for SMP2 can achieve a good result both cases that $k = 2$ and $k = 3$.

Theorem 4.2. *The competitive ratio of on-line algorithm SMH2 for SMP2 with $k = 2$ is 2, which is on-line optimal.*

Proof of Theorem 4.2: Let η be the schedule obtained by the algorithm SMH2. Suppose that the jobs of Customer 1 are completed earlier in opt , which implies that $C_{max}(\mathcal{J}^{(1)}) = C_{max}^{(1)}(opt) \leq C_{max}(\mathcal{J}^{(2)})$. In η , at the time of $\max\{r_j^{(i)} | J_j^{(i)} \in \mathcal{J}^{(i)}, i = 1, 2\}$, all the jobs are released and there are uncompleted jobs for Customer 2. By the algorithm SMH2, the jobs of Customer 1 will have the top priority at this moment. Therefore, the jobs of Customer 1 are also completed earlier in η .

Suppose that l_1D is the last idle delivery point (l_1D is idle time) before $C_{max}^{(1)}(opt)$ in η (if there is no such l_1D , let $l_1 = 0$), where l_1 is a non-negative integer. For the interval $(l_1D, C_{max}^{(1)}(\eta)]$, the schedule can be modified such that the jobs of Customer 2 has a higher priority. According to the algorithm, the completion time of Customer 1 will not be changed, and the processing of Customer 2's jobs in this interval will be before $C_{max}^{(1)}(opt)$. Meanwhile, as all the jobs processed in this interval are released after l_1D , $P_{(l_1D, C_{max}^{(1)}(\eta))}^{(1)}(\eta) \leq C_{max}^{(1)}(opt) - l_1D$. Therefore, in the modified schedule, the machine will

only process Customer 1's jobs in the interval $(C_{max}^{(1)}(opt), C_{max}^{(1)}(\eta)]$.

$$\begin{aligned}
C_{max}^{(1)}(\eta) &\leq C_{max}^{(1)}(opt) + P_{[l_1D, C_{max}^{(1)}(\eta)]}^{(1)}(\eta) \leq 2C_{max}^{(1)}(opt) - l_1D \\
C_{max}^{(2)}(\eta) &= C_{max}^{(2)}(opt) \\
\rho_{max}^{(1)}(\eta) &= \lceil \frac{C_{max}^{(1)}(\eta)}{D} \rceil D \leq C_{max}^{(1)}(\eta) + D \leq 2C_{max}^{(1)}(opt) - l_1D + D \\
\rho_{max}^{(2)}(\eta) &= \lceil \frac{C_{max}^{(2)}(\eta)}{D} \rceil D \leq C_{max}^{(2)}(\eta) + D = C_{max}^{(2)}(opt) + D
\end{aligned} \tag{4.6}$$

The delivery cost in η can be analyzed as follows.

In the interval $(0, l_1D]$, there are at most two batches at every delivery point, so the delivery cost will not be more than $2l_1D$.

In the interval $(l_1D, C_{max}^{(1)}(opt)]$, there is at most one batch at every delivery point, so the delivery cost will not be more than $\lfloor \frac{C_{max}^{(1)}(opt)}{D} \rfloor D - l_1D$.

In the interval $(C_{max}^{(1)}(opt), \rho_{max}^{(2)}(\eta)]$, there is at most one batch for Customer 2 at every deliver point, and there is one batch in total for Customer 1, so the delivery cost will not be more than $\rho_{max}^{(2)}(\eta) - \lfloor \frac{C_{max}^{(1)}(opt)}{D} \rfloor D + D$.

$$\begin{aligned}
TC(\eta) &\leq 2l_1D + \rho_{max}^{(2)}(\eta) - l_1D + D = \rho_{max}^{(2)}(\eta) + l_1D + D \\
&\leq C_{max}^{(2)}(opt) + l_1D + 2D
\end{aligned} \tag{4.7}$$

Therefore,

$$\begin{aligned}
\frac{Z(\eta)}{Z(\text{opt})} &= \frac{D_{\max}^{(1)}(\eta) + D_{\max}^{(2)}(\eta) + TC(\eta)}{D_{\max}^{(1)}(\text{opt}) + D_{\max}^{(2)}(\text{opt}) + TC(\text{opt})} \\
&= \frac{\rho_{\max}^{(1)}(\eta) + 2T_{01} + \rho_{\max}^{(2)}(\eta) + 2T_{02} + TC(\eta)}{C_{\max}^{(1)}(\text{opt}) + 2T_{01} + C_{\max}^{(2)}(\text{opt}) + 2T_{02} + 2D} \\
&\leq \frac{\rho_{\max}^{(1)}(\eta) + \rho_{\max}^{(2)}(\eta) + TC(\eta)}{C_{\max}^{(1)}(\text{opt}) + C_{\max}^{(2)}(\text{opt}) + 2D} \\
&\leq \frac{2C_{\max}^{(1)}(\text{opt}) - l_1D + D + C_{\max}^{(2)}(\text{opt}) + D + C_{\max}^{(2)}(\text{opt}) + l_1D + 2D}{C_{\max}^{(1)}(\text{opt}) + C_{\max}^{(2)}(\text{opt}) + 2D} \\
&= \frac{2C_{\max}^{(1)}(\text{opt}) + 2C_{\max}^{(2)}(\text{opt}) + 4D}{C_{\max}^{(1)}(\text{opt}) + C_{\max}^{(2)}(\text{opt}) + 2D} \\
&= 2
\end{aligned} \tag{4.8}$$

From Corollary 4.1, there is no on-line algorithm with competitive ratio less than 2, which completes the proof. \square

Theorem 4.3. *The competitive ratio of on-line algorithm SMH2 for SMP2 with $k = 3$ is $2 + \frac{2}{27}$.*

Let η be the algorithm schedule, $(1, 2, 3)$ be the order of customer' completion times in η , and (i_1, i_2, i_3) be the order of customer' completion times in opt . Suppose that the last idle delivery point before $C_{\max}^{(1)}(\eta)$ is l_1D (l_1 is a non-negative integer). From the proof of the Theorem 4.2, the transportation time will not affect the result, so in the following it is assumed that $T_{0i} = 0$ for $i = 1, 2, 3$. The proof of this theorem is completed by proving the following three lemmas.

Lemma 4.3. *If there is idle delivery point between $C_{\max}^{(1)}(\eta)$ and $C_{\max}^{(2)}(\eta)$, the algorithm is $2 + \frac{1}{22}$ competitive.*

Proof of Lemma 4.3: Suppose that the last idle delivery point in $(C_{\max}^{(1)}(\eta), C_{\max}^{(2)}(\eta)]$ be l_2D (l_2 is a positive integer and $l_2 > l_1$). The preparation after l_2D is equivalent to the

case $k = 2$, so $(i_1, i_2, i_3) = (1, 2, 3)$ from Theorem 4.2. Figure 4.1 shows the jobs processing in η under this situation.

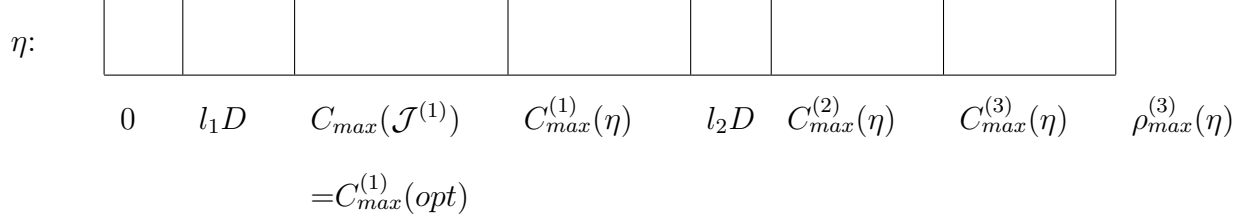


Figure 4.1. The Jobs Processing in η (1)

In the interval $(0, l_1D]$, there are at most three batches for every delivery point, so the deliver cost will not be more than $3l_1D$.

In the interval $(l_1D, C_{max}^{(1)}(opt)]$, there are at most two batches for every delivery point, so the deliver cost will not be more than $2(\lfloor \frac{C_{max}^{(1)}(opt)}{D} \rfloor - l_1)D$.

In the interval $(C_{max}^{(1)}(opt), C_{max}^{(1)}(\eta)]$, all the jobs of Customer 1 are released, so the jobs of Customer 2 and Customer 3 processed in this interval must be released before $C_{max}^{(1)}(opt)$, which implies that there are at most two batches in total for these two customers. Suppose that the delivery cost in this interval is sD , where s is a non-negative integer not greater than 2.

In the interval $(C_{max}^{(1)}(\eta), l_2D]$, there is at most two batch for every delivery point for Customer 2 and Customer 3, and there is one batch in total for Customer 1, so the delivery cost will not be more than $2(l_2 - \lfloor \frac{C_{max}^{(1)}(\eta)}{D} \rfloor)D + D$.

In the interval $(l_2D, \rho_{max}^{(3)}(\eta)]$, there is at most one batch for every delivery point except $\lceil \frac{C_{max}^{(2)}(\eta)}{D} \rceil D$, and there is at most two batches at $\lceil \frac{C_{max}^{(2)}(\eta)}{D} \rceil D$, so the delivery cost will not

be more than $(\lceil \frac{C_{max}^{(3)}(\eta)}{D} \rceil - l_2)D + D$.

$$\begin{aligned}
TC(\eta) &\leq 3l_1D + 2(\lfloor \frac{C_{max}^{(1)}(opt)}{D} \rfloor - l_1)D + sD + 2(l_2 - \lfloor \frac{C_{max}^{(1)}(\eta)}{D} \rfloor)D + D + \\
&\quad (\lceil \frac{C_{max}^{(2)}(\eta)}{D} \rceil - l_2)D + D \\
&\leq l_1D + 2C_{max}^{(1)}(opt) - 2(C_{max}^{(1)}(opt) - \lfloor \frac{C_{max}^{(1)}(opt)}{D} \rfloor D) + l_2D - \\
&\quad 2\lfloor \frac{C_{max}^{(1)}(\eta)}{D} \rfloor D + C_{max}^{(3)}(\eta) + (s+3)D
\end{aligned} \tag{4.9}$$

Therefore,

$$\begin{aligned}
Z(\eta) &= \lceil \frac{C_{max}^{(1)}(\eta)}{D} \rceil D + \lceil \frac{C_{max}^{(2)}(\eta)}{D} \rceil D + \lceil \frac{C_{max}^{(3)}(\eta)}{D} \rceil D + TC(\eta) \\
&\leq 2C_{max}^{(1)}(opt) + 2C_{max}^{(3)}(\eta) + (s+6)D + l_1D + l_2D + C_{max}^{(2)}(\eta) - \\
&\quad \lfloor \frac{C_{max}^{(1)}(\eta)}{D} \rfloor D - 2(C_{max}^{(1)}(opt) - \lfloor \frac{C_{max}^{(1)}(opt)}{D} \rfloor D)
\end{aligned} \tag{4.10}$$

For the jobs processing in the interval $(l_2D, C_{max}^{(2)}(\eta)]$, the similar way in the Theorem 4.2 can be applied to show $C_{max}^{(2)}(\eta) \leq l_2D + 2(C_{max}^{(2)}(opt) - l_2D)$, which implies that $l_2D + C_{max}^{(2)}(\eta) \leq 2C_{max}^{(2)}(opt)$.

Case 1: $C_{max}^{(1)}(\eta) \geq l_1D + D$. In this case, $\lfloor \frac{C_{max}^{(1)}(\eta)}{D} \rfloor \geq l_1 + 1$. Meanwhile, as $l_2D > C_{max}^{(1)}(\eta)$, it should be satisfied that $l_2 \geq l_1 + 2$. Furthermore, there is $C_{max}^{(1)}(opt) \geq l_1D + \frac{1}{3}D$. If not, the similar modifying method from the Theorem 4.2 can show $C_{max}^{(1)}(\eta) \leq l_1D + 3(C_{max}^{(1)}(opt) - l_1D) < l_1D + D$, which contradicts with the assumption. So, one can conclude that $C_{max}^{(1)}(opt) \geq l_1D + \frac{1}{3}D$.

Case 1.1: $C_{max}^{(1)}(opt) < l_1D + D$. Then, $C_{max}^{(1)}(opt) - \lfloor \frac{C_{max}^{(1)}(opt)}{D} \rfloor D = C_{max}^{(1)}(opt) - l_1D > \frac{1}{3}D$.

$$\begin{aligned}
Z(opt) &= C_{max}^{(1)}(opt) + C_{max}^{(2)}(opt) + C_{max}^{(3)}(opt) + 3D \\
&\geq l_1D + \frac{1}{3}D + l_2D + l_2D + 3D \\
&\geq 3l_1D + \frac{22}{3}D \geq \frac{22}{3}D.
\end{aligned} \tag{4.11}$$

Therefore,

$$\begin{aligned}
Z(\eta) &\leq 2C_{max}^{(1)}(opt) + 2C_{max}^{(3)}(opt) + (s+6)D + l_1D + 2C_{max}^{(2)}(opt) - \lfloor \frac{C_{max}^{(1)}(\eta)}{D} \rfloor D \\
&\quad - 2(C_{max}^{(1)}(opt) - \lfloor \frac{C_{max}^{(1)}(opt)}{D} \rfloor D) \\
&\leq 2C_{max}^{(1)}(opt) + 2C_{max}^{(1)}(opt) + 2C_{max}^{(3)}(opt) + 6D + (s-1)D - 2 \cdot \frac{1}{3}D \\
&\leq 2Z(opt) + \frac{1}{3}D \leq (2 + \frac{1}{22})Z(opt).
\end{aligned} \tag{4.12}$$

Case 1.2: $C_{max}^{(1)}(opt) \geq l_1D + D$.

If $\lceil \frac{C_{max}^{(1)}(opt)}{D} \rceil = \lceil \frac{C_{max}^{(1)}(\eta)}{D} \rceil$, there is no delivery in the interval $(C_{max}^{(1)}(opt), C_{max}^{(1)}(\eta)]$, and then $s = 0$.

$$\begin{aligned}
Z(\eta) &\leq 2C_{max}^{(1)}(opt) + 2C_{max}^{(3)}(opt) + 6D + l_1D + 2C_{max}^{(2)}(opt) - \lfloor \frac{C_{max}^{(1)}(\eta)}{D} \rfloor D \\
&\quad - 2(C_{max}^{(1)}(opt) - \lfloor \frac{C_{max}^{(1)}(opt)}{D} \rfloor D) \\
&\leq 2C_{max}^{(1)}(opt) + 2C_{max}^{(1)}(opt) + 2C_{max}^{(3)}(opt) + 6D = 2Z(opt).
\end{aligned} \tag{4.13}$$

If $\lceil \frac{C_{max}^{(1)}(opt)}{D} \rceil < \lceil \frac{C_{max}^{(1)}(\eta)}{D} \rceil$, $\lfloor \frac{C_{max}^{(1)}(\eta)}{D} \rfloor \geq \lceil \frac{C_{max}^{(1)}(opt)}{D} \rceil \geq l_1 + 2$.

$$\begin{aligned}
Z(\eta) &\leq 2C_{max}^{(1)}(opt) + 2C_{max}^{(3)}(opt) + (s+6)D + l_1D + 2C_{max}^{(2)}(opt) - \lfloor \frac{C_{max}^{(1)}(\eta)}{D} \rfloor D \\
&\quad - 2(C_{max}^{(1)}(opt) - \lfloor \frac{C_{max}^{(1)}(opt)}{D} \rfloor D) \\
&\leq 2C_{max}^{(1)}(opt) + 2C_{max}^{(1)}(opt) + 2C_{max}^{(3)}(opt) + 6D + sD - 2D \leq 2Z(opt).
\end{aligned} \tag{4.14}$$

Case 2: $C_{max}^{(1)}(\eta) < l_1D + D$. There is no delivery in the interval $(C_{max}^{(1)}(opt), C_{max}^{(1)}(\eta)]$,

so $s = 0$.

$$\begin{aligned}
Z(\eta) &\leq 2C_{max}^{(1)}(opt) + 2C_{max}^{(3)}(opt) + 6D + l_1D + 2C_{max}^{(2)}(opt) - \lfloor \frac{C_{max}^{(1)}(\eta)}{D} \rfloor D \\
&\quad - 2(C_{max}^{(1)}(opt) - \lfloor \frac{C_{max}^{(1)}(opt)}{D} \rfloor D) \\
&\leq 2C_{max}^{(1)}(opt) + 2C_{max}^{(1)}(opt) + 2C_{max}^{(3)}(opt) + 6D = 2Z(opt).
\end{aligned} \tag{4.15}$$

As such, this lemma is proved. \square

Lemma 4.4. *If there is no idle delivery point between $C_{max}^{(1)}(\eta)$ and $C_{max}^{(2)}(\eta)$ and $(i_1, i_2, i_3) = (1, 2, 3)$, the algorithm is $2 + \frac{2}{27}$ competitive.*

Proof of Lemma 4.4: Figure 4.2 shows the jobs processing in η under this situation.

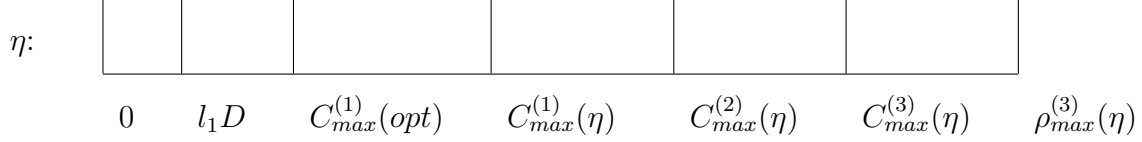


Figure 4.2. The Jobs Preparation in η (2)

In the interval $(0, l_1D]$, there are at most three batches for every delivery point, so the deliver cost will not be more than $3l_1D$.

In the interval $(l_1D, C_{max}^{(1)}(opt)]$, there are at most two batches for every delivery point, so the deliver cost will not be more than $2(\lfloor \frac{C_{max}^{(1)}(opt)}{D} \rfloor - l_1)D$.

In the interval $(C_{max}^{(1)}(opt), C_{max}^{(1)}(\eta)]$, all the jobs of Customer 1 are released, so the jobs of Customer 2 and Customer 3 processed in this interval must be released before $C_{max}^{(1)}(opt)$, which implies that there are at most two batches in total for these two customers. Suppose that the delivery cost in this interval is sD , where s is a non-negative integer not greater than 2.

In the interval $(C_{max}^{(1)}(\eta), \rho_{max}^{(3)}(\eta)]$, there is at most one batch for every delivery point except $\lceil \frac{C_{max}^{(1)}(\eta)}{D} \rceil D$ and $\lceil \frac{C_{max}^{(2)}(\eta)}{D} \rceil D$, and there is at most two batches at $\lceil \frac{C_{max}^{(1)}(\eta)}{D} \rceil D$ and $\lceil \frac{C_{max}^{(2)}(\eta)}{D} \rceil D$, so the delivery cost will not be more than $(\lceil \frac{C_{max}^{(3)}(\eta)}{D} \rceil - \lfloor \frac{C_{max}^{(1)}(\eta)}{D} \rfloor)D + 2D$.

$$\begin{aligned}
TC(\eta) &\leq 3l_1D + 2(\lfloor \frac{C_{max}^{(1)}(opt)}{D} \rfloor - l_1)D + sD + (\lceil \frac{C_{max}^{(3)}(\eta)}{D} \rceil - \\
&\quad \lfloor \frac{C_{max}^{(1)}(\eta)}{D} \rfloor)D + 2D \\
&\leq l_1D + 2C_{max}^{(1)}(opt) - 2(C_{max}^{(1)}(opt) - \lfloor \frac{C_{max}^{(1)}(opt)}{D} \rfloor D) + \\
&\quad \lceil \frac{C_{max}^{(3)}(\eta)}{D} \rceil D - \lfloor \frac{C_{max}^{(1)}(\eta)}{D} \rfloor D + (s+2)D
\end{aligned} \tag{4.16}$$

Therefore,

$$\begin{aligned}
Z(\eta) &= \lceil \frac{C_{max}^{(1)}(\eta)}{D} \rceil D + \lceil \frac{C_{max}^{(2)}(\eta)}{D} \rceil D + \lceil \frac{C_{max}^{(3)}(\eta)}{D} \rceil D + TC(\eta) \\
&\leq 2C_{max}^{(1)}(opt) + 2C_{max}^{(3)}(\eta) + l_1D - 2(C_{max}^{(1)}(opt) - \\
&\quad \lfloor \frac{C_{max}^{(1)}(opt)}{D} \rfloor D) + \lceil \frac{C_{max}^{(2)}(\eta)}{D} \rceil D + (s+5)D
\end{aligned} \tag{4.17}$$

Case 1: $C_{max}^{(1)}(\eta) \geq l_1D + D$. Similar to case 1 in the Lemma 4.3, $C_{max}^{(1)}(opt) \geq l_1D + \frac{1}{3}D$.

Case 1.1: $C_{max}^{(1)}(opt) < l_1D + D$. Then, $C_{max}^{(1)}(opt) - \lfloor \frac{C_{max}^{(1)}(opt)}{D} \rfloor D = C_{max}^{(1)}(opt) - l_1D \geq \frac{1}{3}D$. It is easy to show that $C_{max}^{(2)}(\eta) - l_1D \leq (C_{max}^{(2)}(opt) - l_1D) + (C_{max}(\mathcal{J}^{(2)}) - l_1D)$ which results in $C_{max}^{(2)}(\eta) + l_1D \leq 2C_{max}^{(2)}(opt)$.

Case 1.1.1: $s = 0$.

$$\begin{aligned}
Z(\eta) &\leq 2C_{max}^{(1)}(opt) + 2C_{max}^{(3)}(\eta) + l_1D - 2(C_{max}^{(1)}(opt) - \\
&\quad \lfloor \frac{C_{max}^{(1)}(opt)}{D} \rfloor D) + \lceil \frac{C_{max}^{(2)}(\eta)}{D} \rceil D + 5D \\
&\leq 2C_{max}^{(1)}(opt) + 2C_{max}^{(2)}(opt) + 2C_{max}^{(3)}(opt) + 6D - 2(C_{max}^{(1)}(opt) - \\
&\quad \lfloor \frac{C_{max}^{(1)}(opt)}{D} \rfloor D) \\
&\leq 2Z(opt)
\end{aligned} \tag{4.18}$$

Case 1.1.2: $s = 1$. Modify the jobs processing in the interval $(l_1D, C_{max}^{(1)}(\eta)]$ as the priority order (1, 2, 3), such that all Customer 1's jobs are completed at the time $l_1D + P(1, 1)$, all Customer 2's jobs are completed at the time $l_1D + P(1, 1) + P(1, 2)$, and all

Customer 3's jobs are completed at the time $l_1D + P(1, 1) + P(1, 2) + P(1, 3)$ (see Fig. 4). It is clear that $l_1D + P(1, 1) = C_{max}^{(1)}(opt)$, $P(1, 2) \leq P(1, 1)$, and $P(1, 3) \leq P(1, 1)$. Modify the jobs processing in the interval $(C_{max}^{(1)}(\eta), C_{max}^{(2)}(\eta)]$ as the priority order (2, 3), such that all Customer 2's jobs are completed at the time $C_{max}^{(1)}(\eta) + P(2, 2)$, and all Customer 3's jobs are completed at the time $C_{max}^{(1)}(\eta) + P(2, 2) + P(2, 3)$ (see Figure 4.3). It is clear that $l_1D + P(1, 1) + P(1, 2) + P(2, 2) \leq C_{max}^{(2)}(opt)$, $l_1D + P(1, 2) + P(2, 2) \leq C_{max}(\mathcal{J}^{(2)})$, and $l_1D + P(1, 3) + P(2, 3) \leq C_{max}(\mathcal{J}^{(2)})$.

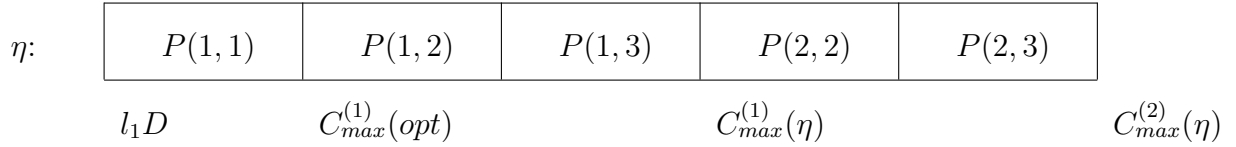


Figure 4.3. The Modified Jobs Processing in The Interval $(l_1D, C_{max}^{(2)}(\eta)]$

If the delivery in the interval $(C_{max}^{(1)}(opt), C_{max}^{(1)}(\eta)]$ is for Customer 3's jobs, then Customer 3's jobs processed in the time period $P(2, 3)$ are released later than $l_1D + D$. Then, $l_1D + D + P(2, 3) \leq C_{max}(\mathcal{J}^{(2)})$.

$$\begin{aligned}
& \lceil \frac{C_{max}^{(2)}(\eta)}{D} \rceil D + l_1D \\
& \leq C_{max}^{(2)}(\eta) + l_1D + D \\
& = l_1D + P(1, 1) + P(1, 2) + P(1, 3) + P(2, 2) + P(2, 3) + l_1D + D \quad (4.19) \\
& = l_1D + P(1, 1) + P(1, 2) + P(2, 2) + l_1D + D + P(2, 3) + P(1, 3) \\
& \leq 2C_{max}^{(2)}(opt) + P(1, 3).
\end{aligned}$$

$$\begin{aligned}
Z(\eta) &\leq 2C_{max}^{(1)}(opt) + 2C_{max}^{(3)}(\eta) + l_1D - 2(C_{max}^{(1)}(opt) - \lfloor \frac{C_{max}^{(1)}(opt)}{D} \rfloor D) + \\
&\quad \lceil \frac{C_{max}^{(2)}(\eta)}{D} \rceil D + 6D \\
&\leq 2C_{max}^{(1)}(opt) + 2C_{max}^{(2)}(opt) + 2C_{max}^{(3)}(opt) + 6D + P(1, 3) - \\
&\quad 2(C_{max}^{(1)}(opt) - \lfloor \frac{C_{max}^{(1)}(opt)}{D} \rfloor D) \\
&\leq 2Z(opt).
\end{aligned} \tag{4.20}$$

If the delivery in the interval $(C_{max}^{(1)}(opt), C_{max}^{(1)}(\eta)]$ is for Customer 2's jobs, then Customer 2's jobs processed in the time period $P(2, 2)$ are released later than $l_1D + D$. Then, $l_1D + D + P(2, 2) \leq C_{max}(\mathcal{J}^{(2)})$.

$$\begin{aligned}
&\lceil \frac{C_{max}^{(2)}(\eta)}{D} \rceil D + l_1D \\
&\leq C_{max}^{(2)}(\eta) + l_1D + D \\
&= l_1D + P(1, 1) + P(1, 2) + P(1, 3) + P(2, 2) + P(2, 3) + l_1D + D \\
&= l_1D + P(1, 3) + P(2, 3) + l_1D + D + P(2, 2) + P(1, 1) + P(1, 2) \\
&\leq 2C_{max}^{(2)}(opt) + P(1, 1) + P(1, 2).
\end{aligned} \tag{4.21}$$

$$\begin{aligned}
Z(\eta) &\leq 2C_{max}^{(1)}(opt) + 2C_{max}^{(3)}(\eta) + l_1D - 2(C_{max}^{(1)}(opt) - \lfloor \frac{C_{max}^{(1)}(opt)}{D} \rfloor D) + \\
&\quad \lceil \frac{C_{max}^{(2)}(\eta)}{D} \rceil D + 6D \\
&\leq 2C_{max}^{(1)}(opt) + 2C_{max}^{(2)}(opt) + 2C_{max}^{(3)}(opt) + 6D + P(1, 1) + P(1, 2) - \\
&\quad 2(C_{max}^{(1)}(opt) - \lfloor \frac{C_{max}^{(1)}(opt)}{D} \rfloor D) \\
&\leq 2Z(opt).
\end{aligned} \tag{4.22}$$

Case 1.1.3: $s = 2$. There must be a delivery in the interval $(C_{max}^{(1)}(opt), C_{max}^{(1)}(\eta)]$ for for Customer 2's jobs, so Customer 2's jobs processed in the time period $P(2, 2)$ are released later than $l_1D + D$ which implies $C_{max}^{(2)}(opt) > l_1D + D + P(2, 2)$. Similarly,

$$C_{max}^{(2)}(opt) > l_1 D + D + P(2, 3).$$

Case 1.1.3.1: $C_{max}^{(2)}(\eta) \leq l_1 D + 2D$. In this case, $\lceil \frac{C_{max}^{(2)}(\eta)}{D} \rceil D + l_1 D = 2l_1 D + 2D < 2C_{max}^{(2)}(opt)$.

As

$$\begin{aligned} Z(opt) &= C_{max}^{(1)}(opt) + C_{max}^{(2)}(opt) + C_{max}^{(3)}(opt) + 3D \\ &> l_1 D + \frac{1}{3}D + l_1 D + D + l_1 D + D + 3D \geq \frac{16}{3}D, \end{aligned} \quad (4.23)$$

$$\begin{aligned} Z(\eta) &\leq 2C_{max}^{(1)}(opt) + 2C_{max}^{(3)}(\eta) + l_1 D - 2(C_{max}^{(1)}(opt) - \lfloor \frac{C_{max}^{(1)}(opt)}{D} \rfloor D) + \\ &\quad \lceil \frac{C_{max}^{(2)}(\eta)}{D} \rceil D + 7D \\ &\leq 2C_{max}^{(1)}(opt) + 2C_{max}^{(2)}(opt) + 2C_{max}^{(3)}(opt) + 6D + D - \\ &\quad 2(C_{max}^{(1)}(opt) - \lfloor \frac{C_{max}^{(1)}(opt)}{D} \rfloor D) \\ &\leq 2Z(opt) + \frac{1}{3}D \leq (2 + \frac{1}{16})Z(opt). \end{aligned} \quad (4.24)$$

Case 1.1.3.2: $C_{max}^{(2)}(\eta) > l_1 D + 2D$.

$$\begin{aligned} &l_1 D + 3P(1, 1) + P(2, 2) + P(2, 3) \\ &\geq l_1 D + P(1, 1) + P(1, 2) + P(1, 3) + P(2, 2) + P(2, 3) \\ &= C_{max}^{(2)}(\eta) > l_1 D + 2D \end{aligned} \quad (4.25)$$

$$\Rightarrow P(2, 2) + P(2, 3) \geq 2D - 3P(1, 1)$$

Meanwhile, $C_{max}^{(2)}(opt) \geq l_1 D + D + \frac{1}{2}(P(2, 2) + P(2, 3)) \geq l_1 D + 2D - \frac{3}{2}P(1, 1)$.

Therefore,

$$\begin{aligned} Z(opt) &= C_{max}^{(1)}(opt) + C_{max}^{(2)}(opt) + C_{max}^{(3)}(opt) + 3D \\ &\geq l_1 D + P(1, 1) + l_1 D + 2D - \frac{3}{2}P(1, 1) + l_1 D + 2D + 3D \\ &\geq 7D - \frac{1}{2}P(1, 1), \end{aligned} \quad (4.26)$$

In addition, from

$$\begin{aligned}
& \lceil \frac{C_{max}^{(2)}(\eta)}{D} \rceil D + l_1 D + D \\
& \leq C_{max}^{(2)}(\eta) + l_1 D + 2D \\
& = l_1 D + P(1, 1) + P(1, 2) + P(1, 3) + P(2, 2) + P(2, 3) + l_1 D + 2D \quad (4.27) \\
& = l_1 D + D + P(2, 2) + l_1 D + D + P(2, 3) + P(1, 1) + P(1, 2) + P(1, 3) \\
& \leq 2C_{max}^{(1)}(opt) + 3P(1, 1).
\end{aligned}$$

$$\begin{aligned}
Z(\eta) & \leq 2C_{max}^{(1)}(opt) + 2C_{max}^{(3)}(\eta) + l_1 D - 2(C_{max}^{(1)}(opt) - \lfloor \frac{C_{max}^{(1)}(opt)}{D} \rfloor D) + \\
& \quad \lceil \frac{C_{max}^{(2)}(\eta)}{D} \rceil D + 7D \\
& \leq 2C_{max}^{(1)}(opt) + 2C_{max}^{(2)}(opt) + 2C_{max}^{(3)}(opt) + 6D + 3P(1, 1) - \\
& \quad 2(C_{max}^{(1)}(opt) - \lfloor \frac{C_{max}^{(1)}(opt)}{D} \rfloor D) \quad (4.28) \\
& \leq 2Z(opt) + P(1, 1).
\end{aligned}$$

and

$$\begin{aligned}
& \lceil \frac{C_{max}^{(2)}(\eta)}{D} \rceil D + l_1 D + D \\
& \leq C_{max}^{(2)}(\eta) + l_1 D + 2D \\
& = l_1 D + P(1, 1) + P(1, 2) + P(1, 3) + P(2, 2) + P(2, 3) + l_1 D + 2D \quad (4.29) \\
& = l_1 D + P(1, 1) + P(1, 2) + P(2, 2) + l_1 D + D + P(2, 3) + D + P(1, 3) \\
& \leq 2C_{max}^{(1)}(opt) + D + P(1, 3).
\end{aligned}$$

$$\begin{aligned}
Z(\eta) &\leq 2C_{max}^{(1)}(opt) + 2C_{max}^{(3)}(\eta) + l_1D - 2(C_{max}^{(1)}(opt) - \lfloor \frac{C_{max}^{(1)}(opt)}{D} \rfloor D) + \\
&\quad \lceil \frac{C_{max}^{(2)}(\eta)}{D} \rceil D + 7D \\
&\leq 2C_{max}^{(1)}(opt) + 2C_{max}^{(2)}(opt) + 2C_{max}^{(3)}(opt) + 6D + D + P(1, 3) - \\
&\quad 2(C_{max}^{(1)}(opt) - \lfloor \frac{C_{max}^{(1)}(opt)}{D} \rfloor D) \\
&\leq 2Z(opt) + D - P(1, 1).
\end{aligned} \tag{4.30}$$

so

$$\frac{Z(\eta)}{Z(opt)} \leq 2 + \min\left\{\frac{P(1, 1)}{7D - \frac{1}{2}P(1, 1)}, \frac{D - P(1, 1)}{7D - \frac{1}{2}P(1, 1)}\right\}. \tag{4.31}$$

where $P(1, 1) \in [\frac{1}{3}D, D)$. When $P(1, 1) = \frac{1}{2}D$, the right term will achieve the maximum value $2 + \frac{2}{27}$.

Case 1.2 $C_{max}^{(1)}(opt) \geq l_1D + D$.

Case 1.2.1: $s = 0$. Similar to case 1.1.1, $Z(\eta) \leq 2Z(opt)$.

Case 1.2.2: $s = 1$.

Case 1.2.2.1: there is a delivery in the interval $(C_{max}^{(1)}(opt), C_{max}^{(1)}(\eta)]$ for Customer 2's jobs. In this case, the delivery point $\lfloor \frac{C_{max}^{(1)}(opt)}{D} \rfloor D$ needs to be considered. If there is no more than one batch at the delivery point $\lfloor \frac{C_{max}^{(1)}(opt)}{D} \rfloor D$, then the result will be similar to the case $s = 0$ which implies $Z(\eta) \leq 2Z(opt)$. Therefore, the case that there are two batches at $\lfloor \frac{C_{max}^{(1)}(opt)}{D} \rfloor D$ should be explored. Modify the jobs processing in the intervals $(\lfloor \frac{C_{max}^{(1)}(opt)}{D} \rfloor D, C_{max}^{(1)}(\eta)]$ and $(C_{max}^{(1)}(\eta), C_{max}^{(2)}(\eta)]$ in a similar way as case 1.1.2 (see Figure 4.4).

η :	$P'(1, 1)$	$P'(1, 2)$	$P'(1, 3)$	$P'(2, 2)$	$P'(2, 3)$
	$\lfloor \frac{C_{max}^{(1)}(opt)}{D} \rfloor D$	$C_{max}^{(1)}(opt)$		$C_{max}^{(1)}(\eta)$	$C_{max}^{(2)}(\eta)$

Figure 4.4. The Modified Jobs Processing in The Interval $(\lfloor \frac{C_{max}^{(1)}(opt)}{D} \rfloor D, C_{max}^{(2)}(\eta)]$

If the two batches are for Customer 1 and Customer 2 at $\lfloor \frac{C_{max}^{(1)}(opt)}{D} \rfloor D$, then $\lfloor \frac{C_{max}^{(1)}(opt)}{D} \rfloor D + P'(1, 1) \leq C_{max}^{(1)}(opt)$, $\lfloor \frac{C_{max}^{(1)}(opt)}{D} \rfloor D + P'(1, 2) \leq C_{max}^{(1)}(opt)$, $\lceil \frac{C_{max}^{(1)}(opt)}{D} \rceil D + P'(2, 2) \leq C_{max}^{(2)}(opt)$ and $l_1 D + P'(1, 3) + P'(2, 3) \leq C_{max}^{(2)}(opt)$.

$$\begin{aligned}
& \lceil \frac{C_{max}^{(2)}(\eta)}{D} \rceil D + l_1 D \\
& \leq C_{max}^{(2)}(\eta) + l_1 D + D \\
& = \lfloor \frac{C_{max}^{(1)}(opt)}{D} \rfloor D + P'(1, 1) + P'(1, 2) + P'(1, 3) + P'(2, 2) + P'(2, 3) + l_1 D + D \quad (4.32) \\
& = l_1 D + P'(1, 3) + P'(2, 3) + \lfloor \frac{C_{max}^{(1)}(opt)}{D} \rfloor D + D + P'(2, 2) + P'(1, 1) + P'(1, 2) \\
& \leq 2C_{max}^{(2)}(opt) + P'(1, 1) + P'(1, 2)
\end{aligned}$$

$$\begin{aligned}
Z(\eta) & \leq 2C_{max}^{(1)}(opt) + 2C_{max}^{(3)}(\eta) + l_1 D - 2(C_{max}^{(1)}(opt) - \lfloor \frac{C_{max}^{(1)}(opt)}{D} \rfloor D) + \\
& \quad \lceil \frac{C_{max}^{(2)}(\eta)}{D} \rceil D + 6D \\
& \leq 2C_{max}^{(1)}(opt) + 2C_{max}^{(2)}(opt) + 2C_{max}^{(3)}(opt) + 6D + P'(1, 1) + P'(1, 2) - \quad (4.33) \\
& \quad 2(C_{max}^{(1)}(opt) - \lfloor \frac{C_{max}^{(1)}(opt)}{D} \rfloor D) \\
& \leq 2Z(opt).
\end{aligned}$$

If the two batches are for Customer 2 and Customer 3 at $\lfloor \frac{C_{max}^{(1)}(opt)}{D} \rfloor D$, then $\lfloor \frac{C_{max}^{(1)}(opt)}{D} \rfloor D + P'(1, 2) \leq C_{max}^{(1)}(opt)$, $\lfloor \frac{C_{max}^{(1)}(opt)}{D} \rfloor D + P'(1, 3) \leq C_{max}^{(1)}(opt)$, $\lfloor \frac{C_{max}^{(1)}(opt)}{D} \rfloor D + P'(1, 3) +$

$$P'(2, 3) \leq C_{max}^{(2)}(opt) \text{ and } \lceil \frac{C_{max}^{(1)}(opt)}{D} \rceil D + P'(2, 2) \leq C_{max}^{(2)}(opt).$$

$$\begin{aligned} & \lceil \frac{C_{max}^{(2)}(\eta)}{D} \rceil D + l_1 D \\ & \leq C_{max}^{(2)}(\eta) + l_1 D + D \\ & = \lfloor \frac{C_{max}^{(1)}(opt)}{D} \rfloor D + P'(1, 1) + P'(1, 2) + P'(1, 3) + P'(2, 2) + \\ & \quad P'(2, 3) + l_1 D + D \\ & = \lfloor \frac{C_{max}^{(1)}(opt)}{D} \rfloor D + P'(1, 3) + P'(2, 3) + l_1 D + P'(1, 1) + \\ & \quad P'(2, 2) + D + P'(1, 2) \end{aligned} \tag{4.34}$$

$$\begin{aligned} & \leq C_{max}^{(2)}(opt) + C_{max}^{(1)}(opt) + P'(2, 2) + D + P'(1, 2) \\ & = C_{max}^{(2)}(opt) + \lceil \frac{C_{max}^{(1)}(opt)}{D} \rceil D + P'(2, 2) + P'(1, 2) + \\ & \quad (C_{max}^{(1)}(opt) - \lfloor \frac{C_{max}^{(1)}(opt)}{D} \rfloor D) \\ & \leq 2C_{max}^{(2)}(opt) + P'(1, 2) + (C_{max}^{(1)}(opt) - \lfloor \frac{C_{max}^{(1)}(opt)}{D} \rfloor D) \end{aligned}$$

$$\begin{aligned} Z(\eta) & \leq 2C_{max}^{(1)}(opt) + 2C_{max}^{(3)}(\eta) + l_1 D - 2(C_{max}^{(1)}(opt) - \lfloor \frac{C_{max}^{(1)}(opt)}{D} \rfloor D) + \\ & \quad \lceil \frac{C_{max}^{(2)}(\eta)}{D} \rceil D + 6D \\ & \leq 2C_{max}^{(1)}(opt) + 2C_{max}^{(2)}(opt) + 2C_{max}^{(3)}(opt) + 6D + P'(1, 2) + \\ & \quad (C_{max}^{(1)}(opt) - \lfloor \frac{C_{max}^{(1)}(opt)}{D} \rfloor D) - 2(C_{max}^{(1)}(opt) - \lfloor \frac{C_{max}^{(1)}(opt)}{D} \rfloor D) \\ & \leq 2Z(opt). \end{aligned} \tag{4.35}$$

If the two batches are for Customer 1 and Customer 3 at $\lfloor \frac{C_{max}^{(1)}(opt)}{D} \rfloor D$, then $\lfloor \frac{C_{max}^{(1)}(opt)}{D} \rfloor D + P'(1, 1) \leq C_{max}^{(1)}(opt)$, $\lfloor \frac{C_{max}^{(1)}(opt)}{D} \rfloor D + P'(1, 3) \leq C_{max}^{(1)}(opt)$, $\lfloor \frac{C_{max}^{(1)}(opt)}{D} \rfloor D + P'(1, 3) +$

$$P'(2, 3) \leq C_{max}^{(2)}(opt) \text{ and } \lceil \frac{C_{max}^{(1)}(opt)}{D} \rceil D + P'(2, 2) \leq C_{max}^{(2)}(opt).$$

$$\begin{aligned} & \lceil \frac{C_{max}^{(2)}(\eta)}{D} \rceil D + l_1 D \leq C_{max}^{(2)}(\eta) + l_1 D + D \\ = & \lfloor \frac{C_{max}^{(1)}(opt)}{D} \rfloor D + P'(1, 1) + P'(1, 2) + P'(1, 3) + P'(2, 2) + \\ & P'(2, 3) + l_1 D + D \\ = & \lfloor \frac{C_{max}^{(1)}(opt)}{D} \rfloor D + P'(1, 3) + P'(2, 3) + l_1 D + P'(1, 2) + \\ & P'(2, 2) + D + P'(1, 1) \end{aligned} \tag{4.36}$$

$$\begin{aligned} & \leq C_{max}^{(2)}(opt) + C_{max}^{(1)}(opt) + P'(2, 2) + D + P'(1, 1) \\ = & C_{max}^{(2)}(opt) + \lceil \frac{C_{max}^{(1)}(opt)}{D} \rceil D + P'(2, 2) + P'(1, 1) + \\ & (C_{max}^{(1)}(opt) - \lfloor \frac{C_{max}^{(1)}(opt)}{D} \rfloor D) \\ & \leq 2C_{max}^{(2)}(opt) + P'(1, 1) + (C_{max}^{(1)}(opt) - \lfloor \frac{C_{max}^{(1)}(opt)}{D} \rfloor D) \end{aligned}$$

$$\begin{aligned} Z(\eta) & \leq 2C_{max}^{(1)}(opt) + 2C_{max}^{(3)}(\eta) + l_1 D - 2(C_{max}^{(1)}(opt) - \lfloor \frac{C_{max}^{(1)}(opt)}{D} \rfloor D) \\ & \quad + \lceil \frac{C_{max}^{(2)}(\eta)}{D} \rceil D + 6D \\ & \leq 2C_{max}^{(1)}(opt) + 2C_{max}^{(2)}(opt) + 2C_{max}^{(3)}(opt) + 6D + P'(1, 1) + \\ & \quad (C_{max}^{(1)}(opt) - \lfloor \frac{C_{max}^{(1)}(opt)}{D} \rfloor D) - 2(C_{max}^{(1)}(opt) - \lfloor \frac{C_{max}^{(1)}(opt)}{D} \rfloor D) \\ & \leq 2Z(opt). \end{aligned} \tag{4.37}$$

Case 1.2.2.2: there is a delivery in the interval $(C_{max}^{(1)}(opt), C_{max}^{(1)}(\eta)]$ for Customer 3's jobs.

$$\begin{aligned}
& \lceil \frac{C_{max}^{(2)}(\eta)}{D} \rceil D + l_1 D \leq C_{max}^{(2)}(\eta) + l_1 D + D \\
& = l_1 D + P(1,1) + P(1,2) + P(1,3) + P(2,2) + P(2,3) + l_1 D + D \\
& = l_1 D + P(1,1) + P(1,2) + P(2,2) + l_1 D + D + P(2,3) + P(1,3) \\
& \leq C_{max}^{(2)}(opt) + \lceil \frac{C_{max}^{(1)}(opt)}{D} \rceil D + P(2,3) + l_1 D + D + P(1,3) - \lceil \frac{C_{max}^{(1)}(opt)}{D} \rceil D \\
& \leq 2C_{max}^{(2)}(opt) + l_1 D + P(1,3) - \lfloor \frac{C_{max}^{(1)}(opt)}{D} \rfloor D \\
& \leq 2C_{max}^{(2)}(opt) + (C_{max}^{(1)}(opt) - \lfloor \frac{C_{max}^{(1)}(opt)}{D} \rfloor D)
\end{aligned} \tag{4.38}$$

$$\begin{aligned}
Z(\eta) & \leq 2C_{max}^{(1)}(opt) + 2C_{max}^{(3)}(\eta) + l_1 D - 2(C_{max}^{(1)}(opt) - \lfloor \frac{C_{max}^{(1)}(opt)}{D} \rfloor D) + \\
& \quad \lceil \frac{C_{max}^{(2)}(\eta)}{D} \rceil D + 6D \\
& \leq 2C_{max}^{(1)}(opt) + 2C_{max}^{(2)}(opt) + 2C_{max}^{(3)}(opt) + 6D - \\
& \quad (C_{max}^{(1)}(opt) - \lfloor \frac{C_{max}^{(1)}(opt)}{D} \rfloor D) \\
& \leq 2Z(opt).
\end{aligned} \tag{4.39}$$

Case 1.2.3: $s = 2$. Then, $\lceil \frac{C_{max}^{(1)}(opt)}{D} \rceil D + P'(2,2) \leq C_{max}^{(2)}(opt)$ and $\lceil \frac{C_{max}^{(1)}(opt)}{D} \rceil D + P'(2,3) \leq C_{max}^{(2)}(opt)$.

Case 1.2.3.1: There is no batch at the delivery point $\lfloor \frac{C_{max}^{(1)}(opt)}{D} \rfloor D$. This is similar to the case $s = 0$, which implies $Z(\eta) \leq 2Z(opt)$.

Case 1.2.3.2: There is one batch at the delivery point $\lfloor \frac{C_{max}^{(1)}(opt)}{D} \rfloor D$.

$$\begin{aligned}
& \lceil \frac{C_{max}^{(2)}(\eta)}{D} \rceil D + l_1 D \leq C_{max}^{(2)}(\eta) + l_1 D + D \\
& = l_1 D + P(1, 1) + P(1, 2) + P(1, 3) + P(2, 2) + P(2, 3) + l_1 D + D \\
& = l_1 D + P(1, 1) + P(1, 2) + P(2, 2) + l_1 D + P(1, 3) + D + P(2, 3) \\
& \leq C_{max}^{(2)}(opt) + C_{max}^{(1)}(opt) + D + P(2, 3) \\
& = C_{max}^{(2)}(opt) + \lceil \frac{C_{max}^{(1)}(opt)}{D} \rceil D + P(2, 3) + (C_{max}^{(1)}(opt) - \lfloor \frac{C_{max}^{(1)}(opt)}{D} \rfloor D) \\
& \leq 2C_{max}^{(2)}(opt) + (C_{max}^{(1)}(opt) - \lfloor \frac{C_{max}^{(1)}(opt)}{D} \rfloor D)
\end{aligned} \tag{4.40}$$

$$\begin{aligned}
Z(\eta) & \leq 2C_{max}^{(1)}(opt) + 2C_{max}^{(3)}(\eta) + l_1 D - 2(C_{max}^{(1)}(opt) - \lfloor \frac{C_{max}^{(1)}(opt)}{D} \rfloor D) + \\
& \quad \lceil \frac{C_{max}^{(2)}(\eta)}{D} \rceil D + 6D \\
& \leq 2C_{max}^{(1)}(opt) + 2C_{max}^{(2)}(opt) + 2C_{max}^{(3)}(opt) + 6D - \\
& \quad (C_{max}^{(1)}(opt) - \lfloor \frac{C_{max}^{(1)}(opt)}{D} \rfloor D) \\
& \leq 2Z(opt).
\end{aligned} \tag{4.41}$$

Case 1.2.3.3: There is two batches at the delivery point $\lfloor \frac{C_{max}^{(1)}(opt)}{D} \rfloor D$.

If the two batches are for Customer 1 and Customer 2 at $\lfloor \frac{C_{max}^{(1)}(opt)}{D} \rfloor D$, then $\lfloor \frac{C_{max}^{(1)}(opt)}{D} \rfloor D + P'(1, 1) \leq C_{max}^{(1)}(opt)$, $\lfloor \frac{C_{max}^{(1)}(opt)}{D} \rfloor D + P'(1, 2) \leq C_{max}^{(1)}(opt)$.

$$\begin{aligned}
& \lceil \frac{C_{max}^{(2)}(\eta)}{D} \rceil D + l_1 D + D \leq C_{max}^{(2)}(\eta) + l_1 D + 2D \\
= & l_1 D + 2D + \lfloor \frac{C_{max}^{(1)}(opt)}{D} \rfloor D + P'(1,1) + P'(1,2) + P'(1,3) + \\
& P'(2,2) + P'(2,3) \\
= & l_1 D + P'(1,3) + P'(2,3) + D + \lfloor \frac{C_{max}^{(1)}(opt)}{D} \rfloor D + P'(2,2) + D + \\
& P'(1,1) + P'(1,2) + D \tag{4.42} \\
\leq & C_{max}^{(1)}(opt) + P'(2,3) + D + \lceil \frac{C_{max}^{(1)}(opt)}{D} \rceil D + P'(2,2) + P'(1,1) + P'(1,2) \\
\leq & \lceil \frac{C_{max}^{(1)}(opt)}{D} \rceil D + P'(2,3) + \lceil \frac{C_{max}^{(1)}(opt)}{D} \rceil D + P'(2,2) + P'(1,1) + P'(1,2) \\
& + (C_{max}^{(1)}(opt) - \lfloor \frac{C_{max}^{(1)}(opt)}{D} \rfloor D) \\
\leq & 2C_{max}^{(2)}(opt) + 3(C_{max}^{(1)}(opt) - \lfloor \frac{C_{max}^{(1)}(opt)}{D} \rfloor D).
\end{aligned}$$

$$\begin{aligned}
Z(\eta) \leq & 2C_{max}^{(1)}(opt) + 2C_{max}^{(3)}(\eta) + l_1 D - 2(C_{max}^{(1)}(opt) - \lfloor \frac{C_{max}^{(1)}(opt)}{D} \rfloor D) + \\
& \lceil \frac{C_{max}^{(2)}(\eta)}{D} \rceil D + 7D \\
\leq & 2C_{max}^{(1)}(opt) + 2C_{max}^{(2)}(opt) + 2C_{max}^{(3)}(opt) + 6D + \\
& (C_{max}^{(1)}(opt) - \lfloor \frac{C_{max}^{(1)}(opt)}{D} \rfloor D) \tag{4.43} \\
\leq & 2Z(opt) + (C_{max}^{(1)}(opt) - \lfloor \frac{C_{max}^{(1)}(opt)}{D} \rfloor D).
\end{aligned}$$

Meanwhile,

$$\begin{aligned}
& \lceil \frac{C_{max}^{(2)}(\eta)}{D} \rceil D + l_1 D + D \leq C_{max}^{(2)}(\eta) + l_1 D + 2D \\
& = l_1 D + 2D + \lfloor \frac{C_{max}^{(1)}(opt)}{D} \rfloor D + P'(1,1) + P'(1,2) + P'(1,3) + \\
& \quad P'(2,2) + P'(2,3) \\
& = l_1 D + P'(1,3) + P'(2,3) + D + \lfloor \frac{C_{max}^{(1)}(opt)}{D} \rfloor D + \\
& \quad P'(1,1) + P'(1,2) + P'(2,2) + D \\
& \leq C_{max}^{(1)}(opt) + P'(2,3) + D + C_{max}^{(2)}(opt) + D \\
& \leq \lceil \frac{C_{max}^{(1)}(opt)}{D} \rceil D + P'(2,3) + C_{max}^{(2)}(opt) + D + (C_{max}^{(1)}(opt) - \lfloor \frac{C_{max}^{(1)}(opt)}{D} \rfloor D) \\
& \leq 2C_{max}^{(2)}(opt) + D + (C_{max}^{(1)}(opt) - \lfloor \frac{C_{max}^{(1)}(opt)}{D} \rfloor D).
\end{aligned} \tag{4.44}$$

$$\begin{aligned}
Z(\eta) & \leq 2C_{max}^{(1)}(opt) + 2C_{max}^{(3)}(\eta) + l_1 D - 2(C_{max}^{(1)}(opt) - \lfloor \frac{C_{max}^{(1)}(opt)}{D} \rfloor D) + \\
& \quad \lceil \frac{C_{max}^{(2)}(\eta)}{D} \rceil D + 7D \\
& \leq 2C_{max}^{(1)}(opt) + 2C_{max}^{(2)}(opt) + 2C_{max}^{(3)}(opt) + 6D + D - \\
& \quad (C_{max}^{(1)}(opt) - \lfloor \frac{C_{max}^{(1)}(opt)}{D} \rfloor D) \\
& \leq 2Z(opt) + D - (C_{max}^{(1)}(opt) - \lfloor \frac{C_{max}^{(1)}(opt)}{D} \rfloor D).
\end{aligned} \tag{4.45}$$

As

$$\begin{aligned}
Z(opt) & = C_{max}^{(1)}(opt) + C_{max}^{(2)}(opt) + C_{max}^{(3)}(opt) + 3D \\
& \geq \lfloor \frac{C_{max}^{(1)}(opt)}{D} \rfloor D + (C_{max}^{(1)}(opt) - \lfloor \frac{C_{max}^{(1)}(opt)}{D} \rfloor D) + \\
& \quad \lceil \frac{C_{max}^{(1)}(opt)}{D} \rceil D + \lceil \frac{C_{max}^{(1)}(opt)}{D} \rceil D + 3D \\
& \geq l_1 D + D + (C_{max}^{(1)}(opt) - \lfloor \frac{C_{max}^{(1)}(opt)}{D} \rfloor D) + \\
& \quad l_D + 2D + l_1 D + 2D + 3D \\
& \geq 8D + (C_{max}^{(1)}(opt) - \lfloor \frac{C_{max}^{(1)}(opt)}{D} \rfloor D).
\end{aligned} \tag{4.46}$$

Therefore,

$$\frac{Z(\eta)}{Z(opt)} \leq 2 + \min \left\{ \frac{(C_{max}^{(1)}(opt) - \lfloor \frac{C_{max}^{(1)}(opt)}{D} \rfloor D)}{8D + (C_{max}^{(1)}(opt) - \lfloor \frac{C_{max}^{(1)}(opt)}{D} \rfloor D)}, \right. \\ \left. \frac{D - (C_{max}^{(1)}(opt) - \lfloor \frac{C_{max}^{(1)}(opt)}{D} \rfloor D)}{8D + (C_{max}^{(1)}(opt) - \lfloor \frac{C_{max}^{(1)}(opt)}{D} \rfloor D)} \right\}. \quad (4.47)$$

where $C_{max}^{(1)}(opt) - \lfloor \frac{C_{max}^{(1)}(opt)}{D} \rfloor D \in [0, D)$. When $C_{max}^{(1)}(opt) - \lfloor \frac{C_{max}^{(1)}(opt)}{D} \rfloor D = \frac{1}{2}D$, the right term achieves the maximum value $2 + \frac{1}{17}$.

If the two batches are for Customer 1 and Customer 3 at $\lfloor \frac{C_{max}^{(1)}(opt)}{D} \rfloor D$, then $\lfloor \frac{C_{max}^{(1)}(opt)}{D} \rfloor D + P'(1, 1) \leq C_{max}^{(1)}(opt)$, $\lfloor \frac{C_{max}^{(1)}(opt)}{D} \rfloor D + P'(1, 3) \leq C_{max}^{(1)}(opt)$.

$$\begin{aligned} & \lceil \frac{C_{max}^{(2)}(\eta)}{D} \rceil D + l_1 D + D \leq C_{max}^{(2)}(\eta) + l_1 D + 2D \\ & = l_1 D + 2D + \lfloor \frac{C_{max}^{(1)}(opt)}{D} \rfloor D + P'(1, 1) + P'(1, 2) + P'(1, 3) + \\ & \quad P'(2, 2) + P'(2, 3) \\ & = l_1 D + P'(1, 2) + P'(2, 2) + D + \lfloor \frac{C_{max}^{(1)}(opt)}{D} \rfloor D + P'(2, 3) + D + \\ & \quad P'(1, 1) + P'(1, 3) + D \quad (4.48) \\ & \leq C_{max}^{(1)}(opt) + P'(2, 2) + D + \lceil \frac{C_{max}^{(1)}(opt)}{D} \rceil D + P'(2, 3) + P'(1, 1) + P'(1, 3) \\ & \leq \lceil \frac{C_{max}^{(1)}(opt)}{D} \rceil D + P'(2, 2) + \lceil \frac{C_{max}^{(1)}(opt)}{D} \rceil D + P'(2, 3) + P'(1, 1) + P'(1, 3) \\ & \quad + (C_{max}^{(1)}(opt) - \lfloor \frac{C_{max}^{(1)}(opt)}{D} \rfloor D) \\ & \leq 2C_{max}^{(2)}(opt) + 3(C_{max}^{(1)}(opt) - \lfloor \frac{C_{max}^{(1)}(opt)}{D} \rfloor D). \end{aligned}$$

$$Z(\eta) \leq 2Z(opt) + (C_{max}^{(1)}(opt) - \lfloor \frac{C_{max}^{(1)}(opt)}{D} \rfloor D). \quad (4.49)$$

Meanwhile,

$$\begin{aligned}
& \lceil \frac{C_{max}^{(2)}(\eta)}{D} \rceil D + l_1 D + D \leq C_{max}^{(2)}(\eta) + l_1 D + 2D \\
& = l_1 D + 2D + \lfloor \frac{C_{max}^{(1)}(opt)}{D} \rfloor D + P'(1,1) + P'(1,2) + P'(1,3) + \\
& \quad P'(2,2) + P'(2,3) \\
& = l_1 D + P'(1,1) + P'(1,2) + P'(2,2) + D + \lfloor \frac{C_{max}^{(1)}(opt)}{D} \rfloor D + \\
& \quad P'(1,3) + P'(2,3) + D \tag{4.50} \\
& \leq C_{max}^{(2)}(opt) + D + C_{max}^{(1)}(opt) + P'(2,3) + D \\
& \leq C_{max}^{(2)}(opt) + D + \lceil \frac{C_{max}^{(1)}(opt)}{D} \rceil D + P'(2,3) + (C_{max}^{(1)}(opt) - \lfloor \frac{C_{max}^{(1)}(opt)}{D} \rfloor D) \\
& \leq 2C_{max}^{(2)}(opt) + D + (C_{max}^{(1)}(opt) - \lfloor \frac{C_{max}^{(1)}(opt)}{D} \rfloor D).
\end{aligned}$$

$$Z(\eta) \leq 2Z(opt) + D - (C_{max}^{(1)}(opt) - \lfloor \frac{C_{max}^{(1)}(opt)}{D} \rfloor D). \tag{4.51}$$

Therefore,

$$\begin{aligned}
\frac{Z(\eta)}{Z(opt)} & \leq 2 + \min \left\{ \frac{(C_{max}^{(1)}(opt) - \lfloor \frac{C_{max}^{(1)}(opt)}{D} \rfloor D)}{8D + (C_{max}^{(1)}(opt) - \lfloor \frac{C_{max}^{(1)}(opt)}{D} \rfloor D)}, \right. \\
& \quad \left. \frac{D - (C_{max}^{(1)}(opt) - \lfloor \frac{C_{max}^{(1)}(opt)}{D} \rfloor D)}{8D + (C_{max}^{(1)}(opt) - \lfloor \frac{C_{max}^{(1)}(opt)}{D} \rfloor D)} \right\} \tag{4.52} \\
& \leq 2 + \frac{1}{17}.
\end{aligned}$$

If the two batches are for Customer 2 and Customer 3 at $\lfloor \frac{C_{max}^{(1)}(opt)}{D} \rfloor D$, then $\lfloor \frac{C_{max}^{(1)}(opt)}{D} \rfloor D +$

$$P'(1,2) \leq C_{max}^{(1)}(opt), \lfloor \frac{C_{max}^{(1)}(opt)}{D} \rfloor D + P'(1,2) \leq C_{max}^{(1)}(opt).$$

$$\begin{aligned}
& \lceil \frac{C_{max}^{(2)}(\eta)}{D} \rceil D + l_1 D + D \leq C_{max}^{(2)}(\eta) + l_1 D + 2D \\
& = l_1 D + 2D + \lfloor \frac{C_{max}^{(1)}(opt)}{D} \rfloor D + P'(1,1) + P'(1,2) + P'(1,3) + \\
& \quad P'(2,2) + P'(2,3) \\
& = l_1 D + P'(1,1) + P'(2,2) + D + \lfloor \frac{C_{max}^{(1)}(opt)}{D} \rfloor D + P'(2,3) + D + \\
& \quad P'(1,2) + P'(1,3) + D \tag{4.53}
\end{aligned}$$

$$\begin{aligned}
& \leq C_{max}^{(1)}(opt) + P'(2,2) + D + \lceil \frac{C_{max}^{(1)}(opt)}{D} \rceil D + P'(2,3) + P'(1,2) + P'(1,3) \\
& \leq \lceil \frac{C_{max}^{(1)}(opt)}{D} \rceil D + P'(2,2) + \lceil \frac{C_{max}^{(1)}(opt)}{D} \rceil D + P'(2,3) + P'(1,2) + P'(1,3) \\
& \quad + (C_{max}^{(1)}(opt) - \lfloor \frac{C_{max}^{(1)}(opt)}{D} \rfloor D) \\
& \leq 2C_{max}^{(2)}(opt) + 3(C_{max}^{(1)}(opt) - \lfloor \frac{C_{max}^{(1)}(opt)}{D} \rfloor D).
\end{aligned}$$

$$Z(\eta) \leq 2Z(opt) + (C_{max}^{(1)}(opt) - \lfloor \frac{C_{max}^{(1)}(opt)}{D} \rfloor D). \tag{4.54}$$

Meanwhile,

$$\begin{aligned}
& \lceil \frac{C_{max}^{(2)}(\eta)}{D} \rceil D + l_1 D + D \leq C_{max}^{(2)}(\eta) + l_1 D + 2D \\
& = l_1 D + 2D + \lfloor \frac{C_{max}^{(1)}(opt)}{D} \rfloor D + P'(1,1) + P'(1,2) + P'(1,3) + \\
& \quad P'(2,2) + P'(2,3) \\
& = l_1 D + P'(1,1) + P'(1,2) + P'(2,2) + D + \lfloor \frac{C_{max}^{(1)}(opt)}{D} \rfloor D + \\
& \quad P'(1,3) + P'(2,3) + D \tag{4.55}
\end{aligned}$$

$$\begin{aligned}
& \leq C_{max}^{(2)}(opt) + D + C_{max}^{(1)}(opt) + P'(2,3) + D \\
& \leq C_{max}^{(2)}(opt) + D + \lceil \frac{C_{max}^{(1)}(opt)}{D} \rceil D + P'(2,3) + (C_{max}^{(1)}(opt) - \lfloor \frac{C_{max}^{(1)}(opt)}{D} \rfloor D) \\
& \leq 2C_{max}^{(2)}(opt) + D + (C_{max}^{(1)}(opt) - \lfloor \frac{C_{max}^{(1)}(opt)}{D} \rfloor D).
\end{aligned}$$

$$Z(\eta) \leq 2Z(opt) + D - (C_{max}^{(1)}(opt) - \lfloor \frac{C_{max}^{(1)}(opt)}{D} \rfloor D). \tag{4.56}$$

Therefore,

$$\begin{aligned} \frac{Z(\eta)}{Z(opt)} &\leq 2 + \min\left\{\frac{(C_{max}^{(1)}(opt) - \lfloor \frac{C_{max}^{(1)}(opt)}{D} \rfloor D)}{8D + (C_{max}^{(1)}(opt) - \lfloor \frac{C_{max}^{(1)}(opt)}{D} \rfloor D)}, \right. \\ &\quad \left. \frac{D - (C_{max}^{(1)}(opt) - \lfloor \frac{C_{max}^{(1)}(opt)}{D} \rfloor D)}{8D + (C_{max}^{(1)}(opt) - \lfloor \frac{C_{max}^{(1)}(opt)}{D} \rfloor D)}\right\} \\ &\leq 2 + \frac{1}{17}. \end{aligned} \tag{4.57}$$

As such, this lemma is proved. \square

Lemma 4.5. *If there is no idle delivery point between $C_{max}^{(1)}(\eta)$ and $C_{max}^{(2)}(\eta)$ and $(i_1, i_2, i_3) \neq (1, 2, 3)$, the algorithm is $2 + \frac{1}{16}$ competitive.*

Proof of Lemma 4.5: As $(i_1, i_2, i_3) \neq (1, 2, 3)$, there must be $i_2 = 3$.

Case 1: $C_{max}^{(i_2)}(opt) \leq C_{max}^{(1)}(\eta)$. Figure 4.5 shows the jobs processing in η under this situation.

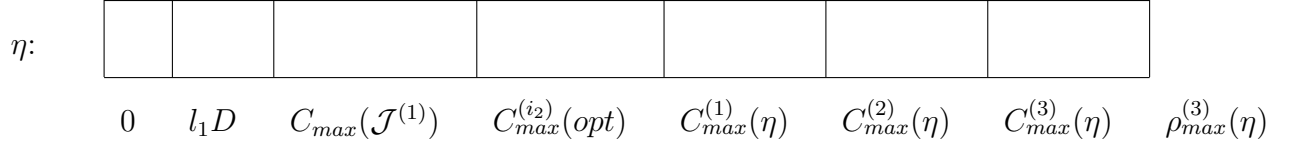


Figure 4.5. The Jobs Processing in η (3)

In the interval $(0, l_1D]$, there are at most three batches for every delivery point, so the deliver cost will not be more than $3l_1D$.

In the interval $(l_1D, C_{max}(\mathcal{J}^{(1)}))$, there are at most two batches for every delivery point, so the deliver cost will not be more than $2(\lfloor \frac{C_{max}(\mathcal{J}^{(1)})}{D} \rfloor - l_1)D$.

In the interval $(C_{max}(\mathcal{J}^{(1)}), C_{max}^{(i_2)}(opt))$, all the jobs of Customer 1 are released, so the jobs of Customer 2 and Customer 3 processed in this interval must be released before

$C_{max}(\mathcal{J}^{(1)})$, which implies that there are at most two batches in total for these two customers. Suppose that the delivery cost in this interval is sD , where s is a non-negative integer not greater than 2.

In the interval $(C_{max}^{(i_2)}(opt), \rho_{max}^{(3)}(\eta)]$, all the jobs are known and there are at most three batches, so the delivery cost will not be more than $3D$.

$$\begin{aligned} TC(\eta) &\leq 3l_1D + 2(\lfloor \frac{C_{max}(\mathcal{J}^{(1)})}{D} \rfloor - l_1)D + sD + 3D \\ &= l_1D + \lfloor \frac{C_{max}(\mathcal{J}^{(1)})}{D} \rfloor D + (s + 3)D. \end{aligned} \quad (4.58)$$

Therefore,

$$\begin{aligned} Z(\eta) &= \lceil \frac{C_{max}^{(1)}(\eta)}{D} \rceil D + \lceil \frac{C_{max}^{(2)}(\eta)}{D} \rceil D + \lceil \frac{C_{max}^{(3)}(\eta)}{D} \rceil D + TC(\eta) \\ &= 2C_{max}(\mathcal{J}^{(1)}) + (s + 3)D + \lceil \frac{C_{max}^{(1)}(\eta)}{D} \rceil D + \lceil \frac{C_{max}^{(2)}(\eta)}{D} \rceil D + \lceil \frac{C_{max}^{(3)}(\eta)}{D} \rceil D + \\ &\quad l_1D - 2(C_{max}(\mathcal{J}^{(1)}) - \lfloor \frac{C_{max}(\mathcal{J}^{(1)})}{D} \rfloor D) \\ &\leq 2C_{max}(\mathcal{J}^{(1)}) + 2C_{max}^{(3)}(\eta) + 6D + C_{max}^{(1)}(\eta) + l_1D + sD - \\ &\quad 2(C_{max}(\mathcal{J}^{(1)}) - \lfloor \frac{C_{max}(\mathcal{J}^{(1)})}{D} \rfloor D). \end{aligned} \quad (4.59)$$

As all the jobs are known after $C_{max}^{(i_2)}(opt)$, the jobs processing in the interval $(C_{max}^{(i_2)}(opt), C_{max}^{(1)}(\eta)]$ are continuous and all for Customer 1.

$$\begin{aligned} l_1D + C_{max}^{(1)}(\eta) &\leq l_1D + C_{max}^{(i_2)}(opt) + C_{max}(\mathcal{J}^{(1)}) - l_1D \\ &= C_{max}^{(i_2)}(opt) + C_{max}(\mathcal{J}^{(1)}) \leq 2C_{max}^{(i_2)}(opt). \end{aligned} \quad (4.60)$$

In addition, if $s \geq 1$, it must be satisfied that $C_{max}^{(i_2)}(opt) \geq \lceil \frac{C_{max}(\mathcal{J}^{(1)})}{D} \rceil D \geq l_1D + D$.

Case 1.1: $s = 0$.

$$\begin{aligned}
Z(\eta) &\leq 2C_{max}(\mathcal{J}^{(1)}) + 2C_{max}^{(3)}(\eta) + 6D + C_{max}^{(1)}(\eta) + l_1D - \\
&\quad 2(C_{max}(\mathcal{J}^{(1)}) - \lfloor \frac{C_{max}(\mathcal{J}^{(1)})}{D} \rfloor D) \\
&\leq 2C_{max}^{(i_1)}(opt) + 2C_{max}^{(i_2)}(opt) + 2C_{max}^{(i_3)}(opt) + 6D - \\
&\quad 2(C_{max}(\mathcal{J}^{(1)}) - \lfloor \frac{C_{max}(\mathcal{J}^{(1)})}{D} \rfloor D) \\
&\leq 2Z(opt).
\end{aligned} \tag{4.61}$$

Case 1.2: $s = 1$.

$$\begin{aligned}
&l_1D + D + C_{max}^{(1)}(\eta) - 2(C_{max}(\mathcal{J}^{(1)}) - \lfloor \frac{C_{max}(\mathcal{J}^{(1)})}{D} \rfloor D) \\
&\leq C_{max}^{(i_2)}(opt) + C_{max}(\mathcal{J}^{(1)}) + D - 2(C_{max}(\mathcal{J}^{(1)}) - \lfloor \frac{C_{max}(\mathcal{J}^{(1)})}{D} \rfloor D) \\
&= C_{max}^{(i_2)}(opt) + \lceil \frac{C_{max}(\mathcal{J}^{(1)})}{D} \rceil D - (C_{max}(\mathcal{J}^{(1)}) - \lfloor \frac{C_{max}(\mathcal{J}^{(1)})}{D} \rfloor D).
\end{aligned} \tag{4.62}$$

$$\begin{aligned}
Z(\eta) &\leq 2C_{max}(\mathcal{J}^{(1)}) + 2C_{max}^{(3)}(\eta) + 6D + D + C_{max}^{(1)}(\eta) + l_1D - \\
&\quad 2(C_{max}(\mathcal{J}^{(1)}) - \lfloor \frac{C_{max}(\mathcal{J}^{(1)})}{D} \rfloor D) \\
&\leq 2C_{max}^{(i_1)}(opt) + 2C_{max}^{(i_2)}(opt) + 2C_{max}^{(i_3)}(opt) + 6D - \\
&\quad (C_{max}(\mathcal{J}^{(1)}) - \lfloor \frac{C_{max}(\mathcal{J}^{(1)})}{D} \rfloor D) \\
&\leq 2Z(opt).
\end{aligned} \tag{4.63}$$

Case 1.3: $s = 2$.

Case 1.3.1: $i_1 = 2$.

If the two batches in the interval $(C_{max}(\mathcal{J}^{(1)}), C_{max}^{(i_2)}(opt)]$ are both delivered at the time $\lceil \frac{C_{max}(\mathcal{J}^{(1)})}{D} \rceil D$, there is $\lceil \frac{C_{max}(\mathcal{J}^{(1)})}{D} \rceil D \leq C_{max}(\mathcal{J}^{(2)})$ and $C_{max}^{(1)}(\eta) \leq \lceil \frac{C_{max}(\mathcal{J}^{(1)})}{D} \rceil D + C_{max}(\mathcal{J}^{(1)}) - l_1D$.

$$\begin{aligned}
Z(\eta) &\leq 2C_{max}(\mathcal{J}^{(1)}) + 2C_{max}^{(3)}(\eta) + 6D + 2D + C_{max}^{(1)}(\eta) + l_1D - \\
&\quad 2(C_{max}(\mathcal{J}^{(1)}) - \lfloor \frac{C_{max}(\mathcal{J}^{(1)})}{D} \rfloor D) \\
&\leq 2C_{max}(\mathcal{J}^{(1)}) + 2C_{max}^{(3)}(\eta) + 6D + 2D + \\
&\quad \lceil \frac{C_{max}(\mathcal{J}^{(1)})}{D} \rceil D + C_{max}(\mathcal{J}^{(1)}) - 2(C_{max}(\mathcal{J}^{(1)}) - \lfloor \frac{C_{max}(\mathcal{J}^{(1)})}{D} \rfloor D) \\
&= 2C_{max}^{(3)}(\eta) + 6D + \lceil \frac{C_{max}(\mathcal{J}^{(1)})}{D} \rceil D + C_{max}(\mathcal{J}^{(1)}) + 2D + 2\lfloor \frac{C_{max}(\mathcal{J}^{(1)})}{D} \rfloor D \\
&\leq 2C_{max}^{(3)}(\eta) + 6D + 2C_{max}(\mathcal{J}^{(2)}) + 2\lceil \frac{C_{max}(\mathcal{J}^{(1)})}{D} \rceil D \\
&\leq 2C_{max}^{(i_3)}(opt) + 6D + 2C_{max}^{(i_1)}(opt) + 2C_{max}^{(i_2)}(opt) \\
&= 2Z(opt).
\end{aligned} \tag{4.64}$$

If at least one batch in the interval $(C_{max}(\mathcal{J}^{(1)}), C_{max}^{(i_2)}(opt)]$ is delivered later than $\lceil \frac{C_{max}(\mathcal{J}^{(1)})}{D} \rceil D$, then $C_{max}^{(i_2)}(opt) \geq \lceil \frac{C_{max}(\mathcal{J}^{(1)})}{D} \rceil D + D$ and $C_{max}^{(1)}(\eta) \leq C_{max}^{(i_2)}(opt) + C_{max}(\mathcal{J}^{(1)}) - l_1D$.

$$\begin{aligned}
Z(\eta) &\leq 2C_{max}(\mathcal{J}^{(1)}) + 2C_{max}^{(3)}(\eta) + 6D + 2D + C_{max}^{(1)}(\eta) + l_1D - \\
&\quad 2(C_{max}(\mathcal{J}^{(1)}) - \lfloor \frac{C_{max}(\mathcal{J}^{(1)})}{D} \rfloor D) \\
&\leq 2C_{max}(\mathcal{J}^{(1)}) + 2C_{max}^{(3)}(\eta) + 6D + 2D + C_{max}^{(i_2)}(opt) + \\
&\quad C_{max}(\mathcal{J}^{(1)}) - 2(C_{max}(\mathcal{J}^{(1)}) - \lfloor \frac{C_{max}(\mathcal{J}^{(1)})}{D} \rfloor D) \\
&= 2C_{max}(\mathcal{J}^{(1)}) + 2C_{max}^{(3)}(\eta) + 6D + C_{max}^{(i_2)}(opt) + \lfloor \frac{C_{max}(\mathcal{J}^{(1)})}{D} \rfloor D + \\
&\quad 2D - (C_{max}(\mathcal{J}^{(1)}) - \lfloor \frac{C_{max}(\mathcal{J}^{(1)})}{D} \rfloor D) \\
&\leq 2C_{max}(\mathcal{J}^{(1)}) + 2C_{max}^{(3)}(\eta) + 6D + 2C_{max}^{(i_2)}(opt) - \\
&\quad (C_{max}(\mathcal{J}^{(1)}) - \lfloor \frac{C_{max}(\mathcal{J}^{(1)})}{D} \rfloor D) \\
&\leq 2C_{max}^{(i_1)}(opt) + 2C_{max}^{(i_3)}(opt) + 6D + 2C_{max}^{(i_2)}(opt) = 2Z(opt).
\end{aligned} \tag{4.65}$$

Case 1.3.2: $i_1 = 1$.

If at least one batch in the interval $(C_{max}(\mathcal{J}^{(1)}), C_{max}^{(i_2)}(opt))$ is delivered later than $\lceil \frac{C_{max}(\mathcal{J}^{(1)})}{D} \rceil D$, it is similar to the above case and there is $Z(\eta) \leq 2Z(opt)$.

If the two batches in the interval $(C_{max}(\mathcal{J}^{(1)}), C_{max}^{(i_2)}(opt))$ are both delivered at the time $\lceil \frac{C_{max}(\mathcal{J}^{(1)})}{D} \rceil D$, the batches at the delivery point $\lfloor \frac{C_{max}(\mathcal{J}^{(1)})}{D} \rfloor D$ need to be investigated.

Case 1.3.2.1: $C_{max}(\mathcal{J}^{(1)}) > l_1 D + D$. In this case, we have $C_{max}^{(i_2)}(opt) \geq l_1 D + 2D$. If there is at most one batch at the delivery point $\lfloor \frac{C_{max}(\mathcal{J}^{(1)})}{D} \rfloor D$, it is similar to the cases $s = 0$ and $s = 1$, and we have $Z(\eta) \leq Z(opt)$. Next, the case that there are two batches at the delivery point $\lfloor \frac{C_{max}(\mathcal{J}^{(1)})}{D} \rfloor D$ is explored.

Suppose that the two batches are for Customer 1 and Customer 2, or for Customer 1 and Customer 3.

$$\begin{aligned} C_{max}^{(1)}(\eta) + l_1 D + 2D &\leq C_{max}^{(i_2)}(opt) + C_{max}(\mathcal{J}^{(1)}) - \lfloor \frac{C_{max}(\mathcal{J}^{(1)})}{D} \rfloor D + l_1 D + 2D \\ &\leq 2C_{max}^{(i_2)}(opt) + (C_{max}(\mathcal{J}^{(1)}) - \lfloor \frac{C_{max}(\mathcal{J}^{(1)})}{D} \rfloor D). \end{aligned} \quad (4.66)$$

$$\begin{aligned} Z(\eta) &\leq 2C_{max}(\mathcal{J}^{(1)}) + 2C_{max}^{(3)}(\eta) + 6D + 2D + C_{max}^{(1)}(\eta) + l_1 D - \\ &\quad 2(C_{max}(\mathcal{J}^{(1)}) - \lfloor \frac{C_{max}(\mathcal{J}^{(1)})}{D} \rfloor D) \\ &\leq 2C_{max}(\mathcal{J}^{(1)}) + 2C_{max}^{(3)}(\eta) + 6D + 2C_{max}^{(i_2)}(opt) + \\ &\quad (C_{max}(\mathcal{J}^{(1)}) - \lfloor \frac{C_{max}(\mathcal{J}^{(1)})}{D} \rfloor D) \\ &\quad - 2(C_{max}(\mathcal{J}^{(1)}) - \lfloor \frac{C_{max}(\mathcal{J}^{(1)})}{D} \rfloor D) \\ &\leq 2C_{max}^{(i_1)}(opt) + 2C_{max}^{(i_3)}(opt) + 6D + 2C_{max}^{(i_2)}(opt) \\ &= 2Z(opt). \end{aligned} \quad (4.67)$$

Suppose that the two batches are for Customer 2 and Customer 3.

$$\begin{aligned}
& C_{max}^{(1)}(\eta) + l_1 D + 2D \\
& \leq \lfloor \frac{C_{max}(\mathcal{J}^{(1)})}{D} \rfloor D + 2(C_{max}(\mathcal{J}^{(1)}) - \lfloor \frac{C_{max}(\mathcal{J}^{(1)})}{D} \rfloor D) + \\
& \quad C_{max}(\mathcal{J}^{(1)}) - l_1 D + l_1 D + 2D \\
& \leq \lceil \frac{C_{max}(\mathcal{J}^{(1)})}{D} \rceil D + C_{max}(\mathcal{J}^{(1)}) + 2(C_{max}(\mathcal{J}^{(1)}) - \lfloor \frac{C_{max}(\mathcal{J}^{(1)})}{D} \rfloor D) + D \\
& \leq 2\lceil \frac{C_{max}(\mathcal{J}^{(1)})}{D} \rceil D + 3(C_{max}(\mathcal{J}^{(1)}) - \lfloor \frac{C_{max}(\mathcal{J}^{(1)})}{D} \rfloor D) \\
& \leq 2C_{max}^{(i_2)}(opt) + 3(C_{max}(\mathcal{J}^{(1)}) - \lfloor \frac{C_{max}(\mathcal{J}^{(1)})}{D} \rfloor D)
\end{aligned} \tag{4.68}$$

$$\begin{aligned}
Z(\eta) & \leq 2C_{max}(\mathcal{J}^{(1)}) + 2C_{max}^{(3)}(\eta) + 6D + 2D + C_{max}^{(1)}(\eta) + l_1 D - \\
& \quad 2(C_{max}(\mathcal{J}^{(1)}) - \lfloor \frac{C_{max}(\mathcal{J}^{(1)})}{D} \rfloor D) \\
& \leq 2C_{max}(\mathcal{J}^{(1)}) + 2C_{max}^{(3)}(\eta) + 6D + 2C_{max}^{(i_2)}(opt) + 3(C_{max}(\mathcal{J}^{(1)}) - \\
& \quad \lfloor \frac{C_{max}(\mathcal{J}^{(1)})}{D} \rfloor D) - 2(C_{max}(\mathcal{J}^{(1)}) - \lfloor \frac{C_{max}(\mathcal{J}^{(1)})}{D} \rfloor D) \\
& = 2C_{max}^{(i_1)}(opt) + 2C_{max}^{(i_3)}(opt) + 6D + 2C_{max}^{(i_2)}(opt) + \\
& \quad (C_{max}(\mathcal{J}^{(1)}) - \lfloor \frac{C_{max}(\mathcal{J}^{(1)})}{D} \rfloor D) \\
& = 2Z(opt) + (C_{max}(\mathcal{J}^{(1)}) - \lfloor \frac{C_{max}(\mathcal{J}^{(1)})}{D} \rfloor D).
\end{aligned} \tag{4.69}$$

Meanwhile,

$$\begin{aligned}
& C_{max}^{(1)}(\eta) + l_1 D + 2D \\
& \leq C_{max}^{(i_2)}(opt) + C_{max}(\mathcal{J}^{(1)}) - l_1 D + l_1 D + 2D \\
& = C_{max}^{(i_2)}(opt) + C_{max}(\mathcal{J}^{(1)}) + D + D \\
& = C_{max}^{(i_2)}(opt) + \lceil \frac{C_{max}(\mathcal{J}^{(1)})}{D} \rceil D + (C_{max}(\mathcal{J}^{(1)}) - \lfloor \frac{C_{max}(\mathcal{J}^{(1)})}{D} \rfloor D) + D \\
& \leq 2C_{max}^{(i_2)}(opt) + D + (C_{max}(\mathcal{J}^{(1)}) - \lfloor \frac{C_{max}(\mathcal{J}^{(1)})}{D} \rfloor D)
\end{aligned} \tag{4.70}$$

$$\begin{aligned}
Z(\eta) &\leq 2C_{max}(\mathcal{J}^{(1)}) + 2C_{max}^{(3)}(\eta) + 6D + 2D + C_{max}^{(1)}(\eta) + l_1D - \\
&\quad 2(C_{max}(\mathcal{J}^{(1)}) - \lfloor \frac{C_{max}(\mathcal{J}^{(1)})}{D} \rfloor D) \\
&\leq 2C_{max}(\mathcal{J}^{(1)}) + 2C_{max}^{(3)}(\eta) + 6D + 2C_{max}^{(i_2)}(opt) + D + \\
&\quad (C_{max}(\mathcal{J}^{(1)}) - \lfloor \frac{C_{max}(\mathcal{J}^{(1)})}{D} \rfloor D) - 2(C_{max}(\mathcal{J}^{(1)}) - \lfloor \frac{C_{max}(\mathcal{J}^{(1)})}{D} \rfloor D) \quad (4.71) \\
&= 2C_{max}^{(i_1)}(opt) + 2C_{max}^{(i_3)}(opt) + 6D + 2C_{max}^{(i_2)}(opt) + D - \\
&\quad (C_{max}(\mathcal{J}^{(1)}) - \lfloor \frac{C_{max}(\mathcal{J}^{(1)})}{D} \rfloor D) \\
&= 2Z(opt) + D - (C_{max}(\mathcal{J}^{(1)}) - \lfloor \frac{C_{max}(\mathcal{J}^{(1)})}{D} \rfloor D).
\end{aligned}$$

As $C_{max}^{(i_1)}(opt) = C_{max}(\mathcal{J}^{(1)}) \geq l_1D + D + (C_{max}(\mathcal{J}^{(1)}) - \lfloor \frac{C_{max}(\mathcal{J}^{(1)})}{D} \rfloor D)$ and $C_{max}^{(i_3)}(opt) \geq C_{max}^{(i_2)}(opt) \geq l_1D + 2D$, there is $Z(opt) = C_{max}^{(i_1)}(opt) + C_{max}^{(i_2)}(opt) + C_{max}^{(i_3)}(opt) + 3D \geq 8D + (C_{max}(\mathcal{J}^{(1)}) - \lfloor \frac{C_{max}(\mathcal{J}^{(1)})}{D} \rfloor D)$.

Therefore,

$$\begin{aligned}
\frac{Z(\eta)}{Z(opt)} &\leq 2 + \min \left\{ \frac{(C_{max}(\mathcal{J}^{(1)}) - \lfloor \frac{C_{max}(\mathcal{J}^{(1)})}{D} \rfloor D)}{8D + (C_{max}(\mathcal{J}^{(1)}) - \lfloor \frac{C_{max}(\mathcal{J}^{(1)})}{D} \rfloor D)}, \right. \\
&\quad \left. \frac{D - (C_{max}(\mathcal{J}^{(1)}) - \lfloor \frac{C_{max}(\mathcal{J}^{(1)})}{D} \rfloor D)}{8D + (C_{max}(\mathcal{J}^{(1)}) - \lfloor \frac{C_{max}(\mathcal{J}^{(1)})}{D} \rfloor D)} \right\} \quad (4.72) \\
&\leq 2 + \frac{1}{17}.
\end{aligned}$$

Case 1.3.2.2: $C_{max}(\mathcal{J}^{(1)}) \leq l_1D + D$. As there are batches in the interval $(C_{max}(\mathcal{J}^{(1)}), C_{max}^{(i_2)}(opt)]$, $C_{max}^{(i_2)}(opt) \geq l_1D + D$ and $C_{max}^{(1)}(\eta) \geq l_1D + D$. Then, $Z(opt) = C_{max}^{(i_1)}(opt) + C_{max}^{(i_2)}(opt) + C_{max}^{(i_3)}(opt) + 3D \geq 5D + (C_{max}(\mathcal{J}^{(1)}) - \lfloor \frac{C_{max}(\mathcal{J}^{(1)})}{D} \rfloor D)$

If $C_{max}^{(1)}(\eta) \leq l_1D + 2D$, $\lceil \frac{C_{max}(\mathcal{J}^{(1)})}{D} \rceil D = l_1D + 2D$. As $C_{max}^{(1)}(\eta) \geq l_1D + D$, $C_{max}(\mathcal{J}^{(1)}) - \lfloor \frac{C_{max}(\mathcal{J}^{(1)})}{D} \rfloor D \geq \frac{1}{3}D$

$$\lceil \frac{C_{max}^{(1)}(\eta)}{D} \rceil D + l_1D + D = 2l_1D + 3D \leq 2C_{max}^{(i_2)}(opt) + D. \quad (4.73)$$

$$\begin{aligned}
Z(\eta) &\leq 2C_{max}(\mathcal{J}^{(1)}) + 2C_{max}^{(3)}(\eta) + 6D + D + \lceil \frac{C_{max}^{(1)}(\eta)}{D} \rceil D + \\
&\quad l_1 D - 2(C_{max}(\mathcal{J}^{(1)}) - \lfloor \frac{C_{max}(\mathcal{J}^{(1)})}{D} \rfloor D) \\
&\leq 2C_{max}(\mathcal{J}^{(1)}) + 2C_{max}^{(3)}(\eta) + 6D + 2C_{max}^{(i_2)}(opt) + \\
&\quad D - 2(C_{max}(\mathcal{J}^{(1)}) - \lfloor \frac{C_{max}(\mathcal{J}^{(1)})}{D} \rfloor D) \\
&= 2Z(opt) + D - 2(C_{max}(\mathcal{J}^{(1)}) - \lfloor \frac{C_{max}(\mathcal{J}^{(1)})}{D} \rfloor D).
\end{aligned} \tag{4.74}$$

Therefore,

$$\begin{aligned}
\frac{Z(\eta)}{Z(opt)} &\leq 2 + \frac{D - 2(C_{max}(\mathcal{J}^{(1)}) - \lfloor \frac{C_{max}(\mathcal{J}^{(1)})}{D} \rfloor D)}{5D + (C_{max}(\mathcal{J}^{(1)}) - \lfloor \frac{C_{max}(\mathcal{J}^{(1)})}{D} \rfloor D)} \\
&\leq 2 + \frac{1}{16}.
\end{aligned} \tag{4.75}$$

If $C_{max}^{(1)}(\eta) > l_1 D + 2D$, $C_{max}(\mathcal{J}^{(1)}) - \lfloor \frac{C_{max}(\mathcal{J}^{(1)})}{D} \rfloor D \geq \frac{2}{3}D$.

$$\begin{aligned}
&C_{max}^{(1)}(\eta) + l_1 D + 2D \\
&\leq \lceil \frac{C_{max}(\mathcal{J}^{(1)})}{D} \rceil D + C_{max}(\mathcal{J}^{(1)}) - l_1 D + l_1 D + 2D \\
&\leq \lceil \frac{C_{max}(\mathcal{J}^{(1)})}{D} \rceil D + C_{max}(\mathcal{J}^{(1)}) + 2D \\
&\leq 2\lceil \frac{C_{max}(\mathcal{J}^{(1)})}{D} \rceil D + D + (C_{max}(\mathcal{J}^{(1)}) - \lfloor \frac{C_{max}(\mathcal{J}^{(1)})}{D} \rfloor D) \\
&\leq 2C_{max}^{(i_2)}(opt) + D + (C_{max}(\mathcal{J}^{(1)}) - \lfloor \frac{C_{max}(\mathcal{J}^{(1)})}{D} \rfloor D).
\end{aligned} \tag{4.76}$$

$$\begin{aligned}
Z(\eta) &\leq 2C_{max}(\mathcal{J}^{(1)}) + 2C_{max}^{(3)}(\eta) + 6D + 2D + C_{max}^{(1)}(\eta) + \\
&\quad l_1 D - 2(C_{max}(\mathcal{J}^{(1)}) - \lfloor \frac{C_{max}(\mathcal{J}^{(1)})}{D} \rfloor D) \\
&\leq 2Z(opt) + D - (C_{max}(\mathcal{J}^{(1)}) - \lfloor \frac{C_{max}(\mathcal{J}^{(1)})}{D} \rfloor D).
\end{aligned} \tag{4.77}$$

Therefore,

$$\begin{aligned}
\frac{Z(\eta)}{Z(opt)} &\leq 2 + \frac{D - (C_{max}(\mathcal{J}^{(1)}) - \lfloor \frac{C_{max}(\mathcal{J}^{(1)})}{D} \rfloor D)}{5D + (C_{max}(\mathcal{J}^{(1)}) - \lfloor \frac{C_{max}(\mathcal{J}^{(1)})}{D} \rfloor D)} \\
&\leq 2 + \frac{1}{17}.
\end{aligned} \tag{4.78}$$

Case 2: $C_{max}^{(i_2)}(opt) > C_{max}^{(1)}(\eta)$. Figure 4.6 shows the jobs processing in η under this situation.

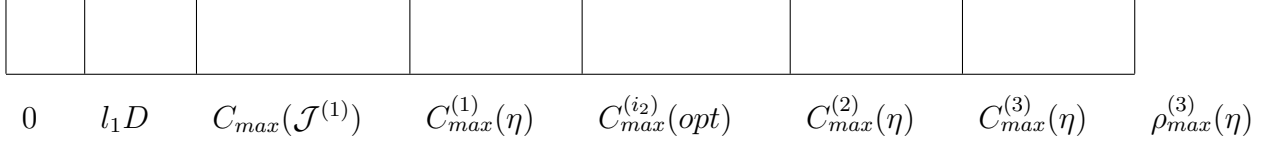


Figure 4.6. The Jobs Processing in η (4)

In the interval $(0, l_1D]$, there are at most three batches for every delivery point, so the deliver cost will not be more than $3l_1D$.

In the interval $(l_1D, C_{max}(\mathcal{J}^{(1)}])$, there are at most two batches for every delivery point, so the deliver cost will not be more than $2(\lfloor \frac{C_{max}(\mathcal{J}^{(1)})}{D} \rfloor - l_1)D$.

In the interval $(C_{max}(\mathcal{J}^{(1)}), C_{max}^{(1)}(\eta)]$, all the jobs of Customer 1 are released, so the jobs of Customer 2 and Customer 3 processed in this interval must be released before $C_{max}(\mathcal{J}^{(1)})$, which implies that there are at most two batches in total for these two aid sites. Suppose that the delivery cost in this interval is sD , where s is a non-negative integer not greater than 2.

In the interval $(C_{max}^{(1)}(\eta), C_{max}^{(i_2)}(opt)]$, there is at most one batch for every delivery point, and there is one batch in total for customer 1, so the delivery cost will not be more than $(\lfloor \frac{C_{max}^{(i_2)}(opt)}{D} \rfloor - \lfloor \frac{C_{max}^{(1)}(\eta)}{D} \rfloor)D + D$

In the interval $(C_{max}^{(i_2)}(opt), \rho_{max}^{(3)}(\eta)]$, all the jobs are known and there are at most two batches, so the delivery cost will not be more than $2D$.

$$\begin{aligned}
TC(\eta) &\leq 3l_1D + 2(\lfloor \frac{C_{max}(\mathcal{J}^{(1)})}{D} \rfloor - l_1)D + sD + \\
&\quad (\lfloor \frac{C_{max}^{(i_2)}(opt)}{D} \rfloor - \lfloor \frac{C_{max}^{(1)}(\eta)}{D} \rfloor)D + D + 2D \\
&= l_1D + 2\lfloor \frac{C_{max}(\mathcal{J}^{(1)})}{D} \rfloor D + (s+3)D + (\lfloor \frac{C_{max}^{(i_2)}(opt)}{D} \rfloor - \lfloor \frac{C_{max}^{(1)}(\eta)}{D} \rfloor)D.
\end{aligned} \tag{4.79}$$

Therefore,

$$\begin{aligned}
Z(\eta) &= \lceil \frac{C_{max}^{(1)}(\eta)}{D} \rceil D + \lceil \frac{C_{max}^{(2)}(\eta)}{D} \rceil D + \lceil \frac{C_{max}^{(3)}(\eta)}{D} \rceil D + TC(\eta) \\
&= 2C_{max}(\mathcal{J}^{(1)}) + 2C_{max}^{(3)}(\eta) + 6D + (s-1)D + l_1D + \lceil \frac{C_{max}^{(1)}(\eta)}{D} \rceil D + \\
&\quad (\lfloor \frac{C_{max}^{(i_2)}(opt)}{D} \rfloor - \lfloor \frac{C_{max}^{(1)}(\eta)}{D} \rfloor)D - 2(C_{max}(\mathcal{J}^{(1)}) - \lfloor \frac{C_{max}(\mathcal{J}^{(1)})}{D} \rfloor D) \\
&\leq 2C_{max}(\mathcal{J}^{(1)}) + 2C_{max}^{(3)}(\eta) + 6D + sD + l_1D + \lfloor \frac{C_{max}^{(i_2)}(opt)}{D} \rfloor D - \\
&\quad 2(C_{max}(\mathcal{J}^{(1)}) - \lfloor \frac{C_{max}(\mathcal{J}^{(1)})}{D} \rfloor D).
\end{aligned} \tag{4.80}$$

Case 2.1: $s = 0$.

As $l_1D + \lfloor \frac{C_{max}^{(i_2)}(opt)}{D} \rfloor D \leq 2C_{max}^{(i_2)}(opt)$,

$$\begin{aligned}
Z(\eta) &\leq 2C_{max}(\mathcal{J}^{(1)}) + 2C_{max}^{(3)}(\eta) + 6D + l_1D + \lfloor \frac{C_{max}^{(i_2)}(opt)}{D} \rfloor D - 2(C_{max}(\mathcal{J}^{(1)}) - \\
&\quad \lfloor \frac{C_{max}(\mathcal{J}^{(1)})}{D} \rfloor D) \\
&\leq 2C_{max}(\mathcal{J}^{(1)}) + 2C_{max}^{(3)}(\eta) + 6D + 2C_{max}^{(i_2)}(opt) - \\
&\quad 2(C_{max}(\mathcal{J}^{(1)}) - \lfloor \frac{C_{max}(\mathcal{J}^{(1)})}{D} \rfloor D) \\
&\leq 2Z(opt).
\end{aligned} \tag{4.81}$$

Case 2.2: $s = 1$. As there is one batch in the interval $(C_{max}(\mathcal{J}^{(1)}), C_{max}^{(1)}(\eta)]$, there is

$C_{max}^{(1)}(\eta) \geq l_1D + D$ and $C_{max}^{(i_2)}(opt) \geq l_1D + D$ which implies $l_1D + D + \lfloor \frac{C_{max}^{(i_2)}(opt)}{D} \rfloor D \leq 2C_{max}^{(i_2)}(opt)$.

$$\begin{aligned}
Z(\eta) &\leq 2C_{max}(\mathcal{J}^{(1)}) + 2C_{max}^{(3)}(\eta) + 6D + D + l_1D + \\
&\quad \lfloor \frac{C_{max}^{(i_2)}(opt)}{D} \rfloor D - 2(C_{max}(\mathcal{J}^{(1)}) - \lfloor \frac{C_{max}(\mathcal{J}^{(1)})}{D} \rfloor D) \\
&\leq 2C_{max}(\mathcal{J}^{(1)}) + 2C_{max}^{(3)}(\eta) + 6D + 2C_{max}^{(i_2)}(opt) - \\
&\quad 2(C_{max}(\mathcal{J}^{(1)}) - \lfloor \frac{C_{max}(\mathcal{J}^{(1)})}{D} \rfloor D) \\
&\leq 2Z(opt).
\end{aligned} \tag{4.82}$$

Case 2.3: $s = 2$. Similarly, $C_{max}^{(i_2)}(opt) \geq l_1D + D$.

Case 2.3.1: $C_{max}^{(i_2)}(opt) \geq l_1D + 2D$. In this case, $l_1D + 2D + \lfloor \frac{C_{max}^{(i_2)}(opt)}{D} \rfloor D \leq 2C_{max}^{(i_2)}(opt)$.

$$\begin{aligned}
Z(\eta) &\leq 2C_{max}(\mathcal{J}^{(1)}) + 2C_{max}^{(3)}(\eta) + 6D + 2D + l_1D + \\
&\quad \lfloor \frac{C_{max}^{(i_2)}(opt)}{D} \rfloor D - 2(C_{max}(\mathcal{J}^{(1)}) - \lfloor \frac{C_{max}(\mathcal{J}^{(1)})}{D} \rfloor D) \\
&\leq 2C_{max}(\mathcal{J}^{(1)}) + 2C_{max}^{(3)}(\eta) + 6D + 2C_{max}^{(i_2)}(opt) - \\
&\quad 2(C_{max}(\mathcal{J}^{(1)}) - \lfloor \frac{C_{max}(\mathcal{J}^{(1)})}{D} \rfloor D) \\
&\leq 2Z(opt).
\end{aligned} \tag{4.83}$$

Case 2.3.2: $C_{max}^{(i_2)}(opt) < l_1D + 2D$. As $C_{max}(\mathcal{J}^{(1)}) \leq l_1D + D$ and $C_{max}^{(1)}(\eta) \geq l_1D + D$, there is $(C_{max}(\mathcal{J}^{(1)}) - \lfloor \frac{C_{max}(\mathcal{J}^{(1)})}{D} \rfloor D) \geq \frac{1}{3}D$. Then, $Z(opt) = C_{max}^{(i_1)}(opt) + C_{max}^{(i_2)}(opt) + C_{max}^{(i_3)}(opt) + 3D \geq 5D + (C_{max}(\mathcal{J}^{(1)}) - \lfloor \frac{C_{max}(\mathcal{J}^{(1)})}{D} \rfloor D)$.

As $l_1D + D + \lfloor \frac{C_{max}^{(i_2)}(opt)}{D} \rfloor D \leq 2C_{max}^{(i_2)}(opt)$,

$$\begin{aligned}
Z(\eta) &\leq 2C_{max}(\mathcal{J}^{(1)}) + 2C_{max}^{(3)}(\eta) + 6D + 2D + l_1D + \lfloor \frac{C_{max}^{(i_2)}(opt)}{D} \rfloor D - \\
&\quad 2(C_{max}(\mathcal{J}^{(1)}) - \lfloor \frac{C_{max}(\mathcal{J}^{(1)})}{D} \rfloor D) \\
&\leq 2C_{max}(\mathcal{J}^{(1)}) + 2C_{max}^{(3)}(\eta) + 6D + 2C_{max}^{(i_2)}(opt) + D - \\
&\quad 2(C_{max}(\mathcal{J}^{(1)}) - \lfloor \frac{C_{max}(\mathcal{J}^{(1)})}{D} \rfloor D) \\
&\leq 2Z(opt) + D - 2(C_{max}(\mathcal{J}^{(1)}) - \lfloor \frac{C_{max}(\mathcal{J}^{(1)})}{D} \rfloor D).
\end{aligned} \tag{4.84}$$

$$\begin{aligned}
\frac{Z(\eta)}{Z(opt)} &\leq 2 + \frac{D - 2(C_{max}(\mathcal{J}^{(1)}) - \lfloor \frac{C_{max}(\mathcal{J}^{(1)})}{D} \rfloor D)}{5D + (C_{max}(\mathcal{J}^{(1)}) - \lfloor \frac{C_{max}(\mathcal{J}^{(1)})}{D} \rfloor D)} \\
&\leq 2 + \frac{1}{16}.
\end{aligned} \tag{4.85}$$

As such, this lemma is proved. \square

Proof of Theorem 4.3: From the above three lemmas, one can conclude that the competitive ratio of SMH2 is not greater than $2 + \frac{2}{27}$ for $k = 3$. At last, it only needs to show that there is an instance such that the ratio of the algorithm result to the optimal result can achieve $2 + \frac{2}{27}$. The instance can be constructed as follows: for Customer 1, there is one job $J_1^{(1)} = (0, \frac{1}{2} + \epsilon)$; for Customer 2, there are two jobs $J_1^{(2)} = (0, \frac{1}{2})$ and $J_2^{(2)} = (1, \frac{1}{4} + \epsilon)$; for Customer 3, there are three jobs $J_1^{(3)} = (0, \frac{1}{2} - \epsilon)$, $J_2^{(3)} = (1, \frac{1}{4})$, and $J_3^{(3)} = (2 + \epsilon, \epsilon)$; the delivery cost $D = 1$ (ϵ is a very small positive number). The algorithm will process the jobs in the order $(J_1^{(3)}, J_1^{(2)}, J_1^{(1)}, J_2^{(3)}, J_2^{(2)}, J_3^{(3)})$, deliver $J_1^{(3)}$ and $J_1^{(2)}$ in tow batches at time 1, deliver $J_1^{(1)}$ and $J_2^{(3)}$ in two batches at time 2 and deliver $J_2^{(2)}$ and $J_3^{(3)}$ at time 3, so $Z(\eta) = 2 + 3 + 3 + 6 = 14$ (see Figure 4.7). However, the off-line optimal schedule should process the jobs in the order $(J_1^{(1)}, J_1^{(2)}, J_2^{(2)}, J_1^{(3)}, J_2^{(3)}, J_3^{(3)})$, deliver three batches for the three customers respectively when their jobs are completed, so $Z(opt) = \frac{1}{2} + \epsilon + \frac{5}{4} + 2\epsilon + 2 + 2\epsilon + 3 = \frac{27}{4} + 5\epsilon$ (see Figure 4.7). Therefore, the ratio $\frac{Z(\eta)}{Z(opt)} = \frac{14}{\frac{27}{4} + 5\epsilon}$ will tend to $\frac{14}{\frac{27}{4}} = 2 + \frac{2}{27}$ as ϵ tends to 0.

η :	$J_1^{(3)}$	$J_1^{(2)}$	$J_1^{(1)}$	$J_2^{(3)}$	$J_2^{(2)}$	$J_3^{(3)}$	
	0	$\frac{1}{2} - \epsilon$	$1 - \epsilon$	$\frac{3}{2}$	$\frac{7}{4}$	$2 + \epsilon$	$2 + 2\epsilon$
opt :	$J_1^{(1)}$	$J_1^{(2)}$	$J_2^{(2)}$	$J_1^{(3)}$	$J_2^{(3)}$	$J_3^{(3)}$	
	0	$\frac{1}{2} + \epsilon$	$1 + \epsilon$	$\frac{5}{4} + 2\epsilon$	$\frac{7}{4} + \epsilon$	$2 + \epsilon$	$2 + 2\epsilon$

Figure 4.7. The Jobs Processing for The Worst Instance

This completes the proof of Theorem 4.3. □

For the case that k is greater than 3, the performance of the algorithm will be shown in the simulated experiment.

4.3 Algorithm for Problem SMP3

SMP3 has the following features: Jobs are released off-line and delivered in "direct" pattern. The capacity of vehicles and the number of vehicles are both enough.

Similar to the induction in Section 4.1, it can be shown that SMP3 is equivalent to the agent scheduling problem $1|r_j|\sum C_{max}^{(i)}$. The assumption that $T_{0i} = 0$ for all i and $D = 0$ can also be applied. As the preemption of job processing is forbidden, the intractability of the problem has increased. Actually, SMP3 is NP-hard even $k = 2$ [Ding and Sun, 2010]. When k is a parameter, SMP3 is SNP-hard and can be proved by polynomially transforming a SNPC problem to the decision version of SMP3.

Theorem 4.4. *The problem SMP3 is SNP-hard when k is a parameter.*

Proof of Theorem 4.4: This statement is proved by showing that 3-Partition problem (a famous SNPC problem [Garey and Johnson , 1975]) can be polynomially transformed to the decision version of SMP3.

3-Partition. Given a set of $3t$ positive integers $S = \{b_1, b_2, \dots, b_{3t}\}$, where $\sum b_l = tB$ and $\frac{B}{4} < b_l < \frac{B}{2}$, can S be partitioned in to t disjoint subsets S_1, S_2, \dots, S_t , which can further cover S such that the sum of the numbers in each subset is equal?

Construct an instance of the decision version of SMP3 from 3-Partition problem as follows. There are $t + 1$ aid sites. For Customer i ($i = 1, 2, \dots, t$), there is a job released at $r_1^{(i)} = iB + (i - 1)\epsilon$ with processing time $p_1^{(i)} = \epsilon$ where $\epsilon = \frac{\min\{|b_l - b_h|\}}{3t}$. For the $(t + 1)th$ customer, he releases $3t$ jobs at $r_j^{(t+1)} = 0$ with processing time $p_j^{(t+1)} = b_j$ ($j = 1, 2, \dots, 3t$), and the last job at $r_{3t+1}^{(t+1)} = tB + t\epsilon$ with processing time $p_{3t+1}^{(t+1)} = \epsilon$. The question is whether there is a feasible solution η such that $Z(\eta) \leq \frac{t(t+3)}{2}(B + \epsilon) + \epsilon$.

If 3-Partition problem has a solution, then the jobs of Customer i ($i = 1, 2, \dots, t$) can be processed immediately after they are released, while the jobs of Customer $t + 1$ can be scheduled on the machine without idle time. Therefore, $Z = \sum_{i=1}^t i(B + \epsilon) + t(B + \epsilon) + \epsilon = \frac{t(t+3)}{2}(B + \epsilon) + \epsilon$.

If the instance of decision version of SMP3 has a feasible solution η such that $Z(\eta) \leq \frac{t(t+3)}{2}(B + \epsilon) + \epsilon$, then it needs to show that the jobs of Customer i ($i = 1, 2, \dots, t$) cannot be delayed. As $C_{max}(\mathcal{J}^{(i)}) = iB + (i - 1)\epsilon + \epsilon = i(B + \epsilon)$ for $i = 1, 2, \dots, t$, and $C_{max}(\mathcal{J}^{(t+1)}) = tB + t\epsilon + \epsilon$, $Z(\eta) = \sum_{i=1}^{t+1} C_{max}^{(i)}(\eta) \geq \sum_{i=1}^t i(B + \epsilon) + tB + t\epsilon + \epsilon = \frac{t(t+3)}{2}(B + \epsilon) + \epsilon$. Therefore, $C_{max}^{(i)}(\eta) = i(B + \epsilon)$ for $i = 1, 2, \dots, t$, and $C_{max}^{(t+1)}(\eta) = tB + t\epsilon + \epsilon$, which implies that the jobs of Customer i ($i = 1, 2, \dots, t$) and the last job of Customer $t + 1$ must be processed immediately after they are released. Then, the first $3t$ jobs of Customer $t + 1$ need to be

processed in the remaining t equal parts such that the last job is not delayed. Thus, the 3-Partition problem has a solution. \square

For any schedule η of SMP3 with customers' completion order $\overset{\circ}{\eta} = (i_1, i_2, \dots, i_k)$, a relax schedule $\overset{\circ}{\eta}_R$ can be constructed as follows. Schedule Customer i_1 's jobs first, and then Customer i_2 's jobs, and so on. The rule is that the processing of every job is as early as possible and do not delay the prepared of previous ones.

In order to develop an algorithm for SMP3, the relationship between SMP3 and the classic scheduling problem $1|r_i|\sum C_i$ will be investigated. Actually, a problem $\bar{P}: 1|\bar{r}_i|\sum \bar{C}_i$ can be constructed from SMP3 through the following steps. For Customer i with job set $\mathcal{J}^{(i)}$, the corresponding job \bar{J}_i with release time $\bar{r}_i = C_{max}(\mathcal{J}^{(i)}) - P^{(i)}$ and processing time $\bar{p}_i = P^{(i)}$ is defined. Furthermore, for any schedule η of SMP3, the completion order $\overset{\circ}{\eta}$ is a schedule for \bar{P} .

To clearly illustrate the relationship among η , $\overset{\circ}{\eta}$ and $\overset{\circ}{\eta}_R$, an example is presented as follows.

An instance of SMP3 with two customers: for Customer 1, there are two jobs $J_1^{(1)} = (0, 1)$ and $J_2^{(1)} = (19, 1)$, while for Customer 2 there are also two jobs $J_1^{(2)} = (0, 19)$ and $J_2^{(2)} = (19, 19)$. Let $\eta = (J_1^{(1)}, J_1^{(2)}, J_2^{(1)}, J_2^{(2)})$, there is $C_{max}^{(1)}(\eta) = 21$, $C_{max}^{(2)}(\eta) = 40$ and $Z(\eta) = 61$ (which is actually optimal). So, as $C_{max}^{(2)}(\eta) > C_{max}^{(1)}(\eta)$, the completion order $\overset{\circ}{\eta}$ is $(1, 2)$. Meanwhile, an corresponding instance of \bar{P} with two jobs can be constructed: $\bar{J}_1 = (18, 2)$ and $\bar{J}_2 = (0, 38)$ ($C_{max}(\mathcal{J}^{(1)}) = 20$, $P^{(1)} = 2$, $C_{max}(\mathcal{J}^{(2)}) = 38$ and $P^{(2)} = 38$). Then, $\overset{\circ}{\eta}$ is an schedule for this instance, which generates the result $\bar{C}_1(\overset{\circ}{\eta}) = 20$ and $\bar{C}_2(\overset{\circ}{\eta}) =$

58. In this situation, the relax schedule $\overset{\circ}{\eta}_R = (J_1^{(1)}, J_2^{(1)}, J_1^{(2)}, J_2^{(2)})$, and $C_{max}^{(1)}(\overset{\circ}{\eta}_R) = 20$, $C_{max}^{(2)}(\overset{\circ}{\eta}_R) = 58$ and $Z(\overset{\circ}{\eta}_R) = 78$.

Based on the relationship constructed above, the following lemma holds.

Lemma 4.6. $C_{max}^{(i)}(\overset{\circ}{\eta}_R) \leq \bar{C}_i(\overset{\circ}{\eta}) \leq 2C_{max}^{(i)}(\eta)$.

Proof of Lemma 4.6: Without loss of generality, assume that the customers' completion order in η is $\overset{\circ}{\eta} = (1, 2, \dots, k)$. The left half of this lemma can be proved by the induction on i . First, the statement of the lemma holds for the case $i = 1$: $C_{max}^{(1)}(\overset{\circ}{\eta}_R) = C_{max}(\mathcal{J}^{(1)}) = \bar{C}_1(\overset{\circ}{\eta})$. Suppose that the lemma is true for $i \leq s$ and consider the case $i = s + 1$.

$$\begin{aligned} C_{max}^{(s+1)}(\overset{\circ}{\eta}_R) &\leq \max \{C_{max}^{(s)}(\overset{\circ}{\eta}_R) + P^{(s+1)}, C_{max}(\mathcal{J}^{(s+1)})\} \\ &\leq \max \{\bar{C}_s(\overset{\circ}{\eta}) + \bar{p}_{s+1}, \bar{r}_{s+1} + \bar{p}_{s+1}\} = \bar{C}_{s+1}(\overset{\circ}{\eta}). \end{aligned} \tag{4.86}$$

For the right half of the lemma,

$$\begin{aligned} \bar{C}_i(\overset{\circ}{\eta}) &= \max_{j \leq i} \left\{ \bar{r}_j + \sum_{h=j}^i \bar{p}_h \right\} \\ &= \max_{j \leq i} \left\{ C_{max}(\mathcal{J}^{(j)}) + \sum_{s=j+1}^i P^{(s)} \right\} \\ &\leq \max_{j \leq i} C_{max}(\mathcal{J}^{(j)}) + \sum_{s=1}^i P^{(s)} \leq 2C_{max}^{(i)}(\eta). \end{aligned} \tag{4.87}$$

□

Note that the construction of the relax schedule only relates to the completion times, so the lemma can also be applied when η is a solution for the preemption case. Next, an algorithm for SMP3 is proposed.

Algorithm SMH3

Solve the preemption case of SMP3 by SMH1 and obtain the optimal solution opt^P .
Construct the corresponding relax schedule opt^P_R .

Theorem 4.5. *A 2-approximate solution for SMP3 can be found by SMH3 in the time $O(nk2^k + n^2)$.*

Proof of Theorem 4.5: As the optimal value of preemption case is the lower bound of that of the non-preemption case,

$$\sum_{i=1}^k C_{max}^{(i)}(opt^P_R) \leq 2 \sum_{i=1}^k C_{max}^{(i)}(opt^P) \leq 2 \sum_{i=1}^k C_{max}^{(i)}(opt). \quad (4.88)$$

From Theorem 4.1, the time of constructing opt^P is $O(nk2^k)$, while the time of constructing the relax schedule is at most $O(n^2)$ as the schedule of every job is only restricted by the previous ones. Then, the total time of the algorithm is $O(nk2^k + n^2)$ which completes the proof. \square

Similarly, this algorithm is not polynomial to k . For large k , SA_SMH1 can be directly applied to solve the preemption case of SMP3 to generate the corresponding simulated annealing algorithm SA_SMH3 for SMP3.

Based on the relationship between SMP3 and \bar{P} , the polynomial approximate algorithm is also developed.

Algorithm PSMH3

Construct the corresponding \bar{P} of SMP3, solve it and obtain the completion order $\hat{\eta}$.
Construct the corresponding relax schedule $\hat{\eta}_R$.

Theorem 4.6. *If $\overset{\circ}{\eta}$ is a ρ -approximate solution for \bar{P} , then $\overset{\circ}{\eta}_R$ is a 2ρ -approximate solution for SMP3.*

Proof of Theorem 4.6: Let opt be the optimal solution for SMP3, and $\overset{\circ}{opt}$ is a feasible solution for \bar{P} .

$$\sum_{i=1}^k C_{max}^{(i)}(\overset{\circ}{\eta}_R) \leq \sum_{i=1}^k C_i(\overset{\circ}{\eta}) \leq \rho \sum_{i=1}^k C_i(\overset{\circ}{opt}) \leq 2\rho \sum_{i=1}^k C_{max}^{(i)}(opt) \quad (4.89)$$

□

The scheme that applying SPRT-rule for the preemption case of \bar{P} can generate a 2-approximate solution for \bar{P} , which means that there is a 4-approximate polynomial algorithm for SMP3.

There is a better result for the special case that $k = 2$.

Algorithm K2SMH3

Construct the relax schedules η_1 and η_2 based on the orders (1, 2) and (2, 1), respectively. Choose the better one from η_1 and η_2 as the final schedule η .

Theorem 4.7. *The approximate ratio of the algorithm K2SMH3 for SMP3 with $k = 2$ is $\frac{4}{3}$.*

Proof of Theorem 4.7: It is obvious that in η_1 , $C_{max}^{(1)}(\eta_1) = C_{max}(\mathcal{J}^{(1)})$. Meanwhile, it can be assumed that $C_{max}^{(2)}(\eta_1) \leq C_{max}(\mathcal{J}^{(1)}) + P^{(2)}$. If not, there is idle time after $C_{max}(\mathcal{J}^{(1)})$ which implies that the jobs processing after $C_{max}(\mathcal{J}^{(1)})$ will not be affected by the jobs processing before. Therefore, the jobs processing after $C_{max}(\mathcal{J}^{(1)})$ is the

same for any schedule and $C_{max}^{(2)}(\eta_1) = C_{max}(\mathcal{J}^{(2)})$. In this case, $Z(\eta_1) \leq C_{max}(\mathcal{J}^{(1)}) + C_{max}(\mathcal{J}^{(2)}) \leq Z(opt)$ which means η_1 is optimal. Similarly, there is $C_{max}^{(2)}(\eta_2) = C_{max}(\mathcal{J}^{(2)})$ and $C_{max}^{(1)}(\eta_2) \leq C_{max}(\mathcal{J}^{(2)}) + P^{(1)}$.

Without loss of generality, assume that $C_{max}^{(1)}(opt) \leq C_{max}^{(2)}(opt)$.

Case 1: $C_{max}(\mathcal{J}^{(2)}) \leq C_{max}^{(1)}(opt)$.

Let $C_{max}^{(2)}(opt) = C_{max}^{(1)}(opt) + \Delta C$, where $\Delta C \geq 0$. Then, before the time $C_{max}^{(1)}(opt)$, the machine at least spend $P^{(2)} - \Delta C$ time for Customer 2's jobs, which means $C_{max}^{(1)}(opt) \geq P^1 + P^2 - \Delta C$. In addition, $P^{(1)} + P^{(2)}$ is a obvious lower bound of $C_{max}^{(2)}(opt)$. Therefore,

$$\begin{aligned}
Z(\eta_1) + Z(\eta_2) &= C_{max}^{(1)}(\eta_1) + C_{max}^{(2)}(\eta_1) + C_{max}^{(1)}(\eta_2) + C_{max}^{(2)}(\eta_2) \\
&\leq 2C_{max}(\mathcal{J}^{(1)}) + 2C_{max}(\mathcal{J}^{(2)}) + P^{(1)} + P^{(2)} \\
&\leq 2C_{max}^{(1)}(opt) + 2C_{max}^{(1)}(opt) + P^{(1)} + P^{(2)} \\
&= 2C_{max}^{(1)}(opt) + 2C_{max}^{(2)}(opt) + P^{(1)} + P^{(2)} - 2\Delta C
\end{aligned} \tag{4.90}$$

$$\begin{aligned}
\frac{Z(\eta_1) + Z(\eta_2)}{Z(opt)} &\leq 2 + \frac{P^{(1)} + P^{(2)} - 2\Delta C}{C_{max}^{(1)}(opt) + C_{max}^{(2)}(opt)} \\
&\leq 2 + \frac{P^{(1)} + P^{(2)} - 2\Delta C}{2P^{(1)} + 2P^{(2)} - \Delta C} \leq 2 + \frac{1}{2}
\end{aligned} \tag{4.91}$$

$$Z(\eta) = \min \{Z(\eta_1), Z(\eta_2)\} \leq \frac{5}{4}Z(opt) \tag{4.92}$$

Case 2: $C_{max}(\mathcal{J}^{(2)}) > C_{max}^{(1)}(opt)$.

Assume that $C_{max}^{(1)}(opt) = C_{max}(\mathcal{J}^{(1)}) + \Delta C_1$, $C_{max}(\mathcal{J}^{(2)}) = C_{max}^{(1)}(opt) + \Delta C_2$, and $C_{max}^{(2)}(opt) = C_{max}(\mathcal{J}^{(2)}) + \Delta C_3$, where ΔC_1 , ΔC_2 , and ΔC_3 are all nonnegative. Then, before the time $C_{max}^{(1)}(opt)$, the machine spends at least $P^{(2)} - \Delta C_2 - \Delta C_3$ time for Customer

2's jobs, which means $C_{max}^{(1)}(opt) \geq P^1 + P^2 - \Delta C_2 - \Delta C_3$.

$$\begin{aligned}
Z(\eta_1) &\leq 2C_{max}(\mathcal{J}^{(1)}) + P^{(2)} \\
&= C_{max}^{(1)}(opt) - \Delta C_1 + C_{max}^{(2)}(opt) - \Delta C_1 - \Delta C_2 - \Delta C_3 + P^{(2)} \quad (4.93) \\
&= Z(opt) + P^{(2)} - 2\Delta C_1 - \Delta C_2 - \Delta C_3
\end{aligned}$$

$$\begin{aligned}
Z(\eta_2) &\leq 2C_{max}(\mathcal{J}^{(2)}) + P^{(1)} \\
&= C_{max}^{(1)}(opt) + \Delta C_1 + C_{max}^{(2)}(opt) - \Delta C_3 + P^{(1)} \quad (4.94) \\
&= Z(opt) + P^{(1)} + \Delta C_1 - \Delta C_3
\end{aligned}$$

$$Z(opt) \geq 2P^{(1)} + 2P^{(2)} - \Delta C_2 - \Delta C_3 \quad (4.95)$$

$$\begin{aligned}
\frac{Z(\eta)}{Z(opt)} &= \frac{\min \{Z(\eta_1), Z(\eta_2)\}}{Z(opt)} \\
&\leq 1 + \min \left\{ \frac{P^{(2)} - 2\Delta C_1 - \Delta C_2 - \Delta C_3}{2P^{(1)} + 2P^{(2)} - \Delta C_2 - \Delta C_3}, \frac{P^{(1)} + \Delta C_1 - \Delta C_3}{2P^{(1)} + 2P^{(2)} - \Delta C_2 - \Delta C_3} \right\} \quad (4.96)
\end{aligned}$$

When $\Delta C_2 = \frac{1}{2}(P^{(2)} - P^{(1)}) - \Delta C_1$, the term of the right hand achieves the maximum value.

$$\begin{aligned}
\frac{Z(\eta)}{Z(opt)} &\leq 1 + \frac{\frac{1}{2}(P^{(1)} + P^{(2)}) - \Delta C_1 - \Delta C_3}{\frac{5}{2}P^{(1)} + \frac{3}{2}P^{(2)} + \Delta C_1 - \Delta C_3} \\
&\leq 1 + \frac{\frac{1}{2}(P^{(1)} + P^{(2)})}{\frac{5}{2}P^{(1)} + \frac{3}{2}P^{(2)}} \leq 1 + \frac{1}{3} = \frac{4}{3} \quad (4.97)
\end{aligned}$$

Next, it needs to show that there is an instance such that the ratio can be achieved. For Customer 1, there are two jobs $J_1^{(1)} = (0, \epsilon)$ and $J_2^{(1)} = (20 - \epsilon, \epsilon)$, while for Customer 2 there are also two jobs $J_1^{(2)} = (0, 20 - \epsilon)$ and $J_2^{(2)} = (20 - \epsilon, 20 - \epsilon)$, where ϵ is a very small positive number. η_1 will process these jobs in the order $(J_1^{(1)}, J_2^{(1)}, J_1^{(2)}, J_2^{(2)})$ and $Z(\eta_1) = 80 - 2\epsilon$. Similarly, η_2 will process these jobs in the order $(J_1^{(2)}, J_2^{(2)}, J_1^{(1)}, J_2^{(1)})$ and $Z(\eta_2) = 80 - 2\epsilon$. However, the optimal solution should process the jobs in the order

$(J_1^{(1)}, J_1^{(2)}, J_2^{(1)}, J_2^{(2)})$ and $Z(opt) = 60 + \epsilon$. Therefore, $\frac{Z(\eta)}{Z(opt)} = \frac{\min \{Z(\eta_1), Z(\eta_2)\}}{Z(opt)} = \frac{80-2\epsilon}{60+\epsilon}$, which will tend to $\frac{4}{3}$ as ϵ tends to 0. This completes the proof. \square

4.4 Algorithm for Problem SMP4

SMP4 has the following features: Jobs are released on-line and delivered in "direct" pattern. The capacity of vehicles and the number of vehicles are both enough. Actually, SMP4 is the on-line version of SMP3. The same lower bound can be applied to this on-line problem (see Appendix).

Corollary 4.2. *No on-line algorithm for SMP4 can have competitive ratio less than 2, even all preparation times being 0.*

As there is more than one customer and the preemption of jobs processing is not allowed, processing every job may delay other customers' completion time. Therefore, there should be a period waiting time for long jobs. The ready job for single machine case is defined as follows.

Definition 4.2. *A job $J_j^{(i)}$ is called ready at time t if it has arrives ($r_j^{(i)} \leq t$), not completed ($C_j^{(i)} > t$) and $\frac{1}{2}p_j^{(i)} \leq t$.*

By combining the concept of ready job with algorithm SMH2, an on-line algorithm for SMP4 is proposed.

Algorithm SMH4

At the time t that a new job arrives, the customers are re-indexed in an increasing order of $C_{max}(\mathcal{J}_{\leq t}^{(i)})$ (If there is more than one customer with the same $C_{max}(\mathcal{J}_{\leq t}^{(i)})$, their

order is the original index order). When the machine is free, process ready jobs of the customer with the highest on-line priority.

At the time of lD where $l \geq 1$ and l is integer, if there is no uncompleted jobs for Customer i , then there must be a batch to deliver all the completed job of Customer i ; otherwise, there is no operation for these jobs.

From the algorithm, the job $J_j^{(i)}$ can only be processed after the time $\max\{r_j^{(i)}, \frac{1}{2}p_j^{(i)}\}$.

Theorem 4.8. *The on-line algorithm SMH4 for SMP4 with $k = 2$ is $2 + \frac{1}{2}$ -competitive.*

Proof of Theorem 4.8: Without loss of generality, suppose that $C_{max}(\mathcal{J}^{(1)}) \leq C_{max}(\mathcal{J}^{(2)})$.

Let η be the algorithm solution, and l_1D be the last idle delivery point before $C_{max}(\mathcal{J}^{(1)})$.

Let (i_1, i_2) be the order of the customers' completion times in the optimal solutions,

and the optimal result is $Z(opt) = C_{max}^{(i_1)}(opt) + C_{max}^{(i_2)}(opt) + 2D$. It is obvious that

$C_{max}(\mathcal{J}^{(1)}) \leq C_{max}^{(i_1)}(opt)$ and $C_{max}(\mathcal{J}^{(1)} \cup \mathcal{J}^{(2)}) \leq C_{max}^{(i_2)}(opt)$. In addition, let $\mathcal{J}_b^{(2)}$ be

the set of Customer 2's jobs which are completed before $C_{max}^{(1)}(\eta)$ in η , and $T_d^{(i)}$ be the

end point of the last period of waiting time before $C_{max}^{(i)}(\eta)$ ($i = 1, 2$). At the time of

$C_{max}(\mathcal{J}^{(1)})$, all the jobs of Customer 1 are released and also satisfy the processing condi-

tion, so $T_d^{(1)} \leq C_{max}(\mathcal{J}^{(1)})$. Meanwhile, after $C_{max}(\mathcal{J}^{(1)})$, the processing of Customer 1's

jobs would be continuous until all are completed. Therefore, the jobs processing in the in-

terval $(C_{max}(\mathcal{J}^{(1)}), C_{max}^{(1)}(\eta)]$ should be a block of Customer 2's jobs followed by a block of

Customer 1's jobs (see Figure 4.8). Simply use P_1 and P_2 to represent $P_{(C_{max}(\mathcal{J}^{(1)}), C_{max}^{(1)}(\eta))}^{(1)}(\eta)$

and $P_{(C_{max}(\mathcal{J}^{(1)}), C_{max}^{(1)}(\eta))}^{(2)}(\eta)$, respectively.

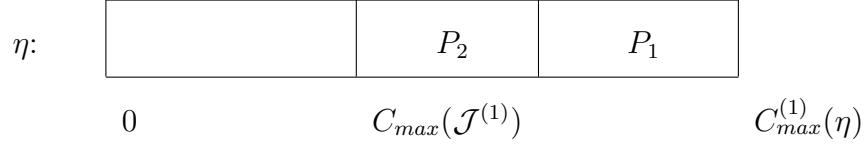


Figure 4.8. The Jobs Processing in The Interval $(C_{max}(\mathcal{J}^{(1)}), C_{max}^{(1)}(\eta)]$ (2)

Case 1: $C_{max}(\mathcal{J}_b^{(2)}) < C_{max}(\mathcal{J}^{(1)})$. As $C_{max}(\mathcal{J}^{(1)}) \leq C_{max}(\mathcal{J}^{(2)})$, there must be $\mathcal{J}^{(2)} \setminus \mathcal{J}_b^{(2)} \neq \emptyset$, which implies that there are Customer 2's jobs completed later than $C_{max}^{(1)}(\eta)$ and thus $C_{max}^{(2)}(\eta) > C_{max}^{(1)}(\eta)$. Therefore, there is $T_d^{(1)} \leq T_d^{(2)}$.

Case 1.1: $T_d^{(1)} = T_d^{(2)}$. The total waiting time in η will not be more than $T_d^{(1)}$, which implies that $C_{max}^{(2)}(\eta) \leq T_d^{(1)} + C_{max}(\mathcal{J}^{(1)} \cup \mathcal{J}^{(2)}) - l_1 D \leq T_d^{(1)} + C_{max}^{(i_2)}(opt) - l_1 D$ (Here assume $T_d^{(1)} > l_1 D$; otherwise the waiting will not affect the completion times of two customers which will lead to a simpler case).

In the interval $(0, l_1 D]$, there are at most two batches for every delivery point, so the delivery cost will not be more than $2l_1 D$.

In the interval $(l_1 D, C_{max}(\mathcal{J}^{(1)}) - \min\{P_1, P_2\}]$, there is at most one batch for every delivery point, so the delivery cost will not be more than $\lfloor \frac{C_{max}(\mathcal{J}^{(1)}) - \min\{P_1, P_2\}}{D} \rfloor D - l_1 D$.

In the interval $(C_{max}(\mathcal{J}^{(1)}) - \min\{P_1, P_2\}, C_{max}(\mathcal{J}^{(1)}])$, there are uncompleted jobs for both customers, so there is no delivery.

In the interval $(C_{max}(\mathcal{J}^{(1)}), C_{max}^{(1)}(\eta)]$, there is at most one batch for Customer 2, so the delivery cost will not be more than D .

In the interval $(C_{max}^{(1)}(\eta), \rho_{max}^{(2)}(\eta)]$, there is at most one batch for Customer 2 for every delivery point, and there is one batch in total for Customer 1, so the delivery cost will not

be more than $D + \lceil \frac{C_{max}^{(2)}(\eta)}{D} \rceil D - \lfloor \frac{C_{max}^{(1)}(\eta)}{D} \rfloor D$.

$$TC(\eta) \leq l_1 D + C_{max}(\mathcal{J}^{(1)}) - \min\{P_1, P_2\} + 2D + \lceil \frac{C_{max}^{(2)}(\eta)}{D} \rceil D - \lfloor \frac{C_{max}^{(1)}(\eta)}{D} \rfloor D \quad (4.98)$$

Therefore,

$$\begin{aligned} Z(\eta) &= \lceil \frac{C_{max}^{(1)}(\eta)}{D} \rceil D + \lceil \frac{C_{max}^{(2)}(\eta)}{D} \rceil D + TC(\eta) + 2T_{01} + 2T_{02} \\ &\leq l_1 D + C_{max}(\mathcal{J}^{(1)}) - \min\{P_1, P_2\} + 2C_{max}^{(2)}(\eta) + 5D + 2T_{01} + 2T_{02} \\ &\leq l_1 D + C_{max}(\mathcal{J}^{(1)}) - \min\{P_1, P_2\} + 2T_d^{(1)} + \\ &\quad 2C_{max}^{(i_2)}(opt) - 2l_1 D + 2T_{01} + 2T_{02} \tag{4.99} \\ &\leq 3C_{max}(\mathcal{J}^{(1)}) + 2C_{max}^{(i_2)}(opt) + 5D + 2T_{01} + 2T_{02} \\ &\leq 2(C_{max}^{(i_1)}(opt) + C_{max}^{(i_2)}(opt) + 2D + 2T_{01} + 2T_{02}) + C_{max}(\mathcal{J}^{(1)}) + D \\ &\leq (2 + \frac{1}{2})Z(opt). \end{aligned}$$

Case 1.2: $T_d^{(1)} < T_d^{(2)}$. If there is no idle time after $T_d^{(2)}$, $C_{max}^{(2)}(\eta) \leq T_d^{(2)} + P(\mathcal{J}^{(2)} \setminus \mathcal{J}_b^{(2)}) \leq T_d^{(2)} + C_{max}(\mathcal{J}^{(2)}) - l_1 D$; otherwise, $C_{max}^{(2)}(\eta) = C_{max}(\mathcal{J}^{(2)}) \leq T_d^{(2)} + C_{max}(\mathcal{J}^{(2)}) - l_1 D$. In particular, there is at least one job of Customer 2 with processing time no less than $2T_d^{(2)}$ which has been released before $T_d^{(2)}$ but processed after $T_d^{(2)}$. Meanwhile, $C_{max}^{(2)}(\eta) \leq T_d^{(1)} + C_{max}(\mathcal{J}_b^{(2)}) + C_{max}(\mathcal{J}^{(1)}) \leq 3C_{max}(\mathcal{J}^{(1)})$.

In the interval $(0, l_1 D]$, there are at most two batches for every delivery point, so the delivery cost will not be more than $2l_1 D$.

In the interval $(l_1 D, T_d^{(2)}]$, there is at most one batch for every delivery point, so the delivery cost will not be more than $\lfloor \frac{T_d^{(2)}}{D} \rfloor D - l_1 D$.

In the interval $(T_d^{(2)}, \rho_{max}^{(2)}(\eta)]$, there is at most one batch for Customer 1 (only when $\lfloor \frac{T_d^{(2)}}{D} \rfloor D < C_{max}^{(1)}(\eta)$), and there is at most one batch for every delivery point after $3T_d^{(2)}$

for Customer 2, so the delivery cost will not be more than $D + \lceil \frac{C_{max}^{(2)}(\eta)}{D} \rceil D - \lfloor \frac{3T_d^{(2)}}{D} \rfloor D$.

$$\begin{aligned} TC(\eta) &\leq l_1 D + T_d^{(2)} + D + \lceil \frac{C_{max}^{(2)}(\eta)}{D} \rceil D - 3T_d^{(2)} + D \\ &\leq l_1 D + C_{max}^{(2)}(\eta) - 2T_d^{(2)} + 3D. \end{aligned} \quad (4.100)$$

Therefore,

$$\begin{aligned} Z(\eta) &= \lceil \frac{C_{max}^{(1)}(\eta)}{D} \rceil D + \lceil \frac{C_{max}^{(2)}(\eta)}{D} \rceil D + TC(\eta) + 2T_{01} + 2T_{02} \\ &\leq C_{max}^{(1)}(\eta) + 2C_{max}^{(2)}(\eta) + l_1 D - 2T_d^{(2)} + 5D + 2T_{01} + 2T_{02} \\ &\leq 3C_{max}(\mathcal{J}^{(1)}) + 2T_d^{(2)} + 2C_{max}(\mathcal{J}^{(2)}) - 2l_1 D + l_1 D - \\ &\quad 2T_d^{(2)} + 5D + 2T_{01} + 2T_{02} \\ &\leq 3C_{max}(\mathcal{J}^{(1)}) + 2C_{max}^{(i_2)}(opt) + 5D + 2T_{01} + 2T_{02} \\ &\leq (2 + \frac{1}{2})Z(opt). \end{aligned} \quad (4.101)$$

Case 2: $C_{max}(\mathcal{J}_b^{(2)}) \geq C_{max}(\mathcal{J}^{(1)})$. Let $J_h^{(2)}$ be the last Customer 2's job which is completed before $C_{max}^{(1)}(\eta)$, and $s_h^{(2)}$ be its start time. For Customer 1's jobs completed after $C_{max}(\mathcal{J}^{(1)})$, the shortest release time will not be more than the time $C_{max}(\mathcal{J}^{(1)}) - P_1$ and the longest processing time will not be more than P_1 , so at least one of them can be processed at the time $\max\{C_{max}(\mathcal{J}^{(1)}) - P_1, \frac{1}{2}P_1\}$. As $J_h^{(2)}$ cannot have a higher priority than these jobs, it must be processed when none of them is ready, $s_h^{(2)} \leq \max\{C_{max}(\mathcal{J}^{(1)}) - P_1, \frac{1}{2}P_1\}$, which also implies that $p_h^{(2)} \leq 2\max\{C_{max}(\mathcal{J}^{(1)}) - P_1, \frac{1}{2}P_1\}$. Furthermore, the jobs pprocessing after $s_h^{(2)}$ is continuous, so $T_d^{(1)} \leq s_h^{(2)}$.

Case 2.1: $C_{max}^{(2)}(\eta) > C_{max}^{(1)}(\eta)$. In this case, there should be $T_d^{(1)} \leq T_d^{(2)}$.

Case 2.1.1: $T_d^{(1)} = T_d^{(2)}$. Similar to the case 1.1, $C_{max}^{(2)}(\eta) \leq T_d^{(1)} + C_{max}^{(i_2)}(opt) - l_1 D$.

In the interval $(0, l_1 D]$, there are at most two batches for every delivery point, so the delivery cost will not be more than $2l_1 D$.

In the interval $(l_1D, C_{max}(\mathcal{J}^{(1)})]$, there is at most one batch for every delivery point before $s_h^{(2)}$, and one possible batch for Customer 1 at the delivery point next to $s_h^{(2)}$, so the delivery cost will not be more than $\lfloor \frac{s_h^{(2)}}{D} \rfloor D - l_1D + D$ if $\lceil \frac{s_h^{(2)}}{D} \rceil D < C_{max}(\mathcal{J}^{(1)})$, or $\lfloor \frac{s_h^{(2)}}{D} \rfloor D - l_1D$ otherwise. Both cases will lead to the delivery cost bounded by $C_{max}(\mathcal{J}^{(1)}) - l_1D$.

In the interval $(C_{max}(\mathcal{J}^{(1)}), C_{max}^{(1)}(\eta)]$, there is at most one batch for Customer 2, so the delivery cost will not be more than D .

In the interval $(C_{max}^{(1)}(\eta), \rho_{max}^{(2)}(\eta)]$, there is at most one batch for Customer 2 for every delivery point, and there is one batch in total for Customer 1, so the delivery cost will not be more than $D + \lceil \frac{C_{max}^{(2)}(\eta)}{D} \rceil D - \lfloor \frac{C_{max}^{(1)}(\eta)}{D} \rfloor D$.

$$TC(\eta) \leq l_1D + C_{max}(\mathcal{J}^{(1)}) + 2D + \lceil \frac{C_{max}^{(2)}(\eta)}{D} \rceil D - \lfloor \frac{C_{max}^{(1)}(\eta)}{D} \rfloor D \quad (4.102)$$

$$\begin{aligned} Z(\eta) &= \lceil \frac{C_{max}^{(1)}(\eta)}{D} \rceil D + \lceil \frac{C_{max}^{(2)}(\eta)}{D} \rceil D + TC(\eta) + 2T_{01} + 2T_{02} \\ &\leq C_{max}(\mathcal{J}^{(1)}) + 2C_{max}^{(2)}(\eta) + 5D + l_1D + 2T_{01} + 2T_{02} \\ &\leq C_{max}(\mathcal{J}^{(1)}) + 2T_d^{(1)} + 2C_{max}^{(i_2)}(opt) + 5D - l_1D + 2T_{01} + 2T_{02} \\ &\leq C_{max}(\mathcal{J}^{(1)}) + 2s_h^{(2)} + 2C_{max}^{(i_2)}(opt) + 5D - l_1D + 2T_{01} + 2T_{02} \\ &\leq 3C_{max}(\mathcal{J}^{(1)}) + 2C_{max}^{(i_2)}(opt) + 5D + 2T_{01} + 2T_{02} \\ &\leq (2 + \frac{1}{2})Z(opt). \end{aligned} \quad (4.103)$$

Case 2.1.2: $T_d^{(1)} < T_d^{(2)}$. Similar to case 1.2, $TC(\eta) \leq l_1D + C_{max}^{(2)}(\eta) - 2T_d^{(2)} + 3D$ and $C_{max}^{(2)}(\eta) \leq T_d^{(2)} + C_{max}(\mathcal{J}^{(2)}) - l_1D$. The completion time of Customer 1 will satisfy that $C_{max}^{(1)}(\eta) = s_h^{(2)} + p_h^{(2)} + P_1 \leq \max\{3C_{max}(\mathcal{J}^{(1)}) - 2P_1, \frac{5}{2}P_1\} \leq 3C_{max}(\mathcal{J}^{(1)})$.

$$\begin{aligned}
Z(\eta) &= \lceil \frac{C_{max}^{(1)}(\eta)}{D} \rceil D + \lceil \frac{C_{max}^{(2)}(\eta)}{D} \rceil D + TC(\eta) + 2T_{01} + 2T_{02} \\
&\leq C_{max}^{(1)}(\eta) + 2C_{max}^{(2)}(\eta) + l_1 D - 2T_d^{(2)} + 5D + 2T_{01} + 2T_{02} \\
&\leq 3C_{max}(\mathcal{J}^{(1)}) + 2C_{max}^{(i_2)}(opt) + 5D + 2T_{01} + 2T_{02} \\
&\leq (2 + \frac{1}{2})Z(opt).
\end{aligned} \tag{4.104}$$

Case 2.2: $C_{max}^{(2)}(\eta) < C_{max}^{(1)}(\eta)$. In this case, there is $\mathcal{J}_b^{(2)} = \mathcal{J}^{(2)}$, $C_{max}^{(2)}(\eta) = s_h^{(2)} + p_h^{(2)}$, and $C_{max}^{(1)}(\eta) = s_h^{(2)} + p_h^{(2)} + P_1$. As $J_h^{(2)}$ is released after $l_1 D$, the preparation time should satisfy $p_h^{(2)} \leq C_{max}(\mathcal{J}^{(2)}) - l_1 D$.

In the interval $(0, l_1 D]$, there are at most two batches for every delivery point, so the delivery cost will not be more than $2l_1 D$.

In the interval $(l_1 D, C_{max}(\mathcal{J}^{(1)})]$, there is at most one batch for every delivery point before $s_h^{(2)}$, and one possible batch for Customer 1 at the delivery point next to $s_h^{(2)}$, so the delivery cost will not be more than $\lfloor \frac{s_h^{(2)}}{D} \rfloor D - l_1 D + D$ if $\lceil \frac{s_h^{(2)}}{D} \rceil D < C_{max}(\mathcal{J}^{(1)})$, or $\lfloor \frac{s_h^{(2)}}{D} \rfloor D - l_1 D$ otherwise. Both cases will lead to the delivery cost bounded by $C_{max}(\mathcal{J}^{(1)}) - l_1 D$.

In the interval $(C_{max}(\mathcal{J}^{(1)}), \rho_{max}^{(1)}(\eta)]$, there are in total two batches for two customers, so the delivery cost will be $2D$, which means $TC(\eta) \leq l_1 D + C_{max}(\mathcal{J}^{(1)}) + 2D$.

$$\begin{aligned}
Z(\eta) &= \lceil \frac{C_{max}^{(1)}(\eta)}{D} \rceil D + \lceil \frac{C_{max}^{(2)}(\eta)}{D} \rceil D + TC(\eta) + 2T_{01} + 2T_{02} \\
&\leq s_h^{(2)} + p_h^{(2)} + s_h^{(2)} + p_h^{(2)} + P_1 + l_1 D + C_{max}(\mathcal{J}^{(1)}) + 4D + 2T_{01} + 2T_{02} \\
&\leq 2s_h^{(2)} + P_1 + C_{max}(\mathcal{J}^{(1)}) + 2C_{max}(\mathcal{J}^{(2)}) + 5D - l_1 D + 2T_{01} + 2T_{02} \\
&\leq \max\{2C_{max}(\mathcal{J}^{(1)}) - P_1, 2P_1\} + C_{max}(\mathcal{J}^{(1)}) + 2C_{max}(\mathcal{J}^{(2)}) + \tag{4.105} \\
&\quad 5D + 2T_{01} + 2T_{02} \\
&\leq 3C_{max}(\mathcal{J}^{(1)}) + 2C_{max}^{(i_2)}(opt) + 5D + 2T_{01} + 2T_{02} \\
&\leq (2 + \frac{1}{2})Z(opt).
\end{aligned}$$

□

For the case that k is greater than 2, the performance of the algorithm will be shown in a simulation to be presented later.

4.5 Algorithm for SMP Problems with Capacity Limited Vehicles

In this section, the above four SMP problems with capacity limited vehicles are considered.

For the case that the capacity of vehicles is constrained by a constant C , the four specific problems can be represented as follows.

$$\text{CSMP1: } 1|r_j, pmtn|V(\infty, C), direct|k| \sum D_{max}^{(i)} + TC$$

$$\text{CSMP2: } 1|r_j, pmtn, on - line|V(\infty, C), direct|k| \sum D_{max}^{(i)} + TC$$

$$\text{CSMP3: } 1|r_j|V(\infty, C), direct|k| \sum D_{max}^{(i)} + TC$$

$$\text{CSMP4: } 1|r_j, \text{ on-line}|V(\infty, C), \text{ direct}|k|\sum D_{max}^{(i)} + TC$$

For the off-line problems CSMP1 and CSMP3, all the jobs of the same customer should be delivered in $\lceil \frac{n_i}{C} \rceil$ batches when they are all completed. Therefore, there are $\sum_{i=1}^k \lceil \frac{n_i}{C} \rceil$ batches in the optimal schedule and $D_{max}^{(i)}(opt) = \rho_{max}^{(i)}(opt) + 2T_{0i} = C_{max}^{(i)}(opt) + 2T_{0i}$ for $i = 1, 2, \dots, k$, which implies that $Z(opt) = \sum_{i=1}^k D_{max}^{(i)}(opt) + TC(opt) = \sum_{i=1}^k C_{max}^{(i)}(opt) + 2\sum_{i=1}^k T_{0i} + \sum_{i=1}^k \lceil \frac{n_i}{C} \rceil D$. Then, it is equivalent with SMP1 and SMP3.

As the on-line problem for single-customer has a lower bound of $\max\{\frac{1+\sqrt{5}}{2}, 2 - \frac{1}{C}\}$ [Han, 2012], the lower bounds of the problems CSMP2 and CSMP4 are at least $\max\{\frac{1+\sqrt{5}}{2}, 2 - \frac{1}{C}\}$.

Corollary 4.3. *No on-line algorithm for CSMP2 and CSMP4 can have competitive ratio less than $\max\{1 + \theta, 2 - \frac{1}{C}\}$, even all processing times being 0.*

For the on-line problems CSMP2 and CSMP4, on-line algorithms modified from SMH2 and SMH4 are proposed as follows.

Algorithm CSMH2

At the time t that a new job arrives, the customers are re-indexed in an increasing order of $C_{max}(\mathcal{J}_{\leq t}^{(i)})$ (If there is more than one customer with the same $C_{max}(\mathcal{J}_{\leq t}^{(i)})$, their order is the original index order). When a new job arrives or the machine is free, process available jobs of the customer with the highest on-line priority.

At the time of lD where $l \geq 1$ and l is integer, if there is no uncompleted job for Customer i , then there are batches to deliver all the completed jobs for Customer i ;

otherwise there are full batches to deliver as many completed jobs as possible.

Algorithm CSMH4

At the time t that a new job arrives, the customers are re-indexed in an increasing order of $C_{max}(\mathcal{J}_{\leq t}^{(i)})$ (If there is more than one customer with the same $C_{max}(\mathcal{J}_{\leq t}^{(i)})$, their order is the original index order). When the machine is free, process ready jobs of the customer with the highest on-line priority.

At the time of lD where $l \geq 1$ and l is integer, if there is no uncompleted job for Customer i , then there are batches to deliver all the completed jobs for Customer i ; otherwise there are full batches to deliver as many completed jobs as possible.

In the following, the competitive analysis of on-line algorithms CSMH2 and CSMH4 are presented. Let η be the schedule of the algorithm CSMH2 (CSMH4) and opt be the optimal schedule. Note that the processing part of CSMH2 (CSMH4) is the same as SMH2 (SMH4), but SMH2 (SMH4) can deliver all the jobs of the same customer in one batch when there is no uncompleted job. Let η^∞ be the schedule of the algorithm SMH2 (SMH4) for and opt^∞ be the corresponding optimal schedule.

Lemma 4.7. $\frac{Z(\eta)}{Z(opt)} \leq \frac{Z(\eta^\infty)}{Z(opt^\infty)}$

Proof of Lemma 4.7: Let a_i and b_i be the number of unfull and full batches for Customer i in η , respectively. From CSMH2 (CSMH4) and SMH2 (SMH4), at every delivery point, if there is an unfull batch for Customer i in η , then there should be a batch for Customer i in η^∞ , which implies that $a_i D \leq TC^{(i)}(\eta^\infty)$. Meanwhile, there is at least one batch for Customer i in η^∞ , $TC^{(i)}(\eta^\infty) \geq D$. For some i , if $a_i = 0$, then $b_i = \lceil \frac{n_i}{C} \rceil = \frac{n_i}{C}$, which

implies that $TC^{(i)}(\eta) = b_i D = \lceil \frac{n_i}{C} \rceil D \leq TC^{(i)}(\eta^\infty) + (\lceil \frac{n_i}{C} \rceil - 1)D$; otherwise, $b_i \leq \lceil \frac{n_i}{C} \rceil - 1$, and $TC^{(i)}(\eta) = (a_i + b_i)D \leq TC^{(i)}(\eta^\infty) + (\lceil \frac{n_i}{C} \rceil - 1)D$. In addition, the same processing part and the same delivery points indicates that $D_{max}^{(i)}(\eta) = D_{max}^{(i)}(\eta^\infty)$.

$$\begin{aligned}
\frac{Z(\eta)}{Z(opt)} &= \frac{\sum_{i=1}^k (D_{max}^{(i)}(\eta) + 2T_{0i} + TC^{(i)}(\eta))}{\sum_{i=1}^k (D_{max}^{(i)}(opt) + 2T_{0i} + TC^{(i)}(opt))} \\
&\leq \frac{\sum_{i=1}^k (D_{max}^{(i)}(\eta) + 2T_{0i} + TC^{(i)}(\eta^\infty) + (\lceil \frac{n_i}{C} \rceil - 1)D)}{\sum_{i=1}^k (D_{max}^{(i)}(opt) + 2T_{0i} + \lceil \frac{n_i}{C} \rceil D)} \\
&= \frac{\sum_{i=1}^k (D_{max}^{(i)}(\eta^\infty) + 2T_{0i} + TC^{(i)}(\eta^\infty) + (\lceil \frac{n_i}{C} \rceil - 1)D)}{\sum_{i=1}^k (D_{max}^{(i)}(opt^\infty) + 2T_{0i} + D + (\lceil \frac{n_i}{C} \rceil - 1)D)} \\
&= \frac{Z(\eta^\infty) + \sum_{i=1}^k ((\lceil \frac{n_i}{C} \rceil - 1)D)}{Z(opt^\infty) + \sum_{i=1}^k ((\lceil \frac{n_i}{C} \rceil - 1)D)} \\
&\leq \frac{Z(\eta^\infty)}{Z(opt^\infty)}.
\end{aligned} \tag{4.106}$$

□

From the above lemma, the competitive analysis of CSMH2 and CSMH4 is presented.

Corollary 4.4. *The competitive ratio of on-line algorithm CSMH2 for CSMP2 with $k = 2$ is 2.*

Corollary 4.5. *The competitive ratio of on-line algorithm CSMH2 for CSMP2 with $k = 3$ is $2 + \frac{2}{27}$.*

Corollary 4.6. *The on-line algorithm CSMH4 for the problem CSMP4 with $k = 2$ is $2 + \frac{1}{2}$ -competitive*

4.6 Simulated Experiment for SMH without Routing

In this section, a simulated experiment or simulation for short is conducted to demonstrate the run-time performance and the ratio (approximate ratio and competitive ratio)

performance of the above algorithms (SMH1-SMH4, SA_SMH1, SA_SMH3, CSMH2 and CSMH4) in normal scenarios and illustrate how the algorithms can be used in practice. An instance was defined by prescribing a set of the foregoing parameters ($n_i, p_j^{(i)}$ and $r_j^{(i)}$ for $j = 1, 2, \dots, n_i, C$ and D). In order to make the discussion applicable to much more general situations, the parameters for the instance were generated by a random engine (uniform distribution and poisson distribution), and such a treatment is also found in the work [Qi, 2005; Shirvani and Shadrokh, 2013]. The instances were generated by these randomly generated parameters. The algorithm was implemented in the Matlab environment. The parameters are thus determined based on the following assumptions (similar assumptions were applied to other simulation experiments in this thesis):

- (1) The release of jobs for Customer i follows the poisson distribution with the parameter λ_i , i.e., the number of jobs released at some time r : $n_i(r) \sim P(\lambda_i)$ and the next release time is $r + r'$, where $r' \sim U(0, \lambda_i)$, λ_i is two times of the mean value of the release intervals for Customer i , and $\lambda_i \sim U(0, \Lambda_i)$ ($i = 1, 2, \dots, k$).
- (2) The job processing time for Customer i follows the uniform distribution in the interval $[0, b_i]$, i.e., $p_j^{(i)} \sim U(0, b_i)$ for $j = 1, 2, \dots, n_i$, where b_i is two times of the mean value of the processing time for Customer i and $b_i \sim U(0, B_i)$ ($i = 1, 2, \dots, k$).
- (3) The number of jobs for Customer i follows the uniform distribution in the set $\{1, 2, \dots, N_i\}$, i.e., $Pr\{n_i = h\} = \frac{1}{N_i}$ for $h = 1, 2, \dots, N_i$ where N_i is two times of the mean value of the number of jobs for Customer i ($i = 1, 2, \dots, k$).
- (4) The delivery cost D is a constant.

(5) The number of customers is of four cases: $k = 2$, $k = 3$, $k = 8$, $k = 10$, and $k = 20$.

By choosing different values for Λ_i , B_i , and N_i , instances are generated and scheduling is then executed. In all cases, 100 instances are generated (the choice of 100 was mainly due to the overhead running time; with 100 instances, the running hours were about 30 hours). Table 4.1 shows the result for the case $k = 2$. Each row in the table is the average of the results of the 100 instances. The columns in the table are (1) the ratio of the algorithm value to the benchmark value, (2) the run-time in seconds, respectively. Notice that the result of SMH1 is applied as the benchmark for the case $k = 2$. As SMH1 solves the problem SMP1 which is a lower bound for the problem SMP3, the ratio of the results of SMH3 and SMH4 to the optimal result (the result of SMH1) should perform better. Meanwhile, the run-time of SMH3 is the sum of the run-time of SMH4 and the time of constructing relax schedules.

From Table 4.1, the ratio of SA_SMH1 is 1, which reflects the optimality of this simulated annealing algorithm. For the off-line algorithm SMH3, the ratio is very close to 1, which means that this algorithm can construct a great solution for problem SMP3 if the optimal solution of SMP1 is known. For the on-line algorithms SMH2 and SMH4 the ratio never exceeds 2, which displays the robustness of the two algorithms, and is also consistent with the results of the Theorem 4.2 and the Theorem 4.8. Actually, for most cases, the ratios of the two algorithms are not greater than 1.6, which shows the excellent performance on the normal instances. The run-time for SA_SMH1 never exceeds 0.015 seconds and the run-time of the analytical algorithms (SMH1, SMH2, SMH3, and SMH4) is much shorter, so the efficiency of the algorithms is very high when $k = 2$.

Table 4.1. Results of Algorithms SMH1-SMH4 with $k = 2$

SMH1		SA_SMH1		SMH2	
ratio	time	ratio	time	ratio	time
1	0.00021	1	0.013	1.486	0.0014
1	0.00018	1	0.012	1.103	0.0013
1	0.00018	1	0.013	1.968	0.0013
1	0.00019	1	0.013	1.418	0.0015
1	0.00017	1	0.0075	1.577	0.00047
1	0.00018	1	0.016	1.075	0.0016
SMH3				SMH4	
ratio	time		ratio	time	
1.017	0.00021+0.00066		1.454	0.0021	
1.0162	0.00018+0.00059		1.103	0.0019	
1.0002	0.00018+0.00067		1.968	0.0019	
1.0104	0.00019+0.00081		1.398	0.0023	
1.0000	0.00017+0.000090		1.577	0.0012	
1.0137	0.00018+0.00071		1.075	0.0020	

Table 4.2 and Table 4.3 give the results with $k = 3$ and $k = 8$, respectively. The performance of the results support the foregoing conclusive discussions. However, as k increases, the run-time of SMH1 increases rapidly, which is consistent with the result of the Theorem 4.1 that the algorithm is exponential to k . In particular, from the perspective of

run-time, SA_SMH1 gradually outperforms SMH1 while the result out of SA_SMH1 keeps optimal. Furthermore, by examining the run-time results of SMH2 and SMH4 for $k = 3$ and $k = 8$, the algorithms tend to have a polynomial time complexity. This means that the two algorithms can well be scaled to a much larger problem.

From the three tables, the algorithm SA_SMH1 performs both the optimality and the efficiency. Therefore, for larger k , the run-time of SMH1 becomes unacceptable, and SA_SMH1 can be applied alternatively. Table 4.4 and Table 4.5 give the results with $k = 10$ and $k = 20$, respectively. In the two tables, the result of SA_SMH1 is applied as the benchmark value and the algorithm SA_SMH3 replaces SMH3. Notice that the run-time of SA_SMH3 is the sum of the run-time of SA_SMH1 and the time of constructing relax schedules. The foregoing conclusive discussions about the performance are still valid. The run-time of SMH2 and SMH4 are very short, and the run-time of SA_SMH1 and SA_SMH3 are both acceptable.

Table 4.2. Results of Algorithms SMH1-SMH4 with $k = 3$

SMH1		SA_SMH1		SMH2	
ratio	time	ratio	time	ratio	time
1	0.00074	1	0.057	1.407	0.0022
1	0.00076	1	0.060	1.105	0.0021
1	0.00071	1	0.059	1.965	0.0022
1	0.00071	1	0.061	1.344	0.0024
1	0.00063	1	0.029	1.552	0.00071
1	0.00074	1	0.079	1.095	0.0030
SMH3				SMH4	
ratio	time		ratio	time	
1.020	0.00074+0.00075		1.351	0.0028	
1.024	0.00076+0.00086		1.105	0.0028	
1.001	0.00071+0.00089		1.965	0.0027	
1.021	0.00071+0.0011		1.309	0.0031	
1.000	0.00063+0.00014		1.552	0.0018	
1.016	0.00074+0.0013		1.095	0.0036	

Table 4.3. Results of Algorithms SMH1-SMH4 with $k = 8$

SMH1		SA_SMH1		SMH2	
ratio	time	ratio	time	ratio	time
1	5.06	1	1.70	1.294	0.0092
1	4.95	1	1.68	1.165	0.0090
1	4.60	1	1.55	1.936	0.0086
1	4.70	1	1.65	1.221	0.0094
1	3.97	1	0.76	1.518	0.0018
1	4.70	1	1.87	1.197	0.012
SMH3				SMH4	
ratio	time		ratio	time	
1.016	5.06+0.0033		1.281	0.010	
1.015	4.95+0.0036		1.169	0.0098	
1.000	4.60+0.0030		1.936	0.0092	
1.0168	4.70+0.0026		1.217	0.0091	
1.000	3.97+0.00032		1.518	0.0044	
1.018	4.70+0.0028		1.200	0.0087	

Therefore, one can conclude that all the algorithms SMH1-SMH4 show robustness for the worst cases and great performance for the normal cases. Meanwhile, the efficiency of the algorithms SMH2 and SMH4 are excellent even for different values of k , which can well be scaled to a realistic application. For the algorithms SMH1 and SMH3, they can deal with

the small k cases in an acceptable time. When k gets larger, SA_SMH1 and SA_SMH3 can be applied instead as they possess the optimality and the acceptable run-time.

Table 4.4. Results of Algorithms SMH1-SMH4 with $k = 10$

SMH1		SA_SMH1		SMH2	
ratio	time	ratio	time	ratio	time
–	–	1	3.41	1.274	0.011
–	–	1	3.48	1.222	0.012
–	–	1	3.46	1.933	0.011
–	–	1	3.65	1.234	0.014
–	–	1	1.98	1.499	0.0026
–	–	1	4.40	1.226	0.019
SA_SMH3				SMH4	
ratio	time		ratio	time	
1.014	3.41+0.0039		1.268	0.012	
1.014	3.48+0.0035		1.225	0.012	
1.000	3.46+0.0038		1.933	0.012	
1.015	3.65+0.0041		1.233	0.013	
1.000	1.98+0.00041		1.499	0.0065	
1.014	4.40+0.0038		1.232	0.013	

Table 4.5. Results of Algorithms SMH1-SMH4 with $k = 20$

SMH1		SA_SMH1		SMH2	
ratio	time	ratio	time	ratio	time
–	–	1	35.21	1.319	0.027
–	–	1	36.02	1.308	0.027
–	–	1	36.19	1.897	0.027
–	–	1	38.56	1.292	0.033
–	–	1	22.09	1.473	0.0057
–	–	1	44.26	1.263	0.055
SA_SMH3				SMH4	
ratio	time		ratio	time	
1.006	35.21+0.0062		1.320	0.026	
1.004	36.02+0.0058		1.310	0.026	
1.000	36.19+0.0062		1.897	0.026	
1.005	38.56+0.0059		1.294	0.027	
1.000	22.09+0.00086		1.473	0.013	
1.005	44.26+0.0059		1.266	0.026	

Next, a simulation of CSMH4 for CSMP4 is conducted to present the performance of the algorithm. Besides the above five assumptions, the assumption of the constraint on vehicle capacity is made.

- (6) The capacity of vehicles is of three cases: $C = 2$, $C = 5$, and $C = 8$.

Table 4.6 shows the result of CSMH4 for the case $C = 2$. Each row in the table is the average of the results of the 100 instances. The algorithm columns of the table are (1) the result of the optimal algorithm for the case that preemption is allowed, (2) the result of the on-line algorithm CSMH4, (3) the ratio (2)/(1), and (4) the run-time of CSMH4 in seconds. The result in column (1) is derived by applying the algorithm SMH1, which is the lower bound of the optimal result of CSMP4. Therefore, the ratio of the results of CSMH4 to the optimal result is better.

From Table 4.6, it is evident that the ratio column of CSMH4 never exceed 1.23, which displays the robustness (the worst case performance) of the algorithm and is also consistent with the result of the Corollary 4.6. Actually, the ratios of the algorithm are much better than the theoretical result, $2 + \frac{1}{2}$, which shows the excellent performance on the normal instances. The run-time for CSMH4 never exceeds 0.1 seconds for $k = 20$ and is much shorter for smaller k , so the efficiency of the algorithms is very high. Furthermore, by examining the run-time results of CSMH4 for the value of k from low to high, the algorithms tend to have a polynomial time complexity, that is, as k increases, the run-time of the algorithms increases as a polynomial of k . This means that the algorithm can well be scaled to a much larger problem.

Table 4.7 and Table 4.8 show the results for $C = 5$ and $C = 8$, respectively, which support the above conclusion from Table 4.6.

Table 4.6. Results of Algorithm CSMH4 with $C = 2$

	$Z(opt^P)$	$Z(CSMH4)$	Ratio	Run-time
$k = 2$	185.16	202.55	1.10	0.0039
	180.90	198.05	1.10	0.0016
	211.26	221.75	1.06	0.0015
	402.86	408.35	1.02	0.0013
	3445.89	3472.55	1.01	0.0011
$k = 5$	458.36	500.95	1.09	0.011
	462.83	506.85	1.10	0.0052
	678.90	724.40	1.07	0.0039
	1091.67	1114.70	1.02	0.0029
	13935.67	14031.35	1.01	0.0031
$k = 10$	918.03	1001.00	1.09	0.030
	919.12	1003.25	1.09	0.013
	1810.42	1926.70	1.07	0.0085
	2402.39	2572.60	1.07	0.0060
	41682.49	41899.45	1.01	0.0058
$k = 20$	1784.30	1949.85	1.09	0.078
	1797.00	1963.40	1.09	0.036
	5120.72	5437.05	1.07	0.019
	6062.34	7310.55	1.20	0.014
	157242.03	157842.25	1.00	0.014

Table 4.7. Results of Algorithm CSMH4 with $C = 5$

	$Z(opt^P)$	$Z(CSMH4)$	Ratio	Run-time
$k = 2$	185.16	202.55	1.10	0.0039
	180.90	198.05	1.10	0.0016
	211.26	221.75	1.06	0.0015
	402.86	408.35	1.02	0.0013
	3445.89	3472.55	1.01	0.0011
$k = 5$	458.36	500.95	1.09	0.011
	462.83	506.85	1.10	0.0052
	678.90	724.40	1.07	0.0039
	1091.67	1114.70	1.02	0.0030
	13935.67	14031.35	1.01	0.0031
$k = 10$	544.95	667.05	1.23	0.026
	547.78	674.65	1.23	0.013
	1310.17	1428.95	1.10	0.0092
	2069.84	2223.70	1.07	0.0060
	43187.34	43406.80	1.01	0.0056
$k = 20$	1048.92	1282.30	1.22	0.076
	1105.54	1358.50	1.23	0.038
	4378.26	4703.00	1.08	0.024
	5412.55	6579.35	1.21	0.034
	147629.56	148174.25	1.00	0.013

Table 4.8. Results of Algorithm CSMH4 with $C = 8$

	$Z(opt^P)$	$Z(CSMH4)$	Ratio	Run-time
$k = 2$	85.29	110.95	1.31	0.0030
	90.21	119.35	1.33	0.0019
	136.34	150.50	1.12	0.0015
	330.83	337.35	1.02	0.0012
	3357.10	3378.30	1.01	0.0011
$k = 5$	223.76	291.95	1.31	0.011
	223.35	294.90	1.32	0.0055
	450.89	492.85	1.11	0.0044
	838.89	861.95	1.03	0.0031
	13008.89	13111.75	1.01	0.0029
$k = 10$	432.67	563.30	1.303	0.027
	454.39	602.15	1.33	0.013
	1227.19	1343.85	1.10	0.0078
	1931.73	2105.70	1.09	0.0057
	43393.48	43587.00	1.00	0.0052
$k = 20$	866.72	1131.05	1.31	0.077
	918.00	1210.25	1.32	0.036
	4190.21	4516.20	1.08	0.019
	5227.55	6394.80	1.22	0.015
	147444.81	147989.50	1.00	0.013

4.7 Algorithm for Problem SMP5

SMP5 has the following features: Jobs are released off-line, processed in "pmtn" pattern and delivered in "routing" pattern. The capacity of vehicles and the number of vehicles are both enough.

When the number of customers k is a parameter, the delivery part is SNP-hard, which implies the problem is at least SNP hard.

Corollary 4.7. *SMP5 is a SNP-hard problem.*

As SMP5 is in off-line environment and the preemption of jobs processing is allowed, a lemma modified from Lemma 4.1 can be proposed.

Lemma 4.8. *There exists an optimal schedule for SMP5 which is a priority schedule and the priority of the customers is consistent with the order of the customers' completion time in the schedule.*

Further, the following properties are proposed to reduce the complexity of a schedule.

Property 4.1. *As there are sufficient vehicles and each vehicle can contain sufficient jobs, all jobs of the same customer should be packed in one batch.*

Property 4.2. *The departure time of a batch is the maximum completion time of the customers in it, that is, $\rho_{max}^{(i)} = \{C_{max}^{(l)} | \text{Customer } l\text{'s jobs are in the same batch with Customer } i\text{'s jobs}\}$.*

To further reduce the complexity of schedule, the analytical property of this problem is proposed as follows.

Theorem 4.9. *If the jobs of Customer i and Customer l are in the same batch in opt , there must be $T_{il} \leq D$.*

Proof of Theorem 4.9: This statement can be proved by contradiction. Suppose that in opt there are two customers i and l , whose jobs are in the same batch, with $T_{il} > D$.

Case 1: Customer i and Customer l are adjacent to each other in the routing path. Without loss of generality, let the routing path of the batch be $(0, 1, \dots, i, l, \dots, m, 0)$. Construct a new schedule opt' as follows: the processing part and the delivery of other batches are the same as opt , but the above batch is divided into two batches with routing paths $(0, 1, \dots, i, 0)$ and $(0, l, \dots, m, 0)$, respectively. Let k_1 and k_2 be the number of customers in the two batches, and T_1 be the transportation time of the path $(0, 1, \dots, i)$ and T_2 be the transportation time of the path $(l, \dots, m, 0)$. It is obvious that $T_1 \geq T_{0i}$ and $T_2 \geq T_{0l}$. Meanwhile, the departure time of the two batches in opt' will not be later than that of the original one in opt . Therefore, the difference between opt and opt' should satisfy that

$$\begin{aligned}
Z(opt') - Z(opt) &\leq [k_1(T_1 + T_{0i}) + D + k_2(T_{0l} + T_2) + D] \\
&\quad - [(k_1 + k_2)(T_1 + T_{il} + T_2) + D] \\
&= k_1T_{0i} + k_2T_{0l} - k_1T_{0l} - k_2T_{0i} - (k_1 + k_2)T_{il} + D \tag{4.107} \\
&= (k_1 - k_2)(T_{0i} - T_{0l}) - (k_1 + k_2)T_{il} + D \\
&\leq |k_1 - k_2||T_{0i} - T_{0l}| - (k_1 + k_2)T_{il} + D
\end{aligned}$$

As $|k_1 - k_2| \leq k_1 + k_2 - 2$, $|T_{0i} - T_{0l}| \leq T_{il}$,

$$Z(opt') - Z(opt) \leq -2T_{il} + D < 0. \tag{4.108}$$

This contradicts with the optimality of opt .

Case 2: Customer i and Customer l are not adjacent to each other in the routing path.

Without loss of generality, let the routing path of the batch be $(0, 1, \dots, i, s, \dots, h, l, \dots, m, 0)$. Construct a new schedule opt' as follows: the processing part and the delivery of other batches are the same as opt , but the above batch is divided into three batches with routing paths $(0, 1, \dots, i, 0)$, $(0, s, \dots, h, 0)$ and $(0, l, \dots, m, 0)$, respectively. Let k_1 , k_2 and k_3 be the number of customers in the three batches, and T_1 be the transportation time of the path $(0, 1, \dots, i, 0)$, T_2 be the transportation time of the path (s, \dots, h) and T_3 be the transportation time of the path $(l, \dots, m, 0)$, respectively. It is obvious that $T_1 \geq T_{0i}$, $T_3 \geq T_{0l}$, $T_1 + T_{is} \geq T_{0s}$, $T_{hl} + T_3 \geq T_{0l}$ and $T_{is} + T_2 + T_{hl} \geq T_{il}$. Meanwhile, the departure time of the three batches in opt' will not be later than that of the original one in opt . Therefore, the difference between opt and opt' should satisfy that

$$\begin{aligned}
Z(opt') - Z(opt) &\leq [k_1(T_1 + T_{0i}) + D + k_2(T_{0s} + T_2 + T_{0h}) + D] + k_3[(T_{0l} \\
&\quad + T_3) + D] - [(k_1 + k_2 + k_3)(T_1 + T_{is} + T_2 + T_{hl} + T_3) + D] \\
&= k_1(T_{0i} - T_2 - T_3 - T_{is} - T_{hl}) + k_2(T_{0s} + T_{0h} - T_1 - \\
&\quad T_3 - T_{is} - T_{hl}) + k_3(T_{0l} - T_1 - T_2 - T_{is} - T_{hl}) + 2D \\
&\leq k_1(T_{0i} - T_{0l} - T_{il}) + k_3(T_{0l} - T_{0i} - T_{il}) + 2D \\
&= (k_1 - k_3)(T_{0i} - T_{0l}) - (k_1 + k_3)T_{il} + D \\
&\leq |k_1 - k_3||T_{0i} - T_{0l}| - (k_1 + k_3)T_{il} + 2D \\
&\leq -2T_{il} + 2D < 0
\end{aligned} \tag{4.109}$$

This contradicts with the optimality of opt .

Therefore, the hypothesis is wrong which proves the statement of this theorem. \square

Based on the first part of Theorem 4.9, a partition of the customer set can be defined as

follows, where the jobs of any two customers in different subsets cannot be in the same batch.

Definition 4.3 [Weak Partition]. *If there is a partition WK_1, WK_2, \dots, WK_o of the customer set K such that for any $i \in WK_p, l \in WK_q$ there is $T_{il} > \frac{D}{2}$, where $p, q \in \{1, 2, \dots, o\}$ and $p \neq q$, then the partition is called a weak partition.*

A weak partition can be constructed by the following steps,

Step 0. Set $q = 1, H = \emptyset$, and goto *Step 1*.

Step 1. Pick up an arbitrary index i from K , let $K = K \setminus \{i\}$ and $WK_q = \{i\}$, and goto *Step 2*.

Step 2. If $K \neq \emptyset$, goto *Step 3*. If $K = \emptyset$ but $H \neq \emptyset, K = H, H = \emptyset, q = q + 1$ and goto *Step 1*. If $K = \emptyset$ and $H = \emptyset$, finish.

Step 3. Pick up an arbitrary index i from K . If for all $l \in WK_q$ there is $T_{il} > \frac{D}{2}$, $H = H \cup \{l\}$ and $K = K \setminus \{l\}$, goto *Step 2*; otherwise, $WK_q = WK_q \cup \{i\}, K = (K \setminus \{i\}) \cup H$ and $H = \emptyset$, goto *Step 3*.

Notice that the existence of the weak partition depend only on the objective function $\sum D_{max}^{(i)} + TC$, delivery characteristics and the routing matrix. Therefore, the analytical property can be extended to problems with different machine configurations and job parameters. Meanwhile, the extendability requires that this property can be expressed independently in the practical implementations.

A genetic algorithm (GA) combined with the above analytical property is proposed to solve SMP5. Therefore, the genetic representation and the fitness function, the popula-

tion initialization, and the genetic operators need to be discussed, respectively.

Algorithm SMH5

At first, the genetic representation of solution domain, i.e., the encoding of the individuals, is constructed. As a solution has three parts: the processing order of customers on the machine, the customers in a batch, the routing path of a batch, the individual encoding also composes three chromosomes.

- (1) The processing order of aid sites can be represented by a permutation σ_p of $\{1, 2, \dots, k\}$, that is, jobs processing is generated from the order $\sigma_p(1), \sigma_p(2), \dots, \sigma_p(k)$.
- (2) Construct a map σ_b on the customer set: $K \xrightarrow{\sigma_b} K$ to represent the allocation of customers to batches, that is, Customer i and Customer l are in the same batch if $\sigma_b(i) = \sigma_b(l)$.
- (3) For all the aid sites, a permutation σ_r of K is applied to represent the routing paths of all batches. Suppose that the customers i_1, i_2, \dots, i_s are in the same batch and they satisfies $\sigma_r(i_1) < \sigma_r(i_2) < \dots < \sigma_r(i_s)$, then the routing path of this batch will be $(0, i_1, i_2, \dots, i_s, 0)$.

As the problem is a minimization problem, $E - (\sum D_{max}^{(i)} + TC)$ is chosen as the fitness function, where E is an upper bound of the values of all meaningful solutions (note that the meaningful solution here means that there is no delay on jobs processing and delivery).

Applying analytical property: In the original GA, the population are initialized randomly, that is, random permutation σ_p , random map σ_b and random permutation σ_r . However, from the analytical property, there are more effective solutions. Therefore, based

on the weak partition WK_1, WK_2, \dots, WK_o in Definition 4.3, the map can be improved as $WK_q \xrightarrow{\sigma_b} WK_q$ for $q = 1, 2, \dots, o$, that is, the map on the subset WK_q are randomly initialized to WK_q .

Genetic operators includes selection operator, crossover operator and mutation operator, which are crucial to the genetic algorithm. The three operators in the proposed algorithm are discussed as follows.

The selection operator is to select a portion of the population to breed a new generation, which is determined by the fitness function. In the algorithm, the bigger the fitness function value is, the more likely the individual is selected.

The crossover operator is to generate two new 'son' individuals from two 'parent' individuals. For different chromosomes, the crossover methods are different. Therefore, the crossover operators for permutation and map are discussed respectively. (1) Suppose that there are two processing order permutations $(\sigma_p^1(1), \sigma_p^1(2), \dots, \sigma_p^1(k))$ and $(\sigma_p^2(1), \sigma_p^2(2), \dots, \sigma_p^2(k))$, and let k_0 be the crossover position. Let $(j_{k_0}^1, j_{k_0+1}^1, \dots, j_k^1)$ be the order of $\{\sigma_p^1(k_0), \sigma_p^1(k_0 + 1), \dots, \sigma_p^1(k)\}$ in σ_p^2 , and $(j_{k_0}^2, j_{k_0+1}^2, \dots, j_k^2)$ be the order of $\{\sigma_p^2(k_0), \sigma_p^2(k_0 + 1), \dots, \sigma_p^2(k)\}$ in σ_p^1 . The two new permutations after crossover are $(\sigma_p^1(1), \sigma_p^1(2), \dots, \sigma_p^1(k_0 - 1), j_{k_0}^1, j_{k_0+1}^1, \dots, j_k^1)$ and $(\sigma_p^2(1), \sigma_p^2(2), \dots, \sigma_p^2(k_0 - 1), j_{k_0}^2, j_{k_0+1}^2, \dots, j_k^2)$. (2) Suppose that there are two batch allocation maps $(\sigma_b^1(1), \sigma_b^1(2), \dots, \sigma_b^1(k))$ and $(\sigma_b^2(1), \sigma_b^2(2), \dots, \sigma_b^2(k))$, and let k_1 be the crossover position. The two new maps after crossover are $(\sigma_b^1(1), \sigma_b^1(2), \dots, \sigma_b^1(k_1 - 1), \sigma_b^2(k_1), \sigma_b^2(k_1 + 1), \dots, \sigma_b^2(k))$ and $(\sigma_b^2(1), \sigma_b^2(2), \dots, \sigma_b^2(k_1 - 1), \sigma_b^1(k_1), \sigma_b^1(k_1 + 1), \dots, \sigma_b^1(k))$. Notice that the two new maps will still be a weak partition. (3) The crossover of the routing permutation will be similar with that of the processing order permutation.

The mutation operator is to generate a new individual by randomly changing several genes, which also depends on the structure of chromosome. (1) Suppose that there is a processing order permutations $(\sigma_p(1), \sigma_p(2), \dots, \sigma_p(k))$ and let k_0 ($k_0 \leq k - 1$) be the mutation position. The new permutation after mutation is $(\sigma_p(1), \dots, \sigma_p(k_0 - 1), \sigma_p(k_0 + 1), \sigma_p(k_0), \sigma_p(k_0 + 2), \dots, \sigma_p(k))$. (2) Suppose that there is a batch allocation map $(\sigma_b(1), \sigma_b(2), \dots, \sigma_b(k))$ and let k_1 be the mutation position. If $k_1 \in WK_q$ ($q = 1, 2, \dots, o$), randomly pick up an index i^q from WK_q . The new map after mutation is $(\sigma_b(1), \dots, \sigma_b(k_1 - 1), i^q, \sigma_b(k_1 + 1), \dots, \sigma_b(k))$, which is a weak partition. (3) The mutation of the routing permutation will be similar with that of the processing order permutation.

After encoding the solution domain, initializing the population and defining the genetic operators, the framework of the genetic algorithm is complete. The combination of the analytical property is reflected in the batch allocation chromosome: the map σ_b . For different kinds of problems, the proposed genetic algorithm can be applied by adjusting the processing order chromosome and batch routing chromosome. Furthermore, if more properties of the processing part and the routing part are explored, they can be directly applied to the corresponding chromosome, which implies the openness of the algorithm. The performance of the algorithm will be presented in the simulated experiment.

4.8 Algorithm for Problem SMP6

SMP6 has the following features: Jobs are released on-line, processed in "pmtn" pattern and delivered in "routing" pattern. The capacity of vehicles and the number of vehicles are both enough. Actually, SMP6 is the on-line version of SMP5. The same lower bound

can be applied to this on-line problem.

Corollary 4.8. *No on-line algorithm for SMP6 can have competitive ratio less than 2, even all processing times being 0.*

Based on Theorem 4.9, especially the first half part, a strong partition of the customer set can be defined as follows.

Definition 4.4 [Strong Partition]. *If a partition SK_1, SK_2, \dots, SK_o of the customer set K satisfies that for any $i, l \in SK_q$, $T_{il} \leq \frac{D}{2}$ and for any $i \in SK_p$ there exists a $l \in SK_{p'}$ ($p \neq p'$) such that $T_{il} > \frac{D}{2}$, then the partition is called a strong partition.*

A strong partition can be constructed by the following steps:

Step 0. Set $q = 1$, $H = \emptyset$, and goto *Step 1*.

Step 1. Pick up an arbitrary index i from K , let $K = K \setminus \{i\}$ and $SK_q = \{i\}$, and goto *Step 2*.

Step 2. If $K \neq \emptyset$, goto *Step 3*. If $K = \emptyset$ but $H \neq \emptyset$, $K = H$, $H = \emptyset$, $q = q + 1$ and goto *Step 1*. If $K = \emptyset$ and $H = \emptyset$, finish.

Step 3. Pick up an arbitrary index i from K . If for all $l \in SK_q$ there is $T_{il} \leq \frac{D}{2}$, $SK_q = SK_q \cup \{i\}$ and $K = K \setminus \{i\}$, goto *Step 2*; otherwise, $H = H \cup \{i\}$ and $K = K \setminus \{i\}$, goto *Step 2*.

For the customers which are in the same subset of a strong partition, their jobs can possibly be in the same delivery batch if they are delivered at the same time point. This is the main behind idea of the algorithm for SMP6.

Algorithm SMH6

At the time t that a new job arrives, the customers are re-indexed in an increasing order of $C_{max}(\mathcal{J}_{\leq t}^{(i)})$ (If there is more than one customer with the same $C_{max}(\mathcal{J}_{\leq t}^{(i)})$, their order is the original index order). When a new job arrives or the machine is free, process jobs of the customer with the highest on-line priority.

Set $l_q = 0$ for $q = 1, 2, \dots, o$. At every time of $\frac{l}{|SK_q|}D$, where $l \geq 1$ and l is integer, if there are s_q customers in SK_q with completed jobs but no uncompleted job, and $l - l_q > |SK_q| - s_q$, deliver all their jobs in a batch, let $l_q = l$; otherwise no operation.

The routing part in the on-line algorithm can be solved by simulated annealing algorithm.

In the following, it is assumed that the optimal routing path can always be obtained.

Next, the fact that on-line algorithm SMH6 for SMP6 with $k = 2$ can achieve a competitive ratio of 2 is proved, which implies the on-line optimality of the algorithm in this special situation. The proof has different cases for whether the two customers' jobs are in the same batch or not. The performance of the algorithm for different values of k will be shown later.

Theorem 4.10. *The competitive ratio of on-line algorithm SMH6 for SMP6 with $k = 2$ is 2, which is on-line optimal.*

Proof of Theorem 4.10: If $T_{12} > \frac{1}{2}D$, from Theorem 4.9, both the problem and the algorithm degenerate to be the case that routing is not allowed. From Theorem 4.2, the competitive ratio of the on-line algorithm is 2. Therefore, in the following, it can be assumed that $T_{12} \leq \frac{1}{2}D$. From the definition of strong partition, $SK_1 = K = \{1, 2\}$.

Let η be the schedule obtained by the algorithm SMH6. Without loss of generality, suppose $C_{max}(\mathcal{J}^{(1)}) \leq C_{max}(\mathcal{J}^{(2)})$. In η , at the time of $\max\{r_j^{(i)} | J_j^{(i)} \in \mathcal{J}^{(i)}, i = 1, 2\}$, all the jobs are released and there are uncompleted jobs for Customer 2. By the algorithm SMH6, the jobs of Customer 1 will have the top priority at this moment. Therefore, the jobs of Customer 1 are also completed earlier in η , that is, $C_{max}^{(1)}(\eta) \leq C_{max}^{(2)}(\eta)$, which implies that $\rho_{max}^{(1)}(\eta) \leq \rho_{max}^{(2)}(\eta)$. Meanwhile, as there are no delay of jobs preparation, $C_{max}^{(2)}(\eta) = C_{max}(\mathcal{J}^{(1)} \cup \mathcal{J}^{(2)})$.

Case 1: In η , the last batch is only for Customer 2. At the time $\rho_{max}^{(1)}(\eta)$, there are uncompleted jobs of Customer 2, so it should be satisfied that $\rho_{max}^{(1)}(\eta) \leq (\lceil \frac{C_{max}^{(2)}(\eta)}{\frac{1}{2}D} \rceil - 1) \frac{1}{2}D$. Meanwhile, the last batch will not wait more than $\frac{1}{2}D$, $\rho_{max}^{(2)}(\eta) \leq (\lceil \frac{C_{max}^{(2)}(\eta)}{\frac{1}{2}D} \rceil + 1) \frac{1}{2}D$.

Case 1.1: In opt , the jobs of Customer 1 and Customer 2 are in the same batch. The optimal schedule opt processes all the jobs without delay and delivery them in one batch, so the optimal result $Z(opt) = 2(C_{max}(\mathcal{J}^{(1)} \cup \mathcal{J}^{(2)}) + T_{01} + T_{12} + T_{02}) + D$.

In the interval $(0, C_{max}^{(2)}(\eta)]$, there is at most one batch at every delivery point, so the delivery cost will not be more than $\lfloor \frac{C_{max}^{(2)}(\eta)}{\frac{1}{2}D} \rfloor D$.

In the interval $(C_{max}^{(2)}(\eta), \rho_{max}^{(2)}(\eta)]$, there is only one batch for Customer 2's jobs, so the delivery cost will not be more than D .

$$TC(\eta) \leq \lfloor \frac{C_{max}^{(2)}(\eta)}{\frac{1}{2}D} \rfloor D + D \leq 2C_{max}^{(2)}(\eta) + D. \quad (4.110)$$

$$\begin{aligned}
\frac{Z(\eta)}{Z(opt)} &\leq \frac{\rho_{max}^{(1)}(\eta) + T_{01} + T_{12} + T_{02} + \rho_{max}^{(2)}(\eta) + 2T_{02} + TC(\eta)}{2(C_{max}(\mathcal{J}^{(1)} \cup \mathcal{J}^{(2)}) + T_{01} + T_{12} + T_{02}) + D} \\
&\leq \frac{2\lceil \frac{C_{max}^{(2)}(\eta)}{\frac{1}{2}D} \rceil \frac{1}{2}D + 2C_{max}^{(2)}(\eta) + D + T_{01} + T_{12} + T_{02} + 2T_{02}}{2(C_{max}(\mathcal{J}^{(1)} \cup \mathcal{J}^{(2)}) + T_{01} + T_{12} + T_{02}) + D} \\
&\leq \frac{4C_{max}^{(2)}(\eta) + 2T_{01} + 2T_{12} + 2T_{02} + 2D}{2C_{max}(\mathcal{J}^{(1)} \cup \mathcal{J}^{(2)}) + 2T_{01} + 2T_{12} + 2T_{02} + D} \leq 2.
\end{aligned} \tag{4.111}$$

Case 1.2: In *opt*, the jobs of Customer 1 and Customer 2 are not in the same batch.

The optimal schedule *opt* processes all the jobs without delay such that Customer 1 has a higher priority, and deliver Customer *i*'s ($i = 1, 2$) jobs in a batch when all of them are completed, so the optimal result $Z(opt) = C_{max}(\mathcal{J}^{(1)}) + 2T_{01} + C_{max}(\mathcal{J}^{(1)} \cup \mathcal{J}^{(2)}) + 2T_{02} + 2D$.

In the interval $(0, C_{max}(\mathcal{J}^{(1)})]$, there is at most one batch at every delivery point, so the delivery cost will not be more than $\lfloor \frac{C_{max}(\mathcal{J}^{(1)})}{\frac{1}{2}D} \rfloor D$.

In the interval $(C_{max}(\mathcal{J}^{(1)}), C_{max}^{(1)}(\eta)]$, there is at most one batch for Customer 2's jobs which are released before $C_{max}(\mathcal{J}^{(1)})$, so the delivery cost will not be more than D .

In the interval $(C_{max}^{(1)}(\eta), \rho_{max}^{(1)}(\eta)]$, there is only one batch at the time $\rho_{max}^{(1)}(\eta)$, so the delivery cost will be D .

In the interval $(\rho_{max}^{(1)}(\eta), \rho_{max}^{(2)}(\eta)]$, there is at most one batch every two delivery points, so the delivery cost will not be more than $\frac{\rho_{max}^{(2)}(\eta) - \rho_{max}^{(1)}(\eta)}{\frac{1}{2}D} \frac{1}{2}D = \rho_{max}^{(2)}(\eta) - \rho_{max}^{(1)}(\eta)$.

$$\begin{aligned}
TC(\eta) &\leq \lfloor \frac{C_{max}(\mathcal{J}^{(1)})}{\frac{1}{2}D} \rfloor D + 2D + \rho_{max}^{(2)}(\eta) - \rho_{max}^{(1)}(\eta) \\
&\leq 2C_{max}(\mathcal{J}^{(1)}) + 2D + \rho_{max}^{(2)}(\eta) - \rho_{max}^{(1)}(\eta).
\end{aligned} \tag{4.112}$$

$$\begin{aligned}
\frac{Z(\eta)}{Z(\text{opt})} &\leq \frac{\rho_{\max}^{(1)}(\eta) + T_{01} + T_{12} + T_{02} + \rho_{\max}^{(2)}(\eta) + 2T_{02} + TC(\eta)}{C_{\max}(\mathcal{J}^{(1)}) + 2T_{01} + C_{\max}(\mathcal{J}^{(1)} \cup \mathcal{J}^{(2)}) + 2T_{02} + 2D} \\
&\leq \frac{2\rho_{\max}^{(2)}(\eta) + T_{01} + T_{12} + T_{02} + 2T_{02} + 2C_{\max}(\mathcal{J}^{(1)}) + 2D}{C_{\max}(\mathcal{J}^{(1)}) + 2T_{01} + C_{\max}(\mathcal{J}^{(1)} \cup \mathcal{J}^{(2)}) + 2T_{02} + 2D} \\
&\leq \frac{2C_{\max}(\mathcal{J}^{(1)}) + 2C_{\max}^{(2)}(\eta) + 4D + 2T_{01} + 4T_{02}}{C_{\max}(\mathcal{J}^{(1)}) + 2T_{01} + C_{\max}(\mathcal{J}^{(1)} \cup \mathcal{J}^{(2)}) + 2T_{02} + 2D} \leq 2.
\end{aligned} \tag{4.113}$$

Case 2: In η , the last batch is for both Customer 1 and Customer 2. Hence, $\rho_{\max}^{(1)}(\eta) = \rho_{\max}^{(2)}(\eta) \leq \lceil \frac{C_{\max}^{(2)}(\eta)}{\frac{1}{2}D} \rceil \frac{1}{2}D \leq C_{\max}^{(2)}(\eta) + \frac{1}{2}D$.

Case 2.1: In opt , the jobs of Customer 1 and Customer 2 are in the same batch. From case 1.1, $Z(\text{opt}) = 2(C_{\max}(\mathcal{J}^{(1)} \cup \mathcal{J}^{(2)}) + T_{01} + T_{12} + T_{02}) + D$.

As there is at most one batch every delivery point, so $TC(\eta) \leq \frac{\rho_{\max}^{(2)}(\eta)}{\frac{1}{2}D} D = 2\rho_{\max}^{(2)}(\eta)$.

$$\begin{aligned}
\frac{Z(\eta)}{Z(\text{opt})} &\leq \frac{2(\rho_{\max}^{(2)}(\eta) + T_{01} + T_{12} + T_{02}) + TC(\eta)}{2(C_{\max}(\mathcal{J}^{(1)} \cup \mathcal{J}^{(2)}) + T_{01} + T_{12} + T_{02}) + D} \\
&\leq \frac{4\rho_{\max}^{(2)}(\eta) + 2T_{01} + 2T_{12} + 2T_{02}}{2(C_{\max}(\mathcal{J}^{(1)} \cup \mathcal{J}^{(2)}) + T_{01} + T_{12} + T_{02}) + D} \\
&\leq \frac{4C_{\max}^{(2)}(\eta) + 2T_{01} + 2T_{12} + 2T_{02} + 2D}{2C_{\max}(\mathcal{J}^{(1)} \cup \mathcal{J}^{(2)}) + 2T_{01} + 2T_{12} + 2T_{02} + D} \leq 2.
\end{aligned} \tag{4.114}$$

Case 2.2: In opt , the jobs of Customer 1 and Customer 2 are not in the same batch.

From case 1.2, $Z(\text{opt}) = C_{\max}(\mathcal{J}^{(1)}) + 2T_{01} + C_{\max}(\mathcal{J}^{(1)} \cup \mathcal{J}^{(2)}) + 2T_{02} + 2D$.

In the interval $(0, C_{\max}(\mathcal{J}^{(1)})]$, there is at most one batch at every delivery point, so the delivery cost will not be more than $\lfloor \frac{C_{\max}(\mathcal{J}^{(1)})}{\frac{1}{2}D} \rfloor D$.

In the interval $(C_{\max}(\mathcal{J}^{(1)}), C_{\max}^{(1)}(\eta)]$, there is at most one batch for Customer 2's jobs which are released before $C_{\max}(\mathcal{J}^{(1)})$, so the delivery cost will not be more than D .

In the interval $(C_{\max}^{(1)}(\eta), \rho_{\max}^{(2)}(\eta)]$, there is one batch at the time $\rho_{\max}^{(2)}(\eta)$, so the delivery cost will not be more than D .

$$TC(\eta) \leq \lfloor \frac{C_{\max}(\mathcal{J}^{(1)})}{\frac{1}{2}D} \rfloor D + 2D \leq 2C_{\max}(\mathcal{J}^{(1)}) + 2D \tag{4.115}$$

$$\begin{aligned}
\frac{Z(\eta)}{Z(opt)} &\leq \frac{2(\rho_{max}^{(2)}(\eta) + T_{01} + T_{12} + T_{02}) + TC(\eta)}{C_{max}(\mathcal{J}^{(1)}) + 2T_{01} + C_{max}(\mathcal{J}^{(1)} \cup \mathcal{J}^{(2)}) + 2T_{02} + 2D} \\
&\leq \frac{2\rho_{max}^{(2)}(\eta) + 2T_{01} + 2T_{12} + 2T_{02} + 2C_{max}(\mathcal{J}^{(1)}) + 2D}{C_{max}(\mathcal{J}^{(1)}) + 2T_{01} + C_{max}(\mathcal{J}^{(1)} \cup \mathcal{J}^{(2)}) + 2T_{02} + 2D} \\
&\leq \frac{2C_{max}(\mathcal{J}^{(1)}) + 2C_{max}^{(2)}(\eta) + 2T_{01} + 2T_{12} + 2T_{02} + 3D}{C_{max}(\mathcal{J}^{(1)}) + 2T_{01} + C_{max}(\mathcal{J}^{(1)} \cup \mathcal{J}^{(2)}) + 2T_{02} + 2D} \leq 2.
\end{aligned} \tag{4.116}$$

From Corollary 4.8, there is no on-line algorithm with competitive ratio less than 2, which completes the proof. \square

4.9 Algorithm for Problem SMP7

SMP7 has the following features: Jobs are released off-line and delivered in "routing" pattern. The capacity of vehicles and the number of vehicles are both enough.

When the number of customers k is a parameter, both the preparation part and the delivery part are strongly NP-hard, respectively, which implies SMP7 is at least strongly NP hard.

Corollary 4.9. *SMP7 is a SNP-hard problem.*

Property 4.1 and 4.2 can still be applied for SMP7, and the jobs of any two aid sites in different subsets of a weak partition cannot be in the same batch, which means that a GA for SMP7 can be modified from SMH5.

Algorithm SMH7

At first, the genetic representation of solution domain, i.e., the encoding of the individuals, is constructed. As a solution has three parts: the processing order of jobs on the processor, the aid sites in a batch, the routing path of a batch, the individual encoding also composes three chromosomes.

- (1) Re-index all the jobs by the release time, and the processing order of jobs can be represented by a permutation σ_p of $\{1, 2, \dots, n\}$, that is, jobs processing order is $\sigma_p(1), \sigma_p(2), \dots, \sigma_p(n)$.
- (2) Construct a map σ_b on the customer set: $K \xrightarrow{\sigma_b} K$ to represent the allocation of customers to batches, that is, Customer i and Customer l are in the same batch if $\sigma_b(i) = \sigma_b(l)$.
- (3) For all the customers, a permutation σ_r of K is applied to represent the routing paths of all batches. Suppose that the customers i_1, i_2, \dots, i_s are in the same batch and they satisfies $\sigma_r(i_1) < \sigma_r(i_2) < \dots < \sigma_r(i_s)$, then the routing path of this batch will be $(0, i_1, i_2, \dots, i_s, 0)$.

Similarly, $E - (\sum D_{max}^{(i)} + TC)$ is chosen as the fitness function, where E is an upper bound of the values of all meaningful solutions.

Applying analytical property: Based on the weak partition WK_1, WK_2, \dots, WK_o in Definition 4.1, the map σ_b can be improved as $WK_q \xrightarrow{\sigma_b} WK_q$ for $q = 1, 2, \dots, o$, that is, the map on the subset WK_q are randomly initialized to WK_q .

The three operators in the proposed algorithm are discussed as follows.

The selection operator is to select a portion of the population to breed a new generation, which is determined by the fitness function. In the algorithm, the bigger the fitness function value is, the more likely the individual is selected.

The crossover operator is to generate two new 'son' individuals from two 'parent' individuals. For different chromosomes, the crossover methods are different. Therefore, the crossover operators for permutation and map are discussed respectively. (1)

Suppose that there are two processing order permutations $(\sigma_p^1(1), \sigma_p^1(2), \dots, \sigma_p^1(n))$ and $(\sigma_p^2(1), \sigma_p^2(2), \dots, \sigma_p^2(n))$, and let n_0 be the crossover position. Let $(j_{n_0}^1, j_{n_0+1}^1, \dots, j_n^1)$ be the order of $\{\sigma_p^1(n_0), \sigma_p^1(n_0 + 1), \dots, \sigma_p^1(n)\}$ in σ_p^2 , and $(j_{n_0}^2, j_{n_0+1}^2, \dots, j_n^2)$ be the order of $\{\sigma_p^2(n_0), \sigma_p^2(n_0 + 1), \dots, \sigma_p^2(n)\}$ in σ_p^1 . The two new permutations after crossover are $(\sigma_p^1(1), \sigma_p^1(2), \dots, \sigma_p^1(n_0 - 1), j_{n_0}^1, j_{n_0+1}^1, \dots, j_n^1)$ and $(\sigma_p^2(1), \sigma_p^2(2), \dots, \sigma_p^2(n_0 - 1), j_{n_0}^2, j_{n_0+1}^2, \dots, j_n^2)$. (2) Suppose that there are two batch allocation maps $(\sigma_b^1(1), \sigma_b^1(2), \dots, \sigma_b^1(k))$ and $(\sigma_b^2(1), \sigma_b^2(2), \dots, \sigma_b^2(k))$, and let k_0 be the crossover position. The two new maps after crossover are $(\sigma_b^1(1), \sigma_b^1(2), \dots, \sigma_b^1(k_0 - 1), \sigma_b^2(k_0), \sigma_b^2(k_0 + 1), \dots, \sigma_b^2(k))$ and $(\sigma_b^2(1), \sigma_b^2(2), \dots, \sigma_b^2(k_0 - 1), \sigma_b^1(k_0), \sigma_b^1(k_0 + 1), \dots, \sigma_b^1(k))$. Notice that the two new maps will still be a weak partition. (3) The crossover of the routing permutation will be similar with that of the preparation order permutation.

The mutation operator is to generate a new individual by randomly changing several genes, which also depends on the structure of chromosome. (1) Suppose that there is a processing order permutations $(\sigma_p(1), \sigma_p(2), \dots, \sigma_p(n))$ and let n_0 ($n_0 \leq n - 1$) be the mutation position. The new permutation after mutation is $(\sigma_p(1), \dots, \sigma_p(n_0 - 1), \sigma_p(n_0 + 1), \sigma_p(n_0), \sigma_p(n_0 + 2), \dots, \sigma_p(n))$. (2) Suppose that there is a batch allocation map $(\sigma_b(1), \sigma_b(2), \dots, \sigma_b(k))$ and let k_0 be the mutation position. If $k_0 \in WK_q$ ($q = 1, 2, \dots, o$), randomly pick up an index i^q from WK_q . The new map after mutation is $(\sigma_b(1), \dots, \sigma_b(k_0 - 1), i^q, \sigma_b(k_0 + 1), \dots, \sigma_b(k))$, which is a weak partition. (3) The mutation of the routing permutation will be similar with that of the processing order permutation.

The performance of the algorithm will be presented in the simulated experiment.

4.10 Algorithm for Problem SMP8

SMP8 has the following features: Jobs are released on-line and delivered in "routing" pattern. The capacity of vehicles and the number of vehicles are both enough. Actually, SMP8 is the on-line version of SMP7. The same lower bound can be applied to this on-line problem.

Corollary 4.10. *No on-line algorithm for SMP8 can have competitive ratio less than 2, even all processing times being 0.*

Similarly with SMP6, for the customers which are in the same subset of a strong partition, their jobs can possibly be in the same delivery batch if they are delivered at the same time point.

Algorithm SMH8

At the time t that a new job arrives, the customers are re-indexed in an increasing order of $C_{max}(\mathcal{J}_{\leq t}^{(i)})$ (If there is more than one customer with the same $C_{max}(\mathcal{J}_{\leq t}^{(i)})$, their order is the original index order). When the machine is free, process ready jobs of the customer with the highest on-line priority.

Set $l_q = 0$ for $q = 1, 2, \dots, o$. At every time of $\frac{l}{|SK_q|}D$, where $l \geq 1$ and l is integer, if there are s_q customers in SK_q with completed jobs but no uncompleted job, and $l - l_q > |SK_q| - s_q$, deliver all their jobs in a batch, let $l_q = l$; otherwise no operation.

Next, the competitive analysis of SMH8 with $k = 2$ is proposed and the performance of the algorithm is presented in the next section. As the preemption of jobs processing is not

allowed, the proof is more complex and the completion times of the two customers need to be considered.

Theorem 4.11. *The on-line algorithm SMH8 for SMP8 with $k = 2$ is $2 + \frac{1}{2}$ -competitive.*

Proof of Theorem 4.11: If $T_{12} > \frac{1}{2}D$, from Theorem 4.9, both the problem and the algorithm degenerate to the case that routing is not allowed. From Theorem 4.8, the on-line algorithm is $2 + \frac{1}{2}$ -competitive. Therefore, in the following, it is assumed that $T_{12} \leq \frac{1}{2}D$. From the definition of strong partition, $SK_1 = K = \{1, 2\}$.

Without loss of generality, suppose that $C_{max}(\mathcal{J}^{(1)}) \leq C_{max}(\mathcal{J}^{(2)})$. Let η be the algorithm solution and $\frac{1}{2}D$ be the last idle delivery point before $C_{max}(\mathcal{J}^{(1)})$. Let (i_1, i_2) be the order of the customers' completion times in the optimal solutions. It is obvious that $C_{max}(\mathcal{J}^{(1)}) \leq C_{max}^{(i_1)}(opt)$ and $C_{max}(\mathcal{J}^{(1)} \cup \mathcal{J}^{(2)}) \leq C_{max}^{(i_2)}(opt)$. In addition, let $T_d^{(i)}$ be the end point of the last period of waiting time before $C_{max}^{(i)}(\eta)$ ($i = 1, 2$) in η . Here assume $T_d^{(1)} > \frac{1}{2}D$; otherwise the waiting will not affect the completion times of two customers and will lead to a simpler case. The total waiting time will not be greater than half of the longest processing time of all jobs. At the time of $C_{max}(\mathcal{J}^{(1)})$, all the jobs of Customer 1 are released and also satisfy the processing condition, so $T_d^{(1)} \leq C_{max}(\mathcal{J}^{(1)})$. Meanwhile, after $C_{max}(\mathcal{J}^{(1)})$, the processing of Customer 1's jobs would be continuous until all of them are completed. Therefore, the jobs processing in the interval $(C_{max}(\mathcal{J}^{(1)}), C_{max}^{(1)}(\eta)]$ should be a block of Customer 2's jobs followed by a block of Customer 1's jobs (see Figure 4.9). Simply use P_1 and P_2 to represent $P_{(C_{max}(\mathcal{J}^{(1)}), C_{max}^{(1)}(\eta))}^{(1)}(\eta)$ and $P_{(C_{max}(\mathcal{J}^{(1)}), C_{max}^{(1)}(\eta))}^{(2)}(\eta)$, respectively.

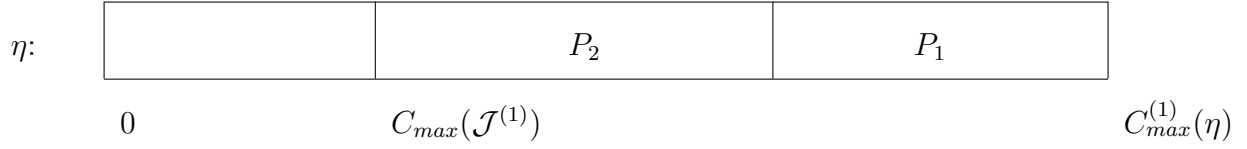


Figure 4.9. The Jobs Processing in The Interval $(C_{max}(\mathcal{J}^{(1)}), C_{max}^{(1)}(\eta)]$ (2)

If $C_{max}(\mathcal{J}^{(1)} \cup \mathcal{J}^{(2)}) \leq \frac{1}{4}D$, the longest processing time will not be more than $\frac{1}{4}D$, and the total waiting time will not be more than $\frac{1}{8}D$, which implies that $\max\{C_{max}^{(1)}(\eta), C_{max}^{(2)}(\eta)\} \leq \frac{3}{8}D$. Further, the first delivery point is at the time $\frac{1}{2}D$, so there is only one batch for all jobs in η at the time $\frac{1}{2}D$. As such, there is $Z(\eta) = 2(\frac{1}{2}D + T_{01} + T_{12} + T_{02}) + D$. As the optimal result $Z(opt) \geq \min\{D + 2(T_{01} + T_{12} + T_{02}), 2D + 2T_{01} + 2T_{02}\} \geq D + 2(T_{01} + T_{12} + T_{02})$,

$$\frac{Z(\eta)}{Z(opt)} \leq \frac{2D + 2T_{01} + 2T_{12} + 2T_{02}}{D + 2T_{01} + 2T_{12} + 2T_{02}} \leq 2.$$

If $C_{max}(\mathcal{J}^{(1)} \cup \mathcal{J}^{(2)}) \leq \frac{1}{2}D$ and $C_{max}(\mathcal{J}^{(1)} \cup \mathcal{J}^{(2)}) > \frac{1}{4}D$, the longest processing time will not be more than $\frac{1}{2}D$ and the total waiting time will not be more than $\frac{1}{4}D$, which implies that $\max\{C_{max}^{(1)}(\eta), C_{max}^{(2)}(\eta)\} \leq \frac{3}{4}D$. All the jobs are known before $\frac{1}{2}D$, so in η there is one batch for all jobs at the delivery point no later than D . Therefore, $Z(\eta) \leq 2(D + T_{01} + T_{12} + T_{02}) + D$. As the optimal result $Z(opt) \geq \min\{2 \cdot \frac{1}{4}D + D + 2(T_{01} + T_{12} + T_{02}), \frac{1}{4}D + 2D + 2T_{01} + 2T_{02}\}$, $\frac{Z(\eta)}{Z(opt)} \leq \max\{\frac{3D + 2T_{01} + 2T_{12} + 2T_{02}}{\frac{3}{2}D + 2T_{01} + 2T_{12} + 2T_{02}}, \frac{3D + 2T_{01} + 2T_{12} + 2T_{02}}{\frac{1}{4}D + 2D + 2T_{01} + 2T_{02}}\} \leq 2$.

It only needs to consider the case that $C_{max}(\mathcal{J}^{(1)} \cup \mathcal{J}^{(2)}) > \frac{1}{2}D$ in the following.

Case 1: $C_{max}^{(1)}(\eta) \leq C_{max}^{(2)}(\eta)$. Therefore, $T_d^{(1)} \leq T_d^{(2)}$ and $C_{max}^{(2)}(\eta) \leq T_d^{(2)} + C_{max}(\mathcal{J}^{(1)} \cup \mathcal{J}^{(2)}) - \frac{1}{2}D$

Case 1.1: In η , the last batch is only for Customer 2. In this case, $\rho_{max}^{(2)}(\eta) \leq (\lceil \frac{C_{max}^{(2)}(\eta)}{\frac{D}{2}} \rceil + 1) \frac{D}{2} \leq C_{max}^{(2)}(\eta) + D$.

In the interval $(0, C_{max}(\mathcal{J}^{(1)}))$, there is at most one batch for every delivery point

before $\frac{l_1}{2}D$ and there is at most one batch for every two delivery points after $\frac{l_1}{2}D$, so the delivery cost will not be more than $\frac{l_1}{2}D + \lfloor \frac{C_{max}(\mathcal{J}^{(1)})}{\frac{l_1}{2}D} \rfloor \frac{D}{2}$.

In the interval $(C_{max}(\mathcal{J}^{(1)}), C_{max}^{(1)}(\eta)]$, there is at most one batch only for jobs of Customer 2, so the delivery cost will not be more than sD ($s = 0$ or $s = 1$).

In the interval $(C_{max}^{(1)}(\eta), \rho_{max}^{(1)}(\eta)]$, there is only one batch at the time $\rho_{max}^{(1)}(\eta)$, so the delivery cost will be D .

In the interval $(\rho_{max}^{(1)}(\eta), \rho_{max}^{(2)}(\eta)]$, there is at most one batch for every two delivery points, so the delivery cost will not be more than $\frac{\rho_{max}^{(2)}(\eta) - \rho_{max}^{(1)}(\eta)}{\frac{l_1}{2}D} \frac{l_1}{2}D = \rho_{max}^{(2)}(\eta) - \rho_{max}^{(1)}(\eta)$.

Case 1.1.1: In opt , the jobs of Customer 1 and Customer 2 are in the same batch. Then, $Z(opt) = 2(C_{max}^{(i_2)}(opt) + T_{01} + T_{12} + T_{02}) + D$.

If $s = 0$, $TC(\eta) \leq \frac{l_1}{2}D + C_{max}(\mathcal{J}^{(1)}) + D + \rho_{max}^{(2)}(\eta) - \rho_{max}^{(1)}(\eta)$.

$$\begin{aligned}
Z(\eta) &\leq \rho_{max}^{(1)}(\eta) + T_{01} + T_{12} + T_{02} + \rho_{max}^{(2)}(\eta) + 2T_{02} + TC(\eta) \\
&\leq 2\rho_{max}^{(2)}(\eta) + \frac{l_1}{2}D + C_{max}(\mathcal{J}^{(1)}) + D + T_{01} + T_{12} + 3T_{02} \\
&\leq 2C_{max}^{(2)}(\eta) + 2D + \frac{l_1}{2}D + C_{max}(\mathcal{J}^{(1)}) + D + T_{01} + T_{12} + 3T_{02} \\
&\leq 2(T_d^{(2)} + C_{max}(\mathcal{J}^{(1)} \cup \mathcal{J}^{(2)}) - \frac{l_1}{2}D) + \frac{l_1}{2}D + C_{max}(\mathcal{J}^{(1)}) + \\
&\quad 3D + T_{01} + T_{12} + 3T_{02} \\
&\leq 2C_{max}(\mathcal{J}^{(1)} \cup \mathcal{J}^{(2)}) + 2T_d^{(2)} + C_{max}(\mathcal{J}^{(1)}) + 3D + T_{01} + T_{12} + 3T_{02} \\
&\leq 4C_{max}(\mathcal{J}^{(1)} \cup \mathcal{J}^{(2)}) + 3D + T_{01} + T_{12} + 3T_{02}.
\end{aligned} \tag{4.117}$$

$$\frac{Z(\eta)}{Z(opt)} \leq \frac{4C_{max}(\mathcal{J}^{(1)} \cup \mathcal{J}^{(2)}) + 3D + T_{01} + T_{12} + 3T_{02}}{2C_{max}^{(i_2)}(opt) + 2T_{01} + 2T_{12} + 2T_{02} + D} \leq 2 + \frac{1}{2}. \tag{4.118}$$

If $s = 1$ and the batch for Customer 2 in the interval $(C_{max}(\mathcal{J}^{(1)}), C_{max}^{(1)}(\eta)]$ is the first batch in η , all Customer 2's jobs completed after this batch are released after it.

Meanwhile, this batch should not happen before the time D , so $C_{max}^{(i_2)}(opt) \geq D$ and $TC(\eta) \leq 2D + \rho_{max}^{(2)}(\eta) - \rho_{max}^{(1)}(\eta)$.

$$\begin{aligned}
Z(\eta) &\leq \rho_{max}^{(1)}(\eta) + T_{01} + T_{12} + T_{02} + \rho_{max}^{(2)}(\eta) + 2T_{02} + TC(\eta) \\
&\leq 2\rho_{max}^{(2)}(\eta) + 2D + T_{01} + T_{12} + 3T_{02} \\
&\leq 2C_{max}^{(2)}(\eta) + 4D + T_{01} + T_{12} + 3T_{02} \\
&\leq 2(T_d^{(2)} + C_{max}(\mathcal{J}^{(1)} \cup \mathcal{J}^{(2)}) - \frac{l_1}{2}D) + 4D + T_{01} + T_{12} + 3T_{02} \\
&\leq 2C_{max}(\mathcal{J}^{(1)} \cup \mathcal{J}^{(2)}) + 2T_d^{(2)} + 4D + T_{01} + T_{12} + 3T_{02} \\
&\leq 3C_{max}(\mathcal{J}^{(1)} \cup \mathcal{J}^{(2)}) + 4D + T_{01} + T_{12} + 3T_{02}.
\end{aligned} \tag{4.119}$$

$$\begin{aligned}
\frac{Z(\eta)}{Z(opt)} &\leq \frac{3C_{max}(\mathcal{J}^{(1)} \cup \mathcal{J}^{(2)}) + 4D + T_{01} + T_{12} + 3T_{02}}{2C_{max}^{(i_2)}(opt) + 2T_{01} + 2T_{12} + 2T_{02} + D} \\
&\leq 2 + \frac{2D - C_{max}^{(i_2)}(opt)}{2C_{max}^{(i_2)}(opt) + 2T_{01} + 2T_{12} + 2T_{02} + D} \\
&\leq 2 + \frac{2D - D}{2D + 2T_{01} + 2T_{12} + 2T_{02} + D} \leq 2 + \frac{1}{3}.
\end{aligned} \tag{4.120}$$

If $s = 1$ and the batch for Customer 2 in the interval $(C_{max}(\mathcal{J}^{(1)}), C_{max}^{(1)}(\eta)]$ is not the first batch in η , all Customer 2's jobs completed after this batch are released after it. Meanwhile, this batch should not happen before the time $\frac{3}{2}D$, so $C_{max}^{(i_2)}(opt) \geq \frac{3}{2}D$ and

$$TC(\eta) \leq \frac{l_1}{2}D + C_{max}(\mathcal{J}^{(1)}) + 2D + \rho_{max}^{(2)}(\eta) - \rho_{max}^{(1)}(\eta).$$

$$\begin{aligned}
Z(\eta) &\leq \rho_{max}^{(1)}(\eta) + T_{01} + T_{12} + T_{02} + \rho_{max}^{(2)}(\eta) + 2T_{02} + TC(\eta) \\
&\leq 2\rho_{max}^{(2)}(\eta) + \frac{l_1}{2}D + C_{max}(\mathcal{J}^{(1)}) + 2D + T_{01} + T_{12} + 3T_{02} \\
&\leq 2C_{max}^{(2)}(\eta) + 2D + \frac{l_1}{2}D + C_{max}(\mathcal{J}^{(1)}) + 2D + T_{01} + T_{12} + 3T_{02} \\
&\leq 2(T_d^{(2)} + C_{max}(\mathcal{J}^{(1)} \cup \mathcal{J}^{(2)}) - \frac{l_1}{2}D) + \frac{l_1}{2}D + C_{max}(\mathcal{J}^{(1)}) + \\
&\quad 4D + T_{01} + T_{12} + 3T_{02} \\
&\leq 2C_{max}(\mathcal{J}^{(1)} \cup \mathcal{J}^{(2)}) + 2T_d^{(2)} + C_{max}(\mathcal{J}^{(1)}) + 4D + T_{01} + T_{12} + 3T_{02} \\
&\leq 4C_{max}(\mathcal{J}^{(1)} \cup \mathcal{J}^{(2)}) + 4D + T_{01} + T_{12} + 3T_{02}.
\end{aligned} \tag{4.121}$$

$$\begin{aligned}
\frac{Z(\eta)}{Z(opt)} &\leq \frac{4C_{max}(\mathcal{J}^{(1)} \cup \mathcal{J}^{(2)}) + 4D + T_{01} + T_{12} + 3T_{02}}{2C_{max}^{(i_2)}(opt) + 2T_{01} + 2T_{12} + 2T_{02} + D} \\
&\leq 2 + \frac{2D}{2C_{max}^{(i_2)}(opt) + 2T_{01} + 2T_{12} + 2T_{02} + D} \\
&\leq 2 + \frac{2D}{2 \cdot \frac{3}{2}D + 2T_{01} + 2T_{12} + 2T_{02} + D} \leq 2 + \frac{1}{2}.
\end{aligned} \tag{4.122}$$

Case 1.1.2: In opt , the jobs of Customer 1 and Customer 2 are not in the same batch.

Then, $Z(opt) = C_{max}^{(i_1)}(opt) + C_{max}^{(i_2)}(opt) + 2T_{01} + 2T_{02} + 2D$.

If $T_d^{(1)} = T_d^{(2)}$, then $T_d^{(2)} \leq C_{max}(\mathcal{J}^{(1)})$.

$$\begin{aligned}
Z(\eta) &\leq 2C_{max}(\mathcal{J}^{(1)} \cup \mathcal{J}^{(2)}) + 2T_d^{(2)} + C_{max}(\mathcal{J}^{(1)}) + 4D + T_{01} + T_{12} + 3T_{02} \\
&\leq 2C_{max}(\mathcal{J}^{(1)} \cup \mathcal{J}^{(2)}) + 3C_{max}(\mathcal{J}^{(1)}) + 4D + T_{01} + T_{12} + 3T_{02}.
\end{aligned} \tag{4.123}$$

$$\begin{aligned}
\frac{Z(\eta)}{Z(opt)} &\leq \frac{2C_{max}(\mathcal{J}^{(1)} \cup \mathcal{J}^{(2)}) + 3C_{max}(\mathcal{J}^{(1)}) + 4D + T_{01} + T_{12} + 3T_{02}}{C_{max}^{(i_1)}(opt) + C_{max}^{(i_2)}(opt) + 2T_{01} + 2T_{02} + 2D} \\
&\leq 2 + \frac{C_{max}(\mathcal{J}^{(1)})}{C_{max}^{(i_1)}(opt) + C_{max}^{(i_2)}(opt) + 2T_{01} + 2T_{02} + 2D} \leq 2 + \frac{1}{2}.
\end{aligned} \tag{4.124}$$

If $T_d^{(1)} < T_d^{(2)}$ and $2T_d^{(2)} \leq \rho_{max}^{(1)}(\eta)$, there is $T_d^{(2)} > C_{max}^{(1)}(\eta)$, as there is no free machine time in the interval $(C_{max}(\mathcal{J}^{(1)}), C_{max}^{(1)}(\eta)]$. From $\rho_{max}^{(1)}(\eta) \leq (\lceil \frac{C_{max}^{(1)}(\eta)}{D} \rceil + 1) \frac{D}{2} \leq$

$C_{max}^{(1)}(\eta) + D$, we have $T_d^{(2)} \leq D$. There is only one delivery point before $C_{max}^{(1)}(\eta)$, which means that there cannot be one batch only for Customer 2, so $s = 0$, i.e., $TC(\eta) \leq \frac{l_1}{2}D + C_{max}(\mathcal{J}^{(1)}) + D + \rho_{max}^{(2)}(\eta) - \rho_{max}^{(1)}(\eta)$.

$$\begin{aligned}
Z(\eta) &\leq 2\rho_{max}^{(2)}(\eta) + D + \frac{l_1}{2}D + C_{max}(\mathcal{J}^{(1)}) + D + T_{01} + T_{12} + 3T_{02} \\
&\leq 2C_{max}^{(2)}(\eta) + 3D + \frac{l_1}{2}D + C_{max}(\mathcal{J}^{(1)}) + D + T_{01} + T_{12} + 3T_{02} \\
&\leq 2(T_d^{(2)} + C_{max}(\mathcal{J}^{(1)} \cup \mathcal{J}^{(2)}) - \frac{l_1}{2}D) + 3D + \frac{l_1}{2}D + C_{max}(\mathcal{J}^{(1)}) + \\
&\quad T_{01} + T_{12} + 3T_{02} \\
&\leq 2C_{max}(\mathcal{J}^{(1)} \cup \mathcal{J}^{(2)}) + C_{max}(\mathcal{J}^{(1)}) + 2T_d^{(2)} + 3D + T_{01} + T_{12} + 3T_{02} \\
&\leq 2C_{max}(\mathcal{J}^{(1)} \cup \mathcal{J}^{(2)}) + C_{max}(\mathcal{J}^{(1)}) + 5D + T_{01} + T_{12} + 3T_{02}.
\end{aligned} \tag{4.125}$$

$$\frac{Z(\eta)}{Z(opt)} \leq \frac{2C_{max}(\mathcal{J}^{(1)} \cup \mathcal{J}^{(2)}) + C_{max}(\mathcal{J}^{(1)}) + 5D + T_{01} + T_{12} + 3T_{02}}{C_{max}^{(i_1)}(opt) + C_{max}^{(i_2)}(opt) + 2T_{01} + 2T_{02} + 2D} \leq 2 + \frac{1}{2}. \tag{4.126}$$

If $T_d^{(1)} < T_d^{(2)}$ and $2T_d^{(2)} > \rho_{max}^{(1)}(\eta)$, there are no batches for Customer 2 in the interval $(T_d^{(2)}, 3T_d^{(2)}]$ as the job (which is in waiting status before $T_d^{(2)}$) cannot be completed before $3T_d^{(2)}$. Therefore, in the interval $(\rho_{max}^{(1)}(\eta), \rho_{max}^{(2)}(\eta)]$, the delivery cost will not be more than $\rho_{max}^{(2)}(\eta) - \lfloor \frac{3T_d^{(2)}}{2} \rfloor \frac{1}{2}D + \frac{1}{2}D \leq \rho_{max}^{(2)}(\eta) - 3T_d^{(2)} + D$.

$$TC(\eta) \leq \frac{l_1}{2}D + C_{max}(\mathcal{J}^{(1)}) + 2D + \rho_{max}^{(2)}(\eta) - 3T_d^{(2)} + D. \tag{4.127}$$

$$\begin{aligned}
Z(\eta) &\leq \rho_{max}^{(1)}(\eta) + 2\rho_{max}^{(2)}(\eta) - 3T_d^{(2)} + 3D + C_{max}(\mathcal{J}^{(1)}) + \frac{l_1}{2}D + \\
&\quad T_{01} + T_{12} + 3T_{02} \\
&\leq 2C_{max}^{(2)}(\eta) - T_d^{(2)} + 5D + C_{max}(\mathcal{J}^{(1)}) + \frac{l_1}{2}D + T_{01} + T_{12} + 3T_{02} \\
&\leq 2(T_d^{(2)} + C_{max}(\mathcal{J}^{(1)} \cup \mathcal{J}^{(2)}) - \frac{l_1}{2}D) - T_d^{(2)} + 5D + C_{max}(\mathcal{J}^{(1)}) + \\
&\quad \frac{l_1}{2}D + T_{01} + T_{12} + 3T_{02} \\
&\leq \frac{5}{2}C_{max}(\mathcal{J}^{(1)} \cup \mathcal{J}^{(2)}) + C_{max}(\mathcal{J}^{(1)}) + 5D + T_{01} + T_{12} + 3T_{02}. \\
\frac{Z(\eta)}{Z(opt)} &\leq \frac{\frac{5}{2}C_{max}(\mathcal{J}^{(1)} \cup \mathcal{J}^{(2)}) + C_{max}(\mathcal{J}^{(1)}) + 5D + T_{01} + T_{12} + 3T_{02}}{C_{max}^{(i_1)}(opt) + C_{max}^{(i_2)}(opt) + 2T_{01} + 2T_{02} + 2D} \leq 2 + \frac{1}{2}. \quad (4.129)
\end{aligned}$$

Case 1.2: In η , the last batch is for both Customer 1 and Customer 2. In this case,

$$\rho_{max}^{(1)}(\eta) = \rho_{max}^{(2)}(\eta) \leq \lceil \frac{C_{max}^{(2)}(\eta)}{\frac{D}{2}} \rceil \frac{D}{2} \leq C_{max}^{(2)}(\eta) + \frac{1}{2}D.$$

In the interval $(0, C_{max}(\mathcal{J}^{(1)})]$, the delivery cost will not be more than $\frac{l_1}{2}D + \lfloor \frac{C_{max}(\mathcal{J}^{(1)})}{\frac{D}{2}} \rfloor \frac{D}{2}$.

In the interval $(C_{max}(\mathcal{J}^{(1)}), C_{max}^{(1)}(\eta)]$, there is at most one batch only for jobs of Customer 2, so the delivery cost will not be more than sD ($s = 0$ or $s = 1$).

In the interval $(C_{max}^{(1)}(\eta), \rho_{max}^{(2)}(\eta)]$, there is only one batch at the time $\rho_{max}^{(2)}(\eta)$, so the delivery cost will be D .

$$TC(\eta) \leq \frac{l_1}{2}D + C_{max}(\mathcal{J}^{(1)}) + (s + 1)D. \quad (4.130)$$

Case 1.2.1: In opt , the jobs of Customer 1 and Customer 2 are in the same batch.

Then, $Z(opt) = 2(C_{max}^{(i_2)}(opt) + T_{01} + T_{12} + T_{02}) + D$.

$$\begin{aligned}
Z(\eta) &\leq \rho_{max}^{(1)}(\eta) + T_{01} + T_{12} + T_{02} + \rho_{max}^{(2)}(\eta) + T_{01} + T_{12} + T_{02} + TC(\eta) \\
&\leq 2C_{max}^{(2)}(\eta) + \frac{l_1}{2}D + C_{max}(\mathcal{J}^{(1)}) + (s+2)D + 2(T_{01} + T_{12} + T_{02}) \\
&\leq 2(T_d^{(2)} + C_{max}(\mathcal{J}^{(1)} \cup \mathcal{J}^{(2)}) - \frac{l_1}{2}D) + \frac{l_1}{2}D + C_{max}(\mathcal{J}^{(1)}) + \\
&\quad (s+2)D + 2(T_{01} + T_{12} + T_{02}) \\
&\leq 4C_{max}(\mathcal{J}^{(1)} \cup \mathcal{J}^{(2)}) + 3D + 2(T_{01} + T_{12} + T_{02}).
\end{aligned} \tag{4.131}$$

$$\frac{Z(\eta)}{Z(opt)} \leq \frac{4C_{max}(\mathcal{J}^{(1)} \cup \mathcal{J}^{(2)}) + 3D + 2(T_{01} + T_{12} + T_{02})}{2(C_{max}^{(i_2)}(opt) + T_{01} + T_{12} + T_{02}) + D} \leq 2 + \frac{1}{2}. \tag{4.132}$$

Case 1.2.2: In opt , the jobs of Customer 1 and Customer 2 are not in the same batch.

Then, $Z(opt) = C_{max}^{(i_1)}(opt) + C_{max}^{(i_2)}(opt) + 2T_{01} + 2T_{02} + 2D$.

If $T_d^{(1)} = T_d^{(2)}$, $T_d^{(2)} \leq C_{max}(\mathcal{J}^{(1)})$.

$$\begin{aligned}
Z(\eta) &\leq 2(T_d^{(2)} + C_{max}(\mathcal{J}^{(1)} \cup \mathcal{J}^{(2)}) - \frac{l_1}{2}D) + \frac{l_1}{2}D + C_{max}(\mathcal{J}^{(1)}) + \\
&\quad (s+2)D + 2(T_{01} + T_{12} + T_{02}) \\
&\leq 3C_{max}(\mathcal{J}^{(1)}) + 2C_{max}(\mathcal{J}^{(1)} \cup \mathcal{J}^{(2)}) + 3D + 2(T_{01} + T_{12} + T_{02}).
\end{aligned} \tag{4.133}$$

$$\begin{aligned}
\frac{Z(\eta)}{Z(opt)} &\leq \frac{3C_{max}(\mathcal{J}^{(1)}) + 2C_{max}(\mathcal{J}^{(1)} \cup \mathcal{J}^{(2)}) + 3D + 2(T_{01} + T_{12} + T_{02})}{C_{max}^{(i_1)}(opt) + C_{max}^{(i_2)}(opt) + 2T_{01} + 2T_{02} + 2D} \\
&\leq 2 + \frac{1}{2}.
\end{aligned} \tag{4.134}$$

If $T_d^{(1)} < T_d^{(2)}$, there is $T_d^{(2)} > C_{max}^{(1)}(\eta)$ as there is no free machine time in the interval $(C_{max}(\mathcal{J}^{(1)}), C_{max}^{(1)}(\eta)]$. Meanwhile, $\rho_{max}^{(2)}(\eta) \leq (\lceil \frac{C_{max}^{(1)}(\eta)}{D} \rceil + 1)\frac{1}{2}D \leq C_{max}^{(1)}(\eta) + D \leq T_d^{(2)} + D$.

$$\begin{aligned}
Z(\eta) &\leq 2T_d^{(2)} + 2D + \frac{l_1}{2}D + C_{max}(\mathcal{J}^{(1)}) + (s+1)D + 2(T_{01} + T_{12} + T_{02}) \\
&\leq 2C_{max}(\mathcal{J}^{(1)}) + C_{max}(\mathcal{J}^{(1)} \cup \mathcal{J}^{(2)}) + 4D + 2(T_{01} + T_{12} + T_{02}).
\end{aligned} \tag{4.135}$$

$$\frac{Z(\eta)}{Z(opt)} \leq \frac{2C_{max}(\mathcal{J}^{(1)}) + C_{max}(\mathcal{J}^{(1)} \cup \mathcal{J}^{(2)}) + 4D + 2(T_{01} + T_{12} + T_{02})}{C_{max}^{(i_1)}(opt) + C_{max}^{(i_2)}(opt) + 2T_{01} + 2T_{02} + 2D} \leq 2. \tag{4.136}$$

Case 2: $C_{max}^{(1)}(\eta) > C_{max}^{(2)}(\eta)$. As there is no free machine time in $(C_{max}(\mathcal{J}^{(1)}), C_{max}^{(1)}(\eta)]$, there is $T_d^{(1)} = T_d^{(2)} \leq C_{max}(\mathcal{J}^{(1)})$ and $C_{max}^{(1)}(\eta) \leq T_d^{(1)} + C_{max}(\mathcal{J}^{(1)} \cup \mathcal{J}^{(2)}) - \frac{l_1}{2}D$.

Case 2.1: In η , the last batch is only for Customer 1. In this case, $\rho_{max}^{(2)}(\eta) \leq (\lceil \frac{C_{max}^{(2)}(\eta)}{\frac{D}{2}} \rceil + 1)\frac{D}{2}$ and $\rho_{max}^{(1)}(\eta) \leq (\lceil \frac{C_{max}^{(1)}(\eta)}{\frac{D}{2}} \rceil - 1)\frac{D}{2}$

In the interval $(0, C_{max}(\mathcal{J}^{(1)}])$, the delivery cost will not be more than $\frac{l_1}{2}D + \lfloor \frac{C_{max}(\mathcal{J}^{(1)})}{\frac{D}{2}} \rfloor \frac{D}{2}$.

In the interval $(C_{max}(\mathcal{J}^{(1)}), \rho_{max}^{(2)}(\eta)]$, there is one batch at $\rho_{max}^{(2)}(\eta)$ only for Customer 2, so the delivery cost will be D .

In the interval $(\rho_{max}^{(2)}(\eta), \rho_{max}^{(1)}(\eta)]$, there is only one batch at the time $\rho_{max}^{(1)}(\eta)$, so the delivery cost will be D .

$$TC(\eta) \leq \frac{l_1}{2}D + C_{max}(\mathcal{J}^{(1)}) + 2D. \quad (4.137)$$

Case 2.1.1: In opt , the jobs of Customer 1 and Customer 2 are in the same batch.

Then, $Z(opt) = 2(C_{max}^{(i_2)}(opt) + T_{01} + T_{12} + T_{02}) + D$.

$$\begin{aligned} Z(\eta) &\leq \rho_{max}^{(1)}(\eta) + 2T_{01} + \rho_{max}^{(2)}(\eta) + 2T_{02} + TC(\eta) \\ &\leq 2C_{max}^{(1)}(\eta) + D + \frac{l_1}{2}D + C_{max}(\mathcal{J}^{(1)}) + 2D + 2(T_{01} + T_{02}) \\ &\leq 2(T_d^{(1)} + C_{max}(\mathcal{J}^{(1)} \cup \mathcal{J}^{(2)}) - \frac{l_1}{2}D) + \frac{l_1}{2}D + C_{max}(\mathcal{J}^{(1)}) + \\ &\quad 3D + 2(T_{01} + T_{02}) \\ &\leq 4C_{max}(\mathcal{J}^{(1)} \cup \mathcal{J}^{(2)}) + 3D + 2(T_{01} + T_{02}) \end{aligned} \quad (4.138)$$

$$\frac{Z(\eta)}{Z(opt)} \leq \frac{4C_{max}(\mathcal{J}^{(1)} \cup \mathcal{J}^{(2)}) + 3D + 2(T_{01} + T_{02})}{2(C_{max}^{(i_2)}(opt) + T_{01} + T_{12} + T_{02}) + D} \leq 2 + \frac{1}{2} \quad (4.139)$$

Case 2.1.2: In opt , the jobs of Customer 1 and Customer 2 are not in the same batch.

Then, $Z(opt) = C_{max}^{(i_1)}(opt) + C_{max}^{(i_2)}(opt) + 2T_{01} + 2T_{02} + 2D$.

$$\begin{aligned}
Z(\eta) &\leq \rho_{max}^{(1)}(\eta) + 2T_{01} + \rho_{max}^{(2)}(\eta) + 2T_{02} + TC(\eta) \\
&\leq 2C_{max}^{(1)}(\eta) + D + \frac{l_1}{2}D + C_{max}(\mathcal{J}^{(1)}) + 2D + 2(T_{01} + T_{02}) \\
&\leq 2(T_d^{(1)} + C_{max}(\mathcal{J}^{(1)} \cup \mathcal{J}^{(2)}) - \frac{l_1}{2}D) + \frac{l_1}{2}D + C_{max}(\mathcal{J}^{(1)}) + \\
&\quad 3D + 2(T_{01} + T_{02}) \\
&\leq 3C_{max}(\mathcal{J}^{(1)}) + 2C_{max}(\mathcal{J}^{(1)} \cup \mathcal{J}^{(2)}) + 3D + 2(T_{01} + T_{02}).
\end{aligned} \tag{4.140}$$

$$\frac{Z(\eta)}{Z(opt)} \leq \frac{3C_{max}(\mathcal{J}^{(1)}) + 2C_{max}(\mathcal{J}^{(1)} \cup \mathcal{J}^{(2)}) + 3D + 2(T_{01} + T_{02})}{C_{max}^{(i_1)}(opt) + C_{max}^{(i_2)}(opt) + 2T_{01} + 2T_{02} + 2D} \leq 2 + \frac{1}{2}. \tag{4.141}$$

Case 2.2: In η , the last batch is for both Customer 1 and Customer 2. In this case,

$$\rho_{max}^{(1)}(\eta) = \rho_{max}^{(2)}(\eta) \leq \lceil \frac{C_{max}^{(1)}(\eta)}{\frac{D}{2}} \rceil \frac{D}{2} \leq C_{max}^{(1)}(\eta) + \frac{1}{2}D.$$

In the interval $(0, C_{max}(\mathcal{J}^{(1)})]$, the delivery cost will not be more than $\frac{l_1}{2}D + \lfloor \frac{C_{max}(\mathcal{J}^{(1)})}{\frac{D}{2}} \rfloor \frac{D}{2}$.

In the interval $(C_{max}(\mathcal{J}^{(1)}), \rho_{max}^{(1)}(\eta)]$, as all the jobs are released before $C_{max}(\mathcal{J}^{(1)})$, there is only one batch at $\rho_{max}^{(1)}(\eta)$, so the delivery cost will be D .

$$TC(\eta) \leq \frac{l_1}{2}D + C_{max}(\mathcal{J}^{(1)}) + D \tag{4.142}$$

Case 2.2.1: In opt , the jobs of Customer 1 and Customer 2 are in the same batch.

Then, $Z(opt) = 2(C_{max}^{(i_2)}(opt) + T_{01} + T_{12} + T_{02}) + D$.

$$\begin{aligned}
Z(\eta) &\leq \rho_{max}^{(1)}(\eta) + T_{01} + T_{12} + T_{02} + \rho_{max}^{(2)}(\eta) + T_{01} + T_{12} + T_{02} + TC(\eta) \\
&\leq 2C_{max}^{(1)}(\eta) + D + \frac{l_1}{2}D + C_{max}(\mathcal{J}^{(1)}) + D + 2(T_{01} + T_{12} + T_{02}) \\
&\leq 2(T_d^{(1)} + C_{max}(\mathcal{J}^{(1)} \cup \mathcal{J}^{(2)}) - \frac{l_1}{2}D) + \frac{l_1}{2}D + C_{max}(\mathcal{J}^{(1)}) + 2D + \\
&\quad 2(T_{01} + T_{12} + T_{02}) \\
&\leq 4C_{max}(\mathcal{J}^{(1)} \cup \mathcal{J}^{(2)}) + 2D + 2(T_{01} + T_{12} + T_{02}).
\end{aligned} \tag{4.143}$$

$$\frac{Z(\eta)}{Z(opt)} \leq \frac{4C_{max}(\mathcal{J}^{(1)} \cup \mathcal{J}^{(2)}) + 2D + 2(T_{01} + T_{02})}{2(C_{max}^{(i_2)}(opt) + T_{01} + T_{12} + T_{02}) + D} \leq 2. \quad (4.144)$$

Case 2.2.2: In opt , the jobs of Customer 1 and Customer 2 are not in the same batch.

Then, $Z(opt) = C_{max}^{(i_1)}(opt) + C_{max}^{(i_2)}(opt) + 2T_{01} + 2T_{02} + 2D$.

$$\begin{aligned} Z(\eta) &\leq \rho_{max}^{(1)}(\eta) + T_{01} + T_{12} + T_{02} + \rho_{max}^{(2)}(\eta) + T_{01} + T_{12} + T_{02} + TC(\eta) \\ &\leq 2C_{max}^{(1)}(\eta) + D + \frac{l_1}{2}D + C_{max}(\mathcal{J}^{(1)}) + D + 2(T_{01} + T_{12} + T_{02}) \\ &\leq 2(T_d^{(1)} + C_{max}(\mathcal{J}^{(1)} \cup \mathcal{J}^{(2)}) - \frac{l_1}{2}D) + \frac{l_1}{2}D + C_{max}(\mathcal{J}^{(1)}) + \\ &\quad 2D + 2(T_{01} + T_{12} + T_{02}) \\ &\leq 3C_{max}(\mathcal{J}^{(1)}) + 2C_{max}(\mathcal{J}^{(1)} \cup \mathcal{J}^{(2)}) + 2D + 2(T_{01} + T_{12} + T_{02}). \end{aligned} \quad (4.145)$$

$$\frac{Z(\eta)}{Z(opt)} \leq \frac{3C_{max}(\mathcal{J}^{(1)}) + 2C_{max}(\mathcal{J}^{(1)} \cup \mathcal{J}^{(2)}) + 2D + 2(T_{01} + T_{12} + T_{02})}{C_{max}^{(i_1)}(opt) + C_{max}^{(i_2)}(opt) + 2T_{01} + 2T_{02} + 2D} \leq 2. \quad (4.146)$$

□

4.11 Simulated Experiment for SMH with Routing

In this section, a simulation is conducted to demonstrate the run-time and the performance of the above algorithms (SMH5-SMH8) in normal scenarios and illustrate how the algorithms are used in practice. An instance can be defined by prescribing a set of the foregoing parameters (n_i , $p_j^{(i)}$, and $r_j^{(i)}$, for $j = 1, 2, \dots, n_i$, T_{il} for $i, l = 1, 2, \dots, k$, C and D). The instances were generated by these randomly generated parameters. The algorithm was implemented in the Matlab environment. The parameters are thus determined based on the following assumptions:

- (1) The release of jobs for Customer i follows the poisson distribution with the parameter λ_i , i.e., the number of jobs released at some time r : $n_i(r) \sim P(\lambda_i)$ and the next

release time is $r + r'$, where $r' \sim U(0, \lambda_i)$, λ_i is two times of the mean value of the release intervals for Customer i , and $\lambda_i \sim U(0, \Lambda_i)$ ($i = 1, 2, \dots, k$).

- (2) The job processing time for Customer i follows the uniform distribution in the interval $[0, b_i]$, i.e., $p_j^{(i)} \sim U(0, b_i)$ for $j = 1, 2, \dots, n_i$, where b_i is two times of the mean value of the processing time for Customer i and $b_i \sim U(0, B_i)$ ($i = 1, 2, \dots, k$).
- (3) The number of jobs for Customer i follows the uniform distribution in the set $\{1, 2, \dots, N_i\}$, i.e., $Pr\{n_i = h\} = \frac{1}{N_i}$ for $h = 1, 2, \dots, N_i$ where N_i is two times of the mean value of the number of jobs for Customer i ($i = 1, 2, \dots, k$).
- (4) The positions of the manufacturer and the customers are randomly located in an square area with side length L , and the transportation network can be directly determined by the Euclidean distance.
- (5) The delivery cost D is a constant.
- (6) The number of customers is of four cases: $k = 2$, $k = 5$, $k = 10$, and $k = 20$.

By choosing different values for Λ_i , B_i , N_i , and L , instances are generated and scheduling is then executed. First, the performance of the developed GAs for off-line problems with the analytical property in comparison with that of original GAs without the analytical property is demonstrated. SMH7 and the corresponding original GA are taken as an example as SMH7 is similar with but more complex than SMH5. In the numerical simulation, the two algorithms have the same initialized population, the same size of population, and the same number of generations. The probabilities of crossover and mutation are chosen as 0.75 and 0.2, respectively. Therefore, the differences in the results are attributed to

the analytical property. To illustrate the results clearly, the discussion will be under the following three cases.

Case 1. The values of processing part ($r_j^{(i)}$ and $p_j^{(i)}$) are small while the values of transportation system (T_{il}) and the unit delivery cost (D) are large. In this case, the delivery part overwhelms the processing part, so it highlights the role of the analytical property. Figures 4.10-4.12 show the results of SMH7 and original GA in this case for $k = 5$, $k = 10$ and $k = 20$, respectively. From the three figures, it can be seen that the convergence values of SMH7 are much greater than those of original GA, and the convergence speeds are almost the same, which implies that SMH7 can achieve a much better result than original GA in the same running time.

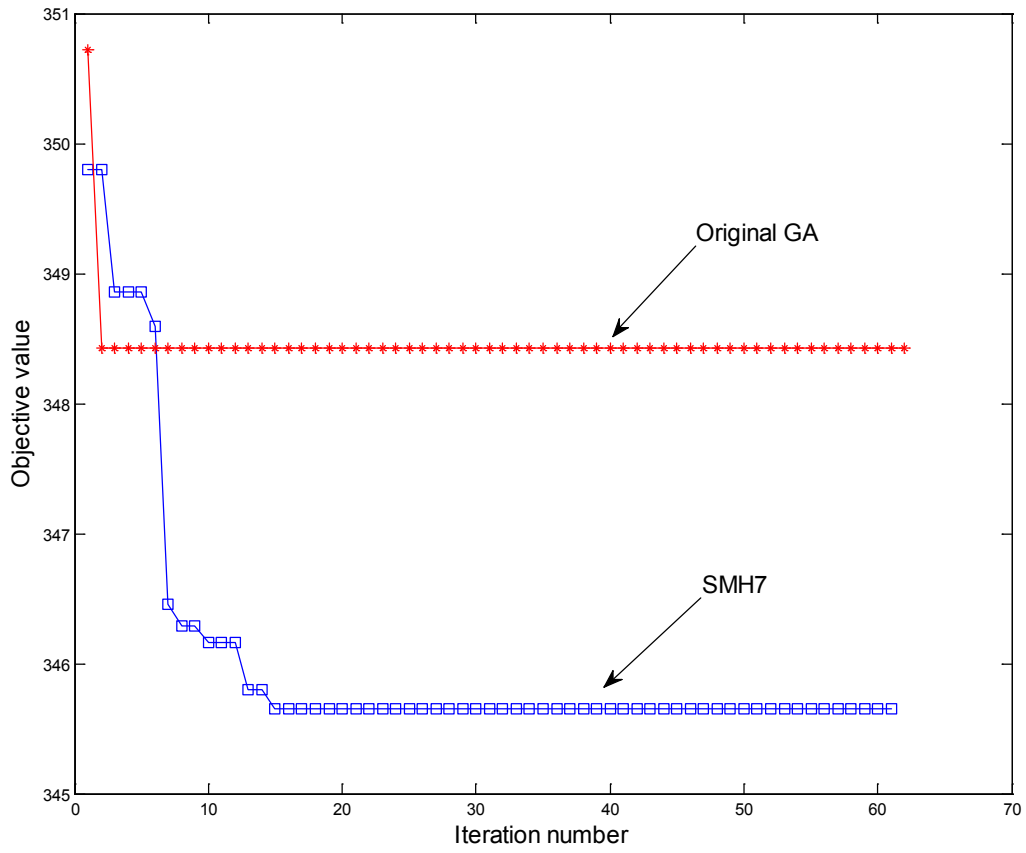


Figure 4.10. The Results of SMH7 and Original GA for Case 1 with $k = 5$

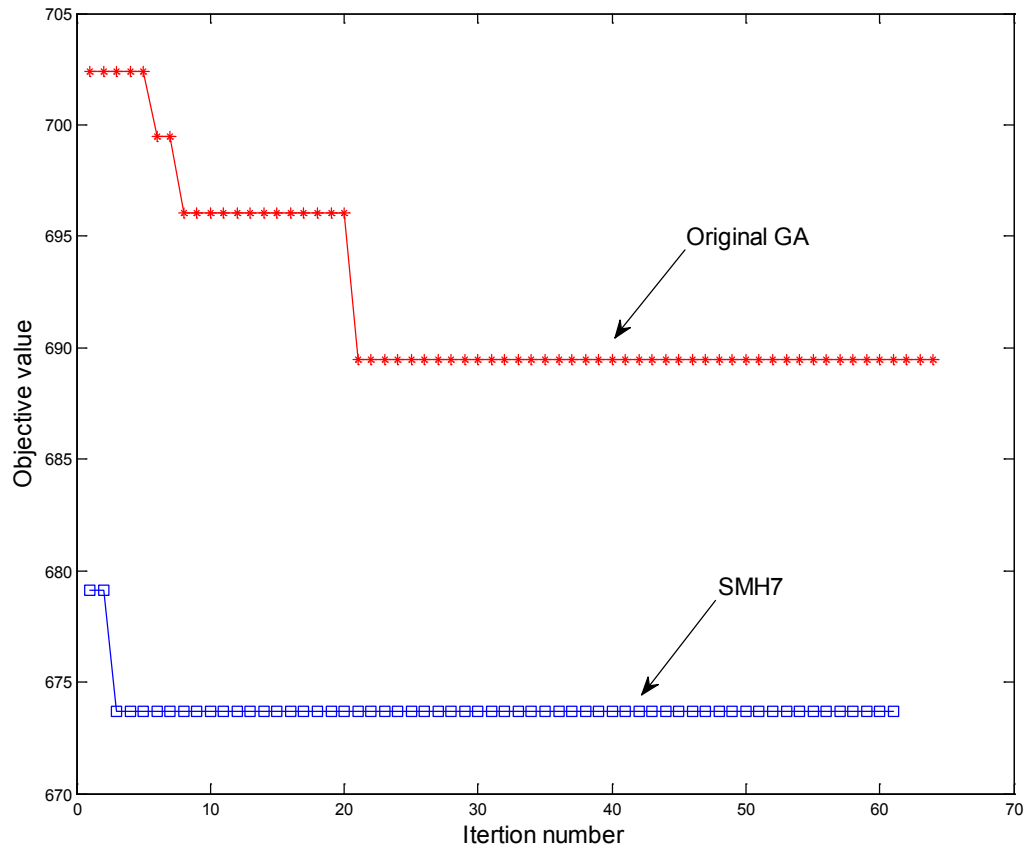


Figure 4.11. The Results of SMH7 and Original GA for Case 1 with $k = 10$

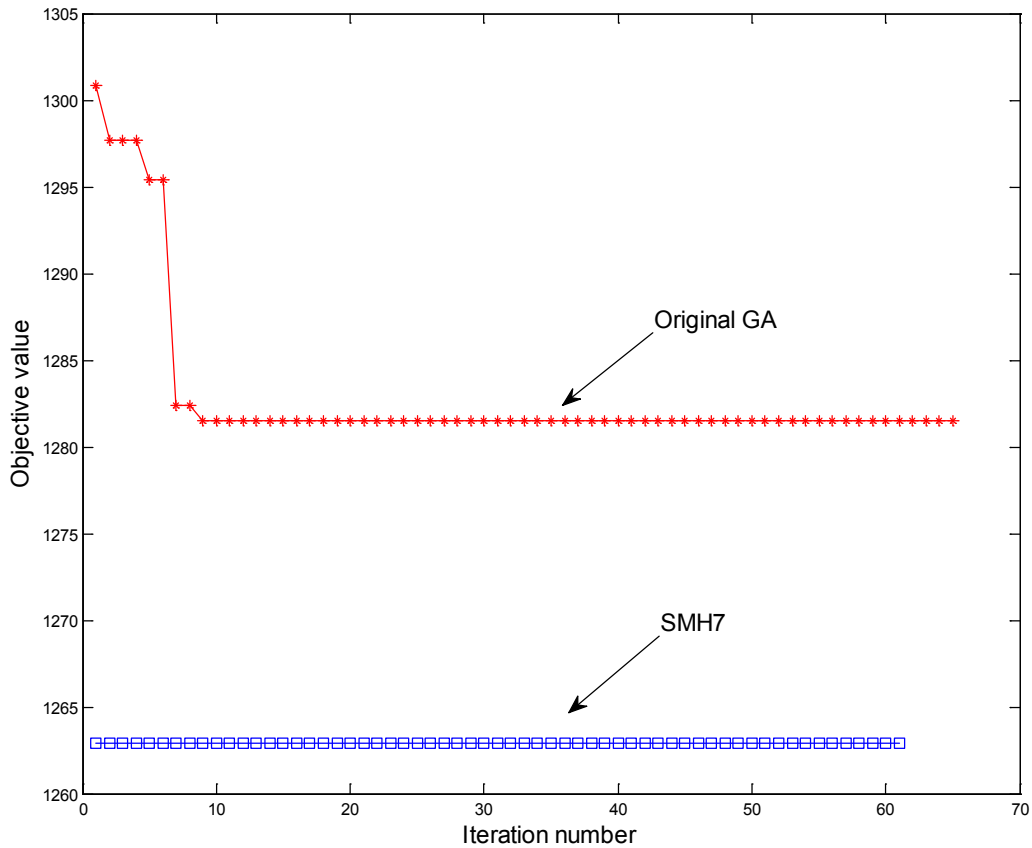


Figure 4.12. The Results of SMH7 and Original GA for Case 1 with $k = 20$

Case 2. The values of processing part ($r_j^{(i)}$ and $p_j^{(i)}$) and the unit delivery cost (D) are small while the values of transportation system (T_{il}) are large. As the delivery cost does not take a major portion, scheduling with a large scale routing should be abandoned and jobs of most customers should be delivered directly to their destinations. Only the customers, who are close to each other, may share a batch. This is consistent with the analytical property that every SK_q ($q = 1, 2, \dots, o$) contains few customers when D is small. Figures 4.16-4.18 show the results of SMH7 and original GA in this case for $k = 5$, $k = 10$ and $k = 20$, respectively. It can tell from the three figures that SMH7 can have

better initial values and converge to the final value more quickly. Therefore, SMH7 can perform greater than original GA for this case.

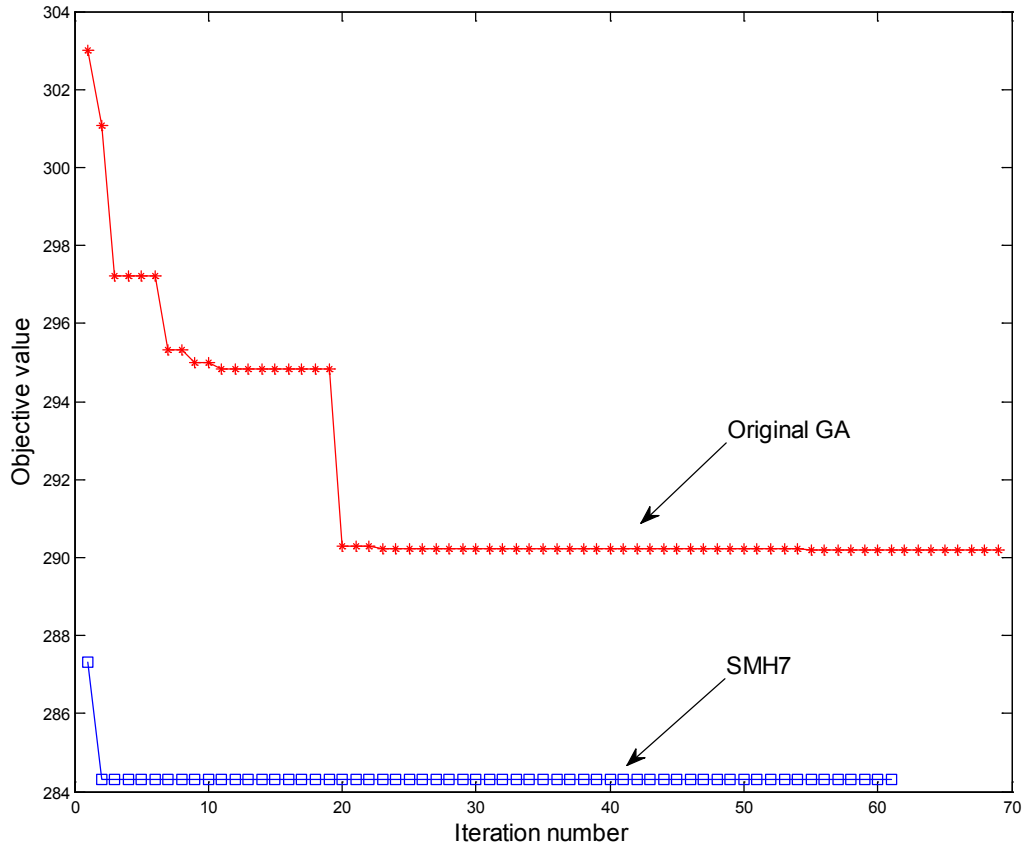


Figure 4.13. The Results of SMH7 and Original GA for Case 2 with $k = 5$

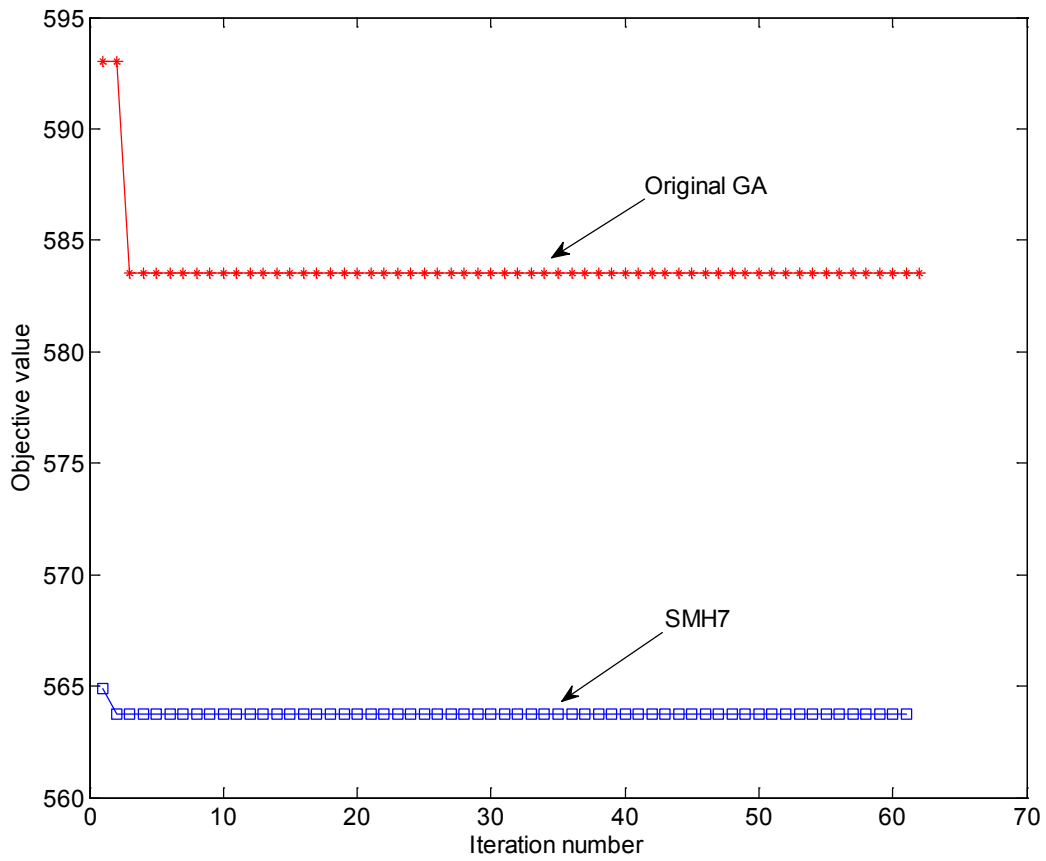


Figure 4.14. The Results of SMH7 and Original GA for Case 2 with $k = 10$

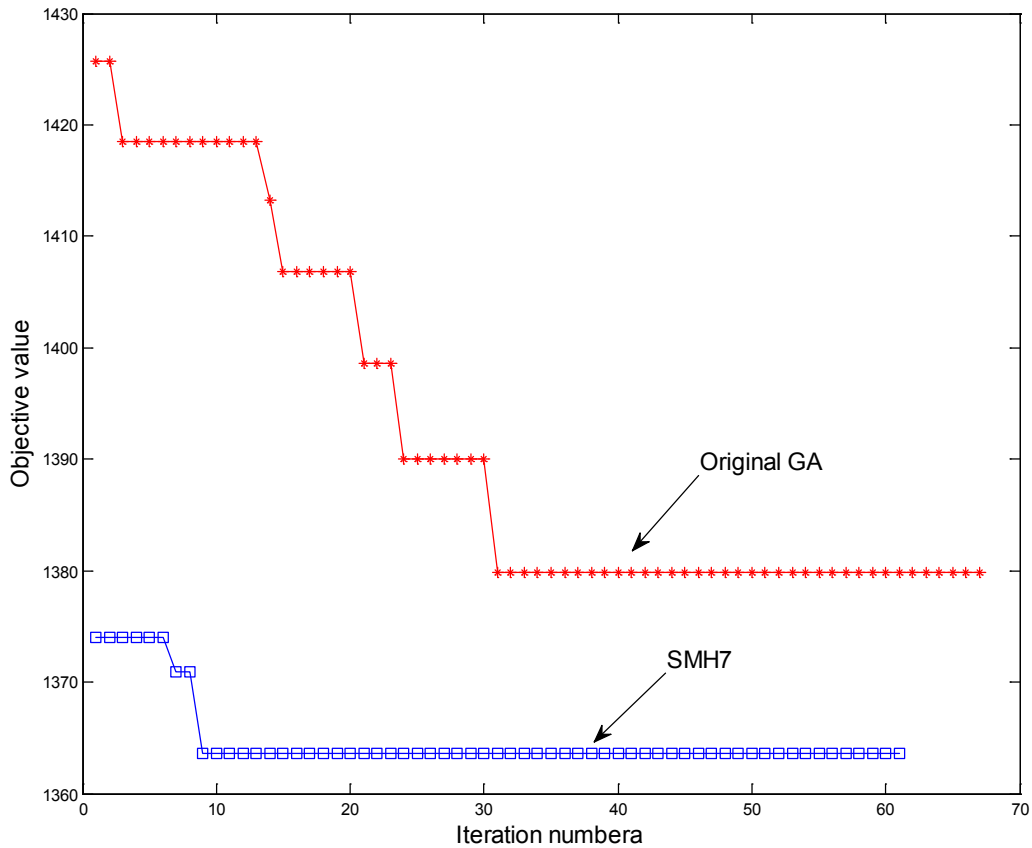


Figure 4.15. The Results of SMH7 and Original GA for Case 2 with $k = 20$

Case 3. The values of processing part ($r_j^{(i)}$ and $p_j^{(i)}$), the unit delivery cost (D), and the values of transportation system (T_{il}) are almost the same. In this case, the processing part and the delivery part account for the similar proportion in the total objective value. Figures show the results of SMH7 and original GA in this case for $k = 5$, $k = 10$ and $k = 20$, respectively. From the three figures, the algorithm SMH7 achieves better results than original GA in the same iterations. Especially, for larger k ($k = 10$ and $k = 20$), the performance of SMH7 significantly exceeds that of original GA.

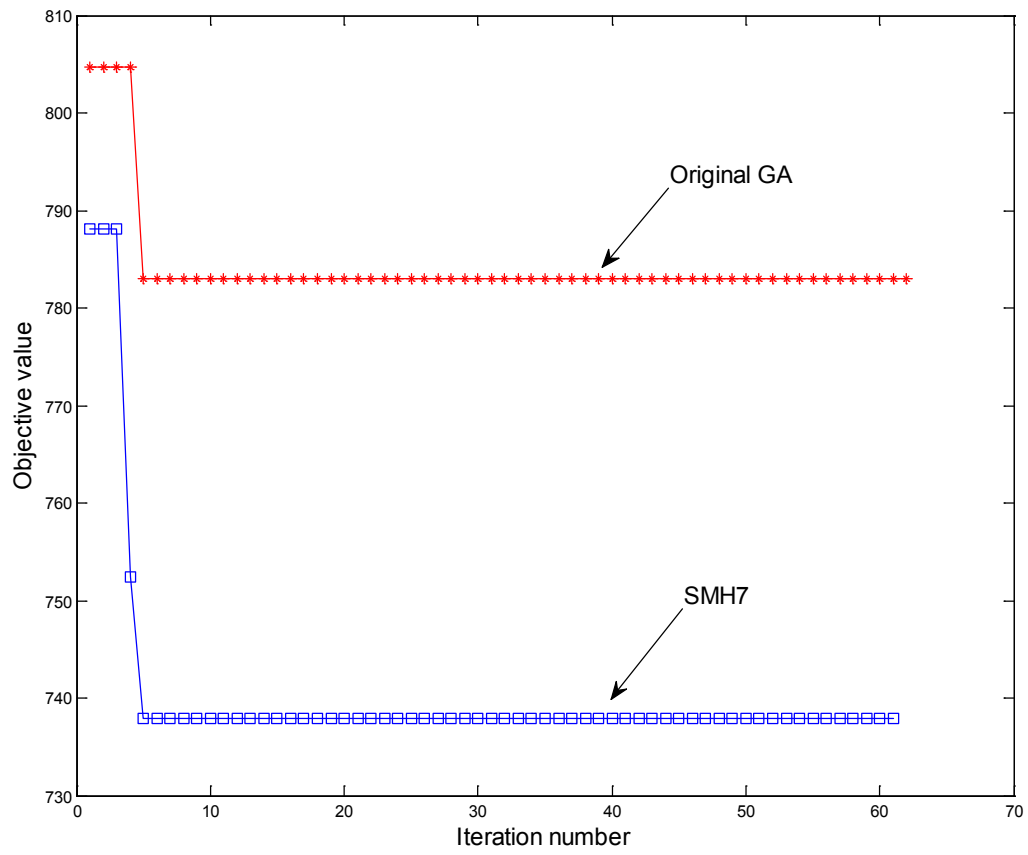


Figure 4.16. The Results of SMH7 and Original GA for Case 3 with $k = 5$

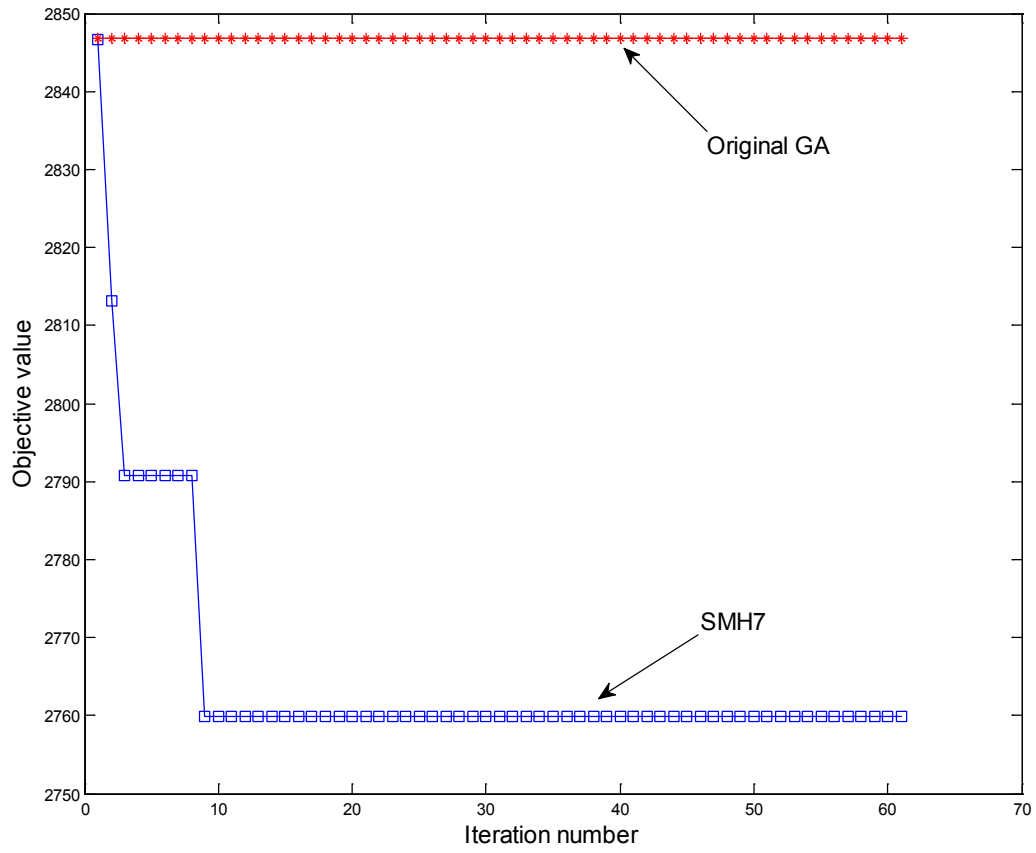


Figure 4.17. The Results of SMH7 and Original GA for Case 3 with $k = 10$

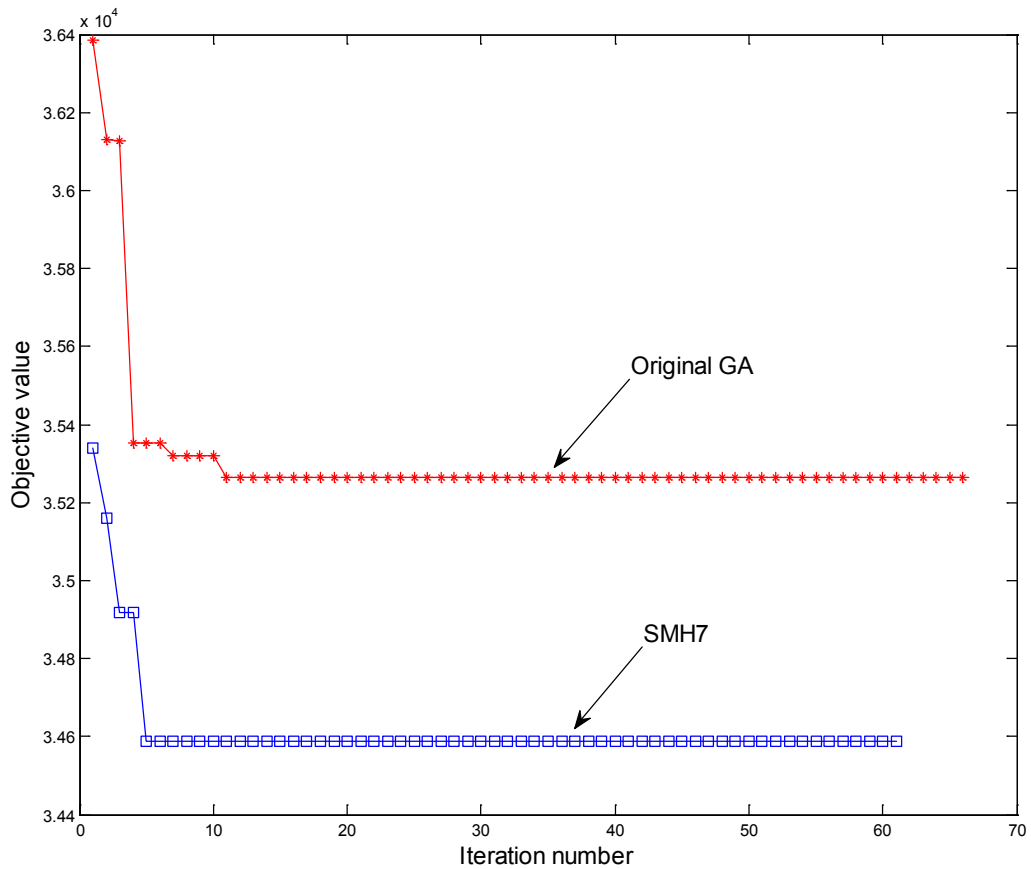


Figure 4.18. The Results of SMH7 and Original GA for Case 3 with $k = 20$

From the above discussions, one can conclude that SMH7 performs much greater than original GA for all different cases (notice that there is no consideration of the case that the value of unit cost is large but the values of processing part and transportation system is small, as the problem will degenerate to the classical scheduling problem). The reason is that the analytical property has exclude solutions which are not good enough. The solution domain will be reduced to be a smaller one, so the algorithm can perform more efficiently. As mentioned in Chapter 2, although GA is global optimal in theory, it cannot achieve the global optimal solution in application. However, the developed GAs with an-

alytical property (SMH5 and SMH7) can outperform the original ones, which implies that much greater local solutions can be found by these algorithms. In the following, these local solutions will be assumed as the global optimal ones for the evaluation of on-line algorithms SMH6 and SMH8.

The above three cases of parameter settings are still considered. In all cases, 100 instances are generated. Table 4.9 shows the result for different values of k and three different parameter setting cases. Each row in the table is the average of the results of the 100 instances. The columns in the table are (1) the ratio of the algorithm value to the benchmark value, (2) the run-time in seconds, respectively. Notice that the result of SMH5 is applied as the benchmark value. As SMH5 solves the problem SMP5 which is the lower bound for off-line version of the problem SMP8, the ratio of the results of SMH8 to the optimal result (the result of SMH5) is better.

Table 4.9. Results of Algorithms SMH5, SMH6 and SMH8

k	Case	SMH5		SMH6		SMH8	
		ratio	time	ratio	time	ratio	time
2	Case 1	1	0.0183	1.34	0.0068	1.34	0.0081
	Case 2	1	0.0165	1.34	0.0043	1.34	0.0052
	Case 3	1	0.0167	1.12	0.0012	1.12	0.0016
5	Case 1	1	0.348	1.33	0.023	1.33	0.025
	Case 2	1	0.379	1.36	0.023	1.36	0.025
	Case 3	1	0.345	1.12	0.0051	1.13	0.0051
10	Case 1	1	5.45	1.30	0.086	1.30	0.088
	Case 2	1	5.44	1.36	0.073	1.36	0.076
	Case 3	1	5.31	1.21	0.017	1.21	0.012
20	Case 1	1	145.356	1.24	0.44	1.30	0.44
	Case 2	1	112.367	1.36	0.28	1.36	0.27
	Case 3	1	115.804	1.33	0.063	1.33	0.032

From Table 4.9, it is evident that the ratio columns of SMH6 and SMH8 for $k = 2$ never exceed 1.37, which exhibits the robustness of the two algorithms and is also consistent with the results of Theorem 4.10 and Theorem 4.11. Actually, the ratios of the two algorithms perform much better than the theoretical result, 2 and $2 + \frac{1}{2}$, respectively, which shows the excellent performance on the normal instances. For the cases that the value of k is greater than 2, the algorithms SMH6 and SMH8 still performs very well on the normal

instances. The run-time for SMH6 and SMH8 never exceeds 0.5 seconds for $k = 20$ and is much shorter for smaller k , so the efficiency of the algorithms is very high. Furthermore, by examining the run-time results of SMH6 and SMH8 for the value of k from low to high, the algorithms tend to have a polynomial time complexity, that is, as k increases, the run-time of the algorithms increases as a polynomial of k . This means that the two algorithms can well be scaled to a much larger problem. Meanwhile, for algorithm SMH5, the run-time increases much more rapidly. The reason is that although GA may have a polynomial complexity, the power will be very high. However, the run-time of SMH5 is short for small value of k ($k=2,5$, and 10) still acceptable for large value of k ($k=20$).

4.12 Concluding Remarks

In this chapter, eight problems (denoted by SMP1-SMP8) for single-machine and multi-customers were proposed. These problems were of different release environment, processing patterns and delivery patterns. Corresponding algorithms were developed for them. A simulation study was conducted for all the algorithms. These algorithms are robust and efficient in terms of the approximate ratio and the competitive ratio analysis. For SMP1, the exact optimal algorithm (SMH1) can be applied for small k case while the simulated annealing algorithm (SA_SMH1) can deal with large k case. For SMP2, the on-line algorithm (SMH2) has competitive ratio 2 for $k = 2$ (on-line optimal) and $2 + \frac{2}{27}$ for $k = 3$. For SMP3, approximate algorithms (SMH3 and K2SMH3) for both general case and special case ($k = 2$) are provided. For SMP4, the on-line algorithm (SMH4) is $2 + \frac{1}{2}$ -competitive for $k = 2$ case. For SMP5 and SMP7, the GAs with analytical property (SMH5 and SMH7) are provided. For SMP6, the on-line algorithm (SMH6) is optimal

for $k = 2$ case. For SMP8, the on-line algorithm (SMH8) is $2 + \frac{1}{2}$ -competitive for $k = 2$ case. In addition, for the cases without routing, two on-line problems with limited vehicle capacity (CSMP2 and CSMP4) are considered and the on-line algorithms (CSMH2 and CSMH4) are 2-competitive and $2 + \frac{1}{2}$ -competitive, respectively. From the simulation study, all algorithms perform robustness for worst instances and great for most normal instances, and possess efficiency even for different values of k .

CHAPTER 5

ALGORITHMS FOR MULTI-MACHINES AND SINGLE-CUSTOMER PROBLEMS

In this chapter, the problems for multi-machines and single-customer are considered and corresponding algorithms are developed. As described in Section 3.3, five on-line problems are defined in terms of different processing patterns, vehicles characteristics and delivery patterns. For all these problems, the corresponding algorithms are developed and the theoretical analysis is proposed. The simulation experiment for one algorithm is presented.

As there is only one customer, the notations $J_j^{(1)}$, $r_j^{(1)}$, $p_j^{(1)}$, $\mathcal{J}_j^{(1)}$, n_1 , $C_j^{(1)}$, $C_{max}^{(1)}$, $\rho_j^{(1)}$, $\rho_{max}^{(1)}$, $D_j^{(1)}$, $D_{max}^{(1)}$ and $2T_{01}$ are replaced by J_j , r_j , p_j , \mathcal{J}_j , n , C_j , C_{max} , ρ_j , ρ_{max} , D_j , D_{max} and T for short (T is the round-trip transportation time between the medical center and the customer).

5.1 Algorithm for Problem MSP1

MSP1 has the following features: Jobs are released on-line and delivered in "direct" pattern. The capacity of vehicles is enough but the number of vehicles is one. The lower bound of MSP1 can be derived from single-machine case [Han, 2012].

Corollary 5.1. *No on-line algorithm for MSP1 can have a competitive ratio less than $\max\{1 + \theta, 1 + \sqrt{\frac{D}{T+D}}\}$, even if all processing times are 0.*

Algorithm MSH1

Jobs are scheduled on the machines with the LPT-rule.

When $T > (1 + \theta)D$, after time $\theta(T + D)$, if there is no uncompleted job and the vehicle is available, then there is a batch to deliver all the completed jobs.

When $T \leq (1 + \theta)D$, at time $l\sqrt{D(T + D)}$, where $l \geq 1$ is an integer, if there is no uncompleted job, then there is a batch to deliver all the jobs.

The on-line algorithm MSH1 for MSP1 is on-line optimal, which is analyzed as follows.

Theorem 5.1. *The competitive ratio for the on-line algorithm MSH1 for MSP1 is $\max\{1 + \theta, 1 + \sqrt{\frac{D}{T+D}}\}$, which is on-line optimal.*

Proof of Theorem 5.1: Let η be the schedule obtained by MSH1. As the algorithm has two different cases, the proof also has two parts for the two cases, respectively.

Case 1: $T > (1 + \theta)D$.

Case 1.1: there is only one batch in η . $Z(\text{opt}) = C_{\max}(\text{opt}) + T + D$ and $Z(\eta) = \max\{\theta(T + D), C_{\max}(\eta)\} + T + D$. As $C_{\max}(\eta) \leq \frac{3}{2}C_{\max}(\text{opt})$, $\frac{Z(\eta)}{Z(\text{opt})} \leq 1 + \theta$.

Case 1.2: there is more than one batch in η . The information of all the jobs is known after $C_{\max}(\text{opt})$, so there is only one batch after $C_{\max}(\text{opt})$. Note that there is at least one batch before $C_{\max}(\text{opt})$. Suppose that there are $h+1$ batches before $C_{\max}(\text{opt})$ and the last delivery time is τ . Then $\tau \geq \theta(T + D) + hT$ and all the jobs completed after τ are released after τ . $D_{\max}(\eta) \leq \max\{C_{\max}(\eta), \tau + T\} + T \leq C_{\max}(\eta) + 2T$ and $TC(\eta) = (h + 2)D$. Furthermore, $C_{\max}(\eta) - \tau \leq \frac{3}{2}(C_{\max}(\text{opt}) - \tau)$ [Chen and Vestjens, 1997].

$$\begin{aligned}
\frac{Z(\eta)}{Z(\text{opt})} &\leq \frac{C_{\max}(\eta) + 2T + (h+2)D}{C_{\max}(\text{opt}) + T + D} \\
&\leq \frac{\frac{3}{2}C_{\max}(\text{opt}) - \frac{1}{2}\tau + 2T + (h+2)D}{C_{\max}(\text{opt}) + T + D} \\
&\leq 1 + \frac{\frac{1}{2}C_{\max}(\text{opt}) - \frac{1}{2}\tau + T + (h+1)D}{C_{\max}(\text{opt}) + T + D}
\end{aligned} \tag{5.1}$$

If $\frac{-\frac{1}{2}\tau + T + (h+1)D}{T+D} < \frac{1}{2}$, $\frac{Z(\eta)}{Z(\text{opt})} \leq 1 + \frac{1}{2}$; else,

$$\begin{aligned}
\frac{Z(\eta)}{Z(\text{opt})} &\leq 1 + \frac{\frac{1}{2}\tau - \frac{1}{2}\tau + T + (h+1)D}{\tau + T + D} \\
&\leq 1 + \frac{T + (h+1)D}{\tau + T + D} \\
&\leq 1 + \frac{T + D + hD}{(1+\theta)(T+D) + hT} \leq 1 + \theta.
\end{aligned} \tag{5.2}$$

Case 2: $T \leq (1+\theta)D$. Suppose that there are h batches before $C_{\max}(\text{opt})$ and the last delivery time is τ . Then $\tau \geq h\sqrt{D(T+D)}$ and all the jobs completed after τ are released after τ . $D_{\max}(\eta) \leq C_{\max}(\eta) + \sqrt{D(T+D)} + T$ and $TC(\eta) = (h+1)D$.

$$\begin{aligned}
\frac{Z(\eta)}{Z(\text{opt})} &\leq \frac{C_{\max}(\eta) + \sqrt{D(T+D)} + T + (h+1)D}{C_{\max}(\text{opt}) + T + D} \\
&\leq \frac{\frac{3}{2}C_{\max}(\text{opt}) - \frac{1}{2}\tau + \sqrt{D(T+D)} + T + (h+1)D}{C_{\max}(\text{opt}) + T + D} \\
&\leq 1 + \frac{\frac{1}{2}C_{\max}(\text{opt}) - \frac{1}{2}\tau + \sqrt{D(T+D)} + hD}{C_{\max}(\text{opt}) + T + D}.
\end{aligned} \tag{5.3}$$

If $\frac{-\frac{1}{2}\tau + \sqrt{D(T+D)} + hD}{T+D} < \frac{1}{2}$, $\frac{Z(\eta)}{Z(\text{opt})} \leq 1 + \frac{1}{2}$; else,

$$\begin{aligned}
\frac{Z(\eta)}{Z(\text{opt})} &\leq 1 + \frac{\frac{1}{2}\tau - \frac{1}{2}\tau + \sqrt{D(T+D)} + hD}{\tau + T + D} \\
&\leq 1 + \frac{\sqrt{D(T+D)} + hD}{\tau + T + D} \\
&\leq 1 + \frac{hD + \sqrt{D(T+D)}}{h\sqrt{D(T+D)} + T + D} \leq 1 + \sqrt{\frac{D}{T+D}}.
\end{aligned} \tag{5.4}$$

According to Corollary 5.1, the competitive ratio of MSH1 can not be less than $\max\{1 + \theta, 1 + \sqrt{\frac{D}{T+D}}\}$, which completes the proof. \square

5.2 Algorithm for Problem MSP2

MSP2 has the following features: Jobs are released on-line and delivered in "direct" pattern. The capacity of vehicles and the number of vehicles are both enough. The lower bound of MSP2 can be derived from single-machine case [Han, 2012].

Corollary 5.2. *No on-line algorithm for MSP2 can have a competitive ratio less than 2, even if all processing times are 0.*

Algorithm MSH2

Jobs are scheduled on the machines with the LPT rule. At time lD , where $l \geq 1$ is an integer, if there is no uncompleted job, then there is a batch to deliver all the jobs .

The on-line algorithm MSH2 for MSP2 is on-line optimal, which is analyzed as follows.

Theorem 5.2. *The competitive ratio for the on-line algorithm MSH2 for MSP2 is 2, which is on-line optimal.*

Proof of Theorem 5.2: Let η be the schedule obtained by MSH2. Suppose that there are h batches before $C_{max}(opt)$ and the last delivery time is τ . Then $\tau \geq hD$ and all the jobs completed after τ are released after τ . $D_{max}(\eta) \leq C_{max}(\eta) + D + T$ and $TC(\eta) = (h+1)D$.

$$\begin{aligned}
 \frac{Z(\eta)}{Z(opt)} &\leq \frac{C_{max}(\eta) + D + T + (h+1)D}{C_{max}(opt) + T + D} \\
 &\leq \frac{\frac{3}{2}C_{max}(opt) - \frac{1}{2}\tau + D + T + (h+1)D}{C_{max}(opt) + T + D} \\
 &\leq 1 + \frac{\frac{1}{2}C_{max}(opt) - \frac{1}{2}\tau + D + hD}{C_{max}(opt) + T + D}.
 \end{aligned} \tag{5.5}$$

If $\frac{-\frac{1}{2}\tau+D+hD}{T+D} < \frac{1}{2}$, $\frac{Z(\eta)}{Z(opt)} \leq 1 + \frac{1}{2}$; else,

$$\begin{aligned} \frac{Z(\eta)}{Z(opt)} &\leq 1 + \frac{\frac{1}{2}\tau - \frac{1}{2}\tau + D + hD}{\tau + T + D} \\ &\leq 1 + \frac{D + hD}{\tau + T + D} \\ &\leq 1 + \frac{hD + D}{hD + T + D} \leq 2. \end{aligned} \tag{5.6}$$

According to Corollary 5.2, the competitive ratio of MSH2 can not be less than 2, which completes the proof. \square

5.3 Algorithm for Problem MSP3

MSP3 has the following features: Jobs are released on-line, processed in "pmtn" pattern and delivered in "direct" pattern. The capacity of vehicles is C and the number of vehicles is one. The lower bound of MSP3 can be derived from single-machine case [Han, 2012].

Corollary 5.3. *No on-line algorithm for MSP3 can have a competitive ratio less than*

$$\max\{1 + \theta, 1 + \sqrt{\frac{D}{T+D}} - \frac{\sqrt{D(T+D)}}{(C-1)\sqrt{D(T+D)+T+D}}\}, \text{ even if all processing times are 0.}$$

Algorithm MSH3

When a new job is released, the McNaughton's algorithm is applied to all the uncompleted jobs.

When $T > (1 + \theta)D$, at time lT , where $l \geq 1$ is an integer, then there is a batch to deliver as many completed jobs as possible.

When $T \leq (1 + \theta)D$, at time $l\sqrt{D(T+D)}$, where $l \geq 1$ is an integer, then there is a batch to deliver as many completed jobs as possible.

The on-line algorithm MSH3 for MSP3 has competitive ratio 2, which is analyzed as follows.

Theorem 5.3. *The competitive ratio for the on-line algorithm MSH3 for MSP3 is 2.*

Proof of Theorem 5.3: Let η be the schedule obtained by algorithm MSH3. The processing part can minimize the completion time which implies $C_{max}(\eta) = C_{max}(opt)$. The proof has two parts for the two different cases of the algorithm.

Case 1: $T > (1 + \theta)D$. Suppose τ is the last delivery time before $\rho_{max}(\eta)$ when there is an unfull batch. Note that if there is no such τ , let $\tau = 0$. In this case, all the jobs completed after τ are released after τ , which means $\tau < C_{max}(opt)$. Meanwhile, for every delivery time between τ and $\rho_{max}(\eta)$, there is either no batch or there is a full batch. Let h be the number of full batches. At last, there will be a possible unfull batch at $\rho_{max}(\eta)$. Therefore, there will be more than hC jobs released after τ . In opt , there are at least $h + 1$ batches after τ , that is, $D_{max}(opt) \geq \max\{\tau + (h + 1)T, C_{max}(opt) + T\}$ and $TC(opt) \geq (h + 1)D$. For η , the worst case is that all these jobs are delivered after $C_{max}(\eta)$: $D_{max}(\eta) \leq \max\{\tau + T, C_{max}(\eta)\} + (h + 1)T \leq C_{max}(\eta) + (h + 2)T$ and $TC(\eta) \leq \frac{\tau}{T}D + (h + 1)D$.

$$\begin{aligned} \frac{Z(\eta)}{Z(opt)} &\leq \frac{C_{max}(\eta) + (h + 2)T + \frac{\tau}{T}D + (h + 1)D}{\max\{\tau + (h + 1)T, C_{max}(opt) + T\} + (h + 1)D} \\ &\leq 1 + \frac{(h + 1)T + \frac{\tau}{T}D}{\max\{\tau + (h + 1)T, C_{max}(opt) + T\} + (h + 1)D} \leq 2. \end{aligned} \quad (5.7)$$

Case 2: $T \leq (1 + \theta)D$. The same τ and h with Case 1 are defined. As such, there is $D_{max}(opt) \geq \max\{\tau + (h + 1)T, C_{max}(opt) + T\}$, $TC(opt) \geq (h + 1)D$, $D_{max}(\eta) \leq C_{max}(\eta) + (h + 1)\sqrt{D(T + D)} + T$ and $TC(\eta) \leq \frac{\tau}{\sqrt{D(T + D)}}D + (h + 1)D$.

$$\begin{aligned} \frac{Z(\eta)}{Z(opt)} &\leq \frac{C_{max}(\eta) + (h + 1)\sqrt{D(T + D)} + T + \frac{\tau}{\sqrt{D(T + D)}}D + (h + 1)D}{\max\{\tau + (h + 1)T, C_{max}(opt) + T\} + (h + 1)D} \\ &\leq 1 + \frac{(h + 1)\sqrt{D(T + D)} + \frac{\tau}{\sqrt{D(T + D)}}D}{\max\{\tau + (h + 1)T, C_{max}(opt) + T\} + (h + 1)D} \leq 1 + \sqrt{\frac{D}{T + D}}. \end{aligned} \quad (5.8)$$

At last, it needs to show the ratio of the algorithm can achieve 2 for the following instance. At time 0, there is a job with NT preparation time released, and at time ϵ there are NC jobs with 0 processing time released, where N is a sufficient large integer and ϵ is a very small number. $Z(\eta) = (2N + 1)T + (N + 1)D$ while $Z(opt) = \epsilon + (N + 1)T + (N + 1)D$. As ϵ and D gets infinitely small but N gets infinitely large, the ratio $\frac{Z(\eta)}{Z(opt)}$ will tend to 2. \square

5.4 Algorithm for Problem MSP4

MSP4 has the following features: Jobs are released on-line and delivered in "direct" pattern. The capacity of vehicles is C and the number of vehicles is one. The same lower bound of MSP3 can be applied directly.

Corollary 5.4. *No on-line algorithm for MSP4 can have a competitive ratio less than $\max\{1 + \theta, 1 + \sqrt{\frac{D}{T+D}} - \frac{\sqrt{D(T+D)}}{(C-1)\sqrt{D(T+D)+T+D}}\}$, even if all processing times are 0.*

Algorithm MSH4

Jobs are scheduled on the machines with the LPT-rule.

When $T > (1 + \theta)D$, at time lT , where $l > 1$ is an integer, if the number of completed jobs is not less than C or there is no uncompleted job, then there is a batch to deliver as many completed jobs as possible.

When $T \leq (1 + \theta)D$, at the time of $l\sqrt{D(T + D)}$, where $l > 1$ is an integer, if the number of completed jobs is not less than C or there is no uncompleted job, then there is a batch to deliver as many completed jobs as possible.

The on-line algorithm MSH4 for MSP4 $\max\{\frac{3}{2} + \theta, \frac{3}{2} + \sqrt{\frac{B}{T+B}}\}$ -competitive, which is

analyzed as follows.

Theorem 5.4. *The on-line algorithm MSH4 for MSP4 is $\max\{\frac{3}{2}+\theta, \frac{3}{2}+\sqrt{\frac{D}{T+D}}\}$ -competitive.*

Proof of Theorem 5.4: Let η be the schedule obtained by algorithm MSH4. The proof has two parts for the two different cases of the algorithm.

Case 1: $T > (1 + \theta)D$. Suppose τ is the last delivery time before $\rho_{max}(\eta)$ when there is an unfull batch. Note that if there is no such τ , let $\tau = 0$. As such, all the jobs completed after τ are released after τ , which means $\tau < C_{max}(opt)$. Meanwhile, for every deliver time between τ and $\rho_{max}(\eta)$, there is either no batch or there is a full batch. Let h be the number of these full batches. At last, there will be a possible unfull batch at $\rho_{max}(\eta)$. Therefore, there will be more than hC jobs released after τ . In opt , there are at least $h + 1$ batches after τ , that is, $D_{max}(opt) \geq \max\{\tau + (h + 1)T, C_{max}(opt) + T\}$ and $TC(opt) \geq (h + 1)D$. For η , the worst case is that all these jobs are delivered after $C_{max}(\eta)$: $D_{max}(\eta) \leq \max\{\tau + T, C_{max}(\eta)\} + (h + 1)T \leq C_{max}(\eta) + (h + 2)T$ and $TC(\eta) \leq \frac{\tau}{T}D + (h + 1)D$.

$$\begin{aligned}
\frac{Z(\eta)}{Z(opt)} &\leq \frac{C_{max}(\eta) + (h + 2)T + \frac{\tau}{T}D + (h + 1)D}{\max\{\tau + (h + 1)T, C_{max}(opt) + T\} + (h + 1)D} \\
&\leq \frac{\frac{3}{2}C_{max}(opt) - \frac{1}{2}\tau + (h + 2)T + \frac{\tau}{T}D + (h + 1)D}{\max\{\tau + (h + 1)T, C_{max}(opt) + T\} + (h + 1)D} \\
&\leq 1 + \frac{\frac{1}{2}C_{max}(opt) - \frac{1}{2}\tau + (h + 1)T + \frac{\tau}{T}D}{\max\{\tau + (h + 1)T, C_{max}(opt) + T\} + (h + 1)D} \\
&\leq \frac{3}{2} + \frac{(h + 1)T + \frac{\tau}{T}D}{\max\{\tau + (h + 1)T, C_{max}(opt) + T\} + (h + 1)D} \leq \frac{3}{2} + \theta.
\end{aligned} \tag{5.9}$$

Case 2: $T \leq (1 + \theta)B$. The same τ and h with Case 1 are defined. As such, there is $D_{max}(opt) \geq \max\{\tau + (h + 1)T, C_{max} + T\}$, $TC(opt) \geq (h + 1)D$, $D_{max}(\eta) \leq C_{max}(\eta) +$

$(h+1)\sqrt{D(T+D)} + T$ and $TC(\eta) \leq \frac{\tau}{\sqrt{B(T+B)}}B + (h+1)B$.

$$\begin{aligned}
\frac{Z(\eta)}{Z(opt)} &\leq \frac{C_{max}(\eta) + (h+1)\sqrt{D(T+D)} + T + \frac{\tau}{\sqrt{D(T+D)}}D + (h+1)D}{\max\{\tau + (h+1)T, C_{max}(opt) + T\} + (h+1)D} \\
&\leq \frac{\frac{3}{2}C_{max}(opt) - \frac{1}{2}\tau + (h+1)\sqrt{D(T+D)} + T + \frac{\tau}{\sqrt{D(T+D)}}D + (h+1)D}{\max\{\tau + (h+1)T, C_{max}(opt) + T\} + (h+1)D} \\
&\leq 1 + \frac{\frac{1}{2}C_{max}(opt) - \frac{1}{2}\tau + (h+1)\sqrt{D(T+D)} + \frac{\tau}{\sqrt{D(T+D)}}D}{\max\{\tau + (h+1)T, C_{max}(opt) + T\} + (h+1)D} \\
&\leq \frac{3}{2} + \frac{(h+1)\sqrt{D(T+D)} + \frac{\tau}{\sqrt{D(T+D)}}D}{\max\{\tau + (h+1)T, C_{max}(opt) + T\} + (h+1)D} \leq \frac{3}{2} + \sqrt{\frac{D}{T+D}}.
\end{aligned} \tag{5.10}$$

□

5.5 Algorithm for Problem MSP5

MSP5 has the following features: Jobs are released on-line and delivered in "direct" pattern. The capacity of vehicles is C and the number of vehicles is enough. The lower bound of MSP5 can be derived from single-machine case [Han, 2012].

Corollary 5.5. *No on-line algorithm for MSP5 can have a competitive ratio less than $\max\{1 + \theta, 2 - \frac{1}{C}\}$, even if all processing times are 0.*

Algorithm MSH5

Jobs are scheduled on the machines with the LPT-rule.

At time lD , where $l \geq 1$ is an integer, if there is no uncompleted job, then there are batches to deliver all completed jobs; otherwise there are only full bathes to deliver as many completed jobs as possible.

The on-line algorithm MSH5 for MSP5 has a competitive ratio 2, which is analyzed as follows.

Theorem 5.5. *The competitive ratio for the on-line algorithm MSH5 for MSP5 is 2.*

Proof of Theorem 5.5: Let η be the schedule obtained by algorithm MSH5. Suppose τ is the last delivery time before $\rho_{max}(\eta)$ when there is no batch or not all batches are full. Note that if there is no such τ , let $\tau = 0$. As such, all the jobs completed after τ are released after τ , which means $\tau < C_{max}(opt)$. Let $h + 1$ be the number of batches for the delivery time after τ , where only one possible batch at $\rho_{max}(\eta)$ is unfull and the other h batches are all full. Meanwhile, suppose that there are a full batches and b unfull batches at the delivery time no later than τ . As at every delivery time, there is at most one unfull batch, $bD \leq \tau$. Because the number of jobs is more than $(a + h)C$, there are at least $a + h + 1$ batches in opt . $D_{max}(\eta) \leq C_{max}(\eta) + D + T$, $TC(\eta) = (a + b + h + 1)D$, $D_{max}(opt) = C_{max}(opt) + T$ and $TC(opt) \geq (a + h + 1)D$.

$$\begin{aligned}
\frac{Z(\eta)}{Z(opt)} &\leq \frac{C_{max}(\eta) + D + T + (a + b + h + 1)D}{C_{max}(opt) + T + (a + h + 1)D} \\
&\leq \frac{\frac{3}{2}C_{max}(opt) - \frac{1}{2}\tau + D + T + (a + b + h + 1)D}{C_{max}(opt) + T + (a + h + 1)D} \\
&\leq 1 + \frac{\frac{1}{2}C_{max}(opt) - \frac{1}{2}\tau + D + bD}{C_{max}(opt) + T + (a + h + 1)D} \\
&\leq 1 + \frac{\frac{1}{2}C_{max}(opt) + \frac{1}{2}\tau + D}{C_{max}(opt) + T + (a + h + 1)D} \leq 2.
\end{aligned} \tag{5.11}$$

At last, it needs to show that the ratio of the algorithm can achieve 2 for the following instance. At time 0, there is a job with 0 processing time released, and the ratio of the result of MSH5 to that of opt will achieve 2 when T tends to 0. This completes the proof

□

5.6 Simulated Experiment for MSH5

In this section, a simulation is conducted to demonstrate the run-time and the performance of the algorithm MSH5 in normal scenarios and illustrate how the algorithm is used in practice. An instance can be defined by prescribing a set of the foregoing parameters (n , p_j , and r_j , for $j = 1, 2, \dots, n$, C and D). The instances were generated by these randomly generated parameters. The algorithm was implemented in the Matlab environment. The parameters are thus determined based on the following assumptions:

- (1) The release of jobs follows the poisson distribution with the parameter λ , i.e., the number of jobs released at some time r : $n(r) \sim P(\lambda)$ and the next release time is $r + r'$, where $r' \sim U(0, \lambda)$, λ is two times of the mean value of the release intervals for the customer i , and $\lambda \sim U(0, \Lambda)$.
- (2) The job processing time follows the uniform distribution in the interval $[0, b]$, i.e., $p_j \sim U(0, b)$ for $j = 1, 2, \dots, n$, where b is two times of the mean value of the processing time and $b \sim U(0, B)$.
- (3) The number of jobs follows the uniform distribution in the set $\{1, 2, \dots, N\}$, i.e., $Pr\{n = h\} = \frac{1}{N}$ for $h = 1, 2, \dots, N$ where N is two times of the mean value of the number of jobs.
- (4) The delivery cost D is a constant.
- (5) The number of machines is of three cases: $m = 2$, $m = 5$, and $m = 8$.
- (6) The capacity of vehicles is of three cases: $C = 2$, $C = 5$, and $C = 8$.

(7) The unit cost of delivery is of three cases: $D = 5$, $D = 500$, and $D = 0.05$.

By choosing different values for Λ , B , and N , instances are generated and scheduling is then executed. In all cases, 100 instances are generated. Table 5.1 shows the results. Each row of the table is the average of the results of the 100 instances. The columns in the table are (1) the ratio of the algorithm value to the benchmark value, where benchmark value is obtained by SA for off-line version of MSP5, (2) the run-time in seconds, respectively.

Table 5.1. Results of Algorithm MSH5

m	D	C	MSH5		D	C	MSH5		D	C	MSH5	
			ratio	time			ratio	time			ratio	time
2	5	2	1.12	0.00092	500	2	1.07	0.00027	0.05	2	1.0037	0.0029
	5	5	1.13	0.00089	500	5	1.07	0.00025	0.05	5	1.0026	0.0023
	5	8	1.13	0.00084	500	8	1.06	0.00027	0.05	8	1.0050	0.0028
5	5	2	1.25	0.00086	500	2	1.14	0.00028	0.05	2	1.0053	0.0029
	5	5	1.27	0.00080	500	5	1.12	0.00026	0.05	5	1.0031	0.0023
	5	8	1.26	0.00079	500	8	1.15	0.00028	0.05	8	1.0067	0.0028
8	5	2	1.31	0.00086	500	2	1.15	0.00031	0.05	2	1.0068	0.0032
	5	5	1.31	0.00092	500	5	1.21	0.00027	0.05	5	1.0055	0.0028
	5	8	1.30	0.00073	500	8	1.14	0.00034	0.05	8	1.0052	0.0031

From Table 5.1, it is evident that the ratio columns of MSH5 never exceed 1.32 for all different values of m , D and C , which exhibits the robustness of the algorithm and is also consistent with the result of Theorem 5.5. Actually, the ratios of the algorithm are much

better than the theoretical result, 2, which shows the excellent performance on the normal instances. The run-time for MSH5 never exceeds 0.0035 seconds , so the efficiency of the algorithms is very high. Furthermore, it can be inferred that the five algorithms MSH can perform well for normal scenarios and are robust to the worst case owing to their similar structure.

5.7 Concluding Remarks

In this chapter, five on-line problems (denoted by MSP1-MSP5) for multi-machines and single-customer were proposed. The problems were of different processing patterns and vehicles characteristics. Corresponding algorithms were developed for the problems and a simulation for one of them was conducted. The algorithms are robust and efficient according to competitive ratio analysis. In particular, the algorithms (MSH1 and MSH2) for MSP1 and MSP2 can achieve optimal results in terms of the competitive ratio and lower bound. For the other problems, the larger the parameter C , the better the result. The simulation of MSH5 for MSP5 shows that the algorithm performs robustness for worst instances and great for most normal instances, and possess efficiency even for different values of m and C . Owing to the similar structure of problems and algorithms, one can conclude the five developed algorithms can perform well for normal scenarios and are robust to the worst case.

CHAPTER 6

ALGORITHMS FOR MULTI-MACHINES AND MULTI-CUSTOMERS PROBLEMS

In this chapter, the problems for multi-machines and multi-customers are considered and corresponding algorithms are developed. As described in Section 3.3, eight problems are defined in terms of different processing patterns, vehicles characteristics and delivery patterns. For all these problems, the corresponding algorithms are developed and the simulation experiment is presented.

6.1 Algorithm for Problem MMP1

MMP1 has the following features: Jobs are released off-line, processed in "pmtn" pattern and delivered in "direct" pattern. The capacity of vehicles and the number of vehicles are both enough.

Similar to the induction in subsection 4.1, it can be shown that MMP1 is equivalent with the agent scheduling problem $Pm|r_j, pmtn|\sum C_{max}^{(i)}$. The assumption that $T_{0i} = 0$ for all i and $D = 0$ can also be applied.

When k is a parameter, the classical scheduling problem $Pm|r_j, pmtn|\sum C_j$ is a special case of MMP1. As $Pm|r_j, pmtn|\sum C_j$ with fixed m is NP-hard [Du et al., 1990], MMP1 with fixed m is at least NP-hard. When both k and m are parameters, MMP1 is SNP-hard

[Baptiste et al., 2007].

Corollary 6.1. *When k is a parameter, MMP1 is a NP-hard problem. When both k and m are parameters, MMP1 is a SNP-hard problem.*

As there are multiple customers and multiple processors, jobs of each customer need to be processed as soon as possible but the number of occupied machines is as few as possible.

For a jobs set U on m machines, if $p_{mean}(U) \geq p_{max}(U)$, the minimum number of processors is m ; otherwise, it is $\lceil \frac{\sum p_i}{p_{max}(U)} \rceil$. Based on this idea, the following algorithm is proposed.

Algorithm MMH1

The customers are re-indexed in an increasing order of $C_{max}(\mathcal{J}^{(i)}, m, opt^{(i)})$, where $opt^{(i)}$ is the optimal schedule for $\mathcal{J}^{(i)}$ on m machines (If there is more than one customer with the same $C_{max}(\mathcal{J}^{(i)}, m, opt^{(i)})$, their order is the original index order). When a new job arrives or a machine is free, all the machines are re-assigned to the customers in terms of the priority such that the jobs of each customer occupy the minimum number of free machines.

6.2 Algorithm for Problem MMP2

MMP2 has the following features: Jobs are released on-line, processed in "pmtn" pattern and delivered in "direct" pattern. The capacity of vehicles and the number of vehicles are both enough. Actually, MMP2 is the on-line version of MMP1. A lower bound of MMP2 can be derived directly from the lower bound of SMP2.

Corollary 6.2. *No on-line algorithm for MMP2 can have competitive ratio less than 2, even all processing times being 0.*

Algorithm MMH2

At the time t that a new job arrives, the customers are re-indexed in the increasing order of $C_{max}(\mathcal{J}_{<t}^{(i)}, m, opt_{<t}^{(i)})$, where $opt_{<t}^{(i)}$ is the optimal schedule for $\mathcal{J}_{<t}^{(i)}$ on m machines (If there is more than one customer with the same $C_{max}(\mathcal{J}_{<t}^{(i)}, m, opt_{<t}^{(i)})$, their order is the original index order). When a new job arrives or a machine is free, all the machines are re-assigned to the customers in terms of the on-line priority such that the jobs of each customer occupy the minimum number of free machines.

At the time of lD where $l \geq 1$ and l is integer, if there is no uncompleted job for Customer i , then there must be a batch to deliver all the completed jobs for Customer i , otherwise there is no operation for these jobs.

6.3 Algorithm for Problem MMP3

MMP3 has the following features: Jobs are released off-line and delivered in "direct" pattern. The capacity of vehicles and the number of vehicles are both enough.

Similarly, MMP3 is equivalent with the agent scheduling problem $Pm|r_j|\sum C_{max}^{(i)}$. The assumption that $T_{0i} = 0$ for all i and $D = 0$ can also be applied. As the preemption of jobs processing is not allowed, MMP3 is at least SNP-hard.

Corollary 6.3. *MMP3 is a SNP-hard problem.*

LPT-rule can generate a great approximate algorithm for single customer case. When there is more than one customer, processing the longest job may delay other customers'

completion time. Therefore, there should be a period waiting time for long jobs. The ready job for multiple machines case is defined as follows.

Definition 6.1. A job $J_j^{(i)}$ is called ready at time t if it has arrives ($r_j^{(i)} \leq t$), not completed ($C_j^{(i)} > t$) and $\frac{1}{m+1}p_j^{(i)} \leq t$.

Algorithm MMH3

The customers are re-indexed in the increasing order of $C_{max}(\mathcal{J}^{(i)}, m, \eta_L^{(i)})$, where $\eta_L^{(i)}$ is the schedule generated by LPT-rule for $\mathcal{J}^{(i)}$ on m machines (If there is more than one customer with the same $C_{max}(\mathcal{J}^{(i)}, m, \eta_L^{(i)})$, their order is the original index order). When a machine is free, prepare the longest ready job of the customer with the highest priority.

6.4 Algorithm for Problem MMP4

MMP4 has the following features: Jobs are released on-line, and delivered in "direct" pattern. The capacity of vehicles and the number of vehicles are both enough. Actually, MMP4 is the on-line version of MMP3. The same lower bound can be applied directly.

Corollary 6.4. No on-line algorithm for MMP4 can have competitive ratio less than 2, even all processing times being 0.

Algorithm MMH4

At the time t that a new job arrives, the customers are re-indexed in the increasing order of $C_{max}(\mathcal{J}_{<t}^{(i)}, m, \eta_{<t,L}^{(i)})$, where $\eta_{<t,L}^{(i)}$ is the schedule generated by LPT-rule for $\mathcal{J}_{<t}^{(i)}$ on m machines (If there is more than one customer with the same $C_{max}(\mathcal{J}_{<t}^{(i)}, m, \eta_{<t,L}^{(i)})$, their order is the original index order). When a machine is free, process the longest ready job of the customer with the highest on-line priority.

At the time of lD where $l \geq 1$ and l is integer, if there is no uncompleted job for Customer i , then there must be a batch to deliver all the completed jobs for Customer i , otherwise there is no operation for these jobs.

6.5 Simulated Experiment for MMH without Routing

In this subsection, a simulation is conducted to demonstrate the run-time and the performance of the above algorithms (MMH1-MMH4) in normal scenarios and illustrate how the algorithms are used in practice. An instance can be defined by prescribing a set of the foregoing parameters ($n_i, p_j^{(i)}$ and $r_j^{(i)}$ for $j = 1, 2, \dots, n_i$, and D). The instances were generated by stochastically choosing the parameters. The algorithm was implemented in the Matlab environment. The parameters are thus determined based on the following assumptions:

- (1) The release of jobs for Customer i follows the poisson distribution with the parameter λ_i , i.e., the number of jobs released at some time r : $n_i(r) \sim P(\lambda_i)$ and the next release time is $r + r'$, where $r' \sim U(0, \lambda_i)$, λ_i is two times of the mean value of the release intervals for Customer i , and $\lambda_i \sim U(0, \Lambda_i)$ ($i = 1, 2, \dots, k$).
- (2) The job processing time for Customer i follows the uniform distribution in the interval $[0, b_i]$, i.e., $p_j^{(i)} \sim U(0, b_i)$ for $j = 1, 2, \dots, n_i$, where b_i is two times of the mean value of the processing time for Customer i and $b_i \sim U(0, B_i)$ ($i = 1, 2, \dots, k$).
- (3) The number of jobs for Customer i follows the uniform distribution in the set $\{1, 2, \dots, N_i\}$, i.e., $Pr\{n_i = h\} = \frac{1}{N_i}$ for $h = 1, 2, \dots, N_i$ where N_i is two times of the mean value of the number of jobs for Customer i ($i = 1, 2, \dots, k$).

- (4) The delivery cost D is a constant.
- (5) The number of customers is of four cases: $k = 2$, $k = 5$, $k = 10$, and $k = 20$.
- (6) The number of machines is of three cases: $m = 2$, $m = 5$, and $m = 8$.

By choosing different values for Λ_i , B_i , and N_i , instances are generated and scheduling is then executed. In all cases, 100 instances are generated. Table 6.1 shows the result for the case $k = 2$. Each row of the table is the average of the results of the 100 instances. The algorithm columns of the table are (1) the ratio of the algorithm value to the benchmark value (BV), (2) the run-time in seconds, respectively. The benchmark value is computed as follows: For every instance I of MMP1, the corresponding instance \bar{I} of SMP1 is constructed, where $\bar{p}_j^{(i)} = \frac{1}{m}p_j^{(i)}$ and all the other parameters are the same. Notice that the off-line optimal value of \bar{I} is a lower bound of that of I . The benchmark value is the result of SMH1 for \bar{I} . As the benchmark value is the lower bound of the off-line optimal value of MMP1-MMP4, the ratios of the results of MMH1-MMH4 to their corresponding optimal result (which are not known) are better.

For the off-line algorithm MMH1 and MMH3, the ratio is very close to 1, which means that the algorithms can construct a great solution. For the on-line algorithms MMH2 and MMH4 the ratio never exceeds 2. Actually, for most cases, the ratios of the two algorithms are not greater than 1.65, which shows the excellent performance on the normal instances. The run-time for all the algorithms (MMH1-MMH4) are very short, so the efficiency of the algorithms is very high when $k = 2$.

Table 6.1. Results of Algorithms MMH1-MMH4 for $k = 2$

m	BV	MMH1		MMH2		MMH3		MMH4	
		ratio	time	ratio	time	ratio	time	ratio	time
2	61.34	1.01	0.0025	1.39	0.011	1.01	0.0026	1.38	0.0071
	63.36	1.01	0.0026	1.10	0.011	1.02	0.0025	1.10	0.0072
	57.92	1.00	0.0024	1.99	0.0095	1.00	0.0024	1.99	0.0065
	77.52	1.01	0.0022	1.26	0.0079	1.01	0.0027	1.26	0.0061
	50.83	1.00	0.0032	1.80	0.015	1.00	0.0035	1.80	0.017
	913.49	1.00	0.0012	1.01	0.0022	1.03	0.0029	1.04	0.0038
5	56.82	1.04	0.0035	1.52	0.015	1.01	0.0032	1.59	0.0094
	54.27	1.04	0.0032	1.14	0.013	1.02	0.0029	1.13	0.0087
	53.97	1.00	0.0030	1.99	0.013	1.00	0.0029	1.99	0.0089
	65.38	1.03	0.0028	1.36	0.011	1.01	0.0031	1.40	0.0079
	47.31	1.00	0.0032	1.79	0.015	1.00	0.0037	1.79	0.022
	426.49	1.04	0.0012	1.06	0.0019	1.04	0.0028	1.06	0.0037
8	52.45	1.04	0.0033	1.52	0.014	1.02	0.0034	1.64	0.011
	49.21	1.06	0.0030	1.16	0.011	1.02	0.0030	1.13	0.0090
	54.45	1.00	0.0032	1.99	0.015	1.00	0.0033	1.99	0.011
	57.51	1.05	0.0026	1.42	0.0096	1.01	0.0031	1.50	0.0081
	51.10	1.00	0.0033	1.81	0.016	1.00	0.0046	1.81	0.030
	298.40	1.07	0.0011	1.10	0.0018	1.01	0.0028	1.04	0.0038

Table 6.2. Results of Algorithms MMH1-MMH4 for $k = 5$

m	BV	MMH1		MMH2		MMH3		MMH4	
		ratio	time	ratio	time	ratio	time	ratio	time
2	245.82	1.02	0.0058	1.25	0.046	1.03	0.0059	1.23	0.027
	249.00	1.02	0.0059	1.13	0.048	1.04	0.0060	1.14	0.027
	257.20	1.00	0.0072	1.98	0.059	1.00	0.0073	1.98	0.034
	385.58	1.01	0.0055	1.15	0.037	1.03	0.0067	1.15	0.025
	126.35	1.00	0.0085	1.80	0.078	1.00	0.011	1.80	0.084
	3510.95	1.00	0.0027	1.01	0.0076	1.02	0.0062	1.03	0.011
5	147.56	1.04	0.0080	1.40	0.055	1.04	0.0063	1.39	0.031
	151.81	1.05	0.0086	1.16	0.059	1.04	0.0067	1.15	0.033
	158.02	1.00	0.0098	1.99	0.072	1.00	0.0077	1.99	0.039
	192.02	1.04	0.0078	1.31	0.045	1.04	0.0074	1.30	0.030
	126.12	1.00	0.0086	1.81	0.079	1.00	0.012	1.81	0.12
	1539.13	1.02	0.0032	1.05	0.0088	1.06	0.0074	1.08	0.014
8	139.77	1.05	0.0096	1.44	0.068	1.02	0.0073	1.50	0.040
	153.41	1.06	0.012	1.15	0.088	1.03	0.0089	1.14	0.050
	144.74	1.00	0.012	1.99	0.088	1.00	0.0087	1.99	0.049
	165.27	1.06	0.0089	1.37	0.052	1.04	0.0079	1.36	0.034
	129.53	1.00	0.0088	1.81	0.083	1.00	0.014	1.81	0.16
	979.53	1.06	0.0033	1.09	0.0093	1.07	0.0078	1.10	0.014

Table 6.3. Results of Algorithms MMH1-MMH4 for $k = 10$

m	BV	MMH1		MMH2		MMH3		MMH4	
		ratio	time	ratio	time	ratio	time	ratio	time
2	720.73	1.08	0.012	1.23	0.14	1.09	0.015	1.23	0.097
	719.86	1.07	0.011	1.16	0.12	1.08	0.013	1.17	0.084
	765.35	1.00	0.0097	1.97	0.12	1.00	0.013	1.97	0.082
	1300.70	1.03	0.0082	1.11	0.081	1.04	0.012	1.12	0.060
	254.63	1.00	0.017	1.81	0.28	1.00	0.027	1.81	0.28
	10441.31	1.00	0.0056	1.01	0.026	1.01	0.013	1.02	0.030
5	396.63	1.08	0.014	1.37	0.14	1.08	0.015	1.36	0.11
	377.89	1.08	0.016	1.25	0.16	1.09	0.017	1.25	0.12
	371.93	1.00	0.013	1.99	0.12	1.00	0.013	1.99	0.093
	597.20	1.08	0.0099	1.28	0.083	1.09	0.013	1.28	0.074
	250.61	1.00	0.017	1.80	0.27	1.00	0.029	1.80	0.38
	4424.79	1.03	0.0057	1.05	0.025	1.05	0.013	1.07	0.032
8	309.57	1.05	0.019	1.41	0.17	1.04	0.016	1.40	0.13
	308.10	1.06	0.019	1.20	0.17	1.04	0.016	1.19	0.13
	310.21	1.00	0.017	1.99	0.16	1.00	0.015	1.99	0.12
	423.24	1.08	0.012	1.35	0.099	1.08	0.014	1.35	0.084
	251.29	1.00	0.019	1.82	0.30	1.00	0.034	1.82	0.56
	3009.72	1.06	0.0068	1.09	0.029	1.07	0.015	1.09	0.039

Table 6.4. Results of Algorithms MMH1-MMH4 for $k = 20$

m	BV	MMH1		MMH2		MMH3		MMH4	
		ratio	time	ratio	time	ratio	time	ratio	time
2	2705.00	1.09	0.021	1.16	0.44	1.09	0.026	1.16	0.29
	2705.00	1.09	0.021	1.14	0.45	1.10	0.026	1.14	0.29
	2705.00	1.00	0.020	1.95	0.42	1.00	0.025	1.95	0.28
	4142.52	1.04	0.022	1.09	0.41	1.05	0.033	1.09	0.29
	499.76	1.00	0.039	1.81	1.1	1.00	0.081	1.81	1.0
	38872.40	1.01	0.011	1.01	0.085	1.01	0.025	1.01	0.079
5	1169.60	1.18	0.024	1.36	0.43	1.18	0.028	1.35	0.35
	1169.60	1.19	0.024	1.31	0.44	1.19	0.029	1.31	0.36
	1169.60	1.00	0.028	1.98	0.49	1.00	0.032	1.98	0.40
	1666.90	1.16	0.021	1.29	0.30	1.16	0.029	1.28	0.27
	511.32	1.00	0.040	1.82	1.1	1.00	0.088	1.82	1.5
	16047.53	1.03	0.013	1.05	0.095	1.04	0.029	1.05	0.10
8	800.21	1.11	0.031	1.39	0.47	1.11	0.032	1.38	0.45
	800.21	1.13	0.032	1.30	0.49	1.13	0.034	1.30	0.47
	800.21	1.00	0.042	1.98	0.63	1.00	0.043	1.98	0.62
	1111.40	1.14	0.023	1.35	0.31	1.14	0.030	1.35	0.31
	524.49	1.00	0.041	1.82	1.1	1.00	0.094	1.82	2.0
	10214.94	1.07	0.015	1.10	0.11	1.08	0.031	1.10	0.12

Tables 6.2-6.4 give the results with $k = 5$, $k = 10$ and $k = 20$, respectively. The performance of the results support the foregoing conclusive discussions. For $k = 10$ and $k = 20$, the benchmark value is computed by SA_SMH1 for I' . By examining the run-time results of MMH1-MMH4, the algorithms tend to have a polynomial time complexity. This means that the algorithms can well be scaled to a much larger problem.

6.6 Algorithm for Problem MMP5

MMP5 has the following features: Jobs are released off-line, processed in "pmtn" pattern and delivered in "routing" pattern. The capacity of vehicles and the number of vehicles are both enough.

When the number of aid sites k is a parameter, the delivery part is SNP-hard, which implies the problem is at least SNP hard.

Corollary 6.5. *SMP5 is a SNP-hard problem.*

Property 4.1 and 4.2 can still be applied for MMP5, and the jobs of any two customers in different subsets of a weak partition cannot be in the same batch. An algorithm combining SMH5 and MMH1 is applied for this problem.

Algorithm MMH5

For every instance I of MMP5, the corresponding instance \bar{I} of SMP5 is constructed, where $\bar{p}_j^{(i)} = \frac{1}{m}p_j^{(i)}$ and all the other parameters are the same. Then the algorithm SMH5 is applied to \bar{I} to generate the schedule $\bar{\eta}$.

The customers are re-indexed in the increasing order of $C_{max}^{(i)}(\bar{\eta})$ (If there is more than

one customer with the same $C_{max}^{(i)}(\bar{\eta})$, their order is the original index order). When a new job arrives or a machine is free, all the machines are re-assigned to the machines in terms of the priority such that the jobs of each customer occupy the minimum number of free machines.

The batch delivery is the same with $\bar{\eta}$.

6.7 Algorithm for Problem MMP6

SMP6 has the following features: Jobs are released on-line, processed in "pmtn" pattern and delivered in "routing" pattern. The capacity of vehicles and the number of vehicles are both enough. Actually, MMP6 is the on-line version of MMP5. The same lower bound can be applied to this on-line problem.

Corollary 6.6. *No on-line algorithm for MMP6 can have competitive ratio less than 2, even all processing times being 0.*

An algorithm combining SMH6 and MMH2 is applied for this problem.

Algorithm MMH6

At the time t that a new job arrives, the customers are re-indexed in the increasing order of $C_{max}(\mathcal{J}_{<t}^{(i)}, m, opt_{<t}^{(i)})$, where $opt_{<t}^{(i)}$ is the optimal schedule for $\mathcal{J}_{<t}^{(i)}$ on m machines (If there is more than one customer with the same $\mathcal{J}_{<t}^{(i)}$, their order is the original index order). When a new job arrives or a machine is free, all the machines are re-assigned to the customers in terms of the on-line priority such that the jobs of each customer occupy the minimum number of free machines.

Set $l_q = 0$ for $q = 1, 2, \dots, o$. At every time of $\frac{l}{|SK_q|}D$, where $l \geq 1$ and l is integer, if

there are s_q customers in SK_q with completed jobs but no uncompleted job, and $l - l_q > |SK_q| - s_q$, deliver all their jobs in a batch, let $l_q = l$; otherwise no operation.

6.8 Algorithm for Problem MMP7

MMP7 has the following features: Jobs are released off-line and delivered in "routing" pattern. The capacity of vehicles and the number of vehicles are both enough.

For MMP7, both the preparation part and the delivery part are strongly NP-hard, respectively, which implies MMP7 is at least strongly NP hard.

Corollary 6.7. *MMP7 is a SNP-hard problem.*

Similar with MMP5, an algorithm combining SMH7 and MMH3 is applied for this problem.

Algorithm MMH7

For every instance I of MMP7, the corresponding instance \bar{I} of SMP5 is constructed, where $\bar{p}_j^{(i)} = \frac{1}{m}p_j^{(i)}$ and all the other parameters are the same. Then the algorithm SMH5 is applied to \bar{I} to generate the schedule $\bar{\eta}$.

The customers are re-indexed in the increasing order of $C_{max}^{(i)}(\bar{\eta})$ (If there is more than one customer with the same $C_{max}^{(i)}(\bar{\eta})$, their order is the original index order). When a machine free, process the longest ready job of the customer with the highest priority.

The batch delivery is the same with $\bar{\eta}$.

6.9 Algorithm for Problem MMP8

SMP6 has the following features: Jobs are released on-line and delivered in "routing" pattern. The capacity of vehicles and the number of vehicles are both enough. Actually, MMP8 is the on-line version of MMP7. The same lower bound can be applied to this on-line problem.

Corollary 6.8. *No on-line algorithm for MMP6 can have competitive ratio less than 2, even all processing times being 0.*

An algorithm combining SMH8 and MMH4 is applied for this problem.

Algorithm MMH8

At the time t that a new job arrives, the customers are re-indexed in the increasing order of $C_{max}(\mathcal{J}_{<t}^{(i)}, m, \sigma_{<t,L}^{(i)})$, where $opt_{<t,L}^{(i)}$ is the schedule generated by LPT-rule for $\mathcal{J}_{<t}^{(i)}$ on m machines (If there is more than one customer with the same $C_{max}(\mathcal{J}_{<t}^{(i)}, m, \sigma_{<t,L}^{(i)})$, their order is the original index order). When a machine is free, process the longest ready job of the customer with the highest on-line priority.

Set $l_q = 0$ for $q = 1, 2, \dots, o$. At every time of $\frac{l}{|SK_q|}D$, where $l \geq 1$ and l is integer, if there are s_q customers in SK_q with completed jobs but no uncompleted job, and $l - l_q > |SK_q| - s_q$, deliver all their jobs in a batch, let $l_q = l$; otherwise no operation.

6.10 Simulated Experiment for MMH with Routing

In this section, a simulation is conducted to demonstrate the run-time and the performance of the above algorithms (MMH5-MMH8) in normal scenarios and illustrate how

the algorithms are used in practice. An instance can be defined by prescribing a set of the foregoing parameters $(n_i, p_j^{(i)}, \text{ and } r_j^{(i)})$, for $j = 1, 2, \dots, n_i, T_{il}$ for $i, l = 1, 2, \dots, k$, and D . The instances were generated by these randomly generated parameters. The algorithm was implemented in the Matlab environment. The parameters are thus determined based on the following assumptions:

- (1) The release of jobs for Customer i follows the poisson distribution with the parameter λ_i , i.e., the number of jobs released at some time r : $n_i(r) \sim P(\lambda_i)$ and the next release time is $r + r'$, where $r' \sim U(0, \lambda_i)$, λ_i is two times of the mean value of the release intervals for Customer i , and $\lambda_i \sim U(0, \Lambda_i)$ ($i = 1, 2, \dots, k$).
- (2) The job processing time for Customer i follows the uniform distribution in the interval $[0, b_i]$, i.e., $p_j^{(i)} \sim U(0, b_i)$ for $j = 1, 2, \dots, n_i$, where b_i is two times of the mean value of the processing time for Customer i and $b_i \sim U(0, B_i)$ ($i = 1, 2, \dots, k$).
- (3) The number of jobs for Customer i follows the uniform distribution in the set $\{1, 2, \dots, N_i\}$, i.e., $Pr\{n_i = h\} = \frac{1}{N_i}$ for $h = 1, 2, \dots, N_i$ where N_i is two times of the mean value of the number of jobs for Customer i ($i = 1, 2, \dots, k$).
- (4) The positions of the manufacturer and the customers are randomly located in an square area with side length L , and the transportation network can be directly determined by the Euclidean distance.
- (5) The delivery cost D is a constant.
- (6) The number of customers is of four cases: $k = 2$, $k = 5$, $k = 10$, and $k = 20$.
- (7) The number of machines is of three cases: $m = 2$, $m = 5$, and $m = 8$.

By choosing different values for Λ_i , B_i , N_i , and L , instances are generated and scheduling is then executed. In all cases, 100 instances are generated. Table 6.5 shows the result for the case $k = 2$. Each row of the table is the average of the results of the 100 instances. The algorithm columns of the table are (1) the ratio of the algorithm value to the benchmark value (BV), (2) the run-time in seconds, respectively. The benchmark value is computed as follows: For every instance I of MMP5, the corresponding instance \bar{I} of SMP5 is constructed, where $\bar{p}_j^{(i)} = \frac{1}{m}p_j^{(i)}$ and all the other parameters are the same. Notice that the off-line optimal value of \bar{I} is a lower bound of that of I . The benchmark value is the result of SMH5 for \bar{I} . Although SMH5 may not find the global optimal solution for \bar{I} , it will be a good reference.

For the off-line algorithm MMH5 and MMH7, the ratio is very close to 1, which means that the algorithms can construct a great solution. For the on-line algorithms MMH6 and MMH8 the ratio never exceeds 2. Actually, for most cases, the ratios of the two algorithms are not greater than 1.6, which shows the excellent performance on the normal instances. The run-time for all the algorithms (MMH5-MMH8) are very short, so the efficiency of the algorithms is very high when $k = 2$.

Table 6.5. Results of Algorithms MMH5-MMH8 for $k = 2$

m	BV	MMH5		MMH6		MMH7		MMH8	
		ratio	time	ratio	time	ratio	time	ratio	time
2	219.04	1.00	0.03	1.52	0.033	1.00	0.030	1.47	0.058
	112.65	1.00	0.022	1.45	0.016	1.00	0.023	1.36	0.024
	155.63	1.01	0.017	1.21	0.002	1.01	0.017	1.34	0.0018
5	127.13	1.00	0.022	1.14	0.017	1.00	0.023	1.38	0.04
	116.98	1.00	0.032	1.57	0.029	1.00	0.033	1.48	0.048
	128.50	1.00	0.022	1.19	0.0064	1.00	0.024	1.34	0.012
8	123.94	1.00	0.022	1.22	0.014	1.00	0.023	1.55	0.027
	97.66	1.00	0.021	1.45	0.011	1.00	0.022	1.51	0.023
	221.00	1.01	0.026	1.29	0.027	1.00	0.026	1.57	0.044

Tables 6.6-6.8 give the results with $k = 5$, $k = 10$ and $k = 20$, respectively. The performance of the results support the foregoing conclusive discussions. By examining the run-time of MMH6 and MMH8, the algorithms tend to have a polynomial time complexity. This means that the algorithms can well be scaled to a much larger problem. For MMH5 and MMH7, the run-time increases much more rapidly as k gets larger. The reason is that both the two algorithms have called GAs. However, the run-time for these two algorithms are still acceptable even for $k = 20$.

Table 6.6. Results of Algorithms MMH5-MMH8 for $k = 5$

m	BV	MMH5		MMH6		MMH7		MMH8	
		ratio	time	ratio	time	ratio	time	ratio	time
2	412.45	1.00	0.56	1.30	0.100	1.00	0.56	1.30	0.13
	244.43	1.00	0.28	1.33	0.037	1.00	0.28	1.32	0.046
	723.22	1.00	0.37	1.09	0.032	1.00	0.37	1.13	0.027
5	376.35	1.00	0.38	1.31	0.064	1.00	0.39	1.34	0.11
	312.85	1.00	0.46	1.45	0.082	1.00	0.46	1.38	0.14
	421.43	1.02	0.33	1.09	0.029	1.00	0.33	1.28	0.034
8	459.11	1.00	0.35	1.15	0.055	1.00	0.35	1.35	0.14
	292.36	1.00	0.49	1.40	0.084	1.00	0.49	1.39	0.17
	377.35	1.03	0.46	1.34	0.052	1.03	0.45	1.44	0.07

Table 6.7. Results of Algorithms MMH5-MMH8 for $k = 10$

m	BV	MMH5		MMH6		MMH7		MMH8	
		ratio	time	ratio	time	ratio	time	ratio	time
2	767.06	1.00	5.95	1.28	0.28	1.00	5.96	1.33	0.32
	526.62	1.00	4.86	1.26	0.15	1.00	4.87	1.24	0.18
	1676.16	1.00	6.27	1.09	0.11	1.01	6.28	1.14	0.072
5	776.52	1.00	4.34	1.13	0.16	1.00	4.35	1.21	0.29
	657.52	1.00	7.16	1.40	0.38	1.00	7.18	1.35	0.57
	1029.25	1.01	6.88	1.17	0.13	1.01	6.88	1.24	0.14
8	782.41	1.00	6.39	1.41	0.32	1.00	6.40	1.46	0.69
	708.88	1.00	7.18	1.35	0.32	1.00	7.20	1.34	0.69
	918.84	1.02	6.30	1.15	0.13	1.02	6.31	1.32	0.19

Table 6.8. Results of Algorithms MMH5-MMH8 for $k = 20$

m	BV	MMH5		MMH6		MMH7		MMH8	
		ratio	time	ratio	time	ratio	time	ratio	time
2	1681.43	1.00	108.27	1.24	1.15	1.00	108.31	1.27	1.17
	929.37	1.00	136.07	1.41	1.32	1.00	136.12	1.41	1.25
	4069.42	1.00	128.5	1.06	0.36	1.06	128.51	1.07	0.27
5	1514.7	1.00	136.4	1.22	1.42	1.00	136.46	1.33	1.99
	1148.6	1.00	147.4	1.41	1.42	1.00	147.46	1.38	2.11
	2409.18	1.01	82.86	1.08	0.20	1.08	82.87	1.12	0.23
8	1606.87	1.00	123.6	1.20	1.19	1.00	123.67	1.33	2.28
	1311.11	1.00	175.77	1.40	1.85	1.00	175.84	1.39	3.22
	2117.44	1.01	84.22	1.10	0.36	1.10	84.22	1.23	0.41

6.11 Concluding Remarks

In this chapter, eight problems (denoted by MMP1-MMP8) for multi-machines and multi-customers were proposed. These problems were of different release environment, preparation patterns and delivery patterns. The algorithms (denoted by MMH1-MMH8) were modified from corresponding SMH algorithms by combining techniques of parallel-machine scheduling. A simulation study was conducted for all the algorithms. From the simulation study, all algorithms perform robustness for worst instances and great for most normal instances, and possess efficiency even for different values of m and k .

CHAPTER 7

ROBUSTNESS AND RESILIENCE OF ALGORITHMS

7.1 Introduction

In this chapter, the robustness and the resilience of algorithms are discussed. In the assessment of algorithms, the traditional approach is based on the complexity (off-line problem) and competitive ratio (on-line problem). There are then two scenarios for algorithms: normal and worst. Accordingly, there is the measure of the algorithm in the normal scenario and worst scenario by the complexity (off-line problem) and competitive ratio (on-line problem). Such traditional measures are found not enough to account for some phenomena relevant to the performance of the algorithm, particularly (1) noises on the parameters of the problems and (2) change in the structure of the parameters. These two are called robustness and resilience, respectively, borrowed from the systems theory [Zhang and Lin, 2010]. In this thesis, a qualitative definition of the robustness and resilience for algorithms (scheduling algorithms in particular) is proposed. Then, the validation of the proposed definition with the algorithms developed in the thesis, in the previous chapters, is attempted. It is noted that validation is just at the qualitative level, namely, the quantitative part of the measure is not in the scope of this thesis, though in the last chapter, there is a discussion on the quantitative part of the measure of robustness

and resilience for future work.

7.2 Definition of Robustness

There seems to be a gap in definition of the robustness between applications and theories of algorithms in the context of production scheduling. The robustness of the scheduling algorithm refers to the worst case performance of an algorithm, while in application systems or engineering systems, the robustness of a system refers to the sensitivity of the system performance to disturbances or noises. The worst scenario performance does not cover the sensitivity issue or robustness of an application system. In the following, the robustness in engineering systems is extended for algorithms.

Let \mathcal{A} be an algorithm for problem \mathcal{P} . Given an instance I of \mathcal{P} , let $s(\mathcal{A}, I)$ be the solution of \mathcal{A} and $ob(s(\mathcal{A}, I), I)$ be the corresponding objective value. When there are disturbances, the information is uncertain and there is deviation of the parameter values in I . Let I^u represents a corresponding instance of I under uncertainty. Therefore, the objective value becomes $ob(s(\mathcal{A}, I), I^u)$, which is the solution $s(\mathcal{A}, I)$ for the instance I^u . If \mathcal{A} is implemented for the instance I^u , the objective value is $ob(s(\mathcal{A}, I^u), I^u)$. In this sense, the robustness of algorithm \mathcal{A} can be reflected by comparing $ob(s(\mathcal{A}, I), I^u)$ with $ob(s(\mathcal{A}, I^u), I^u)$.

7.3 Definition of Resilience

In engineering systems, the resilience of a system refers to the persistence of the system performance to disruptions. The persistence issue or resilience of an algorithm has never

been considered in literature. In the following, the resilience in engineering systems is extended for algorithms.

Let \mathcal{A} be an algorithm for problem \mathcal{P} . Given an instance I of \mathcal{P} , let $s(\mathcal{A}, I)$ be the solution of \mathcal{A} and $ob(s(\mathcal{A}, I), I)$ be the corresponding objective value. When there are disruptions, the structure of I may be totally changed or broken. Let I^D be a corresponding instance of I under disruptions. Unlike the robustness case, the original solution $s(\mathcal{A}, I)$ may not be feasible and then the algorithm \mathcal{A} needs to be re-implemented, which generates the solution $s(\mathcal{A}, I^D)$ and the objective value $ob(s(\mathcal{A}, I^D), I^D)$. In this sense, the resilience of algorithm \mathcal{A} can be reflected by comparing $ob(s(\mathcal{A}, I^D), I^D)$ with the original value $ob(s(\mathcal{A}, I), I)$.

7.4 Robustness of Algorithms

In the following, a simulation is presented to show the sense of the robustness of algorithms SMH4 and SMH8. The two algorithms are for on-line problems. The mechanism of uncertainty can be constructed as follows: It assumes that at the release time $r_j^{(i)}$ the job $J_j^{(i)}$ arrives but the information of the processing time $p_j^{(i)}$ may not be true because of uncertainty. Indeed, the true value $\bar{p}_j^{(i)}$ will not be known until the job is completed. In the simulation, the uniform distribution is used to describe the uncertainty of the processing time: $\bar{p}_j^{(i)} \sim U[\frac{1}{2}p_j^{(i)}, \frac{3}{2}p_j^{(i)}]$.

In the simulated experiment, the same assumptions and cases (Section 4.6 and 4.11) for generating instances are applied. The running of the two on-line algorithms is like this: at time t , the decisions are made based on the jobs information $(r_j^{(i)}, \bar{p}_j^{(i)})$ if $C_j^{(i)} \leq t$ and

$(r_j^{(i)}, p_j^{(i)})$ if $r_j^{(i)} \leq t$ but $C_j^{(i)} > t$.

Table 7.1 shows the results of SMH4 for different values of k . Each row of the table is the average of the results of the 100 instances. The algorithm columns of the table are (1) the ratio of the algorithm value to the benchmark value, (2) the run-time in seconds, respectively. The benchmark value is the objective value of SMH1 (SA_SMH1), which is the case that all the true information is known beforehand.

Comparing with the results in Tables 4.1-4.5, one can conclude the algorithm SMH4 can still hold the same performance under the uncertainty, which implies the excellent robustness of SMH4.

Table 7.2 shows the results of SMH8 for different values of k and different cases. Each row of the table is the average of the results of the 100 instances. The algorithm columns of the table are (1) the ratio of the algorithm value to the benchmark value, (2) the run-time in seconds, respectively. The benchmark value is the objective value of SMH5, which is the case that all the true information is known beforehand.

Table 7.1. The Results of SMH4 under Uncertainty

k	SMH4		k	SMH4		k	SMH4	
	ratio	time		ratio	time		ratio	time
2	1.48	0.0030	3	1.31	0.0047	5	1.29	0.010
	1.12	0.0029		1.11	0.0050		1.13	0.010
	1.98	0.0029		1.97	0.0053		1.95	0.010
	1.43	0.0034		1.31	0.0053		1.24	0.012
	1.61	0.0023		1.57	0.0045		1.51	0.012
	1.09	0.0033		1.10	0.0058		1.15	0.013
8	1.28	0.030	10	1.27	0.055	20	1.34	0.094
	1.18	0.028		1.22	0.051		1.31	0.099
	1.94	0.028		1.94	0.046		1.90	0.099
	1.23	0.029		1.25	0.050		1.29	0.098
	1.51	0.027		1.51	0.048		1.47	0.083
	1.21	0.026		1.21	0.046		1.27	0.096

Comparing with the results in Table 4.9, the robustness of SMH8 under the uncertainty can also be shown.

Table 7.2. The Results of SMH8 under Uncertainty

k	Case	SMH8		k	Case	SMH8	
		ratio	time			ratio	time
2	Case 1	1.37	0.0174	5	Case 1	1.33	0.056
	Case 2	1.37	0.013		Case 2	1.37	0.053
	Case 3	1.12	0.0096		Case 3	1.14	0.040
10	Case 1	1.30	0.23	20	Case 1	1.23	0.62
	Case 2	1.37	0.17		Case 2	1.36	0.51
	Case 3	1.20	0.14		Case 3	1.35	0.55

7.5 Resilience of Algorithms

In the following, a simulated experiment is conducted to illustrate the sense of the resilience of algorithm SMH7 under disruptions. With respect to the transportation network T in normal case, the broken transportation network T^B in disruption case is constructed as follows: If the road from the place 'i' to the place 'l' is broken, $T_{il}^B = \infty$; otherwise, $T_{il}^B = T_{il}$. For other parameters, the same assumptions (Section 4.11) for generating instances hold to the experiment. The performance of SMH7 under T and T^B is simulated for the roads disruption case (no more than k pairs (i, l) are broken). Figures 7.1-7.3 show the result of the algorithm for three values of k ($k = 5, 10, 20$) under the case that the values of the processing part ($r_j^{(i)}$ and $p_j^{(i)}$), the unit delivery cost (D), and the values of transportation system (T_{il}) are almost the same.

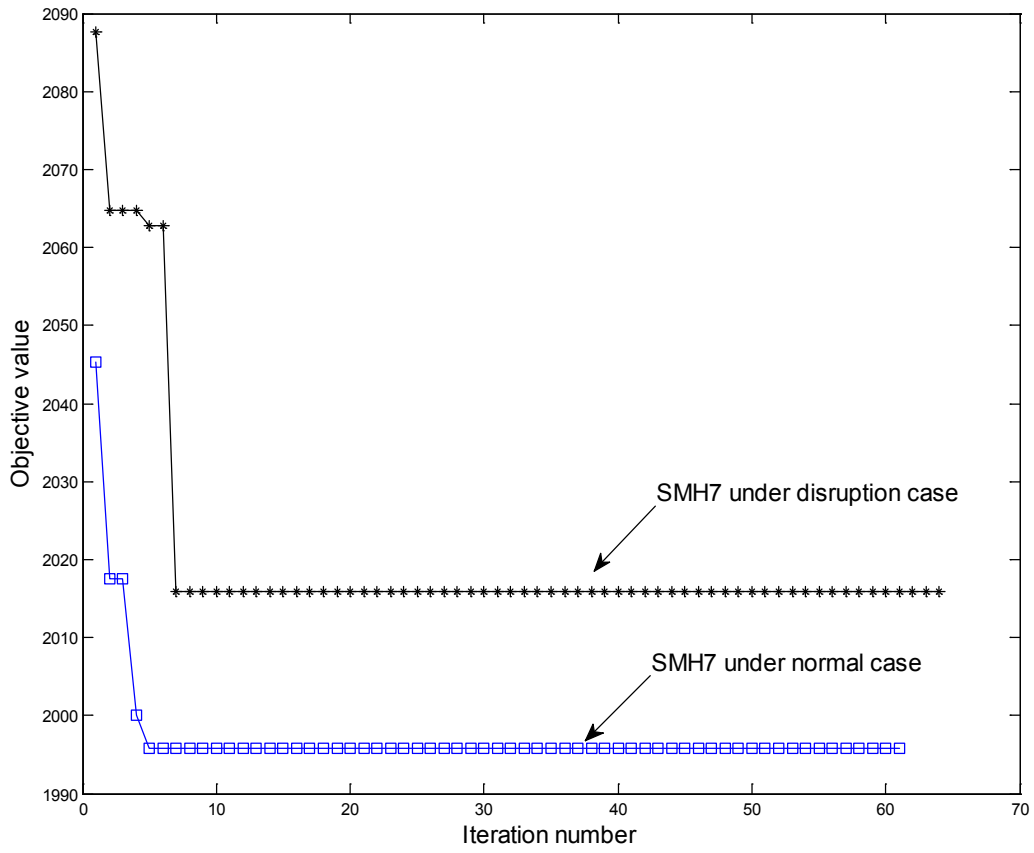


Figure 7.1. The results of SMH7 under normal and disruption case with $k = 5$

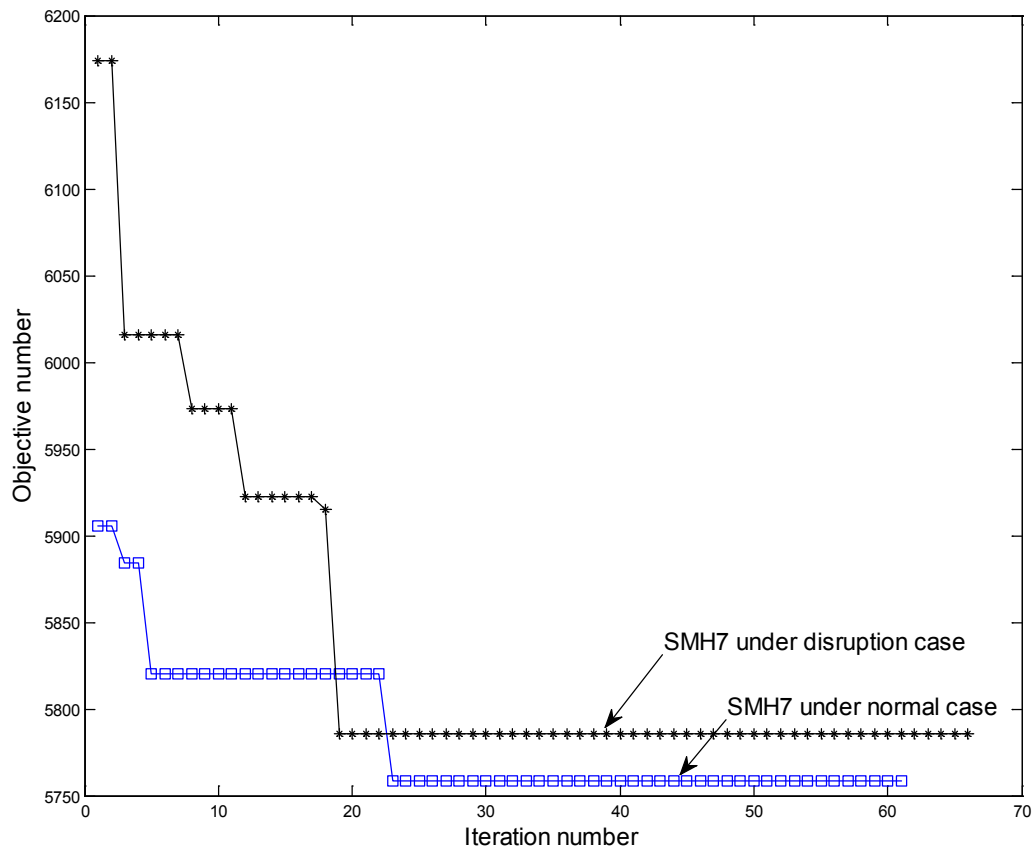


Figure 7.2. The results of SMH7 under normal and disruption case with $k = 10$

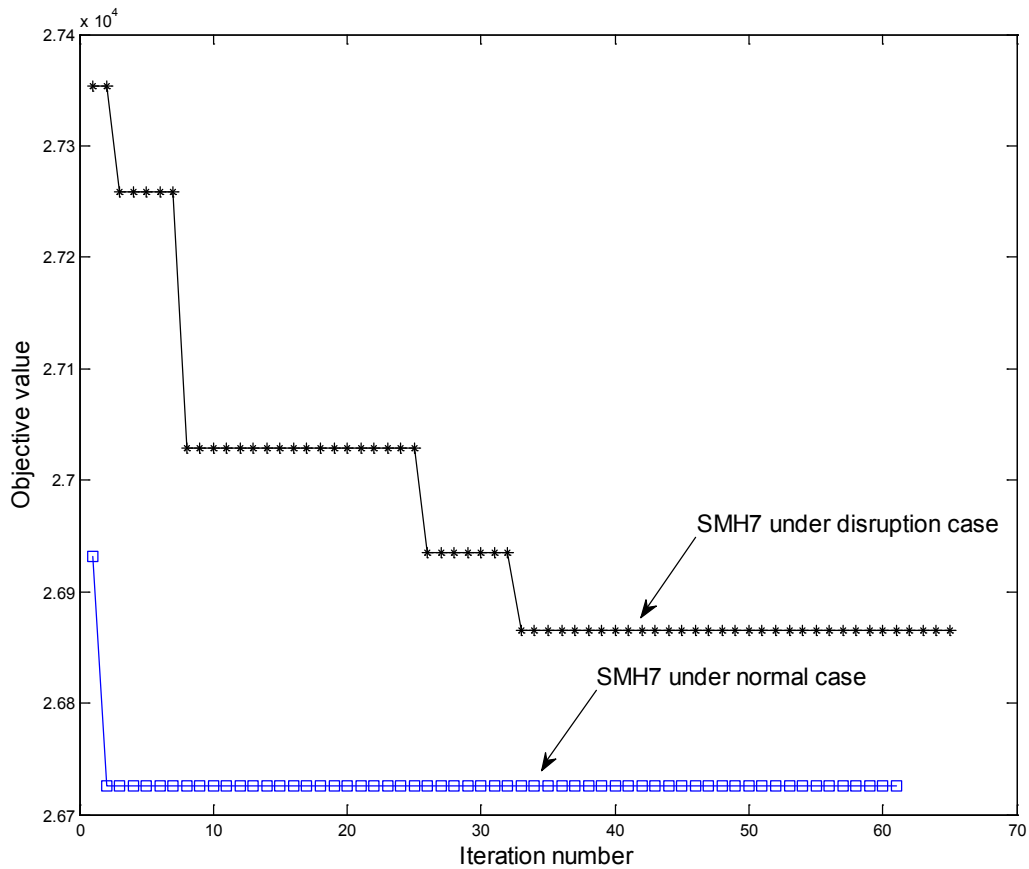


Figure 7.3. The results of SMH7 under normal and disruption case with $k = 20$

From the above three figures, one can conclude that although the initial value of the disruption case is not good enough, the converged value has displayed a great improvement. The reason is that the weak partition under the transportation matrix T is still valid for T^B . To prove this, construct a matrix $M = (M_{il})_{(k+1) \times (k+1)}$ such that M_{il} is the length of the shortest path between i and l in T^B . It is clear that the matrix M satisfies that $M_{il} \leq M_{ih} + M_{hl}$ and $M_{il} \geq T_{il}$. Therefore, replace T by M in the proof of Theorem 4.9, which implies the validity of the property for T^B . Hence, one can conclude that the algorithm SMH7 is resilient for the roads disruption case.

7.6 Concluding Remarks

In this chapter, the simulations were conducted to demonstrate the robustness and the resilience of the algorithms. From the simulation, one can conclude that algorithms SMH4 and SMH8 are robust under the uncertainty while algorithm SMH7 is resilient under the disruption. The conclusions are applicable to the other algorithms because of the similar structure of the problems and algorithms in this thesis. Therefore, all the algorithms developed in this thesis possess a good degree of the robustness and resilience. It is noted that the quantitative part of the robustness and resilience for algorithms has not been given in this chapter. In the last chapter of this thesis, the possible definition of the quantitative part of the robustness and resilience will be given as a future work.

CHAPTER 8

CASE STUDY: MEDICAL RESOURCES ALLOCATION

In this chapter, a case study is presented, which has the two purposes: (1) to illustrate how the algorithms developed can be applied to real world problems and (2) to demonstrate the effectiveness of the algorithms. The case is about applying medical resources allocation in emergency management (EM). In the following, the concept of EM is first introduced along with the related work in literature. Note that the problem of medical resources allocation is in itself very important in emergency management and there are quite an amount of studies on this topic from a perspective other than supply chain scheduling. Later, supply chain scheduling is applied to model the medical resources allocation problem based on several assumptions into a supply chain scheduling problem and the problem is then solved by the algorithms as developed in thesis and described in the previous chapters.

8.1 Emergency Management

The emergency events can be disasters of nature or human, such as earthquake, fire, flood, traffic accident and so on (see Figure 8.1). They may also be acute infectious diseases, such as Severe Acute Respiratory Syndrome (SARS), influenza A (H1N1) virus, Ebola virus disease (Ebola), and so on (see Figure 8.2). As it is impossible to completely eliminate such emergency events, the focus is on how to help people and reduce losses after the

occurrence of emergency events, which results in the research on EM.

For helping people in EM, the victims should be first evacuated from dangerous places (affected area where an emergency event takes places) to safe places (temporary aid sites). Further, the victims may be wounded or infected, which implies that medical resources are required to cure them. However, the existing work in the field of EM only considered the evacuation operations but little research is on the allocation of medical resources.



Figure 8.1. Great Disasters in Recent Years

(<http://www.canadianbusiness.com/companies-and-industries/flooding/>

http://www.boston.com/bigpicture/2010/11/shanghai_apartment_fire.html

<http://www.dailymail.co.uk/news/article-1365546/Japan-earthquake-pictures-Devastation-rescue-workers-fight-fires-search-survivors.html>

<http://o.canada.com/news/photos-50-80-car-accident-on-the-401/>)

After the wounded or infected victims have settled in temporary aid sites, they need the medical resources which can be drugs, medical devices and medical staff. Although the aid sites are safe, there are very limited medical resources to meet the demand. Therefore, the nearby medical centers should supply the medical resources to them.

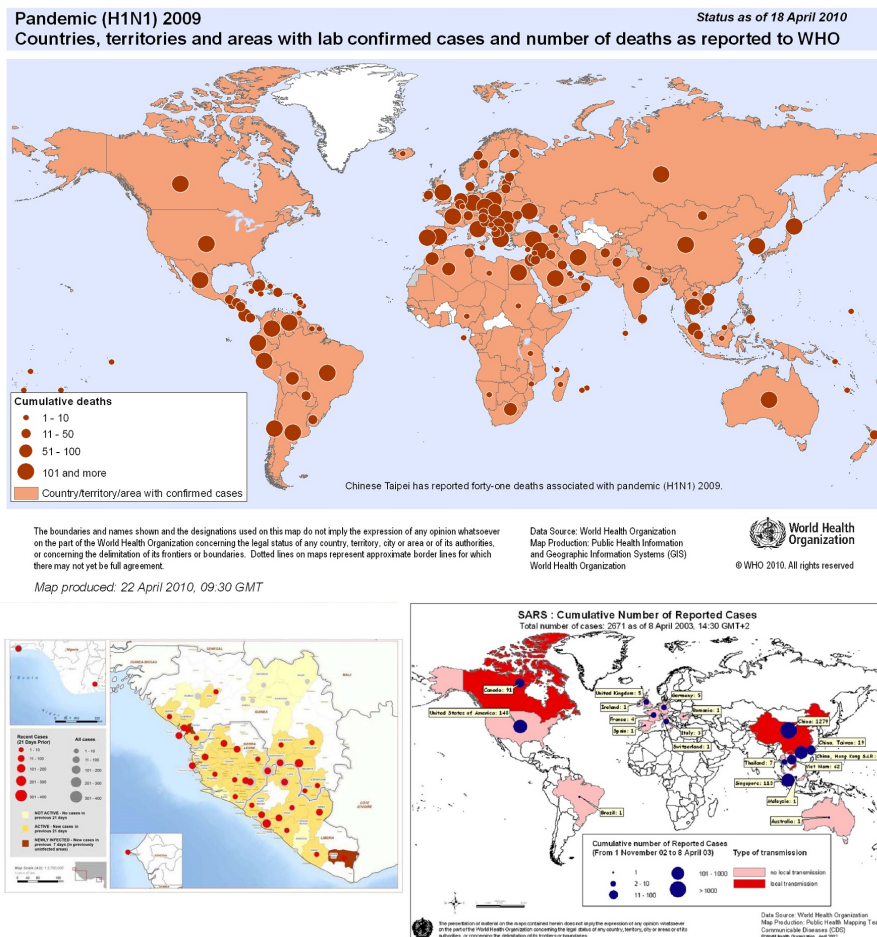


Figure 8.2. Infectious Diseases in Recent Years

(<http://www.lib.utexas.edu/maps/sars.html>)

<http://reliefweb.int/map/world/world-pandemic-h1n1-2009-countries>

-territories-and-areas-lab-confirmed-cases-and-number-11

http://www.un.org/apps/news/story.asp?NewsID=48668#.VR7Sf_nF_z0)

In the view of medical centers, they need to know the information of requirements before

supply. When large areas are affected by the disaster (earthquake, flood, or worldwide infectious diseases) or the connection between aid sites and outside is limited, the related data (the number of victims, the quantity and the type of drugs needed and so on) cannot be obtained in a short time but collected gradually. Obviously, the medical centers cannot wait for all requirements being known and then deliver all the needed resources. Therefore, the on-line mechanism should be considered and on-line decisions should be made accordingly.

After the information being known, there should be a period of time to prepare the required medical resources, e.g., to prescribe drugs, assemble medical devices and gather medical staff. The preparation is managed on processors, which involves different processor configurations. When the medical resources are prepared, vehicles will deliver them to the corresponding aid sites with different delivery patterns and characteristics of the vehicle.

Two performance measures are concerned in the medical resources allocation: time and cost. It is always desired that the whole process has short time and low cost. Similarly, these two objectives conflict with each other.

As the preparation-delivery mechanism is similar with the production-delivery mechanism of supply chain scheduling, this study will apply the supply chain scheduling model to describe the problem of allocating medical resources.

8.2 Evacuation Problems

In the existing work in EM, most researchers focus on evacuation problem but little about medical resources allocation. However, the evacuation problem is close to the medical resources allocation problem and there are some studies on the evacuation problem in literature.

The models of evacuation problems can be divided into two classes: macroscopic and microscopic models. The macroscopic model considers the victims as a homogeneous group where individual differences are ignored [Fahy, 1991; Burkard et al., 1993; Lin et al., 2008] while the microscopic model concerns the individual victims' movement and depends on simulation [Nagel and Schreckenberg, 1992; Lárraga et al., 2005; Lan et al., 2010].

Initially, the evacuation problems are considered as min-max flow problems in a static network. In a static network \mathcal{G} , nodes are used to represent source places, sink places and intermediate places while edges to represent the roads or paths connecting these places. By graph theory, the whole network can be transformed to a node-edge incidence matrix which is convenient for the algorithmic analysis. On source places, there are victims who need to be evacuated; while on intermediate places and sink places, there is a capacity limit for victims. Edges may also be characterized by the attributes such as flow capacity and travel speed. Table 8.1 includes some evacuation problems in the static network.

However, the static network min-max models cannot describe the evacuation problems

in reality because of ignoring the time factor. Thus, the dynamic network \mathcal{G}_T is brought in to model the emergency operations over time, where \mathcal{G}_T is the time expanded version of the static network \mathcal{G} and flows in it. There are two classes of the dynamic network models: discrete-time dynamic network and continuous-time dynamic network. In the continuous time dynamic network flow problems, researchers focus on the special cases with a constant travel time and flow capacity. Some examples of the dynamic network are also shown in Table 8.1.

8.3 Problem Descriptions for The Medical Resources Allocation

In this section, the supply chain scheduling model is applied to the problem of medical resources allocation based on several assumptions.

Suppose that a disaster takes place and victims have been settled to temporary aid sites. There are demands of medical resources to cure wounded/infected victims in the aid sites. However, the medical resources in these temporary aid sites are very limited and it is necessary to appeal to the nearby medical centers. In particular, the case that there are multiple medical centers is considered.

The aid sites need to inform medical centers of their demands of medical resources such as drugs, medical devices and medical staff. In the following, the term 'job' is used to represent the demands of medical resources. The medical resources are continuously required in the whole process of EM. When large areas are affected or the communication is impeded, the information cannot be known beforehand but gradually known during the process. Therefore, the on-line environments for scheduling in this case should be

considered.

Table 8.1. Evacuation Problems for Different Network Models

Static network	Shortest path [Fahy, 1991; Lim et al., 2012] Minimum cost [Yamada, 1996] Quickest path [Chen and Chin, 1990; Chen and Hung, 1993]
Discrete-time dynamic network	Shortest path [Hamacher et al., 2006; Bérubé et al., 2006] Minimum cost maximum flow [Köhler and Skutella, 2006; Dressler et al., 2010] Quickest flow [Baumann and Köhler, 2007] Universally quickest flow [Takizawa et al., 2012] Lexicographically minimal cost [Hamacher and Tufekci, 1987]
Continuous-time dynamic network	Maximum flow with time dependent capacity [Anderson et al., 1982; Philpott, 1990] Universally maximum flow with zero travel time [Ogier, 1988; Fleischer, 2001b] Quickest flow with constant capacity and travel time [Fleischer, 2001a] Maximum flow, Quickest flow, Universally quickest flow, Lexicographically maximum flow [Fleischer and Tardos, 1998]

After knowing the jobs, or the jobs being released, medical centers need time to prepare the jobs, such as prescribe drugs, assemble medical devices, and gather medical staff. This preparation needs to be executed by work resources called processors. After the jobs are prepared, the jobs are then delivered to aid sites by the vehicles through a transportation network.

If jobs preparation on processors is viewed as jobs processing on machines, the processing-delivery in supply chain scheduling can be extended directly to medical resources allocation. Then, the same time-based objective and cost-based objective can be defined for medical resources allocation. The following assumptions are made.

1. The relationship between aid sites and the medical center form an on-line environment.

Because of the well-developed technology of information and communication, the satellite signal can cover almost everywhere in the world [Schiller, 2003]. In particular, the experiment to transfer medical data from Mount Logan (Canada's highest summit) through the satellite was successfully conducted in in the late 90s [Otto and Pipe, 1997]. This implies that the aid sites can connect with the medical center. Although there may be limitation on the communication, it can still assume that the transmission of medical data is valid at some moments in every time interval as the satellites move around the earth periodically. Therefore, the on-line environment makes sense for this situation.

2. All jobs are homogenous.

This is similar with the assumption of the macroscopic model for the evacuation problem. As most of the medical resources (drugs, medical devices and medical staff) are regular, it is possible to consider that every job occupies the unit size of a vehicle in the delivery. Furthermore, only the release time and preparation time of a job are considered in the allocation process.

3. *The processor configurations are single-processor and multi-processor*

In reality, the preparation of jobs can have different patterns. Furthermore, this case study focuses on applying supply chain scheduling model for medical resources allocation problem. To capture the nature of the problem, the basic pattern of preparation is explored. Therefore, particular configurations of single-processor and parallel-processor are considered in this case study. For problems with other preparation patterns, the results of this case study can be meaningful and extendable. In the two configurations, the processors and the jobs are exclusive: one job is prepared by one processor in the center at a time and one processor prepares one job at a time.

Based on the above assumptions, the supply chain scheduling model can be applied to the medical resources allocation problem. The problem can be described as follows.

Suppose there are n_i jobs $J_1^{(i)}, \dots, J_{n_i}^{(i)}$ with the preparation time $p_1^{(i)}, \dots, p_{n_i}^{(i)}$, released at the time $r_1^{(i)}, \dots, r_{n_i}^{(i)}$ from the i th aid site ($i = 1, 2, \dots, k$), respectively, to s medical centers. The k aid sites and the s medical centers are located at different places and form a transportation network. Every medical center has processors to prepare the jobs without preemption. The job release is in an on-line environment, which means that the information of future jobs is not known until their release time. After jobs are prepared in the same medical center, they are divided into batches or shipments and then transported to the aid sites by vehicles. There is a delivery cost for delivering a batch. Jobs of different aid sites can be contained into one batch. The objective is to minimize the total makespans and the total delivery cost.

When there are multiple medical centers, the schedule needs to decide in which medical center that a job should be prepared besides jobs preparation on processors and batch delivery. Therefore, a policy that assign jobs to a certain medical center is implemented. When a job $J_j^{(i)}$ is assigned to a medical center, $J_j^{(i)}$ needs to be prepared in this medical center and then delivered to Aid Site i . Thus, the original problem is decomposed into several sub-problems with single medical center and the algorithms developed in the previous chapter regarding the single manufacturer and multiple customers can then be applied. In the following sections, details of the policy are presented and a simulated experiment is proposed to demonstrate the performance of the policy along with algorithms.

8.4 Policy to Assign Aid Sites

In this section, the policy to assign jobs to the corresponding medical centers is presented. The solution of the above problem is a schedule which should specify when a job is prepared, in which medical center a job is prepared, to which batch a job is assigned, when a batch is transported, and through which path a batch is transported. As there is more than one medical center, it is required to assign a job to a certain medical center. To deal with such a situation, a policy is proposed to complete the assignment. After that, the problem is decomposed into several sub-problems, which are actually problems SMP8, and SMH8 can be applied to solve them. Details of the policy are described as follows.

Policy AAS

Let MJ_z be the sets of all jobs assigned to the z th medical center and set $MJ_z = \emptyset$ ($z = 1, 2, \dots, s$). Let AS_z be the sets of aid sites that release the jobs in MJ_z and set

$AS_z = \emptyset$ ($z = 1, 2, \dots, s$). Set $H_z = 0$ for $z = 1, 2, \dots, s$.

When a new job $J_j^{(i)}$ is released, if $i \in AS_z$, $H_z = C_{max}(MJ_z \cup \{J_j^{(i)}\})$; if $i \notin AS_z$ and $AS_z \neq \emptyset$, $H_z = C_{max}(MJ_z \cup \{J_j^{(i)}\}) + 2T_{is}$; if $AS_z = \emptyset$, $H_z = C_{max}(MJ_z \cup \{J_j^{(i)}\}) + 2T_{is} + D$. Compute $z_0 = \operatorname{argmin}\{H_z | z = 1, 2, \dots, s\}$. Assign the job $J_j^{(i)}$ to Medical Center z_0 and $MJ_{z_0} = MJ_{z_0} \cup \{J_j^{(i)}\}$. If $i \notin AS_{z_0}$, $AS_{z_0} = AS_{z_0} \cup \{i\}$.

For the aid sites in AS_z and jobs in MJ_z ($z = 1, 2, \dots, s$), a schedule to prepare and deliver the jobs is constructed by SMH8. Then s schedule gives a solution of the original problem. Next, a case is built to show the performance of AAS and SMH8.

8.5 Simulated Experiment for Case Study

In this section, a simulation is conducted to show the performance of AAS and SMH8. An instance can be defined by prescribing a set of the foregoing parameters (n_i , $p_j^{(i)}$, and $r_j^{(i)}$, for $j = 1, 2, \dots, n_i$, T_{il} for $i, l = 1, 2, \dots, k, \dots, k+s$, and D). The instances are generated by these randomly generated parameters. The algorithm was implemented in the Matlab environment. The parameters are thus determined based on the following assumptions:

- (1) The release of jobs for Aid Site i follows the poisson distribution with the parameter λ_i , i.e., the number of jobs released at some time r : $n_i(r) \sim P(\lambda_i)$ and the next release time is $r + r'$, where $r' \sim U(0, \lambda_i)$, λ_i is two times of the mean value of the release intervals for Aid Site i , and $\lambda_i \sim U(0, \Lambda_i)$ ($i = 1, 2, \dots, k$).
- (2) The job preparation time for Aid Site i follows the uniform distribution in the interval $[0, b_i]$, i.e., $p_j^{(i)} \sim U(0, b_i)$ for $j = 1, 2, \dots, n_i$, where b_i is two times of the mean value of the preparation time for Aid Site i and $b_i \sim U(0, B_i)$ ($i = 1, 2, \dots, k$).

- (3) The number of jobs for Aid Site i follows the uniform distribution in the set $\{1, 2, \dots, N_i\}$, i.e., $Pr\{n_i = h\} = \frac{1}{N_i}$ for $h = 1, 2, \dots, N_i$ where N_i is two times of the mean value of the number of jobs for Aid Site i ($i = 1, 2, \dots, k$).
- (4) The positions of the medical centers and the aid sites are randomly located in an square area with the side length L , and the transportation network can be directly determined by the Euclidean distance.
- (5) The delivery cost D is a constant.
- (6) The number of aid sites is of one case: $k = 30$.
- (7) The number of medical centers is of one case: $s = 5$.

An instance with 30 aid sites and 5 medical centers is generated. The coordinates of the locations of the aid sites and the medical centers are as follows (see Figure 8.3). In Figure 8.3, x and y are length and width of the concerned area, which does not have a physical dimension but an unit. To a real application area, this unit will need to be multiplied by a ratio (e.g., inch per unit) to scale to a real length.

Aid Site 1:	(81.47,82.35)	Aid Site 2:	(90.58,69.48)	Aid Site 3:	(12.70,31.71)
Aid Site 4:	(91.34,95.02)	Aid Site 5:	(63.24,3.44)	Aid Site 6:	(9.75,43.87)
Aid Site 7:	(27.85,38.16)	Aid Site 8:	(54.69,76.55)	Aid Site 9:	(95.75,79.52)
Aid Site 10:	(96.49,18.69)	Aid Site 11:	(15.76,48.98)	Aid Site 12:	(97.06,44.56)
Aid Site 13:	(95.72,64.63)	Aid Site 14:	(48.54,70.94)	Aid Site 15:	(80.03,75.47)
Aid Site 16:	(14.19,27.60)	Aid Site 17:	(42.18,67.97)	Aid Site 18:	(91.57,65.51)
Aid Site 19:	(79.22,16.26)	Aid Site 20:	(95.95,11.90)	Aid Site 21:	(65.57,49.84)

Aid Site 22: (3.57,95.97) Aid Site 23: (84.91,34.04) Aid Site 24: (93.40,58.53)
 Aid Site 25: (67.87,22.38) Aid Site 26: (75.77,75.13) Aid Site 27: (74.31,25.51)
 Aid Site 28: (39.22,50.60) Aid Site 29: (65.55,69.91) Aid Site 30: (17.12,89.09)

Medical Center 1: (70.60,95.93) Medical Center 2: (31.83,54.72)
 Medical Center 3: (27.69,13.86) Medical Center 4: (4.62,14.93)
 Medical Center 5: (9.71,25.75)

The jobs information of every aid sites is as follows.

Aid Site 1 (13 jobs)

$J_1^{(1)}$ (0.86 , 4.60) $J_2^{(1)}$ (1.66 , 2.49) $J_3^{(1)}$ (4.47 , 6.26) $J_4^{(1)}$ (4.47 , 1.59)
 $J_5^{(1)}$ (8.76 , 5.77) $J_6^{(1)}$ (8.76 , 1.54) $J_7^{(1)}$ (8.76 , 3.10) $J_8^{(1)}$ (10.56 , 5.26)
 $J_9^{(1)}$ (12.12 , 6.56) $J_{10}^{(1)}$ (13.19 , 0.68) $J_{11}^{(1)}$ (13.19 , 7.81) $J_{12}^{(1)}$ (14.02 , 6.52)
 $J_{13}^{(1)}$ (15.40 , 4.09)

Aid Site 2 (13 jobs)

$J_1^{(2)}$ (0.87 , 4.91) $J_2^{(2)}$ (1.89 , 5.79) $J_3^{(2)}$ (1.89 , 1.81) $J_4^{(2)}$ (3.18 , 0.96)
 $J_5^{(2)}$ (3.18 , 2.42) $J_6^{(2)}$ (3.88 , 2.60) $J_7^{(2)}$ (3.88 , 3.45) $J_8^{(2)}$ (5.12 , 4.14)
 $J_9^{(2)}$ (6.19 , 0.70) $J_{10}^{(2)}$ (7.72 , 2.14) $J_{11}^{(2)}$ (8.09 , 6.52) $J_{12}^{(2)}$ (8.09 , 0.24)
 $J_{13}^{(2)}$ (9.13 , 7.56)

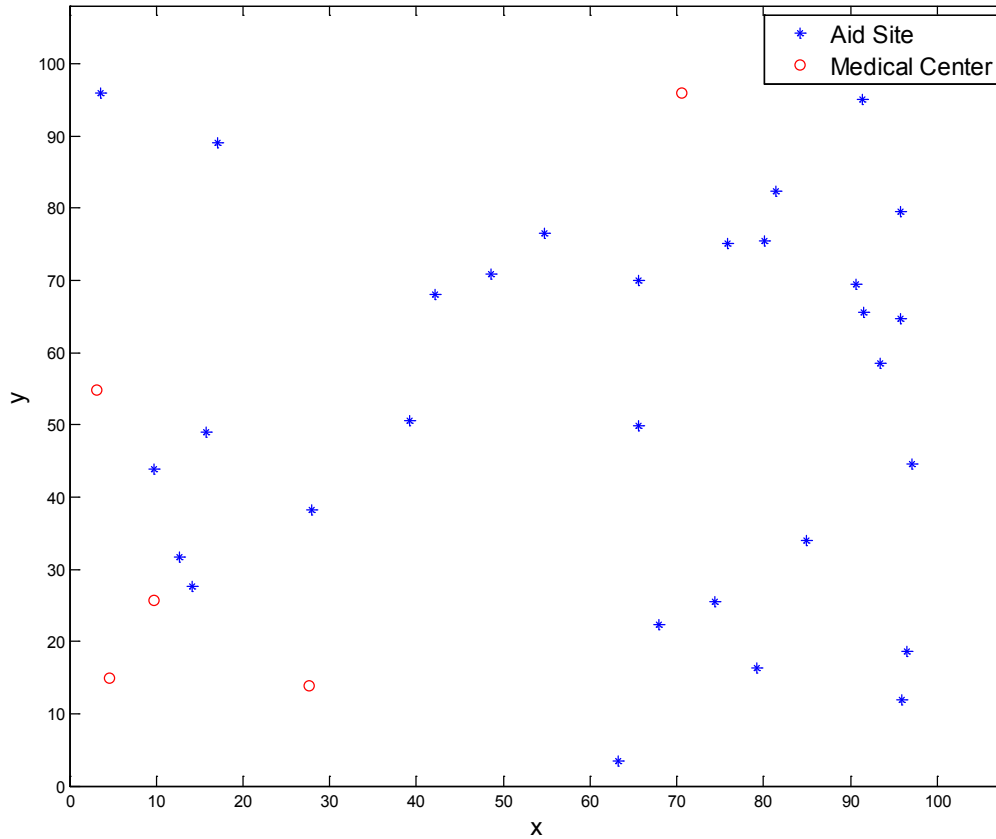


Figure 8.3. The Locations of 30 Aid Sites and 5 Medical Centers

Aid Site 3 (18 jobs)

$J_1^{(3)}$	(1.46 , 2.23)	$J_2^{(3)}$	(1.46 , 8.24)	$J_3^{(3)}$	(2.38 , 0.27)	$J_4^{(3)}$	(2.38 , 4.55)
$J_5^{(3)}$	(4.09 , 1.56)	$J_6^{(3)}$	(4.82 , 9.09)	$J_7^{(3)}$	(6.59 , 6.62)	$J_8^{(3)}$	(6.59 , 4.65)
$J_9^{(3)}$	(11.23 , 4.38)	$J_{10}^{(3)}$	(11.23 , 0.55)	$J_{11}^{(3)}$	(11.23 , 6.34)	$J_{12}^{(3)}$	(13.01 , 0.39)
$J_{13}^{(3)}$	(13.01 , 0.66)	$J_{14}^{(3)}$	(14.07 , 4.85)	$J_{15}^{(3)}$	(14.07 , 0.90)	$J_{16}^{(3)}$	(15.31 , 7.60)
$J_{17}^{(3)}$	(15.31 , 7.60)	$J_{18}^{(3)}$	(15.31 , 6.71)				

Aid Site 4 (13 jobs)

$J_1^{(4)}$	(1.34 , 1.38)	$J_2^{(4)}$	(1.34 , 1.31)	$J_3^{(4)}$	(1.34 , 1.06)	$J_4^{(4)}$	(2.20 , 1.37)
$J_5^{(4)}$	(3.25 , 1.31)	$J_6^{(4)}$	(3.25 , 0.35)	$J_7^{(4)}$	(4.05 , 0.25)	$J_8^{(4)}$	(5.36 , 1.96)
$J_9^{(4)}$	(8.19 , 0.34)	$J_{10}^{(4)}$	(11.82 , 0.06)	$J_{11}^{(4)}$	(13.30 , 1.10)	$J_{12}^{(4)}$	(14.39 , 1.73)
$J_{13}^{(4)}$	(14.39 , 1.32)						

Aid Site 5 (24 jobs)

$J_1^{(5)}$	(1.30 , 4.28)	$J_2^{(5)}$	(3.01 , 0.42)	$J_3^{(5)}$	(3.01 , 1.57)	$J_4^{(5)}$	(3.87 , 1.38)
$J_5^{(5)}$	(5.82 , 4.11)	$J_6^{(5)}$	(6.40 , 5.20)	$J_7^{(5)}$	(8.05 , 2.12)	$J_8^{(5)}$	(8.05 , 4.81)
$J_9^{(5)}$	(8.05 , 4.16)	$J_{10}^{(5)}$	(8.26 , 0.04)	$J_{11}^{(5)}$	(9.90 , 3.71)	$J_{12}^{(5)}$	(12.06 , 2.38)
$J_{13}^{(5)}$	(12.06 , 5.64)	$J_{14}^{(5)}$	(12.76 , 0.01)	$J_{15}^{(5)}$	(13.58 , 2.85)	$J_{16}^{(5)}$	(13.58 , 2.61)
$J_{17}^{(5)}$	(13.58 , 2.84)	$J_{18}^{(5)}$	(15.45 , 4.74)	$J_{19}^{(5)}$	(15.45 , 1.99)	$J_{20}^{(5)}$	(17.60 , 4.83)
$J_{21}^{(5)}$	(17.60 , 2.90)	$J_{22}^{(5)}$	(19.49 , 0.22)	$J_{23}^{(5)}$	(19.49 , 1.08)	$J_{24}^{(5)}$	(20.84 , 4.45)

Aid Site 6 (42 jobs)

$J_1^{(6)}$	(1.63 , 2.36)	$J_2^{(6)}$	(3.11 , 1.54)	$J_3^{(6)}$	(3.64 , 2.93)	$J_4^{(6)}$	(5.37 , 2.70)
$J_5^{(6)}$	(5.37 , 0.59)	$J_6^{(6)}$	(6.66 , 3.03)	$J_7^{(6)}$	(6.66 , 3.48)	$J_8^{(6)}$	(6.66 , 1.81)
$J_9^{(6)}$	(7.08 , 3.11)	$J_{10}^{(6)}$	(8.22 , 2.07)	$J_{11}^{(6)}$	(9.76 , 0.54)	$J_{12}^{(6)}$	(9.76 , 0.70)
$J_{13}^{(6)}$	(11.44 , 1.43)	$J_{14}^{(6)}$	(11.44 , 2.63)	$J_{15}^{(6)}$	(12.61 , 2.90)	$J_{16}^{(6)}$	(12.61 , 2.78)
$J_{17}^{(6)}$	(15.12 , 1.12)	$J_{18}^{(6)}$	(15.12 , 1.88)	$J_{19}^{(6)}$	(15.12 , 0.32)	$J_{20}^{(6)}$	(16.42 , 0.39)
$J_{21}^{(6)}$	(16.42 , 0.48)	$J_{22}^{(6)}$	(18.40 , 2.39)	$J_{23}^{(6)}$	(19.43 , 1.74)	$J_{24}^{(6)}$	(20.96 , 0.67)
$J_{25}^{(6)}$	(20.96 , 1.74)	$J_{26}^{(6)}$	(20.96 , 0.52)	$J_{27}^{(6)}$	(20.96 , 0.19)	$J_{28}^{(6)}$	(22.49 , 2.99)
$J_{29}^{(6)}$	(22.49 , 1.97)	$J_{30}^{(6)}$	(23.83 , 3.27)	$J_{31}^{(6)}$	(23.83 , 2.45)	$J_{32}^{(6)}$	(26.16 , 2.05)
$J_{33}^{(6)}$	(27.54 , 2.87)	$J_{34}^{(6)}$	(27.54 , 3.09)	$J_{35}^{(6)}$	(28.91 , 3.48)	$J_{36}^{(6)}$	(28.91 , 0.00)
$J_{37}^{(6)}$	(31.75 , 3.04)	$J_{38}^{(6)}$	(31.75 , 2.15)	$J_{39}^{(6)}$	(33.29 , 3.48)	$J_{40}^{(6)}$	(33.29 , 1.86)

$$J_{41}^{(6)} \quad (33.29 , 1.69) \quad J_{42}^{(6)} \quad (33.72 , 2.82)$$

Aid Site 7 (28 jobs)

$$\begin{array}{cccc}
 J_1^{(7)} & (2.26 , 2.61) & J_2^{(7)} & (2.26 , 0.32) & J_3^{(7)} & (2.26 , 1.04) & J_4^{(7)} & (2.26 , 3.88) \\
 J_5^{(7)} & (4.84 , 1.94) & J_6^{(7)} & (4.84 , 5.26) & J_7^{(7)} & (4.84 , 0.69) & J_8^{(7)} & (4.84 , 5.78) \\
 J_9^{(7)} & (4.84 , 3.16) & J_{10}^{(7)} & (6.70 , 4.14) & J_{11}^{(7)} & (6.94 , 5.85) & J_{12}^{(7)} & (8.63 , 1.68) \\
 J_{13}^{(7)} & (11.12 , 2.43) & J_{14}^{(7)} & (14.68 , 2.72) & J_{15}^{(7)} & (14.68 , 4.47) & J_{16}^{(7)} & (16.63 , 4.79) \\
 J_{17}^{(7)} & (16.63 , 0.59) & J_{18}^{(7)} & (17.76 , 1.04) & J_{19}^{(7)} & (18.17 , 2.10) & J_{20}^{(7)} & (21.73 , 0.33) \\
 J_{21}^{(7)} & (21.73 , 3.05) & J_{22}^{(7)} & (23.45 , 1.97) & J_{23}^{(7)} & (23.45 , 1.03) & J_{24}^{(7)} & (23.45 , 1.22) \\
 J_{25}^{(7)} & (23.45 , 5.30) & J_{26}^{(7)} & (23.45 , 3.95) & J_{27}^{(7)} & (28.56 , 2.74) & J_{28}^{(7)} & (28.56 , 5.34)
 \end{array}$$

Aid Site 8 (15 jobs)

$$\begin{array}{cccc}
 J_1^{(8)} & (1.68 , 0.67) & J_2^{(8)} & (8.10 , 0.81) & J_3^{(8)} & (8.91 , 7.32) & J_4^{(8)} & (11.12 , 8.65) \\
 J_5^{(8)} & (11.12 , 6.27) & J_6^{(8)} & (13.09 , 1.21) & J_7^{(8)} & (14.74 , 6.63) & J_8^{(8)} & (17.54 , 1.01) \\
 J_9^{(8)} & (17.54 , 1.08) & J_{10}^{(8)} & (19.10 , 5.88) & J_{11}^{(8)} & (20.21 , 3.02) & J_{12}^{(8)} & (21.77 , 6.00) \\
 J_{13}^{(8)} & (24.48 , 6.87) & J_{14}^{(8)} & (24.48 , 5.35) & J_{15}^{(8)} & (24.48 , 6.79) & &
 \end{array}$$

Aid Site 9 (38 jobs)

$$\begin{array}{cccc}
 J_1^{(9)} & (0.47 , 6.13) & J_2^{(9)} & (0.47 , 5.67) & J_3^{(9)} & (1.38 , 0.91) & J_4^{(9)} & (2.96 , 3.98) \\
 J_5^{(9)} & (5.93 , 2.47) & J_6^{(9)} & (10.76 , 4.14) & J_7^{(9)} & (10.76 , 3.02) & J_8^{(9)} & (12.15 , 3.14) \\
 J_9^{(9)} & (12.15 , 1.37) & J_{10}^{(9)} & (12.37 , 1.93) & J_{11}^{(9)} & (12.90 , 0.16) & J_{12}^{(9)} & (14.44 , 6.99) \\
 J_{13}^{(9)} & (15.38 , 4.95) & J_{14}^{(9)} & (17.04 , 7.06) & J_{15}^{(9)} & (18.63 , 1.24) & J_{16}^{(9)} & (18.63 , 6.97) \\
 J_{17}^{(9)} & (21.97 , 6.02) & J_{18}^{(9)} & (21.97 , 4.37) & J_{19}^{(9)} & (22.49 , 3.33) & J_{20}^{(9)} & (25.92 , 1.95) \\
 J_{21}^{(9)} & (28.08 , 5.69) & J_{22}^{(9)} & (28.08 , 1.73) & J_{23}^{(9)} & (31.60 , 0.49) & J_{24}^{(9)} & (31.60 , 5.81) \\
 J_{25}^{(9)} & (33.85 , 5.08) & J_{26}^{(9)} & (33.85 , 5.42) & J_{27}^{(9)} & (34.76 , 4.86) & J_{28}^{(9)} & (35.88 , 3.17)
 \end{array}$$

$$\begin{array}{cccc}
J_{29}^{(9)} & (37.42 , 2.96) & J_{30}^{(9)} & (37.42 , 6.18) & J_{31}^{(9)} & (37.42 , 2.40) & J_{32}^{(9)} & (37.90 , 6.17) \\
J_{33}^{(9)} & (39.38 , 5.97) & J_{34}^{(9)} & (40.74 , 6.45) & J_{35}^{(9)} & (40.94 , 3.83) & J_{36}^{(9)} & (43.06 , 4.81) \\
J_{37}^{(9)} & (43.06 , 7.20) & J_{38}^{(9)} & (44.96 , 3.36) & & & &
\end{array}$$

Aid Site 10 (29 jobs)

$$\begin{array}{cccc}
J_1^{(10)} & (1.38 , 2.22) & J_2^{(10)} & (1.38 , 1.95) & J_3^{(10)} & (2.69 , 0.31) & J_4^{(10)} & (5.27 , 2.74) \\
J_5^{(10)} & (5.27 , 3.79) & J_6^{(10)} & (8.32 , 1.35) & J_7^{(10)} & (8.32 , 3.70) & J_8^{(10)} & (11.13 , 1.32) \\
J_9^{(10)} & (14.21 , 3.37) & J_{10}^{(10)} & (14.21 , 1.73) & J_{11}^{(10)} & (14.21 , 1.57) & J_{12}^{(10)} & (14.21 , 0.83) \\
J_{13}^{(10)} & (15.36 , 0.48) & J_{14}^{(10)} & (16.31 , 1.18) & J_{15}^{(10)} & (16.31 , 2.76) & J_{16}^{(10)} & (18.99 , 2.98) \\
J_{17}^{(10)} & (22.24 , 2.64) & J_{18}^{(10)} & (22.24 , 0.04) & J_{19}^{(10)} & (22.24 , 3.21) & J_{20}^{(10)} & (22.24 , 3.51) \\
J_{21}^{(10)} & (24.97 , 2.93) & J_{22}^{(10)} & (26.94 , 0.16) & J_{23}^{(10)} & (26.94 , 1.44) & J_{24}^{(10)} & (26.94 , 2.68) \\
J_{25}^{(10)} & (26.94 , 2.78) & J_{26}^{(10)} & (26.94 , 0.85) & J_{27}^{(10)} & (28.79 , 1.02) & J_{28}^{(10)} & (29.99 , 2.56) \\
J_{29}^{(10)} & (29.99 , 1.82) & & & & & &
\end{array}$$

Aid Site 11 (3 jobs)

$$J_1^{(11)} \quad (1.60 , 0.30) \quad J_2^{(11)} \quad (8.24 , 0.04) \quad J_3^{(11)} \quad (9.94 , 0.38)$$

Aid Site 12 (39 jobs)

$$\begin{array}{cccc}
J_1^{(12)} & (0.86 , 5.23) & J_2^{(12)} & (0.86 , 3.80) & J_3^{(12)} & (3.43 , 4.45) & J_4^{(12)} & (3.43 , 2.30) \\
J_5^{(12)} & (3.43 , 2.50) & J_6^{(12)} & (4.12 , 2.98) & J_7^{(12)} & (5.58 , 1.43) & J_8^{(12)} & (5.58 , 3.98) \\
J_9^{(12)} & (7.08 , 2.67) & J_{10}^{(12)} & (7.08 , 3.43) & J_{11}^{(12)} & (9.04 , 1.63) & J_{12}^{(12)} & (9.04 , 0.74) \\
J_{13}^{(12)} & (9.76 , 2.52) & J_{14}^{(12)} & (10.74 , 1.92) & J_{15}^{(12)} & (10.74 , 4.18) & J_{16}^{(12)} & (12.42 , 4.14) \\
J_{17}^{(12)} & (12.42 , 3.55) & J_{18}^{(12)} & (13.13 , 0.71) & J_{19}^{(12)} & (13.59 , 0.11) & J_{20}^{(12)} & (14.78 , 2.97) \\
J_{21}^{(12)} & (15.95 , 1.60) & J_{22}^{(12)} & (16.52 , 4.99) & J_{23}^{(12)} & (19.37 , 5.21) & J_{24}^{(12)} & (20.38 , 1.52) \\
J_{25}^{(12)} & (20.38 , 4.25) & J_{26}^{(12)} & (21.27 , 4.76) & J_{27}^{(12)} & (23.22 , 3.17) & J_{28}^{(12)} & (24.39 , 4.69)
\end{array}$$

$$\begin{array}{cccc}
J_{29}^{(12)} & (25.63 , 5.01) & J_{30}^{(12)} & (25.80 , 2.91) & J_{31}^{(12)} & (27.57 , 3.87) & J_{32}^{(12)} & (29.43 , 3.06) \\
J_{33}^{(12)} & (30.12 , 0.14) & J_{34}^{(12)} & (30.12 , 2.37) & J_{35}^{(12)} & (31.49 , 3.43) & J_{36}^{(12)} & (32.49 , 2.77) \\
J_{37}^{(12)} & (34.24 , 1.98) & J_{38}^{(12)} & (36.09 , 4.97) & J_{39}^{(12)} & (36.09 , 4.40) & &
\end{array}$$

Aid Site 13 (7 jobs)

$$\begin{array}{cccc}
J_1^{(13)} & (1.70 , 4.25) & J_2^{(13)} & (1.70 , 2.30) & J_3^{(13)} & (1.70 , 7.33) & J_4^{(13)} & (5.16 , 8.25) \\
J_5^{(13)} & (5.16 , 8.53) & J_6^{(13)} & (5.16 , 5.21) & J_7^{(13)} & (6.67 , 5.59) & &
\end{array}$$

Aid Site 14 (24 jobs)

$$\begin{array}{cccc}
J_1^{(14)} & (0.30 , 3.49) & J_2^{(14)} & (0.71 , 4.66) & J_3^{(14)} & (0.71 , 5.04) & J_4^{(14)} & (1.28 , 5.30) \\
J_5^{(14)} & (2.08 , 1.09) & J_6^{(14)} & (3.87 , 1.47) & J_7^{(14)} & (3.87 , 5.11) & J_8^{(14)} & (5.26 , 3.38) \\
J_9^{(14)} & (5.26 , 2.87) & J_{10}^{(14)} & (5.26 , 3.49) & J_{11}^{(14)} & (6.17 , 4.66) & J_{12}^{(14)} & (6.17 , 3.03) \\
J_{13}^{(14)} & (6.17 , 1.15) & J_{14}^{(14)} & (7.52 , 2.58) & J_{15}^{(14)} & (7.75 , 2.43) & J_{16}^{(14)} & (9.00 , 5.50) \\
J_{17}^{(14)} & (10.82 , 3.53) & J_{18}^{(14)} & (10.82 , 3.96) & J_{19}^{(14)} & (10.82 , 4.10) & J_{20}^{(14)} & (16.63 , 1.97) \\
J_{21}^{(14)} & (16.63 , 2.94) & J_{22}^{(14)} & (16.63 , 3.17) & J_{23}^{(14)} & (16.63 , 0.89) & J_{24}^{(14)} & (20.92 , 3.20)
\end{array}$$

Aid Site 15 (17 jobs)

$$\begin{array}{cccc}
J_1^{(15)} & (1.39 , 0.01) & J_2^{(15)} & (1.39 , 0.06) & J_3^{(15)} & (1.39 , 0.07) & J_4^{(15)} & (3.85 , 0.09) \\
J_5^{(15)} & (3.85 , 0.01) & J_6^{(15)} & (3.85 , 0.08) & J_7^{(15)} & (5.42 , 0.06) & J_8^{(15)} & (6.20 , 0.02) \\
J_9^{(15)} & (11.03 , 0.11) & J_{10}^{(15)} & (13.52 , 0.07) & J_{11}^{(15)} & (15.33 , 0.05) & J_{12}^{(15)} & (15.33 , 0.11) \\
J_{13}^{(15)} & (15.33 , 0.08) & J_{14}^{(15)} & (20.84 , 0.05) & J_{15}^{(15)} & (20.84 , 0.10) & J_{16}^{(15)} & (20.84 , 0.06) \\
J_{17}^{(15)} & (20.84 , 0.07) & & & & & &
\end{array}$$

Aid Site 16 (40 jobs)

$$\begin{array}{cccc}
J_1^{(16)} & (1.36 , 0.73) & J_2^{(16)} & (2.52 , 0.92) & J_3^{(16)} & (2.52 , 0.10) & J_4^{(16)} & (4.21 , 0.80) \\
J_5^{(16)} & (7.80 , 1.04) & J_6^{(16)} & (13.59 , 0.36) & J_7^{(16)} & (13.59 , 1.36) & J_8^{(16)} & (14.34 , 1.57)
\end{array}$$

$J_9^{(16)}$	(16.74 , 1.37)	$J_{10}^{(16)}$	(16.74 , 0.82)	$J_{11}^{(16)}$	(16.74 , 0.45)	$J_{12}^{(16)}$	(18.10 , 1.21)
$J_{13}^{(16)}$	(18.10 , 0.38)	$J_{14}^{(16)}$	(18.53 , 1.55)	$J_{15}^{(16)}$	(19.27 , 1.01)	$J_{16}^{(16)}$	(19.27 , 0.97)
$J_{17}^{(16)}$	(19.91 , 0.28)	$J_{18}^{(16)}$	(19.91 , 0.15)	$J_{19}^{(16)}$	(21.09 , 0.41)	$J_{20}^{(16)}$	(21.98 , 1.39)
$J_{21}^{(16)}$	(23.04 , 1.48)	$J_{22}^{(16)}$	(27.35 , 1.13)	$J_{23}^{(16)}$	(27.35 , 1.18)	$J_{24}^{(16)}$	(30.93 , 0.37)
$J_{25}^{(16)}$	(30.93 , 0.93)	$J_{26}^{(16)}$	(33.03 , 1.31)	$J_{27}^{(16)}$	(33.03 , 0.65)	$J_{28}^{(16)}$	(33.03 , 1.60)
$J_{29}^{(16)}$	(35.61 , 0.15)	$J_{30}^{(16)}$	(35.61 , 0.52)	$J_{31}^{(16)}$	(36.19 , 0.83)	$J_{32}^{(16)}$	(36.19 , 0.10)
$J_{33}^{(16)}$	(37.91 , 1.18)	$J_{34}^{(16)}$	(38.82 , 0.90)	$J_{35}^{(16)}$	(40.25 , 0.86)	$J_{36}^{(16)}$	(40.25 , 1.35)
$J_{37}^{(16)}$	(41.53 , 1.39)	$J_{38}^{(16)}$	(42.72 , 1.28)	$J_{39}^{(16)}$	(44.97 , 0.52)	$J_{40}^{(16)}$	(49.27 , 0.73)

Aid Site 17 (27 jobs)

$J_1^{(17)}$	(2.77 , 1.26)	$J_2^{(17)}$	(2.77 , 2.91)	$J_3^{(17)}$	(3.36 , 1.49)	$J_4^{(17)}$	(3.36 , 0.72)
$J_5^{(17)}$	(5.54 , 1.23)	$J_6^{(17)}$	(5.54 , 2.19)	$J_7^{(17)}$	(10.91 , 1.74)	$J_8^{(17)}$	(12.89 , 2.35)
$J_9^{(17)}$	(15.00 , 3.10)	$J_{10}^{(17)}$	(17.59 , 3.00)	$J_{11}^{(17)}$	(19.01 , 1.67)	$J_{12}^{(17)}$	(19.01 , 3.00)
$J_{13}^{(17)}$	(22.13 , 0.36)	$J_{14}^{(17)}$	(22.66 , 0.16)	$J_{15}^{(17)}$	(22.66 , 0.95)	$J_{16}^{(17)}$	(26.21 , 1.81)
$J_{17}^{(17)}$	(26.21 , 1.65)	$J_{18}^{(17)}$	(26.21 , 2.80)	$J_{19}^{(17)}$	(27.83 , 1.68)	$J_{20}^{(17)}$	(29.01 , 1.34)
$J_{21}^{(17)}$	(29.87 , 1.69)	$J_{22}^{(17)}$	(31.77 , 2.22)	$J_{23}^{(17)}$	(31.77 , 0.05)	$J_{24}^{(17)}$	(32.76 , 2.49)
$J_{25}^{(17)}$	(32.76 , 0.44)	$J_{26}^{(17)}$	(32.76 , 1.49)	$J_{27}^{(17)}$	(32.76 , 0.80)		

Aid Site 18 (31 jobs)

$J_1^{(18)}$	(0.74 , 1.25)	$J_2^{(19)}$	(3.45 , 0.57)	$J_3^{(20)}$	(3.45 , 0.69)	$J_4^{(21)}$	(5.38 , 0.26)
$J_5^{(22)}$	(5.38 , 1.36)	$J_6^{(23)}$	(5.38 , 1.04)	$J_7^{(24)}$	(7.14 , 1.22)	$J_8^{(25)}$	(7.14 , 1.33)
$J_9^{(26)}$	(7.14 , 0.11)	$J_{10}^{(27)}$	(7.14 , 1.57)	$J_{11}^{(28)}$	(7.81 , 0.82)	$J_{12}^{(29)}$	(7.81 , 1.25)
$J_{13}^{(30)}$	(8.56 , 1.23)	$J_{14}^{(31)}$	(8.56 , 1.38)	$J_{15}^{(32)}$	(12.97 , 0.26)	$J_{16}^{(33)}$	(16.81 , 0.76)
$J_{17}^{(34)}$	(16.81 , 1.02)	$J_{18}^{(35)}$	(16.81 , 1.54)	$J_{19}^{(36)}$	(18.79 , 1.38)	$J_{20}^{(37)}$	(20.07 , 1.48)

$J_{21}^{(38)}$	(21.50 , 0.96)	$J_{22}^{(39)}$	(21.50 , 0.97)	$J_{23}^{(40)}$	(23.55 , 1.42)	$J_{24}^{(41)}$	(25.16 , 0.06)
$J_{25}^{(42)}$	(25.16 , 1.47)	$J_{26}^{(43)}$	(27.65 , 0.68)	$J_{27}^{(44)}$	(27.65 , 0.06)	$J_{28}^{(45)}$	(30.92 , 1.24)
$J_{29}^{(46)}$	(32.31 , 0.26)	$J_{30}^{(47)}$	(32.31 , 0.24)	$J_{31}^{(48)}$	(33.68 , 1.00)		

Aid Site 19 (33 jobs)

$J_1^{(19)}$	(2.08 , 0.59)	$J_2^{(19)}$	(3.11 , 0.82)	$J_3^{(19)}$	(4.59 , 1.54)	$J_4^{(19)}$	(4.59 , 2.18)
$J_5^{(19)}$	(4.59 , 0.76)	$J_6^{(19)}$	(5.86 , 1.06)	$J_7^{(19)}$	(5.86 , 2.27)	$J_8^{(19)}$	(6.38 , 1.62)
$J_9^{(19)}$	(8.37 , 2.61)	$J_{10}^{(19)}$	(8.37 , 0.54)	$J_{11}^{(19)}$	(8.37 , 2.18)	$J_{12}^{(19)}$	(10.94 , 1.78)
$J_{13}^{(19)}$	(10.94 , 0.65)	$J_{14}^{(19)}$	(10.94 , 1.25)	$J_{15}^{(19)}$	(10.94 , 1.05)	$J_{16}^{(19)}$	(13.51 , 1.58)
$J_{17}^{(19)}$	(13.51 , 2.11)	$J_{18}^{(19)}$	(15.17 , 0.28)	$J_{19}^{(19)}$	(15.17 , 2.16)	$J_{20}^{(19)}$	(15.17 , 2.21)
$J_{21}^{(19)}$	(16.26 , 0.93)	$J_{22}^{(19)}$	(17.86 , 1.13)	$J_{23}^{(19)}$	(20.33 , 1.50)	$J_{24}^{(19)}$	(20.33 , 1.84)
$J_{25}^{(19)}$	(21.82 , 1.95)	$J_{26}^{(19)}$	(28.50 , 1.99)	$J_{27}^{(19)}$	(29.85 , 1.02)	$J_{28}^{(19)}$	(29.85 , 1.13)
$J_{29}^{(19)}$	(30.73 , 2.51)	$J_{30}^{(19)}$	(31.95 , 1.51)	$J_{31}^{(19)}$	(33.81 , 2.23)	$J_{32}^{(19)}$	(33.81 , 0.73)
$J_{33}^{(19)}$	(33.81 , 1.64)						

Aid Site 20 (38 jobs)

$J_1^{(20)}$	(1.18 , 1.51)	$J_2^{(21)}$	(2.18 , 3.17)	$J_3^{(22)}$	(3.99 , 6.61)	$J_4^{(23)}$	(3.99 , 5.45)
$J_5^{(24)}$	(3.99 , 3.11)	$J_6^{(25)}$	(6.12 , 2.30)	$J_7^{(26)}$	(7.40 , 0.41)	$J_8^{(27)}$	(7.40 , 5.11)
$J_9^{(28)}$	(7.88 , 3.49)	$J_{10}^{(29)}$	(9.82 , 1.38)	$J_{11}^{(30)}$	(9.82 , 2.94)	$J_{12}^{(31)}$	(9.82 , 1.16)
$J_{13}^{(32)}$	(9.82 , 5.18)	$J_{14}^{(33)}$	(11.79 , 2.54)	$J_{15}^{(34)}$	(11.79 , 6.49)	$J_{16}^{(35)}$	(11.79 , 0.12)
$J_{17}^{(36)}$	(11.79 , 5.71)	$J_{18}^{(37)}$	(13.57 , 4.32)	$J_{19}^{(38)}$	(15.55 , 3.71)	$J_{20}^{(39)}$	(15.55 , 4.48)
$J_{21}^{(40)}$	(15.55 , 5.01)	$J_{22}^{(41)}$	(15.55 , 0.65)	$J_{23}^{(42)}$	(15.55 , 6.05)	$J_{24}^{(43)}$	(16.76 , 0.10)
$J_{25}^{(44)}$	(16.76 , 2.03)	$J_{26}^{(45)}$	(16.76 , 1.24)	$J_{27}^{(46)}$	(17.28 , 6.38)	$J_{28}^{(47)}$	(20.16 , 0.47)
$J_{29}^{(48)}$	(20.84 , 4.00)	$J_{30}^{(49)}$	(23.15 , 4.39)	$J_{31}^{(50)}$	(23.15 , 4.49)	$J_{32}^{(51)}$	(23.15 , 5.96)

$$J_{33}^{(52)} (23.38 , 0.39) \quad J_{34}^{(53)} (23.38 , 5.63) \quad J_{35}^{(54)} (26.74 , 3.65) \quad J_{36}^{(55)} (26.74 , 4.79)$$

$$J_{37}^{(56)} (28.54 , 1.46) \quad J_{38}^{(57)} (30.15 , 3.74)$$

Aid Site 21 (5 jobs)

$$J_1^{(21)} (1.41 , 4.30) \quad J_2^{(21)} (1.41 , 0.33) \quad J_3^{(21)} (3.11 , 0.93) \quad J_4^{(21)} (3.11 , 3.49)$$

$$J_5^{(21)} (3.11 , 4.12)$$

Aid Site 22 (46 jobs)

$$J_1^{(22)} (1.57 , 0.25) \quad J_2^{(22)} (3.26 , 0.68) \quad J_3^{(22)} (3.26 , 0.91) \quad J_4^{(22)} (3.90 , 0.96)$$

$$J_5^{(22)} (3.90 , 0.71) \quad J_6^{(22)} (6.44 , 1.59) \quad J_7^{(22)} (6.44 , 0.21) \quad J_8^{(22)} (8.73 , 0.92)$$

$$J_9^{(22)} (8.73 , 0.68) \quad J_{10}^{(22)} (8.73 , 0.70) \quad J_{11}^{(22)} (12.24 , 0.24) \quad J_{12}^{(22)} (12.24 , 1.36)$$

$$J_{13}^{(22)} (12.24 , 0.65) \quad J_{14}^{(22)} (12.24 , 0.36) \quad J_{15}^{(22)} (12.80 , 0.00) \quad J_{16}^{(22)} (13.45 , 0.65)$$

$$J_{17}^{(22)} (16.57 , 1.26) \quad J_{18}^{(22)} (17.80 , 1.99) \quad J_{19}^{(22)} (17.80 , 0.10) \quad J_{20}^{(22)} (20.51 , 2.07)$$

$$J_{21}^{(22)} (20.51 , 0.30) \quad J_{22}^{(22)} (21.67 , 1.91) \quad J_{23}^{(22)} (23.14 , 1.83) \quad J_{24}^{(22)} (24.74 , 2.10)$$

$$J_{25}^{(22)} (27.14 , 0.31) \quad J_{26}^{(22)} (28.60 , 1.16) \quad J_{27}^{(22)} (28.60 , 0.93) \quad J_{28}^{(22)} (32.15 , 0.40)$$

$$J_{29}^{(22)} (33.09 , 1.32) \quad J_{30}^{(22)} (34.60 , 1.39) \quad J_{31}^{(22)} (34.60 , 0.49) \quad J_{32}^{(22)} (36.10 , 1.19)$$

$$J_{33}^{(22)} (36.10 , 2.27) \quad J_{34}^{(22)} (36.10 , 1.12) \quad J_{35}^{(22)} (36.10 , 1.59) \quad J_{36}^{(22)} (37.17 , 0.94)$$

$$J_{37}^{(22)} (37.67 , 0.08) \quad J_{38}^{(22)} (38.27 , 0.67) \quad J_{39}^{(22)} (39.54 , 1.84) \quad J_{40}^{(22)} (39.54 , 0.79)$$

$$J_{41}^{(22)} (41.13 , 0.19) \quad J_{42}^{(22)} (43.03 , 1.17) \quad J_{43}^{(22)} (43.03 , 0.84) \quad J_{44}^{(22)} (44.34 , 1.69)$$

$$J_{45}^{(22)} (44.34 , 1.20) \quad J_{46}^{(22)} (44.34 , 1.84)$$

Aid Site 23 (42 jobs)

$$J_1^{(23)} (1.88 , 0.43) \quad J_2^{(23)} (2.94 , 0.68) \quad J_3^{(23)} (4.47 , 0.90) \quad J_4^{(23)} (4.47 , 1.34)$$

$$J_5^{(23)} (5.93 , 0.71) \quad J_6^{(23)} (5.93 , 0.67) \quad J_7^{(23)} (7.85 , 1.14) \quad J_8^{(23)} (9.22 , 0.71)$$

$$J_9^{(23)} (9.22 , 1.31) \quad J_{10}^{(23)} (11.18 , 0.71) \quad J_{11}^{(23)} (11.18 , 0.76) \quad J_{12}^{(23)} (13.62 , 0.74)$$

$J_{13}^{(23)}$	(14.44 , 0.35)	$J_{14}^{(23)}$	(18.61 , 0.13)	$J_{15}^{(23)}$	(19.97 , 0.10)	$J_{16}^{(23)}$	(19.97 , 1.35)
$J_{17}^{(23)}$	(19.97 , 0.36)	$J_{18}^{(23)}$	(23.65 , 1.31)	$J_{19}^{(23)}$	(24.51 , 1.08)	$J_{20}^{(23)}$	(24.87 , 1.33)
$J_{21}^{(23)}$	(26.93 , 1.43)	$J_{22}^{(23)}$	(28.62 , 0.21)	$J_{23}^{(23)}$	(30.34 , 0.60)	$J_{24}^{(23)}$	(32.96 , 1.49)
$J_{25}^{(23)}$	(35.59 , 0.98)	$J_{26}^{(23)}$	(37.43 , 1.37)	$J_{27}^{(23)}$	(37.43 , 0.73)	$J_{28}^{(23)}$	(37.79 , 0.02)
$J_{29}^{(23)}$	(39.35 , 0.95)	$J_{30}^{(23)}$	(39.35 , 0.35)	$J_{31}^{(23)}$	(41.06 , 0.80)	$J_{32}^{(23)}$	(41.06 , 1.10)
$J_{33}^{(23)}$	(42.47 , 0.93)	$J_{34}^{(23)}$	(44.24 , 0.90)	$J_{35}^{(23)}$	(44.24 , 0.66)	$J_{36}^{(23)}$	(44.24 , 0.37)
$J_{37}^{(23)}$	(44.24 , 0.65)	$J_{38}^{(23)}$	(46.70 , 0.02)	$J_{39}^{(23)}$	(47.38 , 0.93)	$J_{40}^{(23)}$	(49.37 , 1.46)
$J_{41}^{(23)}$	(49.37 , 0.15)	$J_{42}^{(23)}$	(50.27 , 0.05)				

Aid Site 24 (50 jobs)

$J_1^{(24)}$	(1.77 , 2.55)	$J_2^{(24)}$	(5.87 , 4.50)	$J_3^{(24)}$	(7.20 , 2.53)	$J_4^{(24)}$	(7.20 , 2.23)
$J_5^{(24)}$	(8.71 , 2.71)	$J_6^{(24)}$	(14.04 , 0.68)	$J_7^{(24)}$	(15.13 , 0.71)	$J_8^{(24)}$	(16.74 , 4.69)
$J_9^{(24)}$	(16.74 , 3.25)	$J_{10}^{(24)}$	(16.74 , 1.43)	$J_{11}^{(24)}$	(17.31 , 4.66)	$J_{12}^{(24)}$	(19.25 , 0.31)
$J_{13}^{(24)}$	(19.25 , 2.46)	$J_{14}^{(24)}$	(19.77 , 3.89)	$J_{15}^{(24)}$	(24.91 , 1.82)	$J_{16}^{(24)}$	(24.91 , 2.16)
$J_{17}^{(24)}$	(25.27 , 2.84)	$J_{18}^{(24)}$	(27.25 , 4.81)	$J_{19}^{(24)}$	(27.25 , 4.19)	$J_{20}^{(24)}$	(27.69 , 0.37)
$J_{21}^{(24)}$	(28.24 , 1.50)	$J_{22}^{(24)}$	(33.16 , 2.04)	$J_{23}^{(24)}$	(35.86 , 4.65)	$J_{24}^{(24)}$	(35.86 , 2.26)
$J_{25}^{(24)}$	(35.86 , 1.29)	$J_{26}^{(24)}$	(35.86 , 3.22)	$J_{27}^{(24)}$	(35.86 , 2.58)	$J_{28}^{(24)}$	(38.30 , 4.84)
$J_{29}^{(24)}$	(38.30 , 5.03)	$J_{30}^{(24)}$	(39.66 , 4.40)	$J_{31}^{(24)}$	(40.81 , 3.82)	$J_{32}^{(24)}$	(40.81 , 4.00)
$J_{33}^{(24)}$	(41.63 , 4.84)	$J_{34}^{(24)}$	(43.61 , 0.35)	$J_{35}^{(24)}$	(43.61 , 1.81)	$J_{36}^{(24)}$	(43.61 , 0.02)
$J_{37}^{(24)}$	(43.61 , 4.46)	$J_{38}^{(24)}$	(43.61 , 2.73)	$J_{39}^{(24)}$	(43.61 , 1.97)	$J_{40}^{(24)}$	(43.61 , 1.22)
$J_{41}^{(24)}$	(44.37 , 2.88)	$J_{42}^{(24)}$	(45.34 , 1.56)	$J_{43}^{(24)}$	(46.66 , 0.37)	$J_{44}^{(24)}$	(49.50 , 0.46)
$J_{45}^{(24)}$	(49.50 , 0.37)	$J_{46}^{(24)}$	(50.59 , 2.21)	$J_{47}^{(24)}$	(50.59 , 0.66)	$J_{48}^{(24)}$	(50.59 , 2.38)
$J_{49}^{(24)}$	(51.68 , 4.84)	$J_{50}^{(24)}$	(53.40 , 1.90)				

Aid Site 25 (23 jobs)

$J_1^{(25)}$	(1.99 , 0.77)	$J_2^{(25)}$	(3.76 , 0.72)	$J_3^{(25)}$	(3.76 , 0.44)	$J_4^{(25)}$	(4.90 , 0.34)
$J_5^{(25)}$	(4.90 , 0.26)	$J_6^{(25)}$	(4.90 , 0.56)	$J_7^{(25)}$	(4.90 , 0.01)	$J_8^{(25)}$	(6.61 , 0.29)
$J_9^{(25)}$	(7.91 , 0.72)	$J_{10}^{(25)}$	(7.91 , 0.43)	$J_{11}^{(25)}$	(8.89 , 0.37)	$J_{12}^{(25)}$	(8.89 , 0.39)
$J_{13}^{(25)}$	(10.49 , 0.24)	$J_{14}^{(25)}$	(10.49 , 0.74)	$J_{15}^{(25)}$	(12.02 , 0.77)	$J_{16}^{(25)}$	(12.02 , 0.40)
$J_{17}^{(25)}$	(13.01 , 0.78)	$J_{18}^{(25)}$	(13.01 , 0.36)	$J_{19}^{(25)}$	(13.97 , 0.33)	$J_{20}^{(25)}$	(14.18 , 0.17)
$J_{21}^{(25)}$	(16.18 , 0.15)	$J_{22}^{(25)}$	(17.27 , 0.65)	$J_{23}^{(25)}$	(17.27 , 0.57)		

Aid Site 26 (49 jobs)

$J_1^{(26)}$	(1.06 , 0.42)	$J_2^{(26)}$	(1.06 , 0.74)	$J_3^{(26)}$	(3.14 , 0.07)	$J_4^{(26)}$	(3.95 , 0.80)
$J_5^{(26)}$	(6.76 , 0.45)	$J_6^{(26)}$	(6.76 , 0.87)	$J_7^{(26)}$	(7.75 , 0.40)	$J_8^{(26)}$	(9.41 , 0.34)
$J_9^{(26)}$	(9.41 , 1.05)	$J_{10}^{(26)}$	(10.17 , 0.77)	$J_{11}^{(26)}$	(10.17 , 0.44)	$J_{12}^{(26)}$	(12.58 , 0.11)
$J_{13}^{(26)}$	(12.58 , 0.78)	$J_{14}^{(26)}$	(13.95 , 0.68)	$J_{15}^{(26)}$	(13.95 , 0.08)	$J_{16}^{(26)}$	(14.45 , 0.13)
$J_{17}^{(26)}$	(15.51 , 1.05)	$J_{18}^{(26)}$	(15.51 , 0.53)	$J_{19}^{(26)}$	(17.34 , 0.02)	$J_{20}^{(26)}$	(19.15 , 0.06)
$J_{21}^{(26)}$	(20.28 , 0.15)	$J_{22}^{(26)}$	(20.28 , 0.95)	$J_{23}^{(26)}$	(20.28 , 0.50)	$J_{24}^{(26)}$	(23.12 , 0.60)
$J_{25}^{(26)}$	(23.12 , 0.53)	$J_{26}^{(26)}$	(23.12 , 0.07)	$J_{27}^{(26)}$	(24.16 , 0.96)	$J_{28}^{(26)}$	(24.16 , 0.31)
$J_{29}^{(26)}$	(24.16 , 0.29)	$J_{30}^{(26)}$	(25.64 , 0.63)	$J_{31}^{(26)}$	(25.64 , 0.51)	$J_{32}^{(26)}$	(29.30 , 0.39)
$J_{33}^{(26)}$	(31.07 , 0.70)	$J_{34}^{(26)}$	(31.07 , 1.00)	$J_{35}^{(26)}$	(31.57 , 0.66)	$J_{36}^{(26)}$	(32.94 , 0.30)
$J_{37}^{(26)}$	(32.94 , 0.22)	$J_{38}^{(26)}$	(32.94 , 0.47)	$J_{39}^{(26)}$	(33.46 , 0.03)	$J_{40}^{(26)}$	(33.46 , 0.93)
$J_{41}^{(26)}$	(36.18 , 0.65)	$J_{42}^{(26)}$	(36.18 , 0.22)	$J_{43}^{(26)}$	(39.14 , 0.55)	$J_{44}^{(26)}$	(39.14 , 0.06)
$J_{45}^{(26)}$	(40.96 , 0.92)	$J_{46}^{(26)}$	(40.96 , 0.47)	$J_{47}^{(26)}$	(42.65 , 0.58)	$J_{48}^{(26)}$	(42.65 , 0.60)
$J_{49}^{(26)}$	(42.65 , 0.73)						

Aid Site 27 (39 jobs)

$J_1^{(27)}$	(5.04 , 0.05)	$J_2^{(27)}$	(5.04 , 0.02)	$J_3^{(27)}$	(5.04 , 0.02)	$J_4^{(27)}$	(5.56 , 0.01)
$J_5^{(27)}$	(5.56 , 0.03)	$J_6^{(27)}$	(8.26 , 0.03)	$J_7^{(27)}$	(8.26 , 0.03)	$J_8^{(27)}$	(10.64 , 0.01)
$J_9^{(27)}$	(10.64 , 0.02)	$J_{10}^{(27)}$	(11.44 , 0.01)	$J_{11}^{(27)}$	(11.44 , 0.04)	$J_{12}^{(27)}$	(12.96 , 0.02)
$J_{13}^{(27)}$	(16.30 , 0.01)	$J_{14}^{(27)}$	(16.30 , 0.00)	$J_{15}^{(27)}$	(16.30 , 0.04)	$J_{16}^{(27)}$	(19.26 , 0.02)
$J_{17}^{(27)}$	(19.26 , 0.02)	$J_{18}^{(27)}$	(19.92 , 0.04)	$J_{19}^{(27)}$	(20.26 , 0.03)	$J_{20}^{(27)}$	(22.60 , 0.03)
$J_{21}^{(27)}$	(22.60 , 0.01)	$J_{22}^{(27)}$	(22.60 , 0.03)	$J_{23}^{(27)}$	(23.76 , 0.04)	$J_{24}^{(27)}$	(25.45 , 0.03)
$J_{25}^{(27)}$	(26.43 , 0.04)	$J_{26}^{(27)}$	(26.86 , 0.05)	$J_{27}^{(27)}$	(27.47 , 0.04)	$J_{28}^{(27)}$	(27.47 , 0.01)
$J_{29}^{(27)}$	(30.49 , 0.00)	$J_{30}^{(27)}$	(31.87 , 0.02)	$J_{31}^{(27)}$	(31.87 , 0.00)	$J_{32}^{(27)}$	(33.70 , 0.02)
$J_{33}^{(27)}$	(35.30 , 0.04)	$J_{34}^{(27)}$	(35.30 , 0.01)	$J_{35}^{(27)}$	(35.30 , 0.02)	$J_{36}^{(27)}$	(36.72 , 0.02)
$J_{37}^{(27)}$	(36.94 , 0.04)	$J_{38}^{(27)}$	(38.65 , 0.04)	$J_{39}^{(27)}$	(40.36 , 0.02)		

Aid Site 28 (44 jobs)

$J_1^{(28)}$	(0.76 , 0.60)	$J_2^{(28)}$	(0.76 , 6.14)	$J_3^{(28)}$	(0.76 , 6.79)	$J_4^{(28)}$	(0.76 , 7.54)
$J_5^{(28)}$	(2.34 , 2.67)	$J_6^{(28)}$	(2.34 , 6.57)	$J_7^{(28)}$	(3.82 , 4.40)	$J_8^{(28)}$	(3.82 , 3.79)
$J_9^{(28)}$	(6.75 , 6.71)	$J_{10}^{(28)}$	(6.75 , 7.78)	$J_{11}^{(28)}$	(6.75 , 0.62)	$J_{12}^{(28)}$	(8.00 , 5.79)
$J_{13}^{(28)}$	(9.15 , 1.92)	$J_{14}^{(28)}$	(10.50 , 3.26)	$J_{15}^{(28)}$	(12.87 , 2.19)	$J_{16}^{(28)}$	(13.94 , 6.80)
$J_{17}^{(28)}$	(14.98 , 8.14)	$J_{18}^{(28)}$	(16.11 , 5.31)	$J_{19}^{(28)}$	(19.59 , 5.75)	$J_{20}^{(28)}$	(20.00 , 7.62)
$J_{21}^{(28)}$	(21.90 , 5.62)	$J_{22}^{(28)}$	(24.30 , 4.65)	$J_{23}^{(28)}$	(24.30 , 3.11)	$J_{24}^{(28)}$	(24.30 , 5.19)
$J_{25}^{(28)}$	(24.30 , 2.97)	$J_{26}^{(28)}$	(25.21 , 3.33)	$J_{27}^{(28)}$	(26.85 , 3.01)	$J_{28}^{(28)}$	(26.85 , 3.83)
$J_{29}^{(28)}$	(28.22 , 4.11)	$J_{30}^{(28)}$	(28.22 , 7.44)	$J_{31}^{(28)}$	(28.22 , 1.69)	$J_{32}^{(28)}$	(28.22 , 2.77)
$J_{33}^{(28)}$	(28.22 , 4.69)	$J_{34}^{(28)}$	(29.22 , 3.98)	$J_{35}^{(28)}$	(32.25 , 2.14)	$J_{36}^{(28)}$	(33.69 , 4.74)
$J_{37}^{(28)}$	(33.69 , 7.18)	$J_{38}^{(28)}$	(33.69 , 0.50)	$J_{39}^{(28)}$	(34.85 , 3.60)	$J_{40}^{(28)}$	(34.85 , 0.69)
$J_{41}^{(28)}$	(34.85 , 4.60)	$J_{42}^{(28)}$	(34.85 , 4.41)	$J_{43}^{(28)}$	(36.51 , 6.28)	$J_{44}^{(28)}$	(36.51 , 1.91)

Aid Site 29 (20 jobs)

$J_1^{(29)}$	(1.17 , 0.13)	$J_2^{(29)}$	(1.17 , 0.08)	$J_3^{(29)}$	(2.39 , 0.38)	$J_4^{(29)}$	(4.19 , 0.56)
$J_5^{(29)}$	(4.19 , 0.70)	$J_6^{(29)}$	(4.94 , 0.67)	$J_7^{(29)}$	(8.02 , 0.60)	$J_8^{(29)}$	(8.02 , 0.40)
$J_9^{(29)}$	(9.33 , 0.60)	$J_{10}^{(29)}$	(9.33 , 0.81)	$J_{11}^{(29)}$	(14.31 , 0.43)	$J_{12}^{(29)}$	(14.31 , 0.26)
$J_{13}^{(29)}$	(15.20 , 0.67)	$J_{14}^{(29)}$	(16.92 , 0.20)	$J_{15}^{(29)}$	(16.92 , 0.20)	$J_{16}^{(29)}$	(16.92 , 0.39)
$J_{17}^{(29)}$	(16.92 , 0.52)	$J_{18}^{(29)}$	(18.97 , 0.52)	$J_{19}^{(29)}$	(20.18 , 0.10)	$J_{20}^{(29)}$	(22.37 , 0.10)

Aid Site 30 (41 jobs)

$J_1^{(30)}$	(0.57 , 0.25)	$J_2^{(30)}$	(2.23 , 1.00)	$J_3^{(30)}$	(2.23 , 1.30)	$J_4^{(30)}$	(3.37 , 1.48)
$J_5^{(30)}$	(6.23 , 2.54)	$J_6^{(30)}$	(6.23 , 1.28)	$J_7^{(30)}$	(8.06 , 1.04)	$J_8^{(30)}$	(9.85 , 2.59)
$J_9^{(30)}$	(9.85 , 0.68)	$J_{10}^{(30)}$	(9.85 , 1.73)	$J_{11}^{(30)}$	(10.70 , 2.51)	$J_{12}^{(30)}$	(12.40 , 1.74)
$J_{13}^{(30)}$	(12.40 , 0.78)	$J_{14}^{(30)}$	(13.69 , 1.38)	$J_{15}^{(30)}$	(14.93 , 0.00)	$J_{16}^{(30)}$	(17.47 , 2.30)
$J_{17}^{(30)}$	(17.47 , 1.05)	$J_{18}^{(30)}$	(19.03 , 0.78)	$J_{19}^{(30)}$	(19.03 , 2.47)	$J_{20}^{(30)}$	(19.03 , 1.20)
$J_{21}^{(30)}$	(22.64 , 0.75)	$J_{22}^{(30)}$	(22.64 , 0.22)	$J_{23}^{(30)}$	(22.64 , 1.51)	$J_{24}^{(30)}$	(23.87 , 0.40)
$J_{25}^{(30)}$	(27.23 , 0.19)	$J_{26}^{(30)}$	(27.23 , 1.51)	$J_{27}^{(30)}$	(27.23 , 0.75)	$J_{28}^{(30)}$	(27.23 , 0.94)
$J_{29}^{(30)}$	(27.23 , 1.88)	$J_{30}^{(30)}$	(33.30 , 2.23)	$J_{31}^{(30)}$	(33.30 , 0.90)	$J_{32}^{(30)}$	(34.73 , 2.50)
$J_{33}^{(30)}$	(34.73 , 2.48)	$J_{34}^{(30)}$	(34.73 , 0.54)	$J_{35}^{(30)}$	(35.30 , 2.00)	$J_{36}^{(30)}$	(36.31 , 1.60)
$J_{37}^{(30)}$	(36.31 , 2.39)	$J_{38}^{(30)}$	(37.77 , 1.57)	$J_{39}^{(30)}$	(38.04 , 1.82)	$J_{40}^{(30)}$	(38.87 , 1.93)
$J_{41}^{(30)}$	(38.87 , 1.00)						

The unit delivery cost is set to be $D = 5$. After applying AAS to this instance, the assignment of all jobs to 5 medical centers will be as follows.

Medical Center 1

$$\begin{aligned}
 & J_1^{(14)}, J_1^{(9)}, J_2^{(9)}, J_2^{(14)}, J_3^{(14)}, J_1^{(18)}, J_1^{(1)}, J_1^{(12)}, J_2^{(12)}, J_1^{(2)}, J_1^{(26)}, J_2^{(26)}, J_1^{(29)}, J_2^{(29)}, \\
 & J_4^{(14)}, J_1^{(4)}, J_2^{(4)}, J_3^{(4)}, J_3^{(9)}, J_1^{(15)}, J_2^{(15)}, J_3^{(15)}, J_2^{(1)}, J_1^{(8)}, J_1^{(13)}, J_2^{(13)}, J_3^{(13)}, J_1^{(24)}, \\
 & J_2^{(2)}, J_3^{(2)}, J_5^{(14)}, J_4^{(4)}, J_3^{(29)}, J_4^{(9)}, J_3^{(26)}, J_4^{(2)}, J_5^{(2)}, J_5^{(4)}, J_6^{(4)}, J_3^{(12)}, J_4^{(12)}, J_5^{(12)}, \\
 & J_2^{(18)}, J_3^{(18)}, J_4^{(15)}, J_5^{(15)}, J_6^{(15)}, J_6^{(14)}, J_7^{(14)}, J_6^{(2)}, J_7^{(2)}, J_4^{(26)}, J_7^{(4)}, J_6^{(12)}, J_4^{(29)}, J_5^{(29)}, \\
 & J_3^{(1)}, J_4^{(1)}, J_6^{(29)}, J_8^{(2)}, J_4^{(13)}, J_5^{(13)}, J_6^{(13)}, J_8^{(4)}, J_4^{(18)}, J_5^{(18)}, J_6^{(18)}, J_7^{(15)}, J_7^{(12)}, J_8^{(12)}, \\
 & J_2^{(24)}, J_5^{(9)}, J_9^{(2)}, J_8^{(15)}, J_7^{(13)}, J_5^{(26)}, J_6^{(26)}, J_9^{(12)}, J_{10}^{(12)}, J_7^{(18)}, J_8^{(18)}, J_9^{(18)}, J_{10}^{(18)}, J_3^{(24)}, \\
 & J_4^{(24)}, J_{10}^{(2)}, J_7^{(26)}, J_{11}^{(18)}, J_{12}^{(18)}, J_{11}^{(2)}, J_{12}^{(2)}, J_9^{(4)}, J_{13}^{(18)}, J_{14}^{(18)}, J_5^{(24)}, J_5^{(1)}, J_6^{(1)}, J_7^{(1)}, \\
 & J_{11}^{(12)}, J_{12}^{(12)}, J_{13}^{(2)}, J_8^{(26)}, J_9^{(26)}, J_{13}^{(12)}, J_8^{(1)}, J_{14}^{(12)}, J_{15}^{(12)}, J_6^{(9)}, J_7^{(9)}, J_{10}^{(4)}, J_9^{(1)}, J_8^{(9)}, \\
 & J_9^{(9)}, J_{10}^{(9)}, J_{11}^{(9)}, J_{15}^{(18)}, J_{10}^{(1)}, J_{11}^{(1)}, J_{11}^{(4)}, J_6^{(24)}, J_{12}^{(4)}, J_{13}^{(4)}, J_{12}^{(9)}, J_7^{(24)}, J_{13}^{(9)}, J_8^{(24)}, \\
 & J_9^{(24)}, J_{10}^{(24)}, J_{16}^{(18)}, J_{17}^{(18)}, J_{18}^{(18)}, J_{14}^{(9)}, J_{11}^{(24)}, J_{15}^{(9)}, J_{16}^{(9)}, J_{19}^{(18)}, J_{20}^{(18)}, J_{21}^{(18)}, J_{22}^{(18)}, J_{17}^{(9)}, \\
 & J_{18}^{(9)}, J_{19}^{(9)}, J_{23}^{(18)}, J_{24}^{(18)}, J_{25}^{(18)}, J_{20}^{(9)}, J_{26}^{(18)}, J_{27}^{(18)}, J_{21}^{(9)}, J_{22}^{(9)}, J_{28}^{(18)}, J_{23}^{(9)}, J_{24}^{(9)}, J_{29}^{(18)}, \\
 & J_{30}^{(18)}, J_{31}^{(18)}, J_{25}^{(9)}, J_{26}^{(9)}, J_{27}^{(9)}, J_{28}^{(9)}, J_{29}^{(9)}, J_{30}^{(9)}, J_{31}^{(9)}, J_{32}^{(9)}, J_{33}^{(9)}, J_{34}^{(9)}, J_{35}^{(9)}, J_{36}^{(9)}, \\
 & J_{37}^{(9)}, J_{38}^{(9)}
 \end{aligned}$$

Medical Center 2

$$\begin{aligned}
 & J_1^{(30)}, J_1^{(28)}, J_2^{(28)}, J_3^{(28)}, J_4^{(28)}, J_1^{(22)}, J_1^{(11)}, J_1^{(6)}, J_2^{(30)}, J_3^{(30)}, J_5^{(28)}, J_6^{(28)}, J_1^{(17)}, J_2^{(17)}, \\
 & J_2^{(6)}, J_2^{(22)}, J_3^{(22)}, J_3^{(17)}, J_4^{(17)}, J_4^{(30)}, J_3^{(6)}, J_7^{(28)}, J_8^{(28)}, J_4^{(22)}, J_5^{(22)}, J_4^{(6)}, J_5^{(6)}, J_5^{(17)}, \\
 & J_6^{(17)}, J_5^{(30)}, J_6^{(30)}, J_6^{(22)}, J_7^{(22)}, J_6^{(6)}, J_7^{(6)}, J_8^{(6)}, J_9^{(28)}, J_{10}^{(28)}, J_{11}^{(28)}, J_{12}^{(28)}, J_7^{(30)}, J_2^{(11)}, \\
 & J_8^{(22)}, J_9^{(22)}, J_{10}^{(22)}, J_{13}^{(28)}, J_8^{(30)}, J_9^{(30)}, J_{10}^{(30)}, J_3^{(11)}, J_{14}^{(28)}, J_{11}^{(30)}, J_7^{(17)}, J_{11}^{(22)}, J_{12}^{(22)}, J_{13}^{(22)}, \\
 & J_{14}^{(22)}, J_{12}^{(30)}, J_{13}^{(30)}, J_{15}^{(22)}, J_{15}^{(28)}, J_8^{(17)}, J_{16}^{(22)}, J_{14}^{(30)}, J_{16}^{(28)}, J_{15}^{(30)}, J_{17}^{(28)}, J_9^{(17)}, J_{18}^{(28)}, J_{17}^{(22)}, \\
 & J_{16}^{(30)}, J_{17}^{(30)}, J_{10}^{(17)}, J_{18}^{(22)}, J_{19}^{(22)}, J_{11}^{(17)}, J_{12}^{(17)}, J_{18}^{(30)}, J_{19}^{(30)}, J_{20}^{(30)}, J_{19}^{(28)}, J_{20}^{(28)}, J_{20}^{(22)}, J_{21}^{(22)}, \\
 & J_{22}^{(22)}, J_{21}^{(28)}, J_{13}^{(17)}, J_{21}^{(30)}, J_{22}^{(30)}, J_{23}^{(30)}, J_{14}^{(17)}, J_{15}^{(17)}, J_{23}^{(22)}, J_{24}^{(30)}, J_{22}^{(28)}, J_{23}^{(28)}, J_{24}^{(28)}, J_{25}^{(28)},
 \end{aligned}$$

$J_{24}^{(22)}, J_{26}^{(28)}, J_{16}^{(17)}, J_{17}^{(17)}, J_{18}^{(17)}, J_{27}^{(28)}, J_{28}^{(28)}, J_{25}^{(22)}, J_{25}^{(30)}, J_{26}^{(30)}, J_{27}^{(30)}, J_{28}^{(30)}, J_{29}^{(30)}, J_{19}^{(17)},$
 $J_{29}^{(28)}, J_{30}^{(28)}, J_{31}^{(28)}, J_{32}^{(28)}, J_{33}^{(28)}, J_{26}^{(22)}, J_{27}^{(22)}, J_{20}^{(17)}, J_{34}^{(28)}, J_{21}^{(17)}, J_{22}^{(17)}, J_{23}^{(17)}, J_{28}^{(22)}, J_{35}^{(28)},$
 $J_{24}^{(17)}, J_{25}^{(17)}, J_{26}^{(17)}, J_{27}^{(17)}, J_{29}^{(22)}, J_{30}^{(30)}, J_{31}^{(30)}, J_{36}^{(28)}, J_{37}^{(28)}, J_{38}^{(28)}, J_{30}^{(22)}, J_{31}^{(22)}, J_{32}^{(30)}, J_{33}^{(30)},$
 $J_{34}^{(30)}, J_{39}^{(28)}, J_{40}^{(28)}, J_{41}^{(28)}, J_{42}^{(28)}, J_{35}^{(30)}, J_{32}^{(22)}, J_{33}^{(22)}, J_{34}^{(22)}, J_{35}^{(22)}, J_{36}^{(30)}, J_{37}^{(30)}, J_{43}^{(28)}, J_{36}^{(22)},$
 $J_{37}^{(22)}, J_{38}^{(30)}, J_{39}^{(30)}, J_{38}^{(22)}, J_{40}^{(30)}, J_{41}^{(30)}, J_{39}^{(22)}, J_{40}^{(22)}, J_{41}^{(22)}, J_{42}^{(22)}, J_{43}^{(22)}, J_{44}^{(22)}, J_{45}^{(22)}, J_{46}^{(22)}$

Medical Center 3

$J_1^{(20)}, J_1^{(5)}, J_1^{(10)}, J_2^{(10)}, J_1^{(21)}, J_2^{(21)}, J_1^{(23)}, J_1^{(25)}, J_1^{(19)}, J_2^{(20)}, J_3^{(10)}, J_2^{(23)}, J_2^{(5)}, J_3^{(5)},$
 $J_2^{(19)}, J_3^{(21)}, J_4^{(21)}, J_5^{(21)}, J_2^{(25)}, J_3^{(25)}, J_4^{(5)}, J_3^{(20)}, J_4^{(20)}, J_5^{(20)}, J_3^{(23)}, J_4^{(23)}, J_3^{(19)}, J_4^{(19)},$
 $J_5^{(19)}, J_4^{(25)}, J_5^{(25)}, J_6^{(25)}, J_7^{(25)}, J_4^{(10)}, J_5^{(10)}, J_5^{(5)}, J_6^{(19)}, J_7^{(19)}, J_5^{(23)}, J_6^{(23)}, J_6^{(20)}, J_8^{(19)},$
 $J_6^{(5)}, J_8^{(25)}, J_7^{(20)}, J_8^{(20)}, J_7^{(23)}, J_9^{(20)}, J_9^{(25)}, J_{10}^{(25)}, J_7^{(5)}, J_8^{(5)}, J_9^{(5)}, J_{10}^{(5)}, J_6^{(10)}, J_7^{(10)},$
 $J_9^{(19)}, J_{10}^{(19)}, J_{11}^{(19)}, J_{11}^{(25)}, J_{12}^{(25)}, J_8^{(23)}, J_9^{(23)}, J_{10}^{(20)}, J_{11}^{(20)}, J_{12}^{(20)}, J_{13}^{(20)}, J_{11}^{(5)}, J_{13}^{(25)}, J_{14}^{(25)},$
 $J_{12}^{(19)}, J_{13}^{(19)}, J_{14}^{(19)}, J_{15}^{(19)}, J_8^{(10)}, J_{10}^{(23)}, J_{11}^{(23)}, J_{14}^{(20)}, J_{15}^{(20)}, J_{16}^{(20)}, J_{17}^{(20)}, J_{15}^{(25)}, J_{16}^{(25)}, J_{12}^{(5)},$
 $J_{13}^{(5)}, J_{14}^{(5)}, J_{17}^{(25)}, J_{18}^{(25)}, J_{16}^{(19)}, J_{17}^{(19)}, J_{18}^{(20)}, J_{15}^{(5)}, J_{16}^{(5)}, J_{17}^{(5)}, J_{12}^{(23)}, J_{19}^{(25)}, J_{20}^{(25)}, J_9^{(10)},$
 $J_{10}^{(10)}, J_{11}^{(10)}, J_{12}^{(10)}, J_{13}^{(23)}, J_{18}^{(19)}, J_{19}^{(19)}, J_{20}^{(19)}, J_{13}^{(10)}, J_{18}^{(5)}, J_{19}^{(5)}, J_{19}^{(20)}, J_{20}^{(20)}, J_{21}^{(20)}, J_{22}^{(20)},$
 $J_{23}^{(20)}, J_{21}^{(25)}, J_{21}^{(19)}, J_{14}^{(10)}, J_{15}^{(10)}, J_{24}^{(20)}, J_{25}^{(20)}, J_{26}^{(20)}, J_{22}^{(25)}, J_{23}^{(25)}, J_{27}^{(20)}, J_{20}^{(5)}, J_{21}^{(5)}, J_{22}^{(19)},$
 $J_{14}^{(23)}, J_{16}^{(10)}, J_{22}^{(5)}, J_{23}^{(5)}, J_{15}^{(23)}, J_{16}^{(23)}, J_{17}^{(23)}, J_{28}^{(20)}, J_{23}^{(19)}, J_{24}^{(19)}, J_{24}^{(5)}, J_{29}^{(20)}, J_{25}^{(19)}, J_{17}^{(10)},$
 $J_{18}^{(10)}, J_{19}^{(10)}, J_{20}^{(10)}, J_{30}^{(20)}, J_{31}^{(20)}, J_{32}^{(20)}, J_{33}^{(20)}, J_{34}^{(20)}, J_{18}^{(23)}, J_{19}^{(23)}, J_{20}^{(23)}, J_{21}^{(10)}, J_{35}^{(20)}, J_{36}^{(20)},$
 $J_{21}^{(23)}, J_{22}^{(10)}, J_{23}^{(10)}, J_{24}^{(10)}, J_{25}^{(10)}, J_{26}^{(10)}, J_{26}^{(19)}, J_{37}^{(20)}, J_{22}^{(23)}, J_{27}^{(10)}, J_{27}^{(19)}, J_{28}^{(19)}, J_{28}^{(10)}, J_{29}^{(10)},$
 $J_{38}^{(20)}, J_{23}^{(23)}, J_{29}^{(19)}, J_{30}^{(19)}, J_{24}^{(23)}, J_{31}^{(19)}, J_{32}^{(19)}, J_{33}^{(19)}, J_{25}^{(23)}, J_{26}^{(23)}, J_{27}^{(23)}, J_{28}^{(23)}, J_{29}^{(23)}, J_{30}^{(23)},$
 $J_{31}^{(23)}, J_{32}^{(23)}, J_{33}^{(23)}, J_{34}^{(23)}, J_{35}^{(23)}, J_{36}^{(23)}, J_{37}^{(23)}, J_{38}^{(23)}, J_{39}^{(23)}, J_{40}^{(23)}, J_{41}^{(23)}, J_{42}^{(23)}$

Medical Center 4

$J_1^{(27)}, J_2^{(27)}, J_3^{(27)}, J_8^{(14)}, J_9^{(14)}, J_{10}^{(14)}, J_4^{(27)}, J_5^{(27)}, J_{11}^{(14)}, J_{12}^{(14)}, J_{13}^{(14)}, J_8^{(3)}, J_9^{(6)}, J_{14}^{(14)},$

$J_{15}^{(14)}, J_5^{(16)}, J_7^{(29)}, J_8^{(29)}, J_2^{(8)}, J_{10}^{(6)}, J_6^{(27)}, J_7^{(27)}, J_3^{(8)}, J_{16}^{(14)}, J_9^{(29)}, J_{10}^{(29)}, J_{11}^{(6)}, J_{12}^{(6)},$
 $J_8^{(27)}, J_9^{(27)}, J_{17}^{(14)}, J_{18}^{(14)}, J_{19}^{(14)}, J_4^{(8)}, J_5^{(8)}, J_{10}^{(27)}, J_{11}^{(27)}, J_{13}^{(6)}, J_{14}^{(6)}, J_{15}^{(6)}, J_{16}^{(6)}, J_{12}^{(27)},$
 $J_{12}^{(3)}, J_{13}^{(3)}, J_6^{(8)}, J_6^{(16)}, J_7^{(16)}, J_{14}^{(3)}, J_{15}^{(3)}, J_{11}^{(29)}, J_{12}^{(29)}, J_8^{(16)}, J_7^{(8)}, J_{17}^{(6)}, J_{18}^{(6)}, J_{19}^{(6)},$
 $J_{13}^{(29)}, J_{17}^{(3)}, J_{13}^{(27)}, J_{14}^{(27)}, J_{15}^{(27)}, J_{20}^{(6)}, J_{21}^{(6)}, J_{20}^{(14)}, J_{21}^{(14)}, J_{22}^{(14)}, J_{23}^{(14)}, J_9^{(16)}, J_{10}^{(16)}, J_{11}^{(16)},$
 $J_{14}^{(29)}, J_{15}^{(29)}, J_{16}^{(29)}, J_{17}^{(29)}, J_8^{(8)}, J_9^{(8)}, J_{12}^{(16)}, J_{13}^{(16)}, J_{22}^{(6)}, J_{14}^{(16)}, J_{18}^{(29)}, J_{10}^{(8)}, J_{16}^{(27)}, J_{17}^{(27)},$
 $J_{15}^{(16)}, J_{16}^{(16)}, J_{23}^{(6)}, J_{17}^{(16)}, J_{18}^{(16)}, J_{18}^{(27)}, J_{19}^{(29)}, J_{11}^{(8)}, J_{19}^{(27)}, J_{24}^{(14)}, J_{24}^{(6)}, J_{25}^{(6)}, J_{26}^{(6)}, J_{27}^{(6)},$
 $J_{19}^{(16)}, J_{12}^{(8)}, J_{20}^{(16)}, J_{20}^{(29)}, J_{28}^{(6)}, J_{29}^{(6)}, J_{20}^{(27)}, J_{21}^{(27)}, J_{22}^{(27)}, J_{23}^{(27)}, J_{30}^{(6)}, J_{31}^{(6)}, J_{13}^{(8)}, J_{14}^{(8)},$
 $J_{15}^{(8)}, J_{24}^{(27)}, J_{32}^{(6)}, J_{25}^{(27)}, J_{26}^{(27)}, J_{22}^{(16)}, J_{23}^{(16)}, J_{27}^{(27)}, J_{28}^{(27)}, J_{33}^{(6)}, J_{34}^{(6)}, J_{35}^{(6)}, J_{36}^{(6)}, J_{29}^{(27)},$
 $J_{24}^{(16)}, J_{25}^{(16)}, J_{37}^{(6)}, J_{38}^{(6)}, J_{30}^{(27)}, J_{31}^{(27)}, J_{26}^{(16)}, J_{27}^{(16)}, J_{28}^{(16)}, J_{39}^{(6)}, J_{40}^{(6)}, J_{41}^{(6)}, J_{32}^{(27)}, J_{42}^{(6)},$
 $J_{33}^{(27)}, J_{34}^{(27)}, J_{35}^{(27)}, J_{29}^{(16)}, J_{30}^{(16)}, J_{31}^{(16)}, J_{32}^{(16)}, J_{44}^{(28)}, J_{36}^{(27)}, J_{37}^{(27)}, J_{33}^{(16)}, J_{38}^{(27)}, J_{34}^{(16)}, J_{35}^{(16)},$
 $J_{36}^{(16)}, J_{39}^{(27)}, J_{37}^{(16)}, J_{38}^{(16)}, J_{39}^{(16)}, J_{40}^{(16)}$

Medical Center 5

$J_1^{(16)}, J_1^{(3)}, J_2^{(3)}, J_1^{(7)}, J_2^{(7)}, J_3^{(7)}, J_4^{(7)}, J_3^{(3)}, J_4^{(3)}, J_2^{(16)}, J_3^{(16)}, J_5^{(3)}, J_4^{(16)}, J_6^{(3)},$
 $J_5^{(7)}, J_6^{(7)}, J_7^{(7)}, J_8^{(7)}, J_9^{(7)}, J_7^{(3)}, J_{10}^{(7)}, J_{11}^{(7)}, J_{12}^{(7)}, J_{10}^{(26)}, J_{11}^{(26)}, J_9^{(15)}, J_{13}^{(7)}, J_9^{(3)},$
 $J_{10}^{(3)}, J_{11}^{(3)}, J_{16}^{(12)}, J_{17}^{(12)}, J_{12}^{(26)}, J_{13}^{(26)}, J_{18}^{(12)}, J_{10}^{(15)}, J_{19}^{(12)}, J_{14}^{(26)}, J_{15}^{(26)}, J_{12}^{(1)}, J_{16}^{(26)}, J_{14}^{(7)},$
 $J_{15}^{(7)}, J_{20}^{(12)}, J_{16}^{(3)}, J_{18}^{(3)}, J_{11}^{(15)}, J_{12}^{(15)}, J_{13}^{(15)}, J_{13}^{(1)}, J_{17}^{(26)}, J_{18}^{(26)}, J_{21}^{(12)}, J_{22}^{(12)}, J_{16}^{(7)}, J_{17}^{(7)},$
 $J_{19}^{(26)}, J_{18}^{(7)}, J_{19}^{(7)}, J_{20}^{(26)}, J_{12}^{(24)}, J_{13}^{(24)}, J_{23}^{(12)}, J_{14}^{(24)}, J_{21}^{(26)}, J_{22}^{(26)}, J_{23}^{(26)}, J_{24}^{(12)}, J_{25}^{(12)}, J_{14}^{(15)},$
 $J_{15}^{(15)}, J_{16}^{(15)}, J_{17}^{(15)}, J_{26}^{(12)}, J_{20}^{(7)}, J_{21}^{(7)}, J_{21}^{(16)}, J_{24}^{(26)}, J_{25}^{(26)}, J_{26}^{(26)}, J_{27}^{(12)}, J_{22}^{(7)}, J_{23}^{(7)}, J_{24}^{(7)},$
 $J_{25}^{(7)}, J_{26}^{(7)}, J_{27}^{(26)}, J_{28}^{(26)}, J_{29}^{(26)}, J_{28}^{(12)}, J_{15}^{(24)}, J_{16}^{(24)}, J_{17}^{(24)}, J_{29}^{(12)}, J_{30}^{(26)}, J_{31}^{(26)}, J_{30}^{(12)}, J_{18}^{(24)},$
 $J_{19}^{(24)}, J_{31}^{(12)}, J_{20}^{(24)}, J_{21}^{(24)}, J_{27}^{(7)}, J_{28}^{(7)}, J_{32}^{(26)}, J_{32}^{(12)}, J_{33}^{(12)}, J_{34}^{(12)}, J_{33}^{(26)}, J_{34}^{(26)}, J_{35}^{(12)}, J_{35}^{(26)},$
 $J_{36}^{(12)}, J_{36}^{(26)}, J_{37}^{(26)}, J_{38}^{(26)}, J_{22}^{(24)}, J_{39}^{(26)}, J_{40}^{(26)}, J_{37}^{(12)}, J_{23}^{(24)}, J_{24}^{(24)}, J_{25}^{(24)}, J_{26}^{(24)}, J_{27}^{(24)}, J_{38}^{(12)},$
 $J_{39}^{(12)}, J_{41}^{(26)}, J_{42}^{(26)}, J_{28}^{(24)}, J_{29}^{(24)}, J_{43}^{(26)}, J_{44}^{(26)}, J_{30}^{(24)}, J_{31}^{(24)}, J_{32}^{(24)}, J_{45}^{(26)}, J_{46}^{(26)}, J_{33}^{(24)}, J_{47}^{(26)},$

$$J_{48}^{(26)}, J_{49}^{(26)}, J_{34}^{(24)}, J_{35}^{(24)}, J_{36}^{(24)}, J_{37}^{(24)}, J_{38}^{(24)}, J_{39}^{(24)}, J_{40}^{(24)}, J_{41}^{(24)}, J_{42}^{(24)}, J_{43}^{(24)}, J_{44}^{(24)}, J_{45}^{(24)},$$

$$J_{46}^{(24)}, J_{47}^{(24)}, J_{48}^{(24)}, J_{49}^{(24)}, J_{50}^{(24)}$$

The next step is to implement SMH8 for the 5 medical centers, respectively, and the objective values are 2522.7, 1194.6, 1804.7, 1683.5 and 2015.4. Therefore, the total objective value is 9220.9.

As the optimal result of the original value is unknown, a reference result is constructed to evaluate the combination of AAS and SMH8. The reference result is constructed as follows: It is assumed that there is a virtual medical center with 5 processors located at the virtual center position of the original 5 medical centers (see Figure 8.4). Then MMH5 is implemented to solve a off-line problem with single-medical-center (the virtual one) and multi-processors and multi-aid-sites.

It turns out that the reference value is 7495.9, which means the ratio of the above objective value to the reference value is 1.23. Thus, the performance of the combination of AAS and SMH8 for this instance is excellent as it exceed the average of SMH8 (see Table 4.9).

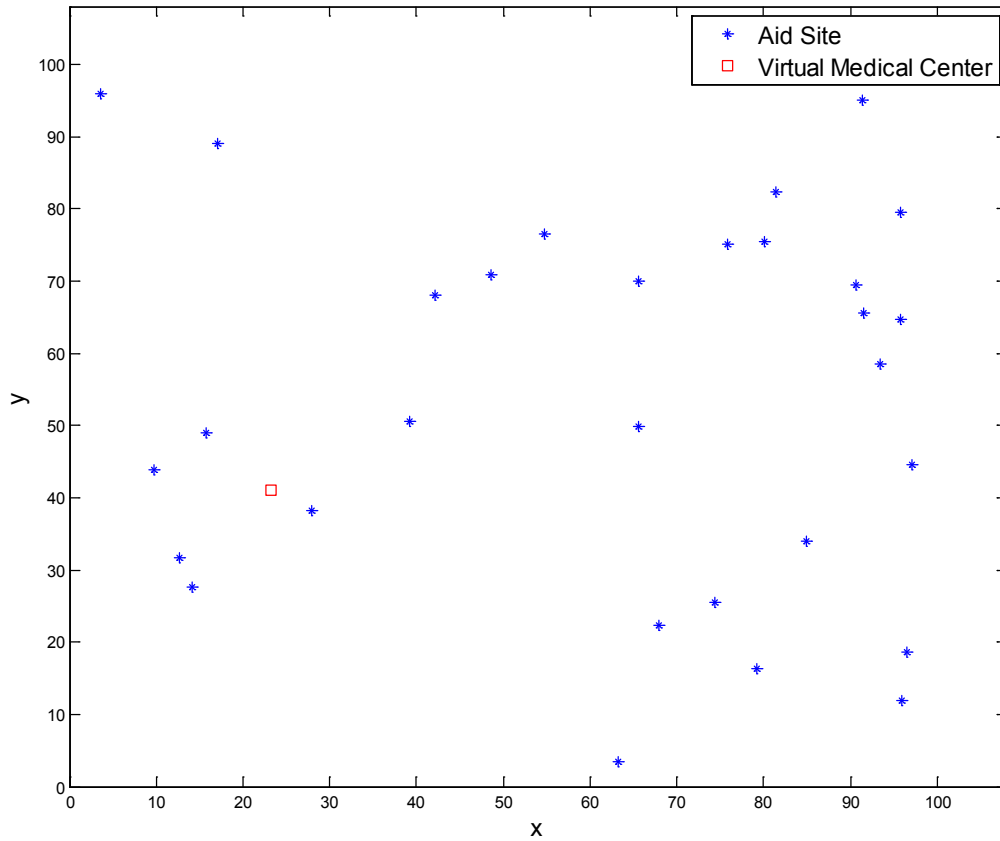


Figure 8.4. The Locations of 30 Aid Sites and The Virtual Medical Center

The simulation is conducted for 100 instances generated randomly and the result is shown in Figure 8.5. From the figure, the ratio of the above objective value to the reference value dose not exceed 1.55. Actually, the average of the ratio is 1.29, which indicates that AAS along with SMH8 can perform well for the normal cases.

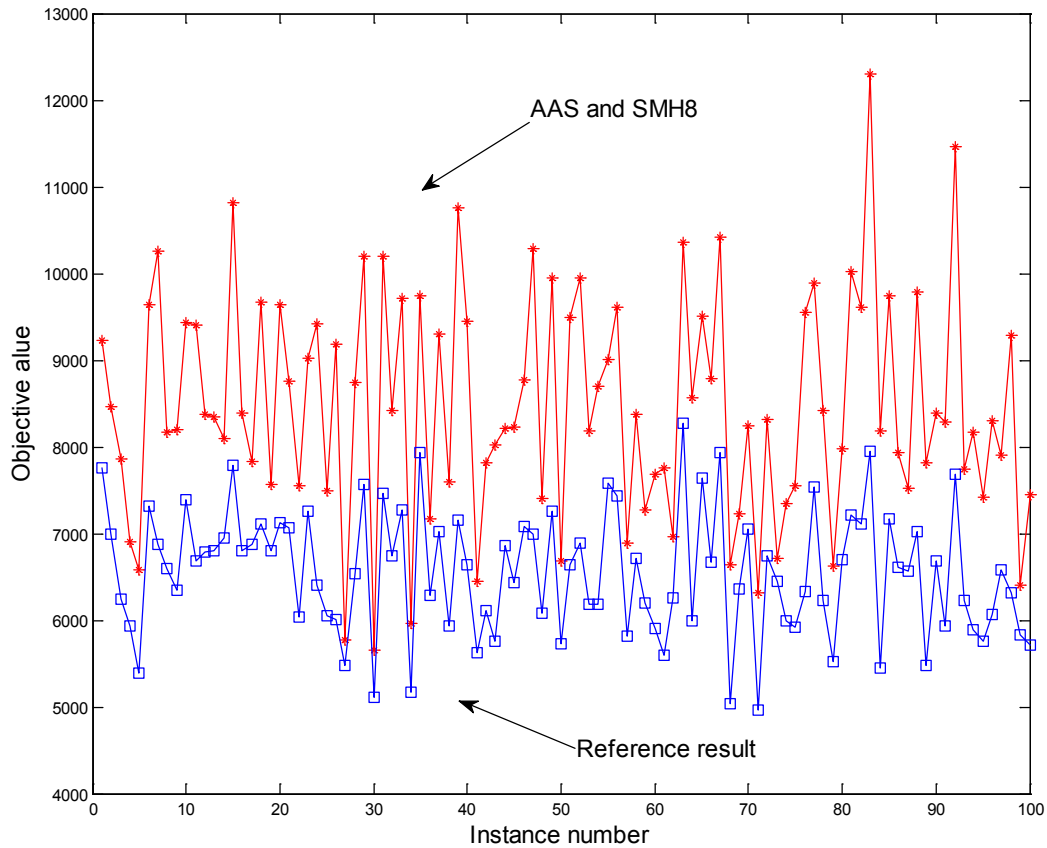


Figure 8.5. The Case Study Result for $D = 5$

Similar simulations are made for $D = 25$ and $D = 50$ (see Figures 8.6 and 8.7). It can be seen that the averages of the ratios are 1.43 and 1.56, respectively. Although the ratio gets larger as D increases, the result is still acceptable. The reason for the ratio getting larger is that the increase of D result in bad cases for SMH8, which can achieve 2 (the lower bound of SMP8) for the worst case. When D gets too large, the delivery part will be the main part and the problem will degenerate to the single medical center case, which can be solved by SMH8 without AAS. This means that AAS with SMH8 does not need to deal with the problem with a large D .

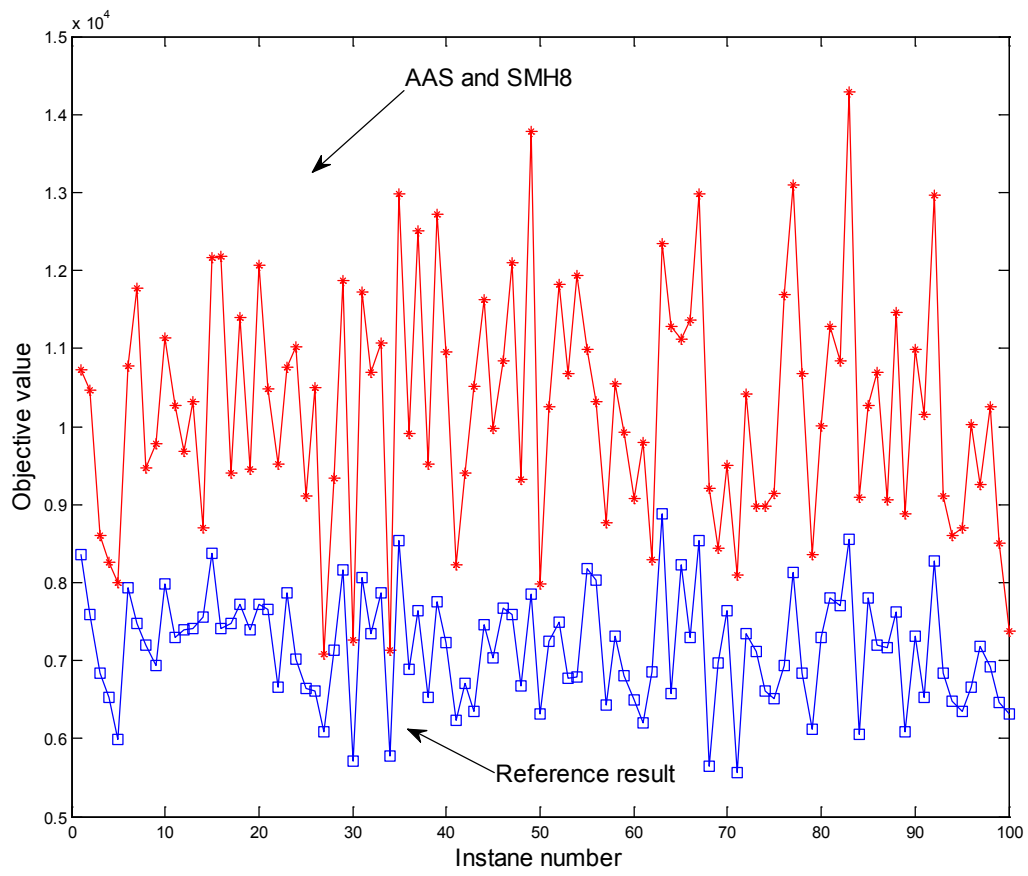


Figure 8.6. The Case Study Result for $D = 25$

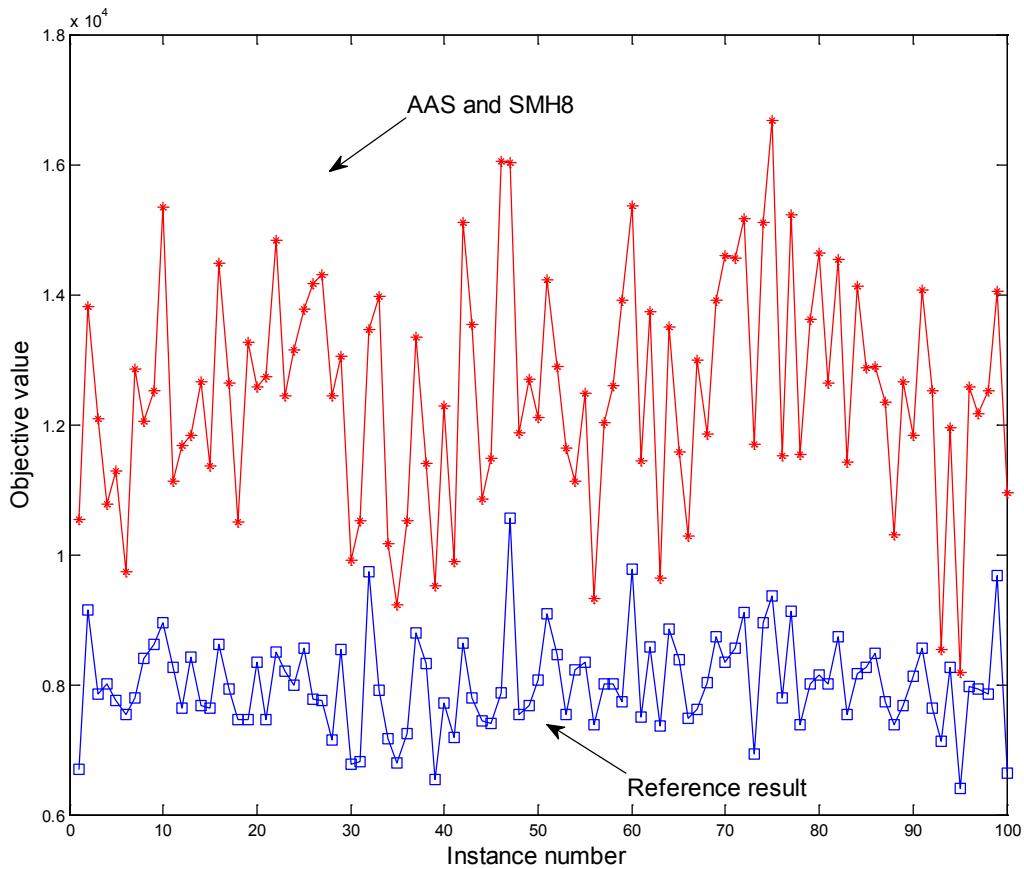


Figure 8.7. The Case Study Result for $D = 50$

8.6 Concluding Remarks

In this chapter, a case study to apply the developed algorithms for a real world problem of medical resources allocation was presented. The problem has a scale of practical situations. A policy to assign jobs to medical centers was proposed and it can achieve an excellent result by combining it with the on-line algorithm SMH8. It is noted that all the algorithms developed in this thesis have a similar structure, and therefore, their scalability should be similar to that of SMH8. Therefore, one can conclude the work described in this thesis

to the topic of supply chain scheduling can be applied not only to many manufacturing problems in practice but also to many emergency management problems.

CHAPTER 9 CONCLUSIONS

9.1 Overview and Contributions

Supply chain scheduling has gained a great popularity in the recent years as the globalization of manufacturing develops rapidly. This thesis studied the supply chain scheduling problem of Class 2, which considers that the manufacturers and the customers are not at the same places. This model has two main parts: processing of jobs in the manufacturer's site and delivery of the completed jobs to customers, which are two subsequent activities. This thesis aimed to define particular problems of this supply chain scheduling model and solve them. After having discussed the background and literature in related research field, three types of problems in terms of different configurations of machines and customers were defined. Algorithms for these problems were then developed along with the analysis and simulation-based verification of the algorithms.

The three configurations considered in this thesis are (1) single-machine multi-customers, (2) multi-machines single-customer and (3) multi-machines multi-customers. For each type of configuration, several specific problems characterized by different release environments, processing patterns, vehicle characteristics and delivery patterns are defined. All these problems integrate the foregoing two activities (processing and delivery) and minimize

both the time-based objective and cost-based objective. The methodology to deal with the problems is: (1) investigate the intractability of problems based on the complexity analysis (off-line) and competitive analysis (on-line); (2) develop algorithms according to different structures of problems; (3) examine the performance by the worst case analysis; (4) verify the effectiveness and the efficiency of algorithms from the simulation; (5) discuss the robustness and the resilience of algorithms; (6) implement the algorithms for the realistic situations in emergency management (EM).

There are several scientific merits with the study presented in this thesis. (1) in all the problem models, the time associated with individual customers is optimized separately, which hits the ultimate goal of manufacturing and service, namely customers satisfaction. (2) the challenge of the supply chain scheduling problem with multiple customers (two customers in particular) and limited capacity of delivery tools is tackled by proving the presence of the optimal algorithm. (3) There is a finding of two new measures for scheduling algorithms, namely robustness and resilience. They seem to be supplemental to the traditional measure (which is essentially based on the performance of algorithms), as the real world problem such as scheduling problem is never perfect and there are always some structural and/or parameter uncertainties.

The study presented in this thesis has a very high potential value for applications. The on-line scheduling problem appears in many occasions in manufacturing and service businesses. It also occurs in micro-systems (e.g., computing resources scheduling in embedded systems) and macro-systems (e.g., service resources scheduling in network systems). The

case study presented in Chapter 8 is an example of the applications in macro-systems.

The following conclusions can be drawn from the study presented in the thesis:

- (1) All the algorithms developed in this thesis have high effectiveness and high efficiency for both the worst scenarios and normal scenarios. Table 9.1 gives a summary of the results of all the algorithms developed in this thesis.
- (2) All the algorithms developed in this thesis have a good sense of robustness and resilience, which are first defined in this thesis.
- (3) The work can be extended to deal with the more realistic situations in practice.

Table 9.1. Results of Algorithms for All Problems

Problem	Intractability	Algorithm	Analysis
SMP1	–	SMH1 SA_SMH1	Exact Algorithm SA
SMP2	lower bound: 2	SMH2	competitive ratio: 2 ($k = 2$) competitive ratio: $2 + \frac{2}{27}$ ($k = 3$)
SMP3	NP-hard ($k = 2$) SNP-hard (general k)	SMH3 SA_SMH3 K2SMH3 ($k = 2$)	2-approximate SA approximation ratio: $\frac{4}{3}$
SMP4	lower bound: 2	SMH4	$2 + \frac{1}{2}$ -competitive ($k = 2$)
SMP5	SNP-hard	SMH5	GA
SMP6	lower bound: 2	SMH6	competitive ratio: 2 ($k = 2$)

continued on next page

Table 9.1. Results of Algorithms for All Problems (*continued*)

Problem	Intractability	Algorithm	Analysis
SMP7	SNP-hard	SMH7	GA
SMP8	lower bound: 2	SMH8	$2 + \frac{1}{2}$ -competitive ($k = 2$)
MSP1	lower bound: $\max\{1 + \theta, 1 + \sqrt{\frac{D}{T+D}}\}$	MSH1	competitive ratio: $\max\{1 + \theta, 1 + \sqrt{\frac{D}{T+D}}\}$
MSP2	lower bound: 2	MSH2	competitive ratio: 2
MSP3	lower bound: $\max\{1 + \theta, 1 + \sqrt{\frac{D}{T+D}} - \frac{\sqrt{D(T+D)}}{(C-1)\sqrt{D(T+D)+T+D}}\}$	MSH3	competitive ratio: 2
MSP4	lower bound: $\max\{1 + \theta, 1 + \sqrt{\frac{D}{T+D}} - \frac{\sqrt{D(T+D)}}{(C-1)\sqrt{D(T+D)+T+D}}\}$	MSH4	$\max\{\frac{3}{2} + \theta, \frac{3}{2} + \sqrt{\frac{D}{T+D}}\}$ -competitive
MSP5	lower bound: $\max\{1 + \theta, 2 - \frac{1}{C}\}$	MSH5	competitive ratio: 2
MMP1	NP-hard	MMH1	–
MMP2	lower bound: 2	MMH2	–
MMP3	SNP-hard	MMH3	–
MMP4	lower bound: 2	MMH4	–
MMP5	SNP-hard	MMH5	–
MMP6	lower bound: 2	MMH6	–
MMP7	SNP-hard	MMH7	–
MMP8	lower bound: 2	MMH8	–

9.2 Future Work

First, several challenges of the on-line problems need to be resolved: (1) optimal algorithms for the number of customers more than two are still hard to be developed; (2) different machine configurations should be of interest and need to be addressed; (3) more complex situations with constraints on the transportation network should be of interest and need to be addressed.

Second, scheduling on the holistic supply chain or network needs to be studied. The concept of the holistic supply network was first elaborated in the thesis of Muddada [2010], and the first paper on this concept refers to [Wang et al., 2013a]. The feature with the holistic supply network is that (a) the configuration of a supply chain is dynamic and (b) several supply chains may cross-link.

Third, the mathematical model for the robustness of algorithms need to be constructed. This thesis presented a qualitative analysis for the robustness of algorithms but not a quantitative measurement. The mathematical model to measure the robustness in engineering system should be extended into the algorithm design field. A system is divided into two levels: infrastructure and substance [Wang, 2013]. The infrastructure system refers to the machine system in the context of production, and the substance system refers to the job which "flow" over the machine. Noises may present on the infrastructure system and/or the substance system. Operations are applied on both the infrastructure system and substance system. Operations are based on the plan and schedule. Therefore, the measure of the robustness of a schedule should be a deviation of the performance of the

system (I-S system) when the system has noise. Suppose that a underlying system has a target to achieve. Then, the performance of the operation is the closeness of the output of the system with respect to the target, which may be denoted by $P = \|A - T\|$, where A : output; T : target; P : performance. Let δ denote the noise. Then the robustness may be measured by how large δ can go given $P \leq O(P)$, $\max \delta|_{P \leq O(P)}$, where $O(P)$ is the tolerance of P . If the performance satisfaction refers to the convergence of $O \rightarrow T$, then the robustness may be measured by the largest δ to have $O \rightarrow T$, $\max \delta|_{O \rightarrow T}$. If the performance satisfaction is that the performance should satisfy X under the worst scenario, then the robustness is $\max \delta|_{\text{worst scenario satisfies } X}$. The above shows some idea of the robustness of a schedule beyond the worst case performance, and these ideas may be interesting to be closely examined, as they are close to the real situations.

Fourth, the mathematical model for resilience of algorithms should also be studied. With respect to the I-S architecture of the supply chain system, the resilience of a schedule is the ability of a schedule to recover from a disruption which could be on the infrastructure system and/or the substance system [Wang, 2013; Zhang and Lin, 2010]. The resilience of a schedule is thus examined from two angles: the performance and the disruption. There are two general types of disruptions: the infrastructure and the substance. The performance has three types as mentioned before: Suppose the disruption can be measured by something called Y . Then, the resilience of a schedule can be expressed by $\max Y|_{P \leq O(P)}$, $\max Y|_{O \rightarrow T}$ and $\max Y|_{\text{worst scenario}}$. The Y needs to be studied in future. In [Wang, 2013], the Y is defined as the cost and/or time to re-balance the supply and demand.

Last, for the problems with more than two activities (Class 3), the systematic analysis of the performance and property of algorithms is needed, as the literature has shown that analytical results are almost for the first two classes. Though this has something to do with the complexity of these problems, still more accurate results of scheduling are desired. This means to study algorithms for trade-off between accuracy and efficiency through the exploration of the properties of the problems (and thus heuristics).

LIST OF REFERENCES

- Agnetis, A., Mirchandani, P. B., Pacciarelli, D., and Pacifici, A., Nondominated Schedules for A Job-Shop with Two Competing Users, *Comput. Math. Organ. Theory*, vol. 6, no. 2, pp. 191-217, 2000.
- Agnetis, A., Mirchandani, P. B., Pacciarelli, D., and Pacifici, A., Scheduling Problems with Two Competing Agents, *Oper. Res.*, vol. 52, no. 2, pp. 229-242, 2004.
- Albers, S., Online Algorithms: A Survey, *Math. Prog.*, vol. 97, no. 1, pp. 3-26, 2003.
- Alonso-Syuso, A., Tirado, G., and Udías, Á., On A Selection and Scheduling Problem in Automatic Storage and Retrieval Warehouses, *Int. J. Prod Res.*, vol. 51, no. 17, pp. 5337-5353, 2013.
- Anderson, E. J., Nash, P., and Philpott, A. B., A Class of Continuous Network Flow Problems, *Math. Oper. Res.*, vol. 7, pp. 501-514, 1982.
- Arora, S., Nearly Linear Time Approximation Schemes for Euclidean TSP and Other Geometric Problems, *Proc. 38th IEEE FOCS*, pp. 554-563, 1997.
- Ausiello, G., Feuerstein, E., Leonardi, S., Stougie, L., and Talamo, M., Algorithms for The On-Line Traveling Salesman, *Algorithmica*, vol. 29, no. 4, pp. 560-581, 2001.
- Averbakh, I., On-line Integrated Production-distribution Scheduling Problems with Capacitated Deliveries, *Eur. J. Oper. Res.*, vol. 200, no. 2, pp. 377-384, 2010.
- Averbakh, I. and Baysan, M., Semi-online two-level supply chain scheduling problems, *J. Sched.*, vol. 15, no. 3, pp. 381-390, 2012.
- Averbakh, I. and Baysan, M., Approximation Algorithm for The On-line Multi-customer Two-level Supply Chain Scheduling Problem, *Oper. Res. Lett.*, vol. 41, no. 6, pp. 710-714, 2013a.
- Averbakh, I. and Baysan, M., Batching and Delivery in Semi-online Distribution Systems, *Discrete Appl. Math.*, vol. 161, no. 1-2, pp. 28-42, 2013b.

- Averbakh, I. and Xue, Z.-H., On-line Supply Chain Scheduling Problems with Preemption, *Eur. J. Oper. Res.*, vol. 181, no. 1, pp. 500-504, 2007.
- Baghalian, A., Rezapour, S., and Farahani, R. Z., Robust Supply Chain Network Design with Service Level against Distruptions and Demand Uncertainties: A Real-Life Case, *Eur. J. Oper. Res.*, vol. 227, no. 1, pp. 199-215, 2013.
- Baker, J. W., Schubert, M., and Faber, M. H., On The Assessment of Robustness, *Struct. Saf.*, vol. 30, no. 3, pp. 253-267, 2008.
- Baker, K. R., and Smith, J. C., A Multiple-Criterion Model for Machine Scheduling, *J. Sched.*, vol. 6, no. 1, pp. 7-16, 2003.
- Baptiste, P., Brucker, P., Chrobak, C. D., Kravchenko, S. A., and Sourd, F., The Complexity of Mean Flow Time Scheduling Problems with Release Dates, *J. Sched.*, vol. 10, no. 2, pp. 139-146, 2007.
- Baumann, N. and Köhler, E., Approximating Earliest Arrival Flows with Flow-dependent Transit Times, *Discrete Appl. Math.*, vol. 155, no. 2, pp. 161-177, 2007.
- Ben-Tal, A., El Ghaoui, L. and Nemirovski, A., Robust Optimization, *Princeton Ser. Appl. Math.*, Princeton University Press, 2009.
- Berman, P. and Karpinsk, M., 8/7-Approximation Algorithm for (1,2)-TSP, *Proc. 17th ACM-SIAM Symposium on Discrete Algorithms (SODA '06)*, pp. 641-648, 2006.
- Bertazzi, L., Paletta, G., and Speranza, M. G., Minimizing the Total Cost in an Integrated VendorManaged Inventory System, *J. Heuristics*, vol. 11 no. 5-6, pp. 393-419, 2005.
- Bertsimas, D. and Sim, M., The Price of Robustness, *Oper. Res.*, vol. 52, no. 1, pp. 35-53, 2004.
- Bérubé, J.-F., Potvin, J.-Y., and Vaucher, J., Time-dependent Shortest Paths through A Fixed Sequence of Nodes: Application to A Travel Planning Problem, *Comput. Oper. Res.*, vol. 3, no. 6, pp. 1838-1856, 2006.
- Bhattacharyya, S. P., Chapellat, H., and Keel, L. H., *Robust Control-The Parametric Approach*, Prentice Hall PTR, 2000.

- Borodin, A. and El-Yaniv, R., *Online Computation and Competitive Analysis*, Cambridge University Press, 1998.
- Burkard, R. E., Dlaska, K., and Klinz, B., The Quickest Flow Problem, *Z. Oper. Res.*, vol. 37, pp. 31-58, 1993.
- Celikbilek, C., Concurrent Optimization of Manufacturing Scheduling and Transportation Mode with Capacity Constraint, *POMS 15th Annual Conference*, 2014.
- Černý, V., Thermodynamical Approach to The Traveling Salesman Problem: An Efficient Simulation Algorithm, *J. Optimiz. Theory App.*, vol. 45, no. 1, pp. 45-51, 1985.
- Chang, Y.-C., Li, V. C., and Chiang, C.-J., An Ant Colony Optimization Heuristic for An Integrated Production and Distribution Scheduling Problem, *Eng. Optimiz.*, vol. 46, no. 4, pp. 503-520, 2014.
- Chen, B. and Vestjens, A. P. A., Scheduling on Identical Machines: How Good Is LPT in An On-line Setting, *Oper. Res. Lett.* vol. 21, no. 4, pp. 165-169, 1997.
- Chen, G. H. and Hung, Y. C., On The Quickest Path Problem, *Inform. Process. Lett.*, vol. 46, pp. 125-128, 1993.
- Chen, Y. L. and Chin, Y. H., The Quickest Path Problem, *Comput. Oper. Res.*, vol. 17, pp. 153-161, 1990.
- Chen, Z.-L., Integrated Production and Outbound Distribution Scheduling: Review and Extensions, *Oper. Res.*, vol. 58, no. 1, pp. 130-148, 2010.
- Chen, Z.-L. and Pundoor, G., Order Assignment and Scheduling in a Supply Chain, *Oper. Res.*, vol. 54, no. 3, pp. 555-572, 2006.
- Chen, Z.-L. and Pundoor, G., Integrated Order Scheduling and Packing, *Prod. Oper. Manag.*, vol. 18, no. 6, pp. 672-692, 2009.
- Chen, Z.-L. and Vairaktarakis, G. L., Integrated Scheduling of Production and Distribution Operations, *Manage. Sci.*, vol. 51, no. 4, pp. 614-628, 2005.
- Cheng, T. C. E., Gordon, V. S., and Kovalyov, M. Y., Single Machine Scheduling with Batch Deliveries, *Eur. J. Oper. Res.*, vol. 94, no. 2, pp. 277-283, 1996.

- Christofides, N., Worst-Case Analysis of A New Heuristic for The Traveling Salesman Problem, *Report 388*, Grad School of Industrial Administration, CMU, 1976.
- Condotta, A., Knust, S., Meier, D., and Shakhlevich, N. V., Tabu Search and Lower Bounds for A Combined Production/Transportation Problem, *Transport Sch.* vol. 40, no. 3, pp. 886-900, 2013.
- Cowling P. and Johansson, M., Using Real Time Information for Effective Dynamic Scheduling, *Eur. J. Oper. Res.*, vol. 139, no. 2, pp. 230-244, 2002.
- Delavar, M. R., Hajiaghaei-Keshteli, M., and Molla-Alizadeh-Zavardehi, S., Genetic Algorithms for Coordinated Scheduling of Production and Air Transportation, *Expert Syst. Appl.*, vol. 37, no. 12, pp. 8255-8266, 2010.
- Ding, G. S. and Sun, S. J., Single-Machine Scheduling Problems with Two Agents Competing for Makespan, *Lect. Notes Comput. Sc.*, vol. 6328, pp. 244-255, 2010.
- Dong, J., Zhang, A., Chen, Y., and Yang, Q., Approximation Algorithms for Two-machine Open Shop Scheduling with Batch and Delivery Coordination, *Theor. Comput. Sci.*, vol. 49, pp. 194-102, 2013.
- Dressler, D., Groß, M., Kappmeier, J.-P., Kelter, T., Kulbatzki, K., Plümpe, D., Schlechter, G., Schmidt, M., Skutella, M., and Temme, S., On the Use of Network Flow Techniques for Assigning Evacuees to Exits, *Procedida Engineering*, vol. 3, pp. 205-215, 2010.
- Du, J., Leung, J. Y.-T., and Young, G. H., Minimizing Mean Flow Time with Release Time Constraint, *Theor. Comput. Sci.*, vol. 75, no. 3, pp. 347-355, 1990.
- Fahy, R. F., An Evacuation Model for High Rise Buildings, *Proceedings of the Third International Symposium on Fire Safety Science*, Elsevier, London, 815-823, 1991.
- Fan, C., Lei, L., and Liu, S., On The Integrated Supply and Distribution Problem with Heterogeneous Vessels, *The 4th International Conference on Operations and Supply Chain Management*, 2010.
- Fan, C., Lei, L., and Liu, S., Solving a Variation of the Integrated Supply and Distribution Problem with Heterogeneous Vessels, *JOSCM*, vol. 11, no. 1, pp. 13-32, 2013.

- Fan, J., Integrated Production and Delivery Scheduling Problem with Two Competing Agents, *Advanced Management Science (ICAMS), 2010 IEEE International Conference*, vol. 2, pp. 157-160, 2010.
- Fan, J., Lu, X., and Liu, P., Integrated Scheduling of Production and Delivery On A Single Machine with Availability Constraint, *Theor. Comput. Sci.*, vol. 562, no. 11, pp. 581-589, 2015.
- Félix, M.-A. and Wagner, A., Robustness and Evolution: Concepts, Insights and Challenges from A Developmental Model System, *Heredity*, vol. 100, no. 2, pp. 132-140, 2006.
- Feng, X. and Zheng, F., Integrated Job Scheduling with Parallel-Batch Processing and Batch Deliveries, *Lect. Comput. Sc.*, vol. 8287, pp. 72-83, 2013.
- Fleischer, L., Faster Algorithms for the Quickest Transshipment Problem, *SIAM Journal on Optimization*, vol. 12, no. 1, pp. 18-35, 2001.
- Fleischer, L., Universally Maximin Flows with Piecewise-Constant Capacities, *Networks*, vol. 38, no. 3, pp. 115-125, 2001.
- Fleischer, L. and Tardos, É., Efficient Continuous-time Dynamic Network Flow Algorithms, *Oper. Res. Lett.*, vol. 23, no. 3-5, pp. 71-80, 1998.
- Frederickson, G. N., Scheduling Unit-time Tasks with Integer Release Times and Deadlines, *Inform. Process. Lett.*, vol. 16, no. 4, pp. 171-173, 1983.
- Friesen, D. K., Tighter Bounds for LPT Scheduling on Uniform Processors, *SIAM J. Comput.*, vol. 16, no. 3, pp. 554-560, 1987.
- Gao, S., *On The Integrated Production and Distribution Problems in Make-to-order Businesses*, Rutgers, The State University of New Jersey, 2011.
- Gao, S., Qi, L., and Lei, L., Integrated Batch Production and Distribution Scheduling with Limited Vehicle Capacity, *Int. J. Prod. Econ.*, vol. 160, pp. 13-25, 2015.
- Garey, M. R. and Johnson, D. S., Complexity Results for Multiprocessor Scheduling under Resource Constraints, *SIAM J. Comput.*, vol. 4, no. 4, pp. 397-411, 1975.

- Garey, M. R. and Johnson, D. S., Scheduling Tasks with Nonuniform Deadlines on Two Processors, *J. ACM*, vol. 23, no. 3, pp. 461-467, 1976.
- Garey, M. R. and Johnson, D. S., *Computers and Intractability: A Guide to the Theory of NP-Completeness*, New York: W.H. Freeman, 1979.
- Gasarch, I. W., The P=?NP Poll, *SIGACT News*, vol. 33, no. 2, pp. 34-47, 2002.
- Goetschalckx, M., Vidal, C. J., and Dogan, K., Modeling and Design of Global Logistics Systems: A Review of Integrated Strategic and Tactical Models and Design Algorithms, *Eur. J. Oper. Res.*, vol. 143, no. 1, pp. 1-18, 2002.
- Goldreich, O., *Computational Complexity: A Conceptual Perspective*, Cambridge University Press, 2008.
- Gonzalez, T., Ibarra, O. H., and Sahni, S., Bounds for LPT Schedules on Uniform Processors, *SIAM J. Comput.*, vol. 6, no. 1, pp. 155-166, 1977.
- Graham, R. L., Bounds on Multiprocessing Timing Anomalies, *SIAM J. Appl. Math.*, vol. 17, no. 2, pp. 416-429, 1969.
- Graham, R. L., Optimization and Approximation in Deterministic Sequencing and Scheduling: A Survey, *Ann. Discrete Math.*, vol. 5, pp. 287-326, 1979.
- Granville, V., Krivanek, M., and Rasson, J.-P., Simulated annealing: A proof of convergence, *IEEE T. Pattern Anal.*, vol. 16, no. 6, pp. 652-656, 1994.
- Hajiaghayi-Keshteli, M. and Aminnayeri, M., Solving The Integrated Scheduling of Production and Rail Transportation Problem by Keshtel Algorithm, *Appl. Soft Comput.*, vol. 25, pp. 184-203, 2014.
- Hajiaghayi-Keshteli, M., Aminnayeri, M., and Ghomi, S. M. T.-F., Integrated Scheduling of Production and Rail Transportation, *Comput. Ind. Eng.*, vol. 74, pp. 240-256, 2014.
- Hall, L. A. and Shmoys, D. B., Jackson's Rule for Single-machine Scheduling: Make A Good Heuristic better, *Math. Oper. Res.*, vol. 17, no. 1, pp. 22-35, 1999.
- Hall, N. G., Lesaoana, M., and Potts, C. N., Scheduling with Fixed Delivery Dates, *Oper. Res.*, vol. 49, no. 1, pp. 134-144, 2001.

- Hall, N. G. and Potts, C. N., Supply Chain Scheduling: Batching and Delivery, *Oper. Res.*, vol. 51, no. 4, pp. 566-584, 2003.
- Hall, N. G. and Potts, The Coordination of Scheduling and Batch Deliveries, *Annals of Oper. Res.*, vol. 135, no. 1, pp. 41-64, 2005.
- Hamacher, H. W., Ruzika, S., and Tjandra, S. A., Algorithms for Time-depedent Bicriteria Shortest Path Problems, *Discrete Optimiz.*, vol. 3, no. 3, pp. 239-254, 2006.
- Hamacher, H. W. and Tufekci, S., On the Use of Lexicographic Min Cost Flows in Evacuation Modeling, *Nav. Res. Log.*, vol. 34, pp. 487-503, 1987.
- Han, B., *On-line Network Scheduling in Emergency Operatoin for Medical Resources with Single-Processor Single-Destination*, M.Sc. Thesis, University of Saskatchewan, Saskatoon, Canada, 2012.
- Holland, J. H., *Adaptation in Natural and Artificial Systems*, MIT Press, 1975.
- Holling, C. S., Resilience and Stability of Ecological Systems, *Annu Rev. Ecol. Syst.*, vol. 4, pp. 1-23, 1973.
- Hong, K. S. and Leung, J. Y-T., On-Line Scheduling of Real-Time Tasks, *IEEE T. Comput.*, vol. 41, no. 10, pp. 1326-1331, 1992.
- Hu, T. C., Parallel Sequencing and Assembly Line Problems, *Oper. Res.*, vol. 9, no. 6, pp. 841-848, 1961.
- Ivanov, D., Dolgui, A., and Sokolov, B., Applicability of Optimal Control Theory to Adaptive Supply Chain Planning and Scheduling, *Annu. Rev. Contr.*, vol. 36, no. 1, pp. 73-84, 2012.
- Ivanov, D. and Sokolov, B., Dynamic Supply Chain Scheduling, *J. Sched.*, vol. 15, no. 2, pp. 201-216, 2012.
- Jackson, J. R., Scheduling A Production Line to Minimize Maximum Tardiness. Research Report 43, *Management Science Research Project*, University of California, Los Angeles, 1955.
- Ji, M., He, Y., and Cheng, T. C. E., Batch Delivery Scheduling with Batch Delivery Cost on A Single Machine, *Eur. J. Oper. Res.*, vol. 176, no. 2, pp. 745-755, 2007.

- Johnson, S. M., Optimal Two- and Three-stage Production Schedules with Setup Times Included, *Nav. Res. Log.*, vol. 1, no. 1, pp. 61-68, 1954.
- Karp, R. M., Reducibility Among Combinational Problems, *Complexity of Computer Computations*, pp. 85-103, 1972.
- Jim, E.-S. and Oron, D., Coordinating Multi-location Production and Customer Delivery, *Optim. Lett.*, vol. 7, no. 1, pp. 39-50, 2013.
- Jim, E.-S. and Oron, D., Multi-location Production and Delivery with Job Selection, *Comput. Oper. Res.*, vol. 40, no. 5, pp. 1461-1466, 2013.
- Kirkpatrick, S., Gelatt Jr, C. D., and Vecchi, M. P., Optimization by Simulated Annealing, *Science*, vol. 220, no. 5598, pp. 671-680, 1983.
- Kitano, H., Biological Robustness, *Nat. Rev. Genet.*, vol. 5, no. 11, pp. 826-837, 2004.
- Knuth, D. E., Big Omicron and Big Omega and Big Theta, *ACM SIGACT News*, vol. 8, no. 2, pp. 18-24, 1976.
- Köhler, E. and Skutella, M., Flows over Time with Load-Dependent Transit Times, *SIAM J. Optimiz.*, vol. 15, no. 4, pp. 1185-1202, 2006.
- Lan, L. W., Chiou, Y.-C., Lin, Z.-S., and Hsu, C.-C., Cellular Automaton Simulations for Mixed Traffic with Erratic Motorcycles' Behaviours, *Physica A: Statistical Mechanics and its Applications*, vol. 389, no. 10, pp. 2077-2089, 2010.
- Lárraga, M. E., del Río, J. A., and Alvarez-Icaza, L., Cellular Automata for One-lane Traffic Flow Modeling, *Transportation Res. C-Emer.*, vol. 13, no. 1, pp. 63-74, 2005.
- Lee, C.-Y. and Chen, Z.-L., Machine Scheduling with Transportation Considerations, *J. Sched.*, vol. 4, pp. 3-24, 2001.
- Lee, C.-Y., Çetinkaya, S., and Jaruphongsa, W., A Dynamic Model for Inventory Lot Sizing and Outbound Shipment Scheduling at a Third-Party Warehouse, *Oper. Res.*, vol. 51, no. 5, pp. 735-747, 2003.
- Lee, I. S. and Yoon, S. H., Coordinated Scheduling of Production and Delivery Stages with Stage-dependent Inventory Holding Costs, *Omega*, 38(6) (2010) 509-521.

- Lee, K., Choi, B.-C., Leung, J. Y.-T., Pinedo, M. L., and Briskorn, D., Minimizing The Total Weighted Delivery Time in Container Transportation Scheduling, *Nav. Res. Log.*, 59(3-4) (2012) 266-277.
- Lee, K., Lei, L., and Dong, H., A Solvable Case of Emergency Supply Chain Scheduling Problem with Multi-stage Lead Times, *JOSCM*, vol. 11, no. 2, pp. 30-45, 2013.
- Li, C.-L. and Ou, J., Machine Scheduling with Pickup and Delivery, *Nav. Res. Log.*, vol. 52, no. 7), pp. 617-630, 2005.
- Li, C.-L. and Ou, J., Coordinated Scheduling of Customer Orders with Decentralized Machine Locations, *IIE Trans.*, vol. 39, no. 9, pp. 899-909, 2007.
- Li, C.-L., Vairaktarakis, G., and Lee, C.-Y., Machine Scheduling with Deliveries to Multiple Customer Locations, *Eur. J. Oper. Res.*, vol. 164, no. 1, pp. 39-51, 2005.
- Li, S. and Li, M., Integrated Production and Distribution Scheduling Problems Related with Fixed Delivery Departure Dates and Number of Late Orders, *. Inequal. Appl.*, (Online), DOI: 10.1186/1029-242X-2014-409, 2014.
- Li, Q., Zhang, W. J., and Chen, L., Design for Control (DFC): a Concurrent Engineering Approach for Mechatronic System Design, *IEEE/ASME Trans. Mechatron.*, vol. 6, no. 2, pp. 161-169, 2001.
- Li, S., Yuan, J., and Fan, B., Unbounded Parallel-batch Scheduling with Family Jobs and Delivery Coordination, *Inform. Process. Lett.*, vol. 111, no. 12, pp. 575-582, 2011.
- Lim, G. J., Zangeneh, S., Baharnemati, M. R., and Assavapokee, T., A Capacitated Network Flow Optimization Approach for Short Notice Evacuation Planning, *Eur. J. Oper. Res.*, vol. 223, no. 1, pp. 234-245, 2012.
- Lin, P., Lo, S. M., Huang, H. C., and Yuen, K. K., On The Use of Multi-stage Time-varying Quickest Time Approach for Optimization of Evacuation Planning, *Fire Safety J.*, vol. 43, pp. 282-290, 2008.
- Lipmann, M., *On-line Routing*, Phd thesis, Technical University of Eindhoven, 2003.
- Liu, Q., Wan, L., and Wei, L., Online Scheduling on a Single Machine with Grouped Processing Times, *Discrete Dyn. Nat. Soc.*, (Online), Article ID 805294, 2014.

- Mazdeh, M. M., Sarhadi, M. and Hindi, K. S., A Branch-and-bound Algorithm for Single-machine Scheduling with Batch Delivery Minimizing Flow Times and Delivery Costs, *Eur. J. Oper. Res.*, vol. 183, no. 1, pp. 74-86, 2007.
- Mazdeh, M. M., Shashaani, S., Ashouri, A., and Hindi, K. S., Single-machine Batch Scheduling Minimizing Weighted Flow Times and Delivery Costs, *Appl. Math. Model.*, vol. 35, no. 1, pp. 563-570, 2011.
- McNaughton, R., Scheduling with Deadlines and Loss Functions, *Manag. Sci.*, vol. 6, no. 1, pp. 1-12, 1959.
- Meinecke, C. and Scholz-Reiter, B. A Heuristic for the Integrated Production and Distribution Scheduling Problem, *IJIRSE* vol. 8, no. 2, pp. 278-285, 2014.
- Mor, B. and Mosheiov, G., Sheduling Problems with Two Competing Agents to Minimize Minmax and Minsum Earliness Measures, *Eur. J. Oper. Res.*, vol. 206, vol. 3, pp. 540-546, 2010.
- Muddada, R. R., *Towards Resilient Supply Chain Networks*, M.Sc. Thesis, University of Saskatchewan, Saskatoon, Canada, 2010.
- Nagel, K. and Schreckenberg, M., A Cellular Automaton Model for Freeway Traffic, *J. Phys. I France*, vol. 2, pp. 2221-2229, 1992.
- Ng, C. T., Cheng, T. C. E., and Yuan, J. J., A Note on The Complexity of The Problem of Two-Agent Scheduling on A Single Machine, *J. Comb. Optim.*, vol.12, no. 4, pp. 387-394, 2006.
- Ng, C. T. and Lu, L., On-line Integrated Production and Outbound Distribution Scheduling to Minimize The Maximum Delivery Completion Time, *J. Sched.*, vol. 15, no. 3, pp. 391-398, pp. 2012.
- Nong, Q. Q., Yuan, J. J., Fu, R. Y., Lin, L., and Tian, J., The Single-machine Parallel Batching On-line Scheduling Problem with Family Jobs to Minimize Makespan, *Int. J. Prod. Econ.*, vol. 111, no. 2, pp. 435-440, 2008.
- Ogier, R. G., Minimu-Delay Routing in Continuous-Time Dynamic Networks with Piecewise-Constant Capacities. *Networks*, vol. 18, pp. 303-318, 1988.

- Orponen, P. and Mannila, H., On Approximation Preserving Reductions: Complete Problems and Robust Measures, *Technical Report*, University of Helsinki, 1990.
- Otto, C. and Pipe, A., Remote, Mobile Telemedicine: The Satellite Transmission of Medical Data from Mount Logan, *J. Telemed. Telecare.*, vol. 3, suppl 1, pp. 84-85, 1997.
- Pei, J., Liu, X., Pardalos, P. M., Fan, W., Wang, L., and Yang, S., Solving A Supply Chain Scheduling Problem with Non-identical Job Sizes and Release Times by Applying A Novel Effective Heuristic Algorithm, *Int. J. Syst. Sci.*, (Online), DOI: 10.1080/00207721.2014.902553, 2014.
- Pei, J., Liu, X., Pardalos, P. M., Fan, W., Wang, L., and Yang, S., Application of An Effective Modified Gravitational Search Algorithm for The Coordinated Scheduling Problem in A Two-stage Supply Chain, *Int. J. Adv. Manuf. Technol.*, vol. 70, no. 1-4, pp. 335-348, 2014.
- Philpott, A. B., Continuous-Time Flows in Networks, *Math. Oper. Res.*, vol. 15, no. 4, pp. 640-661, 1990.
- Potts, C. N., Analysis of A Heuristic for One Machine Sequencing with Release Dates and Delivery Times, *Oper. Res.*, vol. 28, no. 6, pp. 1436-1441, 1980.
- Pruhs, K., Sgall, J. and Torng, E., *Handbook of scheduling: Algorithms, Models, and Performance Analysis*, CRC Press, 2004.
- Pundoor, G. and Chen, Z.-L., Scheduling A Production-distribution System to Optimize The Tradeoff between Delivery Tardiness and Total Distribution Cost, *Nav. Res. Log.*, vol. 52, no. 6, pp. 571-589, 2005.
- Qi, X., A Logistics Scheduling Model: Inventory Cost Reduction by Batching, *Nav. Res. Log.*, vol. 52, no. 4, pp. 312-320, 2005.
- Rainey, L. B. and Andreas, T., *Modeling and Simulation Support for System of Systems Engineering Applications*, John Wiley & Sons, 2015.
- Rasti-Barzoki, M. and Hejazi, S. R., Minimizing The Weighted Number of Tardy Jobs with Due Date Assignment and Capacity-constrained Deliveries for Multiple Customers in Supply Chains, *Eur. J. Oper. Res.*, vol. 228, no. 2, pp. 345-357, 2013.

- Rasti-Barzoki, M., Hejazi, S. R., and Mazdeh, M. M., A Branch and Bound Algorithm to Minimize the Total Weighed Number of Tardy Jobs and Delivery Costs, *Appl. Math. Model.*, vol. 37, no. 7, pp. 4927-4937, 2013.
- Ray, L. R. and Stengel, R. F., Stochastic Robustness of Linear-Time-Invariant Control Systems, *IEEE Trans. Automat. Contr.*, vol. 36. no. 1, pp. 82-82, 1991.
- Rosenberger, J., P vs. NP Poll Results, *Commun. ACM*, vol. 55, no. 5, pp. 10, 2012.
- Sarmiento, A. M. and Nagi, R., A Review of Intergrated Analysis of Production-distribution Systems, *IIE Trans.*, vol. 31, no. 1, pp. 1061-1074, 1999.
- Sawik, T., Coordinated supply chain scheduling, *Int. J. Prod. Econ.*, vol. 120, no. 2, pp. 437-451, 2009.
- Selvarajah, E. and Zhang, R., Supply Chain Scheduling to Minimize Holding costs with Outsourcing, *Ann. Oper. Res.*, vol. 217, no. 1, pp. 479-490, 2014.
- Schiller, J. H., *Mobile Communications (Second Edition)*, Addison-Wesley, 2003.
- Schmitt, L. M., Theory of Genetic Algorithms II: models for genetic operators over the string-tensor representation of populations and convergence to global optima for arbitrary fitness function under scaling, *Theor. Comput. Sci.*, vol. 310, no. 1-3, pp. 181-231, 2004.
- Shirvani, N. and Shadrokh, S., Coordination of a Cyclic Three-stage Supply Chain for Fast Moving Consumer Goods, *Iranian Journal of Operations Research*, vol. 4, no. 2, pp. 175-190, 2013.
- Smith, W. E., Various Optimizers for Single Stage Production, *Nav. Res. Log.*, vol. 3, no. 1-2, pp. 59-66, 1956.
- Sniedovich, M., *Dynamic Programming: Foundations and Principles*, CRC Press, 2010.
- Sørensen, J. D., Framework for Robustness Assessment of Timber Structures, *Eng. Struct.*, vol. 33, no. 11, pp. 3087-3092, 2011.
- Steiner, G. and Zhang, R., Approximation Algorithms for Minimizing The Total Weighted Number of Late Jobs with Late Deliveries in Two-level Supply Chains, *J. Sched.*, vol. 12, no. 6, pp. 565-574, 2009.

- Steiner, G. and Zhang, R., Minimizing The Weighted Number of Tardy Jobs with Due Date Assignment and Capacity-constrained Deliveries, *Ann. Oper. Res.*, vol. 191, no. 1, pp. 171-181, 2011.
- Takizawa, A., Inoue, M., and Katoh, N., An Emergency Evacuation Planning Model Using the Universally Quickest Flow, *Rev. Socionetwork Strat.*, vol. 6, no. 1, pp. 15-28, 2012.
- Tang, L., Gong, H., Liu, J., and Li, F., Bicriteria Scheduling on A Single Batching Machine with Job Transportation and Deterioration Considerations, *Nav. Res. Log.*, vol. 61, no. 4, pp. 269-285, 2014.
- Ullrich, C. A., Supply Chain Scheduling: Makespan Reduction Potential, *Int. J. logist.*, vol. 15, no. 5, pp. 323-336, 2012.
- Ullrich, C. A., Integrated Machine Scheduling and Vehicle Routing with Time Windows, *Eur. J. Oper. Res.*, vol. 227, no. 1, pp. 152-165, 2013.
- Viswanadham, N., The Past, Present, and Future of Supply-Chain Automation, *IEEE Robot. and Autom. Mag.*, vol. 9, no. 2, pp. 48-56, (2002).
- Wan, L. and Zhang, A., Coordinated Scheduling on Parallel Machines with Batch Delivery, *Int. J. Prod. Econ.*, vol. 150, pp. 199-203, 2014.
- Wang, D., Grunder, O., and Moudni, A. E., Using Genetic Algorithm for Lot Sizing and Scheduling Problem with Arbitrary Job Volumes and Distinct Job Due Date Considerations, *Int. J. of Sys. Sci.*, vol. 45, no. 8, pp. 1694-1707, 2014.
- Wang, G., Vakati, R. S. R., Leund, J. Y.-T., and Pinedo, M., Scheduling Two Agents with Controllable Processing Times. *Eur. J. Oper. Res.*, vol. 205, pp. 528-539, 2010.
- Wang, J. W., *Towards A Resilient Networked Service System, in the Department of Mechanical Engineering*, Ph.D. Thesis, University of Saskatchewan, Saskatoon, Canada, 2013.
- Wang, J. W., Ip, W. H., Muddada, R. R., Huang, J. L., and Zhang, W. J., On Petri Net Implementation of Proactive Resilient Holistic Supply Chain Networks, *Int. J. Adv. Manuf. Tech.*, vol. 69, no. 1-4, pp. 427-437, 2013a.

- Wang, J. W., Wang H. F., Zhang W. J., Ip, W. H., and Furuta, K., Evacuation Planning Based on the Contraflow technique with Consideration of Evacuation Priorities and Traffic Setup Time, *IEEE Intell. Transp. Syst.*, vol. 14, no. 1, pp. 480-485, 2013b.
- Wang, X. L. and Cheng, T. C. E., Machine Scheduling with An Availability Constraint and Job Delivery Coordination, *Nav. Res. Log.*, vol. 54, no. 1, pp. 11-20, 2007.
- Wang, X. L. and Cheng, T. C. E., Production Scheduling with Supply and Delivery Considerations to Minimize The Makespan, *Eur. J. Oper. Res.*, vol. 194, no. 3, pp.743-752, 2009a.
- Wang, X. L. and Cheng, T. C. E., Logistics Scheduling to Minimize Inventory and Transport Costs, *Int. J. Prod. Econ.*, vol. 121, no. 1, pp. 266-273, 2009b.
- Yamada, T., A Network Approach to A City Emergency Evacuation Planning, *Int. J. Sys. Sci.*, vol. 27, no. 10, pp. 931-936, 1996.
- Yang, X., Scheduling with Generalized Batch Delivery Dates and Earliness Penalties, *IIE Trans.*, vol. 32, no. 8, pp. 735-741, 2000.
- Yeung, W.-K., Choi, T.-M., and Cheng, T. C. E., Supply Chain Scheduling and Coordination with Dual Delivery Modes and Inventory Storage Cost, *Int. J. Prod. Econ.*, vol. 132, no. 2, pp. 223-229, 2011.
- Yuan, J., Soukhal, A., Chen, Y., and Lu, L., A Note on The Complexity of Flow Shop Scheduling with Transportation Constraints, *Eur. J. Oper. Res.*, vol. 178, no. 3, pp. 918-925, 2007.
- Zdrzalka, S., Approximation Algorithms for Single-machine Sequencing with Delivery Times and Unit Batch Set-up Times, *Eur. J. Oper. Res.*, vol. 51, no. 2, pp. 199-209, 1991.
- Zdrzalka, S., Preemptive Scheduling with Release Dates, Delivery Times and Sequence Independent Setup Times, *Eur. J. Oper. Res.*, vol. 76, no. 1, pp. 60-71, 1994.
- Zdrzalka, S., Analysis of Approximation Algorithms for Single-machine Scheduling with Delivery Times and Sequence Independent Batch Setup Times, *Eur. J. Oper. Res.*, vol. 80, no. 2, pp. 371-380, 1995.

- Zhang, W. J., Is Resilience The Destiny for Safety Management Paradigm? *Presentation*, Northeastern University, China, 2007.
- Zhang, W. J., Resilience Engineering, *Seminar*, Chinese Natural Science Foundation, 2008.
- Zhang, W. J., Towards a New Paradigm of Theory for Having a Dynamic System: Operation Management Through Design (OMTD), *Seminar Presentation at Shanghai University, China*, www.homepage.usask.ca/~wjz485 (publication, other), PPT, 2010.
- Zhang, W. J. and Lin, Y., Principles of Design of Resilient Systems and its Application to Enterprise Information Systems, *Enterp. Inf. Syst.*, vol.4, no. 2, pp. 99-110, 2010.
- Zhang, W. J., Liu, X., and Van Luttervelt, C. A., A New Theory, Methodology and Computer Aid for Manufacturing Systems Design with Special Reference to Partner Factories Selection in Virtual Enterprises, *Proceedings of Int. Conf. of World Manufacturing Congress 97*, pp. 61-66, 1997.
- Zhong, W.-Y. and Lv, L.-H., Hybrid Flowshop Scheduling with Interstage Job Transportation, *J. Oper. Res. Soc. China*, vol. 2, no. 1, pp. 109-121, 2014.

APPENDIX A
LOWER BOUND FOR SMP2

Consider the performance of an arbitrary on-line algorithm H for the following instance. For Customer i , the instance releases a job with zero processing time at time $r_1^{(i)} = 0$, and if the algorithm H delivers the job at time $\rho_1^{(i)} \geq D$, then there are no jobs coming for customer i . Otherwise, the second job with zero processing time arrives at time $r_2^{(i)} = D$, and if the departure time of this job $\rho_2^{(i)} \geq 2D$, then there are no jobs coming, otherwise, the third job comes at time $r_3^{(i)} = 2D$, and so on. If the algorithm H delivers the j th job with zero preparation time at time $\rho_j^{(i)} \geq jD$, then there are no jobs coming, or the $(j + 1)$ th job with zero processing time comes at time $r_{j+1}^{(i)} = jD$. The process is repeated until at most N jobs have been released and delivered (see Figure 10.1).

If the instance at last has released and delivered l jobs for Customer i , where $l < N$, then the l jobs are delivered in l different batches and $D_l^{(i)} = \rho_l^{(i)} \geq lD$. So there is $D_{max}^{(i)}(\eta) + TC^{(i)}(\eta) = D_l^{(i)}(\eta) + lD \geq 2lD$, where η is the schedule obtained by the algorithm H and $TC^{(i)}(\eta)$ is the delivery cost for Customer i in η . The optimal schedule delivers all the jobs in a batch at time $(l-1)D$ and there is $D_{max}^{(i)}(opt) + TC^{(i)}(opt) = (l-1)D + D = lD$. Therefore, $\frac{D_{max}^{(i)}(\eta) + TC^{(i)}(\eta)}{D_{max}^{(i)}(opt) + TC^{(i)}(opt)} \geq 2$.

If the instance at last has released and delivered N jobs for Customer i , the N jobs are delivered in N batches and $D_N^{(i)} \geq r_N = (N-1)D$. So there is $D_{max}^{(i)}(\eta) + TC^{(i)}(\eta) = D_N^{(i)}(\eta) + ND \geq (2N-1)D$, where η is the schedule obtained by the algorithm H and $TC^{(i)}(\eta)$ is the delivery cost for Customer i in η . The optimal schedule delivered all the requirements in a batch at time $(N-1)D$ and there is $D_{max}^{(i)}(opt) + TC^{(i)}(opt) = (N-1)D + D = ND$. As N gets infinitely large, $\frac{D_{max}^{(i)}(\eta) + TC^{(i)}(\eta)}{D_{max}^{(i)}(opt) + TC^{(i)}(opt)}$ will tend to 2.

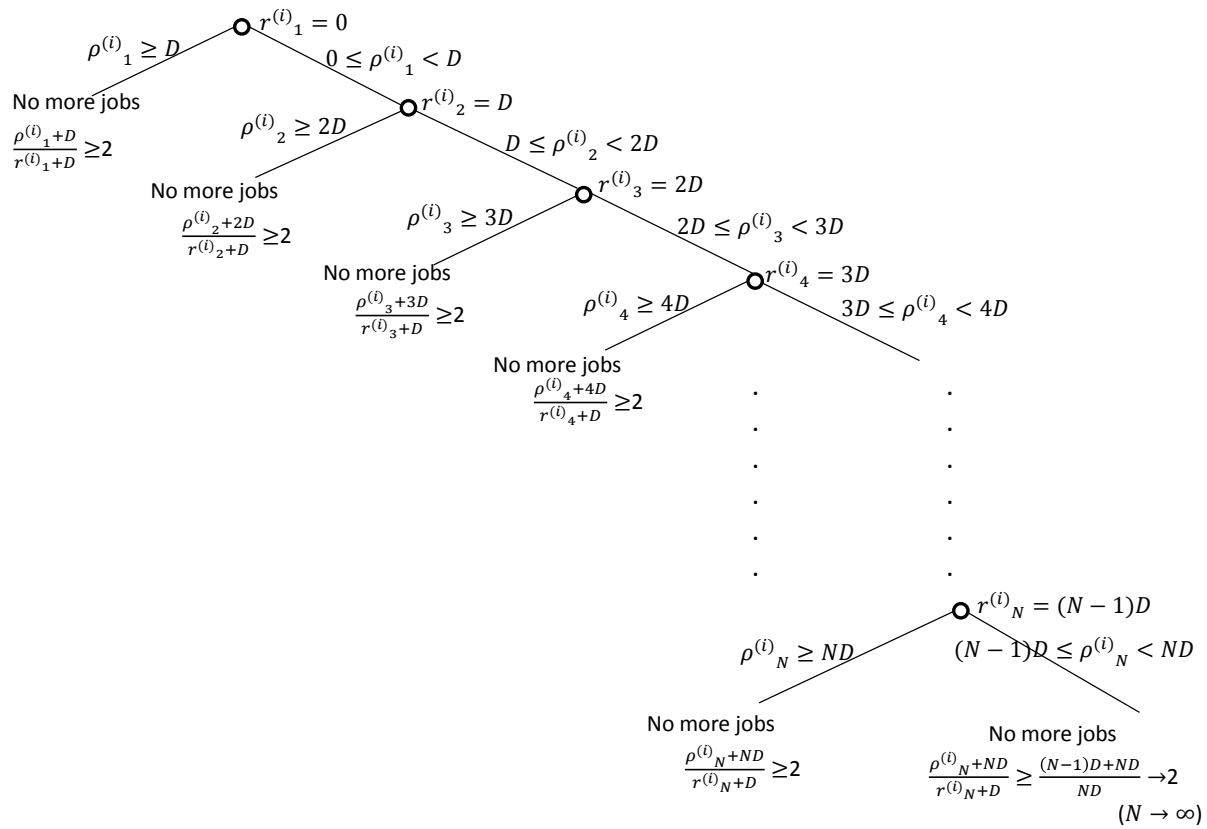


Figure 10.1. The Construction of Lower Bound for SMP2

Therefore, one can conclude $\frac{\sum_i^k D_{max}^{(i)}(\eta) + TC(\eta)}{\sum_i^k D_{max}^{(i)}(opt) + TC^{(i)}(opt)}$ cannot be less than 2, which completes the proof.

APPENDIX B
COPYRIGHT PERMISSIONS

The copyright permissions of the following published papers presented in the thesis:

Han, B., Zhang, W. J., Lu, X. W., and Lin, Y., On-line Supply Chain Scheduling for Single-Machine and Parallel-Machine Configurations with a Single Customer: Minimizing the Makespan and Delivery Cost, *Eur. J. Oper. Res.*, vol. 244, no. 3, pp. 704-714, 2015.

are given in the following pages:

ELSEVIER LICENSE TERMS AND CONDITIONS

Jul 20, 2015

This is a License Agreement between University of Saskatchewan -- Bin Han ("You") and Elsevier ("Elsevier") provided by Copyright Clearance Center ("CCC"). The license consists of your order details, the terms and conditions provided by Elsevier, and the payment terms and conditions.

All payments must be made in full to CCC. For payment instructions, please see information listed at the bottom of this form.

Supplier	Elsevier Limited The Boulevard, Langford Lane Kidlington, Oxford, OX5 1GB, UK
Registered Company Number	1982084
Customer name	University of Saskatchewan
Customer address	57 campus drive saskatoon, SK S7N 5B4
License number	3673201197854
License date	Jul 20, 2015
Licensed content publisher	Elsevier
Licensed content publication	European Journal of Operational Research
Licensed content title	On-line supply chain scheduling for single-machine and parallel-machine configurations with a single customer: Minimizing the makespan and delivery cost
Licensed content author	Bin Han, Wenjun Zhang, Xiwen Lu, Yingzi Lin
Licensed content date	1 August 2015
Licensed content volume number	244
Licensed content issue number	3
Number of pages	11
Start Page	704
End Page	714
Type of Use	reuse in a thesis/dissertation
Portion	full article
Format	both print and electronic
Are you the author of this Elsevier article?	Yes
Will you be translating?	No
Title of your thesis/dissertation	SUPPLY CHAIN SCHEDULING FOR MULTI-MACHINES AND MULTI-CUSTOMERS
Expected completion date	Aug 2015
Estimated size (number of	290 276

pages)

Elsevier VAT number	GB 494 6272 12
Permissions price	0.00 CAD
VAT/Local Sales Tax	0.00 CAD / 0.00 GBP
Total	0.00 CAD

[Terms and Conditions](#)

INTRODUCTION

1. The publisher for this copyrighted material is Elsevier. By clicking "accept" in connection with completing this licensing transaction, you agree that the following terms and conditions apply to this transaction (along with the Billing and Payment terms and conditions established by Copyright Clearance Center, Inc. ("CCC"), at the time that you opened your Rightslink account and that are available at any time at <http://myaccount.copyright.com>).

GENERAL TERMS

2. Elsevier hereby grants you permission to reproduce the aforementioned material subject to the terms and conditions indicated.

3. Acknowledgement: If any part of the material to be used (for example, figures) has appeared in our publication with credit or acknowledgement to another source, permission must also be sought from that source. If such permission is not obtained then that material may not be included in your publication/copies. Suitable acknowledgement to the source must be made, either as a footnote or in a reference list at the end of your publication, as follows:

"Reprinted from Publication title, Vol /edition number, Author(s), Title of article / title of chapter, Pages No., Copyright (Year), with permission from Elsevier [OR APPLICABLE SOCIETY COPYRIGHT OWNER]." Also Lancet special credit - "Reprinted from The Lancet, Vol. number, Author(s), Title of article, Pages No., Copyright (Year), with permission from Elsevier."

4. Reproduction of this material is confined to the purpose and/or media for which permission is hereby given.

5. Altering/Modifying Material: Not Permitted. However figures and illustrations may be altered/adapted minimally to serve your work. Any other abbreviations, additions, deletions and/or any other alterations shall be made only with prior written authorization of Elsevier Ltd. (Please contact Elsevier at permissions@elsevier.com)

6. If the permission fee for the requested use of our material is waived in this instance, please be advised that your future requests for Elsevier materials may attract a fee.

7. Reservation of Rights: Publisher reserves all rights not specifically granted in the combination of (i) the license details provided by you and accepted in the course of this licensing transaction, (ii) these terms and conditions and (iii) CCC's Billing and Payment terms and conditions.

8. License Contingent Upon Payment: While you may exercise the rights licensed immediately upon issuance of the license at the end of the licensing process for the transaction, provided that you have disclosed complete and accurate details of your proposed use, no license is finally effective unless and until full payment is received from you (either

by publisher or by CCC) as provided in CCC's Billing and Payment terms and conditions. If full payment is not received on a timely basis, then any license preliminarily granted shall be deemed automatically revoked and shall be void as if never granted. Further, in the event that you breach any of these terms and conditions or any of CCC's Billing and Payment terms and conditions, the license is automatically revoked and shall be void as if never granted. Use of materials as described in a revoked license, as well as any use of the materials beyond the scope of an unrevoked license, may constitute copyright infringement and publisher reserves the right to take any and all action to protect its copyright in the materials.

9. **Warranties:** Publisher makes no representations or warranties with respect to the licensed material.

10. **Indemnity:** You hereby indemnify and agree to hold harmless publisher and CCC, and their respective officers, directors, employees and agents, from and against any and all claims arising out of your use of the licensed material other than as specifically authorized pursuant to this license.

11. **No Transfer of License:** This license is personal to you and may not be sublicensed, assigned, or transferred by you to any other person without publisher's written permission.

12. **No Amendment Except in Writing:** This license may not be amended except in a writing signed by both parties (or, in the case of publisher, by CCC on publisher's behalf).

13. **Objection to Contrary Terms:** Publisher hereby objects to any terms contained in any purchase order, acknowledgment, check endorsement or other writing prepared by you, which terms are inconsistent with these terms and conditions or CCC's Billing and Payment terms and conditions. These terms and conditions, together with CCC's Billing and Payment terms and conditions (which are incorporated herein), comprise the entire agreement between you and publisher (and CCC) concerning this licensing transaction. In the event of any conflict between your obligations established by these terms and conditions and those established by CCC's Billing and Payment terms and conditions, these terms and conditions shall control.

14. **Revocation:** Elsevier or Copyright Clearance Center may deny the permissions described in this License at their sole discretion, for any reason or no reason, with a full refund payable to you. Notice of such denial will be made using the contact information provided by you. Failure to receive such notice will not alter or invalidate the denial. In no event will Elsevier or Copyright Clearance Center be responsible or liable for any costs, expenses or damage incurred by you as a result of a denial of your permission request, other than a refund of the amount(s) paid by you to Elsevier and/or Copyright Clearance Center for denied permissions.

LIMITED LICENSE

The following terms and conditions apply only to specific license types:

15. **Translation:** This permission is granted for non-exclusive world **English** rights only unless your license was granted for translation rights. If you licensed translation rights you may only translate this content into the languages you requested. A professional translator must perform all translations and reproduce the content word for word preserving the integrity of the article. If this license is to re-use 1 or 2 figures then permission is granted for non-exclusive world rights in all languages.

16. **Posting licensed content on any Website:** The following terms and conditions apply as

follows: Licensing material from an Elsevier journal: All content posted to the web site must maintain the copyright information line on the bottom of each image; A hyper-text must be included to the Homepage of the journal from which you are licensing at <http://www.sciencedirect.com/science/journal/xxxxx> or the Elsevier homepage for books at <http://www.elsevier.com>; Central Storage: This license does not include permission for a scanned version of the material to be stored in a central repository such as that provided by Heron/XanEdu.

Licensing material from an Elsevier book: A hyper-text link must be included to the Elsevier homepage at <http://www.elsevier.com>. All content posted to the web site must maintain the copyright information line on the bottom of each image.

Posting licensed content on Electronic reserve: In addition to the above the following clauses are applicable: The web site must be password-protected and made available only to bona fide students registered on a relevant course. This permission is granted for 1 year only. You may obtain a new license for future website posting.

17. **For journal authors:** the following clauses are applicable in addition to the above:

Preprints:

A preprint is an author's own write-up of research results and analysis, it has not been peer-reviewed, nor has it had any other value added to it by a publisher (such as formatting, copyright, technical enhancement etc.).

Authors can share their preprints anywhere at any time. Preprints should not be added to or enhanced in any way in order to appear more like, or to substitute for, the final versions of articles however authors can update their preprints on arXiv or RePEc with their Accepted Author Manuscript (see below).

If accepted for publication, we encourage authors to link from the preprint to their formal publication via its DOI. Millions of researchers have access to the formal publications on ScienceDirect, and so links will help users to find, access, cite and use the best available version. Please note that Cell Press, The Lancet and some society-owned have different preprint policies. Information on these policies is available on the journal homepage.

Accepted Author Manuscripts: An accepted author manuscript is the manuscript of an article that has been accepted for publication and which typically includes author-incorporated changes suggested during submission, peer review and editor-author communications.

Authors can share their accepted author manuscript:

- immediately
 - via their non-commercial person homepage or blog
 - by updating a preprint in arXiv or RePEc with the accepted manuscript
 - via their research institute or institutional repository for internal institutional uses or as part of an invitation-only research collaboration work-group
 - directly by providing copies to their students or to research collaborators for their personal use

- for private scholarly sharing as part of an invitation-only work group on commercial sites with which Elsevier has an agreement
- after the embargo period
 - via non-commercial hosting platforms such as their institutional repository
 - via commercial sites with which Elsevier has an agreement

In all cases accepted manuscripts should:

- link to the formal publication via its DOI
- bear a CC-BY-NC-ND license - this is easy to do
- if aggregated with other manuscripts, for example in a repository or other site, be shared in alignment with our hosting policy not be added to or enhanced in any way to appear more like, or to substitute for, the published journal article.

Published journal article (JPA): A published journal article (PJA) is the definitive final record of published research that appears or will appear in the journal and embodies all value-adding publishing activities including peer review co-ordination, copy-editing, formatting, (if relevant) pagination and online enrichment.

Policies for sharing publishing journal articles differ for subscription and gold open access articles:

Subscription Articles: If you are an author, please share a link to your article rather than the full-text. Millions of researchers have access to the formal publications on ScienceDirect, and so links will help your users to find, access, cite, and use the best available version.

Theses and dissertations which contain embedded PJAs as part of the formal submission can be posted publicly by the awarding institution with DOI links back to the formal publications on ScienceDirect.

If you are affiliated with a library that subscribes to ScienceDirect you have additional private sharing rights for others' research accessed under that agreement. This includes use for classroom teaching and internal training at the institution (including use in course packs and courseware programs), and inclusion of the article for grant funding purposes.

Gold Open Access Articles: May be shared according to the author-selected end-user license and should contain a [CrossMark logo](#), the end user license, and a DOI link to the formal publication on ScienceDirect.

Please refer to Elsevier's [posting policy](#) for further information.

18. **For book authors** the following clauses are applicable in addition to the above: Authors are permitted to place a brief summary of their work online only. You are not allowed to download and post the published electronic version of your chapter, nor may you scan the printed edition to create an electronic version. **Posting to a repository:** Authors are permitted to post a summary of their chapter only in their institution's repository.

19. **Thesis/Dissertation:** If your license is for use in a thesis/dissertation your thesis may be submitted to your institution in either print or electronic form. Should your thesis be published commercially, please reapply for permission. These requirements include

permission for the Library and Archives of Canada to supply single copies, on demand, of the complete thesis and include permission for Proquest/UMI to supply single copies, on demand, of the complete thesis. Should your thesis be published commercially, please reapply for permission. Theses and dissertations which contain embedded PJAs as part of the formal submission can be posted publicly by the awarding institution with DOI links back to the formal publications on ScienceDirect.

Elsevier Open Access Terms and Conditions

You can publish open access with Elsevier in hundreds of open access journals or in nearly 2000 established subscription journals that support open access publishing. Permitted third party re-use of these open access articles is defined by the author's choice of Creative Commons user license. See our [open access license policy](#) for more information.

Terms & Conditions applicable to all Open Access articles published with Elsevier:

Any reuse of the article must not represent the author as endorsing the adaptation of the article nor should the article be modified in such a way as to damage the author's honour or reputation. If any changes have been made, such changes must be clearly indicated.

The author(s) must be appropriately credited and we ask that you include the end user license and a DOI link to the formal publication on ScienceDirect.

If any part of the material to be used (for example, figures) has appeared in our publication with credit or acknowledgement to another source it is the responsibility of the user to ensure their reuse complies with the terms and conditions determined by the rights holder.

Additional Terms & Conditions applicable to each Creative Commons user license:

CC BY: The CC-BY license allows users to copy, to create extracts, abstracts and new works from the Article, to alter and revise the Article and to make commercial use of the Article (including reuse and/or resale of the Article by commercial entities), provided the user gives appropriate credit (with a link to the formal publication through the relevant DOI), provides a link to the license, indicates if changes were made and the licensor is not represented as endorsing the use made of the work. The full details of the license are available at <http://creativecommons.org/licenses/by/4.0>.

CC BY NC SA: The CC BY-NC-SA license allows users to copy, to create extracts, abstracts and new works from the Article, to alter and revise the Article, provided this is not done for commercial purposes, and that the user gives appropriate credit (with a link to the formal publication through the relevant DOI), provides a link to the license, indicates if changes were made and the licensor is not represented as endorsing the use made of the work. Further, any new works must be made available on the same conditions. The full details of the license are available at <http://creativecommons.org/licenses/by-nc-sa/4.0>.

CC BY NC ND: The CC BY-NC-ND license allows users to copy and distribute the Article, provided this is not done for commercial purposes and further does not permit distribution of the Article if it is changed or edited in any way, and provided the user gives appropriate credit (with a link to the formal publication through the relevant DOI), provides a link to the license, and that the licensor is not represented as endorsing the use made of the work. The full details of the license are available at <http://creativecommons.org/licenses/by-nc-nd/4.0>. Any commercial reuse of Open Access articles published with a CC BY NC SA or CC BY NC ND license requires permission from Elsevier and will be subject to a fee.

Commercial reuse includes:

- Associating advertising with the full text of the Article
- Charging fees for document delivery or access
- Article aggregation
- Systematic distribution via e-mail lists or share buttons

Posting or linking by commercial companies for use by customers of those companies.

20. Other Conditions:

v1.7

Questions? customercare@copyright.com or +1-855-239-3415 (toll free in the US) or +1-978-646-2777.
