ON-LINE NETWORK SCHEDULING IN EMERGENCY OPERATION FOR MEDICAL RESOURCES WITH SINGLE-PROCESSOR SINGLE-DESTINATION

A Thesis Submitted to the College of

Graduate Studies and Research

In Partial Fulfillment of the Requirements

For the Degree of Master of Science

In the Division of Biomedical Engineering

University of Saskatchewan

Saskatoon

By

BIN HAN

PERMISSION TO USE

In presenting this thesis in partial fulfilment of the requirements for a Master 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

Saskatoon, Saskatchewan S7N 5A9

ABSTRACT

Emergency Management has received more and more attention in the recent years. Most research in this field focused on evacuation of victims from dangerous places to safe places, but little on allocation of medical resources to safe places and/or transportation tools to the dangerous places.

This thesis studies the problem of delivering medical resources from medical centers to the temporary aid site in a disaster-affected area to help the wounded victims. In particular, this thesis describes a new algorithm for solving this problem. As requirements of medical resources for a disaster affected area are not known in advance, the problem is in the so-called on-line environment. The algorithm for such a problem is also called on-line algorithm. The evaluation criterion for such an on-line algorithm is the so-called competitive ratio.

This thesis considers four cases of such a problem: (1) the capacity of vehicles for transporting medical resources and the number of vehicles are both infinite, (2) the capacity of vehicles is infinite but the number of vehicles is one, (3) the capacity of vehicles is finite and the number of vehicles is infinite, (4) the capacity of vehicles is finite and the number of vehicles is one. Algorithms for the four cases are called $H1$, $H2$, $H3$, and $H4$,

respectively.

For all these cases, this thesis presents properties, appropriate on-line algorithms and theoretical analysis of these algorithms. The result of the analysis shows that $H1$ and $H3$ are optimal based on the competitive ratio criterion while the other two have a very small gap in terms of the optimum criterion. The thesis also presents a case study for having a sense of the performance of $H2$ and demonstrating practicality of the developed algorithms.

The result of this thesis has contributions to the field of resource planning and scheduling and has application in not only emergency management but also supply chain management in manufacturing and construction.

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 would like to thank the other members of my advisory committee, Professor F.X. Wu, Professor M. Keil, and Professor M. Gupta, for the assistance they provided during the whole process of my graduate research.

I also would lie 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 wish to gratefully acknowledge the support of all my friends.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ACRONYMS

EM:	Emergency Management (pp. 1)
P:	polynomial-time (pp. 8)
NP:	non-deterministic polynomial-time (pp. 8)
NPC:	NP-complete (pp. 9)
SNPC:	Strongly NP-complete (pp. 9)
ONP:	Ordinary NP-hard (pp. 10)
SNP:	Strongly NP-hard (pp. 10)
TSP:	Traveling Salesman Problem (pp. 11)

NOMENCLATURE

$\mathcal{R}_j$: the jth requirement $(j = 1, 2, \cdots, n)$

r_j: the release time of requirement $\mathcal{R}_j$

p_j: the preparation time for requirement $\mathcal{R}_j$

η: a feasible schedule

opt: an off-line optimal schedule

$C_j(\eta)$: the completion time for requirement $\mathcal{R}_j$ in schedule η

$C_{max}(\eta)$: $\underset{\mathcal{R}_j \in I}{max} C_j(\eta)$

$\rho_j(\eta)$: the departure time for requirement $\mathcal{R}_j$ in schedule η

$\rho_{max}(\eta)$: $\underset{\mathcal{R}_j \in I}{max} \rho_j(\eta)$

T: the round-trip transportation time between the medical center and the aid post

$D_j(\eta)$: the return time of the vehicle delivering requirement $\mathcal{R}_j$ in schedule η, that is, $D_j(\eta) = \rho_j(\eta) + T$

$D_{max}(\eta)$: $\underset{\mathcal{R}_j \in I}{max} D_j(\eta)$

$V(x, y)$: there are x vehicles available, each with a capacity y, where $x \in \{1, \infty\}$, and $y \in \{C, \infty\}$

$pmtn$: the requirement preparation that is preempted and resumed later

D:	the cost of a delivery, which is a constant
$TC(\eta)$:	the total cost of all deliveries in schedule η, which is the number of deliveries multiplied by D
θ:	the positive root of equation $\theta^2 + \theta - 1 = 0$, i.e. $\theta = \frac{\sqrt{5}-1}{2} \approx 0.618$
full batch:	the number of requirements in one delivery batch equals to the capacity of vehicles
unfull batch:	a batch which is not full
$Z(\eta)$:	the total cost of schedule η, that is $Z(\eta) = D_{max}(\eta) + TC(\eta)$
$U[a, b]$:	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 Emergency Management

More and more attention has been paid to Emergency Management (EM) in the recent years. The particular problem in EM studied by this thesis is to deliver medical resources to the temporary aid site after victims are evacuated to such a site. EM has gained a growing interest especially after 911 terrorist attacks in 2001 and hurricane in Florida in 2005. The study in EM is much more necessary in places where natural disasters happen frequently. No matter the terrible earthquakes in Wenchuan of Sichuan and in Yushu of Qinghai or the fire hazard of building in Shanghai, rescue mission needs not only to evacuate the victims from dangerous places to safe ones but also to allocate medical resources from medical centers to the temporary aid sites where wounded victims stay. Hamacher and Tjandra [2002] make a survey of the recent work about EM, almost all of which were about evacuation of victims from one place (source place) to safe places (destination places). Indeed there is little research on allocation of medical resources in EM.

Fig 1.1: Great Disasters in Recent Years

(http://frontpagemag.com/2010/05/31/911-sacrilege/

http://howmanyarethere.net/how-many-people-died-on-the-hurricane-katrina/

http://www.chinadaily.com.cn/china/2008-05/14/content_6682723.htm

http://penguinsabroad.blogspot.ca/2010/04/yushu-some-good-news.html)

1.2 Evacuation Operations

One can reasonably assume that there are many routes to evacuate victims given a transportation system which is highly networked. Further, transportation needs tools (e.g., ground vehicles), and they also take a part of the transportation system. Evacuation problems are studied in different fields such as network flow, traffic assignment and cellular automata. Mathematical models of evacuation problems are divided into two classes: microscopic models and macroscopic models, according to [2002]. Microscopic models simulate behaviors of individual victims for experimental analysis of an evacuation schedule, among which cellular automata simulation models and probabilistic models for pedestrians and traffic movements are widely known [Benjaafar et al. 1997, Klüpfel et al. 2000, Nagel and Schreckenberg 1992]. However, in macroscopic models, victims are considered as a homogenous group with their individual difference being ignored. This thesis presents a study that falls into the category of macroscopic models.

1.3 Allocation of Medical Resources

When wounded victims are evacuated to the temporary aid site, they need medical resources (drugs, medical instruments and medical personnel). Therefore, medical centers should supply these resources to the aid site. In delivering medical resources, medical centers need to prepare medical resources and then deliver the resources to the aid site.

In medical centers, there should be enough time to gather medical personnel, prescribe drugs and assemble medical instruments. The number and the capacity of transportation vehicles are variables that constrain this medical resources delivery.

The existing models for the medical resources delivery only consider known and deterministic situations, which is unrealistic. In reality, situations are unknown, which means that demands on medical resources are unknown. Such situations or environments are called on-line. Accordingly, algorithms for scheduling medical resources in an on-line environment are called on-line algorithms.

There are two attributes in the medical resources delivery: cost and time. Reduction of cost implies fewer vehicles to be put in delivery, which may lead to a longer time. Therefore, the nature of scheduling the medical resources delivery is a multi-objective optimization problem.

1.4 Objectives and Scope of the Thesis

This thesis aims to develop algorithms for scheduling medical resources delivery in EM with a particular context that has one medical center and one aid site. The assumption is further made that demands on medical resources are unknown. The other assumptions are that (1) there is a need of preparation time for medical resources and (2) delivery of medical resources is taken on a transportation network with different routes.

1.5 Organization of the Thesis

Chapter 2 gives a background and literature review on evacuation operations, allocation of medical resources. Some basic concepts will also be discussed in this chapter.

The descriptions and the assumptions of the model will be introduced in Chapter 3. This thesis considers four different problems, proposes algorithms to solve these problems and analyzes these algorithms, which will be described in Chapters 4, 5, 6 and 7, respectively. In particular, Chapter 4 considers the case that the capacity of vehicles and the number of vehicles are both infinite. The algorithm for this case is named $H1$. Chapter 5 considers the case that the capacity of vehicles is infinite but the number of vehicles is one. The algorithm for this case is named $H2$. Chapter 6 considers the case that the capacity of vehicles is finite and the number of vehicles is infinite. The algorithm for this case is named $H3$. Chapter 7 considers the case that the capacity of vehicles is finite, the number of vehicles is one and the preparation of requirements allows pmtn. The algorithm for this case is named $H4$. In Chapter 8, a simulation experiment for $H2$ will be discussed.

Chapter 9 summarizes contributions of this thesis, gives the conclusion, and discusses the future work in this field.

CHAPTER 2
BACKGROUND AND LITERATURE REVIEW

2.1 Introduction

As shown in Chapter 1, little work was focused on allocation of medical resources in EM. This thesis aims to study the problems systematically. The concept of macroscopic models in evacuation operations is applied and the so-called on-line environment or situation is considered. The purpose of this chapter is to provide background regarding and a literature review of macroscopic models, on-line problem and related concepts. In particular, Section 2.2 discusses primary concepts/notions related to medical resources allocation or delivery in EM. Section 2.3 discusses macroscopic models in details. The on-line problem and on-line algorithm are discussed in Section 2.4. In Section 2.5, the problem of supply chain scheduling which has some similar properties as the problem of medical resources delivery studied in this thesis.

2.2 Preliminaries

2.2.1 Analysis of Algorithms

One evaluation method to algorithms is the "run-time" of algorithms. An algorithm can be viewed as a set of operations on "instances". The operation of an algorithm can be further decomposed into a set of basic operations such as addition, subtraction, multiplication, division, evaluation and comparison. Let $n = |I|$ denote the input size an algorithm needs to operate on for an instance I and T denote run-time of the algorithm. In particular, run-time of an algorithm is a function of the size of instances: $T(n)$. It is noted that the size refers to the sum of binary string lengths of an instance's parameters.

Table 2.1: Orders of Common Functions

Notation	**Name**
$O(1)$	constant
$O(loglogn)$	double logarithmic
$O(logn)$	logarithmic
$O(n^c)$, $0 < c < 1$	fractional power
$O(n)$	linear
$O(nlogn) = O(logn!)$	loglinear
$O(n^2)$	quadratic
$O(n^c)$, $c > 1$	polynomial
$O(c^n)$, $c > 1$	exponential
$O(n!)$	factorial

In most cases, it is impossible to get an analytical representation of $T(n)$, so asymptotical bounds for $T(n)$ are usually defined.

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 M and an integer n_0 such that*

$T(n) \leq Mf(n)$ for all $n > n_0$. $T(n) = \Omega(g(n))$ if and only if there exists a positive real number M and an integer n_0 such that $T(n) \geq Mg(n)$ for all $n > n_0$. $T(n) = \Theta(f(n))$ if and only if $T(n) = O(f(n))$ and $T(n) = \Omega(f(n))$.

As a small run-time is desired, the function $f(n)$ such that $T(n) = O(f(n))$ is more meaningful. Table 2.1 shows some common functions $f(n)$.

2.2.2 Computational Complexity

A polynomial function of run-time is noted, as the increase of n gives an accurate sense of the increase of run-time. Therefore, it becomes an evaluation criterion for algorithms in algorithms development. In early years, researchers considered that all problems have their respective polynomial run-time algorithms which means they are polynomial-time (P) problem; but this is not true. Therefore, classification of problems in terms of the polynomial function of run-time is made.

Definition 2.2. *A decision problem is a question with yes-or-no answer depending on the input parameters.*

Definition 2.3. *Decision problem DP is non-deterministic polynomial-time (NP) problem, if for an instance I of DP with yes answer, the proof of the fact that the answer is indeed "yes" can be proved in $p(|I|)$ time.*

To compare the intractability of two decision problems, the polynomial transformation between decision problems is introduced below.

Definition 2.4[Karp 1972]. *Decision problem* DP_1 *is polynomially transformed to decision problem* DP_2*, if for an arbitrary instance I_1 of* DP_1*, an instance I_2 of* DP_2 *can be* *constructed in* $p(|I_1|)$ *time such that the answer of* I_1 *is yes if and only if the answer of* I_2 *is yes.*

It is noted that DP_1 being polynomial transformed to DP_2 means that if DP_2 has a polynomial algorithm, so does DP_1. This also means DP_2 is harder than DP_1.

In NP problems, there is a class of problems which are harder than others.

Definition 2.5[Garey and Johnson 1979]. *Decision problem DP is NP-complete if:*

1. DP is NP problem,

2. Every NP problem can be polynomially transformed to DP.

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

Definition 2.6. *Decision problem 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* $p(|I|)$.

For problems which do not belong to NP, the measurement of intractability would refer to the definition of NP-complete, but polynomial transformation cannot be applied because these problems may not be a decision problem. Therefore, polynomial reduction between problems is defined below.

Definition 2.7[Rogers 1967]. *Suppose that* AP_1 *and* AP_2 *are two problems, if* AP_1*'s algorithm* $\mathcal{A}_1$ *calls* AP_2*'s algorithm* $\mathcal{A}_2$ *a polynomial number of times, then* AP_1 *is polynomially reducible to* AP_2.

It can also be shown that AP_2 is harder than AP_1 if AP_1 is polynomially reducible to AP_2.

For an algorithmic problem AP which may not be in NP, if it is polynomially reducible to a NP-complete problem, then it is NP-hard problem. Accordingly, strongly NP-hard problem (SNP) can be defined. A NP-hard problem which is not SNP is called ordinary NP-hard (ONP).

The decision problems which have polynomial algorithms are in P. It is obvious that $P \subseteq NP$, but whether $P = NP$ is still unknown. Actually, there is a famous conjecture $P \neq NP$, which means there are decision problems which do not have polynomial algorithms [Gasarch 2002, Rosenberger 2012]. Fig 2.1 shows a venn diagram to illustrate the relationship among P, NP, NPC, SNPC, ONP and SNP.

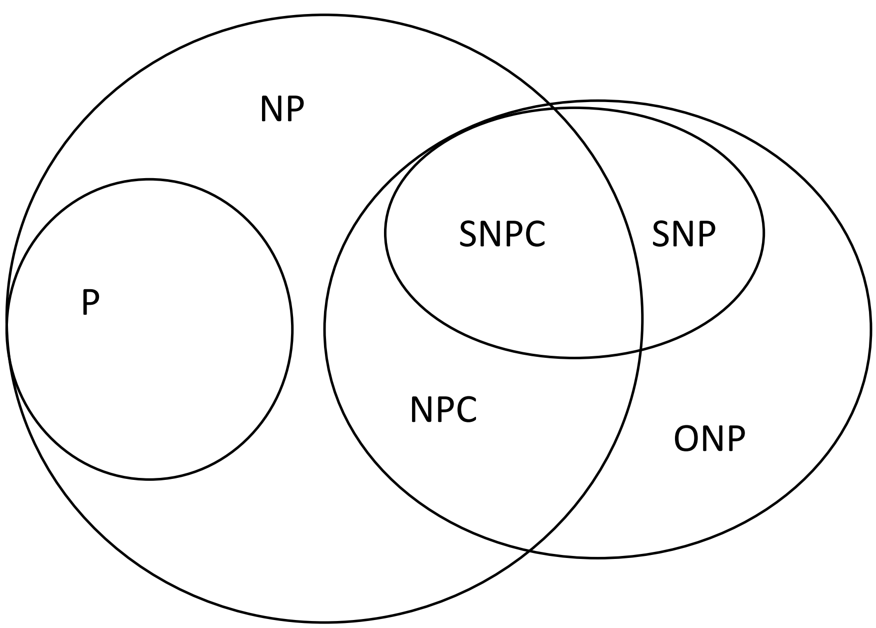

Fig 2.1: Venn Diagram of complexity concepts

2.2.3 Approximation Algorithm

Because (1) NP-hard problems do not have polynomial algorithms unless P=NP and (2) almost all practical problems are NP-hard; enumeration algorithms, branch and bound algorithms and intelligent algorithms are designed for these problems. In the case where time is more important than accuracy, the above methods may not be applicable; therefore efficient approximation algorithms are required.

The precise definition of approximation algorithms for the minimization problem is as follows.

Definition 2.8. *Let $\mathcal{P}$ be a minimization problem and $\mathcal{I}$ be an instance of $\mathcal{P}$, $\mathcal{A}(\mathcal{I})$ be the objective function by algorithm $\mathcal{A}$ for $\mathcal{I}$ and $OPT(\mathcal{I})$ be the optimal solution. If $\frac{\mathcal{A}(\mathcal{I})}{OPT(\mathcal{I})} \leq r$ for all $\mathcal{I}$ and $r \geq 1$, then $\mathcal{A}$ is called a r-approximation algorithm. Furthermore, if $\mathcal{R}_{\mathcal{A}} = inf\{r \geq 1, \frac{\mathcal{A}(\mathcal{I})}{OPT(\mathcal{I})} \leq r$, for all $\mathcal{I}\}$, algorithm $\mathcal{A}$ has approximation ratio $\mathcal{R}_{\mathcal{A}}$.*

A similar definition can be made for the maximization problem.

2.2.4 Traveling Salesman Problem

Traveling Salesman Problem (TSP) is to find the shortest route for a traveling salesman to visits all the nodes of a given transportation network. The polynomial time approximation algorithms of TSP were studied, among which the well-known result comes from from [Christofides 1976]. An instance of the Traveling Salesman Problem is given by a weighted graph and an initial vertex. The goal is to find a tour, i.e., a Hamiltonian circuit, that has a minimum length. The decision problem version of TSP is NP-hard because the Hamiltonian Circuit problem can be polynomial transformed to it. It has been proved that the general TSP is not approximate 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 3/2. The situation is even more favorable in the Euclidean plane, for which Arora (1997) gives a $(1 + \varepsilon)$ approximation scheme.

2.3 Macroscopic Models of Evacuation Problems

In macroscopic models, victims are considered as a homogenous group and only common characteristics of all individuals are taken into account.

Sometimes, evacuation problems are considered as static network flow problems. The static network G is used to represent sources and destinations, and routes which are used to evacuate people from sources to destinations. The routes may have some intermediate points. In the static network flow models, sources, destinations and intermediate points are considered as nodes while routes are considered as paths in graph. Using graph theory, the whole model can be described by a node-arc incidence matrix. On source nodes are victims being evacuated, while intermediate nodes and destination nodes are given upper limits in victims who can stay. Arcs have attributes such as flow capacity and travel time. Such evacuation problems include:

1. Shortest path [Fahy 1991],
2. Minimum cost network flow [Yamada 1996],
3. Quickest path [Chen and Chin 1990, Chen and Hung 1993].

However, the static network flow models are not sufficient to represent the evacuation problems in practice because of the absence of time. Dynamic network G_T is then introduced to model evacuation operations over time. G_T can be seen as the time expanded version of the static network G and the network flows of G. There are two classes of dynamic network models: discrete-time dynamic network and continuous-time dynamic network.

Several discrete-time network models are introduced as follows:

1. Minimum turnstile cost [Chalmet et al. 1982, Chen and Hsueh 1998],
2. Quickest flow [Burkard et al. 1993, Fleischer 1998],
3. Universally maximum flow [Hoppe and Tardos 1994, Minieka 1973],
4. Minimum weight path (multi objectives) [Kostreva and Wiecek 1993],
5. Lexicographically minimal cost [Hamacher and Tufekci 1987],
6. Flow dependent exit capacity [Choi et al. 1984, Choi et al. 1988].

In the field of dynamic network flow problems, researchers considered the models with constant attributes such as constant travel time and constant flows capacity. These special models can be solved efficiently. The following continuous-time dynamic flow models are examples:

1. Maximum flow with time dependent capacity [Anderson et al. 1982, Philpott 1990],
2. Universally maximum flow with zero travel time [Fleischer 2001b, Ogier 1988],
3. Quickest flow with constant capacity and travel time [Fleischer 2001a].

2.4 On-line Problems and On-line Algorithms

The traditional algorithms assume that an algorithm has the complete knowledge of an entire input. However, in practice this assumption is sometimes unrealistic. In many practical problems, the input is only partially known or totally unknown the time a schedule is made. Problems which have this situation are called on-line problems. Accordingly, if for a problem the complete information of the input is available at the beginning, the problem is called the off-line problem. On-line problems arise in many areas [Albers 2003]. The formal definition of on-line problems can be described as follows.

Definition 2.9. *There is a request sequence* $\tau = \tau(t_1), \tau(t_2), \cdots, \tau(t_n)$*, which must be served by a server. At time* t*, no knowledge of any request* $\tau(t')$ *with* $t' > t$ *is known. There will be a cost to serve these requests and the goal of having a service schedule is to minimize the total cost for the entire request sequence.*

Algorithms for on-line problems are called on-line algorithms. According to the above definition of on-line problems, on-line algorithms must decide how to serve the requests $\tau(t')$ $(t' \leq t)$ without any information of the requests $\tau(t'')$ $(t'' > t)$ at time t.

Suppose there are algorithms which have the complete information of an entire input of this on-line problem. Such algorithms are called off-line algorithms. The result of an off-line algorithm is an optimal solution. The difference of an on-line algorithm from the optimal off-line algorithm can be used to evaluate an on-line algorithm.

Definition 2.10. *Let* $\mathcal{P}$ *be a (minimization) on-line problem and* $\mathcal{I}$ *be an instance of* $\mathcal{P}$*,* $\mathcal{A}(\mathcal{I})$ *be the objective function in algorithm* $\mathcal{A}$ *for* $\mathcal{I}$ *and* $OPT(\mathcal{I})$ *be the off-line optimal* *result. If $\frac{\mathcal{A}(\mathcal{I})}{OPT(\mathcal{I})} \leq r$ for all $\mathcal{I}$ and a $r \geq 1$, $\mathcal{A}$ is called a r-competitive algorithm. Furthermore, if $\mathcal{R}_{\mathcal{A}} = inf\{r \geq 1, \frac{\mathcal{A}(\mathcal{I})}{OPT(\mathcal{I})} \leq r$, for all $\mathcal{I}\}$, the on-line algorithm $\mathcal{A}$ has the competitive ratio $\mathcal{R}_{\mathcal{A}}$.*

For an on-line problem, how intractable it can be is first analyzed, and then algorithms for it are designed. Measurement of the intractability of an on-line problem is the competitive ratio of an on-line algorithm for the on-line problem.

Definition 2.11. *For an on-line problem, if no on-line algorithm can achieve a competitive ratio less than L, we say that L is the lower bound of this on-line problem.*

The method to obtain a lower bound is to construct a series of instances of an on-line problem and prove that no algorithm can achieve a competitive ratio less than L for these instances. Construction of the instances should be in accordance with the structure of an on-line algorithm and make L as large as possible.

For each on-line problem the optimal on-line algorithm is the one whose competitive ratio which can be viewed as the upper bound of this on-line problem is equal to its lower bound. It is clear that on-line algorithms are developed such that their competitive ratios are as close to their lower bounds as possible.

2.5 Supply Chain Scheduling Problem

On-line problems for allocation of medical resources in EM are similar to on-line problems in supply chain scheduling. It is reasonable to give comments on the literature of supply chain scheduling.

In a large scale manufacture (Fig. 2.2), jobs are processed on machines in an shop-flow and delivered to customers who order them; thus production cost and delivery cost arise. To customers who want products, both production and delivery activities are in fact relevant. That is to say, both the cost and delivery time of a product to customers would be charged due to a schedule of production and/or a schedule of delivery among many other factors (e.g., quality of production machinery). An analogy between supply chain management and allocation of resources can be made as:

$$\begin{array}{ccc} \text{supply chain} & \leftrightarrow & \text{allocation of resources} \\ \text{product production} & & \text{material preparation} \\ \text{delivery} & & \text{delivery} \end{array}$$

Hall and Potts [2005] give the definition of supply chain scheduling problem and solve some basic off-line versions. The difference of supply chain scheduling from classical scheduling is that because of transportation cost, the processed jobs would not be delivered to customers immediately, but have to wait for other jobs to form batches. Therefore, the decision about a job is not only when to process it and which machine to process it, but also when to deliver it and which batch to deliver it.

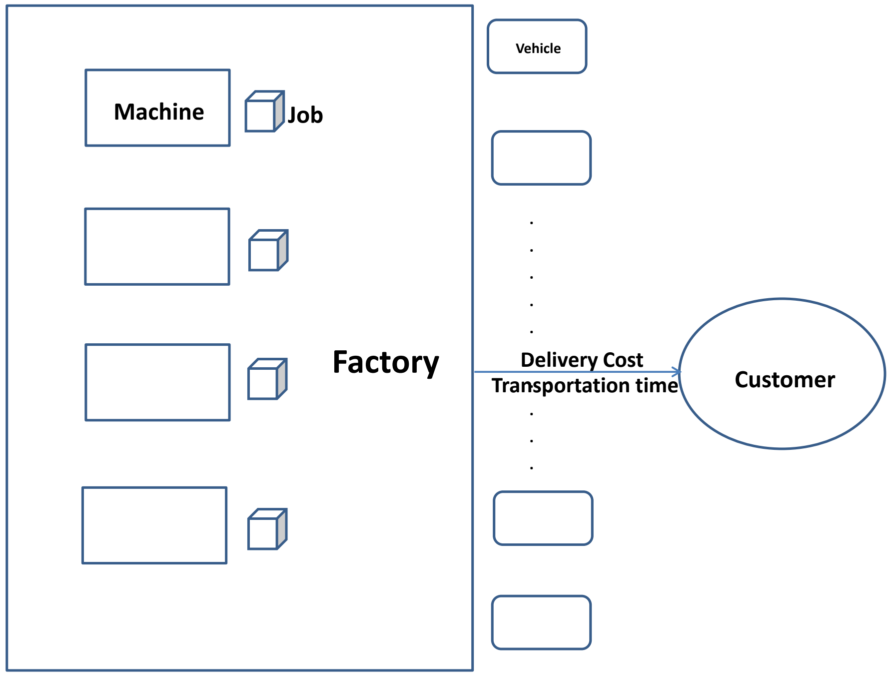

Fig 2.2: Supply Chain Scheduling

When the number of customers is more than one, the supply chain scheduling problem will involve routing which makes the problem more difficult. Chen and Vairaktarakis [2005] studied eight basic supply chain scheduling problems, four of which have the routing problem as their sub-problem; however, they considered the case that the number of customers is constant and solved the routing problem by enumeration. If the number of customers is a variable, the routing sub-problem will be strongly NP-hard as it is equivalent to the traveling salesman problem (TSP).

Chen [2010] made a survey on the supply chain scheduling problem and modified the three-field notation of the classical scheduling problem to be five-field notation $\alpha|\beta|\pi|\delta|\gamma$ to represent the supply chain scheduling problem. The explanation of the five fields is as follows

α: Machine Configuration.

1:	single machine configuration, where all the jobs are precessed by a single machine.
Pm:	parallel-machine configuration, where there are m identical parallel machines such that each job needs to be processed by one of them only.

β: Restrictions and Constraints on Job Parameters.

r_j:	jobs have unequal release times.
$on - line$:	jobs are released on-line.
$pmtn$:	job processing can be preempted and resumed later.
$prec$:	jobs have precedence constraints between them.

π: Delivery Characteristics including vehicle characteristics and delivery methods.

Vehicle characteristics:

$V(x, y)$:	there are x vehicles available, each with a capacity y; so $x \in \{1, \infty\}$ and $y \in \{C, \infty\}$, where the symbol "∞" means "enough" in the engineering sense (this interpretation of "∞" is valid throughout this thesis).
$x = 1$:	a single vehicle available.
$x = v$:	v vehicles are available, where $v < n$ and is finite.
$x = \infty$:	enough vehicles are available such that the number of vehicles is not constraint.
$y = 1$:	each vehicle can accommodate only one job.
$y = C$:	each vehicle can accommodate C jobs, where $C < n$ and is finite.
$y = \infty$:	each vehicle can accommodate any number of jobs.

Delivery methods:

iid: each job is delivered individually and immediately after its completion.

direct: only jobs of the same customer can be delivered together in the same shipment.

routing: jobs of different customers can be delivered together in the same shipment where vehicle routing is a part of the decision.

δ: Number of Customers

1: a single customer.

k: multiple customers, where $2 \leq k \leq n$.

n: n customers, meaning that each job belongs to a different customer.

γ: Objective Function

Time-based:

D_{max}: maximum delivery time of jobs.

$\sum(w_j)D_j$: total (weighted) delivery time of jobs.

L_{max}: maximum delivery lateness of jobs.

$\sum(w_j)T_j$: total (weighted) delivery tardiness of jobs

$\sum(w_j)U_j$: (weighted) number of late jobs.

Cost-based:

TC: total trip-based transportation cost.

VC: total vehicle-based transportation cost.

PC: total production cost of the orders.

The on-line problems consider the supply chain scheduling problem in the on-line environment, where the information of the future jobs is unknown. On-line problems are more realistic today in the manufacturing world as the demand of market is highly dynamic and uncertain. More specifically, suppose that there are n jobs $\mathcal{J}_1, \cdots, \mathcal{J}_n$ with processing times $p_1, \cdots, p_n$ released online at times $r_1, \cdots, r_n$, respectively, to a manufacturer which has machines. The jobs are released by phone or email. As well, suppose at any time there is no information about the number, release time and processing time of future jobs.

When the jobs are completed, they should be transported to the customers with vehicles and there would be a delivery cost thus incurred. In light of the cost, delivery time and "mass customization", the delivery of processed jobs with respect to a particular customers may not be immediately carried. The objective of the scheduling is to minimize both the time and the delivery cost.

Averbakh and Xue [2007] and Averbakh [2010] studied the on-line supply chain scheduling with preemption and obtained the on-line optimal result for the case of single-machine and single-customer. When there is more than one customer, the modified algorithm was proposed but because it scheduled the jobs without differences in the processing part, the result was not good enough according to [Averbakh and Xue 2007, Averbakh 2010].

Table 2.2 summarizes the algorithms proposed in literature for supply chain scheduling problems. From the table, it can be seen that there are systematic works for the off-line problems [Hall and Potts 2005, Chen and Vairaktarakis 2005] but not for on-line problems. This thesis will deal with on-line problems under configuration of machines and customers as the supply chain scheduling problem but with some difference in vehicle characteristics.

Table 2.2: Results of Several Supply Chain Scheduling Problems

Problem	Complexity	Algorithm
$1\|r_j, prec\|V(\infty,1), iid\|n\|D_{max}$	SNP	$\frac{4}{3}$-approximation algorithm [Hall and Shmoys 1992]
$1\|r_j\|V(\infty,1), iid\|n\|D_{max}$	SNP	PTAS [Hall and Shmoys 1992] [Mastrolilli 2003]
$1\|\|V(v,1), iid\|n\|D_{max}$	SNP even if $v=2$	
$Pm\|\|V(\infty,1), iid\|n\|D_{max}$	ONP	$2-\frac{2}{m+1}$-approximation algorithm [Woeginger 1994]
$Pm\|r_j\|V(\infty,1), iid\|n\|D_{max}$	SNP	branch and bound method [Gharbi and Haouari 2002]
$Fm\|\|V(\infty,1), iid\|n\|D_{max}$	SNP	asymptotically optimal heuristic algorithm [Kaminsky 2003]
$1\|\|V(\infty,c), direct\|1\|D_{max}+TC$	P	$O(n)$ [Chen and Vairaktarakis 2005]
$1\|\|V(\infty,\infty), direct\|1\|L_{max}+TC$	P	$O(n^3)$ [Hall and Potts 2005]

continued on the next page

Table 2.2: Results of Several Supply Chain Scheduling Problems (*continued*)

Problem	Complexity	Algorithm
$1\|r_j, pmtn\|V(\infty,\infty), direct\|1\|\sum D_j$	Open	On-line algorithm citeAverbakh and Xue 2007
$1\|\|V(\infty,c), direct\|1\|\sum D_j + TC$	P	$O(nlogn + nc)$ [Chen and Vairaktarakis 2005]
$1\|\|V(\infty,\infty), direct\|1\|\sum D_j + TC$	P	$O(n^2)$ [Hall and Potts 2005]
$1\|\|V(v,\infty), direct\|1\|L_{max} + TC$	P for fixed v Open for arbitrary v	$O(vn^{3v+4})$
$1\|\|V(\infty,\infty), direct\|1\|\sum w_jD_j + TC$	SNP	Polynomial algorithm for special case [Ji et al. 2007]
$1\|\|V(\infty,\infty), direct\|1\|\sum U_j + TC$	P	$O(n^4)$ [Hall and Potts 2005]
$1\|\|V(\infty,c), direct\|1\|\sum U_j + TC$	Open	
$1\|\|V(\infty,\infty), direct\|1\|\sum w_jU_j + TC$	ONP	
$1\|\|V(\infty,\infty), direct\|1\|\sum E_j + TC$	P	$O(n^2logn)$ [Cheng and Kahlbacher 1993]
$1\|\|V(\infty,\infty), direct\|1\|\sum T_j + TC$	ONP	

continued on the next page

Table 2.2: Results of Several Supply Chain Scheduling Problems (*continued*)

Problem	Complexity	Algorithm
$1\|\|V(v,c),direct\|1\|D_{max}$	P	$O(nlogn)$ [Chen and Lee 2008]
$Pm\|\|V(\infty,c),direct\|1\|D_{max}+TC$	ONP	$2-\frac{1}{m}$-approximation algorithm [Chen and Vairaktarakis 2005] asymptotically optimal
$1\|\|V(1,\infty),direct\|1\|L_{max}+TC$	P	$O(n^5)$ [Hall and Potts 2005]
$1\|\|V(1,\infty),direct\|1\|\sum D_j+TC$	P	$O(n^4)$ [Hall and Potts 2005]
$1\|\|V(1,c),direct\|1\|\sum D_j$	P	$O(n^2)$ [Chen and Lee 2008]

Source: [Chen 2010]

CHAPTER 3
PROBLEMS' ASSUMPTIONS AND NOTATIONS

Suppose that a disaster takes place and a rescue team evacuates victims in affected areas and settles them to a temporary aid site. Victims may get hurt during the disaster, and they may need a great deal of medical resources to cure them. Assume that medical centers deliver enough medical resources such as drugs, medical instruments and medical personnel to the aid site.

The medical resources are continuously required in the whole process of EM; thus the information of the requirements is not known at the time scheduling decisions have to be made. Therefore, the on-line mechanism for scheduling decisions should be considered, which means that there are different resources requirements at different moments and they are known when they are released.

After medical centers receive demands or requirements of medical resources, the centers may need to spend time to prepare the resources. Therefore, when a requirement of resources is released, the decision maker should decide when to prepare the requirement and the work force to prepare it. The constraint on the preparation is that the requirement should be known. When a requirement is prepared, 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 aid site. The constraint on the delivery is that the deliver of the requirement should be (1) after the requirement has been prepared and (2) there are available vehicles at that moment.

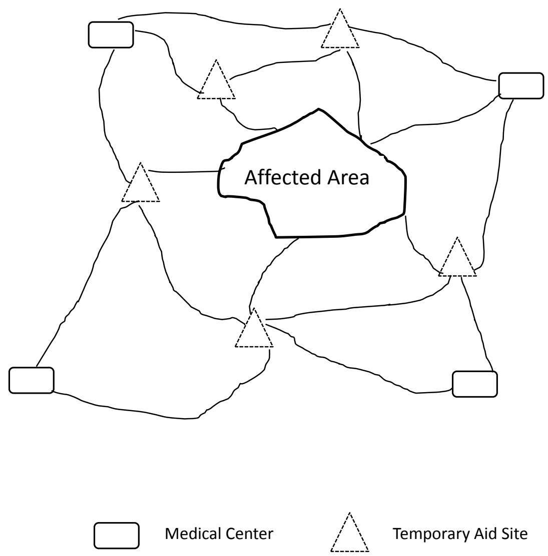

Fig 3.1: Allocation of Medical Resources in Emergency Management

The time that a requirement is known is called "release time" of the requirement. The time that a requirement is prepared is called "completion time" of the requirement The time that a requirement leave the medical center is called "departure time" of the requirement. The time that a shipment returns to the center is called "return time" of all requirements in the shipment. Thus, the time-based objective can be defined as a function of release time, completion time, departure time and return time. Obviously this objective should be minimized to make the whole process efficient.

When a vehicle delivers an requirement to its destination, there is a transportation cost for this delivery. Therefore, the cost-based objective would is proportional to the number of deliveries, which further represents the utility ratio of vehicles and should also be minimized. It is noted that the objective to minimize the time conflicts with the objective to minimize the cost, because delivering every requirement as soon as possible means using vehicles as many as available. The final objective of the problem may have different representations: to minimize a weight sum of the two objectives, Pareto solutions, to minimize one objective under the minimization of the other. The general principle to trade-off the two objectives is: the prepared requirements should be delivered quickly whilst the utilization ratio of vehicles is high.

3.1 Assumptions

In reality, the preparation procedure of medical resources in medical centers is very complicated due to many uncertain factors. The decision maker has to prescribe drugs and assemble medical instruments; meanwhile, relevant personnel should be informed and gathered to the medical centers. These steps are not sequential and may interfere with each other. Uncertain factors include uncertain availability of human workers and so on. If there is more than one medical center, assigning the requirements to different medical centers will be a part of the decision, which is out of the scope of this thesis. After the preparation, the resources should be delivered by the vehicles to aid sites in affected areas, which involves the characteristic of vehicles and transportation networks. The capacity of vehicles and the number of vehicles will determine how to divide the resources into different shipments, while the decision of delivery would be more complex if the delivery has to share the transportation network with the evacuation operations. In addition, parameters of the network may also be stochastic, as the transportation system may be destroyed to a certain extent in the disaster.

In this thesis, the following assumptions are made:

1. All requirements of medical resources are homogenous, which means that the preparation of requirements only considers the preparation time but ignores other factors and every requirement only occupies the unit size of a vehicle in the delivery.

2. All requirements are independent, which means that there are no preceding constraints on requirements, and preparation and delivery of every requirement will be independent of others.

3. There is only one medical center.

4. One requirement is prepared by one processor in the center at a time and one processor prepares one requirement at a time.

5. Delivery of medical resources does not share the transportation network with evacuation operations.

6. All parameters involved in the problem are deterministic.

3.2 Problem Formulation and Notations

Suppose that there are n requirements $\mathcal{R}_1, \cdots, \mathcal{R}_n$ with preparation times $p_1, \cdots, p_n$, released online at times $r_1, \cdots, r_n$; respectively, from the aid sites to medical centers. The requirements are released by phone or email, so medical centers can prepare requirements once the requirements are released. No information regarding the number, release time and preparation time of future requirements is given. After requirements are prepared, they are transported to the aid sites with vehicles and there would be a delivery cost. A schedule should specify when a requirement is prepared and when a requirement is set off. Requirements are are also divided into several batches thus a schedule should decide which requirements are a batch. Further, delivery may not be carried out immediately, though all released requirements are prepared, which means that there may be a period of waiting time for delivery. The problem of scheduling is to minimize a weighted sum of the time-based objective function and the cost-based objective function (as mentioned before). For the time-based objective function, the total time of the procedure is considered, that is upto the return time of the last vehicle. The cost-based objective function is the number of all batches multiplied by the cost of one delivery. It can be shown that the weighted sum of the two objective functions is equivalent to the simple additive sum of the both; see below for derivation,

$$\begin{aligned} w_1 D_{max} + w_2 TC &= w_1 D_{max} + w_2 D \times y \\ &= w_1 (D_{max} + \frac{w_2}{w_1} D \times y) \end{aligned} \tag{3.1}$$

where w_1, w_2 are weights and y is the number of batches. Thus, minimizing $w_1 D_{max} + w_2 TC$ is equivalent to minimizing $D_{max} + \frac{w_2}{w_1} D \times y$. As the unit delivery cost D is a constant, let $D' = \frac{w_2}{w_1} D$ and the objective function becomes $D_{max} + TC'$.

In this thesis four problems with single-processor and single-aid-site but different characteristics of vehicles are studied. The five-field notation [Chen 2010] are employed to describe problems as follows:

(P1) $1|r_j, on-line|V(\infty,\infty), direct|1|D_{max} + TC$

The capacity of vehicles and the number of vehicles are both enough.

(P2) $1|r_j, on-line|V(\infty, C), direct|1|D_{max}+TC$

The capacity of vehicles is finite and the number of vehicles is enough.

(P3) $1|r_j, on-line|V(1, \infty), direct|1|D_{max}+TC$

The capacity of vehicles is enough but the number of vehicles is one.

(P4) $1|r_j, pmtn, on-line|V(1, C), direct|1|D_{max}+TC$

The capacity of vehicless is finite and the number of vehicles is one while the preemption of requirements is allowed.

CHAPTER 4
SOLVING PROBLEM P1

In this chapter, Problem P1 is studied. Definition of Problem P1 is referred to Section 3.2. According to the assumptions made for P1, during the delivery, lacking of vehicles will not happen. As such, the transportation time $T = 0$ is assumed (a nonzero transportation time will certainly increase the cost of a schedule but by the same amount, so it does not affect an optimization of the schedule), leading to $D_j(\eta) = \rho_j(\eta)$ for $\mathcal{R}_j \in I$.

The optimal schedule of the off-line version of P1 is to prepare requirements without idle time and to deliver all the requirements in one batch at the time when the last requirement is completed. So the optimal solution is $Z(opt) = C_{max}(opt) + D$. However, the on-line version, i.e. P1, is much more difficult and, in fact, there is a lower bound for the competitive ratio for all on-line algorithms.

4.1 The Lower Bound for P1

Theorem 4.1. *No on-line algorithm for P1 can have competitive ratio less than 2, even though all processing times are 0.*

Proof Let us consider the performance of an arbitrary on-line algorithm H for the following instance. The instance releases a requirement with zero preparation time at time $r_1 = 0$, and if the algorithm H delivers the requirement at time $\rho_1 \geq D$, then there are no requirements coming. Otherwise, the second requirement with zero preparation time arrives at time $r_2 = D$, and if the departure time of this requirement $\rho_2 \geq 2D$, then there are no requirements coming, otherwise, the third requirement comes at time $r_3 = 2D$, and so on. If the algorithm H delivers the ith requirement with zero preparation time at time $\rho_i \geq iD$, then there are no requirements coming, or the $(i+1)$th requirement with zero preparation time comes at time $r_{i+1} = iD$. The process is repeated until at most N requirements have been released and delivered (see Fig 4.1).

If the instance at last has released and delivered k requirements, where $k < N$, then the k requirements are delivered in k different batches and $D_k = \rho_k \geq kD$. So the solution value of the schedule η obtained by the algorithm H is $Z(\eta) = D_k + kD \geq 2kD$, while the optimal schedule delivers all the requirements in a batch at time $(k-1)D$ and the value is $Z(opt) = (k-1)D + D = kD$. Since $\frac{Z(\eta)}{Z(opt)} \geq 2$, the competitive ratio of the algorithm is no less than 2.

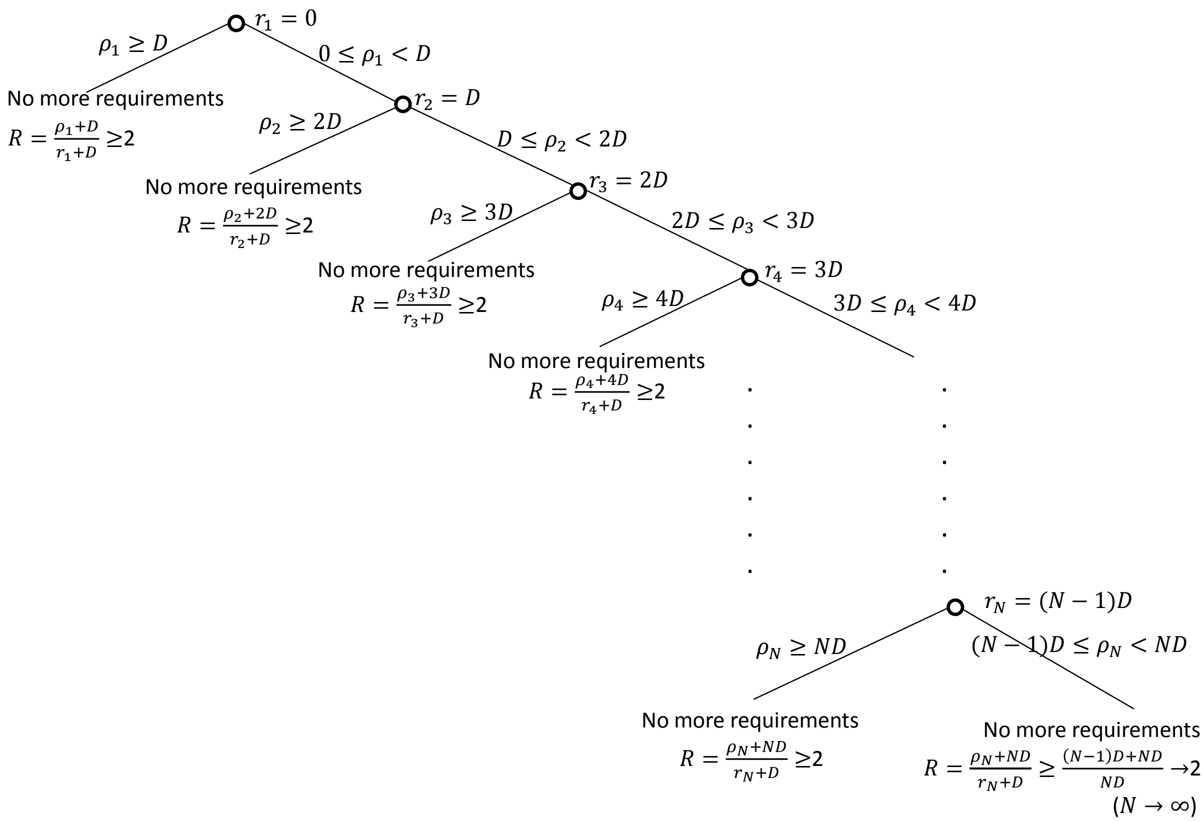

Fig 4.1: The Lower Bound for P1

If the instance at last has released and delivered N requirements, the N requirements are delivered in N batches and $D_N \geq r_N = (N-1)D$. So the solution value of the schedule η obtained by the algorithm H is $Z(\eta) = D_N + ND \geq (2N-1)D$, while the optimal schedule delivered all the requirements in a batch at time $(N-1)D$ and the solution value is $Z(opt) = (N-1)D + D = ND$. As N gets infinitely large, $\frac{Z(\eta)}{Z(opt)} = \frac{(2N-1)D}{ND}$ will tend to 2 which means the competitive ratio of the algorithm is no less than 2. $\square$

4.2 The Upper Bound for P1

In the last section, the lower bound for Problem P1 was given. In this section we will design an on-line algorithm for P1 and give competitive ratio analysis.

Algorithm H1 Requirements are scheduled on the processor without idle time. At the time of lD where $l \geq 1$, if there is no uncompleted requirement , then there must be a batch to deliver all completed requirements; otherwise, there is no operation.

Theorem 4.2. *The competitive ratio of the on-line algorithm $H1$ for P1 is 2.*

Proof Let η be the schedule obtained by the algorithm $H1$ and k be the number of batches delivered in the schedule η. As the deliveries only happen at time of lD where $l \geq 1$, we can get $k \leq \frac{\rho_{max}(\eta)}{D}$. Meanwhile, there should be requirements completed after $\rho_{max}(\eta) - D$ in the schedule η; otherwise there would not be the kth batch.

$$\begin{aligned} Z(\eta) &= D_{max}(\eta) + TC(\eta) \\ &= \rho_{max}(\eta) + kD \\ &\leq \rho_{max}(\eta) + \frac{\rho_{max}(\eta)}{D} D \\ &= 2\rho_{max}(\eta) \end{aligned} \tag{4.1}$$

However,

$$\begin{aligned} Z(opt) &= C_{max}(opt) + D \\ &= C_{max}(\eta) + D \\ &\geq \rho_{max}(\eta) - D + D \\ &= \rho_{max}(\eta) \end{aligned} \tag{4.2}$$

At last, we get $\frac{Z(\eta)}{Z(opt)} \leq 2$.

According to Theorem 4.1, the competitive ratio of $H1$ can not be less than 2 which completes the proof. $\square$

4.3 Discussion and Concluding Remarks

This chapter discussed Problem P1 where both the capacity of vehicles and the number of vehicles are infinite. At first, we constructed a series of instances and showed that no on-line algorithm can obtain a competitive ratio less than 2 for all these instances, which means that the lower bound of P1 is 2. Next, we designed an on-line algorithm $H1$ for P1 and proved that the competitive ratio of $H1$ is 2. Because the competitive ratio of the on-line algorithm $H1$ for P1 is equal to the lower bound of P1, we can conclude that the algorithm $H1$ is on-line optimal. This on-line optimality implies that algorithm $H1$ is robust for all instances of P1 and achieves the best possible solution for the worst case.

CHAPTER 5
SOLVING PROBLEM P2

In this chapter, we study Problem P2, in which the capacity of vehicles is constrained; in particular the capacity is a constant C (when $C = 1$ the problem is trival, so we always assume $C \geq 2$). Further, according to the assumption in P2, the number of vehicles is enough, lacking of vehicles will not happen. For the same reason for Problem P1, we can assume $T = 0$. This further leads to $D_j(\eta) = \rho_j(\eta)$ for $\mathcal{R}_j \in I$.

5.1 The Lower Bound for P2

Theorem 5.1. *No on-line algorithm for P2 can have competitive ratio less than $2 - \frac{1}{C}$, even though all preparing times are 0.*

Proof Let us consider the performance of an arbitrary on-line algorithm H for the following instance. The instance releases a requirement with zero preparation time at time $r_1 = 0$, and if the algorithm H delivers the requirement at time $\rho_1 \geq D$, then there is no requirement coming. Otherwise, the second requirement with zero preparation time arrives at time $r_2 = D$, and if the departure time of this requirement $\rho_2 \geq 2D$, then there is no requirement coming, or else the third requirement comes at time $r_3 = 2D$, and so on (see Fig 5.1). If the algorithm H delivers the ith requirement with zero preparation time at time $\rho_i \geq iD$, then there is no requirement coming, or the $(i + 1)$th requirement with zero preparation time comes at time $r_{i+1} = iD$. The process is repeated until at most C requirements have been released and delivered.

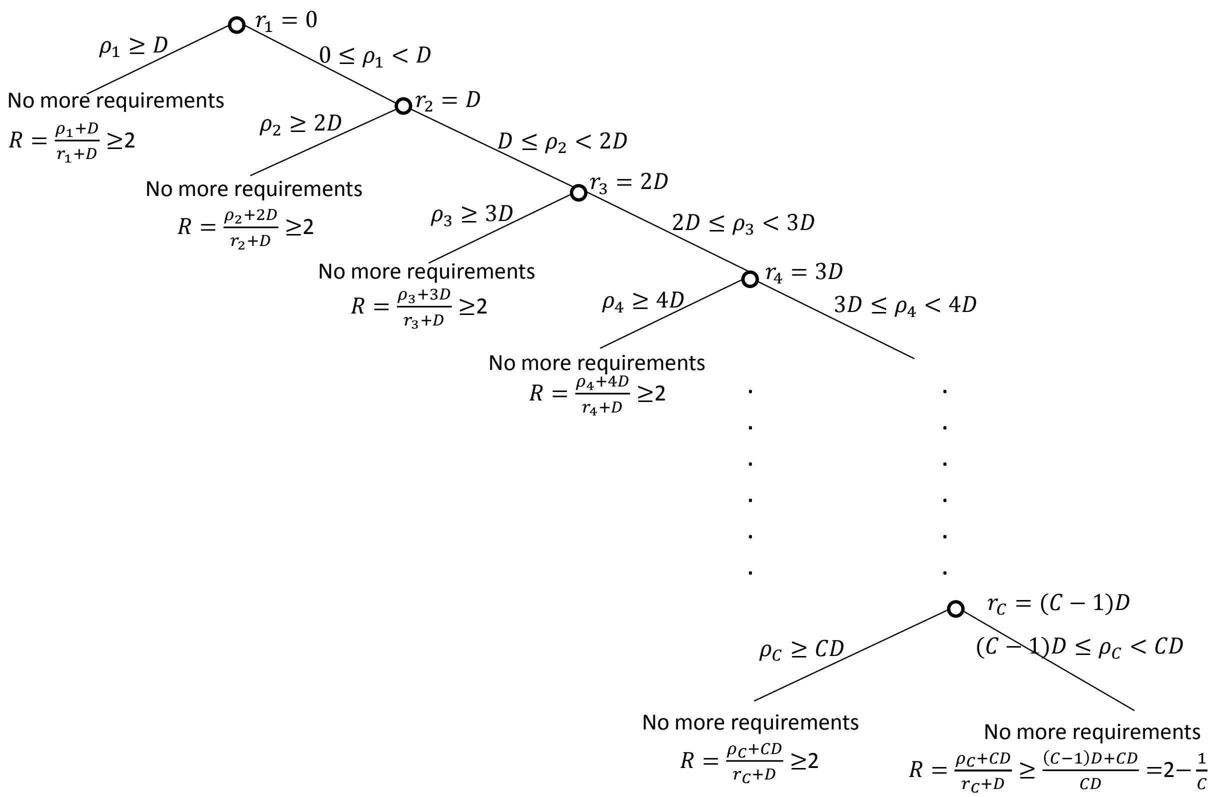

Fig 5.1: The Lower Bound for P2

If the instance at last has released and delivered k requirements, where $k < C$, then we get that the k requirements are delivered in k different batches and $D_k = \rho_k \geq kD$. So the solution value of the schedule η obtained by the algorithm H is $Z(\eta) = D_k + kD \geq 2kD$, while the optimal schedule has delivered all the requirements in a batch at time $(k-1)D$ and the optimal value is $Z(opt) = (k-1)D + D = kD$. The ratio of the algorithm value and the optimal value is greater than 2.

If the instance at last has released and delivered C requirements, we can get that the C requirements are delivered in C batches and $D_C \geq r_C = (C-1)D$. So the solution value of the schedule η obtained by the algorithm H is $Z(\eta) = D_C + CD \geq (2C-1)D$, while the optimal schedule has delivered all the requirements in a batch at time $(C-1)D$ and the optimal value is $Z(opt) = (C-1)D + D = CD$. Thus, the competitive ratio of H cannot be less than $2 - \frac{1}{C}$. $\square$

5.2 The Upper Bound for P2

In this section, we will give an upper bound for P2.

Algorithm H2 Requirements are scheduled on the processor without idle time. At the time of lD where $l \geq 1$, if there is no uncompleted requirement, then there are batches to deliver all completed requirements; otherwise, there are some full bathes to deliver as many completed requirements as possible.

Theorem 5.2. *The competitive ratio of the on-line algorithm $H2$ for P2 is 2.*

Proof Let η be the schedule obtained by the algorithm $H2$, and there are k_f full batches and k_u unfull batches in the schedule η. There is at most one unfull batch at every delivery time, so $k_u \leq \frac{\rho_{max}(\eta)}{D}$. Every unfull batch at least contains one requirement, so $k_f C + k_u \leq n$.

Obviously, the optimal schedule of the off-line version of P2 is to prepare the requirements without idle time and to deliver all requirements at the time when the last requirement is prepared. So the optimal value is $Z(opt) = C_{max}(opt) + \lceil \frac{n}{C} \rceil D$. According to the different value rages of $\lceil \frac{n}{C} \rceil$, we analyze the ratio for the following two cases, respectively.

Case 1: $\lceil \frac{n}{C} \rceil = 1$

If there is only one full batch in the schedule η, then $TC(\eta) = D \leq \rho_{max}(\eta)$, or else the batches in η are all unfull and $TC(\eta) = k_u D \leq \frac{\rho_{max}(\eta)}{D} D = \rho_{max}(\eta)$.

$$\begin{aligned} Z(\eta) &= \rho_{max}(\eta) + TC(\eta) \\ &\leq 2\rho_{max}(\eta) \end{aligned} \tag{5.1}$$

With the same analysis of Theorem 4.2, we know $C_{max}(opt) = C_{max}(\eta) \geq \rho_{max}(\eta) - D$.

$$\begin{aligned} Z(opt) &= C_{max}(opt) + \lceil \frac{n}{C} \rceil D \\ &\geq \rho_{max}(\eta) - D + D \\ &= \rho_{max}(\eta) \end{aligned} \tag{5.2}$$

So, $\frac{Z(\eta)}{Z(opt)} \leq 2$.

Case 2: $\lceil \frac{n}{C} \rceil \geq 2$

In this case, $TC(\eta) = (k_f + k_u)D$. Combining the two inequalities $k_u \leq \frac{\rho_{max}(\eta)}{D}$ and $k_f C + k_u \leq n$, we have $k_f + k_u \leq \frac{1}{C}(n + \frac{C-1}{D}\rho_{max}(\eta))$, which means $TC(\eta) \leq \frac{n}{C}D + \frac{C-1}{C}\rho_{max}(\eta)$. Therefore,

$$\begin{aligned} Z(\eta) &= \rho_{max}(\eta) + TC(\eta) \\ &\leq \frac{2C-1}{C}\rho_{max}(\eta) + \frac{n}{C}D \\ &\leq \frac{2C-1}{C}C_{max}(\eta) + (\frac{2C-1}{C} + \frac{n}{C})D. \end{aligned} \tag{5.3}$$

As

$$\begin{aligned} \frac{2C-1}{C} + \frac{n}{C} &\leq 2 + \lceil \frac{n}{C} \rceil \\ &\leq 2\lceil \frac{n}{C} \rceil \end{aligned} \tag{5.4}$$

then

$$\begin{aligned} Z(\eta) &\leq \frac{2C-1}{C}C_{max}(\eta) + 2\lceil \frac{n}{C} \rceil D \\ &\leq 2(C_{max}(\eta) + \lceil \frac{n}{C} \rceil D) = 2Z(opt) \end{aligned} \tag{5.5}$$

For a instance which only contains a requirement with zero preparation time released at time 0, the solution of the algorithm $H2$ is $2D$, while the off-line optimal solution is D. Therefore, the competitive ratio of $H2$ for Problem P2 is 2. $\square$

Both the lower bound and the algorithm above for problem P2 are modified from those of problem P1. Actually, we can have better results for some special cases discussed below.

5.3 The Lower Bound for P2 with C=2

Theorem 5.3. *No on-line algorithm for P2 with $C = 2$ can have competitive ratio less than $1 + \theta$, even though all processing times are 0.*

Proof Consider instances of P2 with $C = 2$ as follows. Requirement R_1 with zero preparation time arrives at time $r_1 = 0$. Suppose that an on-line algorithm H delivers R_1 at time ρ_1. If $\rho_1 \geq \theta D$, there is no requirement coming and the off-line optimal schedule will deliver the requirement R_1 at time 0; thus the ratio of H satisfies that $R = \frac{\rho_1 + D}{D} \geq \frac{\theta D + D}{D} = 1 + \theta$ (see Fig 5.2).

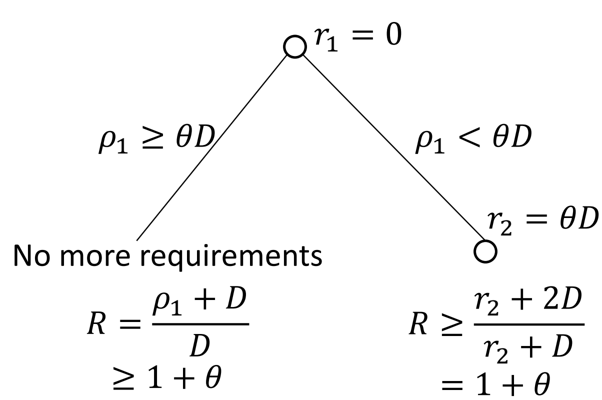

Fig 5.2: The Lower Bound for P2 with $C = 2$

If $\rho_1 < \theta D$, the instance releases another requirement R_2 with zero preparation time at time $r_2 = \theta D$ and the off-line optimal schedule will deliver the two requirements in one batch at time θD, while H deliver the two requirements in different batches and the competitive ratio satisfies that $R \geq \frac{r_2+2D}{\theta D+D} \geq \frac{\theta D+2D}{\theta D+D} = 1 + \frac{D}{\theta D+2D} = 1 + \frac{1}{\theta+1} = 1 + \theta$. Thus, we can conclude that the competitive ratio can not be less than $1 + \theta$. $\square$

5.4 The Upper Bound for P2 with C=2

In this section we give an on-line algorithm which aims at Problem P2 with $C = 2$ and show that this algorithm is on-line optimal.

Algorithm CTH Requirements are scheduled on the processor without idle time and re-indexed by the completion time. When a requirement $\mathcal{R}_j$ is prepared and the delivery rules are instituted as follows:

If there are two prepared requirements, then there is a batch to deliver them at once.

If $\mathcal{R}_j$ is the only prepared requirement but there are other unprepared requirements, then do nothing.

If $\mathcal{R}_j$ is the only prepared requirement, there is no unprepared requirement and j is even, then there is a batch to deliver it at once.

If $\mathcal{R}_j$ is the only prepared requirement, there is no unprepared requirements and j is odd, then wait until $C_j + \theta D$ and deliver it in a batch; or other requirements arrive during the period.

Let η be the schedule obtained by the algorithm CTH, $k_j(\eta)$ be the number of batches to deliver $\mathcal{R}_1, \cdots, \mathcal{R}_j$. We first give a lemma about the requirements which are in a batch alone.

Lemma 5.1. *If requirement $\mathcal{R}_j$ is in a batch alone in the schedule η, then we have*

$$\frac{\rho_j(\eta)+k_j(\eta)D}{C_j(opt)+\lceil \frac{j}{2}\rceil D}\leq 1+\theta \tag{5.6}$$

Proof Noticing that $C_j(opt)=C_j(\eta)$, it is equivalent to prove

$$\frac{\rho_j(\eta)+k_j(\eta)D}{C_j(\eta)+\lceil \frac{j}{2}\rceil D}\leq 1+\theta \tag{5.7}$$

Suppose that there are s requirements $\mathcal{R}_{l_i}$ $(i=1,2,\ldots,s)$ which are in a batch alone in the schedule η, and we will use induction on i.

First, let us show that the statement of the lemma holds for the case $i=1$. It is obvious that l_1 is odd and $k_{l_1}(\eta)=\frac{l_1+1}{2}$. Therefore,

$$\begin{aligned}
&\frac{\rho_{l_1}(\eta)+k_{l_1}(\eta)D}{C_{l_1}(\eta)+\lceil \frac{l_1}{2}\rceil D}\\
=&\frac{C_{l_1}(\eta)+\theta D+\frac{l_1+1}{2}D}{C_{l_1}(\eta)+\frac{l_1+1}{2}D}\\
\leq& 1+\frac{\theta D}{C_{l_1}(\eta)+\frac{l_1+1}{2}D}\\
\leq& 1+\frac{\theta D}{D}=1+\theta
\end{aligned} \tag{5.8}$$

Assume now that the statement of the lemma is proven for $i\leq m$ (this is the induction hypothesis), and consider the case $i=m+1$.

Case 1: l_{m+1} is odd

Obviously l_m is even and we have $\rho_{l_{m+1}}(\eta)=C_{l_{m+1}}(\eta)+\theta D$, $\rho_{l_m}(\eta)=C_{l_m}(\eta)$ and $k_{l_{m+1}}(\eta)=k_{l_m}(\eta)+\frac{l_{m+1}-l_m+1}{2}$

From the induction hypothesis, we have that

$$\begin{aligned}
&C_{l_m}(\eta)+k_{l_m}(\eta)D\leq(1+\theta)C_{l_m}(\eta)+(1+\theta)\frac{l_m}{2}D\\
\Rightarrow\ & k_{l_m}(\eta)D\leq\theta C_{l_m}(\eta)+(1+\theta)\frac{l_m}{2}D\leq\theta C_{l_{m+1}}(\eta)+(1+\theta)\frac{l_m}{2}D.
\end{aligned} \tag{5.9}$$

Consequently, we have

$$
\begin{aligned}
&\rho_{l_{m+1}}(\eta) + k_{l_{m+1}}(\eta)D \\
&= C_{l_{m+1}}(\eta) + \theta D + k_{l_m}(\eta)D + \frac{l_{m+1} - l_m + 1}{2}D \\
&\leq C_{l_{m+1}}(\eta) + \theta D + \theta C_{l_{m+1}}(\eta) + (1+\theta)\frac{l_m}{2}D + \frac{l_{m+1} - l_m + 1}{2}D \\
&\leq (1+\theta)C_{l_{m+1}}(\eta) + (1+\theta)\frac{l_m}{2}D + (1+\theta)\frac{l_{m+1} - l_m + 1}{2}D \qquad \left(\frac{l_{m+1} - l_m + 1}{2} \geq 1\right) \\
&= (1+\theta)(C_{l_{m+1}}(\eta) + \frac{l_{m+1}+1}{2}D) \\
&= (1+\theta)(C_{l_{m+1}}(\eta) + \lceil\frac{l_{m+1}}{2}\rceil D)
\end{aligned}
\tag{5.10}
$$

Case 2: l_{m+1} is even

Obviously l_m is odd and we have $\rho_{l_{m+1}}(\eta) = C_{l_{m+1}}(\eta)$, $\rho_{l_m}(\eta) = C_{l_m}(\eta) + \theta D$ and $k_{l_{m+1}}(\eta) = k_{l_m}(\eta) + \frac{l_{m+1}-l_m+1}{2}$.

From the induction hypothesis, we have that

$$
C_{l_m}(\eta) + \theta D + k_{l_m}(\eta)D \leq (1+\theta)C_{l_m}(\eta) + (1+\theta)\frac{l_m+1}{2}D \tag{5.11}
$$

$$
\begin{aligned}
\Rightarrow k_{l_m}(\eta)D &\leq \theta C_{l_m}(\eta) + (1+\theta)\frac{l_m+1}{2}D - \theta D \\
&= \theta(C_{l_m}(\eta) + \theta D) + (1+\theta)\frac{l_m+1}{2}D - (\theta^2+\theta)D \\
&= \theta\rho_{l_m}(\eta) + (1+\theta)\frac{l_m+1}{2}D - D \\
&\leq \theta C_{l_{m+1}}(\eta) + (1+\theta)\frac{l_m+1}{2}D - D
\end{aligned}
\tag{5.12}
$$

Consequently, we have

$$
\begin{aligned}
&\rho_{l_{m+1}}(\eta) + k_{l_{m+1}}(\eta)D \\
&= C_{l_{m+1}}(\eta) + k_{l_m}(\eta) + \frac{l_{m+1} - l_m + 1}{2}D \\
&\le C_{l_{m+1}}(\eta) + \theta C_{l_{m+1}}(\eta) + (1+\theta)\frac{l_m + 1}{2}D - D + \frac{l_{m+1} - l_m + 1}{2}D \\
&= (1+\theta)C_{l_{m+1}}(\eta) + (1+\theta)\frac{l_m + 1}{2}D + \frac{l_{m+1} - l_m - 1}{2}D \\
&\le (1+\theta)C_{l_{m+1}}(\eta) + (1+\theta)\frac{l_m + 1}{2}D + (1+\theta)\frac{l_{m+1} - l_m - 1}{2} \qquad (\frac{l_{m+1} - l_m - 1}{2} \ge 0) \\
&= (1+\theta)(C_{l_{m+1}}(\eta) + \frac{l_{m+1}}{2}D) \\
&= (1+\theta)(C_{l_{m+1}}(\eta) + \lceil\frac{l_{m+1}}{2}\rceil D)
\end{aligned}
\tag{5.13}
$$

This completes the proof. □

Theorem 5.4. *The competitive ratio of the on-line algorithm CTH for P2 with $C = 2$ is $1 + \theta$.*

Proof Noticing that $Z(opt) = C_{max}(opt) + \lceil\frac{n}{2}\rceil D = C_{max}(\eta) + \lceil\frac{n}{2}\rceil D$.

Case 1: There is no requirement which is in a batch alone in η, then η is optimal.

Case 2: If requirement $\mathcal{R}_n$ is in a batch alone in η, then from Lemma 5.1 the statement of the theorem holds obviously.

Case 3: If requirement $\mathcal{R}_n$ is not in a batch alone in η but there is a requirement which is in a batch alone before, then we assume the biggest index of such requirements is m

where n and m have the same parity. Therefore, we have

$$\begin{aligned}
Z(\eta) &= D_{max}(\eta) + k_n(\eta)D \\
&= \rho_n(\eta) + k_n(\eta)D \\
&= C_n(\eta) + k_m(\eta)D + \frac{n-m}{2}D \\
&= \rho_m(\eta) + k_m(\eta)D + (C_n(\eta) - \rho_m(\eta)) + \frac{n-m}{2}D \\
&\le (1+\theta)(C_m(\eta) + \lceil\frac{m}{2}\rceil D) + (C_n(\eta) - C_m(\eta)) + \frac{n-m}{2}D \\
&\le (1+\theta)(C_m(\eta) + \lceil\frac{m}{2}\rceil D) + (1+\theta)((C_n(\eta) - C_m(\eta)) + \frac{n-m}{2}D) \\
&= (1+\theta)(C_n(\eta) + \lceil\frac{n}{2}\rceil D) \\
&= (1+\theta)Z(opt)
\end{aligned} \tag{5.14}$$

According to Theorem 5.3, the ratio of CTH can not be less than $1+\theta$, which completes the proof. □

5.5 Discussion and Concluding Remarks

For Problem P2, the number of vehicles is infinite but the capacity of vehicles is a fixed constant. The lower bound of P2 is different from that of P1. Even though we modify the proof of lower bound for P1, we can not attain a good enough lower bound, as the fixed capacity is a very important constraint.

Neither the lower bound nor the upper bound for P2 in the first two sections are the best. Actually, through the study on the special case of P2 with $C = 2$, we can conclude that the better lower bound and upper bound should be according to C.

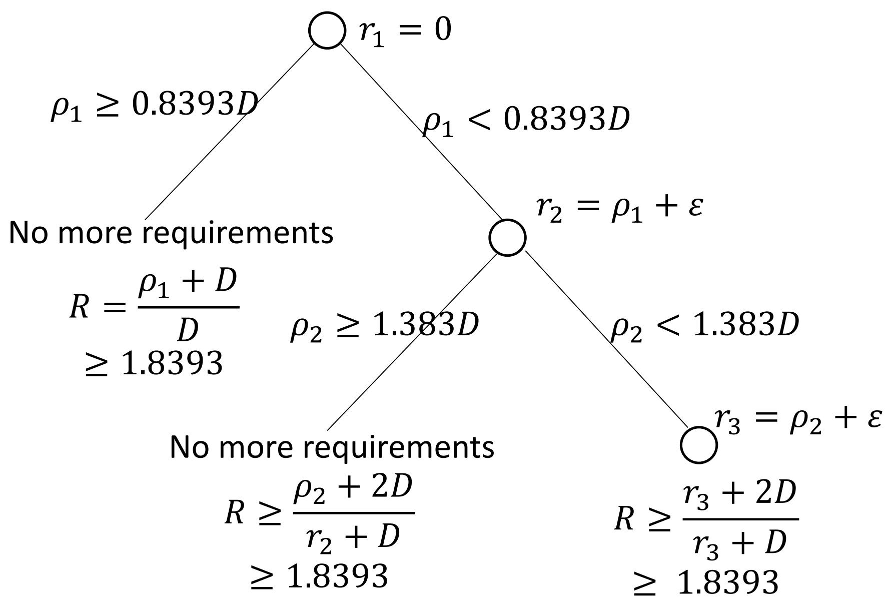

Fig 5.3: Improved Lower Bound for P2 with $C = 3$

Using the similar method of Theorem 5.3, we can also get a greater lower bound of ratio for P2 with C being a fixed constant, as 1.8393 for $C = 3$ (Fig 5.3), 1.9275 for $C = 4$ (Fig 5.4) and 1.9756 for $C = 5$. We can see that as C tends to infinity, the lower bound tends to 2 which agrees with the result of Theorem 4.1. However, for a fixed C that is greater than 2, it is hard to get an optimal on-line algorithm just like $C = 2$ and the algorithm $H2$ is good enough for these conditions.

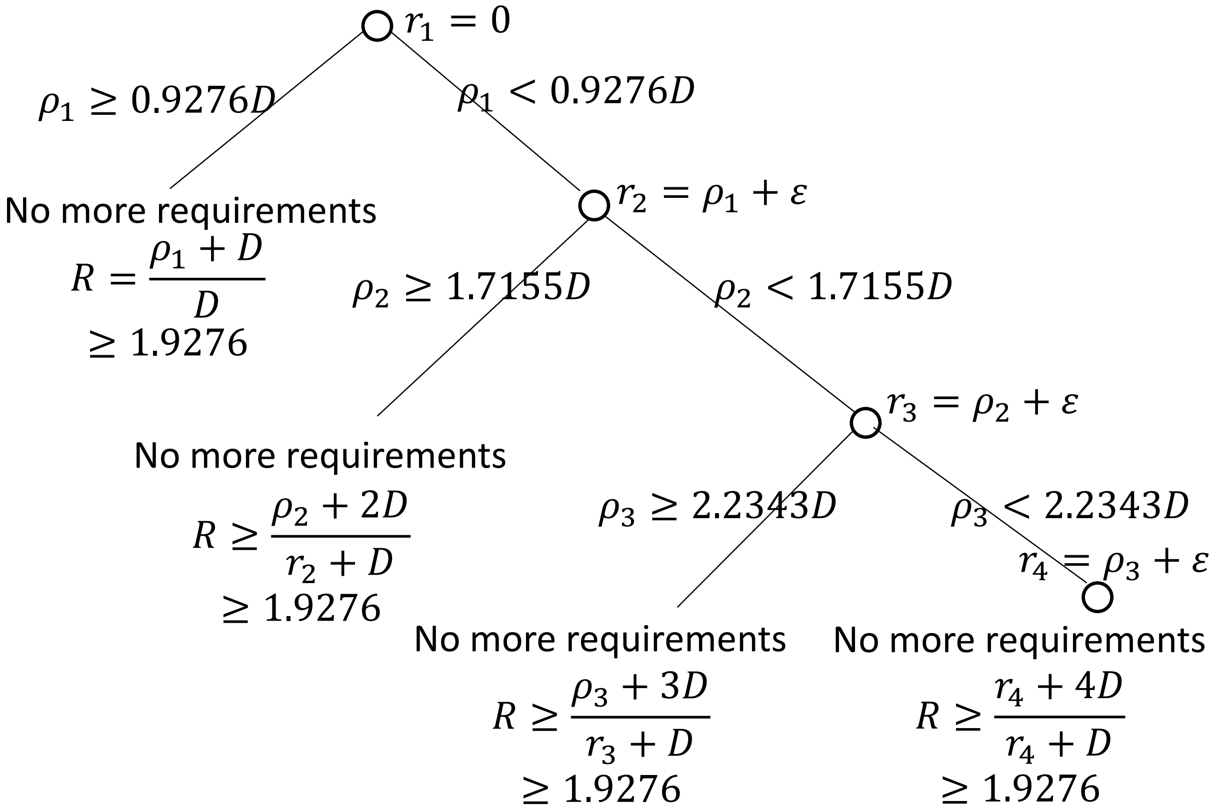

Fig 5.4: Improved Lower Bound for P2 with $C = 4$

CHAPTER 6
SOLVING PROBLEM P3

In the next two chapters, we begin to consider the cases where the number of vehicles is one. In this situation,the same assumption $T = 0$ is no longer valid. The optimal schedule of the off-line version of P3 is to prepare requirements without idle time and to deliver all requirements in one batch at the time when the last requirement is prepared. So the off-line optimal value is $Z(opt) = C_{max}(opt) + T + D$.

6.1 The Lower Bound for P3

Theorem 6.1. *No on-line algorithm for P3 can have competitive ratio less than* $max\{1+\theta, 1+\sqrt{\frac{D}{T+D}}\}$, *even though all preparing times are 0.*

Proof At first, we show that $1+\theta$ is a lower bound for the problem P3. Requirement $\mathcal{R}_1$ with zero preparation time arrives at time $r_1 = 0$. Suppose that an on-line algorithm H delivers $\mathcal{R}_1$ at time ρ_1. If $\rho_1 \geq \theta(T+D)$, there is no requirement coming and the off-line optimal schedule will deliver the requirement $\mathcal{R}_1$ at time 0, therefore the ratio of H satisfies that $R = \frac{\rho_1+T+D}{T+D} \geq \frac{\theta(T+D)+T+D}{T+D} = 1+\theta$.

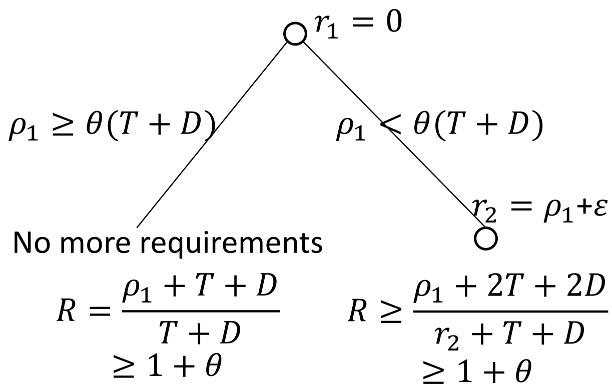

Fig 6.1: The Lower Bound for P3 (1)

If $\rho_1 < \theta(T+D)$, the instance releases another requirement $\mathcal{R}_2$ with zero preparation time at time $r_2 = \rho_1 + \varepsilon$ (where ε is a sufficiently small number) and the off-line optimal schedule will deliver the two requirements in one batch at time r_2 (see Fig 6.1). H delivers the two requirements in different batches and the competitive ratio satisfies that

$$
\begin{aligned}
R &\geq \frac{\rho_1 + 2T + 2D}{r_2 + T + D} \\
&= \frac{\rho_1 + 2T + 2D}{\rho_1 + \varepsilon + T + D} \\
&= 1 + \frac{T + D - \varepsilon}{\rho_1 + T + D + \varepsilon} \\
&> 1 + \frac{T + D - \varepsilon}{\theta(T+D) + T + D + \varepsilon} \qquad\qquad (\rho_1 < \theta(T+D)) \\
&= 1 + \frac{T + D - \varepsilon}{(1+\theta)(T+D) + \varepsilon}
\end{aligned}
\tag{6.1}
$$

As ε tends to 0, the right side of the inequality tends to $1+\theta$. Thus, we conclude that the competitive ratio cannot be less than $1+\theta$.

Then, we will use the similar method of Theorem 4.1 to prove the the other part of the theorem. The instance releases a requirement with zero preparation time at time $r_1 = 0$, and if an on-line algorithm H delivers the requirement at time $\rho_1 \geq \sqrt{D(T+D)}$, then there is no requirement coming. Otherwise, the second requirement with zero preparation time arrives at time $r_2 = \sqrt{D(T+D)}$, and if the departure time of this requirement $\rho_2 \geq 2\sqrt{D(T+D)}$, then there is no requirement coming, or else the third requirement with zero preparation time comes at time $r_3 = 2\sqrt{D(T+D)}$, and so on (see Fig 6.2). If the algorithm H delivers the ith requirement with zero preparation time at time $\rho_i \geq i\sqrt{D(T+D)}$, then there is no requirement coming, otherwise the $(i+1)$th job with zero preparation time comes at time $r_{i+1} = i\sqrt{D(T+D)}$. The process is repeated until at most N requirements have been released and delivered.

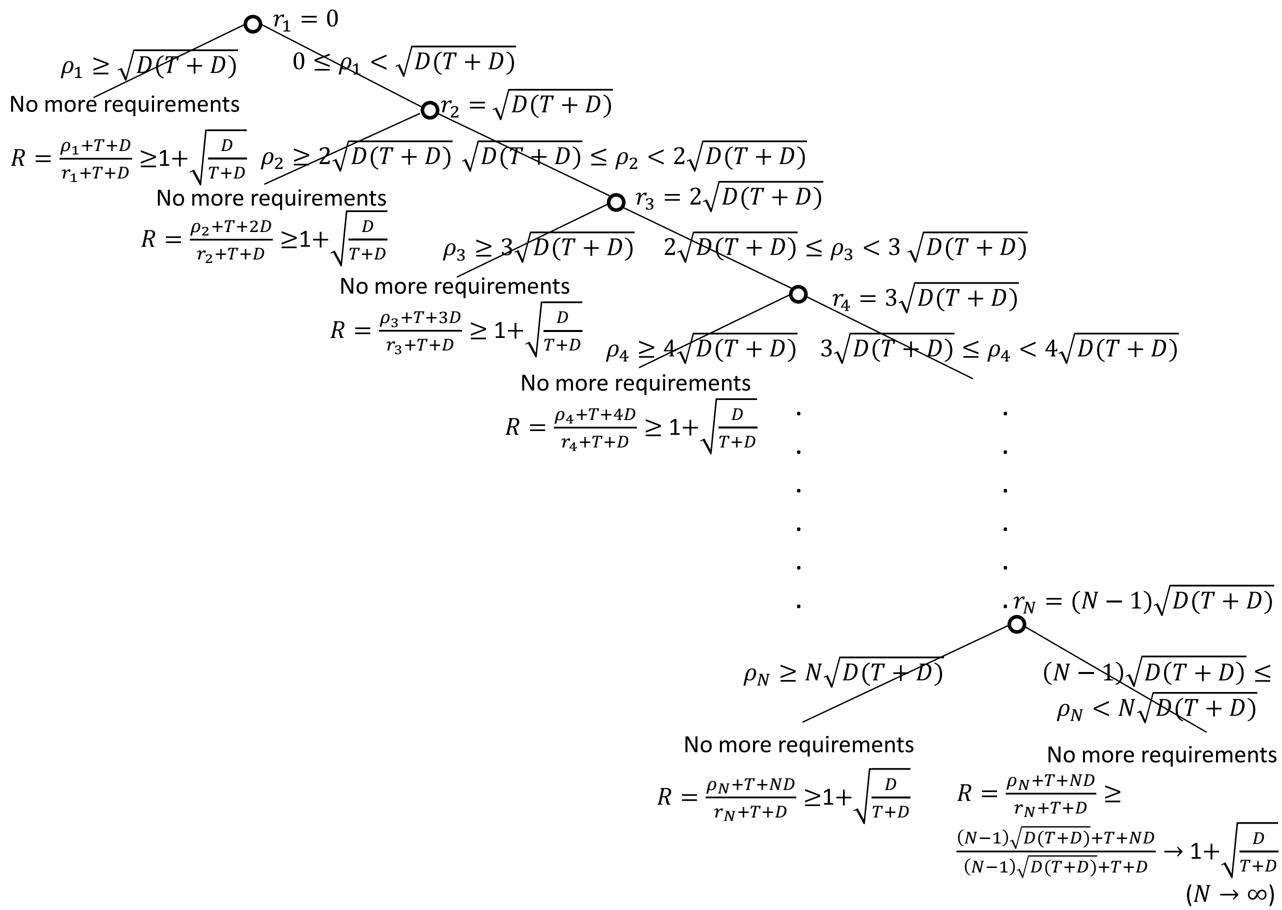

Fig 6.2: The Lower Bound for P3 (2)

If the instance at last has released and delivered k requirements, where $k < N$, then we can get that the k requirements are delivered in k different batches and $\rho_k \geq k\sqrt{D(T+D)}$. So the objective value of the schedule η obtained by the algorithm H is $Z(\eta) = \rho_k + T + kD \geq k(D + \sqrt{D(T+D)}) + T$, while the off-line optimal schedule delivers all the requirements in a batch at time $(k-1)\sqrt{D(T+D)}$ and the value is $Z(opt) = (k-1)\sqrt{D(T+D)} + T + D$. Then we can get that

$$\begin{aligned} R &\geq \frac{k(D + \sqrt{D(T+D)}) + T}{(k-1)\sqrt{D(T+D)} + T + D} \\ &= 1 + \frac{(k-1)D + \sqrt{D(T+D)}}{(k-1)\sqrt{D(T+D)} + T + D} \\ &= 1 + \sqrt{\frac{D}{T+D}} \end{aligned} \tag{6.2}$$

If the instance at last has released and delivered N requirements, then we can get that the N requirements are delivered in N batches and $\rho_N \geq r_N = (N-1)\sqrt{D(T+D)}$. So the objective value of the schedule η obtained by the algorithm H is $Z(\eta) = \rho_N + T + ND \geq (N-1)\sqrt{D(T+D)} + T + ND$, while the off-line optimal schedule delivered all the requirements in a batch at time $(N-1)\sqrt{D(T+D)}$ and the value is $Z(opt) = (N-1)\sqrt{D(T+D)} + T + D$. As N gets infinitely great, the ratio will tend to $\frac{\sqrt{D(T+D)}+D}{\sqrt{D(T+D)}} = 1 + \sqrt{\frac{D}{T+D}}$

Then we have proved the statement of this theorem. $\square$

From Theorem 6.1, we actually get that when $T > (1+\theta)D$ the lower bound for problem P3 is $1+\theta$, and when $T \leq (1+\theta)D$ the lower bound is $1 + \sqrt{\frac{D}{T+D}}$. Next, we will give an algorithm which achieves the best possible solution.

6.2 The Upper Bound for P3

Algorithm H3 Requirements are scheduled on the processor without idle time.

When $T > (1+\theta)D$, at the time of $\theta(T+D)+lT$ where $l \geq 0$, if there is no uncompleted requirement, then there is a batch to deliver all completed requirements; otherwise, there is no operation.

When $T \leq (1+\theta)D$, at the time of $l\sqrt{D(T+D)}$ where $l \geq 1$, if there is no uncompleted requirement, then there is a batch to deliver all completed requirements; otherwise, there is no operation.

Noticing that when $T \leq (1+\theta)D$, $\sqrt{D(T+D)} \geq T$, so the algorithm $H3$ is feasible.

Theorem 6.2. *The competitive ratio of on-line algorithm $H3$ for P3 is* $max\{1+\theta, 1+\sqrt{\frac{D}{T+D}}\}$.

Proof Let η be the schedule obtained by the algorithm $H3$. As the algorithm has two different cases, the proof also has two cases.

Case 1: $T > (1+\theta)D$

Suppose that $\rho_{max}(\eta) = \theta(T+D) + kT$.

Case 1.1: $k = 0$

As there is no batch after time $\theta(T+D)$, the only delivery in η happens at $\theta(T+D)$. Therefore, $Z(\eta) = \theta(T+D) + T + D = (1+\theta)(T+D)$. Because the optimal value $Z(opt) \geq T + D$, the statement holds.

Case 1.2: $k \geq 1$

In this case, we get that $\rho_{max}(\eta) = \theta(T+D) + kT$, and $D_{max}(\eta) = \rho_{max}(\eta) + T =$

$\theta(T+D)+(k+1)T$. As there are at most k batches after $\theta(T+D)$, with one possible batch at $\theta(T+D)$ the cost of delivery in η satisfies that $TC(\eta) \leq (k+1)D$. While $C_{max}(opt) \geq \rho_{max}(\eta) - T = \theta(T+D)+(k-1)T$, so we have

$$\begin{aligned}\frac{Z(\eta)}{Z(opt)} &\leq \frac{\theta(T+D)+(k+1)(T+D)}{\theta(T+D)+kT+D} \\ &= 1+\frac{T+kD}{\theta(T+D)+kT+D}\end{aligned} \tag{6.3}$$

The term

$$\begin{aligned}&\frac{T+kD}{\theta(T+D)+kT+D} \\ =&\frac{T+\frac{D}{T}(kT+\theta(T+D)+D-\theta(T+D)-D)}{\theta(T+D)+kT+D} \\ =&\frac{D}{T}+\frac{T^2-\theta D(T+D)-D^2}{T(\theta(T+D)+kT+D)}\end{aligned} \tag{6.4}$$

Because $T > (1+\theta)D$, $T^2-\theta D(T+D)-D^2 > 0$ and the whole term increases as k decreases.

Thus,

$$\begin{aligned}\frac{Z(\eta)}{Z(opt)} &\leq 1+\frac{T+D}{\theta(T+D)+T+D} \\ &= 1+\theta\end{aligned} \tag{6.5}$$

Case 2: $T \leq (1+\theta)D$

Using the similar analysis of Theorem 4.1, we have $TC(\eta) \leq \frac{\rho_{max}(\eta)}{\sqrt{D(T+D)}}D$ and $C_{max}(opt) \geq$

$\rho_{max}(\eta) - \sqrt{D(T+D)}$. Therefore,

$$\begin{aligned}
\frac{Z(\eta)}{Z(opt)} &\leq \frac{\rho_{max}(\eta) + T + \frac{\rho_{max}(\eta)}{\sqrt{D(T+D)}}D}{\rho_{max}(\eta) - \sqrt{D(T+D)} + T + D} \\
&= 1 + \frac{D}{\sqrt{D(T+D)}} + \frac{T - (1 + \frac{D}{\sqrt{D(T+D)}})(T + D - \sqrt{D(T+D)})}{\rho_{max}(\eta) - \sqrt{D(T+D)} + T + D} \\
&= 1 + \frac{D}{\sqrt{D(T+D)}} + \frac{T - (T + D - \sqrt{D(T+D)} + \sqrt{D(T+D)} - D)}{\rho_{max}(\eta) - \sqrt{D(T+D)} + T + D} \qquad (6.6) \\
&= 1 + \frac{D}{\sqrt{D(T+D)}} \\
&= 1 + \sqrt{\frac{D}{T+D}}
\end{aligned}$$

This completes the proof. □

6.3 Discussion and Concluding Remarks

For the problem in this chapter, the assumption $T = 0$ is not correct. Therefore, solving P3 has to involve to two parameters D and T. Actually, the lower bound of P3 has two parts according to different relationships between D and T. To achieve the on-line optimality, the algorithm $H3$ operates to the two cases respectively, and it was proved that the performance of it is the best for the worst case as the competitive ratio equals the lower bound $max\{1 + \theta, 1 + \sqrt{\frac{D}{T+D}}\}$. In other words, $H3$ is robust with all possible instances and different pairs of (T, D).

CHAPTER 7
SOLVING PROBLEM P4

When there is only one vehicle and the capacity of the vehicle is finite, the problem changes considerably. It is reasonable to allow in this case the preparation of medical resources to be prepared not once but in several actions, the concept of which is called preemption. The abbreviation *pmtn* is used in the five-field notation for the problem description.

The shortest remaining processing time (SRPT) rule is to prepare the requirement with the smallest remaining preparation time among all already released uncompleted requirements at any instant t. The SRPT rule is important to the preemption problem and it was applied in the off-line version of P4 [Lu et al. 2008]. The off-line optimal solution prepares the requirements with the SRPT rule. All prepared requirements are divided into batches by the completion time. Every batch, apart from the first batch, contains exactly C jobs. A ready batch is delivered whenver the vehicle is available.

7.1 The Lower Bound for P4

Theorem 7.1. *No on-line algorithm for P4 can have 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 though all preparation times are 0.*

Proof The proof of the $1+\theta$ part is the same with Theorem 6.1. Therefore, we only need to prove the part of $1+\sqrt{\frac{D}{T+D}}-\frac{\sqrt{D(T+D)}}{(C-1)\sqrt{D(T+D)}+T+D}$. The instance releases a requirement with zero preparation time at time $r_1 = 0$, and if an on-line algorithm H delivers the requirement at time $\rho_1 \geq \sqrt{D(T+D)}$, then there is no requirement coming. Otherwise, the second requirement with zero preparation time arrives at time $r_2 = \sqrt{D(T+D)}$, and if the departure time of this requirement $\rho_2 \geq 2\sqrt{D(T+D)}$, then there is no requirement coming, or else the third requirement with zero preparation time comes at time $r_3 = 2\sqrt{D(T+D)}$, and so on (see Fig 7.1). If the algorithm H delivers the ith requirement with zero preparation time at time $\rho_i \geq i\sqrt{D(T+D)}$, then there is no requirement coming, otherwise the $(i+1)$th job with zero preparation time comes at time $r_{i+1} = i\sqrt{D(T+D)}$. The process is repeated until at most C requirements have been released and delivered.

If the instance at last has released and delivered k requirements, where $k < C$, then we can get that the k requirements are delivered in k different batches and $\rho_k \geq k\sqrt{D(T+D)}$. So the objective value of the schedule η obtained by the algorithm H is $Z(\eta) = \rho_k + T + kD \geq k(D+\sqrt{D(T+D)}) + T$, while the off-line optimal schedule delivers all the requirements in a batch at time $(k-1)\sqrt{D(T+D)}$ and the value is $Z(opt) = (k-1)\sqrt{D(T+D)} + T + D$. Then we can get that

$$\begin{aligned} R &\geq \frac{k(D+\sqrt{D(T+D)})+T}{(k-1)\sqrt{D(T+D)}+T+D} \\ &= 1+\frac{(k-1)D+\sqrt{D(T+D)}}{(k-1)\sqrt{D(T+D)}+T+D} \\ &= 1+\sqrt{\frac{D}{T+D}} \end{aligned} \tag{7.1}$$

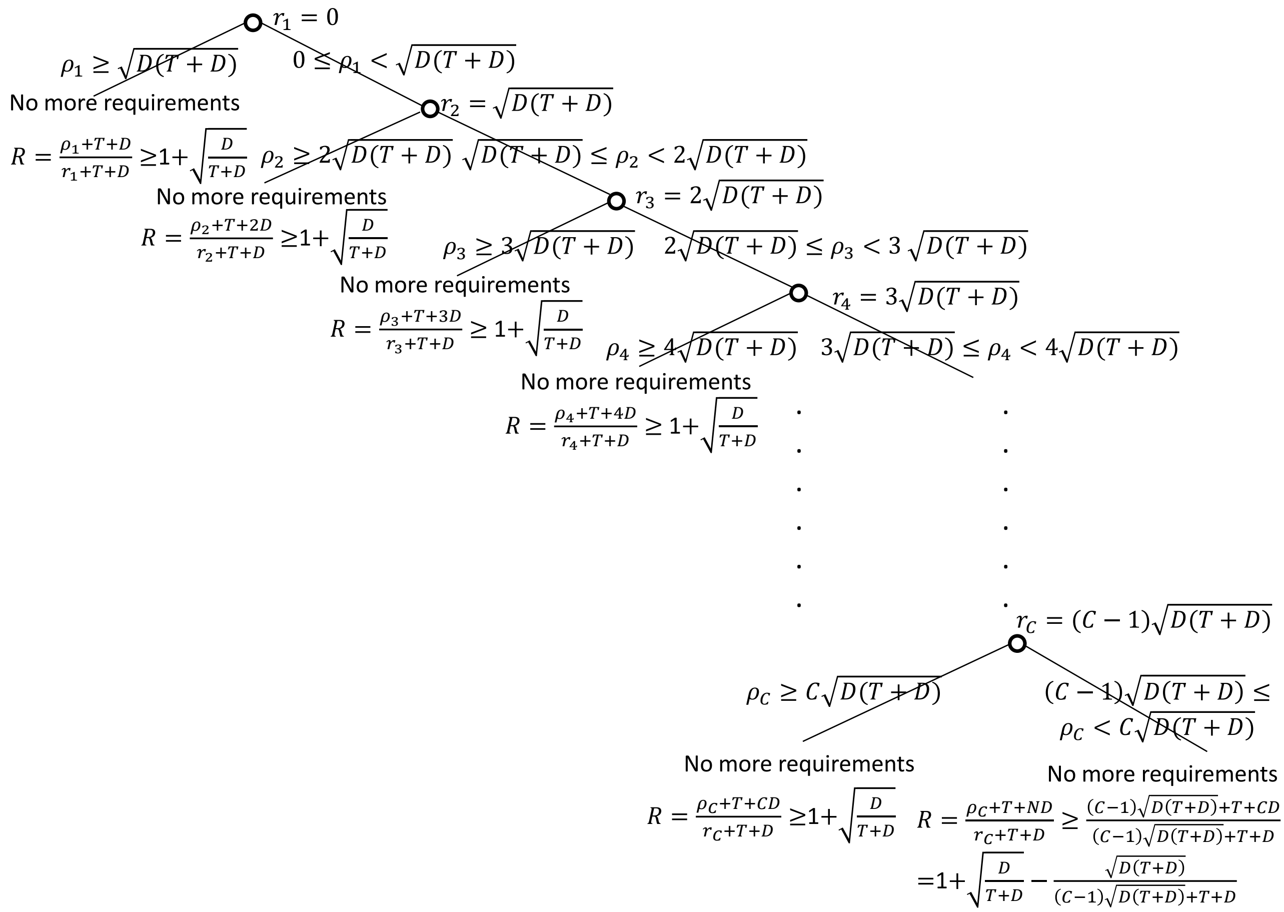

Fig 7.1: The Lower Bound for P4

If the instance at last has released and delivered C requirements, we can get that the C requirements are delivered in C batches and $\rho_C \geq r_C = (C-1)\sqrt{D(T+D)}$. So the objective value of the schedule η obtained by the algorithm H is $Z(\eta) = \rho_C + T + CD \geq (C-1)\sqrt{D(T+D)} + T + CD$, while the off-line optimal schedule delivered all the requirements in a batch at time $(C-1)\sqrt{D(T+D)}$ and the value is $Z(opt) = (C-1)\sqrt{D(T+D)} + T + D$. Thus the competitive ratio is at least $1 + \sqrt{\frac{D}{T+D}} - \frac{\sqrt{D(T+D)}}{(C-1)\sqrt{D(T+D)}+T+D}$. $\square$

7.2 The Upper Bound for P4

Next, we modify $H3$ to give an algorithm for P4 which has competitive ratio of $max\{1+\theta, 1+\sqrt{\frac{D}{T+D}}\}$.

Algorithm H4 Requirements are scheduled on the machine with the SRPT rule.

When $T > (1+\theta)D$, at the time of $\theta(T+D)+lT$ where $l \geq 0$, there is a batch to deliver as many completed requirements as possible.

When $T \leq (1+\theta)D$, at the time of $l\sqrt{D(T+D)}$ where $l \geq 1$, there is a batch to deliver as many completed requirements as possible.

Theorem 7.2. *The competitive ratio of the on-line algorithm $H4$ for P4 is $max\{1+\theta, 1+\sqrt{\frac{D}{T+D}}\}$.*

Proof Let η be the schedule obtained by the algorithm $H4$. As the algorithm has two different cases, the proof also have two cases.

Case 1: $T > (1+\theta)D$

Suppose there are k periods of T in the schedule η after time $\theta(T+D)$.

Case 1.1: $k = 0$

As there are no batches after time $\theta(T+D)$, the only delivery in η happens at $\theta(T+D)$. Therefore, $Z(\eta) = \theta(T+D)+T+D = (1+\theta)(T+D)$. Because the off-line optimal value $Z(opt) \geq T+D$, the statement holds.

Case 1.2: $k \geq 1$

Suppose ν is the last delivery time before $\rho_{max}(\eta) = \theta(T+D)+kT$ when there is an unfull batch. If there is always a full batch at all the deliver times before $\rho_{max}(\sigma)$, then

$\nu = 0$. Let m be the number of full batches between ν and $\rho_{max}(\eta)$, so there are $m+1$ batches after time ν, which means $\rho_{max}(\eta) \leq \nu + (m+1)T$ $(\theta(T+D) < T)$.

As π also prepares the requirements with SRPT rule, there are more than mC requirements completed after ν. Therefore, $D_{max}(opt) \geq \nu + (m+1)T \geq \rho_{max}(\eta)$ and $TC(opt) \geq (m+1)D$.

$$\begin{aligned} \frac{Z(\eta)}{Z(opt)} &\leq \frac{\theta(T+D) + (k+1)(T+D)}{\theta(T+D) + kT + (m+1)D} \\ &\leq \frac{\theta(T+D) + (k+1)(T+D)}{\theta(T+D) + kT + D} \\ &\leq 1 + \theta \end{aligned} \tag{7.2}$$

Case 2: $T \leq (1+\theta)D$

In the following, we will use the similar analysis for case 1.2 to get our result. Suppose that ν is the last delivery time before $\rho_{max}(\eta)$ when there is an unfull batch. Let m be the number of full batches between ν and $\rho_{max}(\eta)$, so there are $m+1$ batches after time ν, that means $\rho_{max}(\eta) = \nu + (m+1)\sqrt{D(T+D)}$. While $D_{max}(opt) \geq \nu + (m+1)T = \rho_{max}(\eta) + (m+1)(T - \sqrt{D(T+D)})$ and $TC(opt) \geq (m+1)D$.

$$\begin{aligned} \frac{Z(\eta)}{Z(opt)} &\leq \frac{\rho_{max}(\eta) + T + \frac{\rho_{max}(\eta)}{\sqrt{D(T+D)}}D}{\rho_{max}(\eta) + (m+1)(T + D - \sqrt{D(T+D)})} \\ &= 1 + \frac{D}{\sqrt{D(T+D)}} + \frac{T - (1 + \frac{D}{\sqrt{D(T+D)}})(m+1)(T + D - \sqrt{D(T+D)})}{\rho_{max}(\eta) + (m+1)(T + D - \sqrt{D(T+D)})} \\ &\leq 1 + \frac{D}{\sqrt{D(T+D)}} \\ &= 1 + \sqrt{\frac{D}{T+D}} \end{aligned} \tag{7.3}$$

□

7.3 Discussion and Concluding Remarks

When there are constraints on both the capacity of vehicles and the number of vehicles, the problem becomes more difficult. Problem P4 allows the requirements to be interrupted and restart later; otherwise even the off-line version of P4 is strongly NP-hard. We modified the lower bound of P3 to reach the lower bound of P4 and designed the corresponding algorithm $H4$; however, there is still a gap between the competitive ratio of this algorithm and the lower bound. This situation is similar to that with $H2$. To get better results, we need to improve both the lower bound and the algorithm in terms of C. When C is large, the original lower bound will approach the competitive ratio of $H4$; which represents the robustness of $H4$.

CHAPTER 8

APPLICATIONS

8.1 Introduction

In the preceding chapters, four problems of allocating medical resources in EM were proposed and algorithms for these problems were developed. For each algorithm, analysis of competitive ratio and lower bound were carried out. This analysis shows that the algorithms are robust. The analysis has also shown that some of the four problems are optimal.

All the four algorithms have a polynomial function of run-time performance. This is even true for Problem P4. In the algorithm for P4, the SRPT rule interrupts the preparation of every requirement at most n times. The time of scheduling is a low order polynomial of n ($O(n)$ for $H1$, $H2$, $H3$ or $O(n^2)$ for $H4$). For the delivery operation, the batches depart at the time the multiple of D (T, $\sqrt{D(T+D)}$), thus the number of departure points is at most $max\{n, \frac{max\{r_j\}+\sum p_j}{D}$ $(\frac{max\{r_j\}+\sum p_j}{T}, \frac{max\{r_j\}+\sum p_j}{\sqrt{D(T+D)}})\}$, which is also a polynomial function.

Chapter will present a simulated experiment to demonstrate the run-time performance of the algorithm ($H2$ for P2 in particular) in normal instances or scenarios and illustrate how the algorithms are used in practice.

8.2 Simulated Experiment

The hypothetic example meets the problem definition of P2. Characteristics of the example problem are as follows:

1. The release of requirements follows poisson distribution, i.e., $r_j - r_{j-1} \sim P(\lambda)$ for $j = 1, 2, \cdots, n$ where $r_0 = 0$.
2. The preparation time of requirements follows $p_j \sim U(0, b)$.
3. The delivery cost follows $D \sim N(\mu, \sigma^2)$.

Referring back to Chapter 5, we consider the capacity $C = 2$, $C = 5$, $C = 12$ for $H2$. And all the results are average of 100 program runs. We assume that the cost of one shipment is much stable, so we choose $\mu = 20$ and $\sigma = 1$ for D. The implementation of the algorithm was done in Matlab environment (see Appendix). Table 8.1 shows the result of $H2$ with $C = 2$

From Table 8.1, it can be seen that the competitive ratio does not exceed 1.23. In most cases, the algorithm value is very close to the optimal value of off-line version of P2. The worst case happens when $\lambda = 20$ and $b = 8$, i.e., the average time interval of requirement release is almost equal to the unit delivery cost D ($D \sim N(20, 1)$), and the preparation time is not very long. This case is consistent with the proof of competitive ratio in Theorem 5.2 in Chapter 5 where the requirements with zero preparation time release with the period of D. The result can be explained. When the requirement release has low frequency and the preparation time is not long, at every delivery point of $H2$ many requirements are delivered in unfull batches while the off-line optimal schedule could wait to deliver as many requirements as possible in full batches.

Table 8.1: Results of Algorithm $H2$ for Problem P2 with $C = 2$

Parameters	$Z(\eta)$	$Z(opt)$	Competitive Ratio	Run-time
$\lambda = 3,\ b = 8,\ n = 100$	1412.98	1401.90	1.0079	0.000310s
$\lambda = 3,\ b = 8,\ n = 1000$	14003.71	13992.78	1.0008	0.004719s
$\lambda = 10,\ b = 8,\ n = 100$	2133.67	2000.24	1.0668	0.000129s
$\lambda = 10,\ b = 8,\ n = 1000$	21153.30	20026.18	1.0564	0.004015s
$\lambda = 20,\ b = 8,\ n = 100$	3685.01	3008.62	1.2249	0.000133s
$\lambda = 20,\ b = 8,\ n = 1000$	36511.09	29927.65	1.2201	0.003968s
$\lambda = 50,\ b = 8,\ n = 100$	7012.19	5999.90	1.1687	0.000131s
$\lambda = 50,\ b = 8,\ n = 1000$	70054.60	60015.05	1.1672	0.003966s
$\lambda = 20,\ b = 30,\ n = 100$	3186.93	3019.67	1.0554	0.000130s
$\lambda = 20,\ b = 30,\ n = 1000$	31620.73	30020.82	1.0533	0.003943s

Further, the period of the requirement release in this case approximates to the unit delivery cost, the time-based objective function and the cost-based objective function may account for a similar weight and approach the worst competitive ratio in Theorem 5.2 in Chapter

5. When there are significant differences between the release period and the unit delivery cost, the competitive ratio will be much better ($\lambda = 3, 10, 50$). In addition, when the release period approximates to the unit delivery cost but the preparation times is long ($\lambda = 20$, $b = 30$), the results will come near to those off-line optimal schedules. It is clear that both the two conditions (i.e., particular preparation time and requirement release period) are necessary for the worst case.

Table 8.2: Results of Algorithm $H2$ for Problem P2 with $C = 5$

Parameters	$Z(\eta)$	$Z(opt)$	Competitive Ratio	Run-time
$\lambda = 3$, $b = 8$, $n = 100$	810.38	799.80	1.0147	0.000121s
$\lambda = 3$, $b = 8$, $n = 1000$	7977.87	7968.00	1.0013	0.004034s
$\lambda = 10$, $b = 8$, $n = 100$	1717.80	1403.34	1.2286	0.000123s
$\lambda = 10$, $b = 8$, $n = 1000$	16973.00	13977.36	1.2139	0.004099s
$\lambda = 20$, $b = 8$, $n = 100$	3548.58	2403.15	1.4771	0.000124s
$\lambda = 20$, $b = 8$, $n = 1000$	35268.61	23973.75	1.4736	0.004039s
$\lambda = 50$, $b = 8$, $n = 100$	7012.25	5401.65	1.2989	0.000126s
$\lambda = 50$, $b = 8$, $n = 1000$	69976.00	53986.18	1.2961	0.004103s
$\lambda = 20$, $b = 30$, $n = 100$	2730.44	2417.48	1.1305	0.000126s
$\lambda = 20$, $b = 30$, $n = 1000$	26843.49	24048.93	1.1154	0.004012s

The run-time for $n = 1000$ never exceeds 0.005 seconds and the run-time for $n = 100$ is much shorter, so the efficiency of $H2$ is high.

Table 8.3: Results of Algorithm $H2$ for Problem P2 with $C = 12$

Parameters	$Z(\eta)$	$Z(opt)$	Competitive Ratio	Run-time
$\lambda = 3$, $b = 8$, $n = 100$	597.14	587.55	1.0163	0.000128s
$\lambda = 3$, $b = 8$, $n = 1000$	5682.74	5672.69	1.0018	0.004035s
$\lambda = 10$, $b = 8$, $n = 100$	1626.08	1180.69	1.3772	0.000131s
$\lambda = 10$, $b = 8$, $n = 1000$	16062.67	11674.61	1.3759	0.004053s
$\lambda = 20$, $b = 8$, $n = 100$	3556.84	2184.58	1.6281	0.000130s
$\lambda = 20$, $b = 8$, $n = 1000$	35339.65	21679.14	1.6301	0.004072s
$\lambda = 50$, $b = 8$, $n = 100$	7016.41	5183.73	1.3535	0.000130s
$\lambda = 50$, $b = 8$, $n = 1000$	70043.07	51666.43	1.3557	0.004063s
$\lambda = 20$, $b = 30$, $n = 100$	2565.48	2195.61	1.1682	0.000140s
$\lambda = 20$, $b = 30$, $n = 1000$	25216.13	21696.36	1.1622	0.004022s

The results of Table 8.2 and Table 8.3 support the conclusions drawn from Table 8.1. However, it should be noted that from the three tables the competitive ratio increases as the capacity of vehicle C increases. This may imply that $H2$ will deliver more unfull batches at delivery points as C increases, while the off-line optimal schedule delivers fewer batches instead. The result about C may also imply that a better performance $H2$ can be designed in terms of C value, which agrees with the discussion in Chapter 5.

8.3 Concluding Remarks

In this chapter, we presented a simulate experiment for algorithm $H2$ for different instances. The following conclusions may be drawn from the discussion:

(1) The worst case happens when two conditions are satisfied: (a) the release period approximates to the unit delivery cost, and (b) the preparation time of requirements are short (which may rarely appear in reality).

(2) For the worst case, the proof of Theorem 5.2 in Chapter 5 has given an upper bound, for the competitive ratio of $H2$ which is close to the lower bound. Therefore, the algorithm $H2$ is also robust for worst case.

(3) There ia an assurance that the four algorithms designed in this thesis may perform well for normal instances, and have robustness for the worst case owing to their similar structures of problems and algorithms.

CHAPTER 9
CONCLUSIONS

9.1 Overview

Allocation of medical resources in EM is an important task in EM. In reality, a safe place may not be a hospital; therefore medical resources need to be delivered to the safe place to treat victims. Allocation of medical resources is similar to allocation of jobs in manufacturing systems such as supply chain systems. There have been many studies on allocation of jobs in supply chain systems or scheduling in supply chain systems as opposed to little work in EM. This thesis was therefore developed in the context of scheduling in supply chain systems. Through a comprehensive analysis of literature in supply chain systems, four new on-line problems were defined. Algorithms were developed for these problems, including analysis of algorithms based on the competitive ratio.

An entire process of allocating medical resources involves three activities: data collection from affected areas, preparation of medical resources in the medical centers, and delivery of medical resources in a network transportation system. This thesis only addressed the latter two activities but considered that future information is not available while scheduling. Such a scheduling problem is called on-line problem. Further, the two activities (preparation and delivery) were jointly considered in the sense the total cost and time of both the resource preparation and delivery are optimized. The methodology the thesis took to tackle the problems is: (1) taking the competitive ratio as a measure of the algorithms; (2) constructing instances for determining the lower bound of algorithms for the problems; (3) analyzing the gap between the competitive ratio and the lower bound to give the implication of the run-time performance of the algorithms.

The following conclusions can be drawn from the thesis:

(1) All the four algorithms have polynomial run-time functions. Table 9.1 gives a summary of the performances of the algorithms.

(2) Capacity limit for vehicles is a critical characteristics, which is responsible for the gap between the competitive ratio and lower bound.

(3) Run-times of the algorithms for normal situations are practically adequate. For P2, the algorithm $H2$ takes 0.005s for the number of requirements $n = 1000$.

Table 9.1: Analytical Results of The Problems

	P1	P2	P2 ($C = 2$)	P3	P4
Lower Bound	2	$2 - \frac{1}{C}$	$\frac{\sqrt{5}+1}{2}$	$max\{\frac{\sqrt{5}+1}{2}, 1 + \sqrt{\frac{D}{T+D}}\}$	$max\{\frac{\sqrt{5}+1}{2}, 1 + \sqrt{\frac{D}{T+D}} - \frac{\sqrt{D(T+D)}}{(C-1)\sqrt{D(T+D)}+T+D}\}$
Upper Bound	2	2	$\frac{\sqrt{5}+1}{2}$	$max\{\frac{\sqrt{5}+1}{2}, 1 + \sqrt{\frac{D}{T+D}}\}$	$max\{\frac{\sqrt{5}+1}{2}, 1 + \sqrt{\frac{D}{T+D}}\}$
Evaluation	optimal	gap	optimal	optimal	gap
run-time	$O(n)$	$O(n)$	$O(n)$	$O(n)$	$O(n^2)$

9.2 Contribution

This thesis has made contributions in the field of resources scheduling in an environment where demands on resources are unknown at the time decisions have to be taken. In particular, four new problems are defined and algorithms for them are developed. These algorithms are robust and efficient in terms of the competitive ratio measure and the lower bound. In particular, for P1 and P3, the algorithms ($H1$ and $H3$) can achieve the optimal result in terms of the competitive ratio and lower bound. For P2, its special case can achieve the optimal result. For P4, it has been found the larger the C parameter is, the better the result is.

Evacuation management in EM is an important operation management problem. The state of arts of solving this problem is such that evacuation of victims from dangerous places to safe places has been studied in literature without consideration of medical resources and transportation tools. This thesis has advanced the state of arts of EM by addressing the problem of allocation medical resources. The scope of the problems studied is in scheduling decisions for two operations: preparation and delivery of medical resources. Although these decisions are a part of whole decisions in EM, they are practically useful to facilitate that part of decisions more accurately and quickly so that human decision resource can be more allocated on those parts where computer models have yet to be developed.

9.3 Future Work

First, study on optimal algorithms for P4 with some special case is interesting. This is possible as the optimal algorithm for P2 with $C = 2$ has been successfully developed.

Second, the four problems all assumed a single-processor in medical centers and there was only one aid site where medical resources are sent to. Further work on the cases where there are multiple processors and many aid sites is worthwhile.

Third, the assumptions underlying the present study may be relaxed to be more in line with real situations. For example, there should be more than one medical center to supply medical resources.

LIST OF REFERENCES

Albers, S., Online Algorithms: A Survey, *Mathematical Programming*, vol. 97, no. 1, pp. 3-26, 2003.

Anderson, E. J., Nash P., and Philpott, A. B., A Class of Continuous Network Flow Problems, *Mathematics of Operation Research*, vol. 7, pp. 501-514, 1982.

Averbakh, I. and Xue, Z.-H., On-line Supply Chain Scheduling Problems with Preempition, *EJOR*, vol. 181, pp. 500-504, 2007.

Averbakh, I., On-line Integrated Production-Distribution Scheduling Problems with Capacitated Deliveries, *EJOR*, 2010, vol. 200 pp. 377-384.

Benjaafar, S., Dooley, K., and Setyawan, W., Cellular Automata for Traffic Flow Modeling, *University of Minnesota, Mineapolis*, 1997.

Burkard, R. E., Dlaska, K., and Klinz, B., The Quikest Flow Problem, *ZOR-Methods and Models of Operations Research*, vol. 37, pp. 31-58, 1993.

Chalmet, L. G., Francis, R. L., and Saunders, P. B., Network Models for Building Evacuation, *Management Science*, vol. 28, pp. 86-105, 1982.

Chen, Y. L. and Chin, Y. H., The Quickest Path Problem, *Computers and Operations Research*, vol. 17, pp. 153-161, 1990.

Chen, G. H. and Hung, Y. C., On the Quickest Path Problem, *Information Processing Letters*, vol. 46, pp. 125-128, 1993.

Chen, H. K. and Hsueh, C. F., A Model and An Algorithm for The Dynamic User-Optimal Route Choice Problem, *Transportation Research B*, vol. 32, no. 3, pp. 219-234, 1998.

Chen, B., Lee and C.-Y., Logistics Scheduling with Batching and Transportation, *Eur. J. Oper. Res.* vol. 189, pp. 871-876, 2008.

Chen, Z.-L., Vairaktarakis, G. L., Integrated Scheduling of Production and Distribution Operations, *Management Science*, vol. 51 pp. 614-628, 2005.

Chen, Z.-L., Integrated Production and Outbound Distribution Scheduling: Review and Extensions, *Operations Research*, vol.58, pp. 130-148, 2010.

Cheng, T. C. E. and Kahlbacher, H. G., Scheduling with delivery and earliness penalties, *Asia-Pacific J. Oper. Res.*, vol. 10, pp. 145-152, 1993.

Choi, W., Francis, R.L., Hamacher, H.W., Tufekci, S., Network Models of Building Evacuation Problems With Flow-Dependent Exit Capacities, *Operational Research*, pp. 1047-1059, 1984.

Choi, W., Francis, R. L., Hamacher, H. W., Tufekci, S., Modelling of Building Evacuation Problems with Side Constraints, *European Journal of Operation Research*, vol. 35, pp. 98-110, 1988.

Christofides, N., Worst-case Analysis of A New Heuristic for The Traveling Salesman Problem, *Report 388, Grad School of Industrial Administration, CMU*, 1976.

Fahy, R. F., An Evacuation Model for High Rise Buildings, *Proceedings of the Third International Symposium on Fire Safety Science*, Elsevier, London, pp. 815-823, 1991.

Fleischer, L., Effient Continuous-Time Dynamic Network Flow Algorithms, *Operation Research Letters*, vol. 23, pp. 71-80, 1998.

Fleischer, L., Faster Algorithms for the Quickest Transshipment Problem, *SIAM Journal on Optimization*, vol. 12, pp. 18-35, 2001.

Fleischer, L., Universally Maximnin Flows with Piecewise-Constant Capacities. *Networks*, vol. 38, pp, 115-125, 2001.

Garey, M. R. and Johnson, D. S., Computers and Intractability: A Guide to the Theory of NP-Completeness, *New Yrok: W.H. Freeman*, 1979.

Gasarch, I. W., The P=?NP Poll, *SIGACT News*, vol. 33, no. 2, pp. 34-47, 2002.

Gharbi, A., Haouari, M., Minimizing makespan on parallel machines subject to release dates and delivery times, *J. Scheduling*, vol. 5, pp. 329-355, 2002.

Hall, N., Potts, C., The Coordination of Scheduling and Batch Deliveies, *Annals of Operations Research*, vol. 135, pp. 41-64, 2005.

Hall, L. A., Shmoys,D.B., Jackson?s Rule for Single-Machine Scheduling: Making A Good Heuristic Better, *Math. Oper. Res.*, vol. 17, pp. 22-35, 1992.

Hamacher, H. W. and Tufekci, S., On the Use of Lexicographic Min Cost Flows in Evacuation Modeling, *Naval Research Logistics*, vol. 34, pp. 487-503, 1987.

Hamacher, H. W., Tjandra, S. A., Mathematical Modelling of Evacuation Problems: A State of the Art, In: *Pedestriori and Eurrcuation Dynamics* (M. Schreckenberg and S. D. Sharma, eds., Springer, 2002), pp. 227-266, 2002.

Hoppe, B. and Tardos, E., Polinomial Time Algorithms for Some Evacuation Problems, *Proc. of 5th Ann. ACM-SIAM Symp. on Discrete Algorithms*, pp. 433-441, 1994.

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, pp. 745-755, 2007.

Kaminsky, P., The Effectiveness of The Longest Delivery Time Rule for The Flow Shop Delivery Time Problem, *Naval Res. Logist.*, vol. 50, pp. 257-272, 2003.

Karp, R. M., Reducibility Among Combinational Problems, *Complexity of Computer Computations*, pp. 85-103, 1972.

Klüpfel, H., König, T.M., Wahle, J., Schreckenberg, M., Microscopic Simulation of E-vacuation Processes on Passenger Ships, *Fourth International Conference on Cellular Automata for Research and Industry*, October, Karlsruhe, Germany, 2000.

Knuth, D. E., Big Omicron and Big Omega and Big Theta, *ACM SIGACT News*, vol. 8, no. 2, pp. 18-24, 1976.

Kostreva, M. M., and Wiecek, M. M., Time Dependency In Multiple Objective Dynamic Programming, *Journal of mathematical Analysis and Application*, vol. 173, no. 1, pp. 289-307, 1993.

Lu, Lingfa, Yuan, Jinjiang, Zhang, Liqi, Single Machine Scheduling with Release Dates and Job Delivery to Minimize The Makespan. *Theoretical Computer Science*, vol. 393, pp. 102-108, 2008.

Mastrolilli, M., Efficient Approximation Schemes for Scheduling Problems with Release Dates and Delivery Times, *J. Scheduling*, vol. 6, pp. 521-531, 2003.

Minieka, E., Maximal, Lexicographic, and Dynamic Network Flows, *Operations Research*, vol. 21, pp. 517-527, 1973.

Nagel, K. and Schreckenberg, M., A Cellular Automaton Model for Freeway Traffic, *J. Phys. I France*, vol. 2, pp. 2221-2229, 1992.

Ogier, R. G., Minimutn-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.

Philpott, A. B., Continuous-Time Flows in Networks, *Mathematics of Operation Research*, vol. 15, no. 4, pp. 640-661, 1990.

Rogers, H., Theory of Recursive Functions and Effective Computability, *New York: McGraw-Hill*, 1967.

Rosenberger, J., P vs. NP Poll Results, *Communications of the ACM*, vol. 55, no. 5, pp. 10, 2012.

Woeginger, G. J., Heuristics for parallel machine scheduling with delivery times, *Acta Information*, vol. 31, pp. 503-512, 1994.

Yamada, T., A Network Approach To A City Emergency Evacuation Planning, *International Journal of Systems Science*, vol. 27, no. 10, pp. 931-936, 1996.

APPENDIX
PROGRAM FOR $H2$

The program in Appendix is to an implementation of the algorithm $H2$. It consists of three functions (running on MATLAB 2008 and later version): **Alg_P2**, **Req_data**, and **Test**. The main function is **Alg_P2**(Req, D, C), in which Req is the information of requirements, D is the unit delivery cost, and C is the capacity of vehicle. Function **Req_data**(lambda, b, n) is used to generate requirements where lambda (λ) is the expectation of poisson distribution, b is the upper bound of uniform distribution, and n is the number of requirements. A brief description for each function is introduced at the beginning of each function module.

```
% Program for algorithm H2
% MATLAB M-file
% August 2012
% Main function Alg_P2(Req, D, C)
% – Req is the information of requirements
% – D is the unit delivery cost
% – C is the capacity of vehicle

function [Z_alg,Z_opt,ratio]=Alg_P2(Req,D,C)
% Input — Req: the requirement data (release time and preparing time)
%         — D: the unit delivery cost
%         — C: the capacity of vehicle
% Output — Z_alg: the value of algorithm H2
%           — Z_opt: the off-line optimal value
%           — ratio: competitive ratio
Req_no=size(Req,1);
```

```
n=size(Req,1);
C_max=0;
D_max=0;
TC=0;
Next_Delivery=D;
Complete_no=0;
while(Req_no~=0)
    if C_max < Req(1,1)
        if Next_Delivery < Req(1,1)
            TC=TC+ceil(Complete_no/C)*D;
            Complete_no=0;
            D_max=Next_Delivery;
            Next_Delivery=ceil(Req(1,1)/D)*D;
        end
        C_max=Req(1,1);
    end
    while C_max + Req(1,2) > Next_Delivery
        TC=TC+floor(Complete_no/C)*D;
        Complete_no=Complete_no-floor(Complete_no/C)*C;
        D_max=Next_Delivery;
        Next_Delivery=Next_Delivery+D;
    end
    C_max=C_max + Req(1,2);
    Complete_no=Complete_no+1;
    Req=Req(2:Req_no,:);
    Req_no=Req_no-1;
    if Req_no==0
      TC=TC+ceil(Complete_no/C)*D;
      D_max=Next_Delivery;
    end
end
Z_alg=D_max+TC;
Z_opt=C_max+ceil(n/C)*D;
```

```
ratio=Z_alg/Z_opt;
```

function Req=Req_data(lambda,b,n)

```
% Input — lambda: the expectation of poisson distribution
%       — b: the upper bound of uniform distribution
%       — n: the number of requirements
% Output — Req: the release time and preparing time of requirements
Pos=poissrnd(lambda,n,1);
Release=zeros(n,1);
Release(1)=Pos(1);
for i=2:n
    Release(i)=Release(i-1)+Pos(i);
end
Prepare=b*rand(n,1);
Req=[Release,Prepare];
```

function Summary=Test(lambda,b,n,C)

```
% Input — the parameters of the above two functions
% Output — Summary: results of 100 running
Summary=[];
for i=1:100
    Req=Req_data(lambda,b,n);
    D=normrnd(20,1);
    if D<=0
        D=1;
    end
    tic;
    [Z_alg,Z_opt,ratio]=Alg_P2(Req,D,C);
    t=toc; % time counting
    Summary=[Summary;Z_alg,Z_opt,ratio,t];
end
```