XIANDONG ZHANG

Scheduling with Time Lags

Scheduling with Time Lags

Scheduling with Time Lags

Machinevolgordeproblemen met vertragingen

Proefschrift

ter verkrijging van de graad van doctor
aan de Erasmus Universiteit Rotterdam
op gezag van de rector magnificus
Prof.dr. H.G. Schmidt
en volgens besluit van het College voor Promoties.

De openbare verdediging zal plaatsvinden op
donderdag 1 juli 2010 om 11.30 uur

door
Xiandong Zhang
Geboren te Xuzhou, Jiangsu, China

ERASMUS UNIVERSITEIT ROTTERDAM

Promotor:
Prof.dr. S.L. van de Velde

Overige leden:
Prof.dr. M.B.M. de Koster
Prof.dr. A.P.M. Wagelmans
Prof.dr. B. Chen

Erasmus Research Institute of Management – ERIM
Rotterdam School of Management (RSM)
Erasmus School of Economics (ESE)
Erasmus University Rotterdam
Internet: http://www.erim.eur.nl

ERIM Electronic Series Portal: http://hdl.handle.net/1765/1

ERIM PhD Series in Research in Management, 206
Reference number ERIM: EPS-2010-206-LIS
ISBN 978-90-5892-244-1

Design: B&T Ontwerp en advies www.b-en-t.nl
Print: Haveka www.haveka.nl

Scheduling with Time Lags

For *Yuhong* & *Jiayuan*

Preface

It was July 1999 when I first came to Erasmus University Rotterdam. At that time, Professor Willem Lammerts van Bueren, the founder and honorary chairman of the China-Holland Education & Research Center (CHERC), organized a master class in Rotterdam for potential Chinese Ph.D. candidates. I was kindly invited to join this class. And, from then on, I decided to pursue a Ph.D. degree from Erasmus University. In the years that followed, I have been continuously encouraged and supported by Willem and Professor Qifan Wang to achieve this goal; I am very grateful for their encouragement and support.

This thesis is based on work that began in January 2007, when I came to Erasmus University for the second time. My promotor, Professor Steef van de Velde, led me to the research field of scheduling with time lags. At the time, we called it the barges and container terminals problem, as its motivation lied in port operations. The topic fitted me very well. Some initial results were found almost immediately. Steef gave me a lot of suggestions to improve these initial and ensuing results and my English writing style. After nearly four years of research in this area, I think I have really become a scheduling guy. I would like to thank Steef for cultivating my skills and taste for academic research.

I express my thanks to the other members of my Ph.D. committee for reading the draft version and providing me with useful feedback: Professor René M.B.M. de Koster, Professor Albert P.M. Wagelmans and Professor Bo Chen.

Since I had to visit the Rotterdam School of Management (RSM), Erasmus University twice a year to continue my study, I express my gratitude towards

many people who have contributed a lot to making this happen. Firstly, I thank Weichia Tseng, the managing director of CHERC, who sent me invitation letters twice a year to invite me to come here and helped me to cope with the administrative issues of my Ph.D. Secondly, I thank Xiongwen Lu, the dean of the School of Management, Fudan University (FDSM); Yihong Yu, the vice dean; Qiuzhi Xue, the vice dean; and Yifan Xu, the chairman of the Department of Management Science at FDSM, for their continuous support for my visits to the Netherlands. Thirdly, I thank Huang Lei, the senior officer for foreign affairs at FDSM, who kindly helped me with my visa applications every time. Then, I thank Mengfei Yu, my best friend in Rotterdam, who generously provided free accommodation for my short stays and has much better cooking skills than I. Next, I would like to thank Carmen Meesters-Mirasol, the secretary of Department 6, who kindly arranged office room for me every time and helped with the many administrative issues at the department.

I thank my colleagues in the Department of Management of Technology and Innovation at RSM and in the Department of Management Science, Fudan University, for the great working atmosphere on both sides. I really enjoy the lunch time we spend together in the L-Building and in the Starr Building. I enjoyed our many fruitful discussions very much.

A special word of thanks goes the Chinese group in the Department of Management of Technology and Innovation. I learned a lot from the discussions with Yugang Yu, whose knowledge on integer programming deepened my understandings of this topic. Yeming Gong generously provided the LaTex template for this thesis and shared his many experiences in doing research with me. With Mengfei Yu, who used AutoMod in his research, I had the pleasure of discussing common topics in simulation.

Also, I would like to thank Professor Meine Pieter van Dijk, who acted as my supervisor at Erasmus University in the very beginning. After his move to the UNESCO-IHE Institute, we almost lost contact with each other. I think he will be gratified when he sees this thesis.

I thank both Erasmus University and FDSM for supporting my study financially. The research was also supported in part by projects of the National Science Foundation of China (No. 70832002, No. 10971034 and No. 70771028).

Finally, I owe a world of thanks to my family members. My parents, my wife, *Yuhong*, and my adorable daughter, *Jiayuan*, had to bear family separations twice a year, especially in the Spring Festivals holidays season in the past four years. I appreciate their continuous support from the deepest of my heart.

Xiandong Zhang
Rotterdam, the Netherlands
2010

Contents

Part I Introduction and Review

1 Introduction 3
- 1.1 Machine scheduling 3
- 1.2 Computational complexity 6
- 1.3 Performance measurements 8
- 1.4 Time lags 10
  - 1.4.1 Classification of time lags 11
  - 1.4.2 Complexity results 12
  - 1.4.3 Approximation algorithms 12
- 1.5 Some practical problems 13
- 1.6 Outline of the thesis 17

Part II On-Line Algorithms

2 Two-Machine Job Shop 21
- 2.1 Introduction 21
- 2.2 Performance analysis of a greedy algorithm 24
- 2.3 The worst-case performance of on-line delay algorithms 27
- 2.4 Conclusions 29

3 Two-Machine Open Shop 31
- 3.1 Introduction 31

3.2 Notations and preliminaries 34
3.3 Performance analysis of the greedy algorithm 35
3.3.1 On-line case 35
3.3.2 Semi-online case 36
3.4 The worst-case performance of on-line delay algorithms 43
3.5 Conclusions 45

Part III Off-Line Algorithms

4 Parallel Flow Shop 49
4.1 Introduction 49
4.2 A $\frac{3}{2}$-approximation algorithm for $m = 2$ 52
4.3 A $1\frac{5}{7}$-approximation algorithm for $m = 3$ 64
4.4 Conclusions 70

5 Polynomial-Time Approximation Schemes 75
5.1 Introduction 75
5.1.1 Problem description 75
5.1.2 Complexity 77
5.1.3 Approximability 78
5.2 Our contribution 79
5.3 PTAS for makespan problems in class $\mathcal{M}$ 80
5.4 PTAS for problems in class $\mathcal{F}$ 84
5.5 Conclusions 87

6 Fixed Interval Shop Scheduling 89
6.1 Introduction 89
6.2 Fixed interval two-machine flow shop scheduling 92
6.2.1 The case of equal time lags 93
6.3 Fixed interval two-machine job shop scheduling 95
6.4 Fixed interval two-machine open shop problem 99
6.5 Conclusions 101

Part IV Conclusions and Future Directions

7 Conclusions and Future Directions 109
7.1 Conclusions 109
7.2 Directions for future works 111

References 113

Summary 119

Summary in Dutch 121

List of Tables

1.1 Complexity results for flow shop problems with minimal time lags . 12
1.2 Complexity results for open shop problems with minimal time lags 13
1.3 Minimum and maximum storage times in hours 15
1.4 Minimum and maximum degassing times in hours 16

5.1 Some problems in class $\mathcal{M}$ 80
5.2 Two earlier studied problems in class $\mathcal{F}$ 81

List of Figures

1.1 Layout of the Port of Rotterdam. (Adapted from Douma (2008)) . . 14
1.2 Material flow in coffee production. (Adapted from Simeonov and Simeonovová (1997)) 16

2.1 The illustration of Δ_k, Q, T and busy periods. 26

4.1 A hybrid two-stage flow shop. 51
4.2 A parallel two-stage flow shop. 51
4.3 Cutting the Johnsonian schedule as prescribed in Lemma 7. 54
4.4 Cutting the Johnsonian schedule as prescribed in Lemma 8. 55
4.5 Cutting the delayed Johnsonian schedule as prescribed in Lemma 9 if $S'_{1h} \geq T_1$. The top schedule is the Johnsonian schedule σ, the bottom schedule is the delayed Johnsonian schedule σ'. 56
4.6 Cutting the delayed Johnsonian schedule as prescribed in Lemma 9 if $S'_{1h} < T_1$. The top schedule is the Johnsonian schedule σ, the bottom schedule is delayed Johnsonian schedule σ'. 56
4.7 Illustration of a Johnsonian schedule σ (the top schedule) and a delayed Johnsonian schedule σ' (the bottom schedule) for a job J_h with $S_{2h} \leq T_2 \leq C_{2h}$, $S_{1h} < T_1$, $C_{1h} < S_{2h}$, $S'_{1h} < T_1$ and $C'_{1h} < S'_{2h}$. 57
4.8 Illustration of the jobs J_u,J_v,J_d,J_e, with $J_u = J_d = J_h$, as they occur in Lemma 11. 58
4.9 Cutting the Johnsonian schedule as prescribed in Lemma 12. 59

4.10 Cutting the Johnsonian schedule as prescribed in Lemma 13. 60
4.11 Cutting the Johnsonian schedule as prescribed in Lemma 14. ($v < e$) 61
4.12 Splitting of the Johnsonian schedule according to Lemma 14. ($v \geq e$) 61
4.13 Cutting the Johnsonian schedule as prescribed in Lemma 15. 62
4.14 Cutting the Johnsonian schedule as indicated in Lemma 16. 63
4.15 Cutting the Johnsonian schedule as prescribed in Lemma 18. 65
4.16 Cutting the Johnsonian schedule as indicated in Lemma 19. 66
4.17 Cutting the Johnsonian schedule as indicated in Lemma 20. 66
4.18 The remaining case with jobs J_h and J_t. 67
4.19 Cutting the Johnsonian schedule as indicated in Lemma 21. 68
4.20 Cutting the Johnsonian schedule as indicated in Lemma 22. 69

6.1 J_h and J_i are crossing jobs. The arrows represent the equal time lags $l > 0$. 95
6.2 Crossing jobs for the case $l = 0$. 96
6.3 An instance of fixed interval two-machine flow shop scheduling problem with time lags. 104

Part I

Introduction and Review

1

Introduction

All things under heaven sprang from it as existing, the existence sprang from it as non-existent.

- Lao Tzu (around 500BC)

Scheduling theory has been developed along with the evolution of modern manufacturing, aimed to solve problems of allocating scarce resources to activities over time. The ubiquity of resource scarcity makes scheduling theory highly relevant for a wide variety of industrial settings. The resources and activities are commonly referred as *machines* and *jobs*, respectively, where a job may consist of multiple operations. This thesis is concerned with the theory of scheduling with time lags, where a time lag specifies a minimum delay between the execution of two consecutive operations of the same job. Such a time lag may for instance arise if the transfer of a job from one machine to the next requires a certain amount of transportation time.

1.1 Machine scheduling

The distinguishing features of machine scheduling problems are captured by a well-known three-field framework, which describes the machine environment, the job characteristics, and the scheduling objective (Graham et al (1979)). In the remainder of this section we give a short overview of the framework. For more elaborate descriptions we refer to Hoogeveen et al (1997) and Pinedo (1995).

Machine environment The machine environment specifies the type(s) of machine(s) available for processing jobs. A machine may be continuously available or have break downs. It is commonly assumed that each machine can perform at most one operation at any time and each operation requires at most one machine at any time. The simplest machine environment is the *single machine*, on which n jobs J_j $(j = 1, \ldots, n)$, each consisting of a single operation, have to be processed for a processing time p_j. In a *parallel machine* environment, there are m machines, all jobs consist of a single operation and each job has to be assigned to exactly one of the machines. These m machines may be identical or not. In flow shops, job shops and open shops, there are also m machines but job J_j $(j = 1, \ldots, n)$ consists of a set of m operations, each of which requires processing on a specified machine and no job can undergo more than one operation at a time. In a *flow shop*, the order for processing the operations of job J_j $(j = 1, \ldots, n)$ is fixed and the same for all jobs; in a *job shop*, the order for processing the operations of job J_j $(j = 1, \ldots, n)$ is fixed and *not* the same for all jobs; in a *open shop*, the order for processing the operations of job J_j $(j = 1, \ldots, n)$ is free and hence up to the scheduler. A flow shop is called a *permutation* flow shop if the order in which the jobs go through the first machine need to be maintained throughout the entire system. A permutation schedule is defined analogously. A *flexible flow shop* is a flow shop with a number of parallel machines in each stage. *Flexible job shops* and *flexible open shops* are defined analogously.

Job characteristics A job J_j may have a release date r_j, by which job J_j is available for processing, a due date d_j, by which job J_j is expected to be finished, and a deadline $\bar{d}_j$, by which job J_j has either to be finished or discarded. If a job completes after its due date, it is called late or tardy. The lateness of a job is measured by the difference between its completion time and its due date. Furthermore, each job J_j may have a weight w_j, by which job J_j is weighted in an objective function. For single machine problems, p_j denotes the required processing time of job J_j, while for problems with m machines, p_{ij} is the required processing time of job J_j on machine M_i $(i = 1, \ldots, m)$.

If the scheduler has all relevant information for scheduling the n jobs available before scheduling, the scheduling problem is called an *off-line* problem. If the processing of a job can be interrupted at any point in time, the problem is a preemptive scheduling problem. If the scheduler has no access to all information of a problem instance and has to react to new job scheduling requests with only a partial knowledge of the problem instance, the scheduling problem is called an *on-line* problem.

Sgall (1998) distinguishes the following on-line paradigms.

(i) *Jobs arriving in a list.* Jobs are presented to the scheduler one by one. As soon as a job is revealed, the scheduler knows all its characteristics, such as its processing time, deadline, weight, etc. The scheduler has no information of the next job and the total number of jobs. The scheduler has to assign the job to a time interval and a machine while satisfying all constraints. Once the scheduler sees the next job he cannot change the assignment of the previous jobs.

(ii) *Jobs arriving over time.* Each job becomes available at its release time. Before the release time of a job, the scheduler has no information about it. After the job is revealed, the scheduler may start the job immediately or delay it. If the on-line problem is of the *clairvoyant* type, the processing time of a job becomes known at its arrival. If the on-line problem is of the *non-clairvoyant* type, then the processing time of each job is unknown until it finishes.

If no information of the jobs is available but additional knowledge of about the structure of the problem is known, the scheduling problem is called a *semi-online* problem. For example, Seiden et al (2000) investigated the semi-online problem of scheduling jobs on m identical parallel machines where the jobs are known to arrive in order of decreasing processing times. In this case, jobs arrive in a list but the scheduler knows in advance that the processing time of incoming jobs are becoming smaller and smaller.

Objective functions A *schedule* is a specification of the job start and completion times such that all machine and job requirements are met. The quality of a schedule can be measured in many different ways. One is the *makespan* or length of the schedule, denoted by $C_{\max}$. If we let C_j denote the time when job J_j finishes its processing, then the makespan is defined as $\max\{C_1, \ldots, C_n\}$, that is, the completion time of the last completed job. A small makespan usually implies a high utilization of the machine(s). If the scheduler cares about the total inventory costs incurred by the schedule, the *total weighted completion time*, $\sum w_j C_j$, may work as an objective function, where w_j is the weight of job J_j.

The scheduling objective may also be a function of the due dates. For example, the *lateness* of job J_j is defined as $L_j = C_j - d_j$, and the maximum lateness $L_{\max}$, which is defined as $\max\{L_1, \ldots, L_n\}$, is a measure of the largest violation of the due dates. The *tardiness* of job J_j is defined as $T_j = \max\{C_j - d_j, 0\} = \max\{L_j, 0\}$.
The *weighted number of tardy jobs*, $\sum w_j U_j$, is a measure for the weighted number of jobs that are finished late, where U_j is a variable that takes the value 1 if job J_j is finished after its due date and the value 0 otherwise. For other objective functions, please refer to Pinedo (1995).

Given the machine environment, job characteristics, and scheduling objectives, a scheduling problem can be denoted by the three-field notation $\alpha|\beta|\gamma$, introduced by Graham et al (1979). The first field α specifies the machine environment, the second field β describes the job characteristics, and the third field γ refers to the optimality criterion. For example, $F2||C_{\max}$ is the well-known two-machine flow shop scheduling problem to minimize the makespan. $P2||C_{\max}$ is the parallel two-machine problem to minimize the makespan, where the two machines are identical. $O2|\mathit{on\text{-}line}, r_j, l_j|C_{\max}$ is the on-line two-machine open shop scheduling problem to minimize the makespan, where jobs have release times and there is a time lag l_j between the two operations of job J_j, for $j = 1, \ldots, n$.

1.2 Computational complexity

The main goal of scheduling research is to find efficient algorithms for scheduling problems. For any off-line non-preemptive deterministic scheduling problem, the total number of feasible sequences is finite. Although possibly very large, an optimal sequence can in principle always be found by exhaustive enumeration, by examining all possible sequences. This is not a practical approach when the search space is large. For many combinatorial optimization problems, enumerating all feasible solutions may take thousands of years even with the most powerful computer in the world.

An algorithm is an $O(g(n))$ algorithm if there exists a constant $c > 0$, a function $g(n)$, and an integer $n_0 > 0$ such that the maximum number of iterations $f(n)$ needed to find an optimal solution satisfies $f(n) \leq cg(n)$ for all $n \geq n_0$, where n is the *size* of a problem instance (the number of bits required to represent the instance by a computer as an input to the algorithm). If g is a polynomial function, the algorithm is called a *polynomial-time algorithm*. According to Garey and Johnson (1979), a problem has not been "well-solved" until a polynomial-time algorithm is known for it, since polynomial-time algorithms typically require much less computing time than non-polynomial time algorithms.

Encoding scheme The size of a problem instance is related to the encoding scheme used. For example, consider an instance of $P2||C_{\max}$ with 5 jobs ($n = 5$) and processing times 2, 3, 5, 9, and 8. An *unary encoding* scheme uses k ones to represent the number k. The instance of the $P2||C_{\max}$ problem is then encoded as

$$11, 111, 11111, 111111111, 11111111.$$

Therefore, the size of the instance is $2 + 3 + 5 + 9 + 8 = 27$.

If all the data were presented in *binary encoding*, the above instance would be encoded as

$$10, 11, 101, 1001, 1000.$$

The size of the instance is then $2 + 2 + 3 + 4 + 4 = 15$. Hence, the value k has size k in unary encoding but has size $\lfloor log_2 k \rfloor + 1$ in binary encoding.

Suppose an algorithm is a polynomial-time algorithm where the time complexity is a function of the size of the problem instance in unary encoding, i.e., an $O(g(n))$ algorithm where n is the size in unary encoding. If the encoding scheme is changed to binary encoding, the problem size may decrease to $n' = \lfloor log_2 n \rfloor + 1$ with $n \geq 2^{n'-1}$. The algorithm then becomes an $O(g(2^{n'}))$ algorithm, which is not a polynomial-time algorithm any more in binary encoding.

Usually, an algorithm is said to be polynomial-time only if the encoding scheme is binary. An algorithm that runs in polynomial time with respect to an unary encoding scheme is called a *pseudo-polynomial algorithm*.

Classes P and NP In complexity theory, a *decision problem* is a problem that requires a "yes" or "no" answer. Every combinatorial optimization problem has a corresponding decision problem. For example, for the $F2||C_{\max}$ problem, the corresponding decision problem is: is there a schedule that completes within the given makespan C?

If for any *yes* instance of a decision problem, there exists a verifier algorithm that can check in polynomial time whether the instance is indeed a yes instance, the decision problem is said to belong to the *class NP*. For example, if a schedule for a problem instance of $F2||C_{\max}$ is claimed to have a makespan less than C, we can easily verify this claim in polynomial. Hence, $F2||C_{\max}$ belongs to class NP.

Class P is a subset of class NP. A decision problem is said to belong to *class P*, if it can be answered by a polynomial-time algorithm which can also be used to verify any *yes* instance of the problem. Problems in class P are so-called easy problems.

Class NP-complete is another subset of class NP. A decision problem A in NP is called NP-complete if all other problems in NP *polynomially reduce* to A. We say that a decision problem A polynomially reduces to another decision problem B if, given any instance x of A, we can construct an instance y of B within polynomial (in size of x) time such that x is a yes instance of A if and only if y is a yes instance of B (Papadimitriou and Steiglitz (1998)). Therefore, if any NP-complete problem can be solved in polynomial time, then all problems in class NP can be solved in polynomial time. Since no polynomial-time algorithm has ever been found for an NP-complete problem, problems in class NP-complete are called hard problems.

A decision problem A is called *NP-hard* if all other problems in NP *polynomially reduce* to A, and problem A may be in class NP or not. Therefore, class NP-complete is a subset of class NP-hard.

For a subset of problems in class NP-hard, pseudo-polynomial algorithms may exist. A decision problem is so called *weakly NP-hard/NP-complete* if it has a pseudo-polynomial algorithm. A *strongly NP-hard/NP-complete* problem cannot have a pseudo-polynomial algorithm unless P=NP.

An optimization problem is called strongly/weakly NP-hard/NP-complete if its corresponding decision problem is strongly/weakly NP-hard/NP-complete.

1.3 Performance measurements

Given a scheduling problem, we first need to identify whether the problem is in class P or class NP-hard, and if it is in class NP-hard, we need to further explore whether it is weakly or strongly NP-hard. Brucker and Knust (2010) have made a near complete list of the complexity results of the best-known scheduling problems. Scheduling problems are grouped by machine environments, such as single machine problems, parallel machine problems, etc., and then classified into five categories: maximal polynomially solvable, maximal pseudo-polynomially solvable, minimal NP-hard, minimal open and maximal open.

For NP-hard problems, optimization algorithms may work for small problem instances only. For large scale problem instances, the approach is typically aimed at finding a near-optimal solution, since the search for an optimal solution may take a prohibitively large amount of time. To evaluate the performance of an approximation algorithm, we use *worst case ratio* for off-line algorithms and *competitive ratio* for on-line algorithms.

Worst case ratio Given an off-line NP-hard scheduling problem A of minimizing an objective function f, we define $f^H(I)$ to be the objective value of the solution given by a polynomial-time algorithm H for an instance I, and $f^*(I)$ to be the value of an optimum schedule. If for all $I \in \mathcal{X}$, where $\mathcal{X}$ is the set of all instances of the problem, it holds that

$$f^H(I) \leq \rho f^*(I)$$

for a specific constant $\rho \geq 1$, then algorithm H is called a *ρ-approximation* algorithm for problem A. The worst case (performance) ratio of algorithm H is defined as

$$R^H = \sup_{I \in \mathcal{X}} (\frac{f^H(I)}{f^*(I)}).$$

By definition, we have $R^H \leq \rho$. If we have found an instance I' such that $\frac{f^H(I')}{f^*(I')} = \rho$, then the approximation ratio of algorithm H is said to be *tight* and $R^H = \rho$. The ratio is used to measure the quality of an algorithm and to compare different algorithms.

For any $\varepsilon > 0$, algorithm H is said to be an *approximation scheme* for problem A if

$$f^H(I) \leq (1 + \varepsilon) f^*(I).$$

H is said to be a *polynomial time approximation scheme*, abbreviated as PTAS, if for each fixed $\varepsilon > 0$, its running time is bounded by a polynomial in the size of the instance I. If the running time of H is bounded by a polynomial in the size of the instance I and $1/\varepsilon$, then H is a *fully polynomial time approximation scheme*, abbreviated as FPTAS. No strongly NP-hard optimization problem can have an FPTAS unless P=NP, if the value of the objective function is bounded by a polynomial in the unary size of instance I (Vazirani (2004)).

Competitive ratio The competitive ratio in an on-line scheduling environment is the equivalent of the worst case ratio in an off-line scheduling environment. For any on-line scheduling problem A of minimizing an objective function f, let $f^H(I)$ denote the objective value of the schedule give by some polynomial-time algorithm H for an instance I, and let $f^*(I)$ be the optimal objective value of the off-line version of the problem. If for all $I \in \mathcal{X}$, where $\mathcal{X}$ is the set of all instances of the problem

$$f^H(I) \leq cf^*(I),$$

for some constant $c \geq 1$, then algorithm H is called a *c-competitive* algorithm for problem A. The competitive ratio of algorithm H is defined as

$$R^H = \sup_{I \in \mathcal{X}} (\frac{f^H(I)}{f^*(I)}),$$

and we have $R^H \leq c$. If there exists an instance I' such that $\frac{f^H(I')}{f^*(I')} = c$, then the competitive ratio of algorithm H is *tight* and $R^H = c$.

1.4 Time lags

A *time lag* is the minimum time delay required between the execution of two consecutive operations of the same job. Mitten (1959) was the first to consider a scheduling problem with time lags; in his problem, jobs need to be processed by a number of non-bottleneck machines in between two bottleneck machines. The total time required for processing a job J_j ($j = 1, \ldots, n$) on the intermediate non-bottleneck machines may then be represented as a certain time lag l_j. For this problem, denoted by $F2|l_j|C_{\max}$, Mitten (1959) shows that an optimal permutation schedule can be found in polynomial time. Yu (1996) proved that the problem $F2|l_j, p_j = 1|C_{\max}$ is strongly NP-hard if non-permutation schedules are allowed.

Time lags can represent the following:

(i) *Transportation delays.* When the time needed to move a job from one machine to another is not negligible, we have to take transportation delays into account when constructing a schedule. The implicit assumption is that there is a sufficient number of vehicles to carry out the transportation, or, that jobs may travel by themselves, like trucks and barges loading and unloading at terminals and traveling between them.

(ii) *Activities that require no limited resources.* In many industries, activities require no limited resources other than time. Examples include the fermentation process in the food industry, the cooling down process in metal casting, and the sun-drying process in the agricultural industry.

(iii) *Intermediate processes.* In a manufacturing/service system, processes between two bottleneck machines may often be modeled as activities that require no limited resources.

1.4.1 Classification of time lags

When we talk about time lags, we usually refer to *minimal* time lags, that is, the minimal amount of time required to elapse between two consecutive operations of a job. In a feasible sequence, the actual time between two consecutive operations may be larger than the minimal time lag. A *maximal time lag* specifies an upper bound on the time delay between two consecutive operations of a job.

A scheduling problem may include both *minimal and maximal time lags*. The classical scheduling setting is one in which all minimal time lags are equal to zero and all maximal time lags are infinite, whereas the *no-wait* situation implies that all the minimal and maximal time lags are equal to zero. In the *fixed interval* scheduling problem, job J_j has a minimal time lag l_j and a maximal time lag τ_j, where $l_j = \tau_j$ for $j = 1, \ldots, n$.

The minimal (maximal) time lag between two consecutive operations of a job may be *job-dependent* or *job-independent*, *machine-dependent* or *machine-independent*. Let l_{jkl} denote the minimum time delay required by job J_j when transferring from machine M_k to machine M_l. The time lag is denoted by l_j if it is machine-independent, while it is denoted by l_{kl} if it is job-independent.

A time lag requires no additional resources/machines. If jobs would need additional resources during their time lags, the additional resources could be modeled as machines. For example, the problem where a single robot transports jobs from one machine to the next in the two-machine flow shop problem of minimizing the makespan is equivalent to the classical three-machine flow shop problem

$F3||C_{\max}$. Brucker and Knust (2010) summarized NP-hardness results for flow shop problems with transportation times and a single robot.

The focus of this thesis is on scheduling problems with minimal time lags with no bottlenecks resources required during those time lags.

1.4.2 Complexity results

Tables 1.1 and 1.2, adapted from Brucker and Knust (2010), list the complexity results for flow and open shop scheduling problems with minimal time lags.

Table 1.1. Complexity results for flow shop problems with minimal time lags

• Polynomially solvable problems	
$F2\|p_{ij} = 1; l_j \in \{T_1, T_2\}\|C_{\max}$	Yu (1996)
$F2\|l_j = T\|C_{\max}$	Johnson (1954)
$F\|p_{ij} = 1, l_j\| \sum C_j$	Brucker et al (2004)
$F\|p_{ij} = 1, l_k, r_j\| \sum w_j U_j$	Single-machine problem
$F\|P_{ij} = 1, l_k, r_j\| \sum w_j T_j$	Single-machine problem
• Strongly NP-hard problems	
$F2\|p_{ij} = 1, l_j\|C_{\max}$	Yu (1996)
$F2\|l_j \in \{T_1, T_2\}\|C_{\max}$	Yu (1996)
$F2\|r_j\|C_{\max}$	Lenstra et al (1977)
$F3\|\|C_{\max}$	Garey et al (1976)
$F2\|\|L_{\max}$	Lenstra et al (1977)
$F2\|\| \sum C_j$	Garey et al (1976)
$F2\|p_{ij} = 1, l_j, r_j\| \sum C_j$	Brucker et al (2004)
$F2\|p_{ij} = 1, l_j\| \sum w_j C_j$	Brucker et al (2004)

1.4.3 Approximation algorithms

Since most scheduling problems with time lags are strongly NP-hard, we focus on the design and analysis of approximation algorithms. Only few worst-case performance guarantee results are known for approximation algorithms for scheduling problems with time lags.

Table 1.2. Complexity results for open shop problems with minimal time lags

• Polynomially solvable problems	
$O\|p_{ij} = 1; l_{jkl} = T\|C_{\max}$	Rayward-Smith and Rebaine (1992)
$O2\|p_{ij} = 1, l_{jkl}, r_j\|C_{\max}$	Brucker et al (2004)
$O2\|p_{ij} = 1, l_{kl}\|C_{\max}$	Knust (1999)
$O2\|p_{ij} = 1, l_{kl}\|\sum w_j C_j$	Brucker et al (2004)
$O2\|p_{ij} = 1; l_{jkl} = T\|\sum U_j$	Brucker et al (2004)
• Strongly NP-hard problems	
$O2\|l_{jkl} = T\|C_{\max}$	Yu (1996)
$O2\|l_j = T_1, T_2\|C_{\max}$	Yu (1996)
$O3\|\|C_{\max}$	Gonzalez and Sahni (1976)
$O\|p_{ij} = 1, l_{kl} = l_{lk}\|C_{\max}$	Rayward-Smith and Rebaine (1992)
$O2\|p_{ij} = 1, l_j\|C_{\max}$	Yu (1996)
$O2\|r_j\|C_{\max}$	Lawler et al (1981)
$O2\|\|\sum C_j$	Achugbue and Chin (1982)
$O2\|p_{ij} = 1, l_j, r_j\|\sum C_j$	Brucker et al (2004)
$O2\|p_{ij} = 1, l_j\|\sum w_j C_j$	Brucker et al (2004)
$O\|p_{ij} = 1, r_j\|\sum U_j$	Kravchenko (2000)

Flow shop For the two-machine flow shop environment, Dell'Amico (1996) provided a 2-approximation algorithm for $F2|l_j|C_{\max}$. Karuno and Nagamochi (2003) improved on this and gave an $\frac{11}{6}$-approximation algorithm. Ageev (2008) showed that the worst case ratio could be improved to $\frac{3}{2}$ if $p_{1j} = p_{2j}$ for each job J_j $(j = 1, \ldots, n)$.

Open shop For the two-machine open shop environment, Strusevich and Rebaine (1995) presented a $\frac{7}{4}$-approximation algorithm for $O2|l_{hij} = l_{hji}|C_{\max}$. This bound was improved to $\frac{3}{2}$ by Strusevich (1999). Rebaine and Strusevich (1999) presented an $\frac{8}{5}$-approximation algorithm for $O2|l_{ij}|C_{\max}$. Rebaine (2004) presented a 2-approximation algorithm for $O2|l_{hij}|C_{\max}$ and a $(\frac{7}{4} - \frac{1}{2n})$-approximation algorithm for $O2|p_{ij} = p, l_{hij}|C_{\max}$, where n is the number of jobs.

1.5 Some practical problems

We now give two examples of machine scheduling problems with time lags.

Example 1.1: Barges loading and unloading containers at terminals in the Port of Rotterdam (Douma (2008)) In the Port of Rotterdam, there are three clusters of container terminals: Maasvlakte, Botlek, and Vierhavens/Mervehaven (see Figure 1.1).

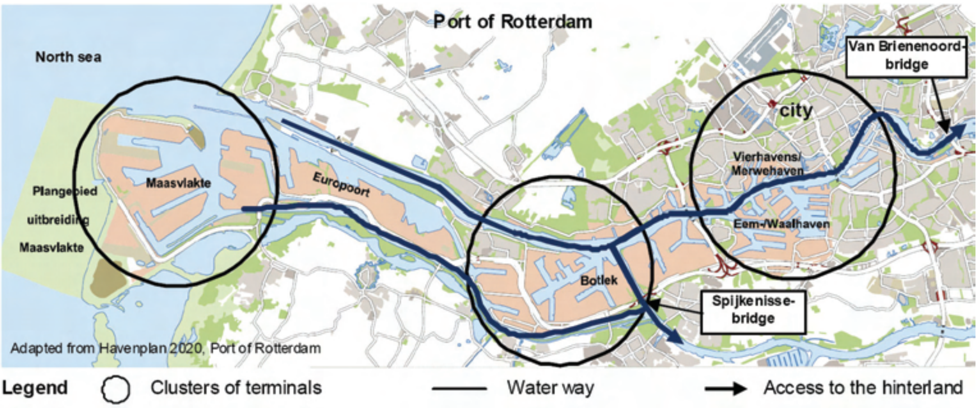

Fig. 1.1. Layout of the Port of Rotterdam. (Adapted from Douma (2008))

Each barge has a sailing plan that specifies the container terminals that need to be called upon for unloading and loading containers, and sailing between clusters takes about one or two hours, whereas sailing from one terminal to another terminal in the same cluster takes about 20 minutes. The main objective of a barge operator is to minimize possible delays in the sailing schedule, which determines the time at which a barge is planned to be in the port and at its hinterland destinations. The way containers are stacked on the barge limits the sequence in which terminals can be visited. If a barge operator decides to visit terminals in a different order, it might happen that a barge has not unloaded enough containers to load the containers available at the next terminal. The loading and unloading time of a barge depends on the speed of the terminal cranes, which may vary between 35 and 45 containers per hour. The Port of Rotterdam's main objective is to minimize the congestion in the port (Van Groningen (2006)). The key performance indicators include the fraction of barges leaving the port late, the average barge tardiness, the average level of congestion per hour, and the maximum level of congestion per hour.

The problem of scheduling barges along container terminals can be modeled as a machine scheduling problem with time lags, where the jobs are the barges coming to and sailing in the Port of Rotterdam, and the machines are the container terminals in the port. If a barge's sequence of container terminals to be called upon is fixed, then the scheduling environment is a *job shop*. Otherwise the scheduling environment is an *open shop*. If the processing times, release times, and the total number of barges are known before they arrive, the scheduling problem is an *off-line* problem; otherwise, the problem is an *on-line* problem. If the container terminal handling times are fixed, the on-line problem is *clairvoyant*. If the processing time of each barge is unknown until it is finishes, the on-line problem is *non-clairvoyant*. The scheduling objective could be to minimize $\sum w_j U_j$, $\sum w_j T_j$, $\sum w_j C_j$, $C_{\max}$, etc.

Example 1.2: The coffee production process (Simeonov and Simeonovová (1997)). Figure 1.2 shows the material flow in a coffee production facility with a single roasting machine. Its per hour output is known and identical for all types of coffee. A pipeline system transports the ground coffee to silos. Coffee is degassed immediately after roasting. For this purpose, it is stored in eight degassing silos of known maximum capacity. Only the same type of coffee can be stored in a silo. The minimum duration of degassing after roasting is specified by the technology and must be strictly adhered to. Besides this, the maximum period of keeping coffee in the silo is also limited. Refer to Table 1.3 for an example with three types of coffee with their minimum and maximum storage times in hours.

Table 1.3. Minimum and maximum storage times in hours

Type of coffee	minimum storage time	maximum storage time
Coffee type A	8	50
Coffee type B	10	50
Coffee type C	12	50

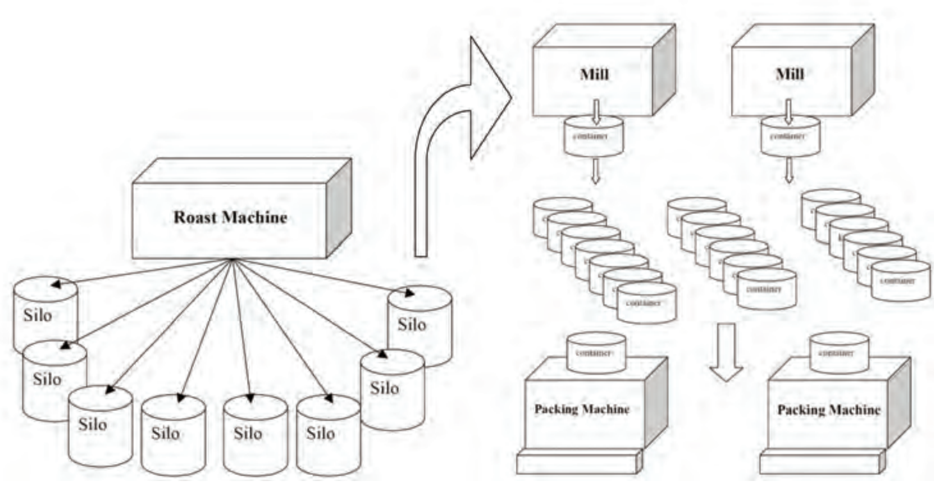

Fig. 1.2. Material flow in coffee production. (Adapted from Simeonov and Simeonovová (1997))

Coffee is transported from the silos to the mills by a tube system controlled by the dispatcher. Coffee is ground in two mills, and their per hour output is known. Ground coffee is transported via the pipeline system to the containers, where additional degassing is carried out. The maximum capacity of each container is known. The duration of degassing after grinding is again specified by the technology. For the three types of coffee it is as follows:

Table 1.4. Minimum and maximum degassing times in hours

Type of coffee	minimum degassing time	maximum degassing time
Coffee type A	2.5	14.5
Coffee type B	20	30
Coffee type C	6	17

After degassing the containers are transported to three packaging machines, where coffee is poured into the loader of the packaging machine. The per hour output of the packaging machine depends on the type of coffee. Some products require a certain packaging machine; others can be processed on any packaging machine.

The problem is to generate a feasible production schedule that maximizes the production throughput. This problem can be modeled as a scheduling problem with time lags. Jobs are production lots, and the machines are the roasting machine, silos, mills, containers, and packaging machines. Since there are a number of parallel machines in each stage, the environment is a *flexible flow shop*. The transportation times between machines can be modeled as *minimal time lags*. The scheduling problem also includes *maximal time lags* induced by the maximal degassing times. Since all data and information are available beforehand, the problem is an *off-line* problem. The scheduling objective function could be to minimize $\sum w_j U_j$, $\sum w_j T_j$, $\sum w_j C_j$, $C_{\max}$, etc.

1.6 Outline of the thesis

The thesis contains four parts and seven chapters. The first part is the introductory chapter. The second part includes Chapters 2 and 3, both devoted to on-line problems, while the third part includes Chapters 4 to 6, devoted to off-line problems. The concluding chapter is the fourth and last part.

In Chapter 2, we consider the on-line two-machine job shop makespan scheduling problem with minimal time lags. We prove that the greedy algorithm is 2-competitive. For the non-clairvoyant variant of the problem, no on-line algorithm can do better. For the clairvoyant variant, no on-line delay algorithm has a competitive ratio better than $\frac{\sqrt{5}+1}{2} \approx 1.618$, and the greedy algorithm is still the best on-line non-delay algorithm. We also show that the same results hold for two machine flow shop problem with time lags. This chapter is based on Zhang and Van de Velde (2010c).

In Chapter 3, we consider the on-line two-machine open shop makespan scheduling problem with minimal time lags. The competitive ratio for the greedy algorithm is 2, and it can be reduced to 5/3 if the maximum time lag is less than the minimum positive processing time of any operation. These ratios are tight.

We also prove that no on-line non-delay algorithm can have a better competitive ratio. As far as delay algorithms are concerned, no algorithm can do better than the greedy algorithm for the non-clairvoyant variant of the problem. For the clairvoyant variant, no on-line delay algorithm has a competitive ratio better than $\sqrt{2}$. This chapter is based on Zhang and Van de Velde (2010d).

In Chapter 4, we consider the NP-hard problem of scheduling n jobs on m parallel two-stage flow shops so to minimize the makespan. This problem decomposes into two subproblems; assigning each job to one of the parallel flow shops, and then scheduling the jobs assigned to the same flow shop by use of Johnson's rule. For $m = 2$, we present a $\frac{3}{2}$-approximation algorithm, and for $m = 3$, we present a $1\frac{5}{7}$-approximation algorithm. Both algorithms run in $O(n \log n)$ time. This chapter is based on Zhang and Van de Velde (2010a).

In Chapter 5, we identify two classes of machine scheduling problems with time lags that possess Polynomial-Time Approximation Schemes (PTASs). These classes together, one for minimizing makespan and one for minimizing total completion time ($\sum C_j$), include many well-studied time lag scheduling problems. The running times of these approximation schemes are polynomial in the number of jobs but exponential in the number of machines and the ratio between the largest time lag and the smallest positive operation time. These classes constitute the first PTAS results for scheduling problems with time lags. This chapter is based on Zhang and Van de Velde (2009).

In Chapter 6, we introduce and analyze the fixed interval shop scheduling problem, where the objective is to maximize the weighted number of jobs that can be processed in a two-stage flow, job, or open shop, if each job has a fixed start and finish time and requires a given transportation time, a time lag, for moving from one stage to the other. We prove that the two-machine fixed interval flow shop problem is NP-hard in the strong sense for general time lags, even in case of unit processing times. The problem is solvable in polynomial time if all time lags are equal. The two-machine fixed interval job shop and open shop problems are solvable in $O(n^3)$ time if the time lags are identical and relatively small. This chapter is based on Zhang and Van de Velde (2010b)

Finally, in Chapter 7, we summarize our results for machine scheduling problems with time lags. We also point out promising directions for further research.

Part II

On-Line Algorithms

2

Two-Machine Job Shop[1]

In the kingdom of the blind, the one-eyed man is king.

- Desiderius Erasmus(1466-1536)

In this chapter, we consider the on-line two-machine job shop scheduling problem with time lags so as to minimize the makespan. Each job consists of no more than two operations and time lags exist between the completion time of the first and the start time of the second operation of any two-operation job. We prove that any greedy algorithm is 2-competitive. For the non-clairvoyant variant of the problem, no on-line algorithm can do better. For the clairvoyant variant, no on-line delay algorithm has a competitive ratio better than $\frac{\sqrt{5}+1}{2} \approx 1.618$, and a greedy algorithm is still the best on-line non-delay algorithm. We also show that the same results hold for the two-machine flow shop problem with time lags.

2.1 Introduction

We consider the two-machine job shop problem with time lags, where jobs arrive over time, to minimize the makespan. In such a system, there is a set $\mathcal{J}$ of n independent jobs $J_1, \ldots, J_n$ that needs scheduling on two machines M_1 and M_2. Each job $J_j \in \mathcal{J}$ consists of no more than two operations O_{ij} ($i = 1, 2$), and

[1] This chapter is based on a paper that has been published in the *Information Processing Letters*. (Zhang and Van de Velde (2010c))

operation O_{ij} requires processing on machine M_i during an uninterrupted non-negative processing time p_{ij} $(i = 1, 2; j = 1, \dots, n)$. The sequence of operations for each job is prescribed. Let $\mathcal{J}^1$ be the set containing all jobs for which O_{1j} has to be scheduled before O_{2j} or O_{2j} is missing (hence need not be scheduled), and let $\mathcal{J}^2$ be the set containing all jobs for which O_{2j} has to be scheduled before O_{1j} or O_{1j} is missing, where $j = 1, \dots, n$. We have $\mathcal{J} = \mathcal{J}^1 \cup \mathcal{J}^2$.

Either machine is available from time 0 onwards and can handle only one job at a time. For each job J_j there is a time lag l_j required between the completion of its first and the start of its second operation. All jobs have release times, which means that the first operation of any job J_j cannot be started before its release time r_j $(j = 1, \dots, n)$. Preemption of jobs, that is, interrupting a job and resuming in at a later point in time, is not allowed. The objective is to minimize the maximum completion time $C_{\max}$, that is, to find a schedule of minimum length or makespan. Following the standard three-field $\alpha|\beta|\gamma$ scheduling notation Graham et al (1979)), we denote the problem as $J2|o_j \le 2, r_j, l_j|C_{\max}$, where o_j is the number of operations of job J_j $(j = 1, \dots, n)$. If $\mathcal{J}^2 = \emptyset$ or $\mathcal{J}^1 = \emptyset$, the problem reduces to the corresponding two-machine flow shop problem, denoted as $F2|r_j, l_j|C_{\max}$.

Time lags have several practical interpretations. They can model the transportation times between machines if the number of vehicles is not restrictive, or if the jobs can travel by themselves, like for example barges sailing between port terminals for loading and unloading containers. Time lags can also model required heating or cooling down times.

The complexity of the *off-line* version of $J2|o_j \le 2, r_j, l_j|C_{\max}$, where all job data are known a priori, is relatively well understood. It is strongly NP-hard, even in the case of unit processing times, since the two-machine flow shop problem $F2|p_{ij} = 1, l_j|C_{\max}$ is already strongly NP-hard (Yu (1996); Yu et al (2004)). Dell'Amico (1996) showed that any instance of $J2|o_j \le 2, l_j|C_{\max}$ can be solved by solving two instances of $F2|l_j|C_{\max}$. Therefore, if all time lags are equal or if the solution of $F2|l_j|C_{\max}$ is restricted to the class of permutation schedules, the related two cases of $J2|o_j \le 2, l_j|C_{\max}$ are polynomially solvable (See Table 1.2). Panwalkar (1973) identified another well-solvable special case of the job shop problem with time lags. As far as we know, no approximation algorithm has been developed for the general $J2|o_j \le 2, l_j|C_{\max}$ problem.

We study the *on-line* version, where the jobs dynamically arrive at a priori unknown points in time (the so-called *release times*) and the job data are not known a priori. We also do not know the number of jobs to be scheduled. In particular, we study the *non-clairvoyant* variant, in which the processing time of an operation is unknown until it has finished, and the required time lag is unknown until it has elapsed.

The quality of an on-line algorithm is typically measured by its *competitive ratio*, and an on-line algorithm is called ρ-competitive if the objective value of the schedule produced by the on-line algorithm is at most ρ times the value of an optimal off-line solution, for any instance of the problem. An on-line algorithm is called *best possible* if no one-line algorithm has a lower competitive ratio.

Results for on-line job shop and flow shop scheduling problems with time lags are very scarce. For the case with unit execution time and arbitrary time lags without release times, Rayward-Smith and Rebaine (2008) present $(2-\frac{3}{n+2})$-competitive algorithms for $F2|on\text{-}line, p_{ij} = 1, l_j|C_{\max}$, where n is the number of jobs. The competitive ratio is proved to be tight, which means that the ratio holds with equality for specific instances of the problem. For the case without time lags, Sgall (1998) shows that no deterministic algorithm is better than 2-competitive for $F2|on\text{-}line|C_{\max}$. For the on-line two-machine open shop problem with time lags, Zhang and Van de Velde (2010d) prove that any greedy algorithm has a tight competitive ratio of 2 and this ratio is 5/3 in the case of small time lags, that is, if the maximum time lag is no larger than the smallest processing time. *A greedy algorithm* for an on-line scheduling problem with time lags assigns to a machine any available operation as soon as the machine becomes available. Zhang and Van de Velde (2010d) also prove that no on-line non-delay algorithm can have a better competitive ratio. As far as delay algorithms are concerned, that is, algorithms that allow a machine to be idle while an operation is available for processing, no delay algorithm can do better than a greedy algorithm for the non-clairvoyant variant of the problem. For the clairvoyant variant, no on-line delay algorithm has a competitive ratio better than $\sqrt{2}$.

In this chapter, we analyze the performance of a greedy algorithm for the on-line version of $J2|o_j \leq 2, r_j, l_j|C_{\max}$ that processes an available operation as soon as possible. If there are more than two operations available, the algorithm processes one of them randomly. Accordingly, the resulting schedule is *non-delay*, that is, no machine is kept idle while an operation is waiting to be processed.

We prove that the competitive ratio of any greedy algorithm is 2, this bound is tight, and no on-line non-delay algorithm can do better. Using an adversary strategy argument, we also prove that no on-line delay algorithm can have a better performance guarantee for the non-clairvoyant variant of the problem. For the clairvoyant version of the problem, we prove that no on-line delay algorithm can have a better competitive ratio than $\frac{\sqrt{5}+1}{2} \approx 1.618$. We prove that these results apply to $F2|on\text{-}line, r_j, l_j|C_{\max}$ also.

2.2 Performance analysis of a greedy algorithm

Let G be any greedy algorithm. We prove that G is 2-competitive for the on-line two machine job shop scheduling problem with time lags.

Let r_j be the arrival time of job J_j $(j = 1, \ldots, n)$. For a given instance, let $C^*_{\max}$ denote the optimized completion time of n jobs and $C^G_{\max}$ denote the completion time of the schedule given by the greedy algorithm G. Due to symmetry of the argument, we can assume without loss of generality that machine M_2 finishes last. For the schedule constructed by G, let S_{ij} and C_{ij} denote the starting and completing time of O_{ij}, respectively $(i = 1, 2; j = 1, \ldots, n)$.

For any subset $\mathcal{H} \subseteq \{J_1, \ldots, J_n\}$, we define

$$\begin{aligned}
r(\mathcal{H}) &= \min_{J_j \in \mathcal{H}} r_j \\
p_1(\mathcal{H}) &= \sum_{J_j \in \mathcal{H}} p_{1j} \\
p_2(\mathcal{H}) &= \sum_{J_j \in \mathcal{H}} p_{2j} \\
C(\mathcal{H}) &= \max_{J_j \in \mathcal{H}} (r_j + p_{1j} + l_j + p_{2j}).
\end{aligned}$$

Clearly, we have that

$$C^*_{\max} \geq \max_{\mathcal{H} \in \mathcal{J}} \{r(\mathcal{H}) + p_1(\mathcal{H}), r(\mathcal{H}) + p_2(\mathcal{H}), C(\mathcal{H})\} \tag{2.1}$$

Lemma 1 *If there is no idle time before $C_{\max}^G$ on machine M_2, then $C_{\max}^* = C_{\max}^G$.* □

Lemma 1 is obviously true. So, in the remainder we suppose machine M_2 has idle time before $C_{\max}^G$.

Let T denote the last point in time such that M_2 is busy throughout the time interval $[T, C_{\max}^G]$ but idle immediately before time T. Consider now the jobs with $S_{2j} \geq T$ on machine M_2. We divide these jobs into two disjoint subsets: subset $\mathcal{X}$ contains all the jobs with $r_j < T$, and subset $\mathcal{Y}$ contains all the jobs with $r_j \geq T$.

Lemma 2 *If $\mathcal{X} = \emptyset$, we have $C_{\max}^* = C_{\max}^G$.*

Proof. Note that if $\mathcal{X} = \emptyset$, then $\mathcal{Y}$ cannot be empty, and hence we have that

$$C_{\max}^G = T + p_2(\mathcal{Y}) \leq r(\mathcal{Y}) + p_2(\mathcal{Y}) \leq C_{\max}^*.$$

So, if $\mathcal{X} = \emptyset$, then the greedy algorithm G has returned an optimal schedule. □

Lemma 3 *If $\mathcal{Y} \neq \emptyset$, we have $C_{\max}^G \leq 2C_{\max}^*$.*

Proof. In this case, we have that

$$C_{\max}^G = T + p_2(\mathcal{X}) + p_2(\mathcal{Y}) \leq r(\mathcal{Y}) + p_2(\mathcal{Y}) + r(\mathcal{X}) + p_2(\mathcal{X}) \leq 2C_{\max}^*,$$

and we are done. □

So, we need to analyze the case $\mathcal{X} \neq \emptyset$ and $\mathcal{Y} = \emptyset$ in the remaining discussion.

Let Q denote the completion time of machine M_1, that is, the completion time of the last job scheduled on M_1. Note that we have assumed without loss of generality that $Q \leq C_{\max}^G$. Let $[B_k, E_k]$ for $k = 1, \ldots, h$, with $B_1 = r(\mathcal{X})$ and $E_h = Q$, be the busy intervals of machine M_1 ($B_1 < B_2 < \cdots < B_h$), and correspondingly let $[E_k, B_{k+1}]$ for $k = 1, \ldots, h-1$ be the idle intervals of M_1 between time $r(\mathcal{X})$ and time Q. Let $\mathcal{X}^k = \{J_j \in \mathcal{X} | B_k \leq S_{1j} < E_k\}$ for $k = 1, \ldots, h$; and let $\Delta_k = B_{k+1} - E_k$, for $k = 1, \ldots, h-1$. If $Q < T$, let $\Delta_h = T - Q$; otherwise, let $\Delta_h = 0$. For an illustration of these concepts, see Figure 2.1.

Lemma 4 *If $\mathcal{X} \neq \emptyset$, $\mathcal{Y} = \emptyset$, then $r(\mathcal{X}) + \sum_{k=1}^{h} \Delta_k + p_2(\mathcal{X}) \leq C_{\max}^*$.*

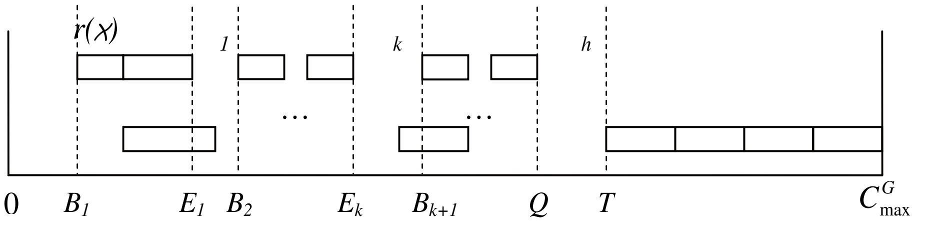

Fig. 2.1. The illustration of Δ_k, Q, T and busy periods.

Proof. We branch into two cases: (1) $h = 1$; (2) $h > 1$.

Case 1: $h = 1$. Machine M_1 is busy in the period $[r(\mathcal{X}), Q]$ in this case. If $Q \geq T$, we have $\Delta_h = 0$. Due to (2.1), it holds that $r(\mathcal{X}) + p_2(\mathcal{X}) \leq C^*_{\max}$. If $Q < T$, we have $\Delta_h \leq \min_{J_j \in \mathcal{X}} l_j$; otherwise, at least one job in $\mathcal{X}$ could has been started before time T on machine M_2. Then it holds that $r(\mathcal{X}) + \Delta_h + p_2(\mathcal{X}) \leq C^*_{\max}$.

Case 2: $h > 1$. For any job $J_j \in \mathcal{X}^1$ we have $l_j \geq \sum_{k=1}^{h} \Delta_k$; otherwise, its second operation could have been started before time T on machine M_2. Hence, we have that $S_{2j} \geq r(\mathcal{X}) + \sum_{k=1}^{h} \Delta_k$ for any job $J_j \in \mathcal{X}^1$. By the same token, for any job $J_j \in \mathcal{X}^i (i = 2, \ldots, h)$, we have $l_j \geq \sum_{k=i}^{h} \Delta_k$, otherwise, its second operation could have been started before time T on machine M_2. On the other hand, for each $J_j \in \mathcal{X}^i$ $(i = 2, \ldots, h)$, $r_j \geq B_i \geq r(\mathcal{X}) + \sum_{k=1}^{i-1} \Delta_k$, and we have $S_{2j} \geq r_j + l_j \geq r(\mathcal{X}) + \sum_{k=1}^{i-1} \Delta_k + \sum_{k=i}^{h} \Delta_k = r(\mathcal{X}) + \sum_{k=1}^{h} \Delta_k$ for any job $J_j \in \mathcal{X}^i$ $(i = 2, \ldots, h)$.

Accordingly, $r(\mathcal{X}) + \sum_{k=1}^{h} \Delta_k + p_2(\mathcal{X})$ is a lower bound on $C^*_{\max}$. □

Lemma 5 *If $\mathcal{X} \neq \emptyset$, $\mathcal{Y} = \emptyset$, then $C^*_{\max}/C^G_{\max} \leq 2$.*

Proof. In this case, we have

$$C_{\max}^{G} = r(\mathcal{X}) + (T - r(\mathcal{X})) + p_2(\mathcal{X}) \le r(\mathcal{X}) + \sum_{k=1}^{h} \Delta l_k + p_1(\mathcal{J}) + p_2(\mathcal{X}).$$

Using Lemma 9 and (3.1), we obtain $C_{\max}^{G} \le 2C_{\max}^{*}$. □

Theorem 1 *Any greedy algorithm G is 2-competitive for $J2|on\text{-}line, o_j \le 2, r_j, l_j|C_{\max}$, this bound is tight, and no non-delay algorithm can do better.*

Proof. For any instance of $J2|on\text{-}line, o_j \le 2, r_j, l_j|C_{\max}$, a greedy algorithm G generates a schedule for which Lemma 1, 2, 3 or 5 holds. Therefore, any greedy algorithm is 2-competitive. To see that this bound is tight, consider the following instance of the problem. At time $t = 0$, job $J_1 \in \mathcal{J}^1$ arrives with $p_{11} = 1, l_1 = 0$ and $p_{21} = 0$; at time $t = \epsilon$, job $J_2 \in \mathcal{J}^1$ arrives with $p_{12} = \epsilon, l_2 = 0$ and $p_{22} = 1-\epsilon$, for some $0 < \epsilon < 1/2$. Any non-delay algorithm gives a schedule of length 2, but the optimal makespan is $1 + 2\epsilon$. Therefore, the ratio $\frac{2}{1+2\epsilon}$ is best possible for any non-delay algorithm. Since ϵ may be arbitrarily small, the ratio can be arbitrarily close to 2. □

Since we used an instance of $F2|on\text{-}line, r_j, l_j|C_{\max}$ as an adversary example in the proof of Theorem 1, we have the following corollary.

Corollary 1 *Any greedy algorithm is 2-competitive for $F2|on\text{-}line, r_j, l_j|C_{\max}$, this bound is tight, and no non-delay algorithm can do better.* □

2.3 The worst-case performance of on-line delay algorithms

In this section, we analyze on-line delay algorithms. Kanet (1986) and Kanet and Sridharan (2000)) showed that allowing tactical delays may lead to better schedules. We consider two versions of the problem: the non-clairvoyant version and the clairvoyant version.

Theorem 2 *For the non-clairvoyant version of $J2|on\text{-}line, o_j \le 2, r_j, l_j|C_{\max}$, no on-line delay algorithm has a better competitive ratio than an on-line greedy algorithm.*

Proof. We use an adversary strategy against which any on-line delay algorithm must perform poorly. Consider the following instance of the problem. At time

$t = 0$, job $J_1 \in \mathcal{J}^1$ arrives with $p_{11}, l_1 = 0$ and $p_{21} = 0$. Let algorithm H be any delay algorithm. Suppose algorithm H imposes a deliberate delay $S_1 \geq 0$ to schedule job J_1.

If $S_1 > 0$, let $n = 1$ and $p_{11} = S_1$. The minimum makespan is now S_1, but algorithm H will finish the job at time $2S_1$. Its competitive ratio is therefore at least 2.

If $S_1 = 0$, let $n = 2$ and job $J_2 \in \mathcal{J}^1$ arrives at time $t = \epsilon$ with $p_{12} = \epsilon, l_2 = 0$ and $p_{22} = 1 - \epsilon$. The instance then boils down to the one used in Theorem 1 to prove the competitive ratio 2 for greedy algorithms, and algorithm H does not outperform a greedy algorithm. □

Accordingly, any greedy algorithm is a best possible algorithm for the non-clairvoyant variant of the problem.

For the clairvoyant version, we prove that any on-line delay algorithm has a competitive ratio of at least $\frac{\sqrt{5}+1}{2} \approx 1.618$. However, this results leaves open the question whether an on-line delay algorithm has a better competitive ratio than a greedy algorithm.

Theorem 3 *For the clairvoyant version of $J2|on\text{-}line, o_j \leq 2, r_j, l_j|C_{\max}$, no on-line delay algorithm has a competitive ratio better than $\frac{\sqrt{5}+1}{2}$.*

Proof. Consider the following instance of the problem. At time $t = 0$, job $J_1 \in \mathcal{J}^1$ arrives with $p_{11} = 1, l_1 = 0$ and $p_{21} = 0$. Let algorithm H be any delay algorithm, and suppose algorithm H imposes a deliberate delay $S_1 \geq 0$ before job J_1. Suppose that either no further job arrives, or job $J_2 \in \mathcal{J}^1$ arrives at time $S_1 + \epsilon$, $0 < \epsilon \leq 1/2$, with $p_{12} = \epsilon, l_2 = 0$, and $p_{22} = 1 - \epsilon$. In the former case, the schedule given by H has makespan $1 + S_1$, while in the latter case the makespan is at least $2 + S_1$. The optimal makespans are 1 and $1 + S_1 + 2\epsilon$, respectively. So, the competitive ratio of algorithm H is larger than or equal to $\max\{\frac{1+S_1}{1}, \frac{2+S_1}{1+S_1+2\epsilon}\}$. The minimum of this expression is achieved for $S_1 = \frac{\sqrt{5}-1}{2}$, and the competitive ratio is therefore at least $\frac{\sqrt{5}+1}{2}$.

□

Since we used instances of $F2|on\text{-}line, r_j, l_j|C_{\max}$ as adversary examples in the proofs of Theorem 2 and 3, similar results hold for the corresponding flow shop problem.

2.4 Conclusions

We have proved that any greedy algorithm for the on-line two-machine job shop scheduling problem with time lags has a tight competitive ratio of 2. For the non-clairvoyant versions of these problems, any greedy algorithm is a best possible algorithm. For the clairvoyant versions, any greedy algorithm is a best possible non-delay algorithm; however, since we could prove a lower bound of only $\frac{\sqrt{5}+1}{2} \approx 1.618$ on the competitive ratio in general, a delay algorithm may have a better competitive ratio than a greedy algorithm. We have proved that these results also apply to the on-line two-machine flow shop scheduling problem with time lags.

3

Two-Machine Open Shop[1]

Information is the resolution of uncertainty.

- Claude Elwood Shannon (1916-2001)

In this chapter, we analyze the performance of the greedy algorithm for the on-line two-machine open shop scheduling problem of minimizing makespan, in which time lags exist between the completion time of the first and the start time of the second operation of any job. The competitive ratio for the greedy algorithm is 2, and it can be reduced to 5/3 if the maximum time lag is less than the minimum positive processing time of any operation. These ratios are tight. We also prove that no on-line non-delay algorithm can have a better competitive ratio.

As far as delay algorithms are concerned, no algorithm can do better than the greedy algorithm for the non-clairvoyant variant of the problem. For the clairvoyant variant, no on-line delay algorithm has a competitive ratio better than $\sqrt{2}$.

3.1 Introduction

We consider the two-machine open shop problem with time lags to minimize the makespan. In such a system, there is a set $\mathcal{J}$ of n independent jobs $J_1, \ldots, J_n$ that

[1] This chapter is based on a paper that has been published in the *European Journal of Operational Research.* (Zhang and Van de Velde (2010d))

need scheduling on two machines M_1 and M_2. Each job $J_j \in \mathcal{J}$ consists of two operations O_{ij} $(i = 1, 2)$, and operation O_{ij} requires processing on M_i during an uninterrupted processing time p_{ij} $(i = 1, 2; j = 1, \ldots, n)$. The operations of a job may be scheduled in either order, as long as their execution does not overlap. The machines are available from time 0 onwards and can handle only one job at a time. For each job J_j, there is a time lag l_j required between the completing of its first and the starting of its second operation; accordingly, the time lag is independent of the order in which the two operations are scheduled. Preemption of jobs, that is, interrupting a job and resuming in at a later point in time, is not allowed. The objective is to minimize the maximum completion time $C_{\max}$, that is, to find a schedule of minimum length or makespan.

The *off-line* version of this problem, where all job data are known a priori and all jobs are available at time zero, is NP-hard, even in the case of unit processing times (Yu (1996)) and all time lags equal (Rayward-Smith and Rebaine (1992)). Most of the research has therefore been focused on obtaining polynomial-time approximation algorithms with constant performance guarantees. An approximation algorithm for a minimization problem is said to have performance guarantee or performance ratio ρ for some real $\rho > 1$, if it always delivers a solution with objective function value at most ρ times the optimal value. It is then called a ρ-approximation algorithm. Strusevich (1999) presents an algorithm for this basic problem with performance ratio 3/2. Rebaine and Strusevich (1999) give a 8/5-approximation algorithm for the case of asymmetrical (or route-dependent) time lags, that is, the case in which the time lag between the two operations of a job depends on the order in which they are executed. They also study the problem with small time lags, in which any time lag is smaller than or equal to the smallest processing time and show that this special case is solvable to optimality in $O(n)$ time.

We study the *on-line* version, where the jobs dynamically arrive at a priori unknown points in time (the so-called *release times*) and the job data are not known a priori. In particular, we study the *non-clairvoyant* variant, in which the processing time of either operation of a job is unknown until it finishes, and the required time lag is unknown until it has elapsed.

The quality of an on-line algorithm is typically measured by its *competitive ratio*, and an on-line algorithm is called ρ-competitive if the objective value of the schedule produced by the on-line algorithm is at most ρ times the value of an optimal off-line solution, for any instance of the problem. An on-line algorithm is called *best possible* if no one-line algorithm has a lower competitive ratio.

Thus far, results are known only for on-line open shop makespan problems without time lags. For the clairvoyant problem, Chen et al (1998) present 5/4- and 3/2-competitive algorithms for the preemptive and non-preemptive version, respectively; for the non-clairvoyant model, they present an algorithm with a competitive ratio 3/2 for both the preemptive and non-preemptive version. All algorithms are proved to be best possible. For the case in which only permutation schedules are allowed, Chen et al (2001) propose a 1.848-competitive algorithm for the non-preemptive problem and show that no permutation algorithm can have a competitive ratio better than 1.754. They also develop a 27/19-competitive algorithm for the preemptive three-machine problem, which is shown to be best possible.

Ours is the first result on on-line open shop scheduling problem with time lags. We analyze the performance of the greedy algorithm that processes an available operation as soon as possible, with ties broken arbitrarily. Note that an operation is available for processing as soon as the corresponding job has arrived and the other operation of this job either has not been processed yet, or has been completed and sufficient time, that is, the required time lag, has elapsed since its completion. Accordingly, the resulting schedule is *non-delay*, that is, no machine is kept idle at a time when it could begin processing an operation.

We prove that the competitive ratio for the greedy algorithm is 2, this bound is tight, and no on-line non-delay algorithm can do better. We also analyze a special case of the problem introduced by Rebaine and Strusevich (1999), where the maximum time lag is less than the minimum positive processing time of any operation. This is in fact a *semi-online* problem, since the algorithm will have further knowledge about the structure of the problem in advance (Pruhs and Torng (1994)). We prove that the competitive ratio is 5/3, this ratio is tight, and no non-delay algorithm can have a better competitive ratio.

Using an adversary strategy argument, we also prove that no on-line delay algorithm, that is, an algorithm that allows a machine to be idle while an operation is available for processing, can have a better performance guarantee for the non-clairvoyant variant of the problem. For the clairvoyant version of the problem, we prove that no on-line delay algorithm can have a better competitive ratio than $\sqrt{2}$.

3.2 Notations and preliminaries

In this section, we introduce notation and preliminaries that are useful for the remainder of the chapter.

Definition
A greedy algorithm is an algorithm in which the available operations are processed as soon as the machines become idle. Ties are broken arbitrarily.

Let r_j be the arrival time of job J_j $(j = 1, \ldots, n)$. For a given instance, let $C^*_{\max}$ denote the minimum makespan and $C^G_{\max}$ denote the makespan of the schedule given by the greedy algorithm. Also, for the schedule constructed by the greedy algorithm, let S_{ij} and C_{ij} denote the starting and completing time of O_{ij}, respectively $(i = 1, 2; j = 1, \ldots, n)$.

For any subset $\mathcal{H} \subseteq \{J_1, \ldots, J_n\}$, we define

$$\begin{aligned} r(\mathcal{H}) &= \min_{J_j \in \mathcal{H}} r_j \\ p_1(\mathcal{H}) &= \sum_{J_j \in \mathcal{H}} p_{1j} \\ p_2(\mathcal{H}) &= \sum_{J_j \in \mathcal{H}} p_{2j} \\ C(\mathcal{H}) &= \max_{J_j \in \mathcal{H}} (r_j + p_{1j} + l_j + p_{2j}). \end{aligned}$$

Clearly, we have that

$$C^*_{\max} \geq \max_{\mathcal{H} \in \mathcal{J}} \{r(\mathcal{H}) + p_1(\mathcal{H}), r(\mathcal{H}) + p_2(\mathcal{H}), C(\mathcal{H})\} \tag{3.1}$$

Consider any schedule constructed by a greedy algorithm. Due to symmetry of the argument, we can assume without loss of generality that machine M_2 finishes last. If there is no idle time on M_2 before $C^G_{\max}$, then the greedy algorithm produces an optimal schedule. So, in the remainder we suppose there is idle time on M_2.

Let T denote the last point in time such that M_2 is busy in the time interval $[T, C^G_{\max}]$ but idle immediately before time T. Consider now the jobs with $C_{2j} \geq T$. We divide these jobs into two disjoint subsets: subset $\mathcal{X}$ contains all the jobs with $r_j < T$; and subset $\mathcal{Y}$ contains all the jobs with $r_j \geq T$.

Furthermore, note that if $\mathcal{X} = \emptyset$, then $\mathcal{Y}$ cannot be empty, and hence we have that

$$C^G_{\max} = T + p_2(\mathcal{Y}) \leq r(\mathcal{Y}) + p_2(\mathcal{Y}) \leq C^*_{\max}.$$

So, if $\mathcal{X} = \emptyset$, then the greedy algorithm has returned an optimal schedule. The remainder of the analysis will therefore focus on the case that $\mathcal{X} \neq \emptyset$.

Note that we must have

$$C_{1j} + l_j \geq T, \text{ for each } J_j \in \mathcal{X}, \tag{3.2}$$

for otherwise O_{2j} would have been started before time T.

We are now ready to proceed with the specific parts of the proofs.

3.3 Performance analysis of the greedy algorithm

3.3.1 On-line case

Theorem 4 *The greedy algorithm has a competitive ratio of 2, this bound is tight, and no non-delay algorithm can do better.*

Proof. As indicated in the previous subsection, we may assume without loss of generality that there is idle time on M_2 and $\mathcal{X} \neq \emptyset$. We analyze two disjoint cases: (i) $\mathcal{Y} \neq \emptyset$; and (ii) $\mathcal{Y} = \emptyset$.

If $\mathcal{Y} \neq \emptyset$, we have that

$$C^G_{\max} = T + p_2(\mathcal{X}) + p_2(\mathcal{Y}) \leq r(\mathcal{Y}) + p_2(\mathcal{Y}) + p_2(\mathcal{X}) \leq 2C^*_{\max},$$

and we are done.

If $\mathcal{Y} = \emptyset$, then let O_{2k} be the operation started at time T; hence, $S_{2k} = T$. We also must have that $C_{1k} + l_k = S_{2k}$; after all, if $C_{1k} + l_k < S_{2k}$, then O_{2k} would have started before time T. Accordingly, we must have that

$$C^G_{\max} = S_{1k} + p_{1k} + l_k + p_{2k} + p_2(\mathcal{X} - J_k).$$

Also, we know that both machines are busy in the interval $[r_k, S_{1k}]$ for otherwise job J_k would have started earlier. Since $r_k + p_{1k} + l_k + p_{2k} \le C^*_{\max}$ and $(S_{1k} - r_k) + p_2(\mathcal{X} - J_k) \le p_2(\mathcal{J}) \le C^*_{\max}$, we have

$$\begin{aligned} C^G_{\max} &= (S_{1k} - r_k) + r_k + p_{1k} + l_k + p_{2k} + p_2(\mathcal{X} - J_k) \\ &\le 2C^*_{\max}. \end{aligned}$$

Hence, any greedy algorithm is a 2-competitive algorithm.

To see that this bound is tight, consider the following instance of the problem. At time $t = 0$, jobs $J_1 = (p_{11} = 1, l_1 = 0, p_{21} = 0)$ and $J_2 = (p_{12} = 0, l_1 = 0, p_{22} = 1)$ arrive; at time $t = \epsilon$, job $J_3 = (p_{13} = \epsilon, l_1 = 1 - 2\epsilon, p_{23} = \epsilon)$ arrives, with some $0 < \epsilon \le 1/2$. Any non-delay algorithm gives a schedule with length equal to 2, but the optimal makespan is $1 + 2\epsilon$. Therefore, the ratio $\frac{2}{1+2\epsilon}$ is best possible for any non-delay algorithm. Since ϵ may be arbitrarily small, the ratio can be arbitrarily close to 2.

This completes the proof. $\square$

3.3.2 Semi-online case

As Pruhs and Torng (1994) point out, on-line algorithms may typically behave poorly because of a large variance of job parameters, which may be exceptional in real life. It is therefore of interest to analyze special cases in which the algorithm is fed additional knowledge about the structure of the problem in order to further our understanding of the general problem. Such an algorithm is called a semi-online algorithm. This consideration has motivated us to analyze the special case introduced and justified by Rebaine and Strusevich (1999), in which the time lags are relatively small for all jobs. Specifically, they assume that

$$l_k \le \min_{i=1,2;1\le j\le n} p_{ij}, \text{ for each } k = 1, \ldots, n, \text{ and } p_{ij} > 0, \tag{3.3}$$

and show that the off-line version is solvable in $O(n)$ time. We consider the semi-online setting of this problem, where the release times, processing times and time lags are not known before the jobs arrive. Furthermore, we assume that operations with zero processing time require no processing at all. We prove the following result.

Theorem 5 *The greedy algorithm has a competitive ratio of 5/3 if condition (3.3) is satisfied, this bound is tight, and no non-delay algorithm can do better.*

Proof. Since $C_{1j} + l_j \geq T$ for each $J_j \in \mathcal{X}$ together with condition (3.3), there can be no more than two jobs in $\mathcal{X}$.

Case 1. There is only one job in $\mathcal{X}$, say, J_k.

We know that during the time interval $[r_k, S_{1k}]$ both machines are busy, otherwise the algorithm would have started J_k earlier. We split this case up in two:

Case 1.1 $\mathcal{Y} \neq \emptyset$;

In this case, one or more jobs are released after time T, and hence we have

$$\begin{aligned} C_{\max}^G &= T + p_{2k} + p_2(\mathcal{Y}) \\ &\leq r(\mathcal{Y}) + p_2(\mathcal{Y}) + p_{2k} \\ &\leq C_{\max}^* + p_{2k}. \end{aligned}$$

If $p_{2k} \leq \frac{2}{3}C_{\max}^*$, then $C_{\max}^G \leq \frac{5}{3}C_{\max}^*$, and we are done.

Alternatively, if $p_{2k} > \frac{2}{3}C_{\max}^*$, we must have that $r_k + p_{1k} + l_k < \frac{1}{3}C_{\max}^*$, because of (3.1). Furthermore, let I_2 denote the total idle time on M_2; note that idle time on M_2 must occur in the intervals $[0, r_k]$ and $[S_{1k}, T]$. Since $C_{1k} + l_k = S_{1k} + p_{1k} + l_k \geq T$, we derive that

$$\begin{aligned} I_2 &\leq r_k + T - S_{1k} \\ &\leq r_k + p_{1k} + l_k \\ &< \frac{1}{3}C_{\max}^*. \end{aligned} \tag{3.4}$$

Hence we have that

$$C_{\max}^G = p_2(\mathcal{J}) + I_2 < \frac{4}{3}C_{\max}^*.$$

Case 1.2 $\mathcal{Y} = \emptyset$.

In this case, it must be that $C_{1k} + l_k = T$, since J_k is the only job in $\mathcal{X}$. Hence, we have

$$C_{\max}^G = S_{1k} + p_{1k} + l_k + p_{2k} = (S_{1k} - r_k) + r_k + p_{1k} + l_k + p_{2k}.$$

If $S_{1k} - r_k \leq \frac{2}{3}C^*_{\max}$, since $r_k + p_{1k} + l_k + p_{2k} \leq C^*_{\max}$, we have $C^G_{\max} \leq \frac{5}{3}C^*_{\max}$, and we are done. Therefore, we assume $S_{1k} - r_k > \frac{2}{3}C^*_{\max}$ in the following discussion.

We define $\mathcal{V}$ as the set of jobs that have operations finishing in the time interval $(r_k, S_{1k}]$ on machine M_1, that is, $\mathcal{V} = \{J_j | r_k < C_{1j} \leq S_{1k}\}$; remember that both machines are busy during this time interval. Because of (3.1), we have that

$$r(\mathcal{V} + J_k) + p_1(\mathcal{V} + J_k) \leq C^*_{\max}. \tag{3.5}$$

Assume that $r_\theta = r(\mathcal{V})$ for some $J_\theta \in \mathcal{V}$. Inequality (3.5) boils then down to

$$r_\theta + p_1(\mathcal{V}) + p_{1k} \leq C^*_{\max}.$$

Since $p_1(\mathcal{V}) \geq S_{1k} - r_k > \frac{2}{3}C^*_{\max}$, we obtain that

$$r_\theta + p_{1k} < \frac{1}{3}C^*_{\max}, \tag{3.6}$$

and hence also that

$$r_\theta + l_k < \frac{1}{3}C^*_{\max}. \tag{3.7}$$

Apparently, it holds that $r_\theta \leq r_k$, and we need to branch into three subcases.

Case 1.2.1 Machine M_1 is either busy during the period of $[r_\theta, r_k]$, or $r_\theta = r_k$.

Let I_1 denote the total idle time on M_1; note that idle time on M_1 must occur in the intervals $[0, r_\theta]$ and $[C_{1k}, C^G_{\max}]$. Since $C_{1k} + l_k = T$ and $C^G_{\max} = T + p_{2k}$, we derive that

$$I_1 \leq r_\theta + l_k + p_{2k}.$$

With $p_{2k} < \frac{1}{3}C^*_{\max}$ due to $S_{1k} - r_k > \frac{2}{3}C^*_{\max}$ and (3.7), we have that

$$C^G_{\max} = p_1(\mathcal{J}) + I_1 < \frac{5}{3}C^*_{\max}.$$

Case 1.2.2 Machine M_2 is busy during the period of $[r_\theta, r_k]$.

Let I_2 denote the total idle time on M_2; note that idle time on M_2 must occur in the intervals $[0, r_\theta]$ and $[S_{1k}, C_{1k} + l_k]$. We derive that

$$I_2 \leq r_\theta + p_{1k} + l_k.$$

With $l_k < \frac{1}{3}C^*_{\max}$ due to $p_{2k} < \frac{1}{3}C^*_{\max}$ and (3.6), we have that

$$C^G_{\max} = p_2(\mathcal{J}) + I_2 < \frac{5}{3}C^*_{\max}.$$

Case 1.2.3 Both machines have idle time in the period $[r_\theta, r_k]$, that is, they are not fully occupied in this interval.

In this case, the analysis is substantially more intricate. Since $O_{1\theta}$ is finished in the period of $[r_k, S_{1k}]$ and job J_θ is released at r_θ, $O_{2\theta}$ must start before $O_{1\theta}$ for otherwise $O_{1\theta}$ would have started earlier on machine M_1, and $O_{2\theta}$ must end before r_k, for otherwise machine M_2 would be fully occupied in the interval $[r_\theta, r_k]$. Due to $S_{1k} - r_k > \frac{2}{3}C^*_{\max}$, it holds that

$$p_{2\theta} + p_{2k} < \frac{1}{3}C^*_{\max}, \tag{3.8}$$

and hence

$$l_\theta + l_k < \frac{1}{3}C^*_{\max}. \tag{3.9}$$

During the period $[r_\theta, S_{2\theta}]$, both machines must be busy otherwise job J_θ would have started earlier. Define

$$\Delta_\theta = \max\{r_k - (C_{2\theta} + l_\theta), 0\}.$$

If $\Delta_\theta = 0$, we have $I_2 \leq r_\theta + l_\theta + p_{1k} + l_k$. Using (3.7) and (3.9), we have $C^G_{\max} = P_2(\mathcal{J}) + I_2 \leq \frac{5}{3}C^*_{\max}$ and we are done. Hence, we assume $\Delta_\theta > 0$ in the following discussion.

If $\Delta_\theta > 0$, machine M_1 must be busy during the period $[C_{2\theta} + l_\theta, r_k]$, otherwise $O_{1\theta}$ would have started earlier. Therefore, in this case, we have that

$$I_1 \leq r_\theta + p_{2\theta} + l_\theta + l_k + p_{2k}, \text{ and} \tag{3.10}$$

$$I_2 \leq r_\theta + l_\theta + \Delta_\theta + p_{1k} + l_k. \tag{3.11}$$

We further branch this case into three subcases.

Case 1.2.3.1 There is an operation O_{1a} finished before $C_{2\theta} + l_\theta$ in machine M_1.

In this case, we have $p_{1a}+\Delta_\theta+p_{1k} < \frac{1}{3}C^*_{\max}$ due to $S_{1k}-r_k > \frac{2}{3}C^*_{\max}$. Hence, we get

$$l_\theta + \Delta_\theta + p_{1k} < \frac{1}{3}C^*_{\max},$$

since $l_\theta \le p_{1a}$ due to (3.3). Together with (3.7) and (3.11), we have $C^G_{\max} = P_2(\mathfrak{J}) + I_2 < \frac{5}{3}C^*_{\max}$.

Case 1.2.3.2 There is an operation O_{2b} finished before $S_{2\theta}$ in machine M_2.

In this case, we have $p_{2b}+p_{2\theta}+p_{2k} < \frac{1}{3}C^*_{\max}$ due to $S_{1k}-r_k > \frac{2}{3}C^*_{\max}$. Hence, we get

$$l_\theta + p_{2\theta} + p_{2k} < \frac{1}{3}C^*_{\max},$$

since $l_\theta \le p_{2b}$ due to (3.3). Together with (3.7) and (3.10), we have $C^G_{\max} = P_1(\mathfrak{J}) + I_1 < \frac{5}{3}C^*_{\max}$.

Case 1.2.3.3 No operation finishes before $C_{2\theta}+l_\theta$ on machine M_1 and no operation finishes before $S_{2\theta}$ on machine M_2.

In this case, remember that there is idle time before $C_{2\theta}+l_\theta$ on machine M_1, and no job is released before r_θ, for otherwise there would be an operation finished either on machine M_1 before $C_{2\theta}+l_\theta$, or on machine M_2 before $S_{2\theta}$.

Hence we have that

$$r_\theta + p_1(\mathfrak{J}) < C^*_{\max}.$$

Together with the maximal idle time on machine M_1 after r_θ, we have

$$C^G_{\max} \le r_\theta + p_1(\mathfrak{J}) + p_{2\theta} + l_\theta + l_k + p_{2k}.$$

Using the inequalities (3.8) and (3.9), we get $C^G_{\max} < \frac{5}{3}C^*_{\max}$. This completes the case with only a single job in $\mathcal{X}$.

Case 2. We now consider the case that there are two jobs in $\mathcal{X}$, say, J_u and J_v. Assume without loss of generality that J_u precedes J_v on M_1. First of all, note that no other job can be executed between O_{1u} and O_{1v}. To see this, suppose there is some other job, say, J_c, executed between them. Since the time lags are small, see inequality (3.3), and since $r_v < T$, and $C_{1u}+l_u \ge T$, job J_c would delay the start of job J_v on machine M_1 to a time later than T. In such a case, however, the greedy algorithm would have scheduled J_v on machine M_2 such that

$S_{2v} < T$ and $C_{2v} \geq T$, which contradicts the property that M_2 is busy in the time interval $[T, C_{\max}^G]$ but idle immediately before time T.

Case 2.1 $\mathcal{Y} \neq \emptyset$.

In this case, if we have $p_{2u} > \frac{1}{3}C_{\max}^*$, we get $r_u + p_{1u} + l_u < \frac{2}{3}C_{\max}^*$. Note that the idle time in machine M_2 is bounded by $I_2 \leq r_u + p_{1u} + l_u$, since M_2 is busy in the period $[r_u, S_{1u}]$ and after $C_{1u} + l_u$. This immediately gives $C_{\max}^G < \frac{5}{3}C_{\max}^*$. Therefore, we consider

$$p_{2u} \leq \frac{1}{3}C_{\max}^* \tag{3.12}$$

in the remainder of this case.

Accordingly, we have that

$$C_{\max}^G = T + p_{2u} + p_{2v} + p_2(\mathcal{Y}). \tag{3.13}$$

If $p_{2u} + p_{2v} \leq \frac{2}{3}C_{\max}^*$, we obtain that

$$C_{\max}^G = T + p_{2u} + p_{2v} + p_2(\mathcal{Y}) \leq r(\mathcal{Y}) + p_2(\mathcal{Y}) + \frac{2}{3}C_{\max}^* \leq \frac{5}{3}C_{\max}^*,$$

and we are done. Hence, we assume $p_{2u} + p_{2v} > \frac{2}{3}C_{\max}^*$ in this case. This implies that

$$(S_{1v} - r_v) + p_2(\mathcal{Y}) < \frac{1}{3}C_{\max}^*, \tag{3.14}$$

since both machines must be busy in the interval $[r_v, S_{1v}]$ and $S_{1v} < T$.

In addition, we have $S_{1v} + p_{1v} + l_v \geq T$, since otherwise operation O_{2v} would have started before time T. Substituting this in (3.13), we obtain

$$C_{\max}^G \leq S_{1v} + p_{1v} + l_v + p_{2u} + p_{2v} + p_2(\mathcal{Y}).$$

Together with $r_v + p_{1v} + l_v + p_{2v} \leq C_{\max}^*$, (3.12) and (3.14), it holds that $C_{\max}^G < \frac{5}{3}C_{\max}^*$ and we are done.

Case 2.2 $\mathcal{Y} = \emptyset$.

In this case, it must be that $C_{1u} + l_u = T$ and $C_{1v} + l_v \leq C_{2u}$, for otherwise there would be idle time on M_2 between O_{2u} and O_{2v}, contradicting our definition of T. Hence, we write $C_{\max}^G$ as

$$C_{\max}^{G} = r_u + (S_{1u} - r_u) + p_{1u} + l_u + p_{2u} + p_{2v}, \text{ and} \tag{3.15}$$

$$C_{\max}^{G} = r_v + (S_{1v} - r_v) + p_{1v} + l_v + (C_{2u} - (C_{1v} + l_v)) + p_{2v}. \tag{3.16}$$

If $(S_{1u} - r_u) + p_{2v} \leq \frac{2}{3}C_{\max}^{*}$ or $(S_{1v} - r_v) + (C_{2u} - (C_{1v} + l_v)) \leq \frac{2}{3}C_{\max}^{*}$, then we conclude from (3.15) or (3.16) that $C_{\max}^{G} \leq \frac{5}{3}C_{\max}^{*}$.

So, it only remains to consider the situation that $(S_{1u} - r_u) + p_{2v} > \frac{2}{3}C_{\max}^{*}$ and $(S_{1v} - r_v) + (C_{2u} - (C_{1v} + l_v)) > \frac{2}{3}C_{\max}^{*}$.

These two inequalities imply that $r_v < S_{1u}$; otherwise, M_2 would have been busy during the disjoint periods $[r_u, S_{1u}]$, $[r_v, S_{1v}]$, $[S_{2v}, C_{2v}]$ and $[C_{1v} + l_v, C_{2u}]$, and hence $(S_{1u} - r_u) + p_{2v}$ and $(S_{1v} - r_v) + (C_{2u} - (C_{1v} + l_v))$ cannot each be greater than $\frac{2}{3}C_{\max}^{*}$ at the same time. This means that machine M_2 must be busy in the intervals $[r_u, S_{1u}]$ and $[S_{1u}, C_{1u}]$, for otherwise O_{2v} would have started on M_2 before time C_{1u}.

Let now $\mathcal{W} = \{J_j \mid C_{2j} > r_u\}$. We obtain that

$$r(\mathcal{W}) + (S_{1u} - r_u) + (C_{1u} - S_{1u}) + p_{2u} + p_{2v} \leq C_{\max}^{*},$$

since machine M_2 is busy during the periods $[r_u, S_{1u}]$ and $[S_{1u}, C_{1u}]$. Due to $(S_{1u} - r_u) + p_{2v} > \frac{2}{3}C_{\max}^{*}$, we get that

$$r(\mathcal{W}) + p_{1u} + p_{2u} < \frac{1}{3}C_{\max}^{*}. \tag{3.17}$$

Let now J_ω be a job with $r_\omega = r(\mathcal{W})$. We analyze three further subcases:

Case 2.2.1 $O_{2\omega}$ is scheduled before $O_{1\omega}$, or $O_{1\omega}$ is absent;

In this case, M_2 must be busy in the interval $[r_\omega, r_u]$. For I_2, the total idle time on M_2, this implies that $I_2 \leq r_\omega + l_u$, since the length of the idle time immediately before time T on machine M_2 must be smaller than or equal to l_u, as O_{1v} is started immediately after O_{1u} and cannot be started earlier. Accordingly, by use of (3.3) and (3.17), we obtain that

$$C_{\max}^{G} = p(\mathcal{J}) + I_2 \leq C_{\max}^{*} + r_\omega + l_u \leq C_{\max}^{*} + r_\omega + p_{1u} + p_{2u} < \frac{4}{3}C_{\max}^{*}.$$

Case 2.2.2 $O_{2\omega}$ is scheduled after $O_{1\omega}$ and $C_{1\omega} + l_\omega > r_v$;

In this case, I_1, the total idle time on machine M_1, is bounded from above by

$$I_1 \leq r_\omega + l_\omega + p_{2u} + p_{2v}, \tag{3.18}$$

since M_1 must be busy in the intervals $[r_\omega, C_{1\omega}]$ and $[r_v, C_{1v}]$. From $(S_{1v} - r_v) + (C_{2u} - (C_{1v} + l_v)) > \frac{2}{3}C^*_{\max}$ and $p_2(\mathfrak{J}) \leq C^*_{\max}$, we derive that $p_{2v} < \frac{1}{3}C^*_{\max}$. This inequality together with inequalities (3.3), (3.17) and (3.18) imply that $C^G_{\max} = p_1(\mathfrak{J}) + I_1 < \frac{5}{3}C^*_{\max}$.

Case 2.2.3 $O_{2\omega}$ is scheduled after $O_{1\omega}$ and $C_{1\omega} + l_\omega \leq r_v$.

In this case, I_1 is bounded from above by

$$I_1 \leq r_\omega + l_\omega + (r_v - (C_{1\omega} + l_\omega)) + p_{2u} + p_{2v}. \tag{3.19}$$

It is known that period $[C_{1\omega} + l_\omega, r_v]$ precedes period $[r_v, S_{1v}]$ and period $[C_{1v} + l_v, C_{2u}]$ precedes period $[S_{2v}, C_{2v}]$, and machine M_2 is fully occupied in these periods. From $(S_{1v} - r_v) + (C_{2u} - (C_{1v} + l_v)) > \frac{2}{3}C^*_{\max}$ and $p_2(\mathfrak{J}) \leq C^*_{\max}$, it follows that $p_{2v} + (r_v - (C_{1\omega} + l_\omega)) < \frac{1}{3}C^*_{\max}$. Using this inequality together with (3.3), (3.17) and (3.19), we obtain that $C^G_{\max} = p_1(\mathfrak{J}) + I_1 < \frac{5}{3}C^*_{\max}$.

This concludes the case-by-case analysis to prove the competitive ratio for the case with small time lags.

To conclude, we give an example to show that the bound is tight for non-delay and non-preemptive schedules. Consider the following instance of the problem. At time $t = 0$, jobs $J_1 = (p_{11} = 2, l_1 = 0, p_{21} = 0)$ and $J_2 = (p_{12} = 0, l_1 = 0, p_{22} = 2)$ arrive; at time $t = \epsilon$, job $J_3 = (p_{13} = 1, l_1 = 1, p_{23} = 1)$ arrives, with some $0 < \epsilon \leq 1/2$. Any non-delay and non-preemptive algorithm gives a schedule of length 5, whereas the optimal makespan is $3 + \epsilon$. Therefore, a ratio of $\frac{5}{3+\epsilon}$ is best possible for any non-delay and non-preemptive algorithm. As ϵ could be arbitrarily small, the ratio could be arbitrarily close to 5/3. $\square$

3.4 The worst-case performance of on-line delay algorithms

Does a delay algorithm, that is, an algorithm that is willing to keep a machine deliberately idle while an operation is available for processing, have a better competitive ratio than the greedy algorithm, which is non-delay? After all, allowing tactical delays may lead to better schedules (Kanet (1986); Kanet and Sridharan (2000)). The answer is "no" for the non-clairvoyant version of the problem, both for the general case and for the case with small time lags, as we will prove in the next theorem.

Theorem 6 *For the non-clairvoyant version of the problem, no on-line delay algorithm has a better competitive ratio than an on-line greedy algorithm.*

Proof. We use an adversary strategy against which any on-line delay algorithm must perform poorly. Consider the following instance of the problem. At time $t = 0$, jobs $J_1 = (p_{11}, l_1 = 0, p_{21} = 0)$ and $J_2 = (p_{12} = 0, l_1 = 0, p_{22})$ arrive. Let algorithm H be any delay algorithm. Suppose algorithm H imposes deliberate delays S_1 and S_2 to schedule jobs J_1 and J_2, respectively. Without loss of generality, assume $S_2 \geq S_1 \geq 0$.

If $S_2 > 0$, let $n = 2$ and $p_{11} = p_{22} = S_2$. The minimum makespan is now S_2, but algorithm H will finish the two jobs at time $2S_2$. It has therefore a competitive ratio at least 2.

If $S_2 = 0$, let $n = 3$ and job J_3 arrives at time $t = \epsilon$. For the general case, we let $p_{11} = p_{22} = 1$ and $J_3 = (p_{13} = \epsilon, l_1 = 1 - 2\epsilon, p_{23} = \epsilon)$, whereas for the case with small time lags we let $p_{11} = p_{22} = 2$ and $J_3 = (p_{13} = 1, l_1 = 1, p_{23} = 1)$. Since these two instances boil down to those used in Theorem 4 and Theorem 5 to prove the competitive ratios 2 and 5/3 for the greedy algorithm, algorithm H does not outperform the greedy algorithm. □

Accordingly, the greedy algorithm is a best possible algorithm for the non-clairvoyant variant of the problem.

For the clairvoyant problem, we prove that any on-line delay algorithm has a competitive ratio of at least $\sqrt{2} \approx 1.414$. However, this results leaves open the question whether an on-line delay algorithm has a better competitive ratio than the greedy algorithm.

Theorem 7 *For clairvoyant version of the problem, no on-line delay algorithm has a competitive ratio better than $\sqrt{2}$.*

Proof. Consider the following instance of the problem. At time $t = 0$, jobs $J_1 = (p_{11} = 1, l_1 = 0, p_{21} = 0)$ and $J_2 = (p_{12} = 0, l_2 = 0, p_{22} = 1)$ arrive. Let algorithm H be any delay algorithm, and suppose algorithm H imposes deliberate delays S_1 and S_2 for jobs J_1 and J_2, respectively. Without loss of generality, we assume that $1 > S_2 \geq S_1 \geq 0$. Suppose that either no further jobs arrive, or job $J_3 = (p_{13} = \epsilon, l_3 = 1 - 2\epsilon, p_{23} = \epsilon)$ arrives at time $S_2 + \epsilon$, $0 < \epsilon \leq 1/2$. In the former case, the schedule given by H has $C_{\max} = 1 + S_2$, while in the latter case we have $C_{\max} = 2 + S_1$. The optimal makespans are 1 and $1 + S_2 + 2\epsilon$, respectively. So, the competitive ratio of algorithm H is greater than or equal to $\max\{\frac{1+S_2}{1}, \frac{2+S_1}{1+S_2+2\epsilon}\}$. Accordingly, the minimum competitive ratio is equal to $\sqrt{2}$, which is achieved for $S_1 = 0$ and $S_2 = \sqrt{2} - 1$. $\square$

3.5 Conclusions

We have proven that the greedy algorithm for the on-line two-machine open shop scheduling problem with time lags has a tight competitive ratio of 2; this ratio is 5/3 in the case of small time lags. For the non-clairvoyant versions of these problems, the greedy algorithm is a best possible algorithm. For the clairvoyant versions, the greedy algorithm is a best possible non-delay algorithm; however, since we could only prove a lower bound of $\sqrt{2} \approx 1.414$ on the competitive ratio in general, a delay algorithm may have a better competitive ratio.

Other interesting avenues for further research are the problems with asymmetric time lags and problems with other types of small time lags, such as $l_k \leq \min\{p_{1k}, p_{2k}\}$ for all jobs J_k.

Part III

Off-Line Algorithms

4

Parallel Flow Shop[1]

Eventually everything connects - people, ideas, objects. The quality of the connections is the key to quality per se.

- Charles Eames (1907-1978)

We consider the NP-hard problem of scheduling n jobs in m two-stage parallel flow shops so as to minimize the makespan. This problem decomposes into two subproblems; assigning the jobs to parallel flow shops: and scheduling the jobs assigned to the same flow shop by use of Johnson's rule. For $m = 2$, we present a $\frac{3}{2}$-approximation algorithm, and for $m = 3$, we present a $1\frac{5}{7}$-approximation algorithm. Both these algorithms run in $O(n \log n)$ time. These are the first approximation algorithms with fixed worst-case performance guarantees for the parallel flow shop problem.

4.1 Introduction

Consider the problem of scheduling a set of n independent jobs $\mathcal{J} = \{J_1, \ldots, J_n\}$, in which each job J_j consists of a chain of two operations (O_{1j}, O_{2j}) $(j = 1, \ldots, n)$, in a *hybrid flow shop*, also called a *flexible flow shop*, so as to minimize the length of the schedule, that is, the makespan. A hybrid flow shop is an extension of the

[1] This chapter is based on a paper that has been submitted to the *European Journal of Operational Research.*

classical flow shop, where there are m_1 identical machines M_{i1} $(i = 1, \dots, m_1)$ in stage 1 and m_2 identical machines M_{i2} $(i = 1, \dots, m_2)$ in stage 2. The first operation O_{1j} of any job J_j needs first be processed on one of the machines in stage 1 during an uninterrupted processing time $p_{1j} \geq 0$, and then the second operation O_{2j} needs to be processed on one of the machines in stage 2 during an uninterrupted processing time $p_{2j} \geq 0$.

The hybrid flow shop problem of minimizing makespan has been well studied; for an review of the literature, see Ruiz and Vazquez-Rodriguez (2010). Obviously, if $m_1 = m_2 = 1$, then the problem is polynomially solvable in $O(n \log n)$ time by Johnson's rule Johnson (1954). However, if $m_1 \geq 2$, or by symmetry $m_2 \geq 2$, the problem becomes strongly NP-hard (Hoogeveen et al (1996)). Many researchers have focused on the special case with a single machine in one stage (Chen (1995) and Gupta (1988), Gupta and Tunc (1991), Gupta et al (1997)). For a review of the literature for the hybrid flow shop problem with a single machine in one stage, see Linn and Zhang (1999) or Wang (2005). For the general case, Chen (1994) and Lee and Vairaktarakis (1994) present $O(n \log n)$-time heuristics with worst-case performance guarantee ratio $2 - 1/\max\{m_1, m_2\}$. If, for any instance of the problem, the makespan of the schedule generated by some heuristic does not exceed ρ times the optimal makespan, where ρ is a constant that is as small as possible, then ρ is the worst-case performance ratio of the heuristic. A heuristic with a worst-case performance ratio of ρ is called referred to as a ρ-approximation algorithm.

A hybrid flow shop is a manufacturing system that offers much flexibility, but as Vairaktarakis and Elhafsi (2000) point out, this superior performance comes at the expense of sophisticated material handling systems, like automated guided vehicles and automated transfer lines. As an alternative to the hybrid flow shop, Vairaktarakis and Elhafsi (2000) introduced the *parallel flowline design*, which is a flexible manufacturing environment with m identical parallel two-stage flow shops $F_1, \dots, F_m$, each consisting of a series of two machines M_{1i} and M_{2i} $(i = 1, \dots, m)$. Each job needs first to be assigned to one of the flow shops, and once assigned, it will stay there for both operations. See Figure 4.1 for a hybrid two-stage flow shop, where the arrows indicate the routes that the different jobs may follow, and Figure 4.2 for a parallel two-stage flow shop. In the remainder, we will refer to a parallel flowline design as a *parallel flow shop*.

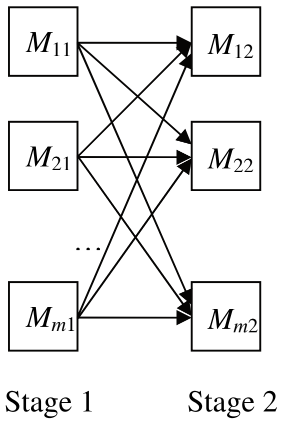

Fig. 4.1. A hybrid two-stage flow shop.

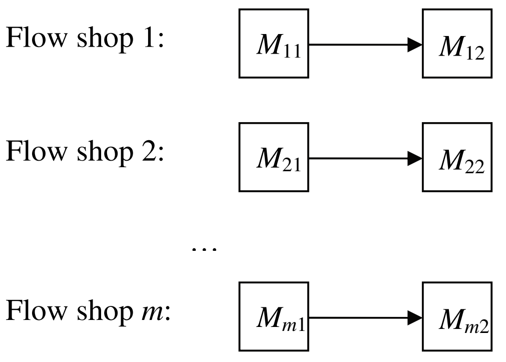

Fig. 4.2. A parallel two-stage flow shop.

The makespan parallel flow shop problem breaks down into two consecutive subproblems; first assigning each job to one of the m flow shops, and then scheduling the jobs in each flow shop so as to minimize the makespan. Whereas this second problem can obviously be solved in polynomial time by Johsnon's rule (Johnson (1954)), the first subproblem makes the problem NP-hard, as proved by Vairaktarakis and Elhafsi (2000), who also presented an $O(n \sum_{j=1}^{n}(p_{1j} + p_{2j})^3)$

time dynamic programming algorithm for its solution. Qi (2008) gave a faster algorithm, running in $O(n\sum_{j=1}^{n}(p_{1j}+p_{2j})^2)$ time.

Vairaktarakis and Elhafsi (2000) concluded empirically, on the basis of computational experiments with several heuristics for both problems, that the parallel flow shop entails only a minor loss in throughput performance in comparison with the hybrid flow shop; accordingly, it is an attractive alternative to the hybrid flow shop, with its complicated routings. Other heuristics for the parallel flow shop problem have been presented by Cao and Chen (2003) and Al-Salem (2004).

In contrast to the makespan hybrid flow shop problem, no approximation results for the makespan parallel flow shop are known. In this chapter, we present a 3/2-approximation algorithm for the parallel flow shop problem with $m = 2$ in Section 4.2. For $m = 3$, we present a $\frac{12}{7}$-approximation algorithm in Section 4.3. These results are the first polynomial-time algorithms with fixed worst-case ratios for the parallel flow shop problem.

Section 4.4 ends the chapter with some conclusions, where we point out that our algorithms and their worst-case performance guarantees also apply to the parallel flow shop problem where each job J_j after the completion of its first operation may be transferred to another flow shop for the processing of its second operation and where such a transfer requires a transportation time $l_j \geq 0$. This transportation time effectively introduces a minimum time lag between the completion time of the first operation and the start time of the second operation of a job. Note that if $l_j = 0$ for each J_j, then the parallel flow shop problem with transportation times boils down to the hybrid flow shop problem. For the hybrid flow shop problem with $m_1 = m_2 = 2$, our approximation algorithm has the same worst-case performance ratio as the one by Chen (1994) and Lee and Vairaktarakis (1994), but our algorithm is less complicated. At the other extreme, if $l_j = \infty$ for each J_j, then transfer between flow shops is effectively prohibited, and we have the original parallel flow shop problem.

4.2 A $\frac{3}{2}$-approximation algorithm for $m = 2$

In the remainder of the chapter, we assume that the job set $\mathcal{J} = \{J_1, \ldots, J_n\}$ has been re-indexed according to Johnson's rule; that is, for any pair of jobs (J_i, J_j) we have that $i < j$ if and only if

$$\min\{p_{1i}, p_{2j}\} \leq \min\{p_{1j}, p_{2i}\}.$$

For any instance of the m parallel two-stage flow shop problem, we refer to the Johnsonian schedule σ as the schedule that is obtained by assigning all the jobs to the first flow shop F_1 and processing them in order of Johnson's rule. $C_{\max}(\mathcal{J})$ denotes the makespan of the Johnsonian schedule for any job set $\mathcal{J} = \{J_1, \ldots, J_n\}$, whereas S_{ij} and C_{ij} denote the completion and the start times of the operations O_{ij} in the Johnsonian schedule, respectively, for $i = 1, 2; j = 1, \ldots, n$.

Lemma 6, which goes with no proof, specifies a simple lower bound on the minimum makespan $C^*_{\max}$ for the m parallel two-stage flow shop problem.

Lemma 6 *We have that*

$$C^*_{\max} \geq \max\{\frac{1}{m}\sum_{j=1}^{n} p_{1j}, \frac{1}{m}\sum_{j=1}^{n} p_{2j}, \frac{1}{m}C_{\max}(\mathcal{J}), \max_{1 \leq j \leq n}\{p_{1j} + p_{2j}\}\}. \tag{4.1}$$

Roughly speaking, the core idea for the 3/2-algorithm is to judiciously cut a Johnsonian schedule σ for $\mathcal{J}$ into two parts. The first part is scheduled on F_1, the second part on F_2. Both parts are scheduled according to Johnson's rule in order to minimize the makespan. The key question of course is where to cut the schedule so as to guarantee the 3/2 performance ratio.

Let now $T_1 = \frac{1}{4}C_{\max}(\mathcal{J})$ and $T_2 = \frac{3}{4}C_{\max}(\mathcal{J})$. Initially, we try to cut the Johnsonian schedule σ at time T_2. We have then the following lemma.

Lemma 7 *If there exists no job J_h with $S_{2h} \leq T_2 \leq C_{2h}$, then let $\mathcal{J}^1 = \{J_1, \ldots, J_{k-1}\}$ and $\mathcal{J}^2 = \{J_k, \ldots, J_n\}$ with J_k such that $S_{1k} \leq T_2 \leq C_{1k}$. We then have that*

$$\max\{C_{\max}(\mathcal{J}^1), C_{\max}(\mathcal{J}^2)\} \leq \frac{3}{2}C^*_{\max}.$$

PROOF. See Figure 4.3 for an illustration of how the two job sets are formed if there is no job J_h such that $S_{2h} \leq T_2 \leq C_{2h}$. By visual inspection of Figure 4.3 and by use of (1), it follows that

$$C_{\max}(\mathcal{J}^1) \leq T_2 = \frac{3}{4}C_{\max}(\mathcal{J}) \leq \frac{3}{2}C^*_{\max}, \text{ and}$$

$$C_{\max}(\mathcal{J}^2) \leq C_{\max}(\mathcal{J}) - T_2 + p_{1k} \leq \frac{1}{4}C_{\max}(\mathcal{J}) + p_{1k} \leq \frac{3}{2}C^*_{\max}. \qquad \square$$

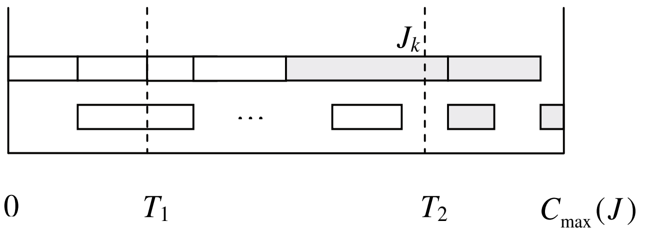

Fig. 4.3. Cutting the Johnsonian schedule as prescribed in Lemma 7.

The implication of Lemma 6 is that if there is no job J_h with $S_{2h} \le T_2 \le C_{2h}$, then we have indeed constructed a schedule with makespan no more than 3/2 times the optimal makespan and we are done. Accordingly, we know need to investigate the case where there such a job J_h does exist. We then have the following result.

Lemma 8 *If there exists a job J_h with $S_{2h} \le T_2 \le C_{2h}$ and if $S_{1h} \ge T_1$ or $C_{1h} = S_{2h}$, then let $\mathcal{J}^1 = \{J_1, \ldots, J_{h-1}\}$ and $\mathcal{J}^2 = \{J_h, \ldots, J_n\}$. It then holds that*

$$\max\{C_{\max}(\mathcal{J}^1), C_{\max}(\mathcal{J}^2)\} \le \frac{3}{2} C^*_{\max}.$$

Proof. Refer to Figure 4.4 for an illustration. Since $S_{2h} \le T_2$, job J_{h-1} is finished before T_2. We have therefore that

$$C_{\max}(\mathcal{J}^1) < T_2 \le \frac{3}{2} C^*_{\max}.$$

If $S_{1h} \ge T_1$, we have that

$$C_{\max}(\mathcal{J}^2) \le C_{\max}(\mathcal{J}) - T_1 = T_2 \le \frac{3}{2} C^*_{\max}.$$

If $C_{1h} = S_{2h}$, then

$$C_{\max}(\mathcal{J}^2) \le p_{1h} + p_{2h} + (C_{\max}(\mathcal{J}) - T_2) \le \frac{3}{2} C^*_{\max}.$$

□

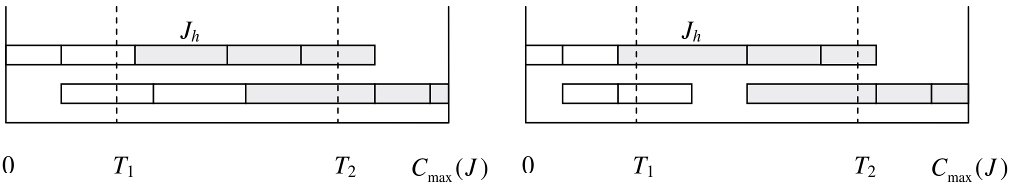

Fig. 4.4. Cutting the Johnsonian schedule as prescribed in Lemma 8.

Lemmata 7 and 8 do not cover the case where there exists a job J_h with $S_{2h} \leq T_2 \leq C_{2h}$, $S_{1h} < T_1$ and $C_{1h} < S_{2h}$. To analyze this case, we transform the Johnsonian schedule σ into the schedule σ' by delaying all operations as much as possible without changing the makespan. Hence, σ' has makespan $C_{\max}(\mathcal{J})$, has no idle time between any two operations on machine M_2, and all jobs are sequenced in order of Johnson's rule. We refer to σ' as the *delayed* Johnsonian schedule. Let now S'_{ij} and C'_{ij} denote the start and completion times of O_{ij} in σ'.

For σ', we have the following result.

Lemma 9 *If $S'_{1h} \geq T_1$ or $C'_{1h} = S'_{2h}$, then let $\mathcal{J}^1 = \{J_1, \ldots, J_{h-1}\}$, $\mathcal{J}^2 = \{J_h, \ldots, J_n\}$. It then holds that*

$$\max\{C_{\max}(\mathcal{J}^1), C_{\max}(\mathcal{J}^2)\} \leq \frac{3}{2} C^*_{\max}.$$

Proof. In this case, there is a job J_h with $S_{2h} \leq T_2 \leq C_{2h}$, therefore we have

$$C_{\max}(\mathcal{J}^1) = C_{2(h-1)} \leq S_{2h} \leq T_2 \leq \frac{3}{2} C^*_{\max}.$$

If $S'_{1h} \geq T_1 = \frac{1}{4} C_{\max}(\mathcal{J})$, then

$$C_{\max}(\mathcal{J}^2) \leq C_{\max}(\mathcal{J}) - S'_{1h} \leq \frac{3}{4} C_{\max}(\mathcal{J}) = \frac{3}{2} C^*_{\max}.$$

This case is illustrated in Figure 4.5, which shows both σ and σ'.

If $S'_{1h} < T_1$ and we have $C'_{1h} = S'_{2h}$, then

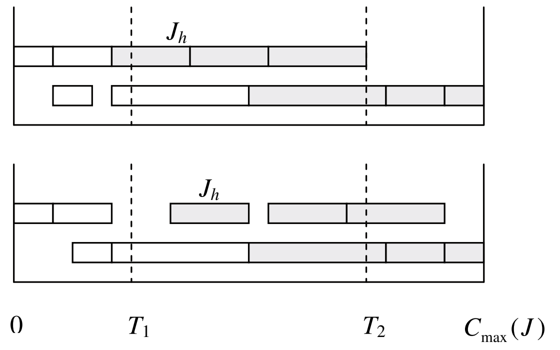

Fig. 4.5. Cutting the delayed Johnsonian schedule as prescribed in Lemma 9 if $S'_{1h} \geq T_1$. The top schedule is the Johnsonian schedule σ, the bottom schedule is the delayed Johnsonian schedule σ'.

$$C_{\max}(\mathcal{J}^2) \leq p_{1h} + p_{2h} + (C_{\max}(\mathcal{J}) - C_{2h}) \leq C^*_{\max} + \frac{1}{4}C_{\max}(\mathcal{J}) = \frac{3}{2}C^*_{\max}.$$

This case is illustrated by Figure 4.6. $\square$

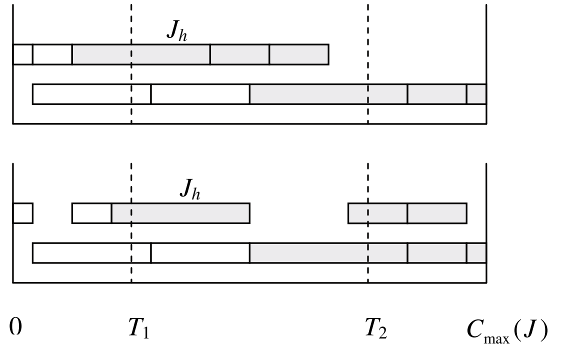

Fig. 4.6. Cutting the delayed Johnsonian schedule as prescribed in Lemma 9 if $S'_{1h} < T_1$. The top schedule is the Johnsonian schedule σ, the bottom schedule is delayed Johnsonian schedule σ'.

We have dealt now with many different subcases. The only case left to consider is the one with a job J_h with $S_{2h} \leq T_2 \leq C_{2h}$, $S_{1h} < T_1$, $C_{1h} < S_{2h}$, $S'_{1h} < T_1$ and $C'_{1h} < S'_{2h}$. See Figure 4.7 for an illustration of this case. In what follows, we will focus on this case.

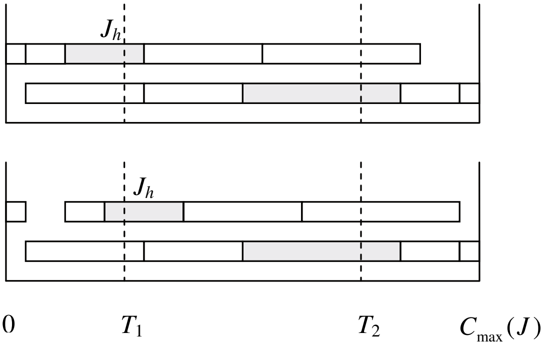

Fig. 4.7. Illustration of a Johnsonian schedule σ (the top schedule) and a delayed Johnsonian schedule σ' (the bottom schedule) for a job J_h with $S_{2h} \leq T_2 \leq C_{2h}$, $S_{1h} < T_1$, $C_{1h} < S_{2h}$, $S'_{1h} < T_1$ and $C'_{1h} < S'_{2h}$.

We then have the following lemma.

Lemma 10 *If there is a job J_h with $S_{2h} \leq T_2 \leq C_{2h}$, $S_{1h} < T_1$, $C_{1h} < S_{2h}$, $S'_{1h} < T_1$ and $C'_{1h} < S'_{2h}$, then machine M_2 is completely busy during the period $[T_1, T_2]$ in schedule σ and machine M_1 is completely busy during the period $[T_1, T_2]$ in schedule σ'.*

Proof. If in schedule σ machine M_2 would not have been busy during the interval $[T_1, T_2]$, then operation O_{2h} could have been started earlier. Similarly, if M_1 would not have been busy during the interval $[T_1, T_2]$ in schedule σ', then operation O_{1h} could have been started later. $\square$

We now separate all n jobs into two subsets $\mathcal{S}^1$ and $\mathcal{S}^2$ with $\mathcal{S}^1 = \{J_j | p_{1j} \leq p_{2j}, j = 1, \ldots, n\}$ and $\mathcal{S}^2 = \{J_j | p_{1j} > p_{2j}, j = 1, \ldots, n\}$. Since all jobs have been indexed in order of Johnson's rule, we can represent these two sets alternatively as $\mathcal{S}^1 = \{J_1, \ldots, J_u\}$ and $\mathcal{S}^2 = \{J_v, \ldots, J_n\}$ with $v = u + 1$. We branch into two cases: $\sum_{j=v}^{n} p_{1j} \geq T_1$; and $\sum_{j=1}^{u} p_{2j} \geq T_1$. Since these two cases are symmetrical, we analyze only the case with $\sum_{j=v}^{n} p_{1j} \geq T_1$.

In this case, we need to find a job J_e with $e \geq v$ such that $\sum_{j=v}^{e-1} p_{1j} < T_1 \leq \sum_{j=v}^{e} p_{1j}$ and a job J_d with $d < v$ such that $\sum_{j=d+1}^{e-1} p_{2j} < T_1 \leq \sum_{j=d}^{e-1} p_{2j}$. If $v = e$, we let $\sum_{j=v}^{e-1} p_{1j} = 0$. If $d = e - 1$, we let $\sum_{j=d+1}^{e-1} p_{2j} = 0$.

Lemma 11 *J_e and J_d exist.*

PROOF. Since $\sum_{j=v}^{n} p_{1j} \geq T_1$, job J_e must exist. To show that J_d exists, too, we branch into two cases. Since machine M_2 is busy in the period $[T_1, T_2]$ and $S_{1h} \leq T_2 \leq C_{2h}$, we have $\sum_{j=1}^{h} p_{2j} \geq T_2 - T_1 > T_1$. If $J_h \in \mathcal{S}^1$, then $v > h$, and we have that $\sum_{j=1}^{v-1} p_{2j} \geq \sum_{j=1}^{h} p_{2j} > T_1$. Hence, job J_d exists. If $J_h \in \mathcal{S}^2$, then $v \leq h$. And since $\sum_{j=v}^{e} p_{1j} \geq T_1$ and $\sum_{j=1}^{h-1} p_{1j} < T_1$ (because $S_{1h} < T_1$), we have that $e \geq h$. Since $C_{1h} < S_{2h}$, we have $\sum_{j=1}^{h-1} p_{2j} > p_{1h} > p_{2h}$. Together with $\sum_{j=1}^{h} p_{2j} \geq T_2 - T_1 = 2T_1$, we get $\sum_{j=1}^{h-1} p_{2j} > T_1$. Therefore, job J_d exists in this case also. For an illustration, see Figure 4.8. □

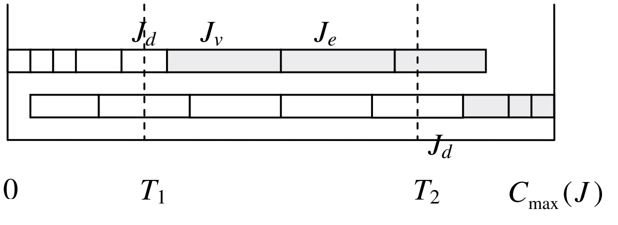

Fig. 4.8. Illustration of the jobs J_u,J_v,J_d,J_e, with $J_u = J_d = J_h$, as they occur in Lemma 11.

We now divide the case $\sum_{j=v}^{n} p_{1j} \geq T_1$ further into 5 different subcases and deal with these subcases in Lemmata 12 to 16.

Lemma 12 *If $\sum_{j=v}^{e} p_{2j} \geq T_1$, let $\mathcal{J}^1 = \{J_v, \ldots, J_e\}$ and $\mathcal{J}^2 = \{\mathcal{J} \backslash \mathcal{J}^1\}$. Then*

$$\max\{C_{\max}(\mathcal{J}^1), C_{\max}(\mathcal{J}^2)\} \leq \frac{3}{2} C_{\max}^*.$$

Proof. In this case, we have $\sum_{j=v}^{e-1} p_{1j} < T_1 \leq \sum_{j=v}^{e} p_{1j}$, $\sum_{j=v}^{e} p_{2j} \geq T_1$, $\mathcal{J}^1 = \{J_v, \ldots, J_e\}$ and $\mathcal{J}^2 = \{\mathcal{J} \backslash \mathcal{J}^1\}$. This can be illustrated by Figure 4.9.

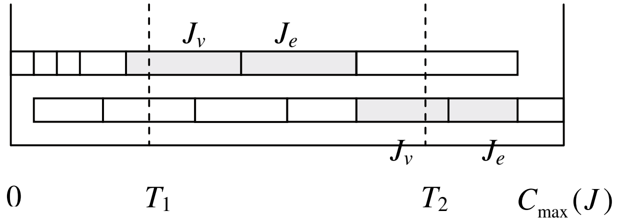

Fig. 4.9. Cutting the Johnsonian schedule as prescribed in Lemma 12.

Let J_w $(v \leq w \leq e)$ be the job for which $C_{\max}(\mathcal{J}^1) = \sum_{j=v}^{w} p_{1j} + \sum_{j=w}^{e} p_{2j}$. This implies that

$$\sum_{j=v}^{w} p_{1j} + \sum_{j=w}^{e} p_{2j} = \max_k \{\sum_{j=v}^{k} p_{1j} + \sum_{j=k}^{e} p_{2j}\},$$

and we refer to J_w as the *critical job* of schedule σ. Since $\mathcal{J}^1 \subset \mathcal{S}^2 = \{J_j | p_{1j} > p_{2j}\}$, we must have that $p_{2e} \leq p_{2w} < p_{1w}$ and $\sum_{j=v}^{w-1} p_{1j} + \sum_{j=w+1}^{e} p_{2j} \leq \sum_{j=v}^{w-1} p_{1j} + \sum_{j=w}^{e-1} p_{2j} < \sum_{j=v}^{e-1} p_{1j} < T_1$. It then holds that

$$C_{\max}(\mathcal{J}^1) = \sum_{j=v}^{w-1} p_{1j} + \sum_{j=w+1}^{e} p_{2j} + p_{1w} + p_{2w} < T_1 + C_{\max}^* \leq \frac{3}{2} C_{\max}^*.$$

Let σ^2 be the minimum makespan schedule for the jobs in $\mathcal{J}^2$, obtained by scheduling the jobs in order of Johnson's rule. For σ^2, let S''_{ij} denote the start time and C''_{ij} the completion time of operation O_{ij} $(i = 1, 2; j = 1, \ldots, v-1, e+1, \ldots, n)$. We have $S''_{ij} = S_{ij}$, $C''_{ij} = C_{ij}$, for $j = 1, \ldots, u$; and $S''_{ij} \leq S_{ij} - T_1$, $C''_{ij} \leq C_{ij} - T_1$, for $j = e+1, \ldots, n$, since job set $\mathcal{J}^1 = \{J_v, \ldots, J_e\}$ is not included in $\mathcal{J}^2$ and $\sum_{j=v}^{e} p_{1j} \geq \sum_{j=v}^{e} p_{2j} \geq T_1$. We have

$$C_{\max}(\mathcal{J}^2) = C''_{2n} \leq C_{\max}(\mathcal{J}) - T_1 = \frac{3}{2} C_{\max}^*.$$

□

Lemma 13 *If* $\sum_{j=d}^{v-1} p_{1j} \geq T_1$*, then let* $\mathcal{J}^1 = \{J_d, \ldots, J_{v-1}\}$ *and* $\mathcal{J}^2 = \{\mathcal{J} \backslash \mathcal{J}^1\}$*. We then have that*

$$\max\{C_{\max}(\mathcal{J}^1), C_{\max}(\mathcal{J}^2)\} \leq \frac{3}{2} C^*_{\max}.$$

Proof. This case is illustrated in Figure 4.10.

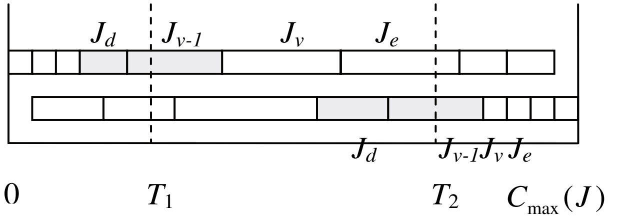

Fig. 4.10. Cutting the Johnsonian schedule as prescribed in Lemma 13.

Since $p_{1j} \leq p_{2j}$ for $j = d, \ldots, v-1$, we have $\sum_{j=d}^{v-1} p_{2j} \geq \sum_{j=d}^{v-1} p_{1j} \geq T_1$. By definition of job J_d, we get $\sum_{j=d+1}^{v-1} p_{2j} < T_1$. The case is then symmetric to the case specified in Lemma 12. □

In the remaining analysis, we therefore assume that $\sum_{j=d}^{v-1} p_{1j} < T_1$.

Lemma 14 *Assume* $\sum_{j=d}^{v} p_{1j} \geq T_1$ *and* $\sum_{j=d}^{v} p_{2j} \geq T_1$*. If* $v < e$*, then let* $\mathcal{J}^1 = \{J_d, \ldots, J_v\}$ *and* $\mathcal{J}^2 = \{J_1, \ldots, J_{d-1}, J_{v+1}, \ldots, J_n\}$*. If* $v = e$*, find a job* J_k *with* $\sum_{j=k+1}^{e} p_{2j} < T_1 \leq \sum_{j=k}^{e} p_{2j}$ *and* $d \leq k < e$*, and let* $\mathcal{J}^1 = \{J_k, \ldots, J_e\}$ *and* $\mathcal{J}^2 = \{\mathcal{J} \backslash \mathcal{J}^1\}$*. It then holds that*

$$\max\{C_{\max}(\mathcal{J}^1), C_{\max}(\mathcal{J}^2)\} \leq \frac{3}{2} C^*_{\max}.$$

Proof. First consider the case $v < e$, illustrated by Figure 4.11.

If $C_{\max}(\mathcal{J}^1) = \sum_{j=d}^{v} p_{1j} + p_{2v} = \sum_{j=d}^{v-1} p_{1j} + p_{1v} + p_{2v}$, we have $C_{\max}(\mathcal{J}^1) < T_1 + C^*_{\max} < \frac{3}{2} C^*_{\max}$. If $C_{\max}(\mathcal{J}^1) = p_{1d} + \sum_{j=d}^{v} p_{2j} = p_{1d} + p_{2d} + \sum_{j=d+1}^{v} p_{2j}$, we have $C_{\max}(\mathcal{J}^1) < C^*_{\max} + T_1 \leq \frac{3}{2} C^*_{\max}$. If $C_{\max}(\mathcal{J}^1) = \sum_{j=d}^{w} p_{1j} + \sum_{j=w}^{v} p_{2j}$

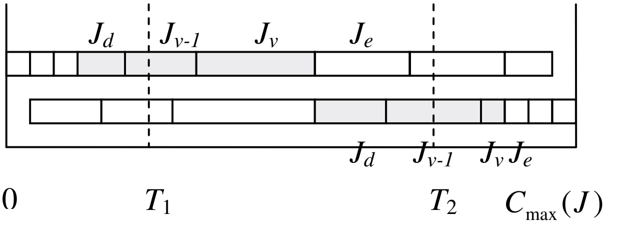

Fig. 4.11. Cutting the Johnsonian schedule as prescribed in Lemma 14. ($v < e$)

and $d < w < v$, where J_w is the critical job, we have $C_{\max}(\mathcal{J}^1) = \sum_{j=d}^{w} p_{1j} + \sum_{j=w}^{v} p_{2j} < T_1 + T_1 \leq C^*_{\max}$, since $\sum_{j=d}^{v-1} p_{1j} < T_1$ and $\sum_{j=d+1}^{v} p_{2j} < T_1$. The proof that $C_{\max}(\mathcal{J}^2) \leq \frac{3}{2} C^*_{\max}$ is similar to the proof of Lemma 12.

Now consider the case $v = e$, which is illustrated by Figure 4.12.

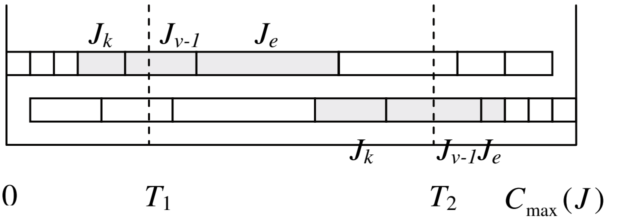

Fig. 4.12. Splitting of the Johnsonian schedule according to Lemma 14. ($v \geq e$)

Since $\sum_{j=d}^{e-1} p_{2j} \geq T_1$, job J_k exists. In this case, we have $\sum_{j=k}^{e-1} p_{1j} < T_1$, which follows from $\sum_{j=d}^{v-1} p_{1j} < T_1$ and $d \leq k < v = e$. Therefore, the proof is analogous to the one for $v < e$. □

In Lemma 14, we consider only the situation that $\sum_{j=d}^{v} p_{1j} \geq T_1$ and $\sum_{j=d}^{v} p_{2j} \geq T_1$. If $\sum_{j=d}^{v} p_{1j} \geq T_1$ and $\sum_{j=d}^{v} p_{2j} < T_1$, it must be that $v \leq e - 2$. Otherwise, if $v = e$ or $v = e - 1$, we would have that $\sum_{j=d}^{v} p_{2j} \geq T_1$. If the subcase in Lemma 14 is not satisfied, we have Lemmata 15 and 16 to solve remaining cases.

Lemma 15 *If* $\sum_{j=d}^{e-1} p_{1j} \geq T_1$*, let* $\mathcal{J}^1 = \{J_d, \ldots, J_{e-1}\}$ *and* $\mathcal{J}^2 = \{\mathcal{J} \backslash \mathcal{J}^1\}$*. It then holds that*

$$\max\{C_{\max}(\mathcal{J}^1), C_{\max}(\mathcal{J}^2)\} \leq \frac{3}{2} C^*_{\max}.$$

PROOF. If $v = e$ or $v = e - 1$, the result is correct due to Lemma 13 and Lemma 14. Hence, we need to consider only the case $v \leq e - 2$, which is illustrated by Figure 4.13.

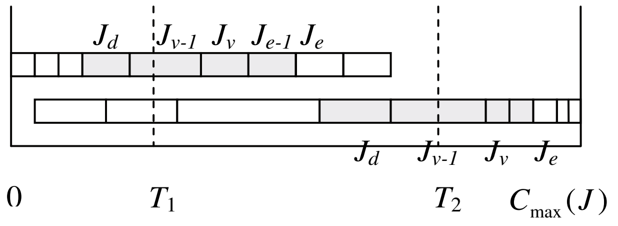

Fig. 4.13. Cutting the Johnsonian schedule as prescribed in Lemma 15.

Consider $C_{\max}(\mathcal{J}^1)$. Let J_w be the critical job in the minimum makespan schedule for $\mathcal{J}^1$. If $C_{\max}(\mathcal{J}^1) = \sum_{j=d}^{w} p_{1j} + \sum_{j=w}^{e-1} p_{2j}$ and $d \leq w < v$, we must have $p_{1d} \leq p_{1w} \leq p_{2w}$ and $\sum_{j=d}^{w-1} p_{1j} + \sum_{j=w+1}^{e-1} p_{2j} \leq \sum_{j=d+1}^{e-1} p_{2j} < T_1$. Then, $C_{\max}(\mathcal{J}^1) = \sum_{j=d}^{w-1} p_{1j} + \sum_{j=w+1}^{e-1} p_{2j} + p_{1w} + p_{2w} < T_1 + C^*_{\max} = \frac{3}{2} C^*_{\max}$.

If $C_{\max}(\mathcal{J}^1) = \sum_{j=d}^{w} p_{1j} + \sum_{j=w}^{e-1} p_{2j}$ and $v \leq w \leq e-1$, we have $\sum_{j=w+1}^{e-1} p_{2j} - \sum_{j=w+1}^{e-1} p_{1j} \leq 0$, since $\{J_w, \ldots, J_{e-1}\} \subset \mathcal{S}^2$. This implies that

$$\begin{aligned} C_{\max}(\mathcal{J}^1) &= \sum_{j=d}^{w} p_{1j} + \sum_{j=w}^{e-1} p_{2j} \\ &= \sum_{j=d}^{v-1} p_{1j} + \sum_{j=v}^{e-1} p_{1j} + p_{2w} + \sum_{j=w+1}^{e-1} p_{2j} - \sum_{j=w+1}^{e-1} p_{1j} \\ &\leq \sum_{j=d}^{v-1} p_{1j} + \sum_{j=v}^{e-1} p_{1j} + p_{2w}. \end{aligned}$$

If $\sum_{j=d}^{v-1} p_{1j} + p_{2w} \geq T_1$, we have $\sum_{j=d}^{v} p_{1j} \geq T_1$ and $\sum_{j=d}^{v} p_{2j} \geq T_1$, since $p_{2w} \leq p_{2v} < p_{1v}$ and $\sum_{j=d}^{v-1} p_{1j} \leq \sum_{j=d}^{v-1} p_{2j}$. We have solved this case in Lemma 14. If $\sum_{j=d}^{v-1} p_{1j} + p_{2w} < T_1$, we have that

$$C_{\max}(\mathcal{J}^1) \leq \sum_{j=d}^{v-1} p_{1j} + p_{2w} + \sum_{j=v}^{e-1} p_{1j} < T_1 + T_1 < C^*_{\max}.$$

Since we have $\sum_{j=d}^{e-1} p_{1j} \geq T_1$ and $\sum_{j=d}^{e-1} p_{2j} \geq T_1$ by definition, the proof of set $\mathcal{J}^2$ is analogous to that of Lemma 12. □

Lemma 16 *If $\sum_{j=d}^{e-1} p_{1j} < T_1$, let J_k with $d \leq k < v$ be such that $\sum_{k+1}^{e} p_{2j} < T_1 \leq \sum_{k}^{e} p_{2j}$, and define $\mathcal{J}^1 = \{J_k, \ldots, J_e\}$ and $\mathcal{J}^2 = \{\mathcal{J} \backslash \mathcal{J}^1\}$. It then holds that*

$$\max\{C_{\max}(\mathcal{J}^1), C_{\max}(\mathcal{J}^2)\} \leq \frac{3}{2} C^*_{\max}.$$

Proof. For a visualization of this case, see Figure 4.14.

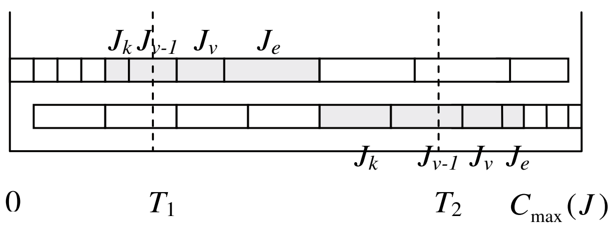

Fig. 4.14. Cutting the Johnsonian schedule as indicated in Lemma 16.

Since $\sum_{j=d}^{e-1} p_{2j} \geq T_1$, job J_k exists. If $C_{\max}(\mathcal{J}^1) = \sum_{j=k}^{w} p_{1j} + \sum_{j=w}^{e} p_{2j}$ and $k \leq w < v$, we must have $p_{1k} \leq p_{1w} \leq p_{2w}$ and $\sum_{j=k}^{w-1} p_{1j} + \sum_{j=w+1}^{e} p_{2j} \leq \sum_{j=k+1}^{e} p_{2j} < T_1$. Then, $C_{\max}(\mathcal{J}^1) = \sum_{j=d}^{w-1} p_{1j} + \sum_{j=w+1}^{e} p_{2j} + p_{1w} + p_{2w} < T_1 + C^*_{\max} = \frac{3}{2} C^*_{\max}$.

If $C_{\max}(\mathcal{J}^1) = \sum_{j=k}^{w} p_{1j} + \sum_{j=w}^{e} p_{2j}$ and $v \leq w \leq e$, we must have $p_{2e} \leq p_{2w} < p_{1w}$ and $\sum_{j=k}^{w-1} p_{1j} + \sum_{j=w+1}^{e} p_{2j} \leq \sum_{j=k}^{e-1} p_{1j} \leq \sum_{j=d}^{e-1} p_{1j} < T_1$. Then, $C_{\max}(\mathcal{J}^1) = \sum_{j=k}^{w-1} p_{1j} + \sum_{j=w+1}^{e} p_{2j} + p_{1w} + p_{2w} < T_1 + C^*_{\max} = \frac{3}{2} C^*_{\max}$.

Since we have $\sum_{j=k}^{e} p_{1j} \geq \sum_{j=v}^{e} p_{1j} \geq T_1$ and $\sum_{j=k}^{e} p_{2j} \geq T_1$, the proof of set $\mathcal{J}^2$ is analogous to that of Lemma 12. $\square$

We are now done with the analysis of the case for which $\sum_{j=v}^{n} p_{1j} \geq T_1$, and for which there exists a job J_h with $S_{2h} \leq T_2 \leq C_{2h}$, $S_{1h} < T_1$, $C_{1h} < S_{2h}$, $S'_{1h} < T_1$ and $C'_{1h} < S'_{2h}$. If $\sum_{j=1}^{u} p_{2j} \geq T_1$, the case is symmetrical to the case $\sum_{j=v}^{n} p_{1j} \geq T_1$, and we can cut the Johnsonian schedule similarly.

Lemma 17 *There is no case with both* $\sum_{j=v}^{n} p_{1j} < T_1$ *and* $\sum_{j=1}^{u} p_{2j} < T_1$.

PROOF. If $\sum_{j=v}^{n} p_{1j} < T_1$ and $\sum_{j=1}^{u} p_{2j} < T_1$, we get $\sum_{j=v}^{n} p_{2j} < T_1$ and $\sum_{j=1}^{u} p_{1j} < T_1$. Then we must have that $\sum_{j=v}^{n} p_{1j} + \sum_{j=1}^{u} p_{2j} + \sum_{j=v}^{n} p_{2j} + \sum_{j=1}^{u} p_{1j} < C_{\max}(\mathcal{J})$, which is a contradiction. $\square$

Using Lemmata 7-17, we have proved that we can split any set $\mathcal{J}$ into two disjoint subsets $\mathcal{J}^1$ and $\mathcal{J}^2$ and guarantee that the minimum makespan schedule for either subset has makespan no larger than $\frac{3}{2}C^*_{\max}$. The full details of the algorithm, referred to as Algorithm $SPLT1$, can be found in Appendix A.

Theorem 8 *Algorithm $SPLT1$ is a $\frac{3}{2}$-approximation for minimizing makespan on two parallel two-stage flow shops.* $\square$

4.3 A $1\frac{5}{7}$-approximation algorithm for $m = 3$

For $m = 3$, we essentially design a similar approach as for Algorithm $SPLT1$; we start by cutting the Johnsonian schedule σ into two parts. We will do this in such a way that the makespan of the first part is bounded from above by $\frac{4}{7}C_{\max}(\mathcal{J}) \leq 1\frac{5}{7}C^*_{\max}$ and the makespan of the second part is bounded from above by $\frac{16}{21}C_{\max}(\mathcal{J}) \leq 2\frac{2}{7}C^*_{\max}$; remember from Lemma 6 that $C_{\max}(\mathcal{J}) \leq 3C^*_{\max}$ if $m = 3$. We then use algorithm $SPLT1$ to cut the second part into two further parts and guarantee that both these further parts can be scheduled with a makespan smaller than $1\frac{5}{7}C^*_{\max}$.

As before, let the Johnsonian schedule be σ, and let S_{ij} and C_{ij} be the earliest start and completion times of operations O_{ij} for $i = 1, 2$ and $j = 1, \ldots, n$. We set $T_1 = \frac{5}{21}C_{\max}(\mathcal{J})$, $T_2 = \frac{16}{21}C_{\max}(\mathcal{J})$.

Lemma 18 *If there exists no job J_h with $S_{1h} \leq T_1 \leq C_{1h}$, then let $\mathcal{J}^1 = \{J_1, \ldots, J_k\}$ and $\mathcal{J}^2 = \{J_{k+1}, \ldots, J_n\}$ with J_k such that $S_{2k} \leq T_1 \leq C_{2k}$. We then have that*

$$C_{\max}(\mathcal{J}^1) \leq 1\frac{5}{7}C^*_{\max} \text{ and } C_{\max}(\mathcal{J}^2) \leq 2\frac{2}{7}C^*_{\max}.$$

PROOF. Since there is no job J_h with $S_{1h} \leq T_1 \leq C_{1h}$, machine M_1 is idle after T_1. Furthermore, there must exist a job J_k with $S_{2k} \leq T_1 \leq C_{2k}$, otherwise machine M_2 would be idle after T_1, too. We then let $\mathcal{J}^1 = \{J_1, \ldots, J_k\}$, and $\mathcal{J}^2 = \{J_{k+1}, \ldots, J_n\}$. This case is illustrated by Figure 4.15.

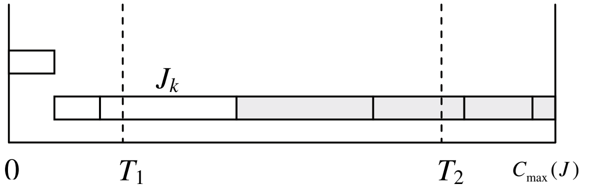

Fig. 4.15. Cutting the Johnsonian schedule as prescribed in Lemma 18.

Since $S_{2k} \leq T_1$, we have $C_{\max}(\mathcal{J}^1) = S_{2k} + p_{2k} \leq T_1 + C^*_{\max} = \frac{5}{21}C_{\max}(\mathcal{J}) + C^*_{\max} \leq 1\frac{5}{7}C^*_{\max}$. And since $C_{2k} \geq T_1$, we have $C_{\max}(\mathcal{J}^2) \leq C_{\max}(\mathcal{J}) - C_{2k} \leq \frac{16}{21}C_{\max}(\mathcal{J})$. $\square$

Lemma 19 *If there is a job J_h with $S_{1h} \leq T_1 \leq C_{1h}$ and $C_{2h} \leq \frac{4}{7}C_{\max}(\mathcal{J})$ or $C_{1h} = S_{2h}$, let $\mathcal{J}^1 = \{J_1, \ldots, J_h\}$, and $\mathcal{J}^2 = \{J_{h+1}, \ldots, J_n\}$. We then have that*

$$C_{\max}(\mathcal{J}^1) \leq 1\frac{5}{7}C^*_{\max} \text{ and } C_{\max}(\mathcal{J}^2) \leq 2\frac{2}{7}C^*_{\max}.$$

$\square$

PROOF. This case is visualized in Figure 4.16.The proof is similar to the one of Lemma 8. $\square$

Suppose now there is a job J_h with $S_{1h} \leq T_1 \leq C_{1h}$ for which $C_{2h} > \frac{4}{7}C_{\max}(\mathcal{J})$ and $C_{1h} < S_{2h}$. Then machine M_2 must be busy in the period $[T_1, \frac{4}{7}C_{\max}(\mathcal{J})]$, i.e.

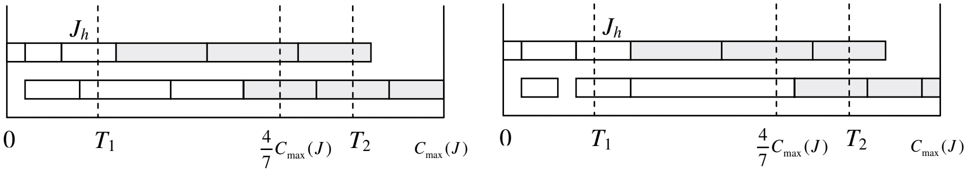

Fig. 4.16. Cutting the Johnsonian schedule as indicated in Lemma 19.

$\sum_{j=1}^{n} p_{2j} \geq \frac{4}{7}C_{\max}(\mathcal{J}) - \frac{5}{21}C_{\max}(\mathcal{J}) = \frac{1}{3}C_{\max}(\mathcal{J}) > T_1$. We now delay all operations O_{ij} in σ as much as possible within the makespan $C_{\max}(\mathcal{J})$. Let S'_{ij} and C'_{ij} denote the modified start and completion times of O_{ij} and let σ' denote the modified schedule.

Lemma 20 *In schedule σ', find a job J_t with $S'_{2t} \leq T_2 \leq C'_{2t}$. If $S'_{1t} \geq \frac{3}{7}C_{\max}(\mathcal{J})$ or $C'_{1t} = S'_{2t}$, let $\mathcal{J}^1 = \{J_t, \ldots, J_n\}$, and $\mathcal{J}^2 = \{J_1, \ldots, J_{t-1}\}$. We then have that*

$$C_{\max}(\mathcal{J}^1) \leq 1\frac{5}{7}C^*_{\max} \text{ and } C_{\max}(\mathcal{J}^2) \leq 2\frac{2}{7}C^*_{\max}.$$

Proof. If there is no such job J_t, we have $\sum_{j=1}^{n} p_{2j} < C_{\max}(\mathcal{J}) - T_2 = T_1$. Since we have assumed there is a job J_h for which $S_{1h} \leq T_1 \leq C_{1h}$ and $C_{2h} \leq \frac{4}{7}C_{\max}(\mathcal{J})$ or $C_{1h} = S_{2h}$, we have already addressed this case in Lemma 19.

Hence, we may suppose such a job J_t exists. This case is visualized in Figure 4.17.

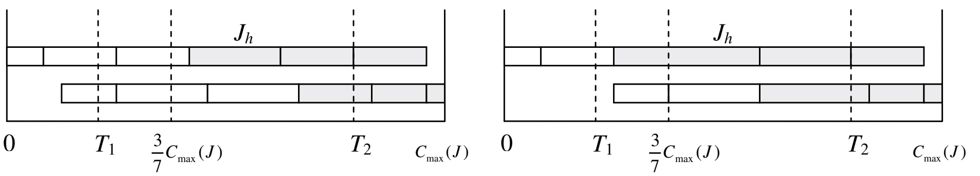

Fig. 4.17. Cutting the Johnsonian schedule as indicated in Lemma 20.

Since $S'_{2t} \leq T_2$, we have

$$C_{\max}(\mathcal{J}^2) = C_{2(t-1)} \leq S'_{2t} \leq T_2 \leq \frac{16}{21} C_{\max}(\mathcal{J}).$$

If $S'_{1t} \geq \frac{3}{7} C_{\max}(\mathcal{J})$, then $C_{\max}(\mathcal{J}^1) \leq C_{\max}(\mathcal{J}) - S'_{1t} \leq \frac{4}{7} C_{\max}(\mathcal{J}) = 1\frac{5}{7} C^*_{\max}$. If $S'_{1t} < \frac{3}{7} C_{\max}(\mathcal{J})$, then we have $C'_{1t} = S'_{2t}$, and hence $C_{\max}(\mathcal{J}^1) \leq p_{1t} + p_{2t} + (C_{\max}(\mathcal{J}) - C_{2t}) \leq C^*_{\max} + \frac{5}{21} C_{\max}(\mathcal{J}) = 1\frac{5}{7} C^*_{\max}$. □

Lemma 18 to Lemma 20 have solved many different cases of this problem. The one remaining case is where there exists a job J_t with $S'_{2t} \leq T_2 \leq C'_{2t}$, $S'_{1t} < \frac{3}{7} C_{\max}(\mathcal{J})$, $C'_{1t} < S'_{2t}$, and a job J_h with $S_{1h} \leq T_1 \leq C_{1h}$, $C_{2h} > \frac{4}{7} C_{\max}(\mathcal{J})$ and $C_{1h} < S_{2h}$. This case is illustrated in Figure 4.18.

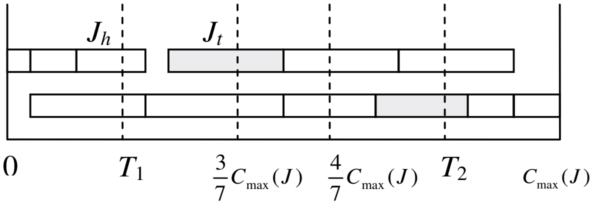

Fig. 4.18. The remaining case with jobs J_h and J_t.

In this remaining case, machine M_2 must be busy in the period $[T_1, \frac{4}{7} C_{\max}(\mathcal{J})]$ in schedule σ, for otherwise, operation O_{2h} could have been started earlier; in schedule σ', machine M_1 is busy in the period $[\frac{3}{7} C_{\max}(\mathcal{J}), T_2]$, for otherwise, operation O_{1t} could have been started later.

In what follows, we deal with the remaining case with jobs J_h and J_t only. We split the n jobs into two subsets $\mathcal{S}^1 = \{J_1, \ldots, J_u\} = \{J_j | p_{1j} \leq p_{2j}, j = 1, \ldots, n\}$ and $\mathcal{S}^2 = \{J_v, \ldots, J_n\} = \{J_j | p_{1j} > p_{2j}, j = 1, \ldots, n\}$. We then branch into two cases: the case $\sum_{j=v}^{n} p_{1j} \geq T_1$, and the case $\sum_{j=1}^{u} p_{2j} \geq T_1$. Since they are symmetrical, we analyze the first case only.

Since $\sum_{j=v}^{n} p_{1j} \geq T_1$, we can find a job J_e with $e \geq v$ such that $\sum_{j=v}^{e-1} p_{1j} < T_1 \leq \sum_{j=v}^{e} p_{1j}$. We have the following Lemma.

Lemma 21 *If* $\sum_{j=v}^{e} p_{2j} \geq T_1$, *then let* $\mathcal{J}^1 = \{J_v, \ldots, J_e\}$ *and* $\mathcal{J}^2 = \{\mathcal{J} \backslash \mathcal{J}^1\}$. *Then*

$$C_{\max}(\mathcal{J}^1) \leq 1\frac{5}{7}C^*_{\max} \text{ and } C_{\max}(\mathcal{J}^2) \leq 2\frac{2}{7}C^*_{\max}.$$

PROOF. This case is illustrated by Figure 4.19.

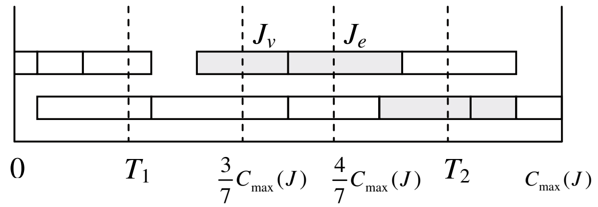

Fig. 4.19. Cutting the Johnsonian schedule as indicated in Lemma 21.

Let $C_{\max}(\mathcal{J}^1) = \sum_{j=v}^{w} p_{1j} + \sum_{j=w}^{e} p_{2j}$ and $v \leq w \leq e$. We must have $p_{2e} \leq p_{2w} < p_{1w}$ and $\sum_{j=v}^{w-1} p_{1j} + \sum_{j=w+1}^{e} p_{2j} < \sum_{j=v}^{e-1} p_{1j} < T_1$. Then, $C_{\max}(\mathcal{J}^1) = \sum_{j=v}^{w-1} p_{1j} + \sum_{j=w+1}^{e} p_{2j} + p_{1w} + p_{2w} < T_1 + C^*_{\max} = 1\frac{5}{7}C^*_{\max}$. The proof for $C_{\max}(\mathcal{J})$ is analogous to the proof of Lemma 12. □

If the condition in Lemma 21 is not satisfied, we need to find a job J_d with $d < v$ such that $\sum_{j=d-1}^{e-1} p_{2j} < T_1 \leq \sum_{j=d}^{e-1} p_{2j}$. If there is no such job J_d, we have the following result.

Lemma 22 *If there is no job J_d with $d < v$ such that $\sum_{j=d-1}^{e-1} p_{2j} < T_1 \leq \sum_{j=d}^{e-1} p_{2j}$, we find a job J_k with $\sum_{j=k+1}^{e} p_{2j} < T_1 \leq \sum_{j=k}^{e} p_{2j}$ and $1 \leq k < e$, and we let $\mathcal{J}^1 = \{J_k, \ldots, J_e\}$ and $\mathcal{J}^2 = \{\mathcal{J} \backslash \mathcal{J}^1\}$. We then have that*

$$C_{\max}(\mathcal{J}^1) \leq 1\frac{5}{7}C^*_{\max} \text{ and } C_{\max}(\mathcal{J}^2) \leq 2\frac{2}{7}C^*_{\max}.$$

PROOF. This case is visualized in Figure 4.20, where $k = v = 1$. In this case, we have $e \geq h$, since $\sum_{j=v}^{e} p_{1j} \geq T_1$ and $\sum_{j=1}^{h-1} p_{1j} \leq T_1$. Furthermore, we have $k < v$, for otherwise we would have $\sum_{j=v}^{e} p_{2j} \geq T_1$, which already has been covered by Lemma 21. With $C_{2h} > \frac{4}{7}C_{\max}(\mathcal{J})$ and machine M_2 being busy in the period $[\frac{5}{21}C_{\max}(\mathcal{J}), \frac{4}{7}C_{\max}(\mathcal{J})]$, we have $\sum_{j=1}^{e} p_{2j} > T_1$. Therefore job J_k exists. Since $\sum_{j=1}^{h} p_{2j} > T_1$ and $\sum_{j=1}^{e-1} p_{2j} < T_1$, we have $e - 1 < h$. Since also $e \geq$

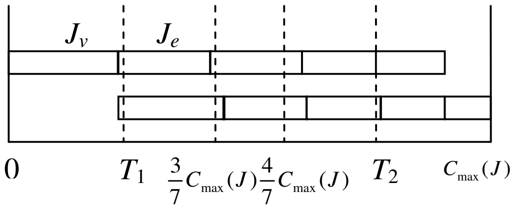

Fig. 4.20. Cutting the Johnsonian schedule as indicated in Lemma 22.

h, we must have that $e = h$. Then we have $\sum_{j=k}^{e-1} p_{1j} \leq \sum_{j=1}^{h-1} p_{1j} < T_1$. If $C_{\max}(\mathcal{J}^1) = \sum_{j=k}^{w} p_{1j} + \sum_{j=w}^{e} p_{2j}$ and $v \leq w \leq e$, we must have $p_{2e} \leq p_{2w} < p_{1w}$ and $\sum_{j=k}^{w-1} p_{1j} + \sum_{j=w+1}^{e} p_{2j} \leq \sum_{j=k}^{e-1} p_{1j} < T_1$. Then, $C_{\max}(\mathcal{J}^1) = \sum_{j=d}^{w-1} p_{1j} + \sum_{j=w+1}^{e} p_{2j} + p_{1w} + p_{2w} < T_1 + C^*_{\max} = 1\frac{5}{7}C^*_{\max}$. If $C_{\max}(\mathcal{J}^1) = \sum_{j=k}^{w} p_{1j} + \sum_{j=w}^{e} p_{2j}$ and $k \leq w < v$, we must have $p_{1k} \leq p_{1w} \leq p_{2w}$ and $\sum_{j=k}^{w-1} p_{1j} + \sum_{j=w+1}^{e} p_{2j} \leq \sum_{j=k+1}^{e} p_{2j} < T_1$. Then, $C_{\max}(\mathcal{J}^1) = \sum_{j=d}^{w-1} p_{1j} + \sum_{j=w+1}^{e} p_{2j} + p_{1w} + p_{2w} < T_1 + C^*_{\max} = 1\frac{5}{7}C^*_{\max}$. Because of $k < v$, we also have $\sum_{j=k}^{e} p_{1j} \geq T_1$. Since $\sum_{j=k}^{e} p_{2j} \geq T_1$, the proof of $C_{\max}(\mathcal{J}^2)$ is analogous to Lemma 12. □

If there exists a job J_e with $e \geq v$ such that $\sum_{j=v}^{e-1} p_{1j} < T_1 \leq \sum_{j=v}^{e} p_{1j}$ and a job J_d with $d < v$ such that $\sum_{j=d-1}^{e-1} p_{2j} < T_1 \leq \sum_{j=d}^{e-1} p_{2j}$, we have the following Lemmata 23 - 26. Their proofs are similar to those of Lemma 13 - 16.

Lemma 23 *If $\sum_{j=d}^{v-1} p_{1j} \geq T_1$, let $\mathcal{J}^1 = \{J_d, \ldots, J_{v-1}\}$ and $\mathcal{J}^2 = \{\mathcal{J} \backslash \mathcal{J}^1\}$. We then have that*

$$C_{\max}(\mathcal{J}^1) \leq 1\frac{5}{7}C^*_{\max} \text{ and } C_{\max}(\mathcal{J}^2) \leq 2\frac{2}{7}C^*_{\max}.$$

□

Lemma 24 *If $\sum_{j=d}^{v} p_{1j} \geq T_1$ and $\sum_{j=d}^{v} p_{2j} \geq T_1$, we have two cases. If $v < e$, let $\mathcal{J}^1 = \{J_d, \ldots, J_v\}$ and $\mathcal{J}^2 = \{\mathcal{J} \backslash \mathcal{J}^1\}$. If $v = e$, find a job J_k with $\sum_{j=k+1}^{e} p_{2j} < T_1 \leq \sum_{j=k}^{e} p_{2j}$ and $d \leq k < e$. Let $\mathcal{J}^1 = \{J_k, \ldots, J_e\}$ and $\mathcal{J}^2 = \{\mathcal{J} \backslash \mathcal{J}^1\}$. We then have that*

$$C_{\max}(\mathcal{J}^1) \leq 1\frac{5}{7}C^*_{\max} \text{ and } C_{\max}(\mathcal{J}^2) \leq 2\frac{2}{7}C^*_{\max}.$$

□

Lemma 25 *In the case of* $\sum_{j=d}^{e-1} p_{1j} \geq T_1$*, let* $\mathcal{J}^1 = \{J_d, \ldots, J_{e-1}\}$ *and* $\mathcal{J}^2 = \{\mathcal{J} \backslash \mathcal{J}^1\}$*. We then have that*

$$C_{\max}(\mathcal{J}^1) \leq 1\frac{5}{7} C^*_{\max} \text{ and } C_{\max}(\mathcal{J}^2) \leq 2\frac{2}{7} C^*_{\max}.$$

□

Lemma 26 *In the case of* $\sum_{j=d}^{e-1} p_{1j} < T_1$*, find a job* J_k *with* $d \leq k < v$ *such that* $\sum_{k+1}^{e} p_{2j} < T_1 \leq \sum_{k}^{e} p_{2j}$*,* $\mathcal{J}^1 = \{J_k, \ldots, J_e\}$ *and* $\mathcal{J}^2 = \{\mathcal{J} \backslash \mathcal{J}^1\}$*. We then have that*

$$C_{\max}(\mathcal{J}^1) \leq 1\frac{5}{7} C^*_{\max} \text{ and } C_{\max}(\mathcal{J}^2) \leq 2\frac{2}{7} C^*_{\max}.$$

□

Using Lemmata 21 - 26, we have solved the case $\sum_{j=v}^{n} p_{1j} \geq T_1$. The algorithm for the case $\sum_{j=1}^{u} p_{2j} \geq T_1$ is symmetrical. For the makespan parallel flow shop problem with $m = 3$, Lemma 17 still holds.

We have now developed an approximation algorithm, referred to as Algorithm $SPLT2$, for the parallel flow shop problem with $m = 3$ with worst-case performance guarantee $1\frac{5}{7}$ (see Appendix A for the exact step-wise description of Algorithm $SPLT2$). Algorithm $SPLT2$ gives two job sets $\mathcal{J}^1$ and $\mathcal{J}^2$, with $C_{\max}(\mathcal{J}^1) \leq 1\frac{5}{7} C^*_{\max}$ and $C_{\max}(\mathcal{J}^2) \leq 2\frac{2}{7} C^*_{\max}$. We can then apply Algorithm $SPLT1$ to the job set $\mathcal{J}^2$, which gives two further job sets for which have makespan bounded by $1\frac{5}{7} C^*_{\max}$. Finally, we have therefore the following result.

Theorem 9 *Algorithm* $SPLT2$ *is a* $1\frac{5}{7}$*-approximation for the problem of minimizing makespan in three parallel two-stage flow shops.* □

4.4 Conclusions

We have developed approximation algorithms with worst-case performance guarantees for scheduling jobs in a flexible manufacturing environment with two and three two-stage parallel flow shops. The key idea is to judiciously cut the Johnsonian schedule in two and three parts, respectively, and schedule each part in a different flow shop.

Our results apply also to the makespan parallel flow shop problem with transportation times, in which the operations of the same job can be performed in different flow shops and where transporting job J_j from one flow shop to another requires a transportation time $l_j \geq 0$ $(j = 1, \ldots, n)$. This is so, since in our algorithms transfer of jobs does not take place. If $l_j = 0$ for each j, then the parallel flow shop problem with transportation times reduces to the hybrid flow shop problem, and our approximation algorithm has the same worst-case performance guarantee as the algorithms by Chen (1994) and Lee and Vairaktarakis (1994) when $m_1 = m_2 = 2$.

Appendix A: Algorithms for splitting the Johnsonian schedule

Algorithm 1 *SPLT1*

Step 1. (Initialization)

Let $S_{11} = 0$, $C_{11} = S_{11} + p_{11}$, $S_{21} = C_{11}$, $C_{21} = S_{21} + p_{21}$.

For $j = 2$ *to* n, *do the following:*

$S_{1j} = C_{1(j-1)}$, $C_{1j} = S_{1j} + p_{1j}$, $S_{2j} = \max\{C_{1j}, C_{2(j-1)}\}$, $C_{2j} = S_{2j} + p_{2j}$.

Let $C_{\max}(\mathcal{J}) = C_{2n}$, $T_1 = \frac{1}{4} C_{\max}(\mathcal{J})$, $T_2 = \frac{3}{4} C_{\max}(\mathcal{J})$.

Step 2. Find the job $\max J_l$ *with* $S_{2h} \leq T_2 \leq C_{2h}$. *If job* J_h *does not exists, find the job* J_k *with* $S_{1k} \leq T_2 \leq C_{1k}$, *and let* $\mathcal{J}^1 = \{J_1, \ldots, J_{k-1}\}$, *and* $\mathcal{J}^2 = \{J_k, \ldots, J_n\}$, *stop; otherwise, go to Step 3 with* J_h.

Step 3.If $S_{1h} \geq T_1$ *or* $C_{1h} = S_{2h}$, *let* $\mathcal{J}^1 = \{J_1, \ldots, J_{h-1}\}$, *and* $\mathcal{J}^2 = \{J_h, \ldots, J_n\}$, *stop; otherwise, go to Step 4 with* J_h.

Step 4. Let $C'_{1n} = S_{2n}$ *and* $S'_{1n} = C'_{1n} - p_{1n}$.

For $j = (n-1)$ *to* 1, *perform the following computations:*

$C'_{1j} = \min\{S'_{1(j+1)}, S_{2j}\}$ *and* $S'_{1j} = C'_{1j} - p_{1j}$, *where* S'_{1j} *and* C'_{1j} *are the latest possible start and completion time of job* J_j *in machine* M_1.

Step 5. If $S'_{1h} \geq T_1$ *or* $C'_{1h} = S'_{2h}$, *let* $\mathcal{J}^1 = \{J_1, \ldots, J_{h-1}\}$, $\mathcal{J}^2 = \{J_h, \ldots, J_n\}$, *and stop; otherwise, go to Step 6.*

Step 6. In schedule σ, *find the job* J_u *with* $p_{1u} \leq p_{2u}$ *and* $p_{1(u+1)} > p_{2(u+1)}$, *and let* $v = u + 1$. *Therefore, in schedule* σ, *we have* $p_{1j} \leq p_{2j}$ *for* $j = 1, \ldots, u$ *and* $p_{1j} > p_{2j}$ *for* $j = v, \ldots, n$. *Then, we branch into the two cases.*

Case 1. $\sum_{j=v}^{n} p_{1j} \geq T_1$. *Find a job* J_e *with* $e \geq v$ *such that* $\sum_{j=v}^{e-1} p_{1j} < T_1 \leq \sum_{j=v}^{e} p_{1j}$ *and a job* J_d *with* $d < v$ *such that* $\sum_{j=d+1}^{e-1} p_{2j} < T_1 \leq \sum_{j=d}^{e-1} p_{2j}$. *We branch into five subcases.*

Subcase 1.1 $\sum_{j=v}^{e} p_{2j} \geq T_1$. *Let* $\mathfrak{J}^1 = \{J_v, \ldots, J_e\}$ *and* $\mathfrak{J}^2 = \{\mathfrak{J} \backslash \mathfrak{J}^1\}$. *Stop.*

Subcase 1.2 $\sum_{j=d}^{v-1} p_{1j} \geq T_1$. *Let* $\mathfrak{J}^1 = \{J_d, \ldots, J_{v-1}\}$ *and* $\mathfrak{J}^2 = \{\mathfrak{J} \backslash \mathfrak{J}^1\}$. *Stop.*

Subcase 1.3 $\sum_{j=d}^{v} p_{1j} \geq T_1$ *and* $\sum_{j=d}^{v} p_{2j} \geq T_1$. *If* $v < e$, *let* $\mathfrak{J}^1 = \{J_d, \ldots, J_v\}$ *and* $\mathfrak{J}^2 = \{\mathfrak{J} \backslash \mathfrak{J}^1\}$. *If* $v = e$, *find a job* J_k *with* $\sum_{j=k+1}^{e} p_{2j} < T_1 \leq \sum_{j=k}^{e} p_{2j}$ *and* $d \leq k < e$. *Let* $\mathfrak{J}^1 = \{J_k, \ldots, J_e\}$ *and* $\mathfrak{J}^2 = \{\mathfrak{J} \backslash \mathfrak{J}^1\}$. *Stop.*

Subcase 1.4 $\sum_{j=d}^{e-1} p_{1j} \geq T_1$. *Let* $\mathfrak{J}^1 = \{J_d, \ldots, J_{e-1}\}$ *and* $\mathfrak{J}^2 = \{\mathfrak{J} \backslash \mathfrak{J}^1\}$. *Stop.*

Subcase 1.5 $\sum_{j=d}^{e-1} p_{1j} < T_1$. *Find a job* J_k *with* $d \leq k < v$ *such that* $\sum_{k+1}^{e} p_{2j} < T_1 \leq \sum_{k}^{e} p_{2j}$, $\mathfrak{J}^1 = \{J_k, \ldots, J_e\}$ *and* $\mathfrak{J}^2 = \{\mathfrak{J} \backslash \mathfrak{J}^1\}$. *Stop.*

Case 2. $\sum_{j=1}^{u} p_{2j} \geq T_1$. *Find a job* J_d *with* $d \leq u$ *such that* $\sum_{j=d+1}^{u} p_{2j} < T_1 \leq \sum_{j=d}^{u} p_{2j}$ *and a job* J_e *with* $e > u$ *such that* $\sum_{j=d+1}^{e-1} p_{1j} < T_1 \leq \sum_{j=d+1}^{e} p_{1j}$. *We branch into five subcases.*

Subcase 2.1 $\sum_{j=d}^{u} p_{1j} \geq T_1$. *Let* $\mathfrak{J}^1 = \{J_d, \ldots, J_u\}$ *and* $\mathfrak{J}^2 = \{\mathfrak{J} \backslash \mathfrak{J}^1\}$. *Stop.*

Subcase 2.2 $\sum_{j=u+1}^{e} p_{2j} \geq T_1$. *Let* $\mathfrak{J}^1 = \{J_{u+1}, \ldots, J_e\}$ *and* $\mathfrak{J}^2 = \{\mathfrak{J} \backslash \mathfrak{J}^1\}$. *Stop.*

Subcase 2.3 $\sum_{j=u}^{e} p_{1j} \geq T_1$ *and* $\sum_{j=u}^{e} p_{2j} \geq T_1$. *If* $d < u$, *let* $\mathfrak{J}^1 = \{J_u, \ldots, J_e\}$ *and* $\mathfrak{J}^2 = \{\mathfrak{J} \backslash \mathfrak{J}^1\}$. *If* $d = u$, *find a job* J_k *with* $\sum_{j=d}^{k-1} p_{1j} < T_1 \leq \sum_{j=d}^{k} p_{1j}$ *and* $d < k \leq e$. *Let* $\mathfrak{J}^1 = \{J_d, \ldots, J_k\}$ *and* $\mathfrak{J}^2 = \{\mathfrak{J} \backslash \mathfrak{J}^1\}$. *Stop.*

Subcase 2.4 $\sum_{j=d+1}^{e} p_{2j} \geq T_1$. *Let* $\mathfrak{J}^1 = \{J_{d+1}, \ldots, J_e\}$ *and* $\mathfrak{J}^2 = \{\mathfrak{J} \backslash \mathfrak{J}^1\}$. *Stop.*

Subcase 2.5 $\sum_{j=d+1}^{e} p_{2j} < T_1$. *Find a job* J_k *with* $u < k \leq e$ *such that* $\sum_{d}^{k-1} p_{2j} < T_1 \leq \sum_{d}^{k} p_{2j}$, $\mathfrak{J}^1 = \{J_d, \ldots, J_k\}$ *and* $\mathfrak{J}^2 = \{\mathfrak{J} \backslash \mathfrak{J}^1\}$. *Stop.*

Algorithm 2 $SPLT2$

Step 1. (Initialization)

Let $S_{11} = 0$, $C_{11} = S_{11} + p_{11}$, $S_{21} = C_{11}$, $C_{21} = S_{21} + p_{21}$.

For $j = 2$ *to* n, *perform the following computations:*

$S_{1j} = C_{1(j-1)}$, $C_{1j} = S_{1j} + p_{1j}$, $S_{2j} = \max\{C_{1j}, C_{2(j-1)}\}$, $C_{2j} = S_{2j} + p_{2j}$.

Let $C_{\max}(\mathfrak{J}) = C_{2n}$, *and* $T_1 = \frac{5}{21} C_{\max}(\mathfrak{J})$, $T_2 = \frac{16}{21} C_{\max}(\mathfrak{J})$.

Step 2. Find a job J_h *with* $S_{1h} \leq T_1 \leq C_{1h}$. *If job* J_h *does not exist, find a job* J_k *with* $S_{2k} \leq T_1 \leq C_{2k}$. *Let* $\mathfrak{J}^1 = \{J_1, \ldots, J_k\}$, *and* $\mathfrak{J}^2 = \{J_{k+1}, \ldots, J_n\}$. *Stop; otherwise, go to Step 3 with job* J_h.

Step 3. For job J_h, *if* $C_{2h} \leq \frac{4}{7} C_{max}$ *or* $C_{1h} = S_{2h}$, *let* $\mathfrak{J}^1 = \{J_1, \ldots, J_h\}$, *and* $\mathfrak{J}^2 = \{J_{h+1}, \ldots, J_n\}$. *Stop; otherwise, go to Step 4.*

Step 4. Let $C'_{1n} = S_{2n}$ *and* $S'_{1n} = C'_{1n} - p_{1n}$.

For $j = (n-1)$ *to* 1, *perform the following computations:*

$C'_{1j} = \min\{S'_{1(j+1)}, S_{2j}\}$ *and* $S'_{1j} = C'_{1j} - p_{1j}$, *where* S'_{1j} *and* C'_{1j} *are the latest possible start and completion time of job* J_j *in machine* M_1.

Step 5. Find a job J_t *with* $S'_{2t} \le T_2 < C'_{2t}$. *If job* J_t *does not exists, we have solved this case in Step 3. If* $S'_{1t} \ge \frac{3}{7} C_{\max}(\mathcal{J})$ *or* $C'_{1t} = S'_{2t}$, *let* $\mathcal{J}^1 = \{J_t, \ldots, J_n\}$, *and* $\mathcal{J}^2 = \{J_1, \ldots, J_t\}$. *Stop; otherwise, go to Step 6.*

Step 6. In schedule σ, *find the job* J_u *with* $p_{1u} \le p_{2u}$ *and* $p_{1(u+1)} > p_{2(u+1)}$, *and let* $v = u + 1$. *Therefore, in schedule* σ, *we have* $p_{1j} \le p_{2j}$ *for* $j = 1, \ldots, u$; *and* $p_{1j} > p_{2j}$ *for* $j = v, \ldots, n$. *Then, we branch into the two cases.*

Case 1. $\sum_{j=v}^{n} p_{1j} \ge T_1$. *Find a job* J_e *with* $e \ge v$ *such that* $\sum_{j=v}^{e-1} p_{1j} < T_1 \le \sum_{j=v}^{e} p_{1j}$ *and a job* J_d *with* $d < v$ *such that* $\sum_{j=d+1}^{e-1} p_{2j} < T_1 \le \sum_{j=d}^{e-1} p_{2j}$. *We branch into six subcases.*

Subcase 1.1 $\sum_{j=v}^{e} p_{2j} \ge T_1$. *Let* $\mathcal{J}^1 = \{J_v, \ldots, J_e\}$ *and* $\mathcal{J}^2 = \{\mathcal{J} \backslash \mathcal{J}^1\}$. *Stop.*

Subcase 1.2 $\sum_{j=1}^{e-1} p_{2j} < T_1$. *Find a job* J_k *with* $\sum_{j=k+1}^{e} p_{2j} < T_1 \le \sum_{j=k}^{e} p_{2j}$ *and* $1 \le k < e$. *Let* $\mathcal{J}^1 = \{J_k, \ldots, J_e\}$ *and* $\mathcal{J}^2 = \{\mathcal{J} \backslash \mathcal{J}^1\}$. *Stop.*

Subcase 1.3 $\sum_{j=d}^{v-1} p_{1j} \ge T_1$. *Let* $\mathcal{J}^1 = \{J_d, \ldots, J_{v-1}\}$ *and* $\mathcal{J}^2 = \{\mathcal{J} \backslash \mathcal{J}^1\}$. *Stop.*

Subcase 1.4 $\sum_{j=d}^{v} p_{1j} \ge T_1$ *and* $\sum_{j=d}^{v} p_{2j} \ge T_1$. *If* $v < e$, *let* $\mathcal{J}^1 = \{J_d, \ldots, J_v\}$ *and* $\mathcal{J}^2 = \{\mathcal{J} \backslash \mathcal{J}^1\}$. *If* $v = e$, *find a job* J_k *with* $\sum_{j=k+1}^{e} p_{2j} < T_1 \le \sum_{j=k}^{e} p_{2j}$ *and* $d \le k < e$. *Let* $\mathcal{J}^1 = \{J_k, \ldots, J_e\}$ *and* $\mathcal{J}^2 = \{\mathcal{J} \backslash \mathcal{J}^1\}$. *Stop.*

Subcase 1.5 $\sum_{j=d}^{e-1} p_{1j} \ge T_1$. *Let* $\mathcal{J}^1 = \{J_d, \ldots, J_{e-1}\}$ *and* $\mathcal{J}^2 = \{\mathcal{J} \backslash \mathcal{J}^1\}$. *Stop.*

Subcase 1.6 $\sum_{j=d}^{e-1} p_{1j} < T_1$. *Find a job* J_k *with* $d \le k < v$ *such that* $\sum_{k+1}^{e} p_{2j} < T_1 \le \sum_{k}^{e} p_{2j}$, $\mathcal{J}^1 = \{J_k, \ldots, J_e\}$ *and* $\mathcal{J}^2 = \{\mathcal{J} \backslash \mathcal{J}^1\}$. *Stop.*

Case 2. $\sum_{j=1}^{u} p_{2j} \ge T_1$. *Find a job* J_d *with* $d \le u$ *such that* $\sum_{j=d+1}^{u} p_{2j} < T_1 \le \sum_{j=d}^{u} p_{2j}$ *and a job* J_e *with* $e > u$ *such that* $\sum_{j=d+1}^{e-1} p_{1j} < T_1 \le \sum_{j=d+1}^{e} p_{1j}$. *We branch into six subcases.*

Subcase 2.1 $\sum_{j=d}^{u} p_{1j} \ge T_1$. *Let* $\mathcal{J}^1 = \{J_d, \ldots, J_u\}$ *and* $\mathcal{J}^2 = \{\mathcal{J} \backslash \mathcal{J}^1\}$. *Stop.*

Subcase 2.2 $\sum_{j=d+1}^{n} p_{1j} < T_1$. *Find a job* J_k *with* $\sum_{j=d}^{k-1} p_{1j} < T_1 \le \sum_{j=d}^{k} p_{1j}$ *and* $u < k \le n$. *Let* $\mathcal{J}^1 = \{J_d, \ldots, J_k\}$ *and* $\mathcal{J}^2 = \{\mathcal{J} \backslash \mathcal{J}^1\}$. *Stop.*

Subcase 2.3 $\sum_{j=u+1}^{e} p_{2j} \ge T_1$. *Let* $\mathcal{J}^1 = \{J_{u+1}, \ldots, J_e\}$ *and* $\mathcal{J}^2 = \{\mathcal{J} \backslash \mathcal{J}^1\}$. *Stop.*

Subcase 2.4 $\sum_{j=u}^{e} p_{1j} \ge T_1$ *and* $\sum_{j=u}^{e} p_{2j} \ge T_1$. *If* $d < u$, *let* $\mathcal{J}^1 = \{J_u, \ldots, J_e\}$ *and* $\mathcal{J}^2 = \{\mathcal{J} \backslash \mathcal{J}^1\}$. *If* $d = u$, *find a job* J_k *with* $\sum_{j=d}^{k-1} p_{1j} < T_1 \le \sum_{j=d}^{k} p_{1j}$ *and* $d < k \le e$. *Let* $\mathcal{J}^1 = \{J_d, \ldots, J_k\}$ *and* $\mathcal{J}^2 = \{\mathcal{J} \backslash \mathcal{J}^1\}$. *Stop.*

Subcase 2.5 $\sum_{j=d+1}^{e} p_{2j} \ge T_1$. *Let* $\mathcal{J}^1 = \{J_{d+1}, \ldots, J_e\}$ *and* $\mathcal{J}^2 = \{\mathcal{J} \backslash \mathcal{J}^1\}$. *Stop.*

Subcase 2.6 $\sum_{j=d+1}^{e} p_{2j} < T_1$. *Find a job* J_k *with* $u < k \leq e$ *such that* $\sum_{d}^{k-1} p_{2j} < T_1 \leq \sum_{d}^{k} p_{2j}$, $\mathcal{J}^1 = \{J_d, \ldots, J_k\}$ *and* $\mathcal{J}^2 = \{\mathcal{J} \backslash \mathcal{J}^1\}$. *Stop.*

5

Polynomial-Time Approximation Schemes[1]

If, from a stick a foot long, you every day take the half of it, it will not be exhausted in a myriad ages.

- Chuang-Tzu (369BC-286BC)

In this chapter, we identify two classes of machine scheduling problems with time lags that possess Polynomial-Time Approximation Schemes (PTASs). These classes together, one for minimizing makespan and one for minimizing total completion time, include many well-studied time lag scheduling problems. The running times of these approximation schemes are polynomial in the number of jobs but exponential in the number of machines and the ratio between the largest time lag and the smallest positive operation time.

These classes constitute the first PTAS results for scheduling problems with time lags.

5.1 Introduction

5.1.1 Problem description

Machine scheduling problems with time lags can arise both in multi-stage and single-stage processing environments as long as the jobs to be processed consist

[1] This chapter is based on a paper that has been accepted for publication in the *Journal of Scheduling*. (Zhang and Van de Velde (2009))

of multiple operations. A time lag, after all, specifies a minimum delay between the execution of two consecutive operations of the same job. Time lags can model the transportation times between machines when the number of vehicles is not restrictive or when the jobs travel by themselves, like for example barges sailing between container terminals along the river bank and trucks traveling between pick up and delivery depots. Time lags can also model activities that require no limited resources, like for instance cooling down times. These practical justifications explain why time lags are sometimes also referred to in the literature as *transportation times* or *delays*.

We assume that there are m machines $M_1, \ldots, M_m$ available from time zero onwards for processing a set of n jobs $J_1, \ldots, J_n$, each consisting of o operations $(O_{1j}, \ldots, O_{oj})$. Every operation O_{ij} $(i = 1, \ldots, o; j = 1, \ldots, n)$ needs to be processed during an uninterrupted processing time $p_{ij} \geq 0$ on a dedicated machine $\mu_{ij} \in \{M_1, \ldots, M_m\}$ and the operations of the same job cannot be processed simultaneously. Each machine can handle only one operation at a time. Each job J_j $(j = 1, \ldots, n)$ may have a release time r_j before which no operation of J_j can be started.

We consider only the basic (and most common) time lag scheduling models with exactly one machine per stage. We consider three such multi-stage scheduling environments with $o = m$: an *open shop* problem, where the operations of a job can be processed in any order; a *job shop* problem, where the operations of every job J_j $(j = 1, \ldots, n)$ need to processed in the order $O_{1j} \to O_{2j} \to \ldots \to O_{oj}$; and a *flow shop*, which is essentially a job shop with the special condition that $\mu_{ij} = M_i$ for each operation O_{ij} $(i = 1, \ldots, m; j = 1, \ldots, n)$. Hence in a flow shop, all jobs pass through the machines in the same order $M_1 \to M_2 \to \ldots \to M_m$.

In the single-stage scheduling environment with time lags that we consider there is only a single machine M_1 available for processing jobs $J_1, \ldots, J_n$, each of which consists of a chain of two operations (O_{1j}, O_{2j}) that need to be processed in the order $O_{1j} \to O_{2j}$ for each j $(j = 1, \ldots, n)$. Hence, in this environment we have $m = 1$, $o = 2$, and $\mu_{1j} = \mu_{2j} = M_1$ for each j $(j = 1, \ldots, n)$. These problems are commonly referred to as single machine *coupled operations* scheduling problems (Potts and Whitehead (2007)).

In an open shop environment, time lags take the form l_{hij} $(j = 1, \ldots, n; h, i = 1, \ldots, m)$, specifying the minimum required delay between the completion of operation O_{hj} and the start of operation O_{ij}. If $l_{hij} = l_{ihj}$ for all jobs J_j $(j = 1, \dots, n)$ and all operations O_{hj} and O_{ij} $(h, i = 1, \dots, m; h \neq i)$, then the time lags are called symmetrical. In flow shops, job shops, and single machine shops with coupled operations, the order of the operations is fixed for every job, and hence the given time lags are of the form l_{ij}, specifying the required minimum time lag between operation O_{ij} and $O_{i+1,j}$. Special time lag cases include the situation where the time lags between operations of job J_j are of the form l_j for each $j = 1, \dots, n$ and the situation where all times lags equal a given positive constant l.

A schedule specifies for every operation when it is executed such that all constraints are satisfied; in other words, it specifies for every operation O_{ij} a starting time S_{ij} and a completion time C_{ij} such that all conditions are met $(i = 1, \dots, o; j = 1, \dots, n)$. In this chapter, we consider two scheduling objectives; the minimization of the *makespan*, or length, $C_{\max}$ of the schedule, and the minimization of the sum of the job completion times or *total flow time* $\sum C_j$, where C_j denotes the completion time of the last processed operation of job J_j $(j = 1, \dots, n)$. Throughout this chapter, we follow the standard three-field $\alpha|\beta|\gamma|$ scheduling notation (Graham et al (1979)). For instance, $F2|l_j|C_{\max}$ denotes the problem of minimizing makespan in a two-machine flow shop with job-dependent time lags; $J2|o_j \leq 2|\sum C_j$ denotes the problem of minimizing total completion time in the two-machine job shop where $o_j \leq 2$ denotes that each job consists of no more than two operations; $O3|l_{hij} = l|C_{\max}$ denotes the problem of minimizing makespan in the three-machine open shop with equal time lags; and $1|o_j = 2, l_j|C_{\max}$ denotes the single-machine problem of scheduling jobs with exactly two coupled operations to minimize makespan subject to job-dependent time lags.

5.1.2 Complexity

Scheduling problems with time lags are strongly NP-hard in their most general forms, and hence it is very unlikely that they can be solved to optimality in polynomial time. For makespan minimization problems, for instance, Yu et al (2004) showed that $F2|l_j|C_{\max}$ and $O2|l_j|C_{\max}$ are NP-hard in the strong sense, even if all processing times are equal to 1. From the first result, it follows of course that $J2|l_j|C_{\max}$ is NP-hard, too. Kern and Nawijn (1991) showed that the problem $1|o_j \leq 2, l_j|C_{\max}$ is NP-hard; Gupta (1996) strengthened the result by proving that the problem is NP-hard in strong sense. Further NP-hardness results are known for even more restrictive problems such as $O2|l_{hij} = l|C_{\max}$ and $O|p_{ij} = 1, l_{hij} = l_{ihj}|C_{\max}$ (Rayward-Smith and Rebaine (1992)), and $F2|p_{1j} = p_{2j}, l_j \in \{l_1, l_2\}|C_{\max}$, which is NP-hard in the strong sense (Yu (1996)).

As far as the total flow time objective is concerned, Garey et al (1976) and Achugbue and Chin (1982) proved that $F2||\sum C_j$ and $O2||\sum C_j$ are strongly -hard, respectively. The implication is that two-machine flow shop, job shop and open shop problems to minimize total flow subject to time lags are NP-hard in the strong sense, too, even if all time lags are the same. Brucker et al (2004) have shown that the problem $F2|p_{ij} = p, r_j, l_j|\sum C_j$ is NP-hard, also.

The best well-known polynomially solvable problem is undoubtedly $F2|l_j|C_{\max}$ if the solution space is restricted to permutation schedules (Mitten (1959)). Other polynomially solvable cases are more restrictive, such as $F2|l_j = l|C_{\max}$, $O2|p_{ij} = p, l_{ij}|C_{\max}$ and $F|p_{ij} = p, l_j|\sum C_j$ (Brucker et al (2004)).

5.1.3 Approximability

In this chapter, we are concerned with the approximability of scheduling models with time lags. We define an approximation algorithm to have *performance ratio* or *worst-case ratio* ρ, with $\rho > 1$, if it always produces a solution with objective value at most ρ times the optimal solution value. If such an algorithm runs in polynomial time, we call it a ρ-approximation algorithm. A *Polynomial-Time Approximation Scheme* (PTAS) is a family of polynomial time $(1+\epsilon)$-approximation algorithms over all $\epsilon > 0$.

The approximability of scheduling problems with time lags, and in particular the design of PTASs, is largely uncharted territory. Approximability results are limited to makespan minimization and concern ρ-approximation results only. For the two-machine flow shop environment, Dell'Amico (1996) provided a 2-approximation algorithm for $F2|l_j|C_{\max}$. Karuno and Nagamochi (2003) improved on this and gave an $\frac{11}{6}$-approximation algorithm. Ageev (2008) showed that the worst case ratio could be improved to $\frac{3}{2}$ if $p_{1j} = p_{2j}$ for each job J_j $(j = 1, \ldots, n)$.

For the two-machine open shop environment, Strusevich and Rebaine (1995) presented a $\frac{7}{4}$-approximation algorithm for $O2|l_{hij} = l_{ihj}|C_{\max}$. This bound was improved to $\frac{3}{2}$ by Strusevich (1999). Rebaine and Strusevich (1999) presented an $\frac{8}{5}$-approximation algorithm for $O2|l_{ij}|C_{\max}$. Rebaine (2004) presented a 2-approximation algorithm for $O2|l_{hij}|C_{\max}$ and a $(\frac{7}{4}-\frac{1}{2n})$-approximation algorithm for $O2|p_{ij}=p, l_{hij}|C_{\max}$.

5.2 Our contribution

As pointed out in the previous section, approximability of scheduling problems with time lags is a largely unexplored area. In this chapter, we present the first PTASs for scheduling problems with time lags, both for minimizing makespan and minimizing total flow time. Specifically, we identify a class $\mathcal{M}$ for makespan minimization problems with time lags and a class for $\mathcal{F}$ for total flow time minimization problems with time lags with the property that every scheduling problem (P) in those classes has a PTAS.

Every problem (P) in class $\mathcal{M}$ and class $\mathcal{F}$ has the following properties:

(i) Problem (P) is a deterministic scheduling problem with time lags, with one machine per manufacturing stage;
(ii) (P) is NP-hard or strongly NP-hard;
(iii) The counterpart problem without time lags, referred to as problem $(\bar{\text{P}})$ is polynomially solvable or has a PTAS;
(iv) All time lags are finite, that is, there exists a real $\mu > 0$ such that

$$l_{\max} \leq \mu p_{\min}, \tag{5.1}$$

where $l_{\max}$ is the maximum time lag and $p_{\min}$ is the smallest positive processing time of any operation. The value μ is fixed, that is, it is not part of the problem instance.

In addition to properties (i) - (iv), problems in class $\mathcal{M}$ have the further properties:

(v) The objective is to minimize makespan;
(vi) The scheduling environment is an m-machine flow shop (with m fixed, that is, m is not part of the problem instance but given a priori), a two-machine open shop, a two-machine job shop, or a single-machine shop with at most two coupled operations;

(vii) If the scheduling environment is a flow shop, then positive release times are allowed; otherwise, $r_j = 0$ for each job J_j $(j = 1, \ldots, n)$;
(viii) If the scheduling environment is a flow shop with $m > 2$ or positive release times, then every job has at least one operation with a positive processing time.

Table 5.1 lists some problems that belong to this class $\mathcal{M}$; the sign $'!'$ indicates that the problem is NP-hard in the strong sense, whereas $\mathcal{P}$ indicates that the problem is solvable in polynomial time.

Table 5.1. Some problems in class $\mathcal{M}$

Problem (P)	Complexity (P)	Complexity (P)
$F2\|l_j\|C_{\max}$	! (Yu (1996))	$\mathcal{P}$ (Johnson (1954))
$F2\|r_j, l_j\|C_{\max}$	! (Yu (1996))	PTAS (Hall (1994), & Kovalyov and Werner (1997))
$Fm\|p_{ij} = 1, tree, l_{jkl}\|C_{\max}$	! (Yu (1996))	$\mathcal{P}$ (Bruno et al (1980))
$O2\|l_{hij}\|C_{\max}$	! (Dell'Amico and Vaessens (1995))	$\mathcal{P}$ (Gonzalez and Sahni (1976))
$J2\|o_j \leq 2, l_j\|C_{\max}$	! (Yu (1996))	$\mathcal{P}$ (Jackson (1956))
$J2\|p_{ij} = 1, r_j, l_j\|C_{\max}$	! (Yu (1996))	$\mathcal{P}$ (Timkovsky (1997))
$1\|o_j = 2, l_j\|C_{\max}$	! (Gupta (1996))	$\mathcal{P}$ (trivial)

In addition to properties (i) - (iv), problems in class $\mathcal{F}$ have the following further properties:

(ix) The scheduling objective is to minimize total flow time;
(x) The scheduling environment is an m-machine flow shop where m is fixed;
(xi) Every job has at least one operation with a positive processing time.

Table 5.2 lists two earlier studied problems belonging to class $\mathcal{F}$.

5.3 PTAS for makespan problems in class $\mathcal{M}$

Consider any makespan minimization scheduling problem (P) belonging to class $\mathcal{M}$. We start with introducing some notation. Let $C^*_{\max}$ denote the optimal

Table 5.2. Two earlier studied problems in class $\mathcal{F}$

Problem (P)	Complexity (P)	Complexity (P)
$F2\|p_{ij}=1,r_j,l_{ij}\|\sum C_j$	! (Brucker et al (2004))	$\mathcal{P}$ (Baptiste and Timkovsky (2004))
$Fm\|l_{ij}\|\sum C_j$	! (Garey et al (1976))	PTAS (Fishkin et al (2002))

makespan for problem (P), and $C_{\max}(\pi)$ denotes the minimum makespan of a feasible schedule π for problem (P). Finally, let $\varepsilon > 0$ be any positive real number.

Define $P_k = \sum_{1\le j\le n, 1\le i\le o|\mu_{ij}=M_k} p_{ij}$ as the work content for machine M_k, for $k = 1, \ldots, m$. Clearly, we must have that

$$P_k \le C^*_{\max}, \text{ for } k = 1, \ldots, m. \tag{5.2}$$

Next, we divide the $n \times o$ operations O_{ij} ($i = 1, \ldots, o; j = 1, \ldots, n$) into $3m$ subsets in the following way:

- $\mathcal{Z}_k = \{O_{ij}|\mu_{ij} = M_k \text{ and } p_{ij} = 0\}$ for $k = 1, \ldots, m$; these are the *zero* operations.
- $\mathcal{S}_k = \{O_{ij}|\mu_{ij} = M_k \text{ and } 0 < p_{ij} \le \frac{\varepsilon}{\mu(o-1)} P_k\}$ for $k = 1, \ldots, m$; these are the *small* operations.
- $\mathcal{L}_k = \{O_{ij}|\mu_{ij} = M_k \text{ and } p_{ij} > \frac{\varepsilon}{\mu(o-1)} P_k\}$ for $k = 1, \ldots, m$; these are the *large* operations.

Note that all zero operations need to be scheduled and executed, although the duration of their processing is zero. The number of large operations per machine is bounded from above; indeed, we have that

$$|\mathcal{L}_k| \le \frac{\mu(o-1)}{\varepsilon}, \text{ for each } k = 1, \ldots, m. \tag{5.3}$$

To see that this is true, assume the opposite, that is, $|\mathcal{L}_k| > \frac{\mu(o-1)}{\varepsilon}$ for some k ($k = 1, \ldots, m$). The work content induced by the large operations on machine M_k would then be at least

$$\sum_{i,j|O_{ij}\in\mathcal{L}_k} p_{ij} \ge |\mathcal{L}_k| \frac{\varepsilon}{\mu(o-1)} P_k > P_k,$$

which would imply that the total processing time of the large operations on M_k exceeds the total processing time of all operations on M_k, which is a contradiction. We now differentiate between two cases: $\sum_{k=1}^{m} |\mathcal{S}_k| = 0$ and $\sum_{k=1}^{m} |\mathcal{S}_k| > 0$.

If $\sum_{k=1}^{m} |\mathcal{S}_k| = 0$, then we have jobs with large and zero operations only, and we can find an optimal solution for this case in time polynomial in n but exponential in $\frac{\mu(o-1)m}{\varepsilon}$ in the following way. First, we use explicit enumeration to schedule all the jobs with at least one large operation, of which there are at most $\sum_{k=1}^{m} |\mathcal{L}_k| \leq \frac{\mu(o-1)m}{\varepsilon}$ by virtue of inequality (5.3). For m-machine flow shop problems with $m > 2$ or flow shop problems where jobs have positive release times, we are then done, since by definition of class $\mathcal{M}$ and in particular by property (viii), no job has zero operations only. For any other type of problem in class $\mathcal{M}$, there may be jobs with zero operations only, and we insert those jobs into the intermediate schedule in such a way that feasibility and optimality of the schedule is maintained; this can easily be achieved, for instance, by scheduling all first operations as early as possible and all second operations as late as possible.

Alternatively, if $\sum_{k=1}^{m} |\mathcal{S}_k| > 0$, then we have at least one operation O_{ij} with $\mu_{ij} = M_k$ and $0 < p_{ij} \leq \frac{\varepsilon}{\mu(o-1)} P_k$, for some i, j, and k. Using (5.1) and (5.2), we have

$$l_{\max} \leq \mu(\frac{\varepsilon}{\mu(o-1)} P_k) \leq \frac{\varepsilon}{o-1} P_k \leq \frac{\varepsilon}{o-1} C^*_{\max}. \tag{5.4}$$

Let now problem $(\bar{\mathrm{P}})$ be the counterpart problem of (P) *without* given time lags; by definition of class $\mathcal{M}$, $(\bar{\mathrm{P}})$ either is solvable in polynomial time or has a PTAS. If $(\bar{\mathrm{P}})$ is solvable in polynomial time, then let $\bar{\sigma}^*$ denote an optimal schedule for this problem. Otherwise, that is, if $(\bar{\mathrm{P}})$ is NP-hard but has a PTAS, let $\bar{\sigma}^\varepsilon$ denote a feasible schedule for $(\bar{\mathrm{P}})$ with makespan at most $(1+\epsilon)\bar{C}^*_{\max}$, where $\bar{C}^*_{\max}$ is the optimal makespan for problem $(\bar{\mathrm{P}})$. Of course, we have that $\bar{C}^*_{\max} \leq C^*_{\max}$.

Let now σ be the schedule obtained from either $\bar{\sigma}^*$ or $\bar{\sigma}^\varepsilon$, whichever is appropriate, by inserting as little idle time as possible between the different operations to ensure that σ is a feasible schedule for problem (P). How much idle time needs to be inserted, and subsequently, how good is $C_{\max}(\sigma)$?

First, suppose that $(\bar{\mathrm{P}})$ is an m-machine flow shop problem. Then we need to insert no more than $l_{\max}$ idle time on each machine $M_2, \ldots, M_m$ to transform either $\bar{\sigma}^*$ or $\bar{\sigma}^\varepsilon$ into a feasible solution σ for problem (P). Accordingly, we have

$$C_{\max}(\sigma) \leq C_{\max}(\bar{\sigma}^*)+(m-1)l_{\max} \leq (1+\varepsilon)C^*_{\max}, \text{ if } \sigma \text{ has been obtained from } \bar{\sigma}^*,$$

and

$$C_{\max}(\sigma) \leq C_{\max}(\bar{\sigma}^\varepsilon)+(m-1)l_{\max} \leq (1+\varepsilon)^2 C^*_{\max}, \text{ if } \sigma \text{ has been obtained from } \bar{\sigma}^\varepsilon.$$

Now suppose that ($\bar{\text{P}}$) is any other type of problem than a flow shop problem, that, is suppose it is a two-machine job shop, two-machine open shop, or a single machine shop with at most two coupled operations. We will first show that no operation in schedule $\bar{\sigma}^*$ or $\bar{\sigma}^\varepsilon$, whichever is appropriate, needs to be delayed by more than $l_{\max}$ time units to obtain a feasible schedule σ for problem (P). First, suppose that ($\bar{\text{P}}$) is a two-machine job shop problem. Then we may assume without loss of generality that in $\bar{\sigma}^*$ or $\bar{\sigma}^\varepsilon$, whichever is appropriate, either machine processes all first operations O_{1j} before any second operation O_{2k}; if $\bar{\sigma}^*$ or $\bar{\sigma}^\varepsilon$ is no such schedule we can easily transform it into an equivalent schedule with the stated property. This property implies that we need to insert no more than $l_{\max}$ idle time on either machine to transform $\bar{\sigma}^*$, or $\bar{\sigma}^\varepsilon$, into a feasible solution σ for problem (P). Second, suppose that ($\bar{\text{P}}$) is a two-machine open shop problem. An argument similar to the one used for the two-machine job shop problem applies; we may assume without loss of generality that in $\bar{\sigma}^*$ or $\bar{\sigma}^\varepsilon$ all first operations precede all second operations on either machine. Accordingly, the second operations need to be delayed by at most $l_{\max}$ time to transform $\bar{\sigma}^*$, or $\bar{\sigma}^\varepsilon$, into a feasible solution σ for problem (P). Finally, let ($\bar{\text{P}}$) be a single-machine problem where jobs have at most two coupled operations. Again, we may assume that all first operations precede all second operations, and hence we need to insert at most $l_{\max}$ time in between to guarantee that the resulting schedule σ is feasible for problem (P). So indeed, if ($\bar{\text{P}}$) is any other problem than an m-machine flow shop problem, we need to insert no more than $l_{\max}$ idle time per machine to obtain a feasible schedule σ for problem (P). This means that

$$C_{\max}(\sigma) \leq C_{\max}(\bar{\sigma}^*) + l_{\max} \leq (1+\varepsilon)C^*_{\max}, \text{ if } \sigma \text{ has been obtained from } \bar{\sigma}^*,$$

and

$$C_{\max}(\sigma) \leq C_{\max}(\bar{\sigma}^\varepsilon) + l_{\max} \leq (1+\varepsilon)^2 C^*_{\max}, \text{ if } \sigma \text{ has been obtained from } \bar{\sigma}^\varepsilon.$$

In conclusion, if $\sum_{k=1}^{m} |\mathcal{S}_k| > 0$, then any makespan minimization problem (P) in class $\mathcal{M}$ has a PTAS, also. Now, we are ready to give a description of our algorithm.

ALGORITHM I: PTAS FOR ANY PROBLEM (P) IN CLASS $\mathcal{M}$

Step 1. If $\sum_{k=1}^{m} |\mathcal{S}_k| > 0$, go to Step 2; otherwise go to Step 3.

Step 2. Find schedule σ, to be obtained either from $\bar{\sigma}^*$, if $(\bar{\mathrm{P}})$ is polynomially solvable, or from $\bar{\sigma}^\varepsilon$, if $(\bar{\mathrm{P}})$ has a PTAS. Stop.

Step 3. Explicitly enumerate all possible sequences for the jobs with at least one large operation. For jobs with zero operations only, we schedule those jobs by scheduling all first operations as early as possible and all second operations as late as possible. Find thus an optimal schedule σ^* for problem (P). Stop.

Theorem 10 *Algorithm I is a Polynomial-Time Approximation Scheme for any makespan minimization scheduling problem with time lags in class $\mathcal{M}$. The running time of the algorithm is polynomial in n but exponential in $\frac{\mu(o-1)m}{\varepsilon}$.* □

5.4 PTAS for problems in class $\mathcal{F}$

Let now (P) be any problem belonging to class $\mathcal{F}$, which consists of m-machine flow shop total flow time minimization problems with time lags; see Section 2. Let F^* denote the optimal solution value for problem (P), and let $\varepsilon > 0$ be any real positive number. Let $p_{i[j]}$ denote the j-th smallest processing time on machine M_i $(i = 1, \dots, m; j = 1, \dots, n)$ and define $P_{ik} = \sum_{j=1}^{k} p_{i[j]}$. We then have that

$$F^* \geq P_{in}, \text{ for each } i = 1, \dots, m. \tag{5.5}$$

Similar to the previous subsection, we divide the $n \times m$ operations into $3m$ subsets:

- $\mathcal{Z}_i = \{O_{ij} | \, p_{ij} = 0\}$ for $i = 1, \dots, m$; these are the *zero* operations.
- $\mathcal{S}_i = \{O_{ij} | \, 0 < p_{ij} \leq \frac{\varepsilon}{\mu(m-1)} \frac{P_{in}}{n}\}$ for $i = 1, \dots, m$; these are the *small* operations.
- $\mathcal{L}_i = \{O_{ij} | \, p_{ij} > \frac{\varepsilon}{\mu(m-1)} \frac{P_{in}}{n}\}$, for $i = 1, \dots, m$; these are the *large* operations.

If $\sum_{i=1}^{m} |\mathcal{S}_i| > 0$, we have at least one operation O_{ij} with $0 < p_{ij} < \frac{\varepsilon}{\mu(m-1)} \frac{P_{in}}{n}$. With (5.1) and (5.5), we have

$$l_{\max} \leq \mu(\frac{\varepsilon}{\mu(m-1)}\frac{P_{in}}{n}) \leq \frac{\varepsilon}{m-1}\frac{P_{in}}{n} \leq \frac{\varepsilon}{m-1}\frac{F^*}{n}. \tag{5.6}$$

Let now problem $(\bar{\mathrm{P}})$ be the counterpart problem of (P) *without* given time lags; accordingly, by definition of class $\mathcal{F}$, $(\bar{\mathrm{P}})$ either is solvable in polynomial time or has a PTAS. If $(\bar{\mathrm{P}})$ is solvable in polynomial time, then let $\bar{\sigma}^*$ denote an optimal schedule for this problem. Otherwise, that is, if $(\bar{\mathrm{P}})$ is NP-hard but has a PTAS, let $\bar{\sigma}^\varepsilon$ denote a feasible schedule for $(\bar{\mathrm{P}})$ with solution value at most $(1+\epsilon)\bar{F}^*$, where $\bar{F}^*$ is the optimal solution value for problem $(\bar{\mathrm{P}})$. Of course, we have that $\bar{F}^* \leq F^*$.

Let σ be the schedule obtained from either $\bar{\sigma}^*$ or $\bar{\sigma}^\varepsilon$, whichever is appropriate, by inserting as little idle time as possible between the different operations to ensure that σ is a feasible schedule for problem (P). Then we need to insert no more than $l_{\max}$ idle time before each operation on each machine $M_2, \ldots, M_m$ to transform either $\bar{\sigma}^*$ or $\bar{\sigma}^\varepsilon$ into a feasible solution σ for problem (P). Accordingly, the completion time of each job J_j in σ is at most $(m-1)l_{\max}$ time later than the completion time of J_j in $\bar{\sigma}^*$ or $\bar{\sigma}^\varepsilon$. Hence, using (5.6), we have

$$F(\sigma) \leq F(\bar{\sigma}^*) + n(m-1)l_{\max} \leq (1+\varepsilon)F^*, \text{ if } \sigma \text{ has been obtained from } \bar{\sigma}^*,$$

and

$$F(\sigma) \leq F(\bar{\sigma}^\varepsilon) + n(m-1)l_{\max} \leq (1+\varepsilon)^2F^*, \text{ if } \sigma \text{ has been obtained from } \bar{\sigma}^\varepsilon.$$

So, if $\sum_{i=1}^m |\mathcal{S}_i| > 0$, then problem (P) has a PTAS.

Now consider the case that $\mathcal{S}_i = \emptyset$ for some i $(i = 1, \ldots, m)$. We have then the following lemma.

Lemma 27 *If $\mathcal{S}_i = \emptyset$, then $|\mathcal{L}_i| \leq \frac{2\mu(m-1)}{\varepsilon} - 1$; that is, if there are no small operations on machine M_i, the number of large operations on machine M_i is bounded from above by $\frac{2\mu(m-1)}{\varepsilon} - 1$.*

Proof. Suppose k is the smallest index such that $p_{i[k]} > \frac{\varepsilon}{\mu(m-1)}\frac{P_{in}}{n}$, and hence suppose there are $K = n-k+1$ large operations to be scheduled on M_i. This implies that

$$\frac{(K)(K+1)}{2}p_{i[k]} \leq P_{in}.$$

We also have that

$$\frac{K(K+1)}{2}p_{i[k]} > (\frac{K(K+1)}{2})(\frac{\varepsilon}{\mu(m-1)})(\frac{P_{in}}{n}).$$

Hence, we must have that

$$\frac{K(K+1)}{2}\frac{\varepsilon}{\mu(m-1)}\frac{P_{in}}{n} < P_{in},$$

which implies that

$$\frac{K(K+1)}{2n} \leq \frac{\mu(m-1)}{\varepsilon}.$$

Since $K \leq n$, we have $K < \frac{2\mu(m-1)}{\varepsilon} - 1$. □

Lemma 27 implies that $n < (\frac{2\mu(m-1)}{\varepsilon} - 1)m$, since each has job at least contains one large one operation (see property (xi) of problems (P) in class $\mathcal{F}$; see Section 2). Hence, for fixed m and μ, we can find the optimal schedule in polynomial time by complete enumeration.

Now, we give a summary of our algorithm.

Algorithm II: PTAS for any problem (P) in class $\mathcal{F}$

Step 1. If $\sum_{i=1}^{m} |\mathcal{S}_i| > 0$, go to Step 2, otherwise go to Step 3.

Step 2. Find schedule σ, to be obtained either from $\bar{\sigma}^*$, if ($\bar{\text{P}}$) is polynomially solvable, or from $\bar{\sigma}^\varepsilon$, if ($\bar{\text{P}}$) has a PTAS. Stop.

Step 3. Enumerate all possible sequences explicitly and find thus a schedule σ^* with minimum total flow time. Stop.

Theorem 11 *Algorithm II is a Polynomial-Time Approximation Scheme for any total flow time minimization scheduling problem with time lags in class $\mathcal{F}$. The running time of the algorithm is polynomial in n but exponential in $(\frac{2\mu(m-1)m}{\varepsilon} - m)$.* □

5.5 Conclusions

In this chapter, we have presented the first PTASs for machine scheduling problems with time lags. Specifically, we have defined two classes of scheduling problems with time lags, one for minimizing makespan and one for minimizing total completion time, such that each problem in those classes possesses a PTAS. Our algorithms mark a step forward for time lag problems without earlier known approximability results, such as $F2|r_j, l_j|C_{\max}$ and $Fm|l_{ij}|\sum C_j$, as well as for problems with known approximability results, such as $F2|l_j|C_{\max}$, if the time lags are relatively restricted in size. For example, the best approximation algorithm for the problem $F2|l_j|C_{\max}$ has a worst-case ratio of $\frac{11}{6}$ (Karuno and Nagamochi (2003)). For $\mu \leq 5$, our algorithm either improves the ratio to $\frac{3}{2}$ in $O(n \log n)$ time, or finds the optimal solution by enumerating at most 10 large jobs. For $\mu \leq 10$, our algorithm either improves the ratio to $\frac{5}{3}$ in $O(n \log n)$ time, or finds the optimal solution by enumerating at most 15 large jobs.

Remember that no explicit enumeration is required if there is at least one job with a small operation. This implies that for any problem (P) whose counterpart ($\bar{\text{P}}$) is polynomially solvable, we can find an $n_0 > 0$ for any given $\mu > 0$ and $\varepsilon > 0$ such that our algorithm requires no explicit enumeration if $n > n_0$. This n_0 is defined by the maximum number of jobs with large operations; accordingly, for problems in class $\mathcal{M}$, $n_0 = \mu(o-1)m/\varepsilon$ (see inequality (5.3)), and for problems in class $\mathcal{F}$, $n_0 = (\frac{2\mu(m-1)}{\varepsilon} - 1)m$ (see Lemma 27). Take for example again the problem $F2|l_j|C_{\max}$ or the problem $O2|l_{hij}|C_{\max}$, which has a known worst-case ratio of 2 (Rebaine (2004)). For $\mu \leq 10$, our algorithm improves the worst-case ratio to $\frac{6}{5}$ in $O(n \log n)$ time for any $n > n_0 = 100$ with no enumeration required. For $\mu \leq 25$, our PTAS improves the worst-case ratio to $\frac{3}{2}$ in $O(n \log n)$ time for any $n > n_0 = 100$, also with no enumeration required.

6

Fixed Interval Shop Scheduling

And so from that, I've always been fascinated with the idea that complexity can come out of such simplicity.

- Will Wright (1960-)

We introduce and analyze the fixed interval shop scheduling problem, where the objective is to maximize the weighted number of jobs that can be processed in a two-stage machine shop, if each job has a fixed start and end time and requires a given transportation time, a time lag, for moving from one stage to the other. This problem is a natural extension of the parallel-machine scheduling problem with fixed start and end times, which is relatively well understood. We prove that the fixed interval two-machine flow shop problem is NP-hard in the strong sense for general time lags, even in the case of unit processing times. The problem is solvable in polynomial time if all time lags are equal. The fixed interval two-machine job shop problem is solvable in $O(n^3)$ time if the time lags are identical and relatively small. For the fixed interval two-machine open shop problem, this is true as well.

6.1 Introduction

We consider the problem of maximizing the weighted number of jobs that can be processed by a two-stage machine shop, with one machine in each stage, if each job has a fixed start time and fixed finish time, and the total processing time a job plus the time needed to transfer the job between the two stages of the shop equals the length of the time interval between the job's start and finish time. Specifically, we assume that there is a set $\mathcal{J}$ of n independent jobs $J_1, \ldots, J_n$ that can be processed in a two-stage machine shop, with machine M_1 in the first stage and machine M_2 in the second. No machine can process more than one job at a time, and no job can be processed by more than one machine at a time. Each job J_j $(j = 1, \ldots, n)$ consists of 2 operations O_{1j} and O_{2j}, operation O_{ij} $(i = 1, 2)$ requires processing on a pre-specified machine $\mu_{ij} \in \{M_1, \ldots, M_2\}$ during an uninterrupted processing time $p_{ij} > 0$, and the transportation time or traveling time between machine M_1 and M_2 is l_j. Furthermore, each job J_j has a fixed start time s_j by which it needs to be started if J_j is selected for processing and a fixed finish time f_j, by which it needs to be finished, if selected. Every job has a value $w_j > 0$, which reflects the benefit gained from processing this job. Each job is to be processed once or not at all. The objective is to find a schedule that maximizes the value of the jobs selected for processing. We refer to this problem as the *fixed interval shop scheduling* problem. If O_{1j} needs to be processed before O_{2j}, O_{1j} needs to be processed by machine M_1 and O_{2j} by machine M_2 for every job J_j $(j = 1, \ldots, n)$, then the scheduling environment is called a *flow shop*. If O_{1j} needs to be processed before O_{2j}, then it is called a *job shop*. If there is no restriction on which operation should be processed first, then the scheduling environment is an *open shop*.

Mitten (1959) was the first to consider a scheduling problem with time lags, in which jobs need to be processed by a number of non-bottleneck machines in between those two bottleneck machines. The time required to complete production on the intermediate machines may then be represented by certain time lag and the problem is effectively equal to the $F2|l_j|C_{\max}$. Mitten (1959) shows that, restricted to *permutation* schedules, an optimal schedule can be found in polynomial time, where a permutation schedule is a schedule in which job sequences in the first and the second machine are the same.

Our main motivation for studying this problem comes from the problem of scheduling barges along container terminals in the Port of Rotterdam; see Douma (2008). Specifically, each barge needs to call upon certain container terminals for the unloading and loading of containers. Depending on the way containers haven been stacked on the barge and the number of containers to be unloaded and loaded at each terminal, the routing along the container terminals may be fixed in advance of may allow some degrees of freedom. Every barge operator submits a proposed sailing plan to the container terminal operators to negotiate the arrival and departure times. Such a sailing plan specifies the container terminals that need to be called upon for unloading and loading containers, the order in which this needs to be done, the times it likes to arrive at and depart from each terminal, and the sailing times between the terminals, which take typically between 20 minutes and two hours. The barge operator's main goal is to avoid possible delays in the sailing schedule. The problem could be modeled as a fixed interval shop scheduling problem, where the barges coming to the Port of Rotterdam are the jobs, the container terminals, or more specifically the crane terminals, are the machines, the times required by the terminal cranes for the unloading and loading of containers as machine processing times, and the sailing times between the different container terminals are the time lags. The objective of maximizing the weighted number of jobs reflects the goal to change as few sailing plans as possible.

Other practical situations in which variants of this problem occur include the coffee production process (Simeonov and Simeonovová (1997)) and the aircraft maintenance and gate assignment process in airports (Kroon et al (1995) Kroon et al (1997)).

In the parallel machine environment, the problem of scheduling jobs with fixed start and finish times has been relatively well-studied. Arkin and Silverberg (1987) analyze the problem of maximizing the value of n jobs completed by k identical machines, where jobs have fixed start and finish times, and give an $O(n^2 \log n)$ time algorithm for its solution. They also prove that the problem is NP-hard in the case of non-identical machines and show that for a fixed number of nonidentical machines k, the problem can be solved in $O(n^{k+1})$ time, which of course is useful for relatively small k only.

The fixed interval shop scheduling problem, while never been studied before, is close to other well-studied problems. For instance, the fixed interval job shop and the fixed interval flow shop problem are essentially the problems of minimizing the weighted number of tardy jobs in a job shop and flow shop, respectively, where every job has a release date r_j and a deadline $\bar{d}_j$ by which the job needs to be finished, and $\bar{d}_j - r_j$ equals the total some of the processing and transportation times for every job J_j $(j = 1, \ldots, n)$.

The fixed interval shop scheduling problem has also features of no-wait shop scheduling problems (See Reddi and Ramamoorthy (1972), Hall and Sriskandarajah (1996)).

In this chapter, we focus on the fixed interval two-stage shop scheduling problem. Note that if in those practical situations mentioned above there are only two bottleneck machines (container terminals, workstations, aircraft, gates), the time required for the activities on the non-bottleneck machines between those two bottleneck machines can be represented by (an increase in the) time lags. Accordingly, results for the fixed interval two-stage shop scheduling problem contribute significantly to the body of knowledge for this type of problem.

In Section 6.2, we prove that fixed interval flow shop problem with general time lags is NP-hard in the strong sense, even for two machines, and present an $O(n^2)$ time dynamic programming algorithm for the problem where the time lags are identical for all jobs. In Section 6.3, we give an $O(n^3)$ time algorithm for the fixed interval two-machine job shop scheduling problem and show that this algorithm also applies if $l_j \leq p_{\min}$, where $p_{\min}$ is the smallest processing time of any operation. In Section 6.4, we study the fixed interval two-machine open shop scheduling problem and show how the algorithm for the fixed interval job shop problem can be modified for the open shop variant.

6.2 Fixed interval two-machine flow shop scheduling

Theorem 12 *The fixed interval two-machine flow shop scheduling problem with general time lags is NP-hard problem in the strong sense, even if $p_{ij} = 1$ for each $i = 1, \ldots, m$ and $j = 1, \ldots, n$.*

Proof. The reduction is from Cubic Planar Monotone 1-in-3 SAT problem, which is a restricted version of the Satisfiability problem (Garey and Johnson (1979)). The proof is given in Appendix A. □

In the remainder of this section, we analyze a restricted variant of the fixed interval flow shop scheduling problem where the time lags have the form $l_j = l$ for j ($j = 1, \ldots, n$). We refer to this variant as the special case with equal time lags.

6.2.1 The case of equal time lags

Note that if J_j is selected for scheduling, then it need to be processed in the interval $[s_j, s_j + p_{1j})]$ on machine M_1 and in the interval $[s_j + p_{1j} + l, f_j]$ on machine M_2. Accordingly, any pair of jobs for which these fixed intervals overlap cannot be selected both. Furthermore, since the time lags between the machines are the same for all jobs, one job can never start before another job and yet finish after it. Also, since the jobs have pre-fixed intervals in which they need to be processed if selected, each feasible schedule can be fully codified by presenting the jobs in order of non-decreasing start times.

These properties possessed by any feasible schedule allows us to develop a forward dynamic programming algorithm that considers the jobs for scheduling one by one in order of non-decreasing starting times. To this end, we re-index the jobs accordingly. For notational convenience, we also introduce a dummy zero-processing time job J_0 with $w_0 = 0$; furthermore, we assume that for this particular job there is no travel time required between the two machines. In our dynamic programming algorithm, we may therefore assume without loss of generality that J_0 is selected for scheduling.

When considering any other job, we have two possible decisions: either to select it for scheduling, in which case it needs to be processed in its fixed interval, or to reject it, in which case it requires no machine time. Our algorithm is predicated upon the following optimality principle. Consider all schedules for the interval scheduling problem defined on the jobs $J_0, \ldots, J_j$ subject to the condition that job J_i $(0 \leq i \leq j)$ was selected last. J_i marks the tail of any such schedule, since the equal time lags structure implies that one job can never start before another job and yet finish after it, on any machine. Accordingly, we have that all such schedules complete at time $s_i + p_{1i}$ on M_1, at time f_i on machine M_2. We define such a schedule to be in state (j, i). To schedule the remaining $n-j$ jobs, we need to consider only a schedule with maximum value among all schedules in state (j, i).

Let $F_j(i)$ denote the optimal solution value of any schedule in state (j, i), and let $\sigma_j(i)$ be any schedule in state (j, i) with solution value $F_j(i)$. By default, we have that $F_j(0) = 0$ for each j $(j = 0, \ldots, n)$ and we let $\sigma_0(0)$ denote the empty schedule.

Consider now a schedule $\sigma_j(i)$ with value $F_j(i)$ for some $0 \leq i \leq j$, $j \geq 1$. Then there are two possibilities:

1. $i = j$. In this case, job J_j has been selected for scheduling, it finishes last on all machines, and $\sigma_j(j)$ has been obtained from some previous schedule $\sigma_{j-1}(h)$ with value $F_{j-1}(h)$. For this job J_h, the job immediately preceding J_j in $\sigma_j(j)$, it must hold that
   a) $s_h + (p_{1h} + l) \leq s_j$ and $f_h \leq s_j + p_{1j} + l$; in other words, the fixed job intervals of J_h and J_j do not overlap each other; and
   b) $F_j(j) = F_{j-1}(h) + w_j$. This means that

$$F_j(j) = F_{j-1}(h) + w_j = \max_{g:J_g \in \mathcal{A}(j)} \{F_{j-1}(g) + w_j\},$$

   where $\mathcal{A}(j)$ contains all jobs J_g that can be scheduled before J_j; i.e., $\mathcal{A}(j) = \{J_g \mid s_g + p_{1g} + l \leq s_j \text{ and } f_g \leq s_j + p_{1j} + l\}$.

   $F_j(i) = F_{j-1}(i)$.

We are now ready to present the dynamic programming recursion. The initialization is $F_j(i) = 0$ for $i = 0, \ldots, n$ and $j = 0, \ldots, n$. For $j = 1, \ldots, n$, $i = 0, \ldots, j$, we do the following

$$F_j(i) = \begin{cases} F_{j-1}(i) & \text{if } i \leq j-1 \\ \max_{g:J_g \in \mathcal{A}(j)} \{F_{j-1}(g) + w_j\} & \text{if } i = j. \end{cases}$$

The optimal solution value is then $\max_{0 \leq i \leq n} F_n(i)$, and the optimal solution can be found by backtracking. Hence, we have the following result.

Theorem 13 *The fixed interval two-machine flow shop scheduling problem with equal time lags is solvable in $O(n^2)$ time and space.* □

The same optimality principles work for the fixed interval flow shop scheduling problem with equal time lags and a general number of machines, and hence the dynamic programming algorithm can be extended to the m-machine case in a straightforward way.

Corollary 2 *The m-machine fixed interval flow shop scheduling problem with equal time lags is solvable in $O(mn^2)$ time and $O(n^2)$ space.* □

6.3 Fixed interval two-machine job shop scheduling

Let $\mathcal{J}^1$ contain all jobs that need to be processed first on machine M_1 and then on machine M_2, whereas $\mathcal{J}^2$ contains all jobs that need to be processed in the reverse order. Again, without loss of generality, we assume for sake of notational convenience that there is a dummy job J_0, with parameters $p_{10} = p_{20} = w_0 = 0$ for which no travel time between M_1 and M_2 is required.

In the flow shop problem, the dynamic programming algorithm leveraged the property that if $s_h < s_i$ then job J_h needs to precede job J_i on both machines, if we decide to select them both. This property in fact still holds for the two sets $\mathcal{J}^1$ and $\mathcal{J}^2$ separately. The complicating design factor is that we can have an optimal schedule with a job $J_h \in \mathcal{J}^1$ and a job $J_i \in \mathcal{J}^2$, in which O_{1h} precedes O_{2i} on one machine, whereas O_{2h} precedes O_{1i} on the other. This situation, however, can happen if and only if the intervals $[s_h, s_h + p_{1h}]$ and $[s_i + p_{1i} + l, f_i]$ do not overlap on machine M_1 and the intervals $[s_i, s_i + p_{1i}]$ and $[s_h + p_{1h} + l, f_h]$ do not overlap on machine M_2. In such a situation, we refer to such jobs J_h and J_i as *crossing jobs*. See Figure 6.1 for an illustration of this concept.

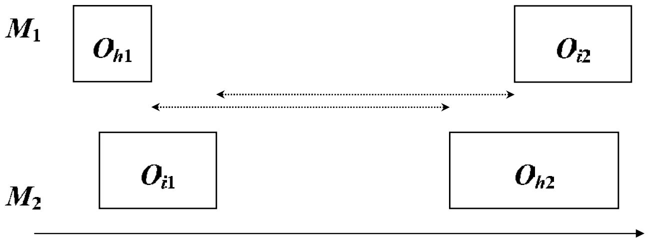

Fig. 6.1. J_h and J_i are crossing jobs. The arrows represent the equal time lags $l > 0$.

If $l = 0$, that is, if there are no time lags, two jobs J_h and J_i can be crossing, if and only $s_h + p_{1h} = s_i + p_{1i}$ and $J_h \in \mathcal{J}^1$ and $J_i \in \mathcal{J}^2$, or vice versa. See Figure 6.2 for an illustration.

Clearly, our dynamic programming algorithm would have to accommodate this possibility. To this end, we again re-index the jobs in order of non-decreasing starting times s_j, setting ties in an arbitrary fashion, but now we need to store the last *two* selected jobs in our dynamic programming recursion. Specifically, we

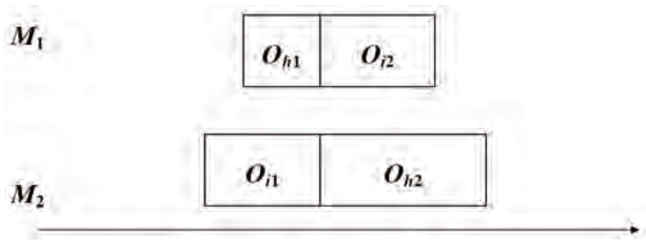

Fig. 6.2. Crossing jobs for the case $l = 0$.

define a schedule to be in state (j, h, i) if it is a feasible schedule for the fixed interval scheduling problem defined on the jobs $J_0, \ldots, J_j$, in which J_i is the last selected job and J_h is the last but one selected job, where $0 \leq h < i \leq j$. By default, a schedule in state $(j, 0, 0)$ is an empty schedule. By tracking the last two selected jobs, we can check whether the first operation $O_{1,j+1}$ of the next job to be considered for selection, J_{j+1}, can be scheduled after J_i, in case J_h and J_i are not crossing, after J_i and J_h in case they are, or in the gap between J_h and J_i on machine $\mu_{1,j+1}$, which is of course possible only if J_h and J_i are not crossing jobs and the gap between J_h and J_i is big enough.

Let $F_j(h, i)$ denote the optimal solution value for the interval scheduling problem for the jobs $J_0, \ldots, J_j$ subject to the condition that J_i has been selected last and J_h has been selected last but one. Let $\sigma_j(h, i)$ be any schedule with minimum objective value $F_j(h, i)$. Considering the optimality principle of dynamic programming, we know that a schedule $\sigma_j(h, i)$ $(0 \leq h \leq i \leq j)$ with value $F_j(h, i)$ must have been obtained from some previous schedule $\bar{\sigma} = F_{j-1}(a, b)$ for some jobs J_a and J_b.

If $l \leq p_{\min} = \min_{i=1,2;1 \leq j \leq n} p_{ij}$, then we can use the same logic, and we can design a dynamic programming algorithm with the same states as for $l = 0$, and the same optimality principle. To see why this is so, note that it is crucial that for such relatively small l there can never be more than one operation $O_{1,j+1}$ scheduled in the gap between J_h and J_i on machine μ_{j+1}, for any $j + 1 > i > h$. Accordingly, it suffices to track the last two selected jobs as long as $0 \leq l \leq p_{\min}$. In what follows, we assume exactly this.

As said, we know that a schedule $\sigma_j(h,i)$ $(0 \le h \le i \le j)$ with value $F_j(h,i)$ must have been obtained from some previous schedule $\bar{\sigma} = F_{j-1}(a,b)$ for some jobs J_a and J_b. We can distinguish the following cases:

1. $j > i$. Then $\sigma_j(h,i)$ must have been obtained from $\bar{\sigma} = \sigma_{j-1}(h,i)$ by rejecting J_j. In this case, we have $F_j(h,i) = F_{j-1}(h,i)$.
2. $j = i$. In this case $\sigma_j(h,j)$ must have been obtained from some previous schedule $\bar{\sigma} = \sigma_{j-1}(g,h)$ for some $0 \le g \le h$, either by appending J_j to the end of schedule, or by inserting J_j, or at least its first operation, between J_h and J_i on machine $\mu_{1,j+1}$ and scheduling J_j concurrently with J_h. Let now $\mathcal{A}(j,h)$ be the set of all such feasible previous schedules $\sigma_{j-1}(g,h)$. In what follows, we characterize this set $\mathcal{A}(j,h)$.
   a) $g = h = 0$. In this particular case, $\sigma_j(0,j)$ was obtained from the empty schedule $\sigma_{j-1}(0,0)$.
   b) $g = 0$, $h \ge 1$, and $J_h \in \mathcal{J}^1$. We differentiate between three subcases:
      i. $J_j \in \mathcal{J}^1$. In this case, J_j can be scheduled after J_h both on M_1 and M_2, if and only if $s_j \ge s_h + p_{1h}$ and $s_j + p_{1j} + l \ge f_h$.
      ii. $J_j \in \mathcal{J}^2$ and $s_j \ge f_h$. This means that J_j can be scheduled after J_h both on M_1 and M_2.
      iii. $J_j \in \mathcal{J}^2$, $s_j + p_{1j} \le s_h + p_{1h} + l$ and $s_j + p_{1j} + l \ge s_h + p_{1h}$. In this case, J_h and J_j can be processed concurrently.
   c) $g = 0$, $h \ge 1$, and $J_h \in \mathcal{J}^2$. We differentiate between three subcases:
      i. $J_j \in \mathcal{J}^1$ and $s_j \ge f_h$. This means that J_j can be scheduled after J_h both on M_1 and M_2.
      ii. $J_j \in \mathcal{J}^1$, $s_j + p_{1j} \le s_h + p_{1h} + l$ and $s_j + p_{1j} + l \ge s_h + p_{1h}$. In this case, J_j can be processed concurrently with J_h on different machines.
      iii. $J_j \in \mathcal{J}^2$. In this case, if $s_j \ge s_h + p_{1h}$ and $s_j + p_{1j} + l \ge f_h$, J_j can be scheduled after J_h, both on M_1 and M_2.
   d) $J_g \in \mathcal{J}^1$, $J_h \in \mathcal{J}^1$. Accordingly, J_h is scheduled after J_g. We have then the following subcases:
      i. $J_j \in \mathcal{J}^1$. In this case, J_j can be added if and only if $s_j \ge s_h + p_{1h}$ and $s_j + p_{1j} + l \ge f_h$, in which case J_j is scheduled after J_h, both on M_1 and M_2.
      ii. $J_j \in \mathcal{J}^2$ and $s_j \ge f_h$. This means that J_j can be scheduled after J_h both on M_1 and M_2.

iii. $J_j \in \mathcal{J}^2$, $s_j + p_{1j} \leq s_h + p_{1h} + l$, $s_j \geq f_g$, and $s_j + p_{1j} + l \geq s_h + p_{1h}$. In this case, the first operation of J_j can be processed on M_2 in the gap between J_g and J_h, and J_j can be scheduled concurrently with J_h.

e) $J_g \in \mathcal{J}^1$, $J_h \in \mathcal{J}^2$, and $s_h \geq f_g$. In this case, J_h is scheduled after J_g both on M_1 and M_2. We identify the following subcases:

i. $J_j \in \mathcal{J}^1$ and $s_j \geq f_h$. This means that J_j can be scheduled after J_h both on M_1 and M_2.

ii. $J_j \in \mathcal{J}^1$, $s_j + p_{1j} \leq s_h + p_{1h} + l$, $s_j \geq s_g + p_{1g}$, and $s_j + p_{1j} + l \geq s_h + p_{1h}$. In this case, the first operation of J_j can be processed on M_1 in the gap between J_g and J_h, and J_j can be scheduled concurrently with J_h.

iii. $J_j \in \mathcal{J}^2$. In this case, J_j can be added if and only if $s_j \geq s_h + p_{1h}$ and $s_j + p_{1j} + l \geq f_h$, in which case J_j is scheduled after J_h, both on M_1 and M_2.

f) $J_g \in \mathcal{J}^2$ and $J_h \in \mathcal{J}^2$. We have the following three subcases:

i. $J_j \in \mathcal{J}^1$ and $s_j \geq f_h$. This means that J_j can be scheduled after J_h both on M_1 and M_2.

ii. $J_j \in \mathcal{J}^1$, $s_j + p_{1j} \leq s_h + p_{1h} + l$, $s_j \geq f_g$, and $s_j + p_{1j} + l \geq s_h + p_{1h}$. In this case, the first operation of J_j can be processed on M_1 in the gap between J_g and J_h, and J_j can be processed concurrently with J_h.

iii. $J_j \in \mathcal{J}^2$. In this case, J_j can be added if and only if $s_j \geq s_h + p_{1h}$ and $s_j + p_{1j} + l \geq f_h$, in which case J_j is scheduled after J_h, both on M_1 and M_2.

g) $J_g \in \mathcal{J}^2$ and $J_h \in \mathcal{J}^1$. There are three subcases to consider:

i. $J_j \in \mathcal{J}^1$. Then, J_j can be added if and only if $s_j \geq s_h + p_{1h}$ and $s_j + p_{1j} + l \geq f_h$, in which case J_j is scheduled after J_h, both on M_1 and M_2.

ii. $J_j \in \mathcal{J}^2$ and $s_j \geq f_h$. This means that J_j can be scheduled after J_h both on M_1 and M_2.

iii. $J_j \in \mathcal{J}^2$, $s_j + p_{1j} \leq s_h + p_{1h} + l$, $s_j \geq s_g + p_{1g}$, and $s_j + p_{1j} + l \geq s_h + p_{1h}$. In this case, the first operation of J_j can be processed on M_2 in the gap between J_g and J_h, and J_j can be processed concurrently with J_h on M_1.

h) $J_g \in \mathcal{J}^1$ and $J_h \in \mathcal{J}^2$, and J_g and J_h are scheduled concurrently. We have two subcases:

i. $J_j \in \mathcal{J}^1$. Then adding J_j is possible if and only if $s_j \geq f_h$, and $s_j + p_{1j} + l \geq f_g$.
ii. $J_j \in \mathcal{J}^2$. Adding J_j is then possible if and only if $s_j \geq f_g$, and $s_j + p_{1j} + l \geq f_h$.

i) $J_g \in \mathcal{J}^2$ and $J_h \in \mathcal{J}^1$, and J_g and J_h are scheduled concurrently. Again, we have two subcases to consider:
i. $J_j \in \mathcal{J}^1$. Appending J_j is then possible if and only if $s_j \geq f_g$, and $s_j + p_{1j} + l \geq f_h$.
ii. $J_j \in \mathcal{J}^2$, Appending J_j is then possible if and only if $s_j \geq f_h$, and $s_j + p_{1j} + l \geq f_g$.

The dynamic programming recursion is then as follows. The initialization is $F_j(h,i) = 0$ for $j = 0, \ldots, n$, $h = 0, \ldots, n$, $i = 0, \ldots, n$, and for $j = 1, \ldots, n$, $i = 0, \ldots, j$, $h = 0, \ldots, i$, we do the following:

$$F_j(h,i) = \begin{cases} F_{j-1}(h,j), & \text{if } j > i, \\ \max_{g:\sigma_{j-1}(g,h) \in \mathcal{A}(j,h)} F_{j-1}(g,h) + w_j, & \text{if } j = i. \end{cases}$$

The optimal solution value is then $\max_{0 \leq h \leq i, h+1 \leq i \leq n} F_n(h,i)$, and the optimal solution can be found by backtracking. Hence, we have the following result.

Theorem 14 *The fixed interval two-machine job shop scheduling problem with equal time lags l and $0 \leq l \leq p_{\min}$ is solvable in $O(n^3)$ time and space.* □

6.4 Fixed interval two-machine open shop problem

In the two-machine open shop problem, the scheduler has for every job J_j ($j = 1, \ldots, n$) a choice; either scheduling the operation that needs to be processed by M_1 first and then scheduling the other operation on M_2, or the other way around. Either way, if we assume equal time lags, then each job still has its fixed interval $[s_j, s_j + p_{1j} + l + p_{2j}]$ in which it needs to be processed. In what follows, we again consider the case that $0 \leq l \leq p_{\min}$.

To a large extent, we apply the same logic as used for the fixed interval two-machine job shop problem to cope with the fixed interval two-machine open shop problem. To this end, we first re-index the jobs in order of non-decreasing start times and then transform an instance for the open shop problem with n jobs into an instance for the job shop problem with $2n$ jobs by creating for each job J_j two copies, job J'_{2j-1} and job J'_{2j} $(j = 1, \ldots, n)$, and by imposing the restriction that at most one of these copies may be selected for scheduling. Each job with an odd index has fixed routing $M_1 \to M_2$ and each job with an even index has fixed routing $M_2 \to M_1$. The jobs with the first type of routing belong to the job set $\mathcal{J}^1$), the others to the job set $\mathcal{J}^2$. Accordingly, each J'_{2j-1} $(j = 1, \ldots, n)$ requires first processing during a time $p'_{1,2j-1} = p_{1j}$ on M_1 and then during a time $p'_{2,2j-1} = p_{2j}$ on M_2, whereas J'_{2j} $(j = 1, \ldots, n)$ requires first processing during a time $p'_{1,2j} = p_{2j}$ on M_2 and then during a time $p'_{2,2j} = p_{1j}$ on M_1. Both copies have the same weight w_j. Any instance of the fixed interval two-machine open shop problem is then equivalent to this transformed instance of the fixed interval two-machine job shop problem subject to the condition that at most one copy of each original job J_j can be selected.

We define a schedule to be in state (j, h, i) if it is a feasible schedule for the transformed fixed interval two-machine job shop scheduling problem defined on the jobs $J_0, \ldots, J_j$ $(j = 1, \ldots, 2n)$, in which J_i is the last selected job, J_h is the last but one selected job. Let $F_j(h, i)$ denote the maximum objective value of any schedule in state (j, h, i), and let $\sigma_j(h, i)$ be any schedule with minimum objective value $F_j(h, i)$.

Based on the optimality principle of dynamic programming, we know that $\sigma_j(h, i)$ must have been obtained from some schedule σ' that is optimal for a previous state. Specifically, we have the following situations:

1. if j is odd, and if $j > i$, then $\sigma_j(h, i)$ must have been obtained from either $\sigma_{j-1}(h, i)$ or, if $j \geq 3$, from $\sigma_{j-2}(h, i)$, in both cases by not selecting J_j for scheduling.
2. if j is odd, and if $j = i$, then J_j has been selected for scheduling, and then $\sigma_j(h, j)$ must have been obtained from either $\sigma_{j-1}(g, h)$ for some J_g $(g = 1, \ldots, 2n)$ or, if $j \geq 3$, from $\sigma_{j-2}(f, h)$ for some J_f $(f = 1, \ldots, n)$, in both cases by either scheduling J_j after J_h or, scheduling it concurrently with J_h.
3. if j is even and if $j > i$, then $\sigma_j(h, i)$ must have been obtained from either $\sigma_{j-2}(h, i)$, or, if $j \geq 4$, from $\sigma_{j-3}(h, i)$, in both cases by not selecting J_j for scheduling.
4. if j is even, and if $j = i$, then J_j has been selected for scheduling, and then $\sigma_j(h, j)$ must have been obtained from either $\sigma_{j-2}(g, h)$ for some J_g

$(g = 1, \ldots, 2n)$ or, if $j \geq 4$, from $\sigma_{j-2}(f, h)$ for some J_f $(f = 1, \ldots, n)$, in both cases by either scheduling J_j after J_h, or scheduling it concurrently with J_h.

The above transitions exclude the possibility that both copies of any original job are selected. Furthermore, the conditions under which a schedule $\sigma_j(h, j)$ can be derived from a previous schedule $\sigma_{j-1}(g, h)$ or $\sigma_{j-2}(f, h)$ in case j is odd or from a previous schedule $\sigma_{j-2}(g, h)$ or $\sigma_{j-3}(f, h)$ in case j is even can be derived in exactly the same way as we did for the fixed interval two-machine job shop problem in Section 2.2. For easy of notation, let $\mathcal{A}(j, h)$ denote the set of feasible previous schedules from which $\sigma_j(h, j)$ can be obtained.

The initialization of the dynamic programming algorithm is $F_j(h, i) = 0$ for $j = 0, \ldots, n$, $h = 0, \ldots, n$, $i = 0, \ldots, n$, and the recursion is to calculate the following for $j = 1, \ldots, n$, $i = 0, \ldots, j$, $h = 0, \ldots, i$:

$$F_j(h,i) = \begin{cases} 0, & \text{if } j = 1,2 \text{ and } i > j, \\ w_j, & \text{if } j = 1,2 \text{ and } i = j, \\ \max\{F_{j-1}(h,i), F_{j-2}(h,i)\}, & \text{if } j > i,\ j \geq 3, \text{ and } j \text{ is odd}, \\ \max\{F_{j-2}(h,i), F_{j-3}(h,i)\}, & \text{if } j > i,\ j \geq 4, \text{ and } j \text{ is even}, \\ \max \left\{ \begin{array}{l} \max_{g:\sigma_{j-1}(g,h) \in \mathcal{A}(j,h)} F_{j-1}(g,h) + w_j, \\ \max_{g:\sigma_{j-2}(g,h) \in \mathcal{A}(j,h)} F_{j-2}(g,h) + w_j, \end{array} \right\} & \text{if } i = j,\ j \geq 3, \text{ and } j \text{ is odd}, \\ \max \left\{ \begin{array}{l} \max_{g:\sigma_{j-2}(g,h) \in \mathcal{A}(j,h)} F_{j-2}(g,h) + w_j, \\ \max_{g:\sigma_{j-3}(g,h) \in \mathcal{A}(j,h)} F_{j-3}(g,h) + w_j, \end{array} \right\} & \text{if } i = j,\ j \geq 4, \text{ and } j \text{ is even}. \end{cases}$$

The optimal solution value is then $\max_{0 \leq h \leq i, h+1 \leq i \leq n} F_n(h, i)$, and the optimal solution can be found by backtracking. Hence, we have the following result.

Theorem 15 *The fixed interval two-machine open shop scheduling problem is solvable in $O(n^3)$ time and space.* □

6.5 Conclusions

In this chapter, we have introduced the fixed interval shop scheduling problem, and we have shown that the problem is solvable in polynomial time in a flow shop environment if the time lags are the same for every job but NP-hard in the strong sense for general time lags, even in the case of unit-processing times. The fixed interval two-machine job and open shop problems are shown to be solvable in polynomial time if the time lags are equal and smaller than the processing time of any operation.

There are several open problems, notably the complexity of the fixed interval two-machine job and open shop problems for general equal time lags.

Another related type of problem, much in the spirit of the objective function analyzed by Arkin and Silverberg (1987) and Kroon et al (1997), is to minimize the number of machines in each stage of the shop in order to be able to process all available jobs.

Appendix A: Fixed Interval Flow Shop Scheduling is NP-hard

We prove that the fixed interval scheduling problem in the two-machine flow shop with general time lags is strongly NP-hard, even in the case of unit processing times. The proof proceeds by reduction from the Cubic Planar Monotone 1-in-3 SAT problem, which is a restricted version of the Satisfiability problem (SAT, for short) and known to be strongly NP-hard.

We follow the notation used in Garey and Johnson (1979).

Let $U = \{u_1, u_2, \ldots, u_m\}$ be a set of Boolean *variables*. A *truth assignment* for U is a function $t : U \to \{T, F\}$. If $t(u) = T$, we say that u is "true" under t; if $t(u) = F$ we say that u is "false." If u is a variable in U, then u and $\overline{u}$ are *literals* over U. A *clause* over U is a set of literals over U, such as $\{u_1, \overline{u}_3, u_8\}$. It represents the disjunction of those literals and is *satisfied* by a truth assignment if and only if at least one of its members is true under that assignment. A collection C of clauses over U is *satisfiable* if and only if there exists some truth assignment for U that simultaneously satisfies all the clauses in C. Such a truth assignment is called a *satisfying truth assignment* for C.

The SAT problem is defined as follows: given a set U of variables and a collection $C = \{c_1, c_2, \ldots, c_m\}$ of clauses over U, is there a satisfying truth assignment for C? The problem was proved to be NP-complete by Cook (1971).

A 3-SAT problem is a restricted version of a SAT problem, which includes the constraints of $|c_i| = 3$, for $1 \leq i \leq m$, i.e., there are exactly 3 literals in each clause. The problem is still NP-complete. (Garey and Johnson (1979))

A monotone 1-in-3 SAT problem is a restricted version of a 3-SAT problem. It is required that each clause in C has exactly one true literal, so called 1-in-3. The monotone means that no $c \in C$ contains a negated literal. The problem is proved to be NP-complete by Schaefer (1978).

A cubic planar monotone 1-in-3 SAT problem is a further restricted version of a monotone 1-in-3 SAT problem. The cubic version of the problem means that every variable has exactly three occurrences. We can think of the instance as a graph, with a vertex for each variable and each clause, and an edge connecting a variable to a clause if it occurs (positively or negatively) in that clause. In planar 1-in-3 SAT, this graph is assumed to be planar. The cubic planar version of monotone 1-in-3 SAT problem was proved to be NP-complete by Moore and Robson (2001).

Theorem 16 *The fixed interval two-machine flow shop problem with general time lags is NP-hard in the strong sense, even in the case of unit processing times.*

Proof. We show that the cubic planar monotone 1-in-3 SAT problem could be polynomially reduced to this problem. Given an instance of a cubic planar monotone 1-in-3 SAT problem. We have a set of clauses $C = \{c_1, c_2, \ldots, c_n\}$. There are three literals in each clause and all the literals are not negative. Because every variable has exactly three occurrences, we have exactly n variables $u_1, u_2, \ldots, u_n$ that appear in the clauses set C, i.e., we have three u_j in C for $j = 1, \ldots, n$. We could replace each occurrence of u_j in C with $u_{j1}, u_{j2},$ *or* u_{j3} respectively for $j = 1, \ldots, n$, such that all literals in C are not equal, and add additional constrains of $u_{j1} = u_{j2} = u_{j3}$ to ensure the problem remains the same. We define a function of $f(u_{jk}) = l$ to indicate that u_{jk} belongs to clause c_l in C, for $l = 1, \ldots, n$.

Now consider an instance of the two-machine flow shop problem with two kinds of jobs, among which there are $3n$ SAT jobs J^S_{jk}, for $j = 1, \ldots, n$ and k=1, 2, 3, and $2n$ auxiliary jobs J^A_{jk}, for $j = 1, \ldots, n$ and k=1, 2. SAT jobs need to be processed in both machines and auxiliary jobs need to be processed on machine M_2 only. We have unit operations time on each machine for all jobs.

The release time for SAT jobs are defined as $r^S_{jk} = f(u_{jk}) - 1$, the deadline for SAT jobs are defined as $d^S_{jk} = n + 3(j-1) + k$, and the time lags between

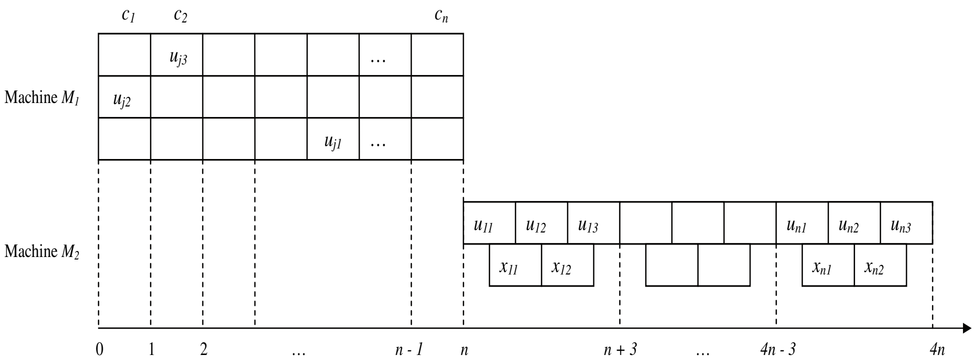

Fig. 6.3. An instance of fixed interval two-machine flow shop scheduling problem with time lags.

two operations for SAT jobs are $l_{jk}^S = d_{jk}^S - r_{jk}^S - 2 = n + 3j + k + f(u_{jk}) - 6$, for $j = 1, \ldots, n$ and k=1, 2, 3.

The release time for auxiliary jobs are defined as $r_{jk}^A = d_{jk}^S - \frac{1}{2}$ and the deadline for auxiliary jobs are defined as $d_{jk}^A = r_{jk}^A + 1 = d_{jk}^S + \frac{1}{2}$, for $j = 1, \ldots, n$ and k=1, 2.

The value of each SAT job is set to 2 and that of auxiliary job is set to 1.5. The instance of the flow shop problem could be illustrated by Figure 6.3. In that figure, SAT jobs are marked with u_{jk} and auxiliary jobs are marked with x_{jk}. In machine M_1, all SAT jobs need to be processed in n unit-time-slots, while in machine M_2, all SAT jobs need to be processed in n 3-unit-intervals.

We first show that if the instance of the cubic planar monotone 1-in-3 SAT problem has a satisfying truth assignment, the instance of two-machine flow shop with time lags would have an objective value of $4n$. If u_j is set to true in the satisfying truth assignment, we have $u_{j1} = u_{j2} = u_{j3} = 1$ and select J_{jk}^S from SAT jobs for processing on both machines. Because there is only one variable which is set to true in each clause, there are exactly n SAT jobs selected. These n SAT jobs will contribute $2n$ to the objective function. On the other hand, all the n SAT jobs are clustered on machine M_2. They occupied one third of the n 3-unit-intervals. In the rest two-third of n 3-unit-intervals in the period of $[n, 4n]$, we can at most process $\frac{4}{3}n$ auxiliary jobs, which contribute another $2n$ to the objective function.

We next show that if this fixed interval two-machine flow shop scheduling problem with time lags is solved with objective value of $4n$, the instance of the cubic planar monotone 1-in-3 SAT problem has a satisfying truth assignment. Let schedule S be a feasible schedule of the instance of two-machine flow shop. Let x be the total number of SAT jobs which are processed in the schedule. Because the maximum number of SAT jobs that could be process in this two-machine problem is bounded by n unit-time-slots in machine M_1, we have $x \leq n$. Then the maximum total time units left in machine M_2 for auxiliary jobs is bounded by $3n - x$. Within these time slots, the total values produced by auxiliary jobs are also bounded by $3n - x$ because there are at most two auxiliary jobs could be finished in every three time slots and each auxiliary jobs contribute 1.5 to the value function. Therefore the total value of any schedule S is bounded by $2x + 3n - x = 3n + x$. If we have a schedule with objective value of $4n$, then we have $x = n$ and the number of auxiliary jobs which are processed is exactly $\frac{4}{3}n$. To process n SAT jobs, we will use n time units on both machines respectively. To process $\frac{4}{3}n$ auxiliary jobs, we only have $2n$ time units in machine M_2. Because the minimum requirement for auxiliary job is 1.5 time units (the 0.5 time units before or after each auxiliary job are useless for both kinds of jobs), any pair of jobs J_{j1}^A and J_{j2}^A must be processed or discarded together. Otherwise, these auxiliary jobs could not be processed within $2n$ time units. Therefore, a schedule with objective value $4n$ means that each clause has exactly one true literal, where u_{jk} is set to true if J_{jk}^S is processed and $u_{j1} = u_{j2} = u_{j3}$ are ensured because jobs J_{j1}^S, J_{j2}^S, and J_{j3}^S are processed or discarded together for $j = 1, \ldots, n$ and $k = 1, 2, 3$, i.e., the cubic planar monotone 1-in-3 SAT problem has a satisfying truth assignment.

This completes the proof. $\square$

Consider now the fixed interval single-machine scheduling problem, where each job consists of two coupled operations with a pre-specified time lag between the completion of the first and the start of the second operation. That is, the first operation of each job J_j ($j = 1, \ldots, n$) needs to be processed in the interval $[s_j, s_j + 1]$ and the second operation in the interval $[f_j - 1, f_j$. The time lag between the two coupled operation is then $l_j = f_j - s_j - 2$. The above proof can then be adjusted to prove the following result.

Corollary 3 *The fixed interval single-machine scheduling problem, where each job consists of two coupled operations, is strongly NP-hard, even in the case of unit processing times.* □

This corollary nicely supplements a result of Potts and Whitehead (2007) who show that the single-machine problem of minimizing makespan for jobs with coupled operations is strongly NP-hard.

Part IV

Conclusions and Future Directions

7

Conclusions and Future Directions

We know accurately only when we know little, with knowledge doubt increases.

- Johann Wolfgang von Goethe (1749-1832)

7.1 Conclusions

This thesis contributes to the body of knowledge of both on-line and off-line machine scheduling problems with time lags.

On-line problems

As far as on-line problems are concerned, we have considered the two-machine flow shop, job shop and open shop scheduling problem with minimal time lags so as to minimize the makespan. For these on-line problems, we have analyzed two variants: the non-clairvoyant and the clairvoyant variant. In the non-clairvoyant variant, we have no information about the processing time of either operation of a job until it finishes, while in the clairvoyant variant, we know the processing times of a job's operations as soon as the job arrives. We have shown that the greedy algorithm is the best possible algorithm for the first variant and that its competitive ratio is 2 for the flow shop, job shop and open shop problem. Furthermore, no non-delay algorithm can do better than the greedy algorithm. As far as delay algorithms are concerned, we have shown that no on-line delay algorithm has a competitive ratio better than $\frac{\sqrt{5}+1}{2} \approx 1.618$ for the flow shop and job shop problem, and that no on-line delay algorithm has a competitive ratio better than $\sqrt{2}$ for the open shop problem. However, these results leave open the question whether an on-line delay algorithm has a better competitive ratio than the greedy algorithm. For the open shop problem, we have also shown that if the time lags are no larger than the maximum positive processing time, the competitive ratio of the greedy algorithm improves to $\frac{5}{3}$.

Off-line problems

As far as off-line machine scheduling problems with time lags are concerned, our contribution is threefold. First, we have presented the first approximation algorithms with fixed worst-case performance guarantees for scheduling m parallel two-stage flow shops to as to minimize the makespan for $m = 2$ and $m = 3$. This NP-hard problem decomposes into two subproblems: first, assigning each job to one of the parallel flow shops, and then scheduling the jobs assigned to the same flow shop by use of Johnson's rule. For $m = 2$, we have presented a $\frac{3}{2}$-approximation algorithm, and for $m = 3$, a $1\frac{5}{7}$-approximation algorithm. Both algorithms run in $O(n \log n)$ time. Our results also apply to the parallel flow shop problem with time lags, which arise when jobs are transported between different flow shops for their processing.

Our second contribution is the identification of two classes of machine scheduling problems with time lags that possess Polynomial-Time Approximation Schemes (PTASs). One class concerns the objective of minimizing the makespan, the other minimizing total completion time. The running times of these approximation schemes are polynomial in the number of jobs but exponential in the number of machines and the ratio between the largest time lag and the smallest positive processing time. These results constitute the first PTAS results for scheduling problems with time lags.

Our third contribution was the introduction and analysis of the fixed interval shop scheduling problem, where the objective is to maximize the weighted number of jobs that can be processed in a two-stage flow, job, or open shop, if each job has a fixed start and finish time and requires a given transportation time, a time lag, for moving from one stage to the other. This problem is a natural extension of the parallel-machine scheduling problem with fixed start and completion times, which is relatively well understood. We prove that the fixed interval two-machine flow shop problem is NP-hard in the strong sense for general time lags, even in the case of unit processing times. The problem is solvable in polynomial time if all time lags are equal. The fixed interval two-machine job shop problem is solvable in $O(n^3)$ time if the time lags are identical and relatively small. For the fixed interval two-machine open shop problem, this is true as well.

7.2 Directions for future works

On-line problems

A vexing open question for the clairvoyant on-line two-machine shop scheduling problem with time lags is whether there exists a delay algorithm with a better competitive ratio that the greedy algorithm. Other interesting questions include:

(1) is the greedy algorithm still the best possible algorithm for m-machine problems ($m \geq 3$) when the on-line environment is clairvoyant or no deliberate delays are allowed?

(2) is the greedy algorithm still the best possible algorithm for other objective functions?

(3) when both minimal and maximal time lags are involved, can we find on-line algorithms with fixed competitive ratios?

(4) if the on-line paradigm is changed to one in which the jobs are presented one by one, according to some list, can we find on-line algorithms with fixed competitive ratios?

Besides these questions, there are many other on-line scheduling problems with time lags remaining untouched, which could be research topics in future.

Off-line problems

For the parallel flow shop problem, open questions include:

(1) do algorithms exist with betters worst-case performance guarantees?

(2) can we design similar algorithms with similar worst-case performance guarantees for the parallel m-stage flow shop problem, if $m \geq 4$?

(3) how can we leverage the possibility of transporting jobs between flow shops to find improved approximation algorithms?

As to the Polynomial-Time Approximation Schemes for scheduling with bounded time lags, relevant open questions are:

(1) does a PTAS exist if there is no upper bound on the time lags?

(2) do any PTAS results exist for other objective functions besides makespan and total completion time?

(3) can we develop algorithms with fixed worst-case performance guarantees for these problems?

For the fixed interval scheduling problems, our algorithms have been developed for two-machine problems to maximize the total weighted number of jobs that can be processed. The following extensions may be of interest:

(1) what are the complexity status of fixed interval two-machine job shop and open shop problems in the case of general time lags?

(2) can we develop efficient algorithm to minimize the number of machines in each stage of the shop in order to be able to process all available jobs?

(3) for fixed interval two-machine shop problems with general time lags, can we develop approximation algorithms with fixed worst-case performance guarantees?

Finally, for each these off-line problems, the corresponding on-line problem is also interesting.

In this thesis, we have only tackled a limited set of machine scheduling problems with time lags. Many problems are still left unsolved. Let's end the thesis with Confucius' (551BC-479BC) wisdom: *to know what you know and what you do not know, that is true knowledge.*

References

Achugbue J, Chin F (1982) Schedulling the open shop to minimize mean flow time. SIAM Journal on Computing 11:709–720

Ageev A (2008) A 3/2-approximation for the proportionate two-machine flow shop scheduling with minimum delays. Lecture Notes in Computer Science 4927:55–66

Al-Salem A (2004) A heuristic to minimize makespan in proportional parallel flow shops. International Journal of Computing & Information Sciences 2(2):98

Arkin E, Silverberg E (1987) Scheduling jobs with fixed start and end times. Discrete Applied Mathematics 18(1):1–8

Baptiste P, Timkovsky V (2004) Shortest path to nonpreemptive schedules of unit-time jobs on two identical parallel machines with minimum total completion time. Mathematical Methods of Operations Research 60(1):145–153

Brucker P, Knust S (2010) Complexity results for scheduling problems. URL: http://www.mathematik.uni-osnabrueck.de/research/or/class/

Brucker P, Knust S, Cheng T, Shakhlevich N (2004) Complexity results for flow-shop and open-shop scheduling problems with transportation delays. Annals of Operations Research 129(1):81–106

Bruno J, Jones III J, So K (1980) Deterministic scheduling with pipelined processors. IEEE Transactions on Computers 29(4):308–316

Cao D, Chen M (2003) Parallel flowshop scheduling using Tabu search. International Journal of Production Research 41(13):3059–3073

Chen B (1994) Scheduling multiprocessor flow shops. In New Advances in Optimization and Approximation(D,-Z Du and J Sun, Eds) pp 1–8

Chen B (1995) Analysis of classes of heuristics for scheduling a two-stage flow shop with parallel machines at one stage. Journal of the Operational Research Society pp 234–244

Chen B, Vestjens A, Woeginger G (1998) On-line scheduling of two-machine open shops where jobs arrive over time. Journal of Combinatorial Optimization 1(4):355–365

Chen B, Du D, Han J, Wen J (2001) On-line scheduling of small open shops. Discrete Applied Mathematics 110(2-3):133–150

Cook S (1971) The complexity of theorem-proving procedures. Proceedings of the Third Annual ACM Symposium on Theory of Computing pp 151–158

Dell'Amico M (1996) Shop problems with two machines and time lags. Operations Research 44(5):777–787

Dell'Amico M, Vaessens R (1995) Flow and open shop scheduling on two machines with transportation times and machine-independent processing times is NP-hard. Materiali di Discussione 141

Douma A (2008) Aligning the operations of barges and terminals through distributed planning. PhD thesis, University of Twente

Fishkin A, Jansen K, Mastrolilli M (2002) A PTAS for shop scheduling with release dates to minimize the average weighted completion time. Technical Report: IDSIA-19-02

Garey M, Johnson D (1979) Computers and Intractability: A Guide to the Theory of NP-Completeness. WH Freeman & Co. New York, NY, USA

Garey M, Johnson D, Sethi R (1976) The complexity of flowshop and jobshop scheduling. Mathematics of Operations Research 1(2):117–129

Gonzalez T, Sahni S (1976) Open shop scheduling to minimize finish time. Journal of the ACM 23(4):665–679

Graham R, Lawler E, Lenstra JK, Rinnooy Kan A (1979) Optimization and approximation in deterministic sequencing and scheduling: a survey. Annals of Discrete Mathematics 5:287–326

Gupta J (1988) Two-stage, hybrid flowshop scheduling problem. Journal of the Operational Research Society pp 359–364

Gupta J (1996) Comparative evaluation of heuristic algorithms for the single machine scheduling problem with two operations per job and time-lags. Journal of Global Optimization 9(3):239–253

Gupta J, Tunc E (1991) Schedules for a two-stage hybrid flowshop with parallel machines at the second stage. International Journal of Production Research 29(7):1489–1502

Gupta J, Hariri A, Potts C (1997) Scheduling a two-stage hybrid flow shop with parallel machines at the first stage. Annals of Operations Research 69:171–191

Hall L (1994) A polynomial approximation scheme for a constrained flow-shop scheduling problem. Mathematics of Operations Research 19(1):68–85

Hall N, Sriskandarajah C (1996) A survey of machine scheduling problems with blocking and no-wait in process. Operations Research 44(3):510–525

Hoogeveen J, Lenstra JK, Veltman B (1996) Preemptive scheduling in a two-stage multiprocessor flow shop is NP-hard. European Journal of Operational Research 89(1):172–175

Hoogeveen J, Lenstra JK, Van de Velde S (1997) Sequencing and scheduling. in: M Dell'Amico, F Maffioli, and S Martello(eds) Annotated bibliographies in combinatorial optimization, Wiley, Chichester pp 181–197

Jackson J (1956) An extension of Johnson's results on job lot scheduling. Naval Research Logistics Quarterly 3(3):201–203

Johnson S (1954) Optimal two-and three-stage production schedules with setup times included. Naval Research Logistics Quarterly 1(1):61–68

Kanet J (1986) Tactically delayed versus non-delay scheduling: An experimental investigation. European Journal of Operational Research 24(1):99–105

Kanet J, Sridharan V (2000) Scheduling with inserted idle Time: problem taxonomy and literature review. Operations Research 48(1):99–110

Karuno Y, Nagamochi H (2003) A better approximation for the two-machine flowshop scheduling problem with time lags. Algorithms and Computation: 14th International Symposium, ISAAC 2003, Kyoto, Japan, December 15-17, 2003: Proceedings

Kern W, Nawijn W (1991) Scheduling multi-operation jobs with time lags on a single machine. 2nd Twente Workshop on Graphs and Combinatorial Optimization p81-86(SEEN 92-24905 15-64)

Knust S (1999) Shop-scheduling problems with transportation. PhDthesis, Universittat Osnabrtuck, Fachbereich Mathematik/Informatik

Kovalyov M, Werner F (1997) A polynomial approximation scheme for problem F2/rj/Cmax. Operations Research Letters 20(2):75–79

Kravchenko S (2000) On the complexity of minimizing the number of late jobs in unit time open shop. Discrete Applied Mathematics 100(1-2):127–132

Kroon L, Salomon M, Van Wassenhove L (1995) Exact and approximation algorithms for the operational fixed interval scheduling problem. European Journal of Operational Research 82(1):190–205

Kroon L, Salomon M, Van Wassenhove L (1997) Exact and approximation algorithms for the tactical fixed interval scheduling problem. Operations Research 45(4):624–638

Lawler E, Lenstra JK, Rinnooy Kan A (1981) Minimizing maximum lateness in a two-machine open shop. Mathematics of Operations Research 6(1):153–158

Lee C, Vairaktarakis G (1994) Minimizing makespan in hybrid flowshops. Operations Research Letters 16(3):149–158

Lenstra JK, Rinnooy Kan A, Brucker P (1977) Complexity of machine scheduling problems. Annuals of Discrete Mathematics 1:343–362

Linn R, Zhang W (1999) Hybrid flow shop scheduling: a survey. Computers & Industrial Engineering 37(1-2):57–61

Mitten L (1959) Sequencing n jobs on two machines with arbitrary time lags. Management Science 5(3):293–298

Moore C, Robson J (2001) Hard tiling problems with simple tiles. Discrete and Computational Geometry 26(4):573–590

Panwalkar S (1973) Job shop sequencing problem on two machines with time lag constraints. Management Science 19(9):1063–1066

Papadimitriou C, Steiglitz K (1998) Combinatorial Optimization: Algorithms and Complexity. Dover Pubns

Pinedo M (1995) Scheduling: Theory, Algorithms, and Systems. Prentice Hall

Potts C, Whitehead J (2007) Heuristics for a coupled-operation scheduling problem. Journal of the Operational Research Society 58(10):1375–1388

Pruhs K, Torng E (1994) Fault-tolerant scheduling. In: Proc. of the 26th Ann. ACM Symp. on Theory of Computing, CRC Press, pp 115–124

Qi X (2008) New results for scheduling two flowlines. Working paper of Hong Kong University of Science and Technology

Rayward-Smith V, Rebaine D (1992) Open shop scheduling with delays. Informatique Théorique et Applications (Imprimé) 26(5):439–448

Rayward-Smith V, Rebaine D (2008) Analysis of heuristics for the UET two-machine flow shop problem with time delays. Computers and Operations Research 35(10):3298–3310

Rebaine D (2004) Scheduling the two-machine open shop problem with non-sysmetric time delays. Congres ASAC 2004, Quebec, Canada

Rebaine D, Strusevich V (1999) Two-machine open shop scheduling with special transportation times. The Journal of the Operational Research Society 50(7):756–764

Reddi S, Ramamoorthy C (1972) On the flow-shop sequencing problem with no wait in process. Operational Research Quarterly pp 323–331

Ruiz R, Vazquez-Rodriguez J (2010) The hybrid flow shop scheduling problem. European Journal of Operational Research 205(1):1–18

Schaefer T (1978) The complexity of satisfiability problems. Proceedings of the Tenth Annual ACM Symposium on Theory of Computing pp 216–226

Seiden S, Sgall J, Woeginger G (2000) Semi-online scheduling with decreasing job sizes. Operations Research Letters 27(5):215–221

Sgall J (1998) On-line scheduling a survey. Online algorithms: the state of the art, Lecture Notes in Computer Science 1442:196–231

Simeonov S, Simeonovová J (1997) Simulation scheduling in food industry application. Czech Journal of Food Science 20(1):31–37

Strusevich V (1999) A heuristic for the two-machine open-shop scheduling problem with transportation times. Discrete Applied Mathematics 93(2-3):287–304

Strusevich V, Rebaine D (1995) Two-machine open shop with transportations times. ECCO VIII, May 8-10, Poznan, Poland

Timkovsky V (1997) A polynomial-time algorithm for the two-machine unit-time release-date job-shop schedule-length problem. Discrete Applied Mathematics 77(2):185–200

Vairaktarakis G, Elhafsi M (2000) The use of flowlines to simplify routing complexity in two-stage flowshops. IIE Transactions 32(8):687–699

Van Groningen L (2006) Performance Measurement in Barge Planning. Master thesis, RSM Erasmus University

Vazirani V (2004) Approximation Algorithms. Springer

Wang H (2005) Flexible flow shop scheduling: optimum, heuristics and artificial intelligence solutions. Expert Systems 22(2):78–85

Yu W (1996) The Two-machine Flow Shop Problem with Delays and the One-machine Total Tardiness Problem. Ph.D. thesis, Eindhoven University of Technology

Yu W, Hoogeveen H, Lenstra JK (2004) Minimizing makespan in a two-machine flow shop with delays and unit-time operations is NP-hard. Journal of Scheduling 7(5):333–348

Zhang X, Van de Velde S (2009) Polynomial-time approximation schemes for scheduling problems with time lags. Journal of Scheduling, Article in Press

Zhang X, Van de Velde S (2010a) Approximation algorithms for the parallel flow shop problem. Working paper of RSM Erasmus University Rotterdam

Zhang X, Van de Velde S (2010b) Fixed interval shop scheduling. Working paper of RSM Erasmus University Rotterdam

Zhang X, Van de Velde S (2010c) On-line two-machine job shop scheduling with time lags. Information Processing Letters 110:510–513

Zhang X, Van de Velde S (2010d) On-line two-machine open shop scheduling with time lags. European Journal of Operational Research 204(1):14–19

Summary

Scheduling is the allocation of limited resources to activities, or jobs to be done, over time. The applications of scheduling theory can be found in many manufacturing and service industries, where production, transportation, distribution, procurement, information processing and communication are restricted by limited resources. This thesis focuses on a subset of scheduling problems where time lags play a role. Time lags are job characteristics, specifying the minimum time delay required between the execution of two consecutive operations of the same activity. Time lags can represent transportation delays, activities that require no limited resources, or intermediate processes between two bottleneck machines. The problems addressed in this study are motivated by the scheduling of barges that sail between terminals in a port area to load and unload cargos.

The research in this thesis addresses two types of problems, the on-line scheduling problems and the off-line scheduling problems. In on-line problems, the decision maker does not have complete information of the problem instance and therefore has to react to new job scheduling requests with a partial knowledge of the problem instance only. In off-line problems, the decision maker has all information available prior to making any scheduling decision.

The thesis presents the first results on on-line scheduling problems with time lags. For the basic on-line two-machine flow shop, job shop and open shop problems to minimize makespan, we show that the greedy algorithm has competitive ratio 2 and that the greedy algorithm is a best possible algorithm, if non-delay is required or the environment is non-clairvoyant. For the case that the environ-

ment is clairvoyant and deliberate delays are allowed, we give lower bounds on the best possible competitive ratios of any on-line algorithm for these problems. This raises the question whether there exists a delay algorithm with a better competitive ratio than the greedy algorithm for the clairvoyant variant.

For the open shop problem, we also analyze a semi-online case, where each time lag is smaller than any positive processing time. We show that the greedy algorithm is a best possible algorithm with competitive ratio $\frac{5}{3}$.

We analyze three sets of off-line scheduling problems with time lags. First, we analyze the NP-hard parallel flow shop problem with transportation delays to minimize the makespan. We design a $\frac{3}{2}$-approximation algorithm for the case of two parallel two-stage flow shops and a $\frac{12}{7}$-approximation algorithm for the case of three parallel two-stage flow shops. Second, we consider scheduling problems with time lags with the objective to minimize the makespan or total completion time, and we prove that for such problems Polynomial-Time Approximation Schemes can be developed if ratio between the largest time lag and the minimum positive processing time is bounded from above by a given fixed constant. These are the first PTAS results for scheduling problems with time lags. Third, we analyze the fixed interval shop scheduling problem, where the objective is to maximize the weighted number of jobs that can be processed by a shop, if each job has an a priori given interval in which it needs to be scheduled if it is selected for scheduling. We show that the problem is solvable in polynomial time in a flow shop environment in the case of equal time lags but NP-hard in the strong sense for general time lags, even in the case of unit-processing times. The fixed interval two-machine job and open shop problems are shown to be solvable in polynomial time if all time lags are equal and smaller than the processing time of any operation.

Summary in Dutch

Planning het toewijzen van een beperkte verzameling hulpmiddelen aan een verzameling van activiteiten over de tijd heen. Hulpmiddelen worden gewoonlijk aangeduid als machines en activiteiten als taken. Het doel daarbij is het vinden van een toegelaten, optimaal of bijna optimaal schedule voor een van te voren gespecificeerde doelstellingsfunctie. Een *schedule* is een toewijzing van taken aan machines over de tijd heen zodanig dat geen enkele machine meer dan een taak tegelijkertijd uitvoert, geen enkele taak op verschillende machines tegelijkertijd wordt verricht, en taken binnen de van te voren vastgestelde tijdvensters worden gedaan.

Planningsvraagstukken doen zich voor in haast alle organisaties, op het gebied van productie, dienstverlening, transport, distributie, inkoop, inzet van personeel, of verwerking van informatie. Hulpmiddelen kunnen dan machines, vrachtwagens, computers, of mensen zijn. In dit proefschrift richten we ons op planningsvraagstukken met minimaal vereiste vertragingen tussen twee opeenvolgende taken. Deze vraagstukken worden gemotiveerd door planningsvraagstukken die optreden bij het plannen van het laden en lossen van binnenvaartschepen (de taken) bij container terminals (de hulpmiddelen) in de Haven van Rotterdam, waarbij de minimaal vereiste vertragingen de vaartijden tussen container terminals vormen.

Het onderzoek in dit proefschrift richt zich op planningsvraagstukken met vertragingen in zowel statische als dynamische omgevingen. Een omgeving heet *statisch* als alle activiteiten en de daarbij behorende informatie van te voren bekend zijn. Dit leidt tot zogenaamde *off-line* problemen. Een omgeving heet *dynamisch* als niet alle activiteiten van te voren bekend zijn. Dit leidt tot zogenaamde *on-line* problemen. We zijn met name geinteresseerd in de *prestatiegarantie* van oplossingsmethoden. Een oplossingsmethode heeft prestatiegarantie ρ als het voor iedere willekeurige instantie van het probleem een schedule genereert waarvan de doelstellingswaarde gegarandeerd niet meer dan ρ maal de doelstellingwaarde van een optimaal schedule is.

Dit proefschrift presenteert de eerste prestatiegrantieresultaten voor on-line planningsvraagstukken met vertragingen. Voor het minimaliseren van de lengte van een schedule in een on-line omgeving met minimaal vereiste vertragingen laten we zien dat een zogenaamde *greedy* oplossingsmethode prestatiegarantie 2 heeft, voor zowel een flow shop, als een job shop, als een open shop met twee machines. We bewijzen dat een dergelijke greedy oplossingsmethode een beste mogelijke oplossingsmethode is; dat wil zeggen, voor deze vraagstukken bestaat er geen oplossingsmethode met een betere prestatiegarantie als een binnenkomende taak direct gealloceerd dient te worden of als we bij binnenkomst van de taak niet bekend wordt hoeveel tijd de taak vergt. Indien een taak niet direkt gealloceerd dient te worden en de duur van een taak bij binnenkomst wel bekend wordt, geven we algemene ondergrenzen op de prestatiegarantie. Geen enkele oplossingsmethode kan dan een lagere prestatiegarantie hebben. We kunnen echter niet uitsluiten dat er in deze situatie een oplossingsmethode bestaat die een lagere prestatiegarantie heeft dan de greedy oplossingsmethode. Voor het open shop probleem analyseren we ook de speciale situatie waarbij de lengte van elke vertraging kleiner of gelijk is aan de kleinste positieve verwerkingstijd van een taak. In dit geval is de prestatiegarantie van een greedy oplossingsmethode $\frac{5}{3}$.

We behandelen drie off-line planningsvraagstukken met vertragingen. Allereerst analyseren we het NP-lastige probleem om de lengte van een schedule te minimaliseren in zogenaamde *parallel flow shops.* We ontwerpen een oplossingsmethode met prestatiegarantie $\frac{3}{2}$ voor de situatie met twee parallelle flow shops die beiden uit twee machines bestaan. Voor de situatie met drie parallelle flow shops elk bestaand uit twee machines, geven we een oplossingsmethode met prestatiegarantie $\frac{12}{7}$. Ten tweede, laten we zien dat voor een ruime klasse van planningsvraagstukken met vertragingen om of de lengte van een schedule of om de gemiddelde doorlooptijd te minimaliseren zogeheten *Polynomial-Time*

Approximation Schemes (PTAS) bestaan, indien de ratio tussen de grootste vertraging en de minimale positieve verwerkingstijd wordt begrensd door een vooraf bepaalde constante wordt begrensd. Een PTAS is een familie van benaderende oplossingsmethoden die een polynomiale hoeveel tijd vergen die voor elk van te voren vastgestelde foutenmarge $\epsilon > 0$ een schedule geeft waarvan de doelstellingswaarde gegarandeerd niet meer dan $(1 + \epsilon)$ maal de doelstellingwaarde van een optimaal schedule is. Onze resultaten vormen de eerste PTAS resultaten voor planningsvraagstukken met vertragingen. Ten slotte, bekijken we het zogenaamde *fixed interval shop* planningsvraagstuk, waarbij elke taak een tijdvenster heeft waarin het dient uitgevoerd te worden en waarbij de lengte van het tijdvenster gelijk is aan de verwerkingstijd van de taak. Het doel is het aantal taken, gewogen naar hun mate van belangrijkheid, te maximaliseren. We tonen aan dat het probleem oplosbaar is indien alle vertragingen gelijk zijn, maar dat het NP-lastig is in de sterke zin in geval van algemene vertragingen, zelfs in het geval van gelijke verwerkingstijden.

About the Author

Xiandong Zhang was born on January 14th, 1971 in Xuzhou, Jiangsu Province of China. He earned a Bachelor degree in Industrial Management and Engineering from Southeastern University (Nanjing, China) in 1992, and a Ph.D. degree in Management Engineering from Tongji University (Shanghai, China) in 1997.

Xiandong Zhang has been employed at Fudan University (Shanghai, China) since 1997, as a post-doctoral researcher (1997-1999), a lecturer (1999-2000) and an associate professor (2000-). In January 2007, Xiandong Zhang started his second Ph.D. study at Erasmus University Rotterdam under the supervision of Professor Steef van de Velde. His research interests focus on machine scheduling and combinatorial optimization. Results from his Ph.D. research have appeared or have been accepted for publication in the *European Journal of Operational Research*, *Journal of Scheduling* and *Information Processing Letters*. He was a visiting scholar at Massachusetts Institute of Technology (Cambridge, USA) (September 2001--January 2002) and at Bocconi University (Milan, Italy) (February--June 2008).

ERASMUS RESEARCH INSTITUTE OF MANAGEMENT (ERIM)

ERIM PH.D. SERIES

RESEARCH IN MANAGEMENT

ERIM Electronic Series Portal: http://hdl.handle.net/1765/1

Agatz, N.A.H., *Demand Management in E-Fulfillment*, Promotor: Prof.dr.ir. J.A.E.E. van Nunen, EPS-2009-163-LIS, ISBN: 978-90-5892-200-7, http://hdl.handle.net/1765/15425

Althuizen, N.A.P., *Analogical Reasoning as a Decision Support Principle for Weakly Structured Marketing Problems*, Promotor: Prof.dr.ir. B. Wierenga, EPS-2006-095-MKT, ISBN: 90-5892-129-8, http://hdl.handle.net/1765/8190

Alvarez, H.L., *Distributed Collaborative Learning Communities Enabled by Information Communication Technology*, Promotor: Prof.dr. K. Kumar, EPS-2006-080-LIS, ISBN: 90-5892-112-3, http://hdl.handle.net/1765/7830

Appelman, J.H., *Governance of Global Interorganizational Tourism Networks: Changing Forms of Co-ordination between the Travel Agency and Aviation Sector*, Promotors: Prof.dr. F.M. Go & Prof.dr. B. Nooteboom, EPS-2004-036-MKT, ISBN: 90-5892-060-7, http://hdl.handle.net/1765/1199

Asperen, E. van, *Essays on Port, Container, and Bulk Chemical Logistics Optimization*, Promotor: Prof.dr.ir. R. Dekker, EPS-2009-181-LIS, ISBN: 978-90-5892-222-9, http://hdl.handle.net/1765/1

Assem, M.J. van den, *Deal or No Deal? Decision Making under Risk in a Large-Stake TV Game Show and Related Experiments*, Promotor: Prof.dr. J. Spronk, EPS-2008-138-F&A, ISBN: 978-90-5892-173-4, http://hdl.handle.net/1765/13566

Baquero, G, *On Hedge Fund Performance, Capital Flows and Investor Psychology*, Promotor: Prof.dr. M.J.C.M. Verbeek, EPS-2006-094-F&A, ISBN: 90-5892-131-X, http://hdl.handle.net/1765/8192

Berens, G., *Corporate Branding: The Development of Corporate Associations and their Influence on Stakeholder Reactions*, Promotor: Prof.dr. C.B.M. van Riel, EPS-2004-039-ORG, ISBN: 90-5892-065-8, http://hdl.handle.net/1765/1273

Berghe, D.A.F. van den, *Working Across Borders: Multinational Enterprises and the Internationalization of Employment*, Promotors: Prof.dr. R.J.M. van Tulder & Prof.dr. E.J.J. Schenk, EPS-2003-029-ORG, ISBN: 90-5892-05-34, http://hdl.handle.net/1765/1041

Berghman, L.A., *Strategic Innovation Capacity: A Mixed Method Study on Deliberate Strategic Learning Mechanisms*, Promotor: Prof.dr. P. Mattyssens, EPS-2006-087-MKT, ISBN: 90-5892-120-4, http://hdl.handle.net/1765/7991

Bezemer, P.J., *Diffusion of Corporate Governance Beliefs: Board Independence and the Emergence of a Shareholder Value Orientation in the Netherlands,* Promotors: Prof.dr.ing. F.A.J. van den Bosch & Prof.dr. H.W. Volberda, EPS-2009-192-STR, ISBN: 978-90-5892-232-8, http://hdl.handle.net/1765/1

Bijman, W.J.J., *Essays on Agricultural Co-operatives: Governance Structure in Fruit and Vegetable Chains*, Promotor: Prof.dr. G.W.J. Hendrikse, EPS-2002-015-ORG, ISBN: 90-5892-024-0, http://hdl.handle.net/1765/867

Bispo, A., *Labour Market Segmentation: An investigation into the Dutch hospitality industry*, Promotors: Prof.dr. G.H.M. Evers & Prof.dr. A.R. Thurik, EPS-2007-108-ORG, ISBN: 90-5892-136-9, http://hdl.handle.net/1765/10283

Blindenbach-Driessen, F., *Innovation Management in Project-Based Firms*, Promotor: Prof.dr. S.L. van de Velde, EPS-2006-082-LIS, ISBN: 90-5892-110-7, http://hdl.handle.net/1765/7828

Boer, C.A., *Distributed Simulation in Industry,* Promotors: Prof.dr. A. de Bruin & Prof.dr.ir. A. Verbraeck, EPS-2005-065-LIS, ISBN: 90-5892-093-3, http://hdl.handle.net/1765/6925

Boer, N.I., *Knowledge Sharing within Organizations: A situated and Relational Perspective,* Promotor: Prof.dr. K. Kumar, EPS-2005-060-LIS, ISBN: 90-5892-086-0, http://hdl.handle.net/1765/6770

Boer-Sorbán, K., *Agent-Based Simulation of Financial Markets: A modular, Continuous-Time Approach,* Promotor: Prof.dr. A. de Bruin, EPS-2008-119-LIS, ISBN: 90-5892-155-0, http://hdl.handle.net/1765/10870

Boon, C.T., *HRM and Fit: Survival of the Fittest!?,* Promotors: Prof.dr. J. Paauwe & Prof.dr. D.N. den Hartog, EPS-2008-129-ORG, ISBN: 978-90-5892-162-8, http://hdl.handle.net/1765/12606

Braun, E., *City Marketing: Towards an Integrated Approach,* Promotor: Prof.dr. L. van den Berg, EPS-2008-142-MKT, ISBN: 978-90-5892-180-2, http://hdl.handle.net/1765/13694

Brito, M.P. de, *Managing Reverse Logistics or Reversing Logistics Management?* Promotors: Prof.dr.ir. R. Dekker & Prof.dr. M. B. M. de Koster, EPS-2004-035-LIS, ISBN: 90-5892-058-5, http://hdl.handle.net/1765/1132

Brohm, R., *Polycentric Order in Organizations: A Dialogue between Michael Polanyi and IT-Consultants on Knowledge, Morality, and Organization,* Promotors: Prof.dr. G. W. J. Hendrikse & Prof.dr. H. K. Letiche, EPS-2005-063-ORG, ISBN: 90-5892-095-X, http://hdl.handle.net/1765/6911

Brumme, W.-H., *Manufacturing Capability Switching in the High-Tech Electronics Technology Life Cycle,* Promotors: Prof.dr.ir. J.A.E.E. van Nunen & Prof.dr.ir. L.N. Van Wassenhove, EPS-2008-126-LIS, ISBN: 978-90-5892-150-5, http://hdl.handle.net/1765/12103

Budiono, D.P., The Analysis of Mutual Fund Performance: Evidence from U.S. Equity Mutual Funds, Promotor: Prof.dr.ir. M.J.C.M. Verbeek, EPS-2010-185-F&A, ISBN: 978-90-5892-224-3, http://hdl.handle.net/1765/1

Burgers, J.H., *Managing Corporate Venturing: Multilevel Studies on Project Autonomy, Integration, Knowledge Relatedness, and Phases in the New Business Development Process,* Promotors: Prof.dr.ir. F.A.J. Van den Bosch & Prof.dr. H.W. Volberda, EPS-2008-136-STR, ISBN: 978-90-5892-174-1, http://hdl.handle.net/1765/13484

Campbell, R.A.J., *Rethinking Risk in International Financial Markets*, Promotor: Prof.dr. C.G. Koedijk, EPS-2001-005-F&A, ISBN: 90-5892-008-9, http://hdl.handle.net/1765/306

Chen, C.-M., *Evaluation and Design of Supply Chain Operations Using DEA*, Promotor: Prof.dr. J.A.E.E. van Nunen, EPS-2009-172-LIS, ISBN: 978-90-5892-209-0, http://hdl.handle.net/1765/1

Chen, H., *Individual Mobile Communication Services and Tariffs*, Promotor: Prof.dr. L.F.J.M. Pau, EPS-2008-123-LIS, ISBN: 90-5892-158-1, http://hdl.handle.net/1765/11141

Chen, Y., *Labour Flexibility in China's Companies: An Empirical Study*, Promotors: Prof.dr. A. Buitendam & Prof.dr. B. Krug, EPS-2001-006-ORG, ISBN: 90-5892-012-7, http://hdl.handle.net/1765/307

Damen, F.J.A., *Taking the Lead: The Role of Affect in Leadership Effectiveness*, Promotor: Prof.dr. D.L. van Knippenberg, EPS-2007-107-ORG, http://hdl.handle.net/1765/10282

Daniševská, P., *Empirical Studies on Financial Intermediation and Corporate Policies*, Promotor: Prof.dr. C.G. Koedijk, EPS-2004-044-F&A, ISBN: 90-5892-070-4, http://hdl.handle.net/1765/1518

Delporte-Vermeiren, D.J.E., *Improving the Flexibility and Profitability of ICT-enabled Business Networks: An Assessment Method and Tool*, Promotors: Prof. mr. dr. P.H.M. Vervest & Prof.dr.ir. H.W.G.M. van Heck, EPS-2003-020-LIS, ISBN: 90-5892-040-2, http://hdl.handle.net/1765/359

Derwall, J.M.M., *The Economic Virtues of SRI and CSR*, Promotor: Prof.dr. C.G. Koedijk, EPS-2007-101-F&A, ISBN: 90-5892-132-8, http://hdl.handle.net/1765/8986

Diepen, M. van, *Dynamics and Competition in Charitable Giving*, Promotor: Prof.dr. Ph.H.B.F. Franses, EPS-2009-159-MKT, ISBN: 978-90-5892-188-8, http://hdl.handle.net/1765/14526

Dietz, H.M.S., *Managing (Sales)People towards Perfomance: HR Strategy, Leadership & Teamwork*, Promotor: Prof.dr. G.W.J. Hendrikse, EPS-2009-168-ORG, ISBN: 978-90-5892-210-6, http://hdl.handle.net/1765/1

Dijksterhuis, M., *Organizational Dynamics of Cognition and Action in the Changing Dutch and US Banking Industries*, Promotors: Prof.dr.ir. F.A.J. van den Bosch & Prof.dr. H.W. Volberda, EPS-2003-026-STR, ISBN: 90-5892-048-8, http://hdl.handle.net/1765/1037

Eijk, A.R. van der, *Behind Networks: Knowledge Transfer, Favor Exchange and Performance*, Promotors: Prof.dr. S.L. van de Velde & Prof.dr.drs. W.A. Dolfsma, EPS-2009-161-LIS, ISBN: 978-90-5892-190-1, http://hdl.handle.net/1765/14613

Elstak, M.N., *Flipping the Identity Coin: The Comparative Effect of Perceived, Projected and Desired Organizational Identity on Organizational Identification and Desired Behavior*, Promotor: Prof.dr. C.B.M. van Riel, EPS-2008-117-ORG, ISBN: 90-5892-148-2, http://hdl.handle.net/1765/10723

Erken, H.P.G., *Productivity, R&D and Entrepreneurship*, Promotor: Prof.dr. A.R. Thurik, EPS-2008-147-ORG, ISBN: 978-90-5892-179-6, http://hdl.handle.net/1765/14004

Fenema, P.C. van, *Coordination and Control of Globally Distributed Software Projects*, Promotor: Prof.dr. K. Kumar, EPS-2002-019-LIS, ISBN: 90-5892-030-5, http://hdl.handle.net/1765/360

Fleischmann, M., *Quantitative Models for Reverse Logistics*, Promotors: Prof.dr.ir. J.A.E.E. van Nunen & Prof.dr.ir. R. Dekker, EPS-2000-002-LIS, ISBN: 35-4041-711-7, http://hdl.handle.net/1765/1044

Flier, B., *Strategic Renewal of European Financial Incumbents: Coevolution of Environmental Selection, Institutional Effects, and Managerial Intentionality*, Promotors: Prof.dr.ir. F.A.J. van den Bosch & Prof.dr. H.W. Volberda, EPS-2003-033-STR, ISBN: 90-5892-055-0, http://hdl.handle.net/1765/1071

Fok, D., *Advanced Econometric Marketing Models*, Promotor: Prof.dr. Ph.H.B.F. Franses, EPS-2003-027-MKT, ISBN: 90-5892-049-6, http://hdl.handle.net/1765/1035

Ganzaroli, A., *Creating Trust between Local and Global Systems*, Promotors: Prof.dr. K. Kumar & Prof.dr. R.M. Lee, EPS-2002-018-LIS, ISBN: 90-5892-031-3, http://hdl.handle.net/1765/361

Gertsen, H.F.M., *Riding a Tiger without Being Eaten: How Companies and Analysts Tame Financial Restatements and Influence Corporate Reputation*, Promotor: Prof.dr. C.B.M. van Riel, EPS-2009-171-ORG, ISBN: 90-5892-214-4, http://hdl.handle.net/1765/1

Gilsing, V.A., *Exploration, Exploitation and Co-evolution in Innovation Networks*, Promotors: Prof.dr. B. Nooteboom & Prof.dr. J.P.M. Groenewegen, EPS-2003-032-ORG, ISBN: 90-5892-054-2, http://hdl.handle.net/1765/1040

Gijsbers, G.W., *Agricultural Innovation in Asia: Drivers, Paradigms and Performance,* Promotor: Prof.dr. R.J.M. van Tulder, EPS-2009-156-ORG, ISBN: 978-90-5892-191-8, http://hdl.handle.net/1765/14524

Gong, Y., *Stochastic Modelling and Analysis of Warehouse Operations*, Promotors: Prof.dr. M.B.M. de Koster & Prof.dr. S.L. van de Velde, EPS-2009-180-LIS, ISBN: 978-90-5892-219-9, http://hdl.handle.net/1765/1

Govers, R., *Virtual Tourism Destination Image: Glocal Identities Constructed, Perceived and Experienced*, Promotors: Prof.dr. F.M. Go & Prof.dr. K. Kumar, EPS-2005-069-MKT, ISBN: 90-5892-107-7, http://hdl.handle.net/1765/6981

Graaf, G. de, *Tractable Morality: Customer Discourses of Bankers, Veterinarians and Charity Workers*, Promotors: Prof.dr. F. Leijnse & Prof.dr. T. van Willigenburg, EPS-2003-031-ORG, ISBN: 90-5892-051-8, http://hdl.handle.net/1765/1038

Greeven, M.J., *Innovation in an Uncertain Institutional Environment: Private Software Entrepreneurs in Hangzhou, China*, Promotor: Prof.dr. B. Krug, EPS-2009-164-ORG, ISBN: 978-90-5892-202-1, http://hdl.handle.net/1765/15426

Groot, E.A. de, *Essays on Economic Cycles*, Promotors: Prof.dr. Ph.H.B.F. Franses & Prof.dr. H.R. Commandeur, EPS-2006-091-MKT, ISBN: 90-5892-123-9, http://hdl.handle.net/1765/8216

Guenster, N.K., *Investment Strategies Based on Social Responsibility and Bubbles*, Promotor: Prof.dr. C.G. Koedijk, EPS-2008-175-F&A, ISBN: 978-90-5892-206-9, http://hdl.handle.net/1765/1

Gutkowska, A.B., *Essays on the Dynamic Portfolio Choice*, Promotor: Prof.dr. A.C.F. Vorst, EPS-2006-085-F&A, ISBN: 90-5892-118-2, http://hdl.handle.net/1765/7994

Hagemeijer, R.E., *The Unmasking of the Other*, Promotors: Prof.dr. S.J. Magala & Prof.dr. H.K. Letiche, EPS-2005-068-ORG, ISBN: 90-5892-097-6, http://hdl.handle.net/1765/6963

Hakimi, N.A, *Leader Empowering Behaviour: The Leader's Perspective: Understanding the Motivation behind Leader Empowering Behaviour*, Promotor: Prof.dr. D.L. van Knippenberg, EPS-2010-184-ORG, http://hdl.handle.net/1765/1

Halderen, M.D. van, *Organizational Identity Expressiveness and Perception Management: Principles for Expressing the Organizational Identity in Order to Manage the Perceptions and Behavioral Reactions of External Stakeholders*, Promotor: Prof.dr. S.B.M. van Riel, EPS-2008-122-ORG, ISBN: 90-5892-153-6, http://hdl.handle.net/1765/10872

Hartigh, E. den, *Increasing Returns and Firm Performance: An Empirical Study*, Promotor: Prof.dr. H.R. Commandeur, EPS-2005-067-STR, ISBN: 90-5892-098-4, http://hdl.handle.net/1765/6939

Hensmans, M., *A Republican Settlement Theory of the Firm: Applied to Retail Banks in England and the Netherlands (1830-2007)*, Promotors: Prof.dr. A. Jolink& Prof.dr. S.J. Magala, EPS-2010-193-ORG, ISBN 90-5892-235-9, http://hdl.handle.net/1765/1

Hermans. J.M., *ICT in Information Services; Use and Deployment of the Dutch Securities Trade, 1860-1970,* Promotor: Prof.dr. drs. F.H.A. Janszen, EPS-2004-046-ORG, ISBN 90-5892-072-0, http://hdl.handle.net/1765/1793

Hernandez Mireles, C., *Marketing Modeling for New Products,* Promotor: Prof.dr. P.H. Franses, EPS-2010-202-MKT, ISBN 90-5892-237-3, http://hdl.handle.net/1765/1

Hessels, S.J.A., *International Entrepreneurship: Value Creation Across National Borders*, Promotor: Prof.dr. A.R. Thurik, EPS-2008-144-ORG, ISBN: 978-90-5892-181-9, http://hdl.handle.net/1765/13942

Heugens, P.P.M.A.R., *Strategic Issues Management: Implications for Corporate Performance*, Promotors: Prof.dr.ir. F.A.J. van den Bosch & Prof.dr. C.B.M. van Riel, EPS-2001-007-STR, ISBN: 90-5892-009-9, http://hdl.handle.net/1765/358

Heuvel, W. van den, *The Economic Lot-Sizing Problem: New Results and Extensions*, Promotor: Prof.dr. A.P.L. Wagelmans, EPS-2006-093-LIS, ISBN: 90-5892-124-7, http://hdl.handle.net/1765/1805

Hoedemaekers, C.M.W., *Performance, Pinned down: A Lacanian Analysis of Subjectivity at Work*, Promotors: Prof.dr. S. Magala & Prof.dr. D.H. den Hartog, EPS-2008-121-ORG, ISBN: 90-5892-156-7, http://hdl.handle.net/1765/10871

Hooghiemstra, R., *The Construction of Reality: Cultural Differences in Self-serving Behaviour in Accounting Narratives*, Promotors: Prof.dr. L.G. van der Tas RA & Prof.dr. A.Th.H. Pruyn, EPS-2003-025-F&A, ISBN: 90-5892-047-X, http://hdl.handle.net/1765/871

Hu, Y., *Essays on the Governance of Agricultural Products: Cooperatives and Contract Farming*, Promotors: Prof.dr. G.W.J. Hendrkse & Prof.dr. B. Krug, EPS-2007-113-ORG, ISBN: 90-5892-145-1, http://hdl.handle.net/1765/10535

Huang, X., *An Analysis of Occupational Pension Provision: From Evaluation to Redesigh,* Promotors: Prof.dr. M.J.C.M. Verbeek & Prof.dr. R.J. Mahieu, EPS-2010-196-F&A, ISBN: 90-5892-239-7, http://hdl.handle.net/1765/1

Huij, J.J., *New Insights into Mutual Funds: Performance and Family Strategies*, Promotor: Prof.dr. M.C.J.M. Verbeek, EPS-2007-099-F&A, ISBN: 90-5892-134-4, http://hdl.handle.net/1765/9398

Huurman, C.I., *Dealing with Electricity Prices*, Promotor: Prof.dr. C.D. Koedijk, EPS-2007-098-F&A, ISBN: 90-5892-130-1, http://hdl.handle.net/1765/9399

Iastrebova, K, *Manager's Information Overload: The Impact of Coping Strategies on Decision-Making Performance*, Promotor: Prof.dr. H.G. van Dissel, EPS-2006-077-LIS, ISBN: 90-5892-111-5, http://hdl.handle.net/1765/7329

Iwaarden, J.D. van, *Changing Quality Controls: The Effects of Increasing Product Variety and Shortening Product Life Cycles,* Promotors: Prof.dr. B.G. Dale & Prof.dr. A.R.T. Williams, EPS-2006-084-ORG, ISBN: 90-5892-117-4, http://hdl.handle.net/1765/7992

Jansen, J.J.P., *Ambidextrous Organizations*, Promotors: Prof.dr.ir. F.A.J. Van den Bosch & Prof.dr. H.W. Volberda, EPS-2005-055-STR, ISBN: 90-5892-081-X, http://hdl.handle.net/1765/6774

Jaspers, F.P.H., *Organizing Systemic Innovation*, Promotor: Prof.dr.ir. J.C.M. van den Ende, EPS-2009-160-ORG, ISBN: 978-90-5892-197-), http://hdl.handle.net/1765/14974

Jennen, M.G.J., *Empirical Essays on Office Market Dynamics*, Promotors: Prof.dr. C.G. Koedijk & Prof.dr. D. Brounen, EPS-2008-140-F&A, ISBN: 978-90-5892-176-5, http://hdl.handle.net/1765/13692

Jiang, T., *Capital Structure Determinants and Governance Structure Variety in Franchising*, Promotors: Prof.dr. G. Hendrikse & Prof.dr. A. de Jong, EPS-2009-158-F&A, ISBN: 978-90-5892-199-4, http://hdl.handle.net/1765/14975

Jiao, T., *Essays in Financial Accounting*, Promotor: Prof.dr. G.M.H. Mertens, EPS-2009-176-F&A, ISBN: 978-90-5892-211-3, http://hdl.handle.net/1765/1

Jong, C. de, *Dealing with Derivatives: Studies on the Role, Informational Content and Pricing of Financial Derivatives*, Promotor: Prof.dr. C.G. Koedijk, EPS-2003-023-F&A, ISBN: 90-5892-043-7, http://hdl.handle.net/1765/1043

Kaa, G. van, *Standard Battles for Complex Systems: Empirical Research on the Home Network,* Promotors: Prof.dr.ir. J. van den Ende & Prof.dr.ir. H.W.G.M. van Heck, EPS-2009-166-ORG, ISBN: 978-90-5892-205-2, http://hdl.handle.net/1765/1

Kagie, M., *Advances in Online Shopping Interfaces: Product Catalog Maps and Recommender Systems,* Promotor: Prof.dr. P.J.F. Groenen, EPS-2010-195-MKT, ISBN: 978-90-5892-233-5, http://hdl.handle.net/1765/1

Keizer, A.B., *The Changing Logic of Japanese Employment Practices: A Firm-Level Analysis of Four Industries,* Promotors: Prof.dr. J.A. Stam & Prof.dr. J.P.M. Groenewegen, EPS-2005-057-ORG, ISBN: 90-5892-087-9, http://hdl.handle.net/1765/6667

Kijkuit, R.C., *Social Networks in the Front End: The Organizational Life of an Idea,* Promotor: Prof.dr. B. Nooteboom, EPS-2007-104-ORG, ISBN: 90-5892-137-6, http://hdl.handle.net/1765/10074

Kippers, J., *Empirical Studies on Cash Payments,* Promotor: Prof.dr. Ph.H.B.F. Franses, EPS-2004-043-F&A, ISBN: 90-5892-069-0, http://hdl.handle.net/1765/1520

Klein, M.H., *Poverty Alleviation through Sustainable Strategic Business Models: Essays on Poverty Alleviation as a Business Strategy,* Promotor: Prof.dr. H.R. Commandeur, EPS-2008-135-STR, ISBN: 978-90-5892-168-0, http://hdl.handle.net/1765/13482

Knapp, S., *The Econometrics of Maritime Safety: Recommendations to Enhance Safety at Sea*, Promotor: Prof.dr. Ph.H.B.F. Franses, EPS-2007-096-ORG, ISBN: 90-5892-127-1, http://hdl.handle.net/1765/7913

Kole, E., *On Crises, Crashes and Comovements*, Promotors: Prof.dr. C.G. Koedijk & Prof.dr. M.J.C.M. Verbeek, EPS-2006-083-F&A, ISBN: 90-5892-114-X, http://hdl.handle.net/1765/7829

Kooij-de Bode, J.M., *Distributed Information and Group Decision-Making: Effects of Diversity and Affect*, Promotor: Prof.dr. D.L. van Knippenberg, EPS-2007-115-ORG, http://hdl.handle.net/1765/10722

Koppius, O.R., *Information Architecture and Electronic Market Performance*, Promotors: Prof.dr. P.H.M. Vervest & Prof.dr.ir. H.W.G.M. van Heck, EPS-2002-013-LIS, ISBN: 90-5892-023-2, http://hdl.handle.net/1765/921

Kotlarsky, J., *Management of Globally Distributed Component-Based Software Development Projects,* Promotor: Prof.dr. K. Kumar, EPS-2005-059-LIS, ISBN: 90-5892-088-7, http://hdl.handle.net/1765/6772

Krauth, E.I., *Real-Time Planning Support: A Task-Technology Fit Perspective,* Promotors: Prof.dr. S.L. van de Velde & Prof.dr. J. van Hillegersberg, EPS-2008-155-LIS, ISBN: 978-90-5892-193-2, http://hdl.handle.net/1765/14521

Kuilman, J., *The Re-Emergence of Foreign Banks in Shanghai: An Ecological Analysis,* Promotor: Prof.dr. B. Krug, EPS-2005-066-ORG, ISBN: 90-5892-096-8, http://hdl.handle.net/1765/6926

Kwee, Z., *Investigating Three Key Principles of Sustained Strategic Renewal: A Longitudinal Study of Long-Lived Firms,* Promotors: Prof.dr.ir. F.A.J. Van den Bosch & Prof.dr. H.W. Volberda, EPS-2009-174-STR, ISBN: 90-5892-212-0, http://hdl.handle.net/1765/1

Langen, P.W. de, *The Performance of Seaport Clusters: A Framework to Analyze Cluster Performance and an Application to the Seaport Clusters of Durban, Rotterdam and the Lower Mississippi*, Promotors: Prof.dr. B. Nooteboom & Prof. drs. H.W.H. Welters, EPS-2004-034-LIS, ISBN: 90-5892-056-9, http://hdl.handle.net/1765/1133

Le Anh, T., *Intelligent Control of Vehicle-Based Internal Transport Systems*, Promotors: Prof.dr. M.B.M. de Koster & Prof.dr.ir. R. Dekker, EPS-2005-051-LIS, ISBN: 90-5892-079-8, http://hdl.handle.net/1765/6554

Le-Duc, T., *Design and Control of Efficient Order Picking Processes*, Promotor: Prof.dr. M.B.M. de Koster, EPS-2005-064-LIS, ISBN: 90-5892-094-1, http://hdl.handle.net/1765/6910

Leeuwen, E.P. van, *Recovered-Resource Dependent Industries and the Strategic Renewal of Incumbent Firm: A Multi-Level Study of Recovered Resource Dependence Management and Strategic Renewal in the European Paper and Board Industry*, Promotors: Prof.dr.ir. F.A.J. Van den Bosch & Prof.dr. H.W. Volberda, EPS-2007-109-STR, ISBN: 90-5892-140-6, http://hdl.handle.net/1765/10183

Lentink, R.M., *Algorithmic Decision Support for Shunt Planning*, Promotors: Prof.dr. L.G. Kroon & Prof.dr.ir. J.A.E.E. van Nunen, EPS-2006-073-LIS, ISBN: 90-5892-104-2, http://hdl.handle.net/1765/7328

Li, T., *Informedness and Customer-Centric Revenue Management,* Promotors: Prof.dr. P.H.M. Vervest & Prof.dr.ir. H.W.G.M. van Heck, EPS-2009-146-LIS, ISBN: 978-90-5892-195-6, http://hdl.handle.net/1765/14525

Liang, G., *New Competition: Foreign Direct Investment and Industrial Development in China,* Promotor: Prof.dr. R.J.M. van Tulder, EPS-2004-047-ORG, ISBN: 90-5892-073-9, http://hdl.handle.net/1765/1795

Liere, D.W. van, *Network Horizon and the Dynamics of Network Positions: A Multi-Method Multi-Level Longitudinal Study of Interfirm Networks,* Promotor: Prof.dr. P.H.M. Vervest, EPS-2007-105-LIS, ISBN: 90-5892-139-0, http://hdl.handle.net/1765/10181

Loef, J., *Incongruity between Ads and Consumer Expectations of Advertising*, Promotors: Prof.dr. W.F. van Raaij & Prof.dr. G. Antonides, EPS-2002-017-MKT, ISBN: 90-5892-028-3, http://hdl.handle.net/1765/869

Maeseneire, W., de, *Essays on Firm Valuation and Value Appropriation,* Promotor: Prof.dr. J.T.J. Smit, EPS-2005-053-F&A, ISBN: 90-5892-082-8, http://hdl.handle.net/1765/6768

Londoño, M. del Pilar, *Institutional Arrangements that Affect Free Trade Agreements: Economic Rationality Versus Interest Groups*, Promotors: Prof.dr. H.E. Haralambides & Prof.dr. J.F. Francois, EPS-2006-078-LIS, ISBN: 90-5892-108-5, http://hdl.handle.net/1765/7578

Maas, A.A., van der, *Strategy Implementation in a Small Island Context: An Integrative Framework,* Promotor: Prof.dr. H.G. van Dissel, EPS-2008-127-LIS, ISBN: 978-90-5892-160-4, http://hdl.handle.net/1765/12278

Maas, K.E.G., *Corporate Social Performance: From Output Measurement to Impact Measurement,* Promotor: Prof.dr. H.R. Commandeur, EPS-2009-182-STR, ISBN: 978-90-5892-225-0, http://hdl.handle.net/1765/1

Maeseneire, W., de, *Essays on Firm Valuation and Value Appropriation,* Promotor: Prof.dr. J.T.J. Smit, EPS-2005-053-F&A, ISBN: 90-5892-082-8, http://hdl.handle.net/1765/6768

Mandele, L.M., van der, *Leadership and the Inflection Point: A Longitudinal Perspective*, Promotors: Prof.dr. H.W. Volberda & Prof.dr. H.R. Commandeur, EPS-2004-042-STR, ISBN: 90-5892-067-4, http://hdl.handle.net/1765/1302

Meer, J.R. van der, *Operational Control of Internal Transport*, Promotors: Prof.dr. M.B.M. de Koster & Prof.dr.ir. R. Dekker, EPS-2000-001-LIS, ISBN: 90-5892-004-6, http://hdl.handle.net/1765/859

Mentink, A., *Essays on Corporate Bonds*, Promotor: Prof.dr. A.C.F. Vorst, EPS-2005-070-F&A, ISBN: 90-5892-100-X, http://hdl.handle.net/1765/7121

Meyer, R.J.H., *Mapping the Mind of the Strategist: A Quantitative Methodology for Measuring the Strategic Beliefs of Executives*, Promotor: Prof.dr. R.J.M. van Tulder, EPS-2007-106-ORG, ISBN: 978-90-5892-141-3, http://hdl.handle.net/1765/10182

Miltenburg, P.R., *Effects of Modular Sourcing on Manufacturing Flexibility in the Automotive Industry: A Study among German OEMs*, Promotors: Prof.dr. J. Paauwe & Prof.dr. H.R. Commandeur, EPS-2003-030-ORG, ISBN: 90-5892-052-6, http://hdl.handle.net/1765/1039

Moerman, G.A., *Empirical Studies on Asset Pricing and Banking in the Euro Area,* Promotor: Prof.dr. C.G. Koedijk, EPS-2005-058-F&A, ISBN: 90-5892-090-9, http://hdl.handle.net/1765/6666

Moitra, D., *Globalization of R&D: Leveraging Offshoring for Innovative Capability and Organizational Flexibility,* Promotor: Prof.dr. K. Kumar, EPS-2008-150-LIS, ISBN: 978-90-5892-184-0, http://hdl.handle.net/1765/14081

Mol, M.M., *Outsourcing, Supplier-relations and Internationalisation: Global Source Strategy as a Chinese Puzzle*, Promotor: Prof.dr. R.J.M. van Tulder, EPS-2001-010-ORG, ISBN: 90-5892-014-3, http://hdl.handle.net/1765/355

Mom, T.J.M., *Managers' Exploration and Exploitation Activities: The Influence of Organizational Factors and Knowledge Inflows*, Promotors: Prof.dr.ir. F.A.J. Van den Bosch & Prof.dr. H.W. Volberda, EPS-2006-079-STR, ISBN: 90-5892-116-6, http://hdl.handle.net/1765

Moonen, J.M., *Multi-Agent Systems for Transportation Planning and Coordination*, Promotors: Prof.dr. J. van Hillegersberg & Prof.dr. S.L. van de Velde, EPS-2009-177-LIS, ISBN: 978-90-5892-216-8, http://hdl.handle.net/1

Mulder, A., *Government Dilemmas in the Private Provision of Public Goods*, Promotor: Prof.dr. R.J.M. van Tulder, EPS-2004-045-ORG, ISBN: 90-5892-071-2, http://hdl.handle.net/1765/1790

Muller, A.R., *The Rise of Regionalism: Core Company Strategies Under The Second Wave of Integration*, Promotor: Prof.dr. R.J.M. van Tulder, EPS-2004-038-ORG, ISBN: 90-5892-062-3, http://hdl.handle.net/1765/1272

Nalbantov G.I., *Essays on Some Recent Penalization Methods with Applications in Finance and Marketing*, Promotor: Prof. dr P.J.F. Groenen, EPS-2008-132-F&A, ISBN: 978-90-5892-166-6, http://hdl.handle.net/1765/13319

Nederveen Pieterse, A., *Goal Orientation in Teams: The Role of Diversity*, Promotor: Prof.dr. D.L. van Knippenberg, EPS-2009-162-ORG, http://hdl.handle.net/1765/15240

Nguyen, T.T., *Capital Structure, Strategic Competition, and Governance*, Promotor: Prof.dr. A. de Jong, EPS-2008-148-F&A, ISBN: 90-5892-178-9, http://hdl.handle.net/1765/14005

Niesten, E.M.M.I., *Regulation, Governance and Adaptation: Governance Transformations in the Dutch and French Liberalizing Electricity Industries*, Promotors: Prof.dr. A. Jolink & Prof.dr. J.P.M. Groenewegen, EPS-2009-170-ORG, ISBN: 978-90-5892-208-3, http://hdl.handle.net/1765/1

Nieuwenboer, N.A. den, *Seeing the Shadow of the Self*, Promotor: Prof.dr. S.P. Kaptein, EPS-2008-151-ORG, ISBN: 978-90-5892-182-6, http://hdl.handle.net/1765/14223

Ning, H., *Hierarchical Portfolio Management: Theory and Applications*, Promotor: Prof.dr. J. Spronk, EPS-2007-118-F&A, ISBN: 90-5892-152-9, http://hdl.handle.net/1765/10868

Noeverman, J., *Management Control Systems, Evaluative Style, and Behaviour: Exploring the Concept and Behavioural Consequences of Evaluative Style*, Promotors: Prof.dr. E.G.J. Vosselman & Prof.dr. A.R.T. Williams, EPS-2007-120-F&A, ISBN: 90-5892-151-2, http://hdl.handle.net/1765/10869

Nuijten, I., *Servant Leadership: Paradox or Diamond in the Rough? A Multidimensional Measure and Empirical Evidence*, Promotor: Prof.dr. D.L. van Knippenberg, EPS-2009-183-ORG, http://hdl.handle.net/1765/1

Oosterhout, J., van, *The Quest for Legitimacy: On Authority and Responsibility in Governance*, Promotors: Prof.dr. T. van Willigenburg & Prof.mr. H.R. van Gunsteren, EPS-2002-012-ORG, ISBN: 90-5892-022-4, http://hdl.handle.net/1765/362

Oosterhout, M., van, *Business Agility and Information Technology in Service Organizations,* Promotor: Prof,dr.ir. H.W.G.M. van Heck, EPS-2010-198-LIS, ISBN: 90-5092-236-6, http://hdl.handle.net/1765/1

Oostrum, J.M., van, *Applying Mathematical Models to Surgical Patient Planning*, Promotor: Prof.dr. A.P.M. Wagelmans, EPS-2009-179-LIS, ISBN: 978-90-5892-217-5, http://hdl.handle.net/1765/1

Paape, L., *Corporate Governance: The Impact on the Role, Position, and Scope of Services of the Internal Audit Function*, Promotors: Prof.dr. G.J. van der Pijl & Prof.dr. H. Commandeur, EPS-2007-111-MKT, ISBN: 90-5892-143-7, http://hdl.handle.net/1765/10417

Pak, K., *Revenue Management: New Features and Models*, Promotor: Prof.dr.ir. R. Dekker, EPS-2005-061-LIS, ISBN: 90-5892-092-5, http://hdl.handle.net/1765/362/6771

Pattikawa, L.H, *Innovation in the Pharmaceutical Industry: Evidence from Drug Introduction in the U.S.*, Promotors: Prof.dr. H.R.Commandeur, EPS-2007-102-MKT, ISBN: 90-5892-135-2, http://hdl.handle.net/1765/9626

Peeters, L.W.P., *Cyclic Railway Timetable Optimization*, Promotors: Prof.dr. L.G. Kroon & Prof.dr.ir. J.A.E.E. van Nunen, EPS-2003-022-LIS, ISBN: 90-5892-042-9, http://hdl.handle.net/1765/429

Pietersz, R., *Pricing Models for Bermudan-style Interest Rate Derivatives*, Promotors: Prof.dr. A.A.J. Pelsser & Prof.dr. A.C.F. Vorst, EPS-2005-071-F&A, ISBN: 90-5892-099-2, http://hdl.handle.net/1765/7122

Poel, A.M. van der, *Empirical Essays in Corporate Finance and Financial Reporting*, Promotors: Prof.dr. A. de Jong & Prof.dr. G.M.H. Mertens, EPS-2007-133-F&A, ISBN: 978-90-5892-165-9, http://hdl.handle.net/1765/13320

Popova, V., *Knowledge Discovery and Monotonicity*, Promotor: Prof.dr. A. de Bruin, EPS-2004-037-LIS, ISBN: 90-5892-061-5, http://hdl.handle.net/1765/1201

Pouchkarev, I., *Performance Evaluation of Constrained Portfolios*, Promotors: Prof.dr. J. Spronk & Dr. W.G.P.M. Hallerbach, EPS-2005-052-F&A, ISBN: 90-5892-083-6, http://hdl.handle.net/1765/6731

Prins, R., *Modeling Consumer Adoption and Usage of Value-Added Mobile Services*, Promotors: Prof.dr. Ph.H.B.F. Franses & Prof.dr. P.C. Verhoef, EPS-2008-128-MKT, ISBN: 978/90-5892-161-1, http://hdl.handle.net/1765/12461

Puvanasvari Ratnasingam, P., *Interorganizational Trust in Business to Business E-Commerce*, Promotors: Prof.dr. K. Kumar & Prof.dr. H.G. van Dissel, EPS-2001-009-LIS, ISBN: 90-5892-017-8, http://hdl.handle.net/1765/356

Quak, H.J., *Sustainability of Urban Freight Transport: Retail Distribution and Local Regulation in Cities*, Promotor: Prof.dr. M.B.M. de Koster, EPS-2008-124-LIS, ISBN: 978-90-5892-154-3, http://hdl.handle.net/1765/11990

Quariguasi Frota Neto, J., *Eco-efficient Supply Chains for Electrical and Electronic Products*, Promotors: Prof.dr.ir. J.A.E.E. van Nunen & Prof.dr.ir. H.W.G.M. van Heck, EPS-2008-152-LIS, ISBN: 978-90-5892-192-5, http://hdl.handle.net/1765/14785

Radkevitch, U.L, *Online Reverse Auction for Procurement of Services*, Promotor: Prof.dr.ir. H.W.G.M. van Heck, EPS-2008-137-LIS, ISBN: 978-90-5892-171-0, http://hdl.handle.net/1765/13497

Rinsum, M. van, *Performance Measurement and Managerial Time Orientation*, Promotor: Prof.dr. F.G.H. Hartmann, EPS-2006-088-F&A, ISBN: 90-5892-121-2, http://hdl.handle.net/1765/7993

Roelofsen, E.M., *The Role of Analyst Conference Calls in Capital Markets*, Promotors: Prof.dr. G.M.H. Mertens & Prof.dr. L.G. van der Tas RA, EPS-2010-190-F&A, ISBN: 978-90-5892-228-1, http://hdl.handle.net/1765/1

Romero Morales, D., *Optimization Problems in Supply Chain Management*, Promotors: Prof.dr.ir. J.A.E.E. van Nunen & Dr. H.E. Romeijn, EPS-2000-003-LIS, ISBN: 90-9014078-6, http://hdl.handle.net/1765/865

Roodbergen, K.J., *Layout and Routing Methods for Warehouses*, Promotors: Prof.dr. M.B.M. de Koster & Prof.dr.ir. J.A.E.E. van Nunen, EPS-2001-004-LIS, ISBN: 90-5892-005-4, http://hdl.handle.net/1765/861

Rook, L., *Imitation in Creative Task Performance*, Promotor: Prof.dr. D.L. van Knippenberg, EPS-2008-125-ORG, http://hdl.handle.net/1765/11555

Rosmalen, J. van, *Segmentation and Dimension Reduction: Exploratory and Model-Based Approaches*, Promotor: Prof.dr. P.J.F. Groenen, EPS-2009-165-MKT, ISBN: 978-90-5892-201-4, http://hdl.handle.net/1765/15536

Rus, D., *The Dark Side of Leadership: Exploring the Psychology of Leader Self-serving Behavior*, Promotor: Prof.dr. D.L. van Knippenberg, EPS-2009-178-ORG, http://hdl.handle.net/1765/1

Samii, R., *Leveraging Logistics Partnerships: Lessons from Humanitarian Organizations,* Promotors: Prof.dr.ir. J.A.E.E. van Nunen & Prof.dr.ir. L.N. Van Wassenhove, EPS-2008-153-LIS, ISBN: 978-90-5892-186-4, http://hdl.handle.net/1765/14519

Schaik, D. van, *M&A in Japan: An Analysis of Merger Waves and Hostile Takeovers,* Promotors: Prof.dr. J. Spronk & Prof.dr. J.P.M. Groenewegen, EPS-2008-141-F&A, ISBN: 978-90-5892-169-7, http://hdl.handle.net/1765/13693

Schauten, M.B.J., *Valuation, Capital Structure Decisions and the Cost of Capital,* Promotors: Prof.dr. J. Spronk & Prof.dr. D. van Dijk, EPS-2008-134-F&A, ISBN: 978-90-5892-172-7, http://hdl.handle.net/1765/13480

Schramade, W.L.J., *Corporate Bonds Issuers,* Promotor: Prof.dr. A. De Jong, EPS-2006-092-F&A, ISBN: 90-5892-125-5, http://hdl.handle.net/1765/8191

Schweizer, T.S., *An Individual Psychology of Novelty-Seeking, Creativity and Innovation,* Promotor: Prof.dr. R.J.M. van Tulder, EPS-2004-048-ORG, ISBN: 90-5892-077-1, http://hdl.handle.net/1765/1818

Six, F.E., *Trust and Trouble: Building Interpersonal Trust Within Organizations*, Promotors: Prof.dr. B. Nooteboom & Prof.dr. A.M. Sorge, EPS-2004-040-ORG, ISBN: 90-5892-064-X, http://hdl.handle.net/1765/1271

Slager, A.M.H., *Banking across Borders*, Promotors: Prof.dr. R.J.M. van Tulder & Prof.dr. D.M.N. van Wensveen, EPS-2004-041-ORG, ISBN: 90-5892-066–6, http://hdl.handle.net/1765/1301

Sloot, L., *Understanding Consumer Reactions to Assortment Unavailability*, Promotors: Prof.dr. H.R. Commandeur, Prof.dr. E. Peelen & Prof.dr. P.C. Verhoef, EPS-2006-074-MKT, ISBN: 90-5892-102-6, http://hdl.handle.net/1765/7438

Smit, W., *Market Information Sharing in Channel Relationships: Its Nature, Antecedents and Consequences*, Promotors: Prof.dr.ir. G.H. van Bruggen & Prof.dr.ir. B. Wierenga, EPS-2006-076-MKT, ISBN: 90-5892-106-9, http://hdl.handle.net/1765/7327

Sonnenberg, M., *The Signalling Effect of HRM on Psychological Contracts of Employees: A Multi-level Perspective*, Promotor: Prof.dr. J. Paauwe, EPS-2006-086-ORG, ISBN: 90-5892-119-0, http://hdl.handle.net/1765/7995

Speklé, R.F., *Beyond Generics: A closer Look at Hybrid and Hierarchical Governance*, Promotor: Prof.dr. M.A. van Hoepen RA, EPS-2001-008-F&A, ISBN: 90-5892-011-9, http://hdl.handle.net/1765/357

Srour, F.J., *Dissecting Drayage: An Examination of Structure, Information, and Control in Drayage Operations*, Promotor: Prof.dr. S.L. van de Velde, EPS-2010-186-LIS, http://hdl.handle.net/1765/1

Stam, D.A., *Managing Dreams and Ambitions: A Psychological Analysis of Vision Communication*, Promotor: Prof.dr. D.L. van Knippenberg, EPS-2008-149-ORG, http://hdl.handle.net/1765/14080

Stienstra, M., *Strategic Renewal in Regulatory Environments: How Inter- and Intra-organisational Institutional Forces Influence European Energy Incumbent Firms,* Promotors: Prof.dr.ir. F.A.J. Van den Bosch & Prof.dr. H.W. Volberda, EPS-2008-145-STR, ISBN: 978-90-5892-184-0, http://hdl.handle.net/1765/13943

Sweldens, S.T.L.R., *Evaluative Conditioning 2.0: Direct versus Associative Transfer of Affect to Brands,* Promotor: Prof.dr. S.M.J. van Osselaer, EPS-2009-167-MKT, ISBN: 978-90-5892-204-5, http://hdl.handle.net/1765/1

Szkudlarek, B.A*., Spinning the Web of Reentry: [Re]connecting reentry training theory and practice*, Promotor: Prof.dr. S.J. Magala, EPS-2008-143-ORG, ISBN: 978-90-5892-177-2, http://hdl.handle.net/1765/13695

Tempelaar, M.P., *Organizing for Ambidexterity: Studies on the Pursuit of Exploration and Exploitation through Differentiation, Integration, Contextual and Individual Attributes,* Promotors: Prof.dr.ing. F.A.J. van den Bosch & Prof.dr. H.W. Volberda, EPS-2010-191-STR, ISBN: 978-90-5892-231-1, http://hdl.handle.net/1765/1

Teunter, L.H., *Analysis of Sales Promotion Effects on Household Purchase Behavior*, Promotors: Prof.dr.ir. B. Wierenga & Prof.dr. T. Kloek, EPS-2002-016-MKT, ISBN: 90-5892-029-1, http://hdl.handle.net/1765/868

Tims, B., *Empirical Studies on Exchange Rate Puzzles and Volatility*, Promotor: Prof.dr. C.G. Koedijk, EPS-2006-089-F&A, ISBN: 90-5892-113-1, http://hdl.handle.net/1765/8066

Tuk, M.A., *Is Friendship Silent When Money Talks? How Consumers Respond to Word-of-Mouth Marketing*, Promotors: Prof.dr.ir. A. Smidts & Prof.dr. D.H.J. Wigboldus, EPS-2008-130-MKT, ISBN: 978-90-5892-164-2, http://hdl.handle.net/1765/12702

Valck, K. de, *Virtual Communities of Consumption: Networks of Consumer Knowledge and Companionship*, Promotors: Prof.dr.ir. G.H. van Bruggen & Prof.dr.ir. B. Wierenga, EPS-2005-050-MKT, ISBN: 90-5892-078-X, http://hdl.handle.net/1765/6663

Valk, W. van der, *Buyer-Seller Interaction Patterns During Ongoing Service Exchange*, Promotors: Prof.dr. J.Y.F. Wynstra & Prof.dr.ir. B. Axelsson, EPS-2007-116-MKT, ISBN: 90-5892-146-8, http://hdl.handle.net/1765/10856

Verheijen, H.J.J., *Vendor-Buyer Coordination in Supply Chains,* Promotor: Prof.dr.ir. J.A.E.E. van Nunen, EPS-2010-194-LIS, ISBN: 90-5892-234-2, http://hdl.handle.net/1765/1

Verheul, I., *Is There a (Fe)male Approach? Understanding Gender Differences in Entrepreneurship*, Prof.dr. A.R. Thurik, EPS-2005-054-ORG, ISBN: 90-5892-080-1, http://hdl.handle.net/1765/2005

Verwijmeren, P., *Empirical Essays on Debt, Equity, and Convertible Securities*, Promotors: Prof.dr. A. de Jong & Prof.dr. M.J.C.M. Verbeek, EPS-2009-154-F&A, ISBN: 978-90-5892-187-1, http://hdl.handle.net/1765/14312

Vis, I.F.A., *Planning and Control Concepts for Material Handling Systems*, Promotors: Prof.dr. M.B.M. de Koster & Prof.dr.ir. R. Dekker, EPS-2002-014-LIS, ISBN: 90-5892-021-6, http://hdl.handle.net/1765/866

Vlaar, P.W.L., *Making Sense of Formalization in Interorganizational Relationships: Beyond Coordination and Control,* Promotors: Prof.dr.ir. F.A.J. Van den Bosch & Prof.dr. H.W. Volberda, EPS-2006-075-STR, ISBN 90-5892-103-4, http://hdl.handle.net/1765/7326

Vliet, P. van, *Downside Risk and Empirical Asset Pricing*, Promotor: Prof.dr. G.T. Post, EPS-2004-049-F&A, ISBN: 90-5892-07-55, http://hdl.handle.net/1765/1819

Vlist, P. van der, *Synchronizing the Retail Supply Chain*, Promotors: Prof.dr.ir. J.A.E.E. van Nunen & Prof.dr. A.G. de Kok, EPS-2007-110-LIS, ISBN: 90-5892-142-0, http://hdl.handle.net/1765/10418

Vries-van Ketel E. de, *How Assortment Variety Affects Assortment Attractiveness: A Consumer Perspective*, Promotors: Prof.dr. G.H. van Bruggen & Prof.dr.ir. A. Smidts, EPS-2006-072-MKT, ISBN: 90-5892-101-8, http://hdl.handle.net/1765/7193

Vromans, M.J.C.M., *Reliability of Railway Systems*, Promotors: Prof.dr. L.G. Kroon, Prof.dr.ir. R. Dekker & Prof.dr.ir. J.A.E.E. van Nunen, EPS-2005-062-LIS, ISBN: 90-5892-089-5, http://hdl.handle.net/1765/6773

Vroomen, B.L.K., *The Effects of the Internet, Recommendation Quality and Decision Strategies on Consumer Choice*, Promotor: Prof.dr. Ph.H.B.F. Franses, EPS-2006-090-MKT, ISBN: 90-5892-122-0, http://hdl.handle.net/1765/8067

Waal, T. de, *Processing of Erroneous and Unsafe Data*, Promotor: Prof.dr.ir. R. Dekker, EPS-2003-024-LIS, ISBN: 90-5892-045-3, http://hdl.handle.net/1765/870

Waard, E.J. de, *Engaging Environmental Turbulence: Organizational Determinants for Repetitive Quick and Adequate Responses*, Promotors: Prof.dr. H.W. Volberda & Prof.dr. J. Soeters, EPS-2010-189-STR, ISBN: 978-90-5892-229-8, http://hdl.handle.net/1765/1

Wall, R.S., *Netscape: Cities and Global Corporate Networks,* Promotor: Prof.dr. G.A. van der Knaap, EPS-2009-169-ORG, ISBN: 978-90-5892-207-6, http://hdl.handle.net/1765/1

Watkins Fassler, K., *Macroeconomic Crisis and Firm Performance,* Promotors: Prof.dr. J. Spronk & Prof.dr. D.J. van Dijk, EPS-2007-103-F&A, ISBN: 90-5892-138-3, http://hdl.handle.net/1765/10065

Weerdt, N.P. van der, *Organizational Flexibility for Hypercompetitive Markets: Empirical Evidence of the Composition and Context Specificity of Dynamic Capabilities and Organization Design Parameters,* Promotor: Prof.dr. H.W. Volberda, EPS-2009-173-STR, ISBN: 978-90-5892-215-1, http://hdl.handle.net/1765/1

Wennekers, A.R.M., *Entrepreneurship at Country Level: Economic and Non-Economic Determinants,* Promotor: Prof.dr. A.R. Thurik, EPS-2006-81-ORG, ISBN: 90-5892-115-8, http://hdl.handle.net/1765/7982

Wielemaker, M.W., *Managing Initiatives: A Synthesis of the Conditioning and Knowledge-Creating View*, Promotors: Prof.dr. H.W. Volberda & Prof.dr. C.W.F. Baden-Fuller, EPS-2003-28-STR, ISBN: 90-5892-050-X, http://hdl.handle.net/1765/1042

Wijk, R.A.J.L. van, *Organizing Knowledge in Internal Networks: A Multilevel Study*, Promotor: Prof.dr.ir. F.A.J. van den Bosch, EPS-2003-021-STR, ISBN: 90-5892-039-9, http://hdl.handle.net/1765/347

Wolters, M.J.J., *The Business of Modularity and the Modularity of Business*, Promotors: Prof. mr. dr. P.H.M. Vervest & Prof.dr.ir. H.W.G.M. van Heck, EPS-2002-011-LIS, ISBN: 90-5892-020-8, http://hdl.handle.net/1765/920

Wubben, M.J.J., *Social Functions of Emotions in Social Dilemmas*, Promotors: Prof.dr. D. De Cremer & Prof.dr. E. van Dijk, EPS-2009-187-ORG, http://hdl.handle.net/1765/1

Xu, Y., *Empirical Essays on the Stock Returns, Risk Management, and Liquidity Creation of Banks*, Promotor: Prof.dr. M.J.C.M. Verbeek, EPS-2010-188-F&A, http://hdl.handle.net/1765/1

Yang, J., *Towards the Restructuring and Co-ordination Mechanisms for the Architecture of Chinese Transport Logistics*, Promotor: Prof.dr. H.E. Harlambides, EPS-2009-157-LIS, ISBN: 978-90-5892-198-7, http://hdl.handle.net/1765/14527

Yu, M., *Enhancing Warehouse Perfromance by Efficient Order Picking*, Promotor: Prof.dr. M.B.M. de Koster, EPS-2008-139-LIS, ISBN: 978-90-5892-167-3, http://hdl.handle.net/1765/13691

Zhang, X., *Strategizing of Foreign Firms in China: An Institution-based Perspective*, Promotor: Prof.dr. B. Krug, EPS-2007-114-ORG, ISBN: 90-5892-147-5, http://hdl.handle.net/1765/10721

Zhu, Z., *Essays on China's Tax System*, Promotors: Prof.dr. B. Krug & Prof.dr. G.W.J. Hendrikse, EPS-2007-112-ORG, ISBN: 90-5892-144-4, http://hdl.handle.net/1765/10502

Zwart, G.J. de, *Empirical Studies on Financial Markets: Private Equity, Corporate Bonds and Emerging Markets*, Promotors: Prof.dr. M.J.C.M. Verbeek & Prof.dr. D.J.C. van Dijk, EPS-2008-131-F&A, ISBN: 978-90-5892-163-5, http://hdl.handle.net/1765/12703

Erasmus Research Institute of Management - ERIM

SCHEDULING WITH TIME LAGS

Scheduling is essential when activities need to be allocated to scarce resources over time. Motivated by the problem of scheduling barges along container terminals in the Port of Rotterdam, this thesis designs and analyzes algorithms for various on-line and off-line scheduling problems with time lags. A time lag specifies a minimum time delay required between the executions of two consecutive operations of the same job. Time lags may be the result of transportation delays (like the time required for barges to sail from one terminal to the next), the duration of activities that require no resources (like drying or cooling down), or intermediate processes on non-bottleneck machines between two bottleneck machines.

For the on-line flow shop, job shop and open shop problems of minimizing the makespan, we analyze the competitive ratio of a class of greedy algorithms. For the off-line parallel flow shop scheduling problem with time lags of minimizing the makespan, we design algorithms with fixed worst-case performance guarantees. For two special subsets of scheduling problems with time lags, we show that Polynomial-Time Approximation Schemes (PTAS) can be constructed under certain mild conditions. For the fixed interval scheduling problem, we show that the flow shop problem is solvable in polynomial time in the case of equal time lags but that it is NP-hard in the strong sense for general time lags. The fixed interval two-machine job shop and open shop problems are shown to be solvable in polynomial time if the time lags are smaller than the processing time of any operation.

Design & layout: B&T Ontwerp en advies (www.b-en-t.nl) Print: Haveka (www.haveka.nl)

ERIM

The Erasmus Research Institute of Management (ERIM) is the Research School (Onderzoekschool) in the field of management of the Erasmus University Rotterdam. The founding participants of ERIM are Rotterdam School of Management (RSM), and the Erasmus School of Economics (ESE). ERIM was founded in 1999 and is officially accredited by the Royal Netherlands Academy of Arts and Sciences (KNAW). The research undertaken by ERIM is focused on the management of the firm in its environment, its intra- and interfirm relations, and its business processes in their interdependent connections.

The objective of ERIM is to carry out first rate research in management, and to offer an advanced doctoral programme in Research in Management. Within ERIM, over three hundred senior researchers and PhD candidates are active in the different research programmes. From a variety of academic backgrounds and expertises, the ERIM community is united in striving for excellence and working at the forefront of creating new business knowledge.

ERIM PhD Series

Research in Management

Erasmus Research Institute of Management - **ERIM**
Rotterdam School of Management (RSM)
Erasmus School of Economics (ESE)
P.O. Box 1738, 3000 DR Rotterdam
The Netherlands

Tel. +31 10 408 11 82
Fax +31 10 408 96 40
E-mail info@erim.eur.nl
Internet www.erim.eur.nl