5-19-2005

Multi-Agent Workload Control and Flexible Job Shop Scheduling

Zuobao Wu
University of South Florida

Multi-Agent Workload Control and Flexible Job Shop Scheduling

by

Zuobao Wu

A dissertation submitted in partial fulfillment
of the requirements for the degree
Doctor of Philosophy
Department of Industrial and Management Systems Engineering
College of Engineering
University of South Florida

Major Professor: Michael X. Weng, Ph.D.
Tapas K. Das, Ph.D.
Grisselle Centeno, Ph.D.
Sudeep Sarkar, Ph.D.
Lihua Li, Ph.D.

Date of Approval:
May 19, 2005

Keywords: Due Date, Multi-agent Method, Make-to-Order, Production Planning and Control, Manufacturing Systems

Dedication

To the memory of my father, Zhangxun Wu for his lifelong pursuit of excellence with honesty.

Acknowledgments

I would like to express my appreciation and gratitude to my major professor, Dr. Michael X. Weng, for his guidance throughout this research. I would also like to thank the other committee members: Dr. Lihua Li, Dr. Tapas K. Das, Dr. Sudeep Sarkar and Dr. Grisselle Centeno for their valuable comments and suggestions. I express my special thanks to Dr. Lihua Li for his support and help.

I would especially like to thank my wife, Hong Qiu, and my son, Tao Wu, for their love. I would also like to thank my mother, Lanxiang Li, my brother, Zuohu Wu, and my sisters: Fuxian Wu, Meixian Wu and Chunxian Wu. Without their understanding and support, none of this would be possible.

Table of Contents

List of Tables iii

List of Figures v

List of Symbols vi

List of Acronyms ix

Abstract xii

Chapter 1 Introduction 1
1.1 Make-to-Order Industry 1
1.2 Workload Control in Make-to-Order Companies 2
1.3 Contributions 4
1.4 Dissertation Overview 5

Chapter 2 Literature Review 7
2.1 Introduction 7
2.2 Production Planning and Control 7
2.3 Due Date Setting 10
2.4 Job Release Control 15
2.5 Earliness and Tardiness Problems 18
2.6 Heuristic Scheduling 21
2.7 Flexible Job Shop Scheduling 24
2.8 Workload Control 26
2.9 Multi-Agent Systems 29

Chapter 3 Multi-Agent Scheduling Method 33
3.1 Introduction 33
3.2 Flexible Job Shop Scheduling 33
3.3 System Framework 35
3.4 Job Agent 36
3.5 Machine Agent 37
3.5.1 TOLJ Insertion Algorithm 38
3.5.2 SOLJ Sequencing Algorithm 40
3.5.3 Numerical Example 44
3.5.4 Machine Agent Protocol 46
3.6 System Coordination 47
3.7 Experimental Design 47
3.8 Analysis of Computational Results 49

3.8.1 WET under Different Utilizations 49
3.8.2 WT under Different Utilizations 51
3.8.3 WET under Different Numbers of Operations 52
3.8.4 WET under Different Processing Time Distributions 54
3.8.5 WET under Different Mean Processing Times 56
3.8.6 Simulation Time under Different Utilizations 58
3.9 Summary 59

Chapter 4 Dynamic Due Date Setting 61
4.1 Introduction 61
4.2 Order Entry in Make-to-Order Companies 61
4.3 DTWK and DPPW Rules 62
4.4 Order Entry Agent 64
4.5 Job Routing and Sequencing Agent 66
4.6 Information Feedback Agent 66
4.7 System Coordination 67
4.8 Analysis of Simulation Study 68
4.8.1 Comparisons among TWK, DTWK and DPPW 68
4.8.2 Comparisons among Four Rules 71
4.8.3 WET and WT Performance of Four Rules 73
4.8.4 Performance of Different Earliness and Tardiness Weights 75
4.9 Summary 78

Chapter 5 Multi-Agent Workload Control Methodology 79
5.1 Introduction 79
5.2 Job Release Control 79
5.3 Job Release Agent 80
5.4 System Architecture 81
5.5 System Coordination 82
5.5.1 Temporal Interdependency 82
5.5.2 Sub-goal Interdependency 82
5.6 Discrete Event Simulation 83
5.7 Simulation Study 84
5.7.1 System Performance Using DFPPW 85
5.7.2 System Performance Using DFTWK 86
5.7.3 System Performance under Different Utilizations 88
5.7.4 Performance under Different Processing Time Distributions 90
5.8 Summary 95

Chapter 6 Conclusions 97
6.1 Summary of Work 97
6.2 Future Research Directions 98

References 100

About the Author End Page

List of Tables

Table 3.1 Numerical Values of Example 46

Table 3.2 Simulation Parameters 48

Table 3.3 WET Performance under Different Shop Utilizations 50

Table 3.4 WT Performance under Different Shop Utilizations 52

Table 3.5 WET Performance under Different Numbers of Operations 54

Table 3.6 WET Performance under Different Processing Time Distributions 56

Table 3.7 WET Performance under Different Means of Processing Times 57

Table 3.8 Simulation Time under Different Shop Utilizations 58

Table 4.1 WET Performance under DTWK and DPPW 68

Table 4.2 WET Performance under Different Processing Time Distributions 71

Table 4.3 WET Performance under Different Scheduling Methods 72

Table 4.4 WET Performance under Different Shop Utilizations 72

Table 4.5 WET Performance under Different Processing Time Distributions 73

Table 4.6 WET Performance under Different Mean Processing Times 73

Table 4.7 WETs and WTs under Different Processing Time Distributions 74

Table 4.8 WETs and WTs under Different Mean Processing Times 75

Table 4.9 WETs and WTs under Same Weights 76

Table 4.10 Performance under Different Mean Processing Times 77

Table 4.11 WETs and WTs under Different Weights 77

Table 4.12 Performance under Different Mean Processing Times 78

Table 5.1 Performance under 90% Shop Utilization Using DFPPW 86

Table 5.2 Performance under 90% Shop Utilization Using DFTWK 87

Table 5.3 Performance under 85% Shop Utilization Using DFPPW 88

Table 5.4 Performance under 85% Shop Utilization Using DFTWK 89

Table 5.5 Performance under Normal Distribution Using DFPPW 91

Table 5.6 Performance under Normal Distribution Using DFTWK 92

Table 5.7 Performance under Uniform Distribution Using DFPPW 93

Table 5.8 Performance under Uniform Distribution Using DFTWK 94

List of Figures

Figure 3.1. Insert New Job by TOLJ Insertion Algorithm 39

Figure 3.2. Example of SOLJ Sequencing Algorithm 45

Figure 3.3. Average WET under Different Scheduling Methods 51

Figure 3.4. Average WT under Different Scheduling Methods 53

Figure 3.5. Average WET under Different Numbers of Operations 55

Figure 3.6. Average WET under Different Processing Time Distributions 57

Figure 3.7. Simulation Times under Different Scheduling Methods 59

Figure 4.1. Average WET Using DTWK 69

Figure 4.2. Average WET Using DPPW 70

Figure 4.3. Average WET under 95% Utilization 70

Figure 5.1. System Architecture of Multi-Agent Workload Control 81

Figure 5.2. Performance under 90% Utilization Using DFPPW 86

Figure 5.3. Performance under 90% Utilization Using DFTWK 87

Figure 5.4. Performance under 85% Utilization Using DFPPW 89

Figure 5.5. Performance under 85% Utilization Using DFTWK 90

Figure 5.6. Performance under Normal Distribution Using DFPPW 91

Figure 5.7. Performance under Normal Distribution Using DFTWK 92

Figure 5.8. Performance under Uniform Distribution Using DFPPW 94

Figure 5.9. Performance under Uniform Distribution Using DFTWK 95

List of Symbols

Δ	WET increase
ΔL_j	average lateness of recently completed jobs when job j arrives
a_j	earliness weight of job j
β_j	tardiness weight of job j
δ	number of waiting SOLJs in a machine queue
ρ	machine/shop utilization
λ	job arrival rate
φ	average number of jobs at each machine
a_k	available time of machine k
c_j	due date tightness factor of job j
C_j	completion time of job j
d_j	due date of job j
e	threshold value
E_j	earliness of job j
f	average flowtime at each machine
f_s	average shop flowtime
i	index of operations
I_j	idle time between job j and the job that follows job j
j	index of jobs
k	index of machines

k_1 planning factor

k_2 planning factor

K number of most recently completed jobs

l number of waiting jobs in a machine queue

l_j lead time of job j

L_j lateness of job j

M number of machines

M_{ij} set of machines that can process operation i of job j

N number of jobs

n average number of operations

n_j number of operations in job j

N_t number of uncompleted jobs at time t

o_{ij} operation i of job j

p average processing time for each operation

p_j total processing time of job j

p_{ijk} processing time for operation i of job j on machine k

Q_j number of jobs in queues at machines on job j's routing

R average interarrival time of jobs

r average remaining processing time per job in the shop

r_j release/ready time of job j

s_{ij} starting time of operation i of job j

$\hat{s}_{n_j,j}$ preferred starting time of operation n_j of job j

t current time

T_j tardiness of job j

w_j waiting time per operation of job j

W average shop workload

WIP average WIP level

List of Acronyms

ATC	apparent tardiness cost
BOM	bill of materials
CAGG	continuous aggregate loading
CONWIP	constant work-in-process
COVERT	cost over time
CR	critical ratio
DFPPW	dynamic feedback processing plus waiting
DFTWK	dynamic feedback total work content
DPPW	dynamic processing plus waiting
DTWK	dynamic total work content
EDD	earliest due date
ERP	enterprise resource planning
ET	earliness and tardiness
FCFS	first come first serve
IDD	internal due date
IFA	information feedback agent
JA	job agent
JIQ	jobs in queue
JIS	jobs in system

JIT	just-in-time
JRA	job release agent
MA	machine agent
MRP	material requirements planning
MRPII	manufacturing resource planning
MTO	make-to-order
MTS	make-to-stock
NOP	number of operations
ODD	operation due date
OEA	order entry agent
PPC	production planning and control
PPW	processing plus waiting
PR	production reservation
RBC	repeat business customizers
RSA	routing and sequencing agent
SOLJ	single operation left job
SPT	shortest processing time
S/OPN	slack per remaining operation
S/RPT	slack per remaining processing time
SSPR	single step production reservation
TOC	theory of constrains
TOLJ	two or more operations left job
TWK	total work content
VMC	versatile manufacturing companies

WET	weighted earliness and tardiness,
WIP	work-in-process
WIQ	work in queue
WLC	workload control
XDD	external due date

Multi-Agent Workload Control and Flexible Job Shop Scheduling

Zuobao Wu

ABSTRACT

In the make-to-order (MTO) industry, offering competitive due dates and on-time delivery for customer orders is important to the survival of MTO companies. Workload control is a production planning and control approach designed to meet the need of the MTO companies. In this dissertation, a multi-agent workload control methodology that simultaneously deals with due date setting, job release and scheduling is proposed to discourage job early or tardy completions. The earliness and tardiness objectives are consistent with the just-in-time production philosophy which has attracted significant attention in both industry and academic community. This methodology consists of the order entry agent, job release agent, job routing and sequencing agent, and information feedback agent.

Two new due date setting rules are developed to establish job due dates based on two existing rules. A feedback mechanism to dynamically adjust due date setting is introduced. Both new rules are nonparametric and easy to be implemented in practice. A job release mechanism is applied to reduce job flowtimes (up to 20.3%) and work-in-process inventory (up to 33.1%), without worsening earliness and tardiness, and lead time performances. Flexible job shop scheduling problems are an important extension of the classical job shop scheduling problems and present additional complexity. A multi-agent scheduling method with job earliness and tardiness objectives in a flexible job shop environment is proposed. A new job

routing and sequencing mechanism is developed. In this mechanism, different criteria for two kinds of jobs are proposed to route these jobs. Two sequencing algorithms based on existing methods are developed to deal with these two kinds of jobs.

The proposed methodology is implemented in a flexible job shop environment. The computational results indicate that the proposed methodology is extremely fast. In particular, it takes less than 1.5 minutes of simulation time on a 1.6GHz PC to find a complete schedule with over 2000 jobs on 10 machines. Such computational efficiency makes the proposed method applicable in real time. Therefore, the proposed workload control methodology is very effective for the production planning and control in MTO companies.

Chapter 1

Introduction

1.1 Make-to-Order Industry

There are two kinds of manufacturing sectors of industry: make-to-stock (MTS) sector and make-to-order (MTO) sector. Production planning and control (PPC) is crucial to help meet increasing customer demands and expectations as markets become more competitive. The desirable objective of PPC is just-in-time (JIT) production, which products should be produced by the right quality at the right time. Most research for PPC has been concentrated on the MTS industry. There has been a relative less attention for the MTO industry, even though this is a sizable sector of manufacturing industry. The basic distinction between MTS and MTO is the timing of the receipt of customer orders. In the MTS industry, the product is already available in stock when an order arrives and can be dispatched immediately to the customer from inventory. Enterprise resource planning or manufacturing resource planning (ERP/MRPII) systems are often applied for PPC in the MTS industry. In ERP/MRPII, the master production schedule provides the demand according to orders. The material requirements planning (MRP) nets demand, determines material requirements, and provides release dates. Capacity requirements planning checks plan feasibility. Thus, orders are translated into shop jobs with associated due dates and planned release dates.

In the MTO industry, some or all production takes place after the order is received. Thus MTO companies have ability to customize their products to meet the specific needs of

individual customers. A customer typically makes an enquiry to several possible MTO companies. The MTO company is thus in a competitive environment in determining how to respond to a customer, especially how to determine due dates. Customers usually desire early due date promises and manufacturers prefer extended due dates to ensure on-time delivery. The diverse and unpredictable nature of order arrivals in MTO companies makes the reliable due date setting and due date guarantee as a crucial task to improve on-time delivery performance. A trade-off has to be made between the customer and manufacturer. This demonstrates that there is the greatest need for sophisticated PPC methods to determine due dates and ensure on-time delivery in MTO companies.

1.2 Workload Control in Make-to-Order Companies

Workload control (WLC) is a sophisticated PPC approach specifically designed for the needs of MTO companies (Hendry and Kingsman 1989, Bertrand and Muntslag 1993). WLC consists of the three PPC levels of order entry, job release and scheduling. At order entry level, customer enquiries are processed, and due dates are determined. To control work-in-process (WIP) inventory, a job release mechanism determines when each job should enter the shop floor. After a job is released, the progress of the job on the shop floor is controlled by scheduling.

Most investigations have treated order entry, job release and scheduling separately. A few investigations focus on the interactions among order entry, job release and scheduling. It is challenging to coordinate order entry, job release and scheduling in real time. Simultaneously solving order entry, job release and scheduling problems by mathematical models may be quite time consuming (Kingsman 2000). No effective method addresses this issue.

Shop floor configuration is a major factor for the applicability of PPC approaches. Job shop is an appropriate configuration for many MTO companies (Muda and Hendry 2003). Due to the existence of considerable amount of overlapping capacities with modern machines and the stochastic nature of the arrival of orders in MTO companies, flexible job shops are common in MTO companies. A flexible job shop is a generalization of the job shop and the parallel machine environments (Pinedo 2002). In particular, there are a set of work centers in a flexible job shop environment. Each work center has a set of parallel machines with possibly different processing efficiency (Kacem, Hammadi and Borne 2002). Flexible job shops allow an operation to be performed by any machine in a work center and thus present two issues. The first is job routing: to assign each operation to a machine. The second is job sequencing: to order the operations assigned to a machine. Thus, flexible job shop scheduling consists of job routing and sequencing. Only a few methods such as tabu search, localization approach and neighborhood functions exist for flexible job shop scheduling. These methods require substantial computation load and are not suitable for large-scale scheduling problems in real time.

Manufacturing environments in MTO companies are real-time, dynamic systems. Multi-agent method has been taken as a promising approach for developing advanced manufacturing systems (Cutkosky, Tenenbaum and Glicksman 1996). Such an approach provides rapid responsive and dynamic reconfigurable structures to facilitate flexible and efficient use of manufacturing resources in a rapidly changing environment. This research focuses on the integration of order entry, job release, and flexible job shop scheduling by a multi-agent method. Minimizing job earliness and tardiness (ET) is the PPC objective.

The following definitions are used throughout this research. If the shop workload exceeds some preset maximum limit, new jobs are not released to the shop floor and wait. Such

unreleased jobs form a pre-shop pool. As jobs in MTO companies might differ significantly from each other in terms of their routings, number of operations and processing times, the workload is defined as the total remaining processing time of all jobs released to the shop floor. The workload norm is defined as the preset maximum limit of the workload. The waiting time in the pool is defined as the pool time of a job. The time between the release and completion of the job is defined as its shop flowtime. Thus, the time between the arrival and the completion of a job is the sum of its pool time and shop flowtime, and is commonly referred to as the manufacturing lead time.

1.3 Contributions

The contributions of this research are summarized as follows.

A new multi-agent WLC methodology that simultaneously deals with due date setting, job release and scheduling is proposed to discourage job early or tardy completions. This methodology consists of the order entry agent, job release agent, job routing and sequencing agent, and information feedback agent.

Two new due date setting rules are developed to establish job due dates based on two existing rules. A feedback mechanism to dynamically adjust due date setting is introduced. Both new rules are nonparametric and easy to be implemented in practice.

A job release mechanism is applied to reduce job flowtimes and shop WIP inventory. At the critical norm, the job release mechanism significantly reduces job flowtimes (up to 20.3%) and WIP inventory (up to 33.1%), without worsening ET and lead time performances.

A new multi-agent scheduling method with job earliness and tardiness objectives in a flexible job shop environment is proposed.

A new job routing and sequencing mechanism is developed in the multi-agent scheduling method. In this mechanism, different criteria for two kinds of jobs are proposed to route these jobs. Two sequencing algorithms based on existing methods are developed to deal with these two kinds of jobs.

The proposed WLC methodology is implemented in a flexible job shop environment. The computational results show that the proposed two new due date setting rules outperform the existing DTWK and DPPW rules for ET objectives. The proposed multi-agent scheduling method also outperforms the existing scheduling methods. Therefore, the proposed WLC methodology is very effective for the PPC in MTO companies.

1.4 Dissertation Overview

The rest of this dissertation is organized into five chapters. Chapter 2 reviews the relevant research in eight fields. First is a review of current PPC methods and their applicability. Second, due date setting approaches and rules are reviewed. Third, reviews of job release mechanisms are presented. Fourth, the studies on ET problems are surveyed. Fifth, heuristic scheduling methods are discussed. Sixth, the existing flexible job shop scheduling methods are briefly described. Seventh, most existing research on the interactions among order entry, job release and scheduling are given. At last, the applications of multi-agent systems in PPC are reviewed.

In Chapter 3, a multi-agent scheduling method with job ET objectives in a flexible job shop environment is proposed. A new job routing and sequencing mechanism for flexible job shops is presented. Two heuristic algorithms for job sequencing are developed. The simulation results are given.

Two new due date setting rules are described in Chapter 4. A feedback mechanism to dynamically adjust due date setting is introduced. The effectiveness of the two due date setting rules is also presented. In Chapter 5, the theory of job release control is described. A job release mechanism is discussed. The multi-agent WLC methodology for MTO companies is proposed. The computational results are also discussed.

Chapter 6 concludes this research and suggests future research directions.

Chapter 2

Literature Review

2.1 Introduction

As mentioned in Chapter 1, the goal of this research is to integrate due date setting, job release, job routing and sequencing. This chapter gives the related literature review.

2.2 Production Planning and Control

The MTO industry can be classified into two types (Amaro *et al.* 1999): repeat business customizers (RBC) and versatile manufacturing companies (VMC). A RBC provides customized products on a continuous basis over the length of a contract. Products are customized but may be made more than once permitting a small degree of predictability. The VMC market is more complex requiring more sophisticated solutions. In VMC, a high variety of products with variable demand are manufactured in small batches with little repetition. Both RBC and VMC allow customization, but RBC is able to establish more stability by enticing customers into a more predictable and committed relationship.

MTO companies produce a high variety of products in lower volume than MTS companies. Unstable market demand means a MTO philosophy would be too costly. Production does not take place until customer orders receive, allowing a greater degree of customization. Customization invariably leads to nonstandard product routings on the shop floor, and lead times are naturally longer than those for MTS companies. The price and due date that a company can quote affect its success in winning orders, resulting in lead times

taking on strategic importance. It is in the MTO industry that there is the greatest need for sophisticated PPC methods.

PPC methods are crucial to help meet increasingly high customer demands and expectations as markets become more competitive. Typical functions of a PPC system include customer enquiry and order processing, material requirements planning, input and output control, and scheduling. Thus, it can be classified by three levels: order entry, job release and scheduling. PPC methods vary at the three levels. Past research had a tendency to skip the order entry and job release levels, as these stages are of little significance in a typical MTS environment. However, the three PPC levels are important to the MTO industry.

There are many PPC methods such as ERP/MRPII, WLC, Kanban, and theory of constrains (TOC), and constant WIP (CONWIP). Their applicability is different for different production environments. A simple, effective solution for one company may be insufficient to solve the planning problems of another. To be successful in companies, a PPC approach should fit to the production environment. Essential elements of the approach should correspond with the characteristics of the production system. For classical methods such as MRP, these elements have become common sense. BOM (bill of materials)-explosion and constant lead times make MRP known to perform best in environments with high material and low capacity complexity. However, a PPC method in MTO companies must cope with many products, variable routing and numerous set ups. For example, once a RBC has established a contract with a customer, it needs less control over the order entry stage, but a VMC must go through the whole process for every order.

WLC is based on principles of input/output control. Input control relates to both accepting orders and releasing them to the shop floor. Once released, the jobs remain on the shop floor. Scheduling will direct orders along their downstream operations. Each operation

relates to a specific capacity group consisting of one or more machines and operators. Both order entry and job release can be accompanied by output control decisions in terms of capacity adjustments. WLC uses a pre-shop pool of jobs to reduce shop floor congestion, making the shop floor more manageable. It stabilizes the performance of the shop floor and makes it independent of variations of incoming orders (Bertrand and Van Ooijen 2002). For most WLC approaches, jobs are only released onto the shop floor if the workload does not exceed its norm, while ensuring jobs do not stay in the pool too long in order to reduce lead times and meet due date objectives. While jobs remain in the pool, unexpected changes to quantity and design specifications can be accommodated at less inconvenience.

A framework is proposed to explore the applicability of WLC in MTO companies (Henrich, Land and Gaalman 2004). The framework supports an initial consideration of WLC in the first phase of a PPC selection and implementation process. It is concluded that the applicability of WLC increases with raising variability, indicated by increased arrival rate fluctuations, due date differences, processing time variability, routing sequence and routing length variability. While routing flexibility has not been widely reported in literature, it can contribute to the applicability of WLC.

As discussed above, MTO companies have to react on dynamic environments: they have to cope with changes in product mix and volume, production rate changes, a high number of rush jobs, and lot of internal uncertainty. Thus, the PPC in MTO companies is rather complex and often based on insecure data. Therefore, WLC is a sophisticated PPC approach specifically designed for the needs of MTO companies (Zapfel and Missbauer 1993, Hendry, Kingsman and Cheung 1998).

2.3 Due Date Setting

A due date can be assigned to an order by first estimating its flowtime and then adding a delivery safety allowance to account for transportation and uncertainties. Due date assignment is one of the main application areas of flowtime estimation. As it is frequently observed in literature, most research efforts directed towards flowtime estimation are within the context of due date assignment.

There are basically two flowtime estimation approaches in literature: analytical approach and simulation approach. Cheng and Gupta (1989) presented an extensive survey of these approaches for the due date assignment problem. There are advantages and disadvantages associated with each approach. The analytical approach offers an exact way of determining means and variances of flowtime estimates. However, the dynamic and stochastic nature of production systems makes it difficult to develop realistic analytical models. On the other hand, simulation approach does not always produce reliable estimates. Moreover, a great number of computer runs may also be needed in the latter case to obtain accurate and precise estimates. Since these two areas are complimentary in nature, the literature has been developed in both directions.

Due date setting methods can be dynamic or static. Dynamic methods employ job characteristics and shop congestion information for determining due dates. Static methods consider only job content information such as arrive time, routing, and processing time. For static methods, a job flow allowance is a fixed amount for given job data and does not depend upon shop status when the job arrives (Baker 1984).

The first simulation-based study in this area was conducted by Conway (1965) who compared four flowtime estimation methods: total work content (TWK), number of operations (NOP), constant, random. The results of this study indicate that the methods which utilize the

job information perform better than the others. Conway also observed the relationship between due date assignment methods and dispatching rules. Later, Eilon and Chowdhury (1976) used shop congestion information in estimating flowtimes. In their work, TWK is compared with three other methods: jobs in queue (JIQ), delay in queue and modified TWK. Results indicate that JIQ, which employs the shop congestion information, outperforms other methods.

Many studies have consistently concluded that assigning due dates based on job content and shop congestion information could lead to better shop performance than assigning due dates based only on job content. Weeks (1979) proposed a method which combines both job and shop information. This method performs very well for the performance metrics such as mean lateness, mean earliness, and number of tardy jobs. The results also indicate that flowtime estimation is affected by the structural complexity of the shop more than the size of the system. Bertrand (1983a) proposed a new method of flowtime estimation which exploits time-phased workload information of the shop. Two factors are used in analyzing the performance of the method: minimum allowance for waiting and capacity loading limit. His results indicate that time-phased workload and capacity information significantly decrease variance of the lateness. Ragatz and Mabert (1984) compared eight different methods: TWK, NOP, TWK-NOP, JIQ, work in queue (WIQ), WEEK's method, jobs in system (JIS), and response mapping rule. Among them, the response mapping rule utilizes the response surface methodology to identify the significant factors in flowtime estimation. The results indicate that the job and workload information are very important for predicting flowtimes.

Kanet and Christy (1989) compared TWK with the processing plus waiting (PPW) rule via computer simulation in a job shop with forbidden early shipment. PPW estimates the flow allowance of a job by adding an estimate of the waiting time to the total processing time of a job. The waiting time is proportional with the number of operations. The results indicate that

TWK is superior to PPW in terms of the mean tardiness, proportion of tardy jobs, and mean inventory level. Fry *et al.* (1989) also investigated the job and shop characteristics which affect job flowtimes in a job shop. They constructed two linear and two multiplicative nonlinear models to estimate the coefficients of the factors. This study shows that models using product structure and shop conditions can estimate more accurate flowtimes than the others, linear models are superior to the multiplicative models, and the predictive ability of the models also improves as the utilization increases.

Vig and Dooley (1991) proposed two flowtime estimation methods: operation flowtime sampling, and congestion and operation flowtime sampling. These methods are also compared with JIQ and TWK-NOP under various shop conditions. The results indicate that congestion and operation flowtime sampling and JIQ yield the best performance. Vig and Dooley (1993) extended their work by combining static and dynamic estimates to obtain job flowtime estimates. Gee and Smith (1993) proposed an iterative procedure for estimating flowtimes when due date dependent dispatching rules are used. Two flowtime estimation methods are employed, the one is based on job related information and the other one utilizes both job and shop related information. Their results indicate that the late method yields better estimation. They also compared the iterative approach with the response mapping rule of Ragatz and Mabert (1984) and found that the quality of flowtime estimation was improved by the iterative approach.

As described above, TWK and PPW are parametric rules and need appropriate parameter selection based on the analysis of historical data which requires preliminary runs. TWK is a static and job characteristic related due date setting method. If two jobs have the same amount of work, the same allowance will be given to them, regardless of what the current shop load is, i.e. whether heavy or moderate. This kind of due date assignment lacks the means

of estimating job flowtimes dynamically. It seems that ET performance, which stress the importance of meeting job due dates as closely as possible, can be improved if due date allowance is set to the dynamically estimated flowtime of each job.

In another study, Bertrand (1983b) provided an analytical model used to establish an internal due date (IDD) for shop floor control and an external due date (XDD) quoted to the customer. It is concluded that the use of workload information can contribute substantially to setting attainable due dates in job shops, and the due date setting rule produces a constant mean lateness. Delivery reliability to the customers can be controlled by making the XDD equal to the IDD plus the mean lateness plus a safety time related to the variance of lateness. Thus a small variance of lateness reduces the quoted XDD. His study also indicates that the best variance performance is obtained with an assignment rule that uses a time-phased representation of the workload in the shop.

Later, Enns (1994, 1995) proposed a dynamic estimation method which employs a dynamic version of PPW (DPPW). By using the feedback of exponentially smoothed flowtime estimation error, the lateness variance is estimated. He also describes a method of setting due dates to achieve of the desired percentage of tardy jobs. Enns (1998) developed a workload balancing dispatch mechanism and a dynamic version of TWK (DTWK). In his dynamic forecasting model, two different mechanisms based on exponentially smoothing errors are used to set safety allowances that will result in the targeted percent of tardy deliveries. If a due date independent dispatching rule is used, the operation lateness variance mechanism is appropriate. Otherwise, the job lateness variance mechanism is appropriate. The results indicate that a shop load balance index which considers both shop load and variability has a very strong relation with lead times. Cheng and Jiang (1998) proposed a similar dynamic forecasting model for DTWK and DPPW.

DTWK and DPPW are capable of adjusting the flowtime estimation by using feedback information about current shop load conditions. Simulation results show that the dynamic due date rules are significantly better than their static counterparts. In addition, DTWK and DPPW are nonparametric and, therefore, are simple to implement without preliminary runs for parameter estimation. However, these models do not consider the pool time of a job in WLC situations.

Recently, several artificial intelligent methods were proposed for due date setting. Philipoom *et al.* (1994) investigated the feasibility of using artificial neural networks in flowtime estimation. The neural network models are used to forecast due dates in a simple flow shop manufacturing system. They estimated the coefficients of the methods with neural networks instead of multiple regressions. The results indicate that the neural network approach offers certain advantages over the conventional approaches. However, job due dates in a flow shop are stable, and the system deviation is smaller than that in a job shop. Huang *et al.* (1999) constructed an artificial neural network model to predict production performance for a wafer fabrication factory. They used a three-layer back-propagation neural network that allows for more accurate prediction of the WIP level and for moving volume in the next period for each wafer fabrication operation stage. There are the following advantages using neural network models: neural networks can obtain a probable result even if the input data are incomplete or noisy; a well-trained neural network model can provide a real-time forecasting result; creating a neural network model does not necessitate understanding the complex relationship among the input variables. Artificial neural network models were also used for estimating lead times in a virtual wafer fabrication system (Hsu and Sha 2004). They suggest that if system information is not difficult to obtain, the artificial neural network models can perform a better due date prediction than conventional rules.

A method to dynamically control the safety allowance through reinforcement learning was provided, in which job flowtimes are estimated by parametric due date setting rules (Moses 1999). The applicability of the method to an unrestricted class of discrete manufacturing systems is preserved by the use of a feedback control paradigm, and control knowledge is acquired using reinforcement learning. The current shop status is considered so that due date performance is improved during transient conditions. Results of simulation experiments demonstrate the effectiveness of the method.

These artificial intelligent methods are more computationally expensive and artificial neural network methods would require a set of training data (Sabuncuoglu and Comlekci 2002).

2.4 Job Release Control

Job release has a significant effect on system performance. Specifically, they reduce WIP inventory and variability on the shop floor. Bergamaschi *et al.* (1997) provided a literature review available on efforts to optimize job release. Sabuncuoglu and Karapinar (1999) classified job release methods into four types. The first is job release mechanisms that do not use any information about shop status or job characteristics. Examples are immediate release and interval release. The second is load-limited job release mechanisms that release jobs to the shop floor according to the current workload in the shop. The third is time-phased job release mechanisms that release jobs at predetermined release times based on flowtime estimates. They utilize information about shop capacity and job due date. The fourth is release mechanisms that consider both the current workload and job due dates. They are the extensions of load-limited release with additional considerations on due dates.

There are three common load-limited job release mechanisms (Land and Gaalman 1996). The first is Bechte release mechanism builds on three parameters: a release period, a

time limit and a workload norm. The decision to release jobs is taken periodically, at the beginning of each release period. All jobs in the pool are sequenced in order of their planned release date. The planned release date is determined by backward scheduling from the job due date. All jobs within the time limit are candidates for release. In the established sequence, jobs are released until the workload norm is exceeded. All other candidates have to wait in the job pool until the next period of release. The selection process goes on for the remaining candidates. The workload considered in this mechanism is the queue length at a machine. The second is Bertrand release mechanism does not discuss the release sequence, but elaborates the workload norms extensively. The release decision is taken periodically and job release is allowed if the workload of each machine is less than its norm. The workload considered in this mechanism differs from the workload considered by Bechte. The workload definition of Bertrand covers the processing time of all jobs on the shop floor which still have to be processed at the considered machine. The corresponding workload norm consist of two components: the planned machine output during the release period and the planned quantity of work upstream or in the queue at the end of the release period. Thus, the norm depends on the average machine position within the job shop. The third is Tatsiopoulos release mechanism formalizes three ways of job release. The common push release takes place periodically. Intermediate push release can be forced by rush jobs or jobs with retarded material availability, and an intermediate pull release can be triggered from the shop floor when a foreman sees his machine threatened by unplanned idleness. The periodic release decision considers jobs in the sequence of their planned latest release dates. Job release is allowed unless a workload norm is exceeded, which applies to the intermediate pull releases as well.

When jobs are released periodically, load-limited release mechanisms have to set the planning period length and the check period length. They greatly influence the shop

performance, thus confirming the necessity of a careful setting of such parameters (Perona and Portioli 1998).

The impact of load-limited job release in job shops was investigated by Kanet (1988), who concluded that controlling the release of new orders should be carefully considered as it could result in increased idle times and thus negatively impact performances such as tardiness while making no real impact on inventory. Raman (1995) defined the notion of critical and non-critical jobs. He then used a bicriteria objective to minimize total tardiness and maximize the sum of release times. The former objective is applied to critical jobs; the latter is applied to non-critical jobs. Several authors used cost functions to evaluate methodology performance. Tardif and Spearman (1997) used the 'capacity feasible' time bucket approach of MRP to determine release times. Land and Gaalman (1998) developed an alternative mechanism building on the approach of Fredenhall and Melnyk (1995), which yields significant performance improvement. In this mechanism, a job is released when the queue at its first workstation is empty and it has the earliest planned release time of the unreleased jobs that start at this workstation, or if no urgent jobs are in the queue and this job has the shortest processing time of all unreleased urgent jobs.

Time-phased release mechanisms can be classified into infinite loading and finite loading. The release time of infinite loading is calculated by subtracting the expected lead time from the due date of a job:

$$r_j = d_j - l_j, \tag{2.1}$$

where r_j is the release time of job j, d_j job due date and l_j its estimated lead time. In particular, backward infinite loading utilizes the following methods to calculate the release time:

$$r_j = d_j - k_1 n_j, \tag{2.2}$$

$$r_j = d_j - k_1 n_j - k_2 Q_j, \tag{2.3}$$

where k_1 and k_2 are the planning factors, n_j number of operations in job j and Q_j the number of jobs in queues at machines on job j's routing.

Finite loading considers available shop capacity over the planning horizon and tries to match machine requirements of the jobs with the available capacity. Two types of finite loading can be identified: forward finite loading and backward finite loading. The first approach loads all operations of the job into available capacity starting from the first operation. The release decision of a particular job is based on the loading period of the last operation and the due date of a job. The job is released if the loading period of the last operation is within a preset time window about the due date. Backward finite loading operates in the opposite direction. That is, each operation is placed into available capacity starting with the last operation of the job and working backward from the job due date. As compared to forward finite loading, the release decision is based on the loading period of first operation and the current time. The job is released if this period is within a preset time window from the current time.

Time-phased release mechanisms focus on determining a release time for each job, regardless of current shop load. They often continuously release jobs to the shop floor. However, load-limited release mechanisms are based on current shop load. They are easier to balance and limit the shop load and therefore control WIP. On the other hand, job release also presents a research paradox. It has been found that the pool time is extensive; therefore, the lead time to produce a job is not reduced (Melnyk *et al.* 1994).

2.5 Earliness and Tardiness Problems

Even though makespan is a well-known performance measure widely used in classical scheduling problems, it does not reflect the main objective concerned with some problems in practice. For example, flowtime represents a speed of response in manufacturing environment

and is a good indicator of production rate. Tardiness is a due date based measure in terms of delivery performance.

The JIT philosophy has been a popular management concept. Motivations for implementing JIT production are to reduce inventories and improve response times (Zhu and Meredith 1995). One research area for JIT implementation that has been widely studied in recent years is to schedule jobs so as to minimize job ET. In general, such problems are broadly called ET problems. The objective of ET problems is consistent with the JIT philosophy where an early or a late delivery of a job results in increase of production costs. However, the majority of articles that address ET problems deal with scheduling problems for single machine and parallel machines (Baker and Scudder 1990, Leung 2002, Ventura and Radhakrishnan 2003, Croce and Trubian 2002). Heady and Zhu (1998) provided a heuristic algorithm for minimizing ET in a multi-machine scheduling problem. The heuristic solution procedure is based on a single machine sequencing heuristic. The single machine heuristic starts by forming a good initial job sequence, and then uses a proven method to optimally time the jobs. Luh, Chen and Thakur (1999) proposed an effective approach, which takes into account such factors as uncertain arrival times, processing times, due dates, and job priorities. A problem formulation was presented with the goal to minimize job ET. Combining Lagrangian relaxation and stochastic dynamic programming, a solution methodology was also developed to obtain dual solutions. Zhu and Heady (2000) developed a mixed integer programming formulation for minimizing job ET in a multi-machine scheduling problem. Ip *et al.* (2000) applied a genetic algorithm in order to obtain an optimal solution for ET performance in a large-scale production planning and scheduling problem. Yoon and Ventura (2002) presented linear programming formulations for minimizing the mean weighted absolute deviation from due dates to find optimal schedules in a lot streaming flow shop. A polynomial time solution to minimize the

maximum ET with unit processing times in a flow shop environment was addressed (Mosheiov 2003). An integer optimization formulation for a job shop scheduling system was developed to maximize on-time delivery, low inventory and small number of setups (Chen *et al.* 2003).

Whether idle time between jobs should be allowed in a schedule is an important issue for ET problems. The issue of inserting idle times depends upon the types of due dates and the workload. Clearly, it is not wise to force an unnecessarily early completion of a job when the workload is not heavy unless the machine has a large startup cost. In general, it is reasonable to insert idle times between jobs when jobs have distinct due dates and to delay the starting time when all jobs have a common due date (Alidee 1994). However, if idle time insertion is not treated properly, an ET solution procedure may fail to minimize job ET. Kutanoglu and Sabuncuoglu (1999) identified the conditions under which it may be better to keep the resource idle for a soon-to-arrive urgent job. Hodgson *et al.* (1998) proposed a simulation-based procedure for minimizing the maximum lateness. It is effective and efficient in providing optimal or near optimal schedules for job shop scheduling. This procedure was also modified to provide better schedules by inserting idle time under certain conditions (Hodgson *et al.* 2000).

A heuristic method for the ET problem on single machine with unequal due dates and ready times was presented by Mazzini and Armentano (2001). A feasible solution is obtained through a constructive stage and then a local search procedure is applied to update its idle times. The main feature of this approach is that idle times are suitably inserted during the constructive stage. When compared with EDD, the computational results have shown that the heuristic presents a good performance for the test problem instances with up to 80 jobs.

2.6 Heuristic Scheduling

Job shop scheduling is an important aspect of production management that has a significant effect on the performance of a job shop. The combinatorial complexity of the scheduling problem has received considerable attention in literature. Various techniques, such as mixed integer programming modeling (Liao and You 1993) and branch-and-bound algorithms (Balas, Lenstra and Vazacopoulos 1995) have been used to overcome the problem of this complexity. Recently, significant improvements have been made with the development of efficient scheduling algorithms using tabu search, simulated annealing, neural networks and genetic algorithms. In general, one of the major drawbacks of above algorithms is that the scheduling problems studied by the majority of these researchers were simplified to provide the conditions upon which these methods could be based (Blazewicz, Dmschke and Pesch 1996). However, analytical results obtained are usually for special cases, and most real-life job shop scheduling problems do not fall into this class of special cases. Furthermore, the computational complexity of a scheduling problem increases exponentially as the size of the problem increases. Thus, heuristics are appropriate methods in large-scale scheduling problems since they create good schedules and are considerably faster than other methods (Shafaei and Brunn 1999).

Dispatching rules are the most common approach in industry (Subramaniam *et al.* 2000). It determines the ranking of the jobs waiting at machine queues. The information needed by dispatching rules is classified (Kutanoglu and Sabuncuoglu 1999) by: arrival times, e.g. first come first serve (FCFS); process times, e.g. shortest processing time (SPT); due dates: allowance based, e.g. earliest due date (EDD); slack based, e.g. SLACK; ratio based, e.g. critical ratio (CR); combination of one or more of the above, e.g. operation due date (ODD).

In many applications, meeting due date and avoiding delay penalty is the most important scheduling goal. Flow allowance of a job is the time between the release date and the due date. The simplest version of allowance based priority is the EDD rule. The simplest slack-based priority rule is the SLACK rule, which gives priority to the job with the smallest slack. The ratio-based rules utilize a kind of ratio in their implementations. For instance, CR rule gives a priority to the job with the smallest flow allowance/remaining processing time. Other ratio-based rules are slack per remaining processing time (S/RPT) and slack per remaining operation (S/OPN). S/RPT gives a priority to the job with the longer remaining processing time, while S/OPN considers the job with more operations remaining as urgent.

Some rules utilize operation due dates. The work content method is generally suggested for mean tardiness among several ways of assigning operation due dates (Baker 1984). According to this method, the initial flow allowance of a job is allocated to the operations proportional to their processing times. The rules such as EDD, SLACK and CR have their operation due date versions. Operation-based rules perform better than their job-based counterparts (Kanet and Hayya 1982).

Most studies have tested simple rules designed for some extreme shop conditions and known to be deficient with certain load levels. For example, EDD, SLACK, and S/RPT rules perform reasonably with light load levels but deteriorate in congested shops; whereas SPT rule performs well in congested shops with tight due dates, but fails with light load levels and loose due dates. Thus, there have been attempts to combine two or more of these simple dispatching rules into a single rule in order to use their individual excellent performance characteristics. Cost over time (COVERT) was specifically developed for tardiness objective (Carroll 1965).

The majority of above studies have been done in a uniform (or balanced) environment. For unbalanced systems, bottleneck dynamics was studied with the development of apparent

tardiness cost (ATC) by Vepsalainen and Morton (1987). ATC is very similar to COVERT with two main differences. First, the slack is local resource constrained slack which takes into account the waiting times on downstream machines. Second, the decay function for the ratio of weight/processing time is exponential rather than linear.

The queuing time of a job in a job shop normally accounts for the major portion of its flowtime. Hence, a job flowtime cannot be accurately determined without some knowledge regarding the expected total queuing time for its remaining operations. Queuing time could be influenced by many factors and are very difficult to estimate correctly (Chang 1997). Some of possible factors for are: scheduling heuristic; total processing time remaining; number of operations remaining; number of jobs currently in the system; number of jobs currently in the machine queues on this job's routing; and total processing time of all jobs currently in the machine queues on this job's routing.

There are several methods to estimate queuing times. Standard estimation method calculates the queuing time of a job as proportional to its processing time. One issue in this method is to select a right multiplier value. In actual systems this can be done by using regression analysis with historically collected queuing times. Lead time iteration method is an iterative procedure which aims to improve queuing time estimation. It estimates queuing times by successive approximations using deterministic simulation. First, a job shop is simulated using SLACK as the sequencing rule. This is a transient simulation starting from the current state of a job shop and running until the completion of all jobs. Queuing times are recorded for each job at each machine visited. A revised slack is then calculated using the queuing times observed from the simulation. The simulation is then rerun using the revised slack from the previous iteration. The process is repeated until the estimations of the queuing times stabilize.

On small problems, the procedure may converge exactly. Usually, queuing estimates tend to stabilize after 3 to 10 iterations (Zozom *et al.* 2003).

2.7 Flexible Job Shop Scheduling

Numerically controlled multi-purpose machines in job shops have a considerable amount of overlapping capabilities. They can be easily reconfigured to perform a variety of operations. In order to maximize job shop performances, management should make use of the flexibility while operating job shops. Otherwise, the advantage of having very capable machines might disappear. On the other hand, consideration of flexibility in job shop scheduling will dramatically increase the complexity of the problem, which is already very hard to solve. This will certainly increase the cost of the solution.

The classical modeling of a job shop scheduling problem does not reflect the requirements of modern job shops. Modeling such scheduling problems without considering overlapping capabilities does not reflect the reality of modern job shops. Classical job shop scheduling methods are generally incapable of addressing such capacity overlapping. There is a need to model and solve flexible job shop scheduling problems. Flexible job shops allow an operation to be performed by any machine in a work center. The corresponding flexible job shop scheduling problems are an important extension of the classical job shop scheduling problems. Although there is a huge amount of literature on classical job shop scheduling problems, flexible job shop scheduling problems do not have much literature.

A tabu search algorithm for flexible job shop scheduling problems was developed (Chambers 1996). In this algorithm, the feasible initial solution with the smallest makespan is obtained by selecting from the 12 priority dispatching solutions. Then, two move neighborhoods are implemented, corresponding to job routing and sequencing in flexible job

shop scheduling. A sequencing move is defined by the exchange of adjacent critical operation pairs. Each machine is scanned successively for candidate exchange pairs. A routing move is also defined by the relocation of a critical operation to a feasible alternate machine position. For a given solution, every each relocation of every reroutable critical operation is considered. The contemplated move that yields a smaller makespan can override a move's tabu status.

Local search techniques and two neighborhood functions for flexible job shop scheduling problems were proposed (Mastrolilli and Gambardella 2000). Local search employs the idea that a given solution may be improved by making small changes. A local search algorithm starts off with an initial solution and then continually tries to find better solutions by searching neighborhoods. In order to minimize the makespan, two neighborhood functions are used in local search methods for the flexible job shop scheduling problems. The computational experiments found 120 new better upper bounds and 116 optimal solutions over 221 benchmark problems.

A linguistic based meta-heuristic modeling and solution approach for solving flexible job shop scheduling problems was presented (Baykasoglu 2002). Makespan is considered as the main performance criteria to evaluate the goodness of the generated solutions. Mean flowtime, number of tardy jobs, and maximum tardiness are also considered. This approach makes use of linguistics, simulated annealing and priority rule-based heuristic. The main contribution is to show how the grammars of linguistics can be utilized in modeling and solving flexible job shop scheduling problems. In his work, the flexible job shop scheduling problem is presented as a grammar and the productions in the grammar are defined as controls. Employing the grammars simplify the model formation. This simplification has enabled the development of meta-heuristic optimization procedures such as simulated annealing for the solution of the

problem. Thus, use these controls and the priority rule-based heuristic, a simulated annealing algorithm is developed to solve flexible job shop scheduling problems.

A localization approach and an evolutionary approach were presented for jointly solving job routing and sequencing problems with total or partial flexibility (Kacem, Hammadi and Borne 2002). The considered objective is to minimize makespan and the total processing time of the machines. The localization approach makes it possible to solve the problem of resource allocation and build an ideal assignment model. When each operation is assigned to the suitable machine, this localization approach takes into account the workloads of machines on which the operations have already been assigned. In the evolutionary approach controlled by the assignment model, advanced genetic manipulations are applied in order to enhance the solution quality. The initial population is constructed starting from the set of assignments found in the localization approach. This study also explains some of the practical and theoretical considerations in the construction of a more robust encoding to solve the flexible job shop problem by applying genetic algorithms. It is worth noting that the scheduling uses different dispatching rules.

In general, one of the major drawbacks of above methods is that they require substantial computation load and are not suitable for solving practical large-scale scheduling problems.

2.8 Workload Control

WLC encapsulates the three planning and control levels of order entry, job release and scheduling. Routing and sequencing are usually studied separately. It can result in many problems due to factors such as conflicting objectives and an inability to communicate in dynamic situations. To overcome these problems, researchers (Nasr and Elsayed 1990, Huang, Zhang and Smith 1995) stressed the need to integrate job routing and scheduling. By taking

into account shop status information, it is possible to increase the effectiveness of routing decisions at scheduling level.

Weintraub *et al.* (1999) presented a procedure for scheduling jobs with alternative processes in job shops. The objective of this procedure is to minimize manufacturing costs while satisfying job due dates. Process plans with alternatives job routes, operations, and sequences are selected according to current shop conditions. The results show that there are substantial differences in scheduling performance between scheduling with alternatives and scheduling without alternatives. Scheduling with alternatives can greatly improve the ability to satisfy due dates under various shop conditions. An integration model of concurrent planning and scheduling was realized through a multi-agent approach (Wu, Fuh and Nee 2002). It provides a practical approach for software integration in a distributed environment.

Many investigations have been done in the interactions among due date setting, job release and scheduling. Melnyk and Ragatz (1989) used simulation to investigate the impact of due date tightness, release mechanism, and shop dispatching rules on a number of performance measures. Their results show that job release mechanisms have a significant impact on performance and the impact is dependent upon the specific mechanism.

Wein and Chevalier (1992) defined a broader scheduling problem that considers three dynamic decisions: assigning due-dates to exogenously arriving jobs, releasing jobs from a job pool to the shop floor, and sequencing jobs at each of two machines in the shop. The job shop is modeled as a multiclass queuing network, and the objective is to minimize both shop WIP and job lead times, subject to an upper bound constraint on the proportion of tardy jobs.

Tagawa (1996) proposed a new concept of job shop scheduling system, which consists of the following five decision systems. Order entry system has the function of screening arrived orders and setting the due dates of accepted orders. Master scheduling system makes a

broad schedule of design, fabrication, and assembly. Once a customer order is accepted, it is changed into a planned job and is turned over to the master scheduling system. Master schedule consists of detailed design schedule, fabrication schedule, and assembly schedule. Firstly, detailed design schedule is made by forward scheduling method. Secondly, assembly schedule is made by backward scheduling method starting from job due dates. Lastly, fabrication schedule is made so as to be inserted between assembly schedule and detailed design schedule. Job scheduling system makes a schedule on job basis and work center basis. In the job scheduling system, the master schedule is broken down into a schedule with possible start day and finish day. Operation scheduling system has the function of making a feasible schedule, which is on operation, machine and work day basis. The main criterion of this system is to keep the due date given by the job scheduling system. The subcriterion is the utilization of the machine. Dispatching system determines the sequence of operations to be done on the specified work day on each machine. In this dispatching system, dispatch is done periodically, such as once half a day, every two days, or others. The criteria of the dispatching system are keeping due dates, elevating the utilization of a machine and easiness of the operation.

A framework to integrate job release, routing, and sequencing was proposed (Shafaei and Brunn 2000). This system consists of an integer programming model that is concerned with job release and routing decisions and a dispatching rule that provides the detailed scheduling. Two heuristics that integrate job release and scheduling were proposed, which are effective at lowering WIP and satisfying due dates (Zozom 2003). However, a practical problem in MTO companies is that the due date setting for possible orders from customer enquiries, job release and scheduling should be coordinated in real time. A more effective method should be explored to meet this need.

2.9 Multi-Agent Systems

The technological advances of distributed information systems have greatly inspired and supported the development of multi-agent systems in production planning, scheduling and control. Shaw (1988) proposed a multi-agent manufacturing scheduling and control mechanism. He pointed out that a manufacturing cell could subcontract work to other cells through a bidding mechanism. A multi-agent virtual manufacturing system was implemented in a simulated form using the MetaMorph mediator-centric federation architecture on a distributed computing platform (Maturana and Norrie 1996). It interfaces with the multi-agent concurrent design environment system. Therefore, design, process planning, routing and scheduling activities are coordinated concurrently across a simulated extended enterprise. In MetaMorph, mediator is a distributing decision-making support system for coordinating the activities of a multi-agent system. This coordination involves three main phase: subtasking, creation of virtual communities of agents, and execution of the processes imposed by the tasks.

Sikora and Shaw (1997) presented a multi-agent framework for achieving system integration. Within this framework, they developed coordination mechanisms for the agents on three levels: the decision level, where several functional modules collaborate on the underlying decision processes; the process level, where agents interact to complete the processes based on their task expertise and mutual interdependence; and, finally, the system level, where different stages coordinate their functioning to achieve desirable system level performance. Sikora and Shaw (1998) also provided a representational formalism, coordination mechanisms, and control schemes necessary for integrated different units of an information system while meeting such performance criteria as overall effectiveness, efficiency, responsiveness, and robustness. Saad, Kawamura and Biswas (1997) made use of a contract-net approach for heterarchical scheduling of flexible manufacturing systems. Their system employed a production reservation (PR)

approach where a job agent schedules all the operations prior to its release to the shop. A problem with the PR approach is that it does not handle the need to reschedule jobs when machine breakdowns occur or there is a need to modify a job. To solve the problem, they also proposed a single step production reservation (SSPR) approach that schedules one operation at a time as a job moves through the system. In terms of average tardiness, they found that SSPR outperformed PR.

Cavalieri *et al.* (2000) compared two multi-agent models: a market-like multi-agent architecture and a multi-agent architecture with supervisor. The former model can be referred as representative of pure heterarchical architecture. The latter, due to the presence of a supervisor agent, is a reference of architectures with a slight degree of hierarchy. The experiments show that the market-like architecture results more robust than the architecture with supervisor.

In this market-like model, an agent with a high decision-making autonomy represents each manufacturing entity. In particular, two main typologies of agents, the part agent and the resource agent, are available. The part agent is the control module of a production batch or a single manufacturing job. It contains all manufacturing data regarding the job, the main information and the decision-making rules for carrying out negotiation processes and controlling on-line production. On the other hand, a resource agent is the logical representation of any of the production resources in a shop floor. Like a part agent, a resource agent collects all the information related to the negotiation tasks. The control strategy is carried out through a contract-net protocol. A part agent activates task announcement when triggered events arrive, these events include part arrival, near completion of an operation, resource breakdown after commitment, etc. The resource agents that receive the announcement construct bids based on their status and system states, and submit bids if they desire. Parts may receive many bids.

They will evaluate the bids and prepare an offer to the chosen resources. When a resource gets an offer, it has the opportunity to accept or deny based on certain circumstances. The negotiation is completed when both part and resource are committed.

In the multi-agent architecture with supervisor, the scheduling phase is distinguished from the real-time control phase. In first phase, a supervisor agent selects resources according to technological and operational criteria. In real-time control phase, it is accomplished by the part agent, since it is the user of the service supplied by all the machines assigned during the scheduling phase. It can solve locally and autonomously unexpected situations due to breakdowns or operation delays. However, if the problem cannot be solved locally, the intervention of the supervisor is requested. In the model, the supervisor can modify the assignments built up during the scheduling phase.

Ren (2000) presented a multi-agent scheduling architecture. In his architecture, every machine, job and control system has its own scheduling agent to determine local scheduling priorities for jobs. Each agent is assigned a weight, called a cooperation weight, and the final priority of a job is a weighted sum of these local priorities. The job with the highest final priority is processed first. To show the effectiveness, the architecture is applied to solve three job shop scheduling problems. One is to schedule so as to minimize the mean tardiness of all jobs. The second is to schedule so as to minimize the mean ET of all jobs. The third is a generalization of the second problem, which replaces single-point due dates with job due windows. The exhaustive search and simulated annealing are used to find the beast cooperation weights.

Lu and Yih (2001) proposed a framework that utilizes autonomous agent and weighted functions for distributed decision-making while all agents work in active and collaborative ways to help their decisions. This collaborative control framework is capable of realizing and

seeking balances among heterogeneous objectives of the production entities within a collaborative manufacturing system. Simple index values, instead of detailed data, were used for information exchange among agents. This can greatly reduce the communication and computation load of the control system and keep detailed production information confidential while the agents in the system could belong to different companies.

Usher (2003) explored two methods of enhancing the negotiation process employed by a multi-agent system to support performance improvements in real-time routing of jobs in a job shop environment. The first method takes advantage of an extended negotiation period to provide a more complete picture of the shop conditions in order to enhance the validity of the decisions made by individual agents. The second approach explores the possibility of process model data to increase the accuracy of time estimates used in the negotiation process.

Maione and Naso (2003) applied genetic algorithms to adapt the decisions strategies of autonomous agents in a heterarchical manufacturing system. Yen and Wu (2004) presented a multi-agent scheduling paradigm to transform existing standalone scheduling systems to Internet scheduling agents that can communicate with each other and solve problems beyond individual capabilities. Subbu and Sanderson (2004) proposed an evolutionary multi-agent planning framework particularly suited to distributed design and manufacturing systems. This framework combines a multi-agent architecture and distributed coevolutionary algorithms.

Chapter 3

Multi-Agent Scheduling Method

3.1 Introduction

Flexible job shop scheduling problems are an important extension of the classical job shop scheduling problems and present additional issues. A multi-agent scheduling method with job ET objectives in a flexible job shop environment is discussed in this chapter. The ET objectives are consistent with the just-in-time production philosophy which has attracted significant attention in both industry and academic community.

3.2 Flexible Job Shop Scheduling

In this research, scheduling consists of job routing and sequencing and is a decision-making process with the objectives of minimizing job ET. Job routing and sequencing is to organize the execution of N jobs on M machines. Each job j consists of n_j operations that need to be done in a given order on predetermined work centers. Let O_{ij} denote operation i in job j. The execution of O_{ij} requires one machine selected from a work center that consists of a set of machines $M_{ij} \subseteq M$. Job routing is to assign each operation O_{ij} to machine k ($k \in M_{ij}$). Job sequencing is to determine the starting time s_{ij} of O_{ij} on machine k.

Defining d_j as the due date and C_j as the completion time of job j, job earliness is given by

$$E_j = \max(0, d_j - C_j), \tag{3.1}$$

and job lateness and tardiness are defined by

$$L_j = C_j - d_j, \tag{3.2}$$

$$T_j = \max(0, L_j). \tag{3.3}$$

The scheduling objectives are to minimize the total weighted earliness and tardiness (WET), which is given by

$$WET = \sum_{j=1}^{n} (\alpha_j E_j + \beta_j T_j), \tag{3.4}$$

where α_j is the earliness weight and β_j is the tardiness weight. When $\alpha_j = \beta_j = 1$ for all j, this reduces to the special case of minimizing the total unweighted ET.

To minimize the WET, it may be necessary to insert idle times. This means holding a job that may be completed too early. One way to accomplish this is to examine the slack of a job. It is obvious that a job should be held from processing if it has a large positive slack, especially when it only has a single operation left. However, the decision can be quite difficult if a job still has many operations left. Under these two situations, jobs are distinguished (Ren 2000) and defined as the following: If a job only has a single operation left, it is called a SOLJ (single operation left job); otherwise it is called a TOLJ (two or more operations left job). Intuitively, the completion time of a SOLJ can be determined accurately once it starts processing. If the earliest possible completion time of any SOLJ is greater than its due date, it is tardy, even if the SOLJ is processed immediately when the machine becomes available. In this case, SOLJ j is preferred to start as early as the machine is available. On the other hand, SOLJ j may have a lot of positive slack. This allows the job to start at an appropriate starting time such that it can be completed exactly at its due date. We define such starting times as preferred starting times. For a job with n_j operations, when the job is a SOLJ, (n_j-1) operations have been finished and the remaining operation of the SOLJ is the last operation n_j, and the starting time

of SOLJ j can be expressed as $s_{n_j,j}$. Let $\hat{s}_{n_j,j}$ denote the preferred starting time of SOLJ j, p_{ijk} the processing time of O_{ij} on machine k, and a_k the available time of machine k. Then $\hat{s}_{n_j,j}$ can be computed by

$$\hat{s}_{n_j,j} = \begin{cases} a_k, & \text{if } a_k + p_{ijk} > d_j; \\ d_j - p_{ijk}, & \text{otherwise.} \end{cases} \tag{3.5}$$

If machine k is idle, a_k is the current time t; otherwise it is equal to the completion time C_{h,j_c} of operation h of current job j_c being processed on k. So a_k is given by

$$a_k = \begin{cases} t, & \text{if } k \text{ is idle}; \\ C_{h,j_c}, & \text{otherwise.} \end{cases} \tag{3.6}$$

Ideally, a SOLJ should be scheduled by making its starting time $s_{n_j,j}$ equal to $\hat{s}_{n_j,j}$. Practically, however, tardiness is usually worse than earliness. This is because tardiness leads to unsatisfied customers, while earliness just means some inventory holding. Therefore, it may be better to set preferred starting times earlier. In particular, we consider reducing the preferred starting times by a threshold value e as given in (3.7).

$$\hat{s}_{n_j,j} = \begin{cases} a_k, & \text{if } a_k + p_{ijk} > d_j; \\ d_j - p_{ijk} - e, & \text{otherwise.} \end{cases} \tag{3.7}$$

3.3 System Framework

The proposed multi-agent method consists of job agents (JAs) and machine agents (MAs). Each agent has its goals and includes three components: a knowledge base, a functional component, and a control unit (Sikora and Shaw 1997). The knowledge base consists of the domain knowledge/data. The functional component consists of computational procedures for decision-making. The control unit consists of protocols that provide the mechanism for agents

to communicate with each other. The protocols of all agents together constitute the system coordination approach.

3.4 Job Agent

A JA communicates with MAs and makes routing decision by selecting a machine for each operation. A JA is created whenever a job is released to the shop. When the job finishes its processing, the JA is destroyed.

A JA maintains a list of machines for each operation. It also has the following knowledge to formulate a bid in the MA: the number of uncompleted operations, the remaining processing time of an operation that is currently processing on the machine, and the uncompleted processing time of a job. The data contained in JA knowledge base consists of the job ID, due date, release time, earliness cost, tardiness cost and process planning of each job. A process planning contains an operation sequence, the work center and processing time for each operation.

The functional component of a JA is for job routing. Job routing selects a machine for the next operation of a job when the current operation is completed. When a job is a TOLJ, TOLJ routing selects the machine with the earliest completion time. If at least two alternatives are tied for the criterion of the earliest completion time, the machine with fewer queuing jobs is selected for shop load balancing.

However, when a new SOLJ selects a machine for its next operation, the machine with the smallest total WET of SOLJs from existing jobs in the machine queue and the new SOLJ is selected. If there is a tie, the machine with fewer waiting SOLJs is selected. The motivation is that fewer waiting SOLJs can reduce the overlapping chance between SOLJs, which means the

SOLJs can start most likely at their preferred starting times and, thus, results in a smaller total WET of SOLJs. If there is still a tie, the criteria of TOLJ routing are applied.

For each operation of any job, the JA requests bids from the machines that can process the operation. For a TOLJ, a bid includes the completion time of the operation and the queue size on a machine. For a SOLJ, a bid includes the total WET of all SOLJs, SOLJ size, completion time of the operation and queue size on a machine. The JA evaluates all the bids and selects a machine. The JA repeats this procedure until all the processing is completed. The JA protocol is given as follows.

JA protocol

1) Send a bid request to MAs.
2) Evaluate the bids from MAs.
3) Select a machine.
4) If all operations of a job are completed, stop; otherwise go to step 1.

3.5 Machine Agent

An MA is responsible for the decisions related to job sequencing. Each machine is represented by an MA. An MA has the knowledge of its status (idle or busy), queuing jobs, number of finished tasks and total machine busy time. The data contained in MA knowledge base consists of the machine ID, machine type, machine capabilities and cost of each machine.

The functional component of an MA is for job sequencing. To minimize the WET, job sequencing should try to make SOLJs finished on time. When a new job (say job v) is scheduled in a machine queue, there are two possibilities: a SOLJ or a TOLJ.

In the case 1, job v is a TOLJ. In this case, schedule job v, without interrupting the current schedule of the existing jobs, to be completed as early as possible.

In the case 2, job v is a SOLJ. First reschedule v and all existing SOLJs. Then insert existing TOLJs one at a time using the algorithm for case 1. The insertion order of existing TOLJs can be determined using some dispatching rules. We tested 5 rules: FIFO, SPT, EDD, SLACK and COVERT (Kutanoglu and Sabuncuoglu 1999), and found that COVERT generally gave significant better results. The COVERT priority index of a job represents the expected tardiness cost per unit of imminent processing time. If a job has zero or negative slack, it is projected to be tardy and its priority index is $1/p_{ijk}$. If the slack exceeds the worst case waiting time, the priority index is zero. If the slack is between these two extremes, the priority changes linearly as the slack changes.

Two heuristic algorithms are presented to schedule jobs: TOLJ insertion algorithm is used for case 1 and SOLJ sequencing algorithm is proposed for case 2.

3.5.1 TOLJ Insertion Algorithm

TOLJ insertion algorithm determines the starting and completion times of the new TOLJ v. It is modified based on the PR approach (Saad, Kawamura and Biswas 1997). In the insertion process of TOLJ v, the current schedule is kept unchanged. There are three possibilities: v is inserted at the head, inside or at the end of the queue to get the earliest completion time.

When TOLJ v is inserted at the head, let job j_1 as the first job in the queue to be processed. If the starting time s_{h,j_1} of operation h of j_1 is equal to or greater than the sum of the available time a_k of machine k and the processing time p_{ivk} of TOLJ v, v can finish its

processing before j_1 starts. Therefore, v should be scheduled at the head of the queue as depicted in Figure 3.1(a).

The completion time C_{iv} is given by

$$C_{iv} = a_k + p_{ivk}. \tag{3.8}$$

In Figure 3.1(a), as a_k is greater than current time t, machine k is busy at time t.

When TOLJ v is inserted inside, an MA searches forward through the current schedule to determine the earliest idle time period that the machine can accommodate the processing of TOLJ v. Let I_j as the idle time between job j and the job that follows j in the current schedule. If I_j is equal to or great than p_{ivk}, TOLJ v can be inserted into the idle time period as depicted in Figure 3.1(b). C_{iv} is given by

$$C_{iv} = C_{hj} + p_{ivk}. \tag{3.9}$$

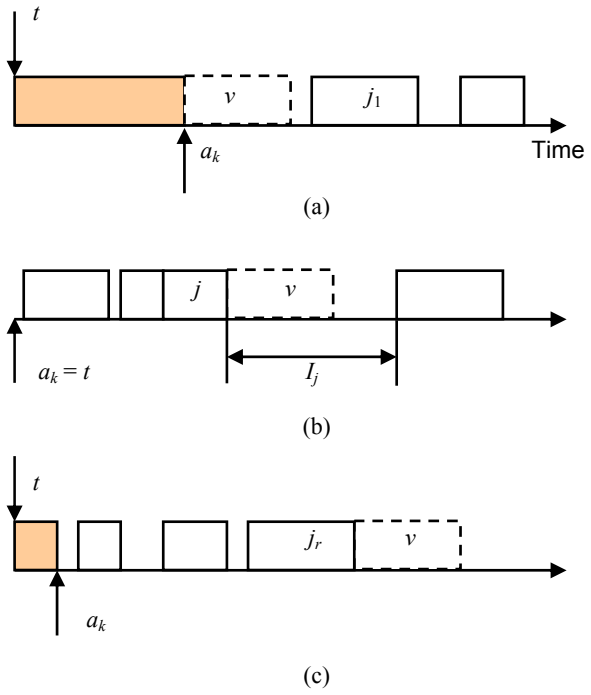

Figure 3.1. Insert New Job by TOLJ Insertion Algorithm

When TOLJ v is inserted at the end, let job j_r be the last job in the queue to be processed. TOLJ v is then scheduled following job j_r, as depicted in Figure 3.1(c). C_{iv} is given by

$$C_{iv} = C_{h,j_r} + p_{ivk}. \tag{3.10}$$

Let l be the number of existing jobs. The TOLJ insertion algorithm is presented as follows.

TOLJ insertion algorithm

1) Set u =1.

2) If $s_{h,j_1} \geq a_k + p_{ivk}$, $C_{iv} = a_k + p_{ivk}$, stop; otherwise go to step 3.

3) If $u = l$, $C_{iv} = C_{h,j_r} + p_{ivk}$, stop; otherwise go to step 4.

4) If $I_j \geq p_{ivk}$, go to step 5; otherwise go to step 6.

5) $C_{iv} = C_{hj} + p_{ivk}$, stop.

6) $u = u + 1$, go to step 3.

In this algorithm the complexity of steps 1 and 2 is O(1). Steps 3-6 constitute a loop that is repeated at most n times. There are O(1) operations in steps 3-6. The resulting complexity for the loop is O(n). So the complexity of TOLJ insertion algorithm is O(n).

3.5.2 SOLJ Sequencing Algorithm

SOLJ sequencing algorithm determines the starting and completion times by rescheduling all jobs including new SOLJ v. It first reschedules all SOLJs including v to minimize the total WET and then inserts the existing TOLJs.

Let l be the number of waiting jobs and δ be the number of waiting SOLJs. Then there are (l-δ) waiting TOLJs. SOLJ sequencing algorithm calculates the preferred starting time $\hat{s}_{n_j,j}$ of SOLJ v and updates $\hat{s}_{n_j,j}$ for the existing SOLJs by (3.5).

If $\delta \geq 1$, there are at least two SOLJs including v. SOLJ sequencing algorithm reschedules all SOLJs by the MA algorithm (Mazzini and Armentano 2001), which consists of the ordering procedure, feasibility procedure, updating procedure and local search procedure. As the local search procedure has little effect on the solutions, it is not implemented in this research. The ordering, feasibility and updating procedures in a flexible job shop environment are restated briefly as follows.

The ordering procedure sequences all SOLJs in non-decreasing values of $\hat{s}_{n_j,j}$. The feasibility procedure then schedules one of the SOLJs at a time by attempting to make their $s_{n_j,j}$ equal to $\hat{s}_{n_j,j}$. If there is no overlapping between j and any other job already in the partial schedule, the procedure schedules another job. Otherwise, it is necessary to eliminate the overlapping. Let j^* be the first job with which j overlaps. The following four possible moves are considered in order to eliminate the overlapping.

In the first move, j remains in its current position and j^* shifts to the right. The new completion time $C'_{n_{j^*},j^*}$ of j^* is the sum of the completion time $C_{n_j,j}$ of j and the processing time $p_{n_{j^*},j^*,k}$ of j^*. The WET increase for j^* is given by

$$\Delta_1=\max\{\alpha_{j^*} E'_{j^*}, \beta_{j^*} T'_{j^*}\}-\max\{\alpha_{j^*} E_{j^*}, \beta_{j^*} T_{j^*}\}, \tag{3.11}$$

where E'_{j^*} is new weighted earliness, T'_{j^*} is new weighted tardiness, E_{j^*} is old weighted earliness and T_{j^*} is old weighted tardiness.

In the second move, j^* remains in its current position and j shifts to the right. The new completion time $C'_{n_j,j}$ of job j is the sum of the completion time $C_{n_{j^*},j^*}$ of j^* and the processing time $p_{n_j,j,k}$ of j. The WET increase for j is computed as

$$\Delta_2=\max\{\alpha_j E'_j, \beta_j T'_j\}-\max\{\alpha_j E_j, \beta_j T_j\}. \quad (3.12)$$

In the third move, j^* shifts to the right and j shifts to the left. It is necessary to determine the appropriate amount of left and right shifts. In order to keep the WET increase to the minimum, one can compute the latest starting time $s'_{n_j,j}$ of job j by

$$s'_{n_j,j}=\max\{C_{n_{j^{**}},j^{**}}, a_k, s_{n_{j^*},j^*}-p_{n_j,j,k}\}, \quad (3.13)$$

where j^{**} is the job before job j^* and $(s_{n_{j^*},j^*}-p_{n_j,j,k})$ is the minimum shift of j to the left to eliminate the overlapping between j^* and j. Note that the job is ready to be processed when a job is routed to a machine in a flexible job shop environment. However, if the machine is busy, the job has to wait. So the ready time of a job in the MA algorithm is replaced by a_k in (3.13).

The completion times of j and j^* in the new partial schedule are

$$C'_{n_j,j}= s'_{n_j,j}+p_{n_j,j,k}. \quad (3.14)$$

$$C'_{n_{j^*},j^*}=C'_{n_j,j}+p_{n_{j^*},j^*,k}. \quad (3.15)$$

The WET increase for j and j^* is given by

$$\Delta_3=\max\{\alpha_{j^*} E'_{j^*}, \beta_{j^*} T'_{j^*}\}-\max\{\alpha_{j^*} E_{j^*}, \beta_{j^*} T_{j^*}\}$$

$$+\max\{\alpha_j E'_j, \beta_j T'_j\}-\max\{\alpha_j E_j, \beta_j T_j\}. \quad (3.16)$$

In the fourth move, j^* shifts to the left and j shifts to the right. This move is similar to Move 3 and the appropriate starting time $s'_{n_{j^*},j^*}$ for j^* is computed as

$$s'_{n_{j^*},j^*} = \max\{C_{n_{j^{**}},j^{**}},\ a_k,\ s_{n_j,j} - p_{n_{j^*},j^*,k}\}. \quad (3.17)$$

The new completion times of j and j^* are

$$C'_{n_{j^*},j^*} = s'_{n_{j^*},j^*} + p_{n_{j^*},j^*,k}. \quad (3.18)$$

$$C'_{n_j,j} = C'_{n_{j^*},j^*} + p_{n_j,j,k}. \quad (3.19)$$

The WET increase Δ_4 is computed by (3.16), using the completion times given by (3.18) and (3.19).

Of the above 4 moves, the move with the minimum WET increase (i.e., move argmin$\{\Delta_i \mid i=1,2,3,4\}$) is selected to eliminate the overlapping. The procedure repeats until all infeasibilities are eliminated.

The updating procedure aims to reduce the total WET and consists of two phases: shifting the jobs to the left and shifting the jobs to the right.

Once the SOLJs are scheduled, if there are TOLJs (i.e., $l > \delta$), TOLJ insertion algorithm inserts one of the TOLJs at a time in the order determined by the CoverT rule.

SOLJ sequencing algorithm is formally stated as follows.

SOLJ sequencing algorithm

1) Calculate $\hat{s}_{n_j,j}$ for each SOLJ.

2) If δ =0, go to step 8; otherwise go to step 3.

3) Order SOLJs in non-decreasing values of $\hat{s}_{n_j,j}$.

4) Set u=1.

5) Insert a SOLJ by feasibility procedure.

6) Update idle times by updating procedure.

7) If $u=\delta+1$, go to step 8; otherwise $u=u+1$, go to step 5.

8) Calculate the total WET of SOLJs.

9) If $l>\delta$, go to step 10; otherwise stop.

10) Calculate the CoverT priority index for each TOLJ.

11) Sequence existing TOLJs by the CoverT rule.

12) Set $u=1$.

13) Insert a TOLJ by TOLJ insertion algorithm.

14) If $u=l-\delta$, stop; otherwise go to step 15.

15) $u=u+1$, go to step 13.

In this sequencing algorithm, the values of $\hat{s}_{n_j,j}$ in step 1 and CoverT priority index in step 10 are computed with complexity O(n). The ordering in step 3 and the sequencing in step 11 are implemented to run in O(n log(n)) time. The complexity of the loop in steps 5-7 is O(n^3). The complexity of the computation in step 8 is O(1). Steps 13-15 form a loop that is repeated at most (n-1) times and at each repetition there are O(n) operations in step 13. The complexity of this loop is O(n^2). Thus, the complexity of SOLJ sequencing algorithm is O(n^3).

3.5.3 Numerical Example

This example is provided to demonstrate SOLJ sequencing algorithm. The current schedule on machine k is shown in Figure 3.2(a). There are 2 SOLJs (S1 and S2) and 3 TOLJs (T1, T2 and T3). A new SOLJ v is to be scheduled. The current time t is 52 and the available time a_k is 60. The values of p_{ijk}, d_j, α_j and β_j corresponding the existing jobs and v are shown in Table 3.1.

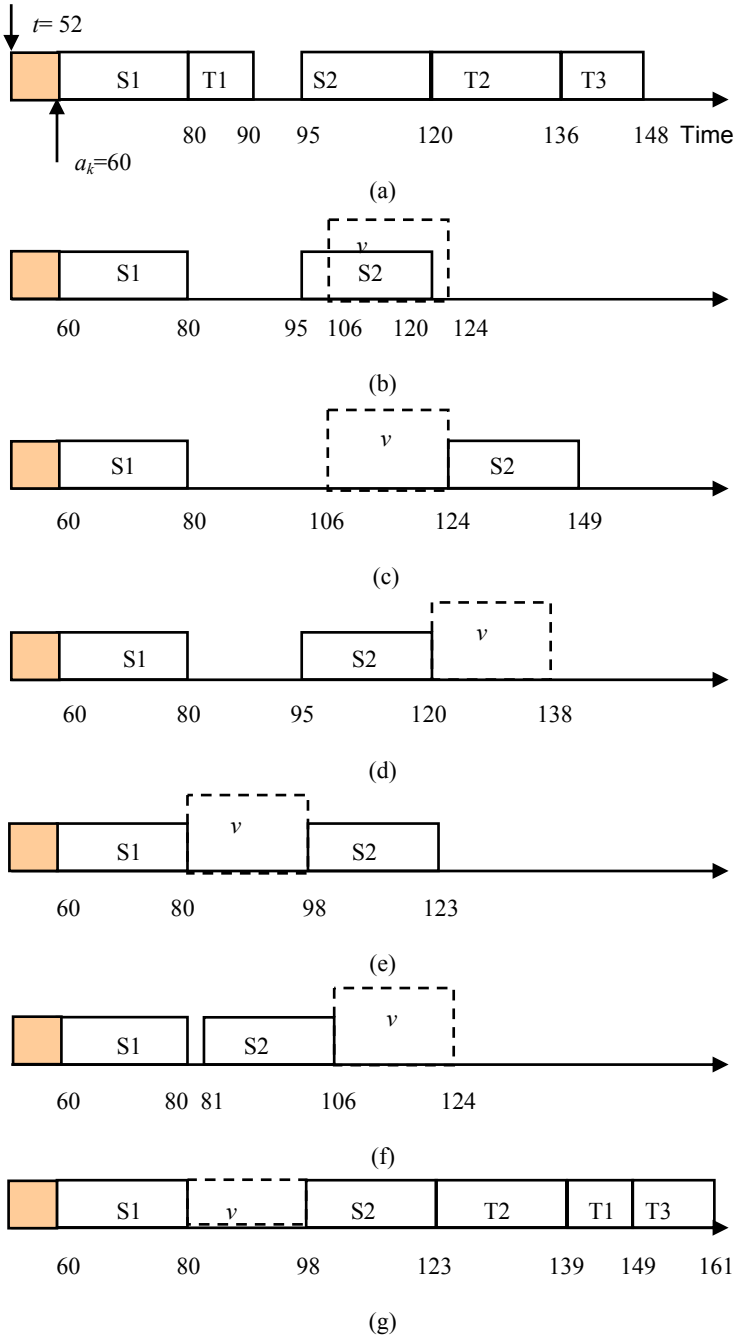

Figure 3.2. Example of SOLJ Sequencing Algorithm

By (3.5), $\hat{s}_{n_v,v}$ =106 and the preferred starting times of S1 and S2 do not changed. Let the SOLJs start at their preferred starting times. The partial schedule of SOLJs depicted in

Figure 3.2(b) is infeasible. In particularly, SOLJ *v* overlaps with S2. The four moves 1-4 are depicted in Figure 3.2(c-f) respectively with Δ_1= 116, Δ_2=42, Δ_3=38 and Δ_4=42. Therefore, move 3 is selected because it gives the minimum WET increase.

Table 3.1 Numerical Values of Example

Job	p_{ijk}	d_j	α_j	β_j
S1	20	68	2	5
S2	25	120	3	4
T1	10	176	5	2
T2	16	157	2	1
T3	12	198	5	1
v	18	124	1	3

Since there is no idle time in the partial schedule by move 3, the updating procedure does not change it. The total WET of SOLJs is 98 by (3.4). The CoverT rule gives the order of (T2, T1, and T3) for the TOLJs. TOLJ insertion algorithm inserts one of the three TOLJs at a time in the order and the final schedule is depicted in Figure 3.2(g).

3.5.4 Machine Agent Protocol

An MA receives the bid request from a JA. The MA formulates and submits a bid. Once a machine is selected, the job is added to the machine queue. When a machine is idle and there are waiting jobs in the machine queue, the machine processes the jobs by the schedule. The MA protocol is described as follows.

MA Protocol

1) Receive a bid request from a JA.

2) Formulate a bid.

3) Submit the bid to the JA.

4) Add a job to the machine queue.

5) Process jobs by the schedule.

3.6 System Coordination

There is a temporal interdependency among the activities of JAs and MAs. There is also a sub-goal interdependency. A JA needs to know the completion time of the operation of a TOLJ or the total WET of SOLJs, which can be determined by the MA. Thus, each agent is dependent on the others, resulting in a circular interdependency.

The system coordination begins when a job is released to the shop. When an MA receives the bid request from a JA, it formulates a bid. Then the JA evaluates all bids from the MAs and selects a machine. Once a machine is selected, the job is moved to the machine. Finally, the machine processes jobs according to the schedule. The coordination activities for a job continue with one operation at a time until the job is finished. It is assumed that the coordination time can be ignored in comparison with the processing time. So a JA initiates coordination for next operation of the job when the current operation is completed.

3.7 Experimental Design

To test the performance of the proposed multi-agent method, we consider the following flexible job shop. It has five work centers. Each work center has two parallel machines with different speeds. Specifically, the processing time of a job on one machine is 10% longer than that on the other machine. Different operations of a job are performed in different work centers. The operation sequence for a job is randomly generated among five work centers. However, there are 5 alternative sequences. No job preemption is allowed. Job reentrance is not allowed. The total work content rule is used to set job due dates. That is,

$$d_j = r_j + c\, p_j, \tag{3.20}$$

where r_j is the release time of job j, c is the due date tightness factor and p_j is the total processing time of job j. The shop load is determined by a job arrival rate. The job arrivals are generated using an exponential distribution for interarrival times. The average interarrival time R can be expressed as

$$R = pn/(M\rho), \tag{3.21}$$

where p is the average processing time for each operation, n the average number of operations and ρ the shop utilization. The simulation parameters are shown in Table 3.2.

Table 3.2 Simulation Parameters

Parameter	Values
ρ	80%, 85%, 90%, 95%
c	2, 4, 6, 8, 10, 12
p_{ijk}	uniform(1, 30)
α_j	uniform(1, 5)
β_j	uniform(1, 5)
n_j	3, 4, 5, uniform(3, 5)

Each simulation experiment consists of twenty replications. As mentioned earlier, job tardiness is generally worse than job earliness. Therefore, it may be better to start jobs before their preferred starting times by some threshold. In our experiments, we considered two threshold values (e=0 and e=2 p). In each replication, the shop is continuously loaded with jobs numbered on their arrivals. The simulation continues until 2200 jobs are completed per run. We are interested in system behavior in steady state. To eliminate the system warm-up effect, the first 200 completed jobs are not recorded. Please note that when each simulation terminates, there are still jobs in the system.

The entire system is implemented using an object-oriented programming approach and C++, which is run on a 1.6GHz PC with 512MB RAM. The PR and SSPR approaches are also implemented as a benchmark for comparing the relative performances of the proposed multi-agent approach. These two approaches are better than a number of common dispatching rules and well-known existing multi-agent methods to route jobs dynamically.

3.8 Analysis of Computational Results

This section reports the results of computational experiments. We analyze these results in detail and provide our findings. In addition to the WET performance, we also report the weighted tardiness (WT) performance to show the robustness of our proposed method, since WT has been used as a primary performance measure against job due dates in literature.

3.8.1 WET under Different Utilizations

When each job has 5 operations, Table 3.3 presents the means and standard deviations (s.d.) of the WETs. When the utilization level is 90%, Figure 3.3 gives the average WETs under different scheduling methods.

From Table 3.3 and Figure 3.3, the proposed multi-agent method significantly outperforms PR and SSPR for all utilization levels and due date settings.

Under the proposed scheduling method, as the due date tightness factor increases, the average WET decreases. This trend also maintains as the shop utilization level decreases. This should be expected as in low utilization shops and with loose due date settings, one can get better schedules to complete jobs closer to their due dates. As a matter of fact, for large due date setting factors and low shop utilization levels (80% and 85%), the proposed multi-agent method found schedules in which jobs are completed very close to their due dates. At high shop

utilization levels (90% and 95%), the proposed multi-agent method performs very well by using a threshold value $e=2p$. However, at 80% and 85% shop utilization levels, the method performs better when $e=0$ unless for very tight due date settings.

Table 3.3 WET Performance under Different Shop Utilizations

	Agent (e=0)		Agent1 (e=2 p)		SSPR		PR	
c	mean	s. d.	mean	s. d.	mean	s. d.	mean	s. d.
				ρ = 95%				
2	1958.76	564.15	1920.74	550.82	2237.29	604.87	3207.28	867.67
4	1590.75	519.39	1462.09	516.18	1813.09	588.84	2764.53	863.77
6	1257.24	398.73	1030.06	435.33	1444.20	547.13	2353.98	837.15
8	1010.04	366.26	653.56	426.56	1164.32	466.39	2000.70	780.61
10	792.07	320.28	474.30	365.24	1013.47	319.53	1722.19	693.61
12	641.82	257.22	291.88	293.32	992.14	190.20	1533.58	568.50
				ρ = 90%				
2	777.54	246.10	749.58	239.92	849.18	262.94	1508.90	489.84
4	516.21	202.38	377.49	189.76	532.58	206.18	1096.35	470.49
6	365.29	159.91	190.56	116.29	492.10	78.68	816.91	378.45
8	260.17	170.25	105.97	72.85	688.77	135.64	733.25	209.96
10	125.62	71.64	53.57	23.63	1029.36	213.58	831.24	93.50
12	55.74	47.79	33.23	5.28	1434.71	252.14	1062.66	198.81
				ρ = 85%				
2	330.14	60.27	291.96	69.67	332.59	82.40	639.92	125.99
4	165.55	62.44	98.39	34.68	320.94	33.24	403.70	94.06
6	83.25	53.59	49.73	13.69	637.89	66.67	508.22	55.52
8	50.03	61.60	38.59	4.53	1057.24	88.09	815.04	73.17
10	15.67	14.18	37.40	3.26	1499.05	97.88	1207.63	102.60
12	6.51	1.06	36.92	2.12	1946.45	103.31	1634.61	121.66
				ρ = 80%				
2	183.97	43.24	159.94	33.53	184.83	35.51	343.11	51.10
4	61.21	26.81	54.98	9.48	381.87	25.63	312.74	22.24
6	17.69	11.85	43.47	1.83	796.90	47.10	627.16	37.44
8	7.41	5.43	42.55	2.23	1241.26	54.67	1044.11	54.59
10	4.87	1.12	43.00	1.84	1689.66	59.58	1484.73	63.64
12	4.56	0.48	43.02	2.21	2138.90	63.76	1931.62	69.15

As for SSPR and PR, SSPR outperforms PR for tight due date settings and high shop utilization levels, and underperforms PR for other due date settings and shop utilization levels.

Note that the performance pattern of these methods in terms of standard deviation is similar to the performances of these methods with respect to the mean WET.

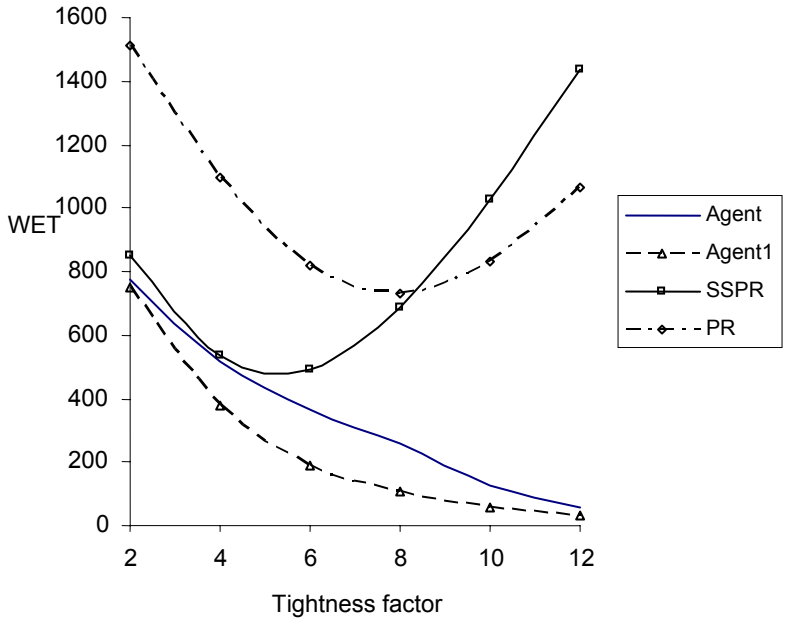

Figure 3.3. Average WET under Different Scheduling Methods

3.8.2 WT under Different Utilizations

When each job has 5 operations, the simulation results of the WTs are presented in Table 3.4. When the utilization level is 90%, Figure 3.4 gives the average WTs under different scheduling methods. From Table 3.4 and Figure 3.4, one can see that as the due date tightness factor increases or shop utilization level decreases, the average WT decreases. This trend holds for all the implemented methods.

For all utilization levels and due date settings, the proposed multi-agent method performs better by using a threshold value $e=2p$ than when $e=0$. In addition, the proposed multi-agent method significantly outperforms PR except for 80% utilization and due date setting $c=2$, and mostly outperforms SSPR except for loose due date settings and low shop utilization levels.

Table 3.4 WT Performance under Different Shop Utilizations

	Agent (e=0)		Agent1 (e=2 p)		SSPR		PR	
c	mean	s. d.	mean	s. d.	mean	s. d.	mean	s. d.
				ρ = 95%				
2	1958.76	564.15	1920.63	550.89	2237.12	605.03	3207.27	867.67
4	1590.73	519.40	1460.39	517.32	1800.36	597.45	2761.33	866.42
6	1257.17	398.77	1026.17	437.75	1391.22	574.22	2331.39	853.52
8	1009.81	366.42	646.81	430.20	1026.53	526.80	1929.94	824.33
10	791.70	320.50	465.67	369.82	726.30	440.08	1565.80	777.43
12	641.19	257.51	282.06	298.92	491.09	334.02	1246.93	709.56
				ρ = 90%				
2	777.53	246.10	748.63	240.36	846.84	263.97	1508.56	490.09
4	516.13	202.41	368.36	194.05	464.12	234.37	1077.98	480.68
6	365.02	159.96	174.63	121.92	219.11	154.05	713.53	432.98
8	259.65	170.35	85.70	77.37	92.56	76.74	446.75	342.77
10	124.81	71.89	30.52	28.33	38.32	34.47	270.55	242.68
12	54.51	48.27	7.51	7.53	16.59	15.44	161.28	161.12
				ρ = 85%				
2	330.12	60.27	284.89	71.36	317.40	85.13	636.44	126.66
4	165.30	62.47	73.83	38.15	87.03	50.45	293.96	109.77
6	82.65	53.63	18.92	15.87	20.50	17.71	121.45	76.42
8	48.99	61.77	6.52	6.17	5.37	5.67	49.99	46.26
10	14.35	14.38	3.56	3.55	1.48	1.78	21.41	26.09
12	4.96	1.30	1.70	0.26	0.49	0.65	10.13	14.43
				ρ = 80%				
2	183.92	43.26	144.80	35.98	146.99	41.41	330.34	52.86
4	60.78	26.81	18.31	11.10	20.57	14.21	90.40	33.55
6	16.87	11.89	3.33	2.12	3.18	3.04	22.68	14.41
8	6.31	5.49	1.71	0.62	0.57	0.72	6.32	5.29
10	3.51	1.29	1.43	0.23	0.06	0.08	1.91	1.75
12	2.99	0.25	1.58	0.20	0.01	0.02	0.67	0.65

3.8.3 WET under Different Numbers of Operations

When jobs have different numbers of operations, Table 3.5 shows the WETs with 90% shop utilization. In Table 3.5, there are 4 scenarios. When n_j = 5, 4, or 3, all jobs have the same number of operations. For the 4th scenario, each job can have 3-5 operations (i.e., n_j ~ uniform(3,5)).

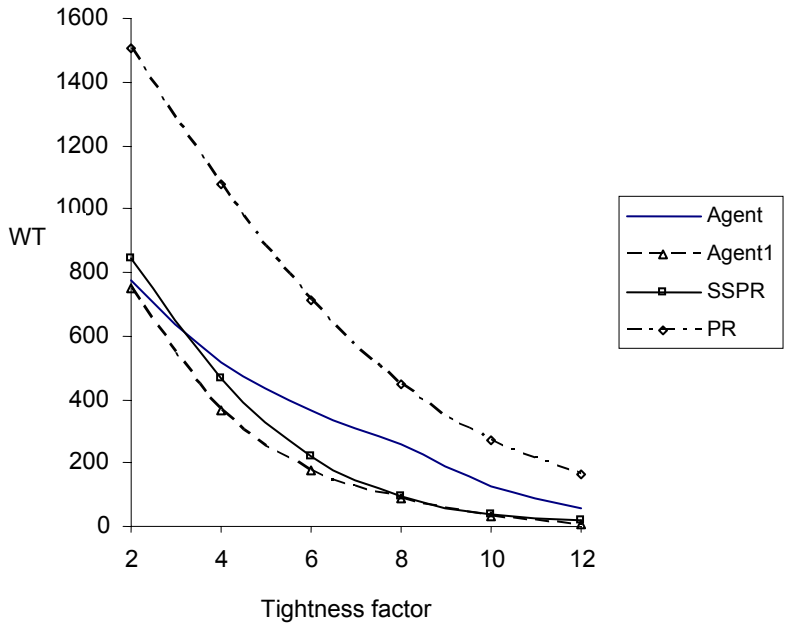

Figure 3.4. Average WT under Different Scheduling Methods

When the multi-agent method is applied and e=0, Figure 3.5 gives the average WET under different numbers of operations. For job j with n_j operations, TOLJ j has no more than (n_j-1) operations remaining and SOLJ j has only one operation to be finished. When the number of operations of a job is decreased from 5 to 3, the operation number ratio of SOLJ/TOLJ is increased on the average and there should be increasing overlapping conflicts and hence decreasing possibilities to schedule SOLJs at their preferred starting times. However, from Table 3.5 and Figure 3.5, the WET performance by the proposed method changes slightly as the number of operations changes.

This observation also holds when jobs have various numbers of operations from the uniform distribution in the range 3-5.

Table 3.5 WET Performance under Different Numbers of Operations

	Agent (e=0)		Agent1(e=2 p)		SSPR		PR	
c	mean	s. d.	mean	s. d.	mean	s. d.	mean	s. d.
				$n_j = 5$				
2	777.54	246.10	749.58	239.92	849.18	262.94	1508.90	489.84
4	516.21	202.38	377.49	189.76	532.58	206.18	1096.35	470.49
6	365.29	159.91	190.56	116.29	492.10	78.68	816.91	378.45
8	260.17	170.25	105.97	72.85	688.77	135.64	733.25	209.96
10	125.62	71.64	53.57	23.63	1029.36	213.58	831.24	93.50
12	55.74	47.79	33.23	5.28	1434.71	252.14	1062.66	198.81
				$n_j = 4$				
2	677.84	359.28	658.64	375.96	843.70	496.50	1123.83	545.80
4	485.29	337.42	371.03	300.69	589.98	406.79	817.83	489.61
6	360.74	297.81	219.08	246.16	522.16	256.55	645.06	357.43
8	229.51	247.71	121.21	160.41	616.23	167.28	628.51	210.94
10	143.30	182.47	71.55	100.17	818.32	221.18	737.39	168.49
12	90.18	135.09	47.29	56.42	1084.85	308.23	937.10	247.61
				$n_j = 3$				
2	698.93	183.51	644.06	190.89	927.42	307.63	1075.95	343.18
4	572.42	178.07	441.98	186.55	696.48	287.81	832.78	330.75
6	475.44	166.67	312.29	150.86	554.21	224.03	659.28	284.17
8	376.13	158.26	209.09	121.36	516.66	131.88	580.55	200.61
10	292.11	127.66	137.06	93.51	574.14	67.04	594.02	107.62
12	229.45	126.28	91.29	73.57	703.06	96.82	683.45	64.77
				n_j = uniform(3,5)				
2	725.08	210.96	681.13	212.08	866.89	296.03	1100.28	308.38
4	507.16	181.32	379.86	166.43	586.58	224.80	783.09	268.51
6	360.18	154.66	204.56	111.06	478.53	91.75	591.16	161.70
8	253.38	125.76	99.39	50.05	543.62	106.56	558.69	65.15
10	172.79	102.18	53.82	20.93	742.82	184.85	668.80	126.08
12	106.52	64.35	33.80	6.94	1022.35	230.24	885.91	193.09

3.8.4 WET under Different Processing Time Distributions

This Section reports the computational results when job processing times follow a different distribution. In particular, we consider uniform, exponential and normal distributions. These computational experiments may reveal the impact of processing time distribution on the ET performance for the proposed method, Table 3.6 shows the WETs with 90% shop utilization.

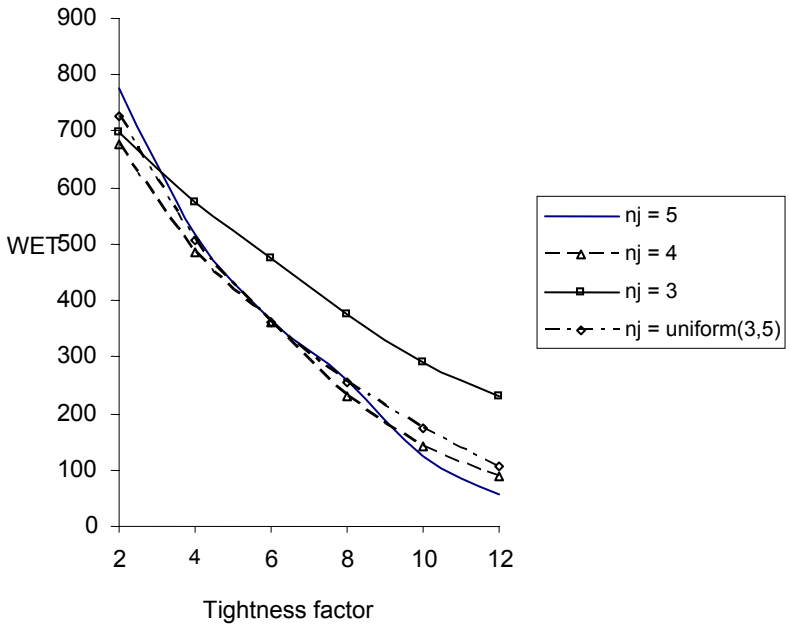

Figure 3.5. Average WET under Different Numbers of Operations

When processing time distribution follows an exponential distribution, we truncate the random number to be a value between 1 and 45. This truncation will assure that the mean operation processing time is roughly 15, the same as the uniform distribution U (1, 30). Similarly, we truncate the random number from a normal distribution N (15, 75). Again, this truncation guarantees that the processing time is at least 1 and the mean is about 15.

From Table 3.6, the proposed multi-agent method significantly outperforms PR and SSPR for all due date settings. When the multi-agent method is applied and e=0, Figure 3.6 gives the average WETs under different processing time distributions. Under the proposed scheduling method, as the due date tightness factor increases, the average WET decreases. When the processing time distributions are exponential and due date setting factors are large (6, 8, 10 and 12), the proposed multi-agent method found schedules in which jobs are completed very close to their due dates. In addition, for uniform distribution and normal distribution, the proposed multi-agent method performs very well by using a threshold value e=2 p . However,

for exponential distribution, the proposed method performs better when e=0 unless for extremely tight due date settings.

Table 3.6 WET Performance under Different Processing Time Distributions

	Agent (e=0)		Agent1(e=2 p)		SSPR		PR	
c	mean	s. d.	mean	s. d.	mean	s. d.	mean	s. d.
p_{ijk} = uniform(1,30)								
2	777.54	246.10	749.58	239.92	849.18	262.94	1508.90	489.84
4	516.21	202.38	377.49	189.76	532.58	206.18	1096.35	470.49
6	365.29	159.91	190.56	116.29	492.10	78.68	816.91	378.45
8	260.17	170.25	105.97	72.85	688.77	135.64	733.25	209.96
10	125.62	71.64	53.57	23.63	1029.36	213.58	831.24	93.50
12	55.74	47.79	33.23	5.28	1434.71	252.14	1062.66	198.81
p_{ijk} = exponential(15)								
2	244.14	51.85	139.66	24.76	175.63	33.08	301.07	38.05
4	92.46	30.70	59.29	9.38	326.55	16.14	286.87	14.72
6	35.25	18.65	48.95	2.80	659.66	36.58	544.53	22.08
8	16.89	9.82	47.17	2.31	1026.72	46.23	884.44	37.57
10	10.33	7.04	46.94	1.30	1400.69	52.52	1247.91	46.72
12	6.83	3.93	47.13	2.68	1776.92	57.28	1616.30	53.24
p_{ijk} = normal(15,75)								
2	507.77	193.31	479.53	194.70	537.18	216.51	985.97	329.34
4	283.27	152.43	186.87	124.02	350.56	122.63	642.47	260.09
6	176.38	107.57	83.74	73.19	509.38	92.08	531.28	131.89
8	102.76	93.40	47.63	36.65	846.45	138.39	651.38	106.07
10	66.11	71.59	36.58	12.69	1248.96	172.32	925.60	178.34
12	41.69	48.40	33.65	5.23	1672.71	190.23	1281.97	234.07

3.8.5 WET under Different Mean Processing Times

When the multi-agent method is used and the mean of the processing time is increased from 15 to 35, Table 3.7 shows the average WETs. For the exponential distribution, processing times are truncated with a mean of 35, a maximum of 65, and a minimum of 20. This truncation assures that the average operation processing time generated is about 35. For the uniform distribution and normal distribution, processing times are truncated with a mean of 35, a maximum of 50, and a minimum of 20.

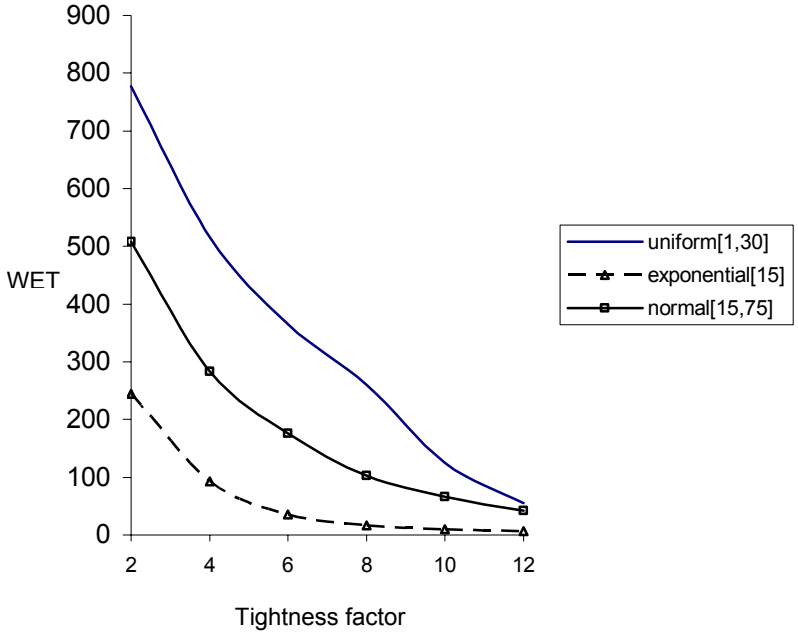

Figure 3.6. Average WET under Different Processing Time Distributions

For every processing time distribution, the average WETs almost increase when the mean of the processing time increases. However, when the mean of the processing time is 15, exponential distribution produces the best results. When the mean of the processing time is 35, uniform distribution produces the best results.

Table 3.7 WET Performance under Different Mean Processing Times

	Multi-agent					
c	2	4	6	8	10	12
Uniform(1,30)	777.54	516.21	365.29	260.17	125.62	55.74
Uniform(20,50)	625.72	525.66	409.46	229.04	168.27	111.15
Exponential(15)	244.14	92.46	35.25	16.89	10.33	6.83
Exponential(35)	1307.44	837.25	627.13	447.42	342.54	261.37
Normal(15,75)	507.77	283.27	176.38	102.76	66.11	41.69
Normal(35,75)	918.82	716.44	686.60	454.66	237.77	190.99

3.8.6 Simulation Time under Different Utilizations

Table 3.8 presents the simulation times to find a complete schedule with over 2000 jobs on 10 machines when each job has 5 operations. When the utilization level is 90%, Figure 3.7 gives the simulation times under different scheduling methods.

Table 3.8 Simulation Time under Different Shop Utilizations (seconds)

	Agent (e=0)			Agent1 (e=2 p)			SSPR			PR		
c	mean	s. d.	max	mean	s. d.	max	mean	s. d.	max	mean	s. d.	max
						ρ = 95%						
2	22.73	5.85	34.44	22.41	5.77	33.45	0.96	0.02	1.00	0.93	0.01	0.94
4	27.00	7.74	42.59	24.05	6.60	37.56	0.98	0.04	1.09	0.93	0.01	0.94
6	35.43	9.49	55.26	27.50	7.48	43.47	0.97	0.05	1.09	0.92	0.02	0.94
8	46.40	14.08	76.38	32.60	9.64	53.84	1.01	0.11	1.34	0.93	0.02	0.97
10	52.41	16.34	87.53	35.47	9.81	55.50	0.99	0.04	1.09	0.93	0.01	0.94
12	63.21	15.36	94.78	40.55	11.29	63.39	0.97	0.03	1.03	0.94	0.02	0.97
						ρ = 90%						
2	10.90	2.69	17.64	10.50	2.22	15.27	0.96	0.03	1.03	0.93	0.01	0.94
4	15.02	4.35	27.16	11.51	2.58	17.04	0.96	0.01	0.97	0.93	0.01	0.94
6	21.78	6.73	38.39	13.77	2.84	20.93	0.97	0.03	1.06	0.92	0.02	0.95
8	28.79	11.79	62.39	17.50	3.24	26.34	1.01	0.05	1.12	0.92	0.02	0.94
10	33.45	10.68	62.53	22.07	3.85	32.90	0.99	0.04	1.06	0.93	0.01	0.94
12	40.73	10.48	67.76	24.35	5.13	38.48	0.96	0.01	0.97	0.93	0.01	0.94
						ρ = 85%						
2	6.13	0.64	7.63	5.86	0.86	7.69	0.96	0.02	1.00	0.93	0.01	0.94
4	9.38	1.10	12.23	7.16	0.58	8.47	0.97	0.02	1.00	0.93	0.01	0.94
6	13.53	2.21	19.61	10.02	0.63	10.94	1.00	0.04	1.06	0.94	0.02	0.96
8	18.22	3.83	29.13	13.45	0.88	14.85	0.97	0.02	1.00	0.93	0.01	0.95
10	22.50	3.03	30.22	16.61	1.18	19.07	0.96	0.02	1.00	0.93	0.01	0.94
12	27.21	4.33	38.56	20.37	1.42	22.93	1.06	0.09	1.29	0.91	0.02	0.93
						ρ = 80%						
2	4.83	0.82	7.05	4.66	0.92	6.94	1.07	0.08	1.22	0.93	0.01	0.94
4	7.56	0.63	8.92	5.66	0.23	6.05	0.96	0.03	1.03	0.93	0.01	0.94
6	10.68	0.50	11.80	8.03	0.28	8.60	0.96	0.02	1.00	0.94	0.02	0.96
8	13.94	1.00	15.77	10.64	0.43	11.61	0.98	0.09	1.25	0.93	0.01	0.94
10	17.33	1.09	19.08	12.98	0.44	13.91	0.95	0.01	0.97	0.94	0.03	0.98
12	21.15	1.06	22.95	15.44	0.68	16.60	0.96	0.02	1.00	0.93	0.01	0.95

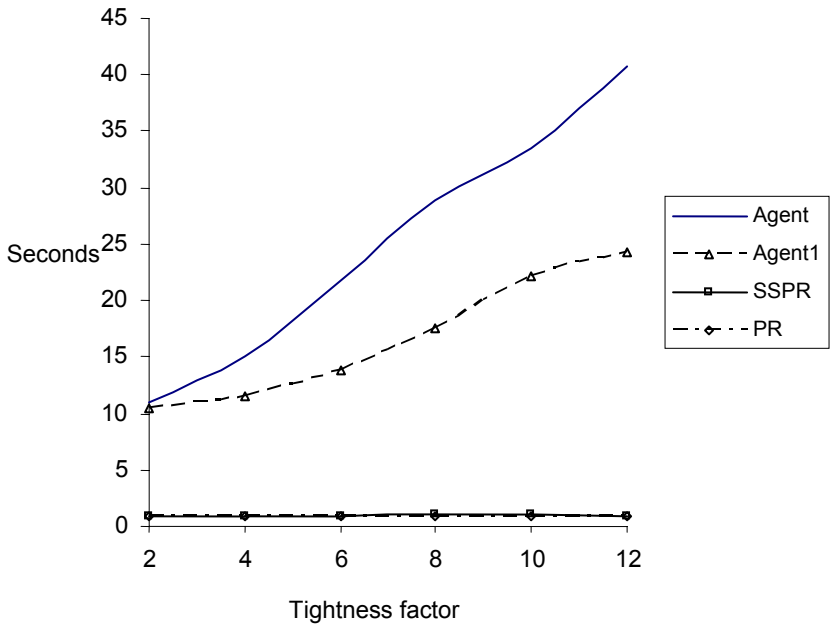

Figure 3.7. Simulation Times under Different Scheduling Methods

From Table 3.8, as the due date tightness factor increases or shop utilization level increases, the simulation time of the proposed method increases. For loose due date settings under all shop utilization levels, the proposed method is faster when $e=2p$ than $e=0$. The largest computer time for all simulation instances is 94.78 seconds. This indicates that the proposed method is computationally efficient from the practical point of view.

One can also see that both PR and SSPR can find schedules for our simulation settings within about 2 seconds. In general, 5 minutes would be a reasonable threshold value for industrial scheduling practice, and thus the propose method can be implemented in real time.

3.9 Summary

A flexible job shop with five work centers is considered to test the proposed method against the existing methods in literature. The computational experiments show that the new

method significantly outperforms the existing methods for WET performance, and the proposed method is insensitive to the number of operations.

In general, the new method also outperforms the implemented methods in terms of WT which has been the primary performance measure against job due dates. This indicates that the proposed method is robust. In addition, the proposed method is very efficient computationally. In particular, it takes less than 1.5 minutes of simulation time on a 1.6GHz PC to find a complete schedule with over 2000 jobs on 10 machines. Such computational efficiency makes the proposed method applicable in real time.

The computational experiments indicate that the due date tightness factor significantly affects the performance of the proposed method for all considered shop load levels. Quick response to customers requires small due date tightness factors. But this can lead to unsatisfactory large tardiness. On the other hand, most jobs can be completed on time with large due date tightness factors. Thus, to set a proper due date tightness factor is a challenging research issue.

Chapter 4

Dynamic Due Date Setting

4.1 Introduction

A multi-agent job routing and sequencing method with ET objectives is discussed in Chapter 3. The method significantly outperforms the existing job routing and sequencing methods. However, the ET performance under high shop utilizations is still not desired and a lot of jobs will be tardy. The reason is that the poor ET performance is not necessary due to bad scheduling. The origin may be wrong commitments of due dates. This chapter discusses dynamic due date setting to address this issue.

4.2 Order Entry in Make-To-Order Companies

In MTO companies, the arrival of customer enquiries cannot be predicted in advance. Whether an enquiry turns into an order depends upon the bid the company gives and how it compares with bids from competitors. Each enquiry from a customer tends to be unique. An enquiry may come with a desired delivery date and merely ask for a price. In this case, MTO companies should check whether the possible order from the enquiry can be manufactured to meet the given due date. If an enquiry requests both a delivery date and a price to be quoted, MTO companies should determine what alternative due dates are feasible and what extra costs in providing extra capacity will be incurred for shorter due dates. It is preferable to select one which best meets the company objectives. This research does not consider price quotation and

just focuses on due date setting. The due date of the corresponding job j for a possible order can be determined by

$$d_j \geq r_j + l_j, \tag{4.1}$$

where r_j is the time when job j is received and l_j its estimated lead time.

4.3 DTWK and DPPW Rules

The TWK and PPW rules are commonly used to set due dates and expressed respectively as

$$d_j = r_j + c_{TWK}\ p_j, \tag{4.2}$$

$$d_j = r_j + p_j + c_{PPW}\ n_j, \tag{4.3}$$

where p_j is the total processing time of job j, n_j the number of operations, c_{TWK} the due date tightness factor for TWK, and c_{PPW} the due date tightness factor for PPW. The due dates may be tight or loose, depending on the value of the parameters c_{TWK} and c_{PPW} set to be small or large.

To eliminate the effect of the due date tightness factor, a dynamic forecasting model to establish IDDs and XDDs is proposed (Bertrand 1983 and Enns 1995). The authors assume that the operation processing times at all machines follow the same distribution and all machines are utilized at the same level. Such a job shop is generally referred to as a uniform shop. Then the average workload W in the shop can be computed by

$$W = M\varphi r, \tag{4.4}$$

where r is the average total remaining processing time per job in the shop, M the number of machines in the shop, and φ the average number of jobs at each machine.

According to Little's law,

$$\varphi = f\lambda\,, \tag{4.5}$$

where f is the average flowtime at each machine and λ is the arrival rate of jobs at each machine. The steady-state utilization ρ of each machine can be expressed as

$$\rho = p\lambda\,, \tag{4.6}$$

where p is the expected processing time per operation. Combining (4.4), (4.5) and (4.6) yields

$$W = Mrf\rho / p\,. \tag{4.7}$$

A dynamic version of (4.7) can be expressed as

$$W_t = Mr_t f_t \rho / p\,. \tag{4.8}$$

If no assembly operations are involved, r_t can be expressed as

$$r_t = W_t / N_t\,, \tag{4.9}$$

where N_t is the number of uncompleted jobs in the shop at time t. Combining (4.8) and (4.9) yields

$$f_t = N_t p /(M\rho). \tag{4.10}$$

Thus the waiting time w_j per operation of job j can be estimated as follows.

$$w_j = f_t - p\,. \tag{4.11}$$

Now, the flowtime of job j can be estimated as $n_j f_t = n_j N_t p /(M\rho) = p_j N_t /(M\rho)$. Setting the flowtime as the estimated lead time for job j leads to the DTWK rule.

$$d_j = r_j + p_j\ N_t /(M\rho)\,. \tag{4.12}$$

Similarly, we can set the DPPW rule as follows.

$$d_j = r_j + p_j + n_j\ (N_t /(M\rho) - 1)p\,. \tag{4.13}$$

Both (4.12) and (4.13) are IDD. The corresponding XDD is obtained by adding a delivery safety allowance.

As pointed out by Enns (1995), when the shop is lightly loaded under DPPW, it is possible that $N_t/(M\rho) < 1$. This means that the flow allowance of a job may be less than its total processing time, which is clearly unreasonable. Therefore, a modified DPPW rule is proposed as follows.

$$d_j = r_j + p_j + n_j \max\{0, N_t/(M\rho) - 1\} p . \tag{4.14}$$

On the other hand, the DTWK and DPPW rules can produce a constant average lateness of jobs (Bertrand 1983). To reduce the constant average lateness, a new due date setting method based on a dynamic feedback mechanism is proposed. It consists of order entry agent (OEA), job routing and sequencing agent (RSA), and information feedback agent (IFA).

4.4 Order Entry Agent

The OEA is responsible for customer enquiries and due date settings for the possible orders. The OEA has the information such as order number and order specification.

The OEA determines job due date by (4.1). The estimated flowtime defined in (4.12) is a good starting point for estimating the lead time of a job. As mentioned earlier, due dates set by (4.12) may be systematically leading to a constant lateness. If the constant lateness is positive, such due dates are set generally too tight. On the other hand, if the constant lateness is negative, such due dates are loose. This motivates to modify (4.12) by introducing a feedback. Consequently a dynamic feedback TWK (DFTWK) rule is obtained as follows.

$$d_j = r_j + p_j N_t/(M\rho) + \Delta L_j , \tag{4.15}$$

where ΔL_j is the average lateness of recently completed jobs at the time when job *j* is received. ΔL_j can be considered as the system feedback in the following sense. If $\Delta L_j < 0$ ($\Delta L_j > 0$), jobs recently completed are early (tardy), which indicates due dates are loose (tight). Therefore,

it is reasonable to adjust due date setting according to (4.15). In addition, DFTWK dose not explicitly include the job pool time, and ΔL_j should be considered to reflect the estimate of the job pool time. For DPPW, we similarly propose a dynamic feedback PPW (DFPPW) rule to assign due dates as follows.

$$d_j = r_j + p_j + n_j \max\{0, N_t/(M\rho) - 1\} p + \Delta L_j . \quad (4.16)$$

The due dates set by (4.15) or (4.16) are job IDDs and job XDDs are obtained by adding some delivery safety allowance. While the original rules (4.2) and (4.3) are parametric, the new rules (4.15) and (4.16) are nonparametric. This is pretty significant in the sense that there is no need to determine a parameter and, therefore, they are easy to be implemented in practice.

An OEA communicates with the RSA and IFA. In particular, when an MTO company receives an enquiry, the OEA requests current shop status and ΔL_j from the RSA and IFA, respectively. It then determines the job due date and releases the job. The OEA protocol is given as follows.

OEA protocol

1) Send the RSA a request to get current shop status.
2) Receive the shop status from the RSA.
3) Send the IFA a request to get ΔL_j.
4) Receive ΔL_j from the IFA.
5) Set the job due date.
6) Release the job.

4.5 Job Routing and Sequencing Agent

The function of job routing and sequencing agent (RSA) is to route and sequence jobs. It has current shop status information such as number of jobs in the pool and on the shop floor, shop utilization, and the total remaining processing time of all jobs at each machine.

The RSA consists of two subagents: job agents (JAs) and machine agents (MAs), which have been discussed in Chapter 3.

The RSA protocol is given as follows.

RSA protocol

1) Receive the request of current shop status from the OEA.
2) Send the shop status to the OEA.
3) Receive a job from the OEA.
4) Select a machine for an operation by the JA.
5) Sequence jobs by the MA.
6) Process an operation by the MA.
7) If a job is not completed, go to step 4.

4.6 Information Feedback Agent

The Information Feedback Agent (IFA) provides a mechanism to estimate the average lateness ΔL_j used in (4.15) and (4.16). It maintains the information of recently completed jobs.

In this research, ΔL_j is determined based on the K most recently completed jobs. In particular, a simple moving average is used as the estimate of ΔL_j. That is,

$$\Delta L_j = \sum_{i=1}^{K} L_i / K. \tag{4.17}$$

The motivation of using the job feedback information is that the average lateness from recently completed jobs reasonably predicts the tightness of job due date setting. The remaining question is to determine what *K* value to use. Observe that some time elapses between a job's arrival and its completion. During this elapsed time, some new jobs may arrive. We define the number of new jobs arrived between the arrival and completion of a job as its lag. Evidently, any job may have an impact on other jobs only when it is in the system. Therefore, it is reasonable to set *K* to be the average lag of all jobs. This can be computed from historical data. Once *K* is determined over a reasonable period of time, it remains stable unless job lags change dramatically.

When the IFA receives a request from the OEA, the IFA determines the average lateness of recently completed jobs and sends it to the OEA. The IFA protocol is given as follows.

IFA protocol

1) Receive a request form the OEA.
2) Determine the average lateness.
3) Send the average lateness to the OEA.

4.7 System Coordination

When a WTO company receives an enquiry for a possible order, the OEA determines the IDD and XDD of the corresponding job of the order, which uses the information from the RSA and IFA. For each operation of any job, a JA requests bids from the machines that can process the operation. When an MA receives the bid request from a JA, it formulates a bid. The JA evaluates all bids from the MAs and selects a machine. Once a machine is selected, the job

is moved to the machine. The machine then sequences and processes the jobs. The coordination activities for a job continue with one operation at a time until the job is finished.

4.8 Analysis of Simulation Study

This section reports the results of computational experiments.

4.8.1 Comparisons among TWK, DTWK and DPPW

The experimental design is considered as the same as in Chapter 3. Each job has 5 operations. Processing times are sampled from a uniform distribution in the range 1-30. When IDDs are determined by (4.12) and (4.14), the means and standard deviations of the WETs are given in Table 4.1.

Table 4.1 WET Performance under DTWK and DPPW

	Multi-agent		SSPR		PR	
	mean	s. d.	mean	s. d.	mean	s. d.
			$\rho = 95\%$			
DTWK	107.94	26.95	481.96	131.29	809.13	210.77
DPPW	48.68	17.11	211.50	35.81	554.32	133.85
			$\rho = 90\%$			
DTWK	38.24	7.19	257.70	59.94	454.78	116.14
DPPW	20.59	3.67	163.93	27.49	333.58	67.70
			$\rho = 85\%$			
DTWK	15.11	4.53	164.47	23.25	278.60	38.86
DPPW	8.32	1.86	127.64	15.78	227.73	29.27
			$\rho = 80\%$			
DTWK	11.86	3.29	129.50	13.75	207.90	19.79
DPPW	7.30	0.77	111.66	10.93	180.43	16.57

From Figure 4.1 and Figure 4.2, the multi-agent method is the best for the WET performance. SSPR significantly outperforms PR under various utilization levels.

From Table 3.3, when due dates are determined by TWK, The computational experiments indicate that the due date tightness factor significantly affects the performances of the multi-agent method, PR and SSPR for all utilization levels. From Table 4.1, when due dates are determined by DTWK and DPPW, they do not select the due date tightness factor. Furthermore, DTWK and DPPW significantly outperform TWK for all considered situations under PR and SSPR. When the multi-agent method is used, DTWK and DPPW outperform TWK except for TWK under loose due date settings and low utilization levels.

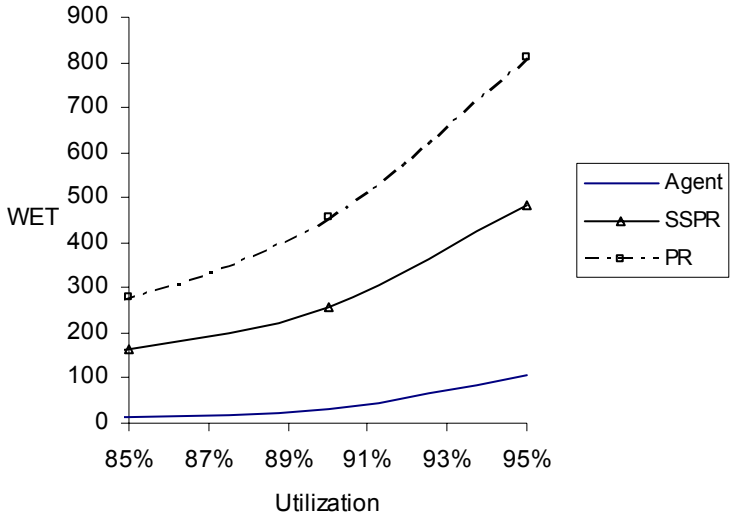

Figure 4.1. Average WET Using DTWK

One can see that DPPW significantly outperforms DTWK for WET performance. When the utilization level is 95%, Figure 4.3 gives the average WETs under DPPW and DTWK. If the multi-agent method is applied, the average WET by DPPW is only 45.1% of that by DTWK. Similarly, under SSPR and PR, the average WETs by DPPW are only 43.9% and 68.5% of those of by DTWK, respectively. This trend can also be observed when the utilization levels are 90%, 85%, and 80%.

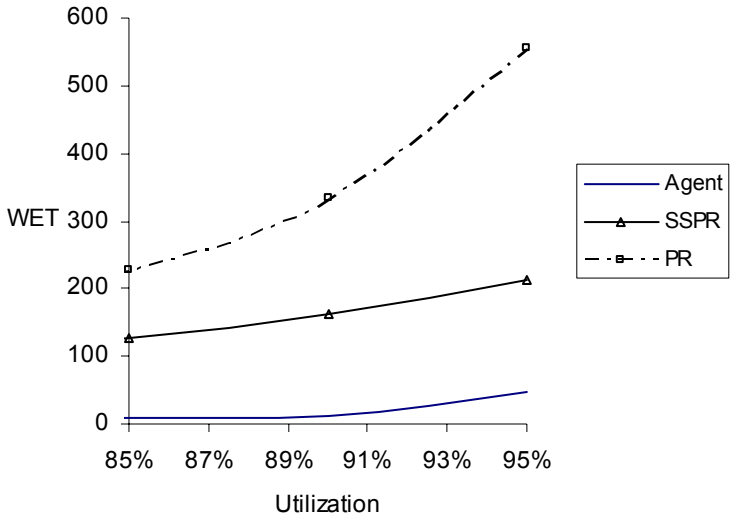

Figure 4.2. Average WET Using DPPW

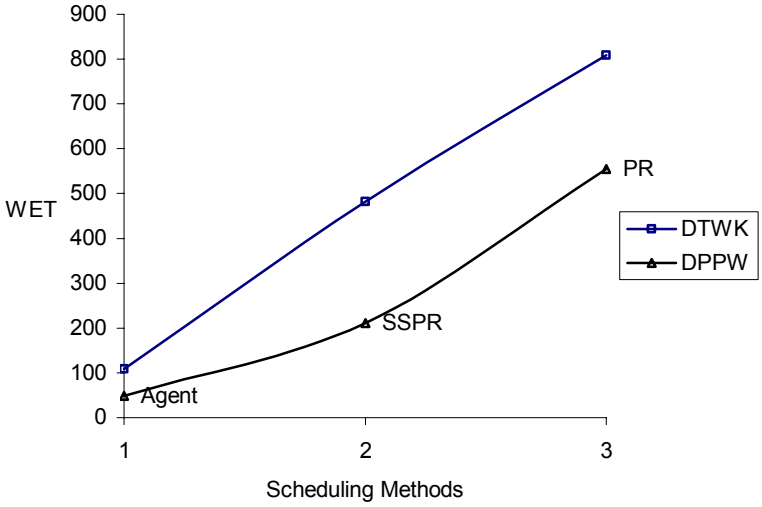

Figure 4.3. Average WET under 95% Utilization

When jobs have different processing time distributions, Table 4.2 shows the WETs with 90% shop utilization. In Table 4.2, there are 3 scenarios. First, processing times are sampled from a uniform distribution in the range 1-30. Second, processing times are randomly generated from a truncated exponential distribution with a mean of 15, a maximum of 45, and a minimum of 1. This truncation assures that the average operation processing time generated is about 15.

Third, processing times are sampled from a truncated normal distribution with a mean of 15, a variance of 75, a maximum of 30, and a minimum of 1.

From Table 4.2, one can also observe that DPPW significantly outperforms DTWK for WET performance for all different processing time distributions and scheduling methods tested.

Table 4.2 WET Performance under Different Processing Time Distributions

	Multi-agent		SSPR		PR	
	mean	s. d.	mean	s. d.	mean	s. d.
p_{ijk} = uniform(1,30)						
DTWK	38.24	7.19	257.70	59.94	454.78	116.14
DPPW	20.59	3.67	163.93	27.49	333.58	67.70
p_{ijk} = exponential(15)						
DTWK	74.68	6.54	186.05	19.48	292.51	27.08
DPPW	26.26	9.69	121.56	9.13	208.80	15.10
p_{ijk} = normal(15,75)						
DTWK	44.11	6.44	130.64	17.17	221.40	28.84
DPPW	22.38	4.98	100.68	9.39	184.04	16.57

When processing time distributions are uniform, the proposed multi-agent method produces the best results. Meanwhile, normal distribution is better than exponential distribution for proposed multi-agent method.

For PR and SSPR, normal distribution is the best. Exponential distribution is better than uniform distribution.

4.8.2 Comparisons among Four Rules

This section presents the effectiveness of the proposed DFTWK and DFPPW versus DTWK and DPPW. When processing times are randomly generated from a truncated exponential distribution with a mean of 15, a maximum of 45, and a minimum of 1, Table 4.3 shows the average WETs with 90% shop utilization.

Table 4.3 WET Performance under Different Scheduling Methods

Methods	DTWK	DFTWK	DPPW	DFPPW
Multi-agent	74.68	48.48	26.26	17.78
SSPR	105.84	99.82	82.75	86.01
PR	162.98	156.38	136.31	139.73

From Table 4.3, when the proposed multi-agent scheduling method is applied, DFPPW and DFTWK significantly outperform DPPW and DTWK, respectively. However, for PR and SSPR, there is no significant difference between DTWK and DFTWK, DPPW and DFPPW.

For the proposed scheduling method, the average WETs are given in Table 4.4 under different shop utilizations. From Table 4.4, it can be seen that under all shop utilizations, DFPPW and DFTWK significantly outperform DPPW and DTWK, respectively. This means introducing the job completion feedback ΔL_j into due date setting works extremely well. Of the 4 due date setting rules considered, DFPPW is the best for WET performance. It is worth noting that DPPW significantly outperforms DTWK for WET performance under all 4 utilization levels tested.

Table 4.4 WET Performance under Different Shop Utilizations

ρ	DTWK	DFTWK	DPPW	DFPPW
95%	83.79	52.40	31.67	20.18
90%	74.68	48.48	26.26	17.78
85%	61.52	39.58	25.62	16.56
80%	53.59	34.99	21.90	15.12

When jobs have different processing time distributions, Table 4.5 shows the WETs with 90% shop utilization. Under different processing time distributions, DFPPW and DFTWK also significantly outperform DPPW and DTWK, respectively.

When the processing time distributions are uniform, the proposed multi-agent method produces the best WET performance. Normal distribution is better than exponential

distribution. In addition, DFPPW produces the least average WET for all processing time distributions and due date setting rules.

Table 4.5 WET Performance under Different Processing Time Distributions

p_{ijk}	DTWK	DFTWK	DPPW	DFPPW
Uniform(1,30)	38.24	26.65	20.59	14.70
Exponential(15)	74.68	48.48	26.26	17.78
Normal(15,75)	44.11	30.61	22.38	15.39

When the mean of the processing time is increased from 15 to 35, Table 4.6 shows the average WETs. For the exponential distribution, processing times are truncated with a mean of 35, a maximum of 65, and a minimum of 20. This truncation assures that the average operation processing time generated is about 35. For the uniform distribution and normal distribution, processing times are truncated with a mean of 35, a maximum of 50, and a minimum of 20.

From Table 4.6, uniform distribution produces the best results in the three processing time distributions.

Table 4.6 WET Performance under Different Mean Processing Times

p_{ijk}	DTWK	DFTWK	DPPW	DFPPW
Uniform(20,50)	56.21	38.14	43.59	28.53
Exponential(35)	91.34	60.80	52.19	35.26
Normal(35,75)	67.58	44.98	52.07	35.63

4.8.3 WET and WT Performance of Four Rules

From (4.12), when $N_t/(M\rho) < 1$, the flow allowance of a job may be less than its total processing time, which is clearly unreasonable. Therefore, a DTWK rule is proposed as follows.

$$d_j = r_j + p_j \max\{1, N_t/(M\rho)\}. \tag{4.18}$$

Similarly, we can set the DFTWK rule as follows.

$$d_j = r_j + p_j \max\{1, N_t/(M\rho)\} + \Delta L_j . \tag{4.19}$$

When jobs have different processing time distributions, Table 4.7 shows the WETs and WTs with 90% shop utilization. In Table 4.7,

$$c_j = (d_j - r_j)/p_j, \tag{4.20}$$

where c_j is the due date tightness factor.

Under different processing time distributions, DFPPW and DFTWK also significantly outperform DPPW and DTWK, respectively. When the processing time distributions are uniform, the proposed multi-agent method produces the best WET performance. In addition, DFPPW produces the least average WET for all processing time distributions and due date setting rules.

From Tables 4.5 and 4.7, there is no significant different results between (4.12) and (4.18), (4.15) and (4.19).

Table 4.7 WETs and WTs under Different Processing Time Distributions

p_{ijk}	DTWK	DFTWK	DPPW	DFPPW
		WET		
Uniform(1,30)	38.15	26.05	20.59	14.70
Exponential(15)	75.13	49.95	26.26	17.78
Normal(15,75)	44.02	30.34	22.38	15.39
		WT		
Uniform(1,30)	34.20	21.53	17.74	11.55
Exponential(15)	72.29	46.23	23.48	14.71
Normal(15,75)	40.57	26.24	19.65	12.40
		c_j		
Uniform(1,30)	4.93	5.41	6.47	7.06
Exponential(15)	3.27	4.04	6.77	7.65
Normal(15,75)	3.62	4.08	5.43	6.05

When the mean of the processing time is increased from 15 to 35, Table 4.8 shows the average WETs. From Table 4.8, uniform distribution produces the best results in the three processing time distributions.

Table 4.8 WETs and WTs under Different Mean Processing Times

p_{ijk}	DTWK	DFTWK	DPPW	DFPPW
		WET		
Uniform(20,50)	56.04	38.97	43.59	28.53
Exponential(35)	91.15	60.71	52.19	35.26
Normal(35,75)	67.56	46.84	52.07	35.63
		WT		
Uniform(20,50)	50.68	33.02	40.07	24.73
Exponential(35)	86.22	55.04	48.17	30.98
Normal(35,75)	62.53	41.17	48.88	32.07
		c_j		
Uniform(20,50)	5.58	6.10	6.29	6.85
Exponential(35)	5.77	6.37	7.05	7.62
Normal(35,75)	5.47	6.01	6.03	6.76

4.8.4 Performance of Different Earliness and Tardiness Weights

The above computational experiments use the earliness and tardiness weights in (3.4) independently sampled from the uniform distribution in the range of 1-5. In practice, however, tardiness penalty should not be smaller than earliness penalty, since tardiness can lead to customer dissatisfaction. Therefore, it is of interest to consider such scenarios that $\alpha_j \le \beta_j$, for all *j*. In particular, two cases are considered in this section: $\alpha_j = \beta_j$ and $\beta_j = 2\alpha_j$. When $\alpha_j = \beta_j$, α_j is sampled from integer uniform distribution in the range of 1-5. Tables 4.9 and 4.10 report the computational results. When $\beta_j = 2\alpha_j$, α_j is sampled from integer uniform distribution in the range of 1-3. Tables 4.11 and 4.12 report the computational results.

When $\alpha_j = \beta_j$, from Table 4.9, DFPPW and DFTWK also significantly outperform DPPW and DTWK respectively under different processing time distributions. When the

processing time distributions are uniform, the proposed multi-agent method produces the best WET and WT performance. Normal distribution is better than exponential distribution.

Table 4.9 WETs and WTs under Same Weights

p_{ijk}	DTWK	DFTWK	DPPW	DFPPW
		WET		
Uniform(1,30)	38.05	26.21	20.42	14.83
Exponential(15)	75.27	49.11	27.60	17.77
Normal(15,75)	44.75	30.15	22.28	15.42
		WT		
Uniform(1,30)	33.92	21.64	17.44	11.70
Exponential(15)	72.33	45.39	24.74	14.65
Normal(15,75)	41.18	25.90	19.45	12.40
		c_j		
Uniform(1,30)	3.29	5.41	6.53	7.07
Exponential(15)	3.24	3.99	6.84	7.68
Normal(15,75)	3.64	4.09	5.46	6.11

From Tables 4.7 and 4.9, there is no significant different results between $\alpha_j = \beta_j$ and when α_j and β_j are independently sampled.

When the mean of the processing time is increased from 15 to 35, Table 4.10 shows the average WETs and WTs. From Table 4.10, uniform distribution produces the best results in the three processing time distributions.

When $\beta_j = 2\alpha_j$, from Table 4.11, DFPPW and DFTWK also significantly outperform DPPW and DTWK respectively under different processing time distributions. When the processing time distributions are uniform, the proposed multi-agent method produces the best WET and WT performance. Normal distribution is better than exponential distribution. In addition, DFPPW produces the least average WET for all processing time distributions and due date setting rules.

Table 4.10 Performance under Different Processing Time Distributions

p_{ijk}	DTWK	DFTWK	DPPW	DFPPW
		WET		
Uniform(20,50)	55.06	39.81	42.30	27.60
Exponential(35)	90.19	61.58	54.93	35.08
Normal(35,75)	69.65	45.53	54.34	36.91
		WT		
Uniform(20,50)	49.58	33.65	38.65	23.82
Exponential(35)	85.12	55.73	50.93	30.92
Normal(35,75)	64.31	39.78	50.99	33.42
		c_j		
Uniform(20,50)	5.52	6.09	6.30	6.74
Exponential(35)	5.73	6.45	7.06	7.65
Normal(35,75)	5.54	6.00	6.09	6.65

From Tables 4.9 and 4.11, compared with $\alpha_j = \beta_j$, WETs and WTs are increased when $\beta_j = 2\alpha_j$. For c_j, there is no significant difference.

Table 4.11 WETs and WTs under Different Weights

p_{ijk}	DTWK	DFTWK	DPPW	DFPPW
		WET		
Uniform(1,30)	49.86	33.62	25.79	19.18
Exponential(15)	98.16	64.90	36.60	24.49
Normal(15,75)	58.92	40.22	29.09	20.12
		WT		
Uniform(1,30)	46.46	29.79	23.01	16.24
Exponential(15)	95.73	61.84	33.80	21.42
Normal(15,75)	56.13	36.85	26.51	17.29
		c_j		
Uniform(1,30)	4.89	5.30	6.37	6.94
Exponential(15)	3.24	3.94	6.64	7.38
Normal(15,75)	3.60	4.08	5.35	5.90

When the mean of the processing time is increased from 15 to 35, Table 4.12 shows the average WETs and WTs. From Table 4.12, uniform distribution produces the best results in the three processing time distributions.

Table 4.12 Performance under Different Mean Processing Times

p_{ijk}	DTWK	DFTWK	DPPW	DFPPW
		WET		
Uniform(20,50)	71.50	50.11	54.08	38.67
Exponential(35)	117.23	81.35	70.17	50.05
Normal(35,75)	91.98	63.20	73.88	49.14
		WT		
Uniform(20,50)	66.46	44.46	50.38	34.58
Exponential(35)	112.54	75.92	65.82	45.18
Normal(35,75)	87.33	57.90	70.57	45.39
		c_j		
Uniform(20,50)	5.48	5.91	6.13	6.63
Exponential(35)	5.68	6.40	6.90	7.62
Normal(35,75)	5.48	5.94	6.01	6.58

4.9 Summary

The ET performances of SSPR, PR and the proposed scheduling method under various shop utilizations are significantly affected by the tightness factor when due dates are determined by TWK. However, DTWK and DPPW do not select the tightness factor and use dynamic shop load information for due date setting. The computational experiments show that DTWK and DPPW outperform TWK. DPPW produces the best WET performance in these three rules.

DTWK and DPPW can produce a constant average lateness. The DFTWK and DFPPW rules do not need to select the tightness factor. They also use the feedback information of recently completed jobs. The simulation results indicate that for WET performance, DFTWK and DFPPW significantly outperform the existing DTWK and DPPW, respectively, and DFPPW performs much better than DFTWK. The strong performance of DFTWK and DFPPW suggests that the job completion feedback mechanism introduced in due date setting works very well.

Chapter 5

Multi-Agent Workload Control Methodology

5.1 Introduction

Job release control has a significant effect on system performance in the PPC of MTO companies. Specifically, they can reduce WIP inventory and job flowtimes. This chapter discusses job release control and then proposes a multi-agent WLC methodology that simultaneously deals with due date setting, job release and scheduling in MTO companies.

5.2 Job Release Control

The arrival of orders in MTO companies is a stochastic process over time. As each order tends to be different and requires varying routings and processing times, the number of jobs does not characterize the total work on the shop floor very well. Instead, we define workload as the total remaining processing time of all jobs on the shop, and workload norm as the maximum amount of workload allowed on the shop floor. Evidently, as the workload norm changes, WIP will also change. According to Little's law, the relationship among WIP, shop flowtime and throughput rate can be expressed as

$$WIP = f_s \lambda , \tag{5.1}$$

where WIP is the average WIP level on the shop, f_s the average job flowtime and λ the throughput rate. At low WIP, a considerable throughput reduction can be expected. However,

when WIP rises to a certain point, the throughput ceases to increase (Bergamaschi *et al.* 1997). On the other hand, the flowtime continues to rise. This means that a critical WIP level exists for a good system performance. This phenomenon may also exist between WIP and other performance measures such as WET. The workload norm corresponding to the critical WIP level is called the critical norm. The critical norm may be determined empirically. The purpose of job release control is to fix WIP at the critical level.

5.3 Job Release Agent

The job release agent (JRA) provides a job release mechanism to control WIP. In this research, continuous aggregate loading (CAGG) is used as the job release mechanism. CAGG performs well for the flowtime and tardiness related criteria (Sabuncuoglu and Karapinar 1999). By CAGG, if current workload falls below the workload norm, jobs are continuously released from the job pool to the shop until the workload reaches its norm. On the other hand, if the workload exceeds its norm when a new job arrives, it will wait in the job pool. Consequently, one can restrain the WIP level while maintaining certain system performance.

The JRA receives jobs from the OEA and puts the jobs in the pool. It also communicates with the RSA and determines current workload. The JRA continuously monitors current workload. If the pool is not empty and the workload is less than its norm, the JRA releases jobs from the pool to the shop. In this research, jobs are released in EDD order (i.e., the job with the earliest due date in the job pool is released first).

The JRA protocol is given as follows.

JRA protocol

1) Receive a job from the OEA.
2) Put the job in the pool.
3) If the pool is not empty, send the RSA a request to get current shop status, and go to step 4; otherwise wait until a new job arrives, and go to step 1.
4) Receive current shop status from the RSA.
5) Calculate current workload.
6) If current workload is less than its norm, release a job in EDD order. Go to step 3.

5.4 System Architecture

To integrate due date setting, job release, and scheduling, a multi-agent WLC methodology is developed. The system architecture is sketched in Figure 5.1.

There are four types of agents in the methodology: OEA, JRA, RSA, and IFA, as defined earlier. All agents consist of three modules. The first is the data module, which carries certain information for the use of the agent. The communication module consists of protocols for the agent to communicate with each other. Finally, the decision module makes decisions using the information from the data module and communication module.

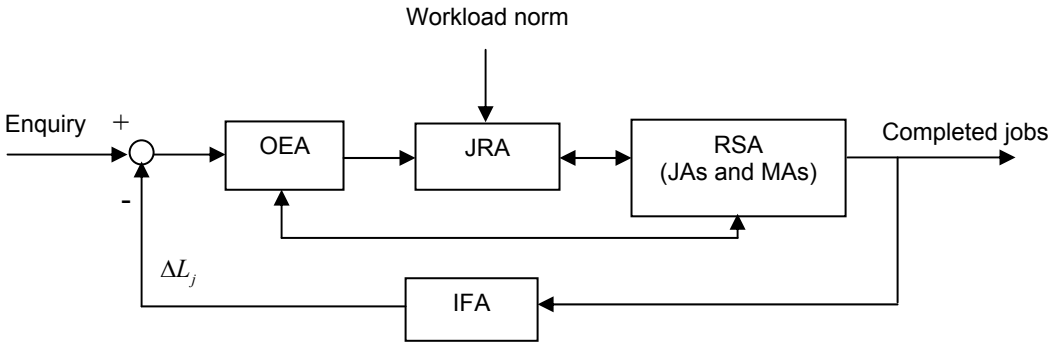

Figure 5.1. System Architecture of Multi-Agent Workload Control

5.5 System Coordination

One challenging issue faced by multi-agent systems is the agent coordination that arises due to the interdependencies and interactions among agents. Individual agents in a multi-agent system have their individual sub-goals. A coordination mechanism integrates the individual sub-goals of agents into the system goal through message-passing.

Three types of interdependencies are identified (Sikora and Shaw 1998). However, there are only two valid interdependencies in multi-agent WLC systems.

5.5.1 Temporal Interdependency

The activities of agents may be interdependent due to the fact that an activity of one agent may be restricted by the activities of other agents. For example, certain activities cannot be started until other activities are finished. In WLC systems, only after the due date of an order is set, the corresponding job can be released, and then the job can be processed on the shop floor.

5.5.2 Sub-goal Interdependency

The sub-goals of agents resulting from task decomposition may be overlapping or interdependent. Agents have to exchange information during the process of decision making. For instance, to determine the due date of a job by (4.14) or (4.15), an OEA needs to know the average job lateness that is determined by the IFA. For job release, the JRA should get current workload that can only be determined by the RSA.

The coordination process in the multi-agent WLC methodology is as follows. When an MTO company receives an enquiry from a customer, the OEA determines the job due date using the information from the RSA and IFA. Then, the job is put in the job pool. If the pool is

not empty and current workload falls below its norm, the JRA releases a job to the shop floor. After a job is released, a JA for the job is created. The JA requests bids from the machines that can process the next operation. When an MA receives the bid request from the JA, it formulates a bid. The JA then evaluates all bids from the MAs and selects a machine. After a machine is selected, the job is moved to the machine. The MA sequences the jobs and the machine processes the job. The routing and sequencing process is repeated until a job is finished.

5.6 Discrete Event Simulation

The WLC simulation system is implemented using an object-oriented approach and C++. The simulation begins by setting the simulation clock to zero, initializing cumulative statistics to zero, generating any initial events, and defining the system state at time zero. The simulation program then cycles, repeatedly passing the current least-time event to the appropriate event subroutines until the simulation is over. At each step, after finding the imminent event but before calling the event subroutine, the simulation clock is advanced to the time of the imminent event. Next, the appropriate event subroutine is called to execute the imminent event, update cumulative statistics, and generate future events. Executing the imminent event means that the system states and entity attributes are changed to reflect the fact that the event has occurred. The simulation algorithm is given as follows.

Simulation algorithm

1) Set CLOCK=0 and cumulative statistics=0.
2) Generate initial system state.
3) Call time advance algorithm to find imminent event.
4) Call appropriate event subroutine.

5) Advance COLOCK to imminent event time.

6) If the simulation is not over, go to step 3; otherwise go to step 7.

7) Generate the simulation report.

Time advance algorithm

1) If a job arrives, go to step 2; otherwise go to step 5.

2) Set its operation sequence, work centers, processing times, earliness and tardiness weights.

3) Call the OEA algorithm to get its due date.

4) Put the job in the job pool.

5) Call the RSA algorithm to get current workload.

6) If current workload is not greater than its norm, go to step 7; otherwise go to step 8.

7) Call the JRA algorithm to release a job to the shop floor.

8) If a job is not completed, call the RSA algorithm to schedule an operation.

5.7 Simulation Study

In order to evaluate the effectiveness of the proposed methodology, we provide an extensive simulation study based on randomly generated problem instances. The shop environment will be similar to the one used in Chapter 3. Each job has 5 operations. However, processing times are randomly generated from a truncated exponential distribution with a mean of 15, a maximum of 30, and a minimum of 1. In addition, results of previous studies indicate that the choice of job release mechanisms is the most critical in the range of 85% to 90% (Ragatz and Mabert 1988). Therefore, 85% and 90% utilizations are considered in the simulation of job release control.

This section presents simulation experiments to investigate the performance of the job release control. The performance measure of WET is important but do not capture all the impact of job release control. Flowtime and maximum WIP in terms of number of jobs provide some indication of shop congestion, and lead time reflects responsiveness to customer orders. Therefore, the average lead time (LT), average flowtime (FT) and maximum WIP are also reported. Immediate release or no job release control is used to compare the performance of CAGG.

Tables 5.1 and 5.2 present the computational results when the shop utilization is 90%, and Tables 5.3 and 5.4 report the computational results at 85% utilization. When the shop utilization is 90%, Tables 5.5-5.8 report the computational results under different processing time distributions. Note that in all 8 tables, the norm of ∞ means no release control. In addition, we only implement the two proposed due date setting rules.

5.7.1 System Performance Using DFPPW

From Table 5.1 and Figure 5.2, we can see that using DFPPW, the average WET and LT almost remains constant when the norm decreases from ∞ to 1000. As the lead time is the sum of the pool time and the flowtime, it means that for a job, the increase in its pool time offsets the reduction in its waiting time on the shop floor for any norm no smaller than 1000.

When the norm further decreases from 1000, the average WET and LT increase quickly. On the other hand, the average FT and maximum WIP continue to decrease as the norm decreases, and both decrease more rapidly as the norm is below 1000. Therefore, 1000 should be considered as the critical norm. Under the critical norm, the average WET and LT are little changed, but the average FT and maximum WIP are reduced by 10.5% and 20.1%, respectively, compared with no release control.

Table 5.1 Performance under 90% Shop Utilization Using DFPPW

Norm	WET	LT	FT	Max WIP
∞	18.13	272.89	272.89	60.55
1700	18.41	271.83	271.46	59.80
1600	18.23	271.65	270.91	59.20
1500	18.35	271.85	270.24	58.90
1400	18.31	272.91	269.88	58.05
1300	18.40	270.57	264.80	56.80
1200	18.06	271.49	261.90	54.95
1100	18.22	270.25	253.10	52.65
1000	*18.04*	*272.84*	*244.14*	*48.40*
900	18.84	278.06	230.27	44.35
800	20.39	295.33	212.90	39.25
700	24.12	357.12	194.29	34.00

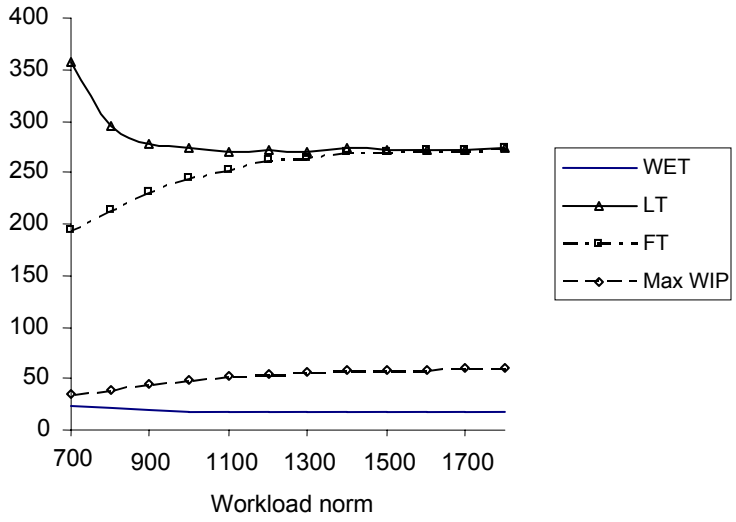

Figure 5.2. Performance under 90% Utilization Using DFPPW

5.7.2 System Performance Using DFTWK

When DFTWK is used for due date setting, one can observe from Table 5.2 and Figure 5.3 that 1000 is also the critical norm at 90% utilization. As the norm is reduced from ∞ to the critical norm, the average WET and LT remain roughly constant, but the average FT and maximum WIP are reduced by 3.7% and 18.0%, respectively.

Table 5.2 Performance under 90% Shop Utilization Using DFTWK

Norm	WET	LT	FT	Max WIP
∞	44.62	171.77	171.77	41.15
1700	45.00	172.00	171.95	42.55
1600	44.92	172.58	172.47	41.45
1500	44.47	171.23	171.00	41.45
1400	44.41	170.93	170.41	40.55
1300	44.67	171.53	170.49	38.95
1200	44.82	172.03	170.28	37.80
1100	44.64	172.76	169.26	35.90
1000	*44.35*	*171.15*	*165.37*	*33.75*
900	45.15	174.29	163.15	31.75
800	47.56	181.34	157.95	28.80
700	49.99	191.83	149.80	25.85

Under both proposed due date setting rules, the job release control reduces flowtime and maximum WIP at no expense of worse WET and lead time performances. Shorter job flowtime and smaller maximum WIP may lead to less shop congestion. This indicates that the job release control is effective.

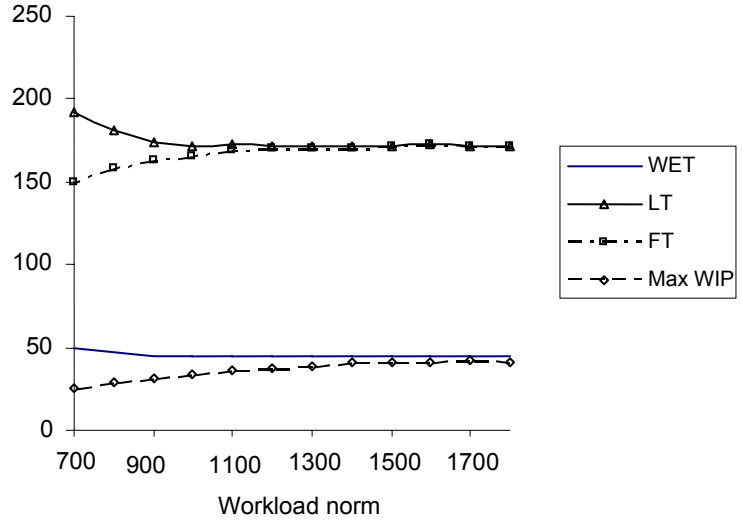

Figure 5.3. Performance under 90% Utilization Using DFTWK

As seen in Chapter 4, DFPPW significantly outperforms its counterpart DFTWK in terms of WET performance, when there is no job release control. From Tables 5.1 and 5.2, the same conclusion holds when there is job release control. For example, when the norm is set equal to the critical norm of 1000, the average WET by DFPPW is only 59.3% of that by DFTWK. However, DFTWK significantly outperforms DFPPW for average LT, FT and maximum WIP performance measures. While less WET means better JIT production, smaller LT and FT indicates quicker customer response and less shop congestion, respectively. Therefore, there is a trade-off between DFPPW and DFTWK, in terms of multiple system performance measures.

5.7.3 System Performance under Different Utilizations

From Tables 5.3 and 5.4, and Figures 5.4 and 5.5, one can see similar observations. However, the critical norm decreases to 800 from 1000 at 90% utilization. The critical norm is affected by shop utilization levels. When the shop utilization is reduced from 90% to 85%, all the average WET, LT, FT and maximum WIP also decrease. This would be expected.

Table 5.3 Performance under 85% Shop Utilization Using DFPPW

Norm	WET	LT	FT	Max WIP
∞	16.56	255.90	255.90	56.05
1600	16.60	255.52	255.38	55.90
1500	16.64	252.66	252.26	55.35
1400	16.67	256.40	255.17	55.55
1300	16.60	253.54	251.28	54.40
1200	16.58	254.14	249.59	53.40
1100	16.35	252.58	244.34	50.95
1000	16.63	247.62	235.19	48.40
900	16.33	254.38	231.57	46.50
800	*17.01*	*252.07*	*215.66*	*41.85*
700	17.74	273.50	201.08	36.15
600	20.87	311.71	178.78	30.30

When the utilization is decreased from 90% to 85%, the advantage of job release control increases. As the workload norm decreases from ∞ to the critical norm, the average WET and LT are almost unchanged at both utilizations. However, the reductions of average FT and maximum WIP increase from 10.5% to 15.7%, and 20.1% to 25.3%, respectively, under DFPPW. Similarly, such reductions in average FT and maximum WIP increase from 3.7% to 5.4% and from 18.0% to 20.7%, respectively, under DFTWK.

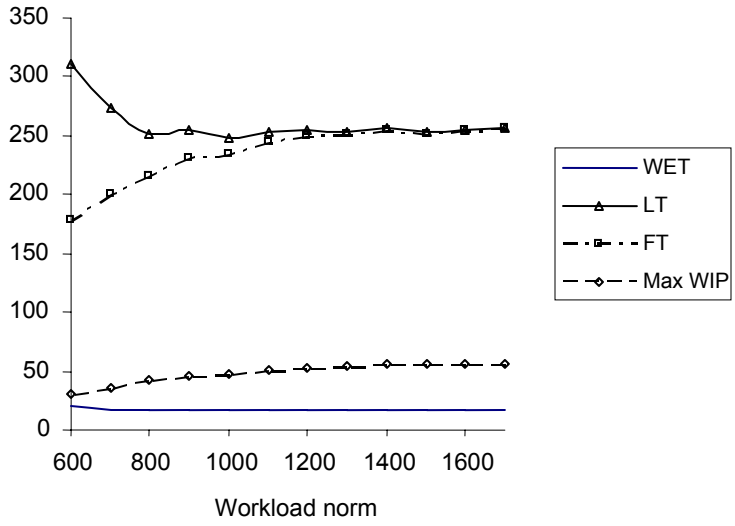

Figure 5.4. Performance under 85 % Utilization Using DFPPW

Table 5.4 Performance under 85% Shop Utilization Using DFTWK

Norm	WET	LT	FT	Max WIP
∞	39.58	155.00	155.00	36.15
1600	39.58	155.10	155.09	36.30
1500	39.56	154.85	154.82	36.10
1400	39.19	153.70	153.60	36.40
1300	38.90	153.75	153.51	36.25
1200	38.96	153.62	153.17	35.65
1100	38.92	153.70	152.61	34.00
1000	38.73	154.01	152.07	33.20
900	39.23	153.86	150.04	30.55
800	*39.07*	*153.96*	*146.56*	*28.65*
700	40.92	159.62	142.92	26.05
600	44.04	170.81	136.11	23.55

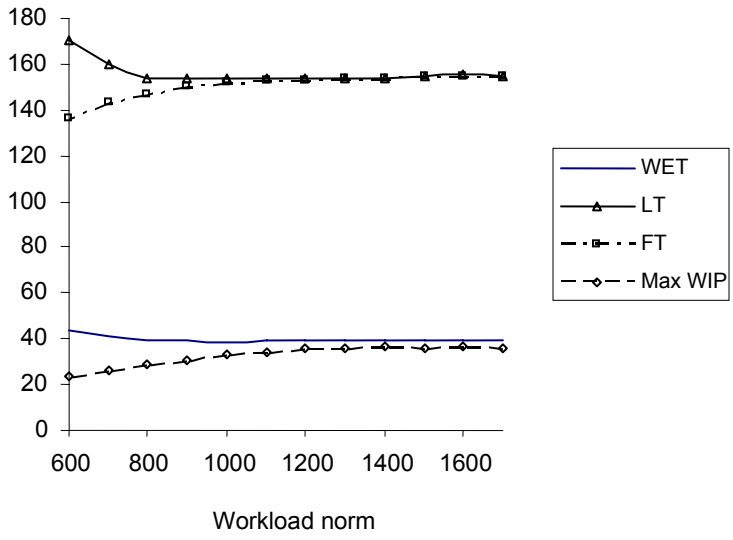

Figure 5.5. Performance under 85 % Utilization Using DFTWK

5.7.4 Performance under Different Processing Time Distributions

The above sections report the results when processing times are randomly generated from a truncated exponential distribution with a mean of 15, a maximum of 30, and a minimum of 1. This section reports the simulation results when jobs have different processing time distributions.

When the shop utilization is 90% and processing times are sampled from a truncated normal distribution with a mean of 15, a variance of 75, a maximum of 30, and a minimum of 1, Tables 5.5 and 5.6 give the WETs using DFPPW and DFTWK, respectively. When the shop utilization is 90% and processing times are sampled from a uniform distribution in the range 1-30, Tables 5.7 and 5.8 give the WETs using DFPPW and DFTWK, respectively.

From Table 5.5 and Figure 5.6, 1700 should be considered as the critical norm. Under the critical norm, compared with no release control, the average WET and LT are little changed, but the average FT and maximum WIP are reduced by 20.3% and 33.1%, respectively.

Compare with Table 5.1, when the processing time distributions are changed from exponential to normal, the critical norm is increased from 1000 to 1700. At the critical norm, average WET is reduced from 18.04 to 15.32. The reduction of average FT increases from 10.5% to 20.3% and the reduction of maximum WIP increases from 20.1% to 33.1%. However, the average LT is also increased from 272.84 to 449.51.

Table 5.5 Performance under Normal Distribution Using DFPPW

Norm	WET	LT	FT	Max WIP
∞	15.39	454.32	454.32	93.60
3500	15.32	454.16	453.50	93.30
3000	15.42	454.92	451.43	93.40
2500	15.46	452.14	437.36	88.80
2200	15.34	452.74	418.96	80.90
2000	15.00	450.58	402.20	75.50
1700	*15.32*	*449.51*	*362.22*	*62.60*
1500	16.59	456.56	333.68	53.80
1400	16.84	463.01	315.90	50.00
1300	17.95	468.18	298.07	46.20
1200	21.32	492.11	278.58	42.70

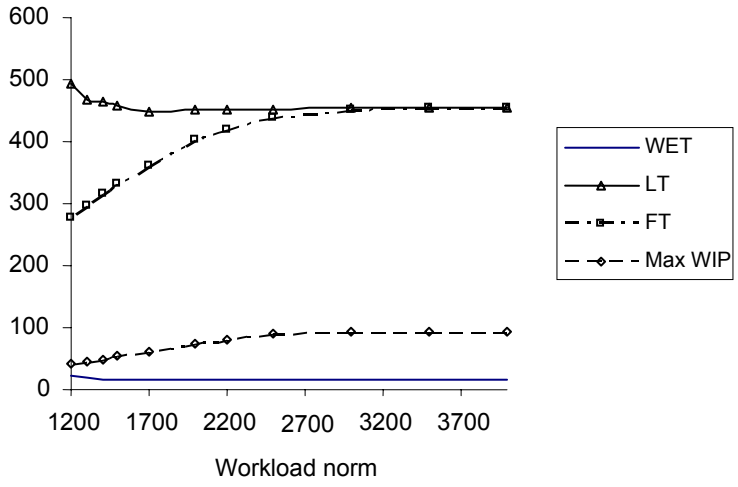

Figure 5.6. Performance under Normal Distribution Using DFPPW

When DFTWK is applied, form Table 5.6 and Figure 5.7, one can see similar observations. From Tables 5.6 and Figure 5.7, 1700 is also the critical norm. Under the critical norm, compared with no release control, the average WET and LT are little changed, but the average FT and maximum WIP are reduced by 15.5% and 31.4%, respectively.

Table 5.6 Performance under Normal Distribution Using DFTWK

Norm	WET	LT	FT	Max WIP
∞	30.61	346.55	346.55	74.80
3500	30.38	345.44	344.89	73.70
3000	30.25	344.43	341.58	72.30
2500	30.33	343.81	336.29	69.00
2200	29.48	335.89	321.39	63.00
2000	30.10	338.91	314.43	58.90
1700	*30.40*	*340.93*	*292.84*	*51.30*
1500	32.19	346.62	277.04	46.50
1400	33.40	351.25	267.21	44.20
1300	35.03	358.41	256.55	41.00
1200	39.46	387.64	245.46	37.90

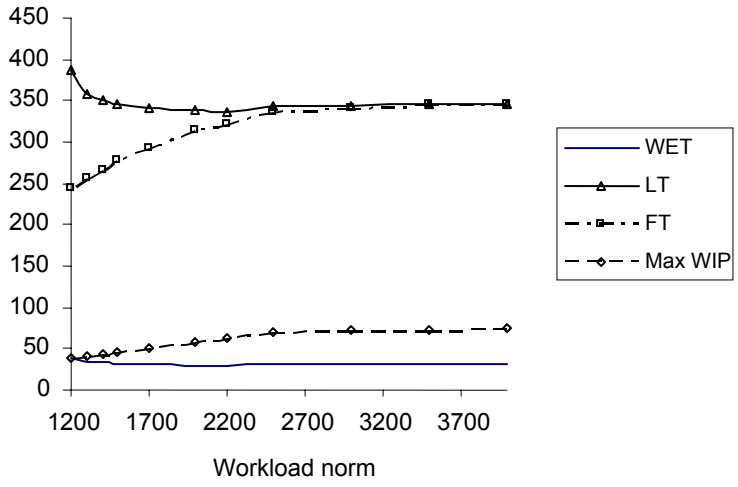

Figure 5.7. Performance under Normal Distribution Using DFTWK

Compare with Table 5.2, when the processing time distributions are changed from exponential to normal, the critical norm is also increased from 1000 to 1700. At the critical norm, average WET is reduced from 44.35 to 30.40. The reduction of average FT increases from 3.7% to 15.5% and the reduction of maximum WIP increases from 18.0% to 31.4%. However, the average LT is also increased from 171.15 to 340.93.

When processing times are sampled from a uniform distribution in the range 1-30, Tables 5.7 gives the WETs using DFPPW.

Table 5.7 Performance under Uniform Distribution Using DFPPW

Norm	WET	LT	FT	Max WIP
∞	14.70	536.49	536.49	99.60
5000	14.62	536.45	536.45	99.60
4500	14.63	536.38	536.18	99.40
4000	14.52	535.70	534.40	99.30
3500	14.10	533.97	530.14	97.40
3000	14.59	535.47	521.50	92.90
2500	14.09	531.72	497.98	81.40
2200	14.41	536.74	468.11	75.30
2000	*14.47*	*534.36*	*437.37*	*66.60*
1700	16.14	546.14	385.61	56.40
1500	21.08	572.23	342.61	50.40

From Tables 5.7 and Figure 5.8, 2000 should be considered as the critical norm. Under the critical norm, compared with no release control, the average WET and LT are little changed, but the average FT and maximum WIP are reduced by 18.5% and 33.1%, respectively.

Compare with Table 5.1, when the processing time distributions are changed from exponential to uniform, the critical norm is increased from 1000 to 2000. At the critical norm, average WET is reduced from 18.04 to 14.47. The reduction of average FT increases from

10.5% to 18.5% and the reduction of maximum WIP increases from 20.1% to 33.1%. However, the average LT is also increased from 272.84 to 534.36.

When DFTWK is applied, one can see similar observations from Table 5.8 and Figure 5.9. 2000 is also the critical norm. Under the critical norm, compared with no release control, the average WET and LT are little changed, but the average FT and maximum WIP are reduced by 16.0% and 31.9%, respectively.

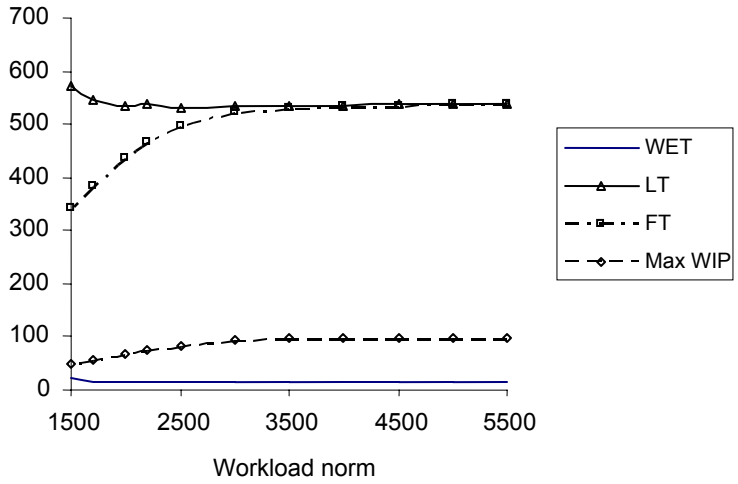

Figure 5.8. Performance under Uniform Distribution Using DFPPW

Table 5.8 Performance under Uniform Distribution Using DFTWK

Norm	WET	LT	FT	Max WIP
∞	26.49	451.73	451.73	88.60
5000	26.49	451.73	451.73	88.60
4500	26.34	451.11	450.98	88.60
4000	26.10	448.91	447.92	87.70
3500	26.21	463.99	460.25	86.88
3000	25.22	443.80	435.46	80.90
2500	24.61	438.88	416.25	72.20
2200	25.26	436.95	395.34	65.20
2000	*25.63*	*440.77*	*379.38*	*60.30*
1700	28.38	453.66	344.67	52.30
1500	33.97	481.96	315.08	47.00

Compare with Table 5.2, when the processing time distributions are changed from exponential to normal, the critical norm is also increased from 1000 to 2000. At the critical norm, average WET is reduced from 44.35 to 25.63. The reduction of average FT increases from 3.7% to 16.0% and the reduction of maximum WIP increases from 18.0% to 31.9%. However, the average LT is also increased from 171.15 to 379.38.

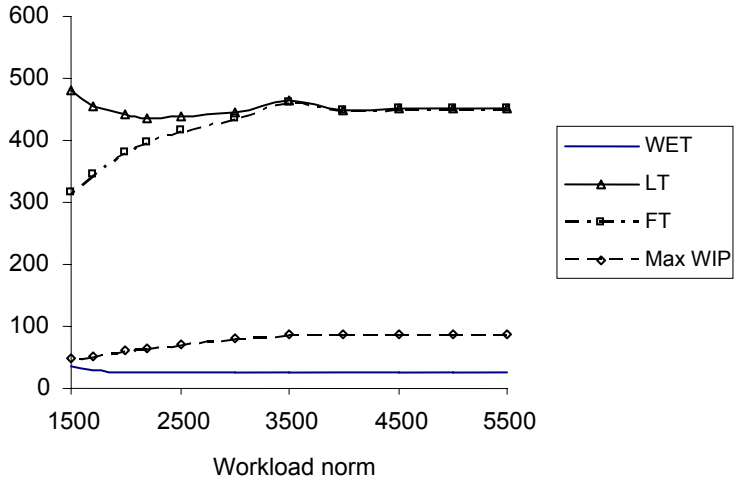

Figure 5.9. Performance under Uniform Distribution Using DFTWK

In addition, there is only a little difference between the uniform distributions and normal distributions of processing times for all performance under DFPPW and DFTWK.

5.8 Summary

This new multi-agent WLC methodology simultaneously deals with due date setting, job release and scheduling in real time. Under job release control, DFTWK and DFPPW can consider job pool times and thus eliminate the need to estimate job pool times in due date setting.

The computational results show that the proposed WLC methodology reduces job flowtimes and shop WIP pretty significantly, without worsening ET and lead time performances. As should be expected, the critical workload norm decreases as shop utilization decreases. However, under the same utilization and processing time distributions, both DFTWK and DFPPW result in the same critical norm. This may indicate that the critical norm is not affected by how due dates are set.

Under the considered utilization levels, DFPPW leads to better WET performance but longer lead times and more congestion than DFTWK. If a company attempts to complete jobs as close to their due dates as possible, DFPPW is much better than DFTWK. However, if the priority is customer response, DFTWK produces better results.

Chapter 6

Conclusions

6.1 Summary of Work

This dissertation has proposed a multi-agent WLC methodology with earliness and tardiness objectives. This methodology simultaneously deals with due date setting, job release and scheduling. It can be used as a PPC method in real time for MTO companies.

Two new due date setting rules, DFTWK and DFPPW, are developed to establish job due dates. They do not need to select the tightness factor and use a feedback mechanism to dynamically adjust due date setting. Both new rules are nonparametric and easy to be implemented in practice. Under job release control, DFTWK and DFPPW can consider job pool times and, thus, eliminate the need to estimate job pool times in due date setting. The simulation results indicate that DFPPW leads to better WET performance but longer lead times and more congestion than DFTWK. According to different performance requirements, MTO companies can choose a suitable one from these two due date setting rules. If a company attempts to complete jobs as close to their due dates as possible, DFPPW is much better than DFTWK. However, if the priority is customer response, DFTWK produces better results.

The job release control significantly reduces job flowtimes and shop WIP inventory, without worsening ET and lead time performances. An important task for job release control is to determine critical workload norm. This research concludes that the critical workload norm

decreases as shop utilization decreases. Under the same utilization and processing time distributions, both DFTWK and DFPPW result in the same critical norm.

A multi-agent scheduling method with job earliness and tardiness objectives in a flexible job shop environment is proposed. A new job routing and sequencing mechanism is developed. In this mechanism, different criteria for two kinds of jobs are proposed to route these jobs. Job sequencing enables to hold a job that may be completed too early. Two sequencing algorithms based on existing methods are developed to deal with these two kinds of jobs. The computational experiments show that the multi-agent scheduling method significantly outperforms PR and SSPR for WET performance, and the proposed method is insensitive to the number of operations. The proposed method also outperforms PR and SSPR in terms of WT performance which has been the primary performance measure against job due dates. This indicates that this proposed method is robust. In addition, the proposed method is very efficient computationally. Such computational efficiency makes the proposed method applicable in real time.

The proposed methodology is implemented in a flexible job shop environment. The computational experiments show that the proposed WLC methodology is very effective for the PPC in MTO companies. In addition, the computational results indicate that the proposed methodology is extremely fast and can be implemented in real time.

6.2 Future Research Directions

In the proposed multi-agent job routing and sequencing method, SOLJ sequencing algorithm reschedules all SOLJs by the MA algorithm (Mazzini and Armentano 2001). However, the MA algorithm results in more tardiness and less earliness. In practice, however, tardiness penalty should not be smaller than earliness penalty, since tardiness can lead to

customer dissatisfaction. Further research will investigate more effective single machine sequencing algorithm to reduce tardiness.

DFPPW usually leads to better ET performance but longer lead times and more congestion than DFTWK. The performance measure for the selection of DFPPW and DFTWK is a very interesting future research direction. The use of economic objectives can be considered.

For the workload control, under the same utilization and processing time distributions, both DFTWK and DFPPW result in the same critical norm. This may indicate that the critical norm is not affected by how due dates are set. Further investigation is necessary to confirm this observation. On the other hand, it is observed that the critical norm decreases as the shop utilization decreases. A future research would be to further investigate this effect.

References

Alidee, B. (1994). Minimizing absolute and squared deviation of completion times from due dates. *Production and Operations Management*, 3, 133-147.

Amaro, G. M., Hendry, L. C. and Kingsman, B. G. (1999). Competitive advantage, customization and a new taxonomy for non make-to-stock companies. *International Journal of Operations and Production Management*, 19, 349- 371.

Anderson, E. J. and Nyirenda, J. C. (1990). Two new rules to minimize tardiness in a job shop. *International Journal of Production Research*, 28, 2277- 2292.

Aydin, M. E. and Oztemel, E. (2000). Dynamic job-shop scheduling using reinforcement learning agents. *Robotics and Autonomous Systems*, 33, 169-178.

Baker, K. R. (1984). Sequencing rules and due date assignments in a job shop. *Management Science*, 30, 1093-1104.

Baker, K. R. and Bertrand, J. W. M. (1982). A dynamic priority rule for sequencing against due dates. *Journal of Operations Management*, 3, 37-42.

Baker, K. R. and Kanet, J. J. (1983). Job shop scheduling with modified due dates. *Journal of Operations Management*, 4, 11-22.

Baker, K. R. and Scudder, G. D. (1990). Sequencing with earliness and tardiness penalties: a review. *Operations Research*, 38, 22-36.

Baykasoglu, A. (2002). Linguistic-based meta-heuristic optimization model for flexible job shop scheduling. *International Journal of Production Research*, 40, 4523-4543.

Balas, E., Lenstra, J. K. and Vazacopoulos, A. (1995). One machine scheduling with delayed precedence constraints. *Management Science*, 41, 94-109.

Bergamaschi, D., Cigolini, R., Perona, M. and Portioli, A. (1997). Order review and release strategies in a job shop environment: a review and a classification. *International Journal of Production Research*, 35, 399–420.

Bertrand, J. W. M. (1983a). The effect of workload dependent due dates on job shop performance. *Management Science***,** 29, 799–816.

Bertrand, J. W. M. (1983b). The use of workload information to control job lateness in controlled and uncontrolled release production systems. *Journal of Operations Management,* 3, 79-92.

Bertrand, J. W. M. and Muntslag, D. R. (1993). Production control in engineering-to-order firms. *International Journal of Production Research*, 30/31, 3-22.

Bertrand, J. W. M. and Van Ooijen, H. P. G. (2002). Workload based order release and productivity: a missing link. *Production Planning and Control*, 13, 665-678.

Blazewicz, J., Dmschke, W. and Pesch, E. (1996). The job shop scheduling problem: conventional and new solution techniques. *European Journal of Operational Research*, 93, 1-33.

Carroll, D. C. (1965). *Heuristic sequencing of jobs with single and multiple components.* Ph. D dissertation, MIT.

Cavalieri, S., Garetti, M., Macchi, M. and Taisch, M. (2000). An experimental benchmarking of two multi-agent architectures for production scheduling and control. *Computers in Industry*. 43, 139-152.

Chambers, J. B. (1996). *Classical and flexible job shop scheduling by tabu search*. Ph.D. dissertation, University of Texas at Austin, Austin, Texas.

Chang, F. C. R. (1997). Heuristics for dynamic job shop scheduling with real-time updated queueing time estimates. *International Journal of Production Research,* 35, 651-665.

Chen, D., Luh, P. B., Thakur, L. S. and Moreno Jr., J. (2003). Optimization-based manufacturing scheduling with multiple resources, setup requirements, and transfer lots. *IIE Transactions*, 35, 973-985.

Cheng, T. C. E. and Gupta, M. C. (1989). Survey of scheduling research involving due date determination decisions. *European Journal of Operational Research,* 38, 156-166.

Cheng, T. C. E. and Jiang, J. (1998). Job shop scheduling for missed due-date performance. *Computers & Industrial Engineering,* 34, 297-307.

Conway, R. W. and Maxwell, W. L. (1962). Network dispatching by the shortest operation discipline. *Operations Research*, 10, 51-73.

Conway, R.W. (1965). Priority dispatching and job lateness in a job shop. *Journal of Industrial Engineering***,** 16, 228–237.

Conway, R. W., Maxwell, W. L. and Miller, L. W. (1967). *Theory of scheduling.* Addison-Wesley Inc., Reading, MA.

Croce, F. D. and Trubian, M. (2002). Optimal idle time insertion in early-tardy parallel machines scheduling with precedence constraints. *Production Planning and Control*. 13, 133-142.

Cutkosky, M. R., Tenenbaum, J. M. and Glicksman, J. (1996). Madefast: collaborative engineering over the Internet. *Communication of the ACM*, 39, 78-87.

Das, T. K., Gosavi, A., Mahadevan, S. and Marchalleck, N. (1999). Solving semi-markov decision problems using average reward reinforcement learning. *Management Science*, 45, 560-574.

Eilon, S. and Chowdhury, I.G. (1976). Due dates in job shop scheduling. *International Journal of Production Research***,** 14, 223–237.

Enns, S.T. (1994). Job shop lead time requirements under conditions of controlled delivery performance. *European Journal of Operational Research***,** 77, 429–439.

Enns, S. T. (1995). A dynamic forecasting model for job shop flowtime prediction and tardiness control. *International Journal of Production Research*, 33, 1295-1312.

Enns, S. T. (1998). Lead time selection and the behavior of work flow in job shops. *European Journal of Operational Research*, 109, 122-136.

Fredenhall, L. D. and Melnyk, S. A. (1995). Assessing the impact of reducing demand variance through improved planning on the performance of a dual resource constrained job shop. *International Journal of Production Research*, 33, 1521–1534.

Fry, T.D., Philipoom, P.R. and Markland, R.E. (1989). Due date assignment in a multistage job shop. *IIE Transactions*, 21, 153–161.

Gee, E. S. and Smith, C. H. (1993). Selecting allowance policies for improved job shop performance. *International Journal of Production Research***,** 31, 1839–1852.

Heady, R. B. and Zhu Z. (1998). Minimizing the sum of job earliness and tardiness in a multimachine system. *International Journal of Production Research*, 36, 1619–1632.

Hendry, L. C. and Kingsman, B. G. (1989). Production planning systems and their applicability to make to order companies, *European Journal of Operational Research*, 40, 1-15.

Hendry, L. C., Kingsman, B. G. and Cheung, P. (1998). The effect of workload control (WLC) on performance in make-to-order companies. *Journal of Operations Management*, 16, 63-75.

Henrich, P., Land, M. and Gaalman G. (2004). Exploring applicability of the workload control concept. *International Journal of Production Economics*, 90, 187–198.

Hodgson, T. J., Cormier, D., Weintraub, A. J. and Zozom, Jr. A. (1998). Note. satisfying due dates in large job shops. *Management Science*, 44, 1442-1446.

Hodgson, T. J., King, R. E., Thoney, K., Stanislaw, N., Weintraub, A. J. and Zozom, Jr. A. (2000). On satisfying due dates in large job shops: idle time insertion. *IIE Transactions*, 32, 177-180.

Hong, J. and Prabhu, V.V. (2004). Distributed reinforcement learning control for batch sequencing and sizing in just-in-time manufacturing systems. *Applied Intelligence,* 20, 71–87.

Hsu, S. Y. and Sha, D. Y. (2004). Due date assignment using artificial neural networks under different shop floor control strategies. *International Journal of Production Research,* 42, 1727-1745.

Huang, S. H., Zhang, H. and Smith, M. L. (1995). A progressive approach for the integration of process planning and scheduling. *IIE Transactions*, 27, 456-464.

Huang, C. L., Huang, Y. H., Chang, T. Y., Chang, S. H., Chung, C. H., Huang, D. T. and Li, R. K. (1999). The construction of production performance prediction system for semiconductor manufacturing with artificial neural networks. *International Journal of Production Research*, 37, 1387–1402.

Ip, W.H., Li, Y., Man, K.F. and Tang, K.S. (2000). Multi-product planning and scheduling using genetic algorithm approach. *Computers and Industrial Engineering*, 38, 283-296.

Jayamohan, M. S. and Rajendran, C. (2004). Development and analysis of cost-based dispatching rules for job shop scheduling. *European Journal of Operational Research*, 157, 307-321.

Kacem, I., Hammadi, S. and Borne, P. (2002). Approach by localization and multiobjective evolutionary optimization for flexible job-shop scheduling problems. *IEEE Transactions on Systems, Man and Cybernetics, Part C: Applications and Reviews*, 31, 1-13.

Kanet, J. J. (1988). Load-limited order release in job shop scheduling systems. *Journal of Operations Management*, 7, 44–58.

Kanet, J. J. and Christy, D.P. (1989). Manufacturing systems with forbidden early shipment: implications for setting manufacturing lead times. *International Journal of Production Research*, 27, 783–792.

Kanet, J. J. and Hayya, J. C. (1982). Priority dispatching with operation due dates in a job shop. *Journal of Operations Management*, 2, 155-163.

Kanet, J. J. and Zhou, Z. (1993). A decision theory approach to priority dispatching for job shop scheduling. *Production and Operations Management*, 2, 2-14.

Kingsman, B. G. (2000). Modelling input-output workload control for dynamic capacity planning in production planning systems. *International Journal of Production Economics*, 68, 73–93.

Krothapalli, N. K. C. and Deshmukh, A.V. (1999). Design of negotiation protocols for multi-agent manufacturing systems. *International Journal of Production Research,* 37, 1601-1624.

Kutanoglu, E. and Sabuncuoglu I. (1999). An analysis of heuristics in a dynamic job shop with weighted tardiness objectives. *International Journal of Production Research,* 37, 165-187.

Land, M. J. and Gaalman, G. (1996). Workload control concepts in job shops: a critical assessment. *International Journal of Production Economics*, 46–47, 535–548.

Land, M. J. and Gaalman, G. (1998). The performance of workload control concepts in job shops: improving the release method. *International Journal of Production Economics*, 56–57, 347–364.

Leung, Joseph Y.-T. (2002). A dual criteria sequencing problem with earliness and tardiness penalties. *Naval Research Logistics*, 49, 422-431.

Li, L., Tang, H., Wu, Z., Gong, J., Gruidl, M., Zou, J. Tockman, M. and Clark, R. A. (2004). Data mining techniques for cancer detection using serum proteomic profiling. *Artificial Intelligence in Medicine*, 32, 71-83.

Liao, C. J. and You, C. T. (1993). An improved formulation for the job shop scheduling problem. *Journal of Production Research Society*, 43, 1047-1054.

Lu, T. P. and Yih, Y. (2001). An agent-based production control framework for multiple-line collaborative manufacturing. *International Journal of Production Research,* 39, 2155-2176.

Luh, P. B., Chen, D. and Thakur, L. S. (1999). Effective approach for job-shop scheduling with uncertain processing requirements. *IEEE Transactions on Robotics and Automation*, 15, 328-339.

Maione, G. and Naso, D. (2003). A genetic approach for adaptive multiagent control in heterarchical manufacturing systems. *IEEE Transactions on Systems, Man and Cybernetics*, *Part A*, 33, 573-588.

Mastrolilli, M. and Gambardella, L. M. (2000). Effective neighborhood functions for the flexible job shop problem. *Journal of Scheduling,* 3, 3-20.

Maturana, F. and Norrie, D. H. (1996). Multi-agent mediator architecture for distributed manufacturing. *Journal of Intelligent Manufacturing,* 7, 257-270.

Mazzini, R. and Armentano, V. A. (2001). A heuristic for single machine scheduling with early and tardy costs. *European Journal of Operational Research*, 128, 129-146.

Melnyk, S. A. and Ragatz, G. L. (1989). Order review/release systems: research issues and perspectives. *International Journal of Production Research*, 27, 1081–1096.

Melnyk, S. A., Tan, K. C., Denzler, D. R. and Fredendall, L. (1994). Evaluating variance control, order review/release and dispatching: a regression analysis. *International Journal of Production Research*, 32, 1045–1061.

Moses, S. A. (1999). Due date assignment using feedback control with reinforcement learning. *IIE Transactions,* 31, 989-999.

Morton, T. E. and Ramnath, P. (1992). *Guided forward tabu/beam search for scheduling very large dynamic job shops*. Technical Report 1992-47, Graduate School of Industrial Administration, Carnegie-Mellon University.

Mosheiov, G. (2003). Scheduling unit processing time jobs on an m-machine flow-shop. *Journal of the Operational Research Society*, 54, 437-44.

Muda, M. S. and Hendry, L. C. (2003). The SHEN model for MTO SME's: a performance improvement tool. *International Journal of Operations and Production Management*, 23, 470-486.

Nagendra Prasad, M.V., Lesser, V.R., and Lander, S.E. (1998). Learning Organizational Roles for Negotiated Search in a Multi-agent System. *International Journal of Human-Computer Studies,* 48, 51-67.

Nasr, N. and Elsayed, E. (1990). Job shop scheduling with alternative machines. *International Journal of Production Research*, 28, 1595-1609.

Perona, M. and Portioli, A. (1998). The impact of parameters setting in load oriented manufacturing control. *International Journal of Production Economics*, 55, 133-142.

Philipoom, P. R., Rees, L. R. and Wiegmann, L. (1994). Using artificial neural networks to determine internally-set due-date assignments for shop scheduling. *Decision Sciences*, 25**,** 825-847.

Pinedo, M. (2002). *Scheduling Theory, Algorithms and Systems*. Prentice Hall.

Pontrandolfo, P., Gosavi, A., Okogbaa, O. G. and Das, T. K. (2002). Global supply chain management: a reinforcement learning approach. *International Journal of Production Research*, 40, 1299-1317.

Ragatz, G. L. and Mabert, V. A. (1984). A simulation analysis of due date assignment rules. *Journal of Operations Management,* 5, 27-39.

Ragatz, G. L. and Mabert, V. A. (1988). An evaluation of order release mechanisms in a job-shop environment. *Decision Science,* 19, 167-189.

Raman, N. (1995). Input control in job shops. *IIE Transactions*, 27, 201–209.

Ren, H. (2000). *Multi-agent scheduling and its applications in job shops with due date related objectives*. Ph.D. dissertation, University of South Florida, Tampa, Florida.

Ronald, R. and Uzsoy, R. (2001). Experimental evaluation of heuristic optimization algorithms: a tutorial. *Journal of Heuristics*, 7, 261-304.

Russell, R. S., Dar-el, E.M. and Taylor II, B.W. (1987). A comparative analysis of the COVERT job sequencing rule using various shop performance measures. *International Journal of Production Research*, 25, 1523-1539.

Saad, A., Kawamura, K. and Biswas, G. (1997). Performance evaluation of contract net-based heterarchical scheduling for flexible manufacturing systems. *Intelligent Automation and Soft Computing*, 3, 229-248.

Sabuncuoglu, I. and Comlekci, A. (2002). Operation-based flowtime estimation in a dynamic job shop. *Omega*, 30, 423-442.

Sabuncuoglu, I. and Karapinar, H. Y. (1999). Analysis of order review/release problems in production systems. *International Journal of Production Economics*, 62, 259-279.

Shafaei, R. and Brunn, P. (1999). Workshop scheduling using practical (inaccurate) data-Part 1: The performance of heuristics scheduling rules in a dynamic job shop environment using a rolling time horizon approach. *International Journal of Production Research*, 37, 3913-3925.

Shafaei, R. and Brunn, P. (2000). Workshop scheduling using practical (inaccurate) data-Part 3: A framework to integrate job release, routing and scheduling functions to create a robust predictive schedule. *International Journal of Production Research*, 38, 85-99.

Shaw, M. J. (1988). Dynamic scheduling in cellular flexible manufacturing systems: a framework for networked decision making. *Journal of Manufacturing Systems,* 7, 83-94.

Shen W. and Barthes J.P. (1997). An experimental environment for exchanging engineering design knowledge by cognitive agents. In Mantyla M., Finger S. and Tomiyama, T., (Eds.), *Knowledge Intensive CAD-2*, Chapman and Hall, 19-38.

Shen, W., Maturana, F. and Norrie D. H. (1998). Learning in Agent-Based Manufacturing Systems. *Proceedings of AI and Manufacturing Research Planning Workshop*, Albuquerque, NM, AAAI Press, 177-183.

Shultz, C. R. (1989). An expediting heuristic for the shortest processing time dispatching rule. *International Journal of Production Research*, 27, 31-41.

Sikora, R. and Shaw, M.J. (1997). Coordination mechanisms for multi-agent manufacturing systems: applications to integrated manufacturing scheduling. *IEEE Transactions on Engineering Management*, 44**,** 175-187.

Sikora, R. and Shaw, M.J. (1998). A multi-agent framework for the coordination and integration of information systems. *Management Science*, 44**,** 65-78.

Subbu, R. and Sanderson, A. C. (2004). Network-based distributed planning using coevolutionary agents: architecture and evaluation. *IEEE Transactions on Systems, Man and Cybernetics*, *Part A,* 34, 257-269.

Subramaniam, V., Lee, G. K., Hong, Y. S., Wong, Y. S. and Ramesh, T. (2000). Dynamic selection of dispatching rules for job shop scheduling. *Production Planning & Control*, 11, 73-81.

Sun, D. and Lin, L. (1994). A dynamic job shop scheduling framework: a backward approach. *International Journal of Production Research*, 32, 967-985.

Tagawa, S. (1996). A new concept of job shop scheduling system — hierarchical decision model. *International Journal of Production Economics*, 44, 17-26.

Tardif, V. and Spearman, M. L. (1997). Diagnostic scheduling in a finite capacity environment. *Computers and Industrial Engineering*, 32, 867–878.

Thomalla, C. S. (2001). Job shop scheduling with alternative process plans. *International Journal of Production Economics*, 74, 125-134.

Usher, J. M. (2003). Negotiation-based routing in job shops via collaborative agents. *Journal of Intelligent Manufacturing*, 14, 485-499.

Ventura, J. A. and Radhakrishnan, S. (2003). Single machine scheduling with symmetric earliness and tardiness penalties. *European Journal of Operational Research*, 144, 598-612.

Vepsalainen, A. P. J. and Morton, T. E. (1987). Priority rules for job shops with weighted tardiness costs. *Management Science*, 33, 1035-1047.

Vepsalainen, A. P. J. and Morton, T. E. (1988). Improving local priority rules with global lead-time estimates: A simulation study. *Journal of Manufacturing and Operations Management*, 1, 102-118.

Vig, M. M. and Dooley, K. J. (1991). Dynamic rules for due date assignment. *International Journal of Production Research***,** 29, 1361–1377.

Vig, M. M. and Dooley, K. J. (1993). Mixing static and dynamic estimates for due date assignment. *Journal of Operations Management***,** 11**,** 67–79.

Wang, D., Fang, S.-C. and Hodgson,T. J. **(**1998). A fuzzy due-date bargainer for the make-to-order manufacturing systems. *IEEE Transactions on Systems, Man and Cybernetics*, *Part C*, 28, 492-497.

Wang, D., Fang, S.-C. and Nuttle, H. L. W. (1999). Soft computing for multicustomer due-date bargaining. *IEEE Transactions on Systems, Man and Cybernetics*, *Part C*, 29, 566-575.

Weeks, J.K. (1979). A simulation study of predictable due dates. *Management Science***,** 25, 363–373.

Wein, L. M. and Chevalier P. B. (1992). A broader view of the job-shop scheduling problem. *Management Science*, 38, 1018-1033.

Weintraub, A. J., Cormier, D., Hodgson, T. J., King, R. E., Wilson, J. and Zozom, Jr. A. (1999). Scheduling with alternatives: a link between process planning and scheduling. *IIE Transactions*, 31, 1093-1102.

Weiss, G. and Sen, S. (1995). Adaptation and Learning in Multi-Agent Systems. *Lecture Notes in Artificial Intelligence*, 1042, Springer-Verlag.

Wu*,* S. H., Fuh*,* J. Y. H. and Nee, A. Y. C. (2002). Concurrent process planning and scheduling in distributed virtual manufacturing. *IIE Transactions*, 34, 77-89.

Wu, Z. and Li, L. (2003). Case-based reasoning for breast cancer prognosis: a web-based multiagent scheme. *Proceedings of the World Congress on Medical Physics and Biomedical Engineering*. Sydney, Australia.

Wu, Z., Weng, M. and Ren, H. (2003). A fuzzy analytic hierarchy process based scheduling and control system. *Proceedings of the 13th International Conference on Flexible Automation & Intelligent Manufacturing*. Tampa, Florida, 62-68.

Wu, Z. and Weng, M. (2005). Multiagent scheduling method with earliness and tardiness objectives in flexible job shops. *IEEE Transactions on Systems, Man and Cybernetics*, *Part B*, 35, 293-301.

Wu, Z. and Weng, M. (2005). Dynamic due date setting and shop scheduling for make-to-order companies. *Proceedings of the 2005 Industrial Engineering Research Conference***,** Atlanta, Georgia.

Wu, Z. and Weng, M. (2005). Multiagent-based workload control for make-to-order manufacturing. *International Journal of Production Research*, submitted.

Yen, B. P.-C. and Wu, O. Q. (2004). Internet scheduling environment with market-driven agents. *IEEE Transactions on Systems, Man and Cybernetics*, *Part A,* 34, 281-289.

Yoon, S. H. and Ventura, J. A. (2002). Minimizing the mean weighted absolute deviation from due dates in lot-streaming flow shop scheduling. *Computers and Operations Research*, 29, 1301-1315.

Zapfel, G. and Missbauer, H. (1993). New concepts for production planning and control. *European Journal of Operational Research*, 67, 297-320.

Zhao, X., Nakashima, K. and Zhang, Z. G. (2002). Allocating kanbans for a production system in a general configuration with a new control strategy. *IEEE Transactions on Systems, Man and Cybernetics*, *Part A*, 32, 446-452.

Zhu Z. and Heady, R. B. (2000). Minimizing the sum of earliness/tardiness in multi-machine scheduling: a mixed integer programming approach. *Computers and Industrial Engineering*, 38, 297-305.

Zhu, Z., and Meredith, P. H. (1995). Defining critical elements in JIT implementation: a survey. *Industrial Management and Data Systems*, 95, 21-28.

Zozom, Jr. A., Hodgson, T. J., King, R. E., Weintraub, A. J. and Cormier, D. (2003). Integrated job release and shop-floor scheduling to minimize WIP and meet due-dates. *International Journal of Production Research*, 41, 31-45.

About the Author

Zuobao Wu received the Bachelor's and Master's degrees from Nanjing University of Science and Technology, Nanjing, China, in 1985 and 1987, respectively, and the Ph.D. degree in mechanical engineering from Zhejiang University, Hangzhou, China, in 1995. From 1995 to 2000, he was with the National CIMS Engineering Research Center, Tsinghua University, Beijing, China, as an Associate Professor. From 2000 to 2001, he was with the Department of Industrial Engineering at Texas Tech University, Lubbock, TX, as a Research Associate.

He has published over 40 refereed journal and conference papers in production planning and control, scheduling, multi-agent systems, concurrent engineering and computer integrated manufacturing. He is also the author of one book and two book chapters. He is a member of IIE.