

國立交通大學  
工業工程與管理學系

博士論文

多工單等級之平行機台排程問題研究  
— 以IC封裝廠為例



The Study on Parallel Machine Scheduling Problem  
with Consideration of Multiple-Priority Jobs  
and Its Application on IC Assembly Scheduling

研究生：賴春美  
指導教授：鍾淑馨 博士  
彭文理 博士

中華民國九十六年四月

多工單等級之平行機台排程問題研究－以IC封裝廠為例

The Study on Parallel Machine Scheduling Problem with  
Consideration of Multiple-Priority Jobs  
and Its Application on IC Assembly Scheduling

研究生：賴春美

Student：Chun-Mei Lai

指導教授：鍾淑馨 博士

Advisor：Dr. Shu-Hsing Chung

彭文理 博士

Dr. Wen Lea Pearn



Submitted to Department of Industrial Engineering and Management

College of Management

National Chiao Tung University

in Partial Fulfillment of the Requirements

for the Degree of

Doctor of Philosophy

in

Industrial Engineering and Management

April 2007

Hsinchu, Taiwan, Republic of China

中華民國九十六年四月

多工單等級之平行機台排程問題研究－以 IC 封裝廠為例

The Study on Parallel Machine Scheduling Problem  
with Consideration of Multiple-priority Jobs  
and Its Application on IC Assembly Scheduling

學生：賴春美

指導教授：鍾淑馨博士

彭文理博士

國立交通大學工業工程與管理系

摘 要

在競爭市場中，晶圓製造在追求利潤時，必須充分考慮產能有效率的被利用。本文主要研究 IC 封裝排程問題，以最小化總工作負荷為衡量績效。IC 封裝製程的生產特色為產品多樣少量且生產時間短。本文探討 IC 封裝實務因素之平行機台排程問題。此排程問題具有多工單等級、工作產品群組、工作產品別相關之處理時間、順序相關之設置時間及產能等限制，其求解比典型的平行機台排程問題更加困難。本文提出目標函數為最小化總工作負荷之 IC 封裝製程排程問題的整數規劃模式。本文利用數學規劃軟體 CPLEX，結合有效的運算策略，則在可接受的時間內，可利用此模式求得實務問題的可行解。此外，本文也提出一啟發式演算法，並測試啟發式演算法之求解品質及其在實務上問題的應用。

由於晶圓製造之製程複雜、週期時間長、再迴流生產及批次機台等生產特性，使得生產週期時間估算困難。IC 封裝須待晶圓之實貨到臨才能加工，封裝之到貨預估模式即為晶圓交期指派模式，因此本文先針對晶圓製造提出交期指派模式，利用此模式可在產品組合隨時間改變之環境下求得符合目標達交率之交期，以作為 IC 封裝排程之依據。

**關鍵詞：** 生產規劃、產品組合、交期指派、平行機台排程問題、整數規劃。

# The Study on Parallel Machine Scheduling Problem with Consideration of Multiple-priority Jobs and Its Application on IC Assembly Scheduling

Student : Chun-Mei Lai

Advisor : Dr. Shu-Hsing Chung

Dr. Wen Lea Pearn

## ABSTRACT

In order to increase a company's competition edge and profitability, an Integrated-Circuit (IC) manufacturer needs to utilize its existed capacity efficiently. This dissertation studies the IC assembly scheduling problem (ICASP) with the objective of minimizing the total machine workload. The IC assembly scheduling problem (ICASP) is a practical generalization of the classical parallel-machine scheduling problem. Since the ICASP involves constraints on precedence, job clusters, job-cluster dependent processing time, machine capacity, and sequence dependent setup times, it is more difficult to solve than the classical parallel machine scheduling problem. In this dissertation, we formulate the ICASP as an integer programming problem with minimizing the total machine workload to simultaneously assign jobs to machines and sequence the jobs on each machine. By using the powerful CPLEX with effective implementation strategies, the feasible solutions of the real-world ICASP problem can be obtained within reasonable amount of time. An effective and efficient algorithm is also proposed for solving large scale problems.

Wafer fabrication determines to a large extent the production plan of the whole semiconductor manufacturing due to its high complexity and long manufacturing process time. The accuracy of due-date assignment for wafer fabrication strongly influences the efficiency of the scheduling of downstream (back-end) operations. In this dissertation, we also proposed a due-date assignment model for wafer fabrication where the product mix periodically changes.

**Keywords** : production planning, product mix, due date assignment, parallel-machine scheduling, integer programming.

## ACKNOWLEDGEMENTS

I am deeply indebted to my advisors, Professor Shu-Hsing Chung and Professor Wen Lea Pearn, for their valuable assistance, guidance, and encouragement in bringing this research work to a successful completion. Their guidance was unfailingly constructive and I remain in awe of their collective academic excellence and professionalism of which I was a most fortunate and grateful beneficiary.

I would like to extend special thanks to Professor Pei-Chann Chang, Professor Jeng-Fung Chen, and professor Tai-Hsi Wu for serving as my committee members. I am grateful for their helpful comments and suggestions.

I would like to thank all classmates and friends whom I have acquired throughout my years at NCTU – Amy Hsin-I Lee, He-Yau Kang, Yi-Shu Yang, Hsin-Yi Ma, Huei-Chun Wang, Shih-Chou Hsu, Tzu-Yun Chang, Yu-Ting Tai – to name a few. Special thanks are due to Hui-Yun Huang for many helpful discussions on practical applications. I am also grateful to the faculty and the staff in the Department of Industrial Engineering and Management and the team of production management laboratory.

I would also like to pay my tribute to all my colleagues of the Department of Industrial and Business Management of Far East University, where I have worked for more than a decade, for their perpetual support.

Most of all, the deepest appreciation is given to my family. Their unconditional love and constant support over the years is something that I cannot thank them enough for. I thank them for having faith in me and in my capabilities, and for offering advice to me over all these years.

# Table of Contents

摘 要 .....	i
ABSTRACT .....	ii
ACKNOWLEDGEMENTS .....	iii
<b>Table of Contents</b> .....	iv
<b>List of Figures</b> .....	vi
<b>List of Tables</b> .....	vii
<b>Notation</b> .....	ix
<b>1. Introduction</b> .....	1
1.1. Motivation.....	1
1.2. Research Scope and Objectives.....	3
1.3. Organization of the Dissertation.....	4
<b>2. Literature Review</b> .....	5
2.1. Due-Date Assignment for Wafer Fabrication .....	5
2.2. IC Assembly Scheduling Problem .....	12
<b>3. Due-date Assignment for Wafer Fabrication</b> .....	17
3.1. The Wafer Fabrication Process.....	17
3.2. Production System Environment and System Input .....	21
3.3. Due-date Assignment for Single Product Mix .....	22
3.4. Due-date Assignment for Periodical Product Mix Changes .....	25
3.5. Simulation Verifications .....	30
<b>4. Scheduling IC Assembly Operations</b> .....	39
4.1. The IC Assembly Process.....	39
4.2. The IC Assembly Scheduling Problem (ICASP).....	42
4.3. An Integer Programming Formulation for ICASP .....	45
4.4. An Illustrative Example .....	50
4.5. A heuristic algorithm .....	54

4.6. Computational test .....	57
<b>5. Conclusions and Future Research</b> .....	<b>67</b>
5.1. Conclusions.....	67
5.2. Future Research.....	68
<b>References</b> .....	<b>70</b>
<b>Appendix</b> .....	<b>77</b>
Appendix A. Product Process Data.....	77
Appendix B. Workstation Data .....	79



## List of Figures

Figure 1-1. The four stages of the IC manufacturing. ....	1
Figure 3-1. Model of Typical Wafer Flow [59].....	18
Figure 3-2. Formation of cycle time for lots. ....	20
Figure 3-3. Due-date assignment based on target on-time delivery rate $\delta$ .....	24
Figure 3-4. The example of contamination model of three gamma distributions with different combinations of $p_1$ , $p_2$ , and $p_3$ .....	27
Figure 3-5. Fitted contamination model versus histogram of the collected data for experiment 1.....	35
Figure 3-6. Fitted contamination model versus histogram of the collected data for experiment 2.....	36
Figure 3-7. Fitted contamination model versus histogram of the collected data for experiment 3.....	37
Figure 4-1. The process flows of plastic packaging products. ....	40
Figure 4-2. Die bonding on the LOC die-bonder. ....	44
Figure 4-3. The ICASP example with two parallel machines, two priority levels, and three job clusters. ....	51
Figure 4-4. The optimal solution on the ICASP example.....	52
Figure 4-5. The optimal solution on the ICASP example with initial status of $r_{00}$ and $r_{30}$ . ....	54
Figure 4-6. The schedule for the real-world ICASP application example.....	65



## List of Tables

Table 2-1. Summary of related literature on traditional analytical methods for the due-date assignment problem.....	7
Table 2-2. Summary of related literature on traditional simulation methods for the due-date assignment problem.....	8
Table 2-3. Summary of related literature on soft computing approach for due-date assignment problem.....	11
Table 2-4. Summary of related literature on identical PMS problems .....	14
Table 2-5. ICASP related papers.....	16
Table 3-1. Estimated parameters for WT distributions.....	25
Table 3-2. Order Information .....	25
Table 3-3. The simulation inputs for single product mix .....	31
Table 3-4. Average and variance of WT collected from simulation for single product mix .....	32
Table 3-5. Estimated parameters for fitted WT distributions for single product mix.....	32
Table 3-6. Comparison of fitted gamma distribution and collected data.....	33
Table 3-7. Product mix composition for the experiments.....	34
Table 3-8. Performance summary for the experiments.....	38
Table 4-1. Setup times required for switching one product type to another for $R_1$ , $R_2$ , and $R_3$ .....	51
Table 4-2. Job processing times for $R_1$ , $R_2$ , and $R_3$ on the machines $m_1$ and $m_2$ .....	51
Table 4-3. The integer programming solution (optimal) for the ICASP example by CPLEX 8.0 .....	53
Table 4-4. Data Set .....	57

Table 4-5. Summary Results.....	58
Table 4- 6 Summary Results for larger-size problem.....	60
Table 4-7. The product types, processing time, and job priority code in the real-world example .....	62
Table 4-8. Setup times required for switching one product type to another in the real-world example .....	64
Table A-1. Sequence and Processing Time for Product A.....	77
Table B-1. Relevant data for each workstation .....	79



## Notation

### Due-Date Assignment for Wafer Fabrication:

$d$  : index of order;

$D_d$  : the due date of order  $d$  ;

$PT_d$  : raw process time of order  $d$  ;

$r_d$  : the release date of the latest batch of order  $d$  ;

$s^2$  : sample variance;

$WTC_d^{-1}(\delta)$  : the inverse of the cumulative function of the fitted contamination distribution of order  $d$  ;

$WTG_d^{-1}(\delta)$  : the inverse of the cumulative function of the fitted gamma distribution of order  $d$  ;

$\bar{x}$  : sample average;

$\alpha$  : shape parameter of gamma distribution;

$\beta$  : scale parameter of gamma distribution;

$\mu$  : population mean;

$\sigma^2$  : population variance;

$\Gamma(\alpha)$  : gamma function,  $\Gamma(\alpha) = \int_0^{\infty} t^{\alpha-1} e^{-t} dt$  .

### IC Assembly Scheduling Problem:

$i$  : product type index,  $i = 0, 1, \dots, I$  ;

$j$  : index of job for product type  $i$  ,  $j = 0, 1, \dots, J_i$  ;

$k$  : machine index,  $k = 1, 2, \dots, K$  ;

$A$  : the set of job priority code;

$G_k$  : the total number of jobs in the schedule  $PS_k$  ;

$h_{ij}$  : the priority code associated with job  $r_{ij}$  ;

- $I$  : the number of job clusters in job set  $R$ ;
- $J_i$  : the number of jobs in job cluster  $R_i$ ;
- $K$  : the number of identical machines;
- $M$  : the set of machine containing identical parallel machines;
- $m_k$  : the  $k^{th}$  machine;
- $n_{ij}$  : lot size of job  $r_{ij}$ ;
- $p_{ik}$  : the unit processing time for job  $r_{ij}$  in  $R_i$  on machine  $m_k$ ;
- $PS_k$  : partial schedule of machine  $m_k$ ;
- $R$  : the set of jobs to be processed;
- $R_i$  : job cluster containing  $J_i$  jobs of product type  $i$  to be processed;
- $r_{ij}$  : the job to be processed in cluster  $R_i$ ;
- $s_{ii'}$  : the sequence dependent setup time between any two consecutive jobs from different job clusters;
- $u_g$  : the job be scheduled at position  $g$  on machine  $m_k$ ,
- $W_k$  : the predetermined machine capacity expressed in terms of processing time unit;
- $x_{ijk}$  : the variable indicating whether a specific job is scheduled on a machine, with  $x_{ijk}=1$  if job  $r_{ij}$  is scheduled on machine  $m_k$ , and  $x_{ijk}=0$  otherwise;
- $y_{ij'j'k}$  : the precedence variable defined on two jobs  $r_{ij}$  and  $r_{i'j'}$  scheduled on machine  $m_k$ , with  $y_{ij'j'k}=1$  if job  $r_{ij}$  precede job  $r_{i'j'}$  (not necessarily directly), and  $y_{ij'j'k}=0$  otherwise;
- $z_{ij'j'k}$  : the direct-precedence variable defined on two jobs  $r_{ij}$  and  $r_{i'j'}$  scheduled on machine  $m_k$ , with  $z_{ij'j'k}=1$  if job  $r_{ij}$  direct precede job  $r_{i'j'}$ , and  $z_{ij'j'k}=0$  otherwise.

# 1. Introduction

## 1.1. Motivation

Semiconductor companies must maintain high-level customer service to gain their competitive edge. Facing the environment with volatile demand, how to deliver order on time justifies the efficiency of the production planning and scheduling in semiconductor manufacturing. Meanwhile, increasing throughput and minimizing setup times are among other managerial and strategic goals. Finding practical scheduling methods that effectively include these sometimes conflicting objectives is a great challenge.

Integrated circuit (IC) is the major product of semiconductor industry. Its process is very complicated and can be divided into four basic manufacturing steps: wafer fabrication, wafer probe, IC assembly, and final test. The four stages of the IC manufacturing are shown in Figure 1-1 [59], [81]. Wafer fabrication and wafer probe are referred as the “front-end”, while IC assembly and final testing are referred as the “back-end”. In the front-end, silicon wafer are chemically processed, and then tested to generate a supply of electronic devices. In the back-end, the wafers are sawed into ICs, and the IC are packaged, branded, and tested.

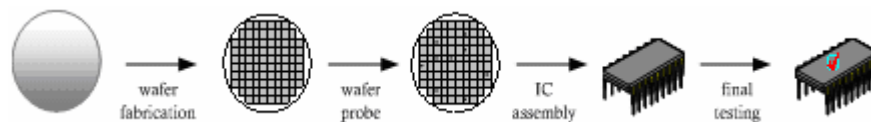


Figure 1-1. The four stages of the IC manufacturing.

Due to different product profit rates and the varied importance level of customers, there often exists more than one priority level of orders in most semiconductor

companies. The multiple-priority job constraint should be included when developing the scheduling methods. On the other hand, the more sophisticated devices being developed are leading to more complex assembly machinery, which is increasing the capital intensity of the IC assembly operations [73]. In order to increase a company's competition edge and profitability, an Integrated-Circuit (IC) manufacturer needs to utilize its existed capacity efficiently and effectively. Therefore, developing efficient scheduling methods simultaneously considering multiple-processing priorities and minimizing the total bottleneck workload is essential.

For the IC assembly scheduling problem (ICASP) investigated in this paper, the jobs are assigned processing priorities and are clustered by their product families with each family containing several product types, which must be processed on a group of identical parallel machines. Further, the job processing time may vary, depending on the product type (job cluster) of the job process on. Setup times for two consecutive jobs of different product types (job clusters) on the same machine are sequence dependent. Since the IC assembly scheduling problem involves constraints on multiple job priorities, job cluster, job-cluster dependent processing time, machine capacity, and sequentially dependent setup times, it is more difficult to solve than the classical parallel-machine scheduling problem.

Wafer fabrication determines to a large extend the production plan of the whole semiconductor manufacturing due to its high complexity and long manufacturing process time. In order to quickly respond to customers' fluctuating demand, companies make changes on the product mix periodically. Because of the complexity of the wafer fabrication process, the due-date assignment problem in semiconductor companies is more difficult to solve compared to other manufacturing industries. Since the accuracy of due-date assignment for wafer fabrication strongly influences the efficiency of the

scheduling of downstream operations, a due-date assignment model for wafer fabrication would be required.

## **1.2. Research Scope and Objectives**

Semiconductor companies can be successful if they only focus on either of the two types: mass manufacturing with high volume and low cost, or high level of product mix that is flexible [81]. High-volume, low-cost fabs produce a few kinds of products, such as memory products, in large quantity in order to have economies of scale; while high-mix flexible fabs mainly produce Application-Specific Integrated Circuit (ASIC) and aim to leverage economies of scope. Hence, the manufacturing strategies and performance measurements of these two types would be significantly different.

This dissertation focuses on the scheduling problems for the manufacturers mainly producing memory products. The purpose of this dissertation is to develop methods for solving two problems that are crucial for efficient scheduling in IC assembly scheduling problem. The objectives of this dissertation include the following:

1. Due-date assignment for the wafer fabrication.

To present a due date assignment model consistent with the target on-time-delivery rate for the environment where product mix changes periodically.

2. Scheduling model for the IC assembly operations.

To design a scheduling model for ICASP with minimizing the total machine workload to simultaneously assign job to machines and to determine the processing sequence on each machine with considerations of the multiple job-priority constraint, and the processing time and the setup time in the capacity constraints.

### 1.3. Organization of the Dissertation

This dissertation includes five chapters that cover the conceptual bases of the study, due-date assignment for wafer fabrication, scheduling of IC assembly operations, and the conclusions. This dissertation is organized as follows.

Chapter 1 provides the background of the research, defines the research domain, problem, and objectives.

Chapter 2 reviews past research work done in the areas related to the study of due-date assignment for wafer fabrication and IC assembly scheduling problems.

Chapter 3 considers the due-date assignment for wafer fabrication and presents a due-date assignment model to set manufacturing due date satisfying the target on-time-delivery rate. Cycle times are first analyzed for each product type under single product mix. A due-date assignment model for periodical product mix changes is then presented by taking the merit of contamination model.

Chapter 4 considers the IC assembly scheduling problem. We first describe the IC assembly process in detail, and capture the characteristics of the ICASP. An integer programming formulation is then proposed to solve the ICASP with minimizing the total machine workload to simultaneously assign jobs to machines and sequence the jobs on each machine. An efficient heuristic is also proposed to obtain the near-optimal solution for large scale problems.

Chapter 5 gives a summary of this research. Conclusions will be drawn based on the results of the research.



## 2. Literature Review

The topic of scheduling for Integrated-Circuit Assembly operations (ICASP) draws upon ideas from two different research areas. As such, this literature review is divided into two areas which contribute to this dissertation. The first subject area is to review the due-date assignment for wafer fabrication. The second subject area is to review on the topics of IC assembly scheduling problem.

### 2.1. Due-Date Assignment for Wafer Fabrication

Due-date assignment has always been an important research topic in production planning and control system, which has attracted abundant research interest. Surveys on recent results of specific aspects of due-date assignment problems, such as Cheng and Gupta [11] and Gordon et al. [28] [29], confirm this continued interest. Conway et al. [21] presented four due-date assignment rules to determine the allowances for cycle time (the difference between due-dates and arrival time) for each job in the following ways: Total-work due-dates (TWK) rule estimates the allowance for cycle-time as a proportion of the expected total processing time of a job. Number-of-operation due-dates (NOP) rule assumes the allowance for cycle-time is proportional to the number of operations. Constant-allowance due-dates (CON) rule assigns a constant cycle time allowance to all jobs. The random-allowance due-dates (RDM) rule randomly assigns allowance within a given range.

The traditional methods of due-date assignment used in the related literature can be classified into two categories: analytical approaches and simulation approaches. The analytical approach offers an exact way that determines mean and variance of flow time estimates and further set due dates. Seidmann and Smith [64] studied the constant

due-date (CON) assignment policy with the objective of minimizing the expected aggregate cost per job subject to restrictive assumptions on the priority discipline and the penalty functions. Cheng [10] proposed a method to assign optimal total work content (TWK) due-dates. Enns [24] used dynamic cycle-time forecasting to set due dates with the objective of minimizing related costs of job shop scheduling. Li and Cheng [42] analyzed the single machine due-date determination and resequencing problem with the objective of minimizing the maximum weighted tardiness and the cost of due-date assignment. Hopp and Roofsturgis [32] developed a due-date quoting method to achieve a target service level by determining lead times as a function of work in process and using a control chart method for adjusting the parameters in this function overtime. Ooijen and Bertrand [50] proposed a method to set the optimal due dates by considering work load, lead time-related and tardiness-related costs. The other trend in analytical approach is to set due dates by determining cycle time prediction errors and distribution functions [25], [37], [39]. Table 2-1 displays the summary of related literature on traditional analytical methods for the due-date assignment problem.

For the simulation approaches, researchers examined the relative performance of various due-date assignment rules, dispatching rules, or sequencing procedures [1], [5], [79]. Other studies of simulation approach are to develop effective cycle-time estimation and due-date assignment policies based on the simulation studies. Weeks [80] proposed a method to assign due date based on expected job cycle time and shop congestion information, and concluded that such due dates were more attainable. Vig and Dooley [75] proposed two new cycle-time estimation methods. They also evaluated relationships between several shop factors and effects on due-date performance via a simulation study. Vig and Dooley [76] further incorporated steady-state with dynamic cycle time estimates to develop cycle-time estimation and provided a regression-based approach for setting job

Table 2-1. Summary of related literature on traditional analytical methods for the due-date assignment problem

Literature	Objective	Major characteristics considered	Solution approach
Seidmann and Smith [64]	Min. the expected aggregate cost per job	due-date cost, tardiness cost, earliness cost.	linear cost model and algorithm
Cheng [10]	Min. the average amount of missed due-dates	dynamic job shop with assembly operations.	queueing networks and Laplace transformation
Enns [24]	Min. job shop scheduling related costs	earliness, tardiness, lead time penalty.	dynamic flow-time forecasting model
Li and Cheng [42]	Min. the maximum weighted tardiness penalty and the due-date assignment cost.	due-dates of the old jobs are treated as given parameters and those of the new jobs are decision variables	algorithm
Hopp and Roofsturgis [32]	Achieving a target service level	leadtimes as a function of WIP	A control chart method
Ooijen and Bertrand [50]	the optimal lead-time related costs per order	workload, lead-time related and tardiness related costs	work-load dependent flow time distribution function
Enns [25]	due date setting rule selection	delivery performance	dynamic flow-time forecasting model
Kaplan and Unal [37]	Best tardiness probability of the job	WIP inventory and tardiness cost	flow time prediction based on correlation analysis
Lawrence [39]	cost minimization, attainment of service level targets, and minimization of mean absolute lateness and mean squared lateness	relevant costs	modeling flowtime estimation as a forecasting problem

Table 2-2. Summary of related literature on traditional simulation methods for the due-date assignment problem

Literature	Performance examined	Major characteristics considered	Solution approach
Ahmed and Fisher [1]	total cost	due date assignment, order release, and sequencing interaction	simulation model
Chang [5]	job completion time	dispatching rule and shop utilization rate	simulation model
Weeks and Fryer [79]	job flow-time cost, job lateness cost, job earliness cost, due date cost, and labor transfer cost	dispatching rule, labor assignment, and due date assignment rules	simulation model, regression model
Weeks [80]	job lateness cost, job earliness cost, and due date cost	dispatching rule, due date assignment rules and shop size and structure	simulation model
Vig and Dooley [75]	estimate job flow time	Shop congestion condition	regression analysis
Vig and Dooley [76]	job lateness	Shop congestion condition, and dispatching heuristic	flow time estimation, regression-based approach
Raghu and Rajendran [60]	job lateness		SA algorithm and regression analysis

shop due dates. Raghu and Rajendran [60] developed a due-date setting policy for a real-life job shop by incorporating the best performing dispatching rule which is selected by simulation. Roman and del Valle [62] presented a rule for the due-date assignment problem of reducing the tardiness and percentage of delayed jobs through a combination of the dispatch rule and assignation of due date. Chang [6] showed that statistical analysis of a simulation model could give valuable insights into the cycle time behavior of jobs through workstations and proposed an approach to provide real-time estimates of

the queueing time for the remaining operations of the jobs. Table 2-2 displays the summary of related literature on traditional simulation methods for the due-date assignment problem.

Owing to the complexity of wafer manufacturing process, the due-date assignment problem in semiconductor companies is more difficult to solve than the classical cycle-time estimation problem. A product mix that varies periodically is an even more complicated problem compared to other manufacturing industries. Chung et al. [19] presented a due date assignment model by using the simulation method and queueing theory. They also proposed a methodology of determining related parameter for cycle time control. Chung and Huang [18], with the application of queueing theory and the observation of the characteristics of material flow, developed a production cycle-time estimation formulation, the Block-Based Cycle Time (BBCT) estimation algorithm. The BBCT algorithm has distinguishable performance in estimating mean cycle time where the product mix is fixed during all the time periods. However, the BBCT model does not consider the product mix periodically changes, which might not be applicable in our planning environment.

Recently, soft computing techniques have been widely applied in the studies of due-date assignment for semiconductor manufacturing. Chang and Hsieh [7] identified influential variables related to the cycle time through regression analysis by using simulation data. A backpropagation neural network model is then established to forecast the due date of each order. Chiu et al. [15] proposed a case based reasoning (CBR) approach which utilized the  $k$ -nearest-neighbors concept with dynamic feature weights and non-linear similarity functions. Sha and Liu [66] presented a CBR with tree-indexing approach for numerical value prediction and illustrated using the due date assignment problem. The approach applied the strength of CBR as a prediction tool

and the tree-indexing approach as assistance in indexing and retrieving cases. Chang et al. [8] applied the fuzzy modeling method proposed by Wang and Mendel [77] for generation of fuzzy rules by using simulation data. The fuzzy modeling method is further evolves by a genetic algorithm for due-date setting. Chang and Liao [9] constructs fuzzy rule bases with the aid of a self-organizing map (SOM) and genetic algorithm for cycle time prediction in semiconductor manufacturing factory. Hsu and Sha [34] and Sha and Hsu [65] presented an artificial neural network (ANN) based due date assignment rule for cycle time prediction. They examined the various combinations of order review/release (ORR) and dispatching rules and concluded that ANN-based due date assignment rules have a better sensitivity and variance. By integrating constraint-based reasoning with genetic algorithm, Hsu et al. [33] proposed an approach for cycle time prediction in consideration of the work flow status. This approach provides a chromosome-filtering mechanism before generating and evaluating a chromosome. Table 2-3 displays the Summary of related literature on soft computing approaches for the due-date assignment problem.

In Chapter 3, we consider a more general version of due-date assignment problem for wafer fabrication. We present a due-date assignment model that is consistent with the target on-time-delivery rate where product mix changes periodically. In the proposed model, the computations can be made manually. In the mean time, quick response and satisfactory accuracy are the advantages of the proposed model.

Table 2-3. Summary of related literature on soft computing approach for due-date assignment problem

Literature	Performance measurement	Result comparisons	Solution approach
Chang and Hsieh [7]	The square root of mean square error	due-date assignment rules: TWK, NOP, JIQ	A backpropagation neural network model
Chiu et al. [15]	The square root of mean square error	GA-CBR, BPN, TWK, NOP, JIQ	GA-CBR
Sha and Liu [66]	The square root of mean square error	T-CBR, CBR, BPN, JIQ, TWK	CBR with tree-indexing
Chang et al. [8]	The square root of mean square error	WM, EFR, MLPNN, CBR	The fuzzy modeling method further evolves by a GA
Chang and Liao [9]	The square root of mean square error	WM, WM&GA, SOM&WM, the multi-layer perceptron	Fuzzy rule bases with the aid of a SOM and GA
Hsu and Sha [34]	Various performance measurement	Regression-based, ANN-based, JIQ, TWK	ANN based due date assignment rule
Sha and Hsu [65]	Various performance measurement	ANN-based, SFM, KFM, TWK, JIQ, JIBQ	ANN based due date assignment rule
Hsu et al. [33]	The square root of mean square error	GA, BPN, CBR, TWK, NOP, JIQ	CBR and GA

## 2.2. IC Assembly Scheduling Problem

Parallel machine scheduling (PMS) is to schedule  $n$  jobs processed on  $m$  machines, with optimized objective. Each job needs to be processed on one of the machines during a given time interval. Classical parallel-machine scheduling problems have been categorized into three types based on the job processing time characteristics, including the identical machines, uniform machines, and unrelated machines [12], [38], [48]:

- (1) Identical parallel-machine scheduling problem: each job requires only a single operation, which may be processed on any of the parallel machines. The processing times of each job are independent of the machine which the job processed on.
- (2) Uniform parallel-machine scheduling problem: the job processing times are determined by the efficiencies of the machines. The processing time for a job processed on one machine is the job processing time divided by that machine speed.
- (3) Unrelated parallel-machine scheduling problem: a generalization of the uniform parallel-machine scheduling problem, the efficiency of the machine depends on the type of jobs processed and the processing times of different jobs on the same machine may not be equal. There is no particular relationship between the processing times for the same job being processed on different machines.

Considering the characteristic of the ICASP investigated in this dissertation, we focus on the identical parallel-machine scheduling problem in this literature review.

A mathematical programming formulation is a natural way to solve machine scheduling problems [4] [61]. Mathematical programming formulations and algorithms are the most used methods for solving parallel-machine scheduling problems. A number of papers addressed the machine scheduling involving sequence-dependent setup times



[2]. Tahar et al. [70] presented a linear programming model for solving the identical parallel-machine scheduling problem with job splitting and sequence-dependent setup times. Omar and Teo [49] presented a mix integer programming model for solving identical parallel-machine scheduling problem considering family setup-time constraints. Pearn et al. [53][54][55] presented integer programming model, network transformation, and algorithms for solving the identical parallel-machine scheduling problem with minimizing the total machine workload subject to sequence-dependent setup time and due date restrictions. Lee and Pinedo [41] considered the identical parallel-machine problem with sequence-dependent setup times. Schutten and Leussink [63] presented a branch-and-bound algorithm for solving the identical parallel machine problem with release date, due dates, and family setup times, with the objective of minimizing the maximum lateness of any job. Webster and Azizoglu [78] presented two dynamic programming algorithms for solving the identical parallel machine problem with family setup times to minimize total weighted cycle time. Dunstall and Wirth [23] proposed a branching scheme to the identical parallel machine problem with family setups to minimize the weighted sum of completion times. Chern and Liu [14] constructed five family-based scheduling rules, and built simulation model to examine these five rules.

The survey papers [12], [48] provide a wide range of parallel machine scheduling with precedence constraint. However, the precedence relations studied in these papers can be presented by graphs in tree-types: in-tree type precedence (each job has at most one successor), out-tree type precedence (each job has at most one predecessor), and chain type precedence (each job has at most one predecessor and at most one successor). In the ICASP, the jobs are completely partitioned and a precedence relation defined by the job priority. Therefore, the algorithms for solving scheduling problem with tree-type precedence constraint may not be applied to the ICASP.

Table 2-4. Summary of related literature on identical PMS problems

Literature	Objective	Major characteristics considered	Solution approach
Tahar et al. [70]	Min. maximum completion time	job splitting, sequence-dependent setup times	linear programming
Omar and Teo [49]	Min. the sum of weighted earliness / tardiness cost.	family setup times	mixed integer programming
Pearn et al. [53]	Min. total machine workload	family setup times, due date restriction	integer programming
Pearn et al. [54]	Min. total machine workload	family setup times, due date restriction	Network transformation
Pearn et al. [55]	Min. total machine workload	family setup times, due date restriction	algorithms
Lee and Pinedo [41]	Min. the sum of weighted tardiness	sequence-dependent setup times, due date restriction	heuristics
Schutten and Leussink [63]	Min. the maximum lateness of any job	family setups, due dates, release dates	branch and bound algorithm
Webster and Azizoglu [78]	Min. total weighted flowtime	Family setups	dynamic programming algorithms
Dunstall and Wirth [23]	Min. the weighted sum of completion time	family setups	branching scheme
Cheng and Diamond [13]	Min. total cycle time	two-class priority jobs	dynamic programming algorithm

Cheng and Diamond [13] considered the parallel machine scheduling problem for minimizing total cycle time, and provided a dynamic programming algorithm. Unfortunately, their model is developed only for two-class priority and does not consider the sequentially dependent setup times. Table 2-4 displays the summary of related

literature on identical parallel-machine scheduling problems. In those research works, the constraint of sequence-dependent setup times and multiple job priorities has never been considered simultaneously, therefore, their models may not be applicable to the ICASP.

Most of the studies have focused on the fab complexity and problems, since the wafer fabrication is the most capital intensive and complex part of IC manufacturing process. The recent increase in device complexity has led to the development of complex capital-intensive assembly systems. The developments in the area of planning and scheduling IC assembly operations, therefore, have seized more attention from the academic world than before. In contrast to the front-end processes being highly reentrant, the back-end process follows a more linear, assembly-line type of flow [56], [72]. Most of the production planning and scheduling methods designed for front-end operations are not applicable for the fundamentally different back-end operations. The IC Assembly Scheduling Problem (ICASP) is a variation of the parallel machine scheduling problem. Since the ICASP involves constraints on multiple-priority jobs, job cluster, job-cluster dependent processing time, machine capacity, and sequentially dependent setup times, a good scheduling model for the back-end operations must capture this complexity.

Scheduling IC assembly operations has been the subject of a series of papers. Table 2-5 displays summary of the ICASP related papers. Liu et al. [45] developed a computer-aided scheduling system for the IC packaging industry. Potoradi et al. [58] developed a simulation-based scheduling to maximize demand fulfillment for the assembly facility. Liu et al. [46] developed a lot release methodology for minimizing machine conversion for the back-end manufacturing. Yin et al. [83] developed a rule-based finite capacity daily scheduling system for semiconductor back-end assembly.

Tovia et al. [72] considered a simple version of the ICASP for the high production volume and high production mix environment, and provided a mathematical model for an IC assembly firm. Tovia et al. [72] also presented a rule-based heuristic approach to solve the problem approximately. However, their models do not consider sequence-dependent setup times and multiple job-priorities simultaneously, which may not reflect the real situation accurately.

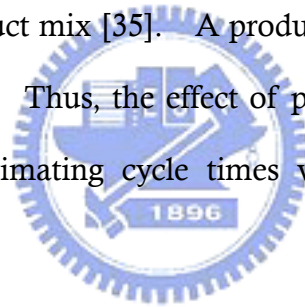
Since in those research works do not consider the sequence-dependent setup times and multiple job-priorities simultaneously, therefore, their models may not applicable to the ICASP. In chapter 4, we formulate the ICASP as an integer programming problem for the complexity of the IC assembly operations. The programming model considers the job priority constraint, the processing time, and the setup times in the capacity constraint.

Table 2-5. ICASP related papers

Research	Objective	Solution approach
Liu et. al. [45]	Reduced scheduling time	STEP enabling technology
Potoradi et al. [58]	Maximizing demand fulfillment	Simulation based scheduling
Yin et al. [83]	Reduction setup times Improving on-time delivery	Rule-based heuristic
Liu et al. [46]	Allocate capacity to lots based on finite capacity constraint	Algorithm
Tovia et al. [72]	Maximizing throughput	Mathematical formulation Two heuristics

### 3. Due-date Assignment for Wafer Fabrication

Semiconductor companies must maintain high-level customer service to gain their competitive edge. In order to quickly respond to customers' fluctuating demand, companies often make changes on the product mix frequently and periodically. In a semiconductor fab, machines are shared by plenty of different products, resulting in a complex queue in the precious resource. The product mix has considerable impact on production throughput, cycle time, and the capability of meeting due dates. Production throughput, cycle time, machine utilization, and work in process (WIP) inventory, are highly interrelated [16], [22]. Under different product mixes, the overall manufacturing performance of the system would be different. The cycle time distribution may shift with the changes in the product mix [35]. A product mix that varies periodically makes the system more complicate. Thus, the effect of product mix changes should be taken into consideration when estimating cycle times where the product mix periodically changes.



In this chapter, we present a due-date assignment model for wafer fabrication. An overview of the wafer fabrication process is first presented. Cycle times are then analyzed for each product type under single product mix. The contamination model is applied to tackle the effect of product mix changes in a periodical fashion. A due-date assignment model is then presented for wafer fabrication where product mix changes periodically.

#### 3.1. The Wafer Fabrication Process

The process of wafer fabrication is a complicated sequence of chemical and physical operations that are performed on a silicon wafer. A designed IC is sent to a wafer fab and requires masking. The basic procedure usually includes about 15-30 repetitive steps

such as the diffusion, photolithography, etch, thin films, ion implant, and polish, to complete the electric circuit on wafers [59], [73]. Through the semiconductor manufacturing process, the completed wafers have a full complement of integrated circuits permanently etched into each silicon wafer, as shown in Figure 3-1. Wafer fabrication determines to a large extent the production plan of the whole semiconductor supply chain due to its high complexity and long manufacturing process time.

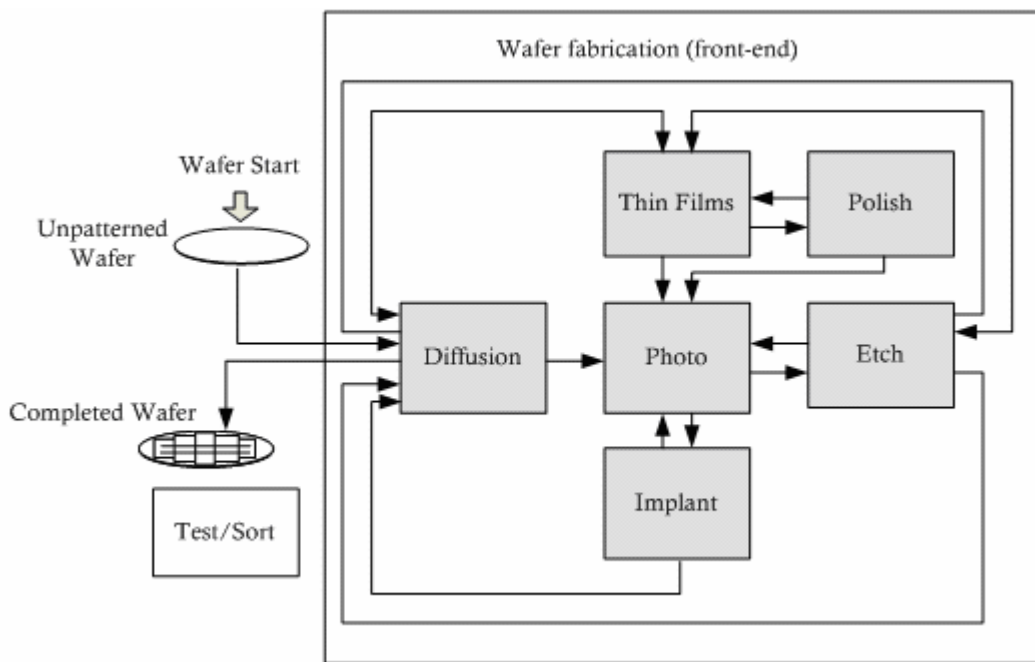


Figure 3-1. Model of Typical Wafer Flow [59].

### 3.1.1 The Characteristics of Wafer Fabrication Process

Typically, the production process has several unique characteristics. First, the process comprises several hundreds of steps on a single wafer. Besides, the manufacturing flow of different products may differ significantly, and the processing time required of the machines for one product may be twice as much as that required for the other products [16]. Second, some of the machines may be used for the same operation more than once as successive circuit layers are added in the production process, and this is termed re-entrant flow property. One problem caused by this property is that different

layers of a wafer have to go through the same machines and to compete with other wafers for the same resources. Finally, based on the number of lots being processed simultaneously, machines are usually categorized into serial or batch types. Batch operations would cause wafer lots additional waiting time owing to batch size transformation. As a result, these interrelated characteristics complicated cycle-time analysis and due-date assignment for the semiconductor fabs.

Furthermore, a product mix that varies periodically makes the system more complicated. The product mix has considerable impact on production throughput, cycle time, and the capability of meeting due dates. Production throughput, cycle time, machine utilization, and work in process (WIP) inventory, are highly interrelated [16], [22]. Under different product mixes, the overall manufacturing performance of the system would be different. The cycle time distribution may shift with the changes in the product mix [35]. Thus, the effect of product mix changes should be taken into consideration when estimating cycle times where the product mix periodically changes.

### ***3.1.2 Cycle Time Analysis for Wafer Fabrication***

Cycle time is the time elapsed from the release of a lot into the plant until its emergence as a finished product [47]. Cycle time for a wafer lot flowing through the entire production process includes raw process time (PT) and waiting time (WT) [18]. PT consists the pure processing time, loading, and unloading times. WT includes the following two parts [18]:

- 1) Load factor waiting time (LFWT):

The LFWT represents the time for a lot waiting for an available workstation.

The load on a workstation reflects the utilization rate and influences the average waiting time of a candidate lot.

2) Batch factor waiting time (BFWT):

The BFWT represents the time for a release batch flowing through the whole process without considering PT and LFWT. BFWT comprises the following two parts:

a) Batch forming waiting time:

The waiting time is caused by gathering lots to form a batch.

b) Batch size transformation waiting time:

The waiting time is caused from transferring lots from an upstream batch workstation to a downstream workstation when the downstream workstation processes a smaller batch size. A temporary peak load thus occurs at the downstream workstation.

The formation of cycle time for lots is depicted in Figure 3-2. PT is a known constant, while WT is the variable that needs to be estimated. Due to the complexity of WT, a simulation-based WT distribution is used to estimate WT in this study.

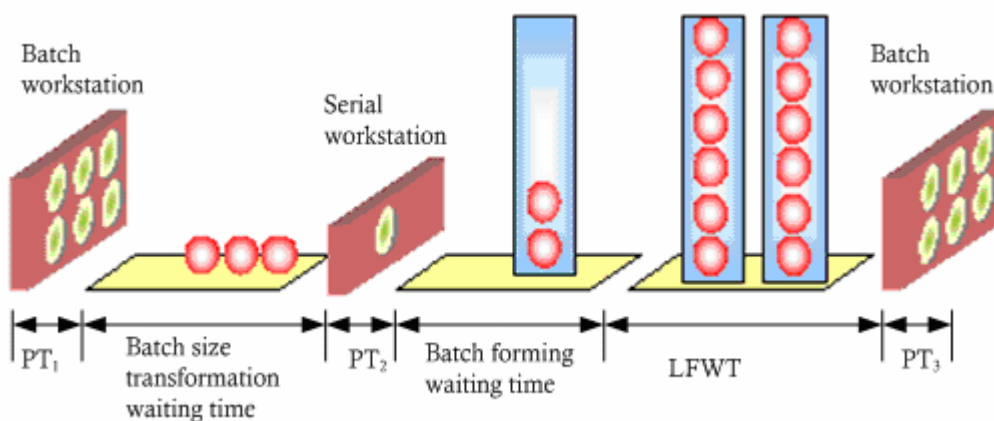


Figure 3-2. Formation of cycle time for lots.



### 3.2. Production System Environment and System Input

A modern fab requires a very high capital investment, usually to a billion dollars or more [31]. Generally, the wafer stepper machines are the most expensive machine in wafer fabrications and are treated as the bottleneck. The tremendous amount of investment makes the manufacturers put emphasis on fully utilizing the bottleneck machine. On the other hand, if the utilization rate of bottleneck machine is set too high, the system may be unstable because of unforeseen disruptions [67]. Therefore, the strategy is to keep the utilization rate of bottleneck in a given range with the consideration of maximizing the utilization of bottleneck while keeping the production system stable.

The batch size of wafer release is set to be six lots. Such a setting could raise the throughput rate of many workstations, which have a maximum batch size of six lots.

Wafer lots are released under a CONWIP (CONstant Work In Process) release policy [69]. By adopting the CONWIP policy, the WIP is kept reasonably constant. As such, the cycle-time distribution should also be reasonably stationary. Based on CONWIP release policy, wafer lots are released into the plant only when WIP level is lower than the planned WIP level,  $L$ . Once the WIP level is lower than  $L$ , six lots (the release batch size) of a product type which has the largest accumulated unreleased quantity is released into the plant. The calculation of “accumulated unreleased quantity” is based on the planned daily release amount. When the product is assigned to release, six lots are deducted from the corresponding unreleased quantity. On the other hand, if there are remaining quantities not released to the plant, the unreleased quantities will be accumulated to the next day.

### 3.3. Due-date Assignment for Single Product Mix

We begin by considering the due-date assignment problem for product mix that is fixed throughout the time periods. As mentioned in Section 3.1.2, owing to the complexity of WT, a simulation-based WT distribution is used to estimate WT. WT of each product type is first modeled by gamma distribution. Due date can then be set based on release date, PT, and WT fitted distribution.

#### 3.3.1. WT Distribution Fitting for Single Product Mix

The gamma distribution is a nonnegative-domain and right skewed probability distribution. The gamma distribution is frequently used as the probability model for waiting times [30]. For instances, in life testing, the waiting time until “death” is the random variable which frequently modeled by a gamma distribution. In addition, the gamma distribution is also a good model for many nonnegative random variables of the continuous type, because the two parameters  $\alpha$  and  $\beta$  provide a great deal of flexibility [30].

A random variable  $x$  is said to have a gamma distribution with parameters  $\alpha > 0$  and  $\beta > 0$ . The probability density function of  $X$  is

$$\phi_X(\alpha, \beta) = \begin{cases} \frac{x^{\alpha-1} e^{-x/\beta}}{\beta^\alpha \Gamma(\alpha)}, & 0 \leq x < \infty \\ 0, & \text{otherwise} \end{cases} \quad (3-1)$$

where  $\Gamma(\alpha)$  is known as gamma function, defined by  $\Gamma(\alpha) = \int_0^\infty t^{\alpha-1} e^{-t} dt$ . In this gamma distribution,  $\mu = E(X) = \alpha\beta$  and  $\sigma^2 = V(X) = \alpha\beta^2$ .

Based on the historical data, WT of each product type is always nonnegative and skewed to the right in the wafer fabrication process, and can be modeled satisfactorily by

the gamma distribution. The method of moments estimators is used for unknown parameters  $\alpha$  and  $\beta$ . The first two moments of the gamma distribution with parameters  $\alpha$  and  $\beta$  are

$$\mu'_1 = \mu = \alpha\beta \quad (3-2)$$

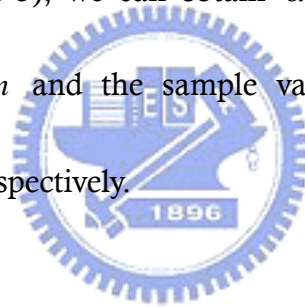
$$\mu'_2 = \sigma^2 + \mu^2 = \alpha\beta^2 + \alpha^2\beta^2 \quad (3-3)$$

Equate these quantities to their corresponding sample moments. Thus,

$$\mu'_1 = \alpha\beta = m'_1 = \bar{x} \quad (3-4)$$

$$\mu'_2 = \alpha\beta^2 + \alpha^2\beta^2 = m'_2 = \frac{1}{n} \sum_{i=1}^n x_i^2 \quad (3-5)$$

From Equations (3-4) and (3-5), we can obtain  $\hat{\alpha} = \bar{x}^2 / S^2$  and  $\hat{\beta} = S^2 / \bar{x}$ , where the sample average  $\bar{x} = \sum_{i=1}^n x_i / n$  and the sample variance  $S^2 = \sum_{i=1}^n (x_i - \bar{x})^2 / n$  are the estimators of  $\mu$  and  $\sigma^2$ , respectively.



### 3.3.2. Due-date Setting

Like firms in other industries, semiconductor companies must meet customers' fluctuating demands in order to survive. Failure to deliver products on time, even with the right quality and quantity, can result in profit penalties or loss of customers. The on-time-delivery rate is an important determinant to measure customer service. The target on-time-delivery rate is therefore chosen as our due-date performance measure. The advantage of this policy is that it combines the competitive advantage of short lead times with the requirement that target numbers of due-date promises can be met [39].

The due date of an order is assigned to the date that equals to release time of the order plus raw process time (PT) and the  $\delta$ -percentile waiting time, where  $\delta$  is the

target fraction of on-time-delivery orders. The  $\delta$ -percentile waiting time can be obtained by taking the inverse of the cumulative function of the fitted gamma distribution. The due date of order  $d$  can then be assigned as

$$D_d = r_d + PT_d + WTG_d^{-1}(\delta), \quad (3-6)$$

where  $D_d$  is the due date of order  $d$ ,  $r_d$  is the release date of the latest batch of order  $d$ ,  $PT_d$  is PT of order  $d$ , and  $WTG_d^{-1}(\delta)$  is the inverse of the cumulative function of the fitted gamma distribution of order  $d$ . Figure 3-3 illustrates the due-date assignment based on the target on-time-delivery rate.

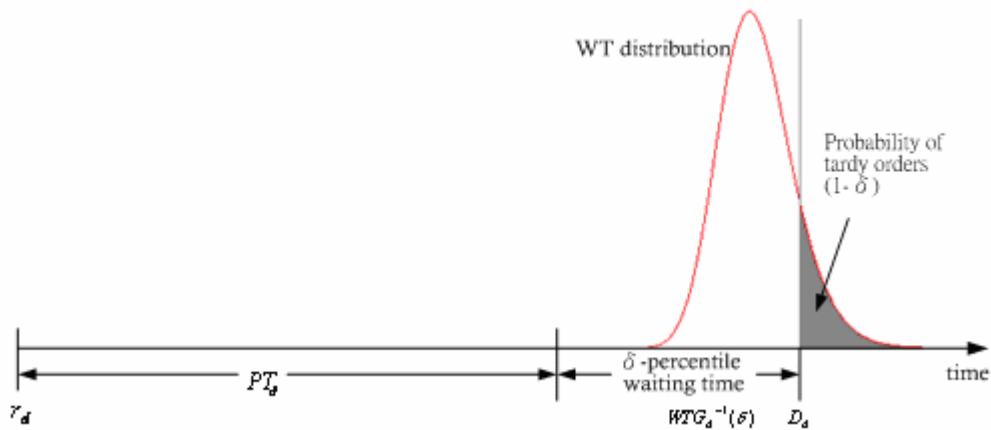


Figure 3-3. Due-date assignment based on target on-time delivery rate  $\delta$ .

Consider the following due-date assignment examples with two product types (L and M) being produced in the plant. PT of these two product types are known as: 120 hours for product L and 145 hours for product M. Table 3-1 displays the estimated parameters for WT fitted distributions under product mix L:M=4:6 and L:M=6:4, respectively. The target on-time-delivery rate is set to 95%.

In the situation where the product mix is L:M=4:6 throughout the planning horizon, due dates need to be assigned to these two orders. Table 3-2 displays the information of

the orders. Since the 95-percentile of  $\text{gamma}(25.0, 2.0)$  is 67.5 and the 95-percentile of  $\text{gamma}(26.0, 2.3)$  is 80.31, based on (4-6), the due dates of order 1 and order 2 (in days) can be obtained as

$$D_1 = 3 + 120/24 + 67.5/24 = 10.8 \text{ (day)} \quad (3-7)$$

$$D_2 = 2 + 145/24 + 80.31/24 = 11.39 \text{ (day)} \quad (3-8)$$

We note that the solution will be different when the product mix is L:M=6:4 throughout the planning horizon. The 95-percentile of  $\text{gamma}(22.0, 1.6)$  is 48.38. The 95-percentile of  $\text{gamma}(28.0, 2.7)$  is 100.53. The due dates of order 1 and order 2 (in days) become 10.02 and 12.23, respectively.

Table 3-1. Estimated parameters for WT distributions

Product mix (L:M)	Product L		Product M	
	$\hat{\alpha}$	$\hat{\beta}$	$\hat{\alpha}$	$\hat{\beta}$
Mix(4:6)	25.0	2.0	26.0	2.3
Mix(6:4)	22.0	1.6	28.0	2.7

Table 3-2. Order Information

Order No.	Product type	Order size (lot)	Planned release date
1	L	6	3
2	M	6	2

### 3.4. Due-date Assignment for Periodical Product Mix Changes

To tackle the effect of periodical changes on product mix, a contamination model is built for estimating WT of each product type. A due-date assignment model is then developed, by which the probability of a job being delivered on-time can be controlled.

### 3.4.1. Contamination Model

The contamination model, a mixture of distributions, provides a rich class of distributions that can be used in modeling data from a population that is composed of several homogeneous subpopulations. The contamination model is useful, particularly for cases with multiple manufacturing processes where the equipment or workmanship are not identical [52], or for cases where there are variable lead time demands in the inventory management function [40]. Such situations often result in production with inconsistent precision in production performances, and the contamination model can be used to characterize the population [40], [51], [52], [57].

Let the observations  $x_1, \dots, x_n$  be a random sample from a contamination model with density function

$$f(x) = \sum_{k=1}^m p_k \phi_X(\theta_k), \quad (3-9)$$

where  $\phi_X(\theta_k)$  is the density of  $X$  in the  $k^{th}$  subpopulation distribution having parameter  $\theta_k$ , and  $p_k$  is the probability of belonging to the  $k^{th}$  subpopulation. Thus,

$$0 \leq p_k \leq 1 \text{ and } \sum_{k=1}^m p_k = 1.$$

Consider the contamination model of three gamma populations, with probability  $p_1$  for population I distributed as gamma( $\alpha_1 = 1, \beta_1 = 1$ ), probability  $p_2$  for population II distributed as gamma( $\alpha_2 = 2, \beta_2 = 1$ ), and probability  $p_3$  for population III distributed as gamma( $\alpha_3 = 3, \beta_3 = 1$ ). The probability density function of the contamination gamma distributions may be expressed as

$$f(x) = p_1[\phi_X(\alpha_1, \beta_1)] + p_2[\phi_X(\alpha_2, \beta_2)] + p_3[\phi_X(\alpha_3, \beta_3)], \quad (3-10)$$

where  $0 \leq p_1 \leq 1$ ,  $0 \leq p_2 \leq 1$ ,  $0 \leq p_3 \leq 1$ ,  $p_1 + p_2 + p_3 = 1$ , and

$$\phi_X(\alpha_k, \beta_k) = \begin{cases} \frac{x^{\alpha_k-1} e^{-x/\beta_k}}{\beta_k^{\alpha_k} \Gamma(\alpha_k)}, & x > 0, k = 1, 2, 3 \\ 0, & \text{otherwise} \end{cases} \quad (3-11)$$

In this contamination model, if  $p_1 = 1$ , then the contamination gamma model reduces to the distribution  $\text{gamma}(\alpha_1, \beta_1)$ . If  $p_2 = 1$ , then the contamination model reduces to the distribution  $\text{gamma}(\alpha_2, \beta_2)$ . On the other hand, if  $p_3 = 1$ , then the contamination model reduces to the distribution  $\text{gamma}(\alpha_3, \beta_3)$ . Figure 3-4 displays various distributions modeled by the contamination of three gamma distributions  $\text{gamma}(1,1)$ ,  $\text{gamma}(2,1)$ , and  $\text{gamma}(3,1)$  with six different combinations of  $p_1$ ,  $p_2$ , and  $p_3$ . We note that the shape of the density differs for the different combinations of  $p_1$ ,  $p_2$ , and  $p_3$ .

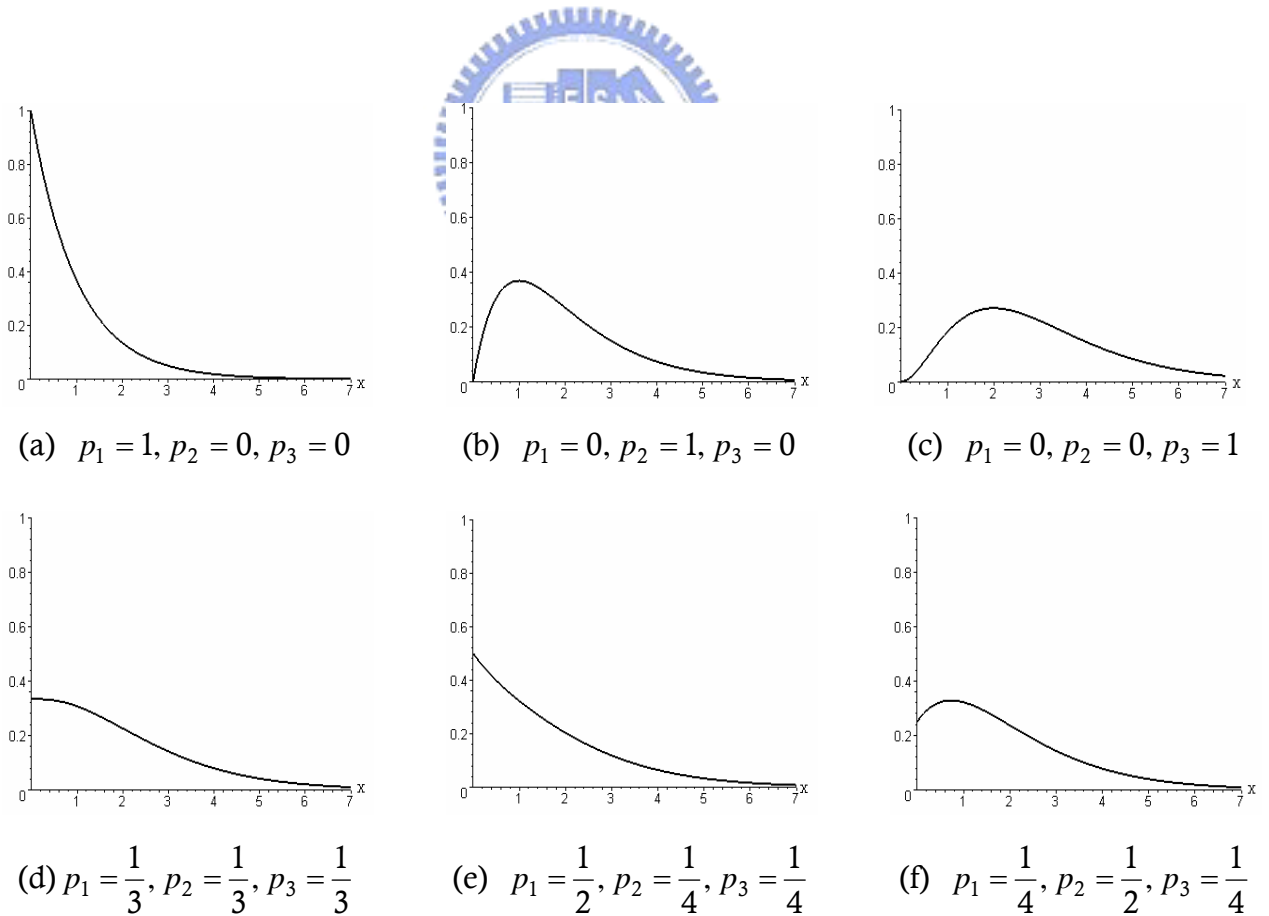


Figure 3-4. The example of contamination model of three gamma distributions with different combinations of  $p_1$ ,  $p_2$ , and  $p_3$ .

### 3.4.2. The Contamination Model for Periodical Product Mix Changes

In wafer fabrication, the lot release time and lot completion time may not belong to the same time period due to the long cycle time. Cycle time of each lot thus may be affected by the product mix settings in successive periods. When estimating the cycle time of each lot, the number of time periods for a lot being processed in the plant should be taken into account for determining the number of components in a contamination model. The number of weeks required for determining the number of components in a contamination model is depending on the type of applications. In the fab we study, the simulation output turns out to be three weeks. Thus, the model of the contamination of three distributions is appropriate for this application. The probability  $p_t$  can be set to 1 divided by numbers of distributions. In the case of releasing job any day during week, the model can be refined by considering each single day.

The contamination model for WT of each product type may be expressed as,

$$f(x) = \sum_{t=1}^N p_t [\phi_X(\alpha_t, \beta_t)], \quad (3-12)$$

where  $t$  is index of time period,  $N$  is the number of components in a contamination model,  $0 \leq p_t \leq 1$ ,  $\sum_{t=1}^N p_t = 1$ , and

$$\phi_X(\alpha_t, \beta_t) = \begin{cases} \frac{x^{\alpha_t-1} e^{-x/\beta_t}}{\beta_t^{\alpha_t} \Gamma(\alpha_t)}, & x > 0 \\ 0, & \text{otherwise.} \end{cases} \quad (3-13)$$

### 3.4.3. Due-date Setting

For periodical product mix changes, the  $\delta$ -percentile waiting time is determined by the fitted contamination model in order to incorporate the effect of product mix changes. The due date of order  $d$  can be assigned as



$$D_d = r_d + PT_d + WTC_d^{-1}(\delta), \quad (3-14)$$

where  $D_d$  is the due date of order  $d$ ,  $r_d$  is the release date of the latest batch of order  $d$ ,  $PT_d$  is PT of order  $d$ , and  $WTC_d^{-1}(\delta)$  is the inverse of the cumulative function of the fitted contamination distribution of order  $d$ . When the product mix is fixed throughout the time periods, the results obtained by (3-6) and (3-14) are identical.

Consider the due-date assignment example described in Section 3.2.2 with two products, L and M. Assume that the cycle times of most lots released into the plant are affected for 2 weeks. Then, the model of the contamination of two gamma distributions is appropriated for this example. In the situation that the product mix is L:M=4:6 in week 1 and L:M=6:4 in week 2, the probability density function of the WT contamination model for order 1 can be expressed as

$$f_1(x) = \frac{1}{2} \times \phi_X(25, 2) + \frac{1}{2} \times \phi_X(22, 1.6) = \frac{1}{2} \times \frac{x^{25-1} e^{-x/2}}{2^{25} \Gamma(25)} + \frac{1}{2} \times \frac{x^{22-1} e^{-x/1.6}}{1.6^{22} \Gamma(22)}. \quad (3-15)$$

The probability density function of the WT contamination model for order 2 can be expressed as

$$f_2(x) = \frac{1}{2} \times \phi_X(26, 2.3) + \frac{1}{2} \times \phi_X(28, 2.7) = \frac{1}{2} \times \frac{x^{26-1} e^{-x/2.3}}{2.3^{26} \Gamma(26)} + \frac{1}{2} \times \frac{x^{28-1} e^{-x/2.7}}{2.7^{28} \Gamma(28)}. \quad (3-16)$$

The 95-percentile of WT distribution of orders can be obtained by taking the inverse of the cumulative fitted contamination function. Based on (3-15), the 95-percentile of WT of order 1 is 63.23. Based on (3-16), the 95-percentile of WT of order 2 is 94.83. According to (3-14), the due dates of order 1 and order 2 (in days) can be solved as 10.63 and 11.99, respectively.

### 3.5. Simulation Verifications

To demonstrate the applicability of the due-date assignment model in real situations, we consider the example taken from a wafer fabrication factory located in the Science-based Industrial Park in Hsinchu, Taiwan.

#### 3.5.1. Simulation Environment

The fab consists of 83 workstations (w1 to w83), and each workstation consists of a given number of identical machines operated in parallel. W46, a stepper in the photolithography area, is the bottleneck. The planned bottleneck utilization rate is set to 90% in this study. The distribution of mean time between failures (MTBF), mean time to repair (MTTR), mean time between preventive maintenance (MTBPM), and mean time to preventive maintenance (MTTPM) for each workstation are known.

Five types of products, namely, A, B, C, D, and E, are manufactured in the system. Each product contains the numbers of circuit layers in a range of 17 to 20. All product types have different process routes and each process route contains process steps in a range of 276 to 338 operations. PT for each product is as follows: 186.8 hours for product A, 201.8 hours for product B, 187.12 hours for product C, 216.23 hours for product D, and 211.78 hours for product E.

Based on CONWIP release policy, for each specific product mix, the planned WIP level,  $L$ , is set by using Little's Law [43],  $L = \lambda \times W$ , where  $\lambda$  is the average releasing rate and  $W$  is the mean cycle time. In this system, the average releasing rate,  $\lambda$ , is equal to the throughput rate because CONWIP is adopted and mean cycle time of each product is estimated by the block-based cycle time estimation algorithm (BBCT) [18].

Based on the system capacity limitation and market demand, seven product mixes are selected. For each single product mix, simulation is run to collect PT and WT. The simulation program used in this work is eM-Plant [71]. Based on the pilot runs, for getting steady-state result, the simulation length is set to be 448 working days, in which the first 224 days are the warm-up period. In order to eliminate simulation errors, 10 replications with different random seeds are run to get adequate statistical results under each product mix. The input data for each product mix is shown in Table 3-3 and the average ( $\bar{x}$ ) and variance ( $S^2$ ) of the collected WT of each product type from running simulation for each product mix are shown in Table 3-4.

Table 3-3. The simulation inputs for single product mix

Product mix (A:B:C:D:E)	Weekly throughput target (lot)	Mean cycle time estimated by BBCT algorithm (hour)					WIP level (lot)
		Product A	Product B	Product C	Product D	Product E	
Mix(8:3:3:3:3)	167	278.19	302.18	282.50	322.23	316.29	293
Mix(6:6:2:5:1)	163	280.80	305.73	282.63	322.40	316.43	292
Mix(6:6:2:2:4)	164	281.02	306.01	282.71	322.32	316.44	292
Mix(5:6:4:4:1)	165	279.47	303.98	282.61	322.25	316.33	292
Mix(5:5:5:3:2)	166	277.79	301.71	282.47	322.08	316.17	293
Mix(5:5:5:1:4)	167	277.91	301.86	282.53	322.04	316.18	293
Mix(3:6:5:2:4)	165	276.58	300.29	282.13	321.74	315.83	292

Table 3-4. Average and variance of WT collected from simulation for single product mix

Product mix (A:B:C:D:E)	Product A		Product B		Product C		Product D		Product E	
	$\bar{x}$	$S^2$	$\bar{x}$	$S^2$	$\bar{x}$	$S^2$	$\bar{x}$	$S^2$	$\bar{x}$	$S^2$
Mix(8:3:3:3:3)	92.08	304.88	107.61	677.82	86.66	298.90	106.22	402.96	101.29	390.75
Mix(6:6:2:5:1)	95.31	333.81	102.15	350.29	86.49	329.24	104.58	332.60	98.91	201.26
Mix(6:6:2:2:4)	95.49	383.99	101.66	312.21	87.55	339.96	103.32	491.26	99.03	291.70
Mix(5:6:4:4:1)	97.26	409.32	100.85	327.00	89.98	451.71	103.31	370.68	97.51	217.46
Mix(5:5:5:3:2)	95.15	344.06	103.18	400.89	87.00	244.98	106.11	414.07	102.25	533.36
Mix(5:5:5:1:4)	95.80	404.42	102.35	411.94	88.23	307.18	100.91	184.76	100.74	339.70
Mix(3:6:5:2:4)	102.27	984.04	99.84	390.44	87.12	302.58	105.82	581.61	99.55	369.08

Table 3-5. Estimated parameters for fitted WT distributions for single product mix

Product mix (A:B:C:D:E)	Product A		Product B		Product C		Product D		Product E	
	$\hat{\alpha}$	$\hat{\beta}$	$\hat{\alpha}$	$\hat{\beta}$	$\hat{\alpha}$	$\hat{\beta}$	$\hat{\alpha}$	$\hat{\beta}$	$\hat{\alpha}$	$\hat{\beta}$
Mix(8:3:3:3:3)	27.81	3.31	17.08	6.30	25.13	3.45	28.00	3.79	26.26	3.86
Mix(6:6:2:5:1)	27.21	3.50	29.79	3.43	22.72	3.81	32.89	3.18	48.61	2.03
Mix(6:6:2:2:4)	23.74	4.02	33.10	3.07	22.55	3.88	21.73	4.76	33.62	2.95
Mix(5:6:4:4:1)	23.11	4.21	31.10	3.24	17.92	5.02	28.79	3.59	43.73	2.23
Mix(5:5:5:3:2)	26.31	3.61	26.56	3.89	30.89	2.82	27.19	3.90	19.60	5.22
Mix(5:5:5:1:4)	22.69	4.22	25.43	4.03	25.34	3.48	55.16	1.83	29.88	3.37
Mix(3:6:5:2:4)	10.63	9.62	25.53	3.91	25.08	3.47	19.25	5.50	26.85	3.71

### 3.5.2. Data Distribution Fitting

By using  $\hat{\alpha} = \bar{x}^2 / S^2$  and  $\hat{\beta} = S^2 / \bar{x}$ , we estimate the parameters for gamma distributions fitted to WT of each product type under each product mix. The estimated parameters are listed in Table 3-5. The theoretical 95-percentile WT of each fitted gamma distribution and the corresponding percentage of number of collected data are shown in Table 3-6. Comparing the 95% on-time-delivery rate with the percentage of collect data less than theoretical 95-percentile of the fitted distribution, we can see from Table 3-6 that the gamma distribution appears to fit the collected WT satisfactorily.

Table 3-6. Comparison of fitted gamma distribution and collected data

Product mix (A:B:C:D:E)	Product A		Product B		Product C		Product D		Product E	
	$T_{95\%}$ *	%**	$T_{95\%}$	%	$T_{95\%}$	%	$T_{95\%}$	%	$T_{95\%}$	%
Mix(8:3:3:3:3)	122.52	96.52	153.70	94.12	116.97	95.23	141.12	95.79	135.94	96.82
Mix(6:6:2:5:1)	127.22	96.28	134.75	95.17	118.34	96.38	136.28	96.62	123.34	96.15
Mix(6:6:2:2:4)	129.84	96.65	132.36	96.02	119.92	95.43	142.27	95.51	128.69	95.16
Mix(5:6:4:4:1)	132.75	95.96	132.32	96.75	127.55	96.07	136.88	96.04	122.97	95.07
Mix(5:5:5:3:2)	127.57	95.38	138.17	96.21	114.24	96.47	141.65	95.55	142.97	96.39
Mix(5:5:5:1:4)	131.10	95.56	137.86	95.64	118.90	96.99	124.26	95.98	132.85	95.55
Mix(3:6:5:2:4)	158.73	94.18	134.41	96.98	116.76	95.68	148.36	95.09	133.11	95.82

\*  $T_{95\%}$  : theoretical 95-percentile WT of the fitted gamma distribution.

\*\* %: percentage of number of collected data  $\leq T_{95\%}$  .

### 3.5.3. Periodical Product Mix Changes

In this section, three experiments are used to demonstrate the effectiveness and accuracy of the due-date assignment model for the environment where the product mix changes periodically. For the experiments, product mix compositions are listed in Table 3-7. Using the input data as displayed in Table 3-3 and Table 3-7, and the release policy described in Section 3.2, the simulation model is run to collect WT of each product type for each experiment.

For each experiment, the contamination model for each product type can be derived from Equations (3-12) and (3-13). The fitted contamination model and collected data distributions for experiment 1, experiment 2, and experiment 3 are plotted in Figure 3-5, Figure 3-6, and Figure 3-7, respectively. The contamination model appears to fit the collected data reasonably well.

In this study, the target on-time-delivery rate is set to 95%. After deriving the contamination model, we can obtain the 95-percentile cycle time by summing up PT and 95-percentile WT by taking the reverse of the cumulative function of the contamination model. Table 3-8 displays the 95-percentile cycle times and the on-time-delivery rate from the simulation data. Since the on-time-delivery rate can meet the target on-time-delivery rate, the due-date assignment model provides a quite good solution.

Table 3-7. Product mix composition for the experiments

Experiment	Week 1	Week 2	Week 3
Experiment 1	Mix(5:5:5:3:2)	Mix(5:5:5:1:4)	Mix(6:6:2:5:1)
Experiment 2	Mix(5:6:4:4:1)	Mix(6:6:2:2:4)	Mix(3:6:5:2:4)
Experiment 3	Mix(5:5:5:1:4)	Mix(8:3:3:3:3)	Mix(3:6:5:2:4)

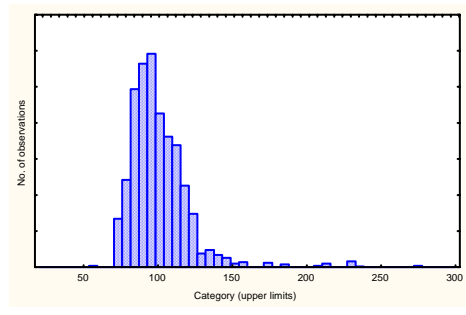
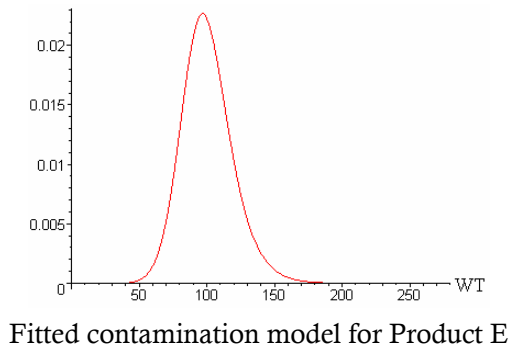
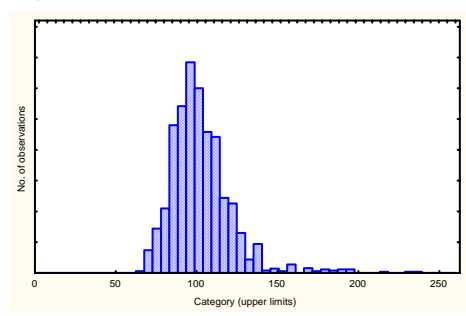
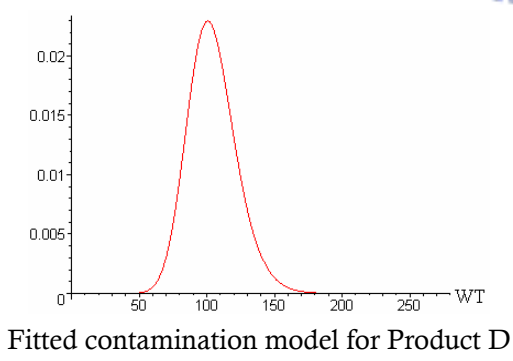
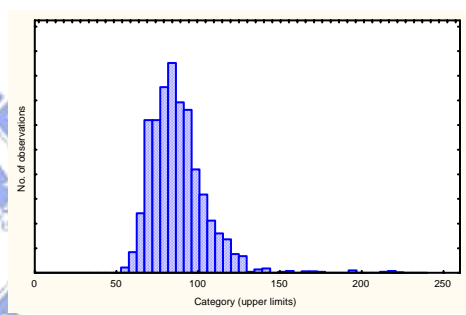
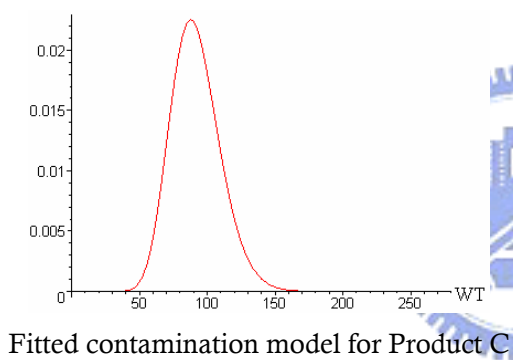
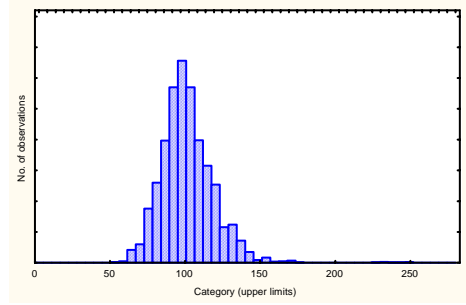
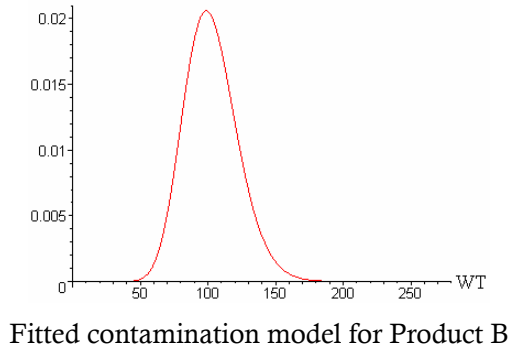
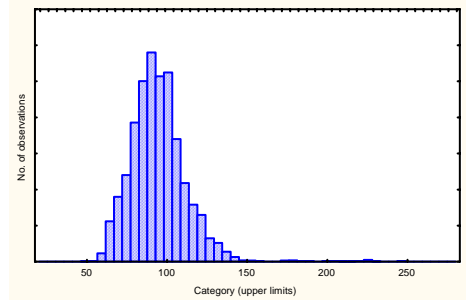
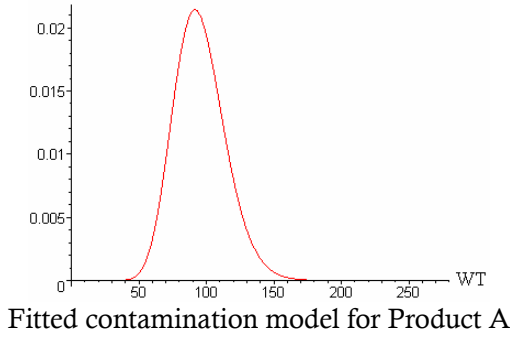


Figure 3-5. Fitted contamination model versus histogram of the collected data for experiment 1.

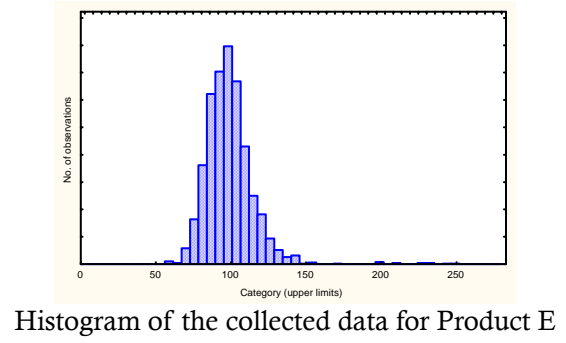
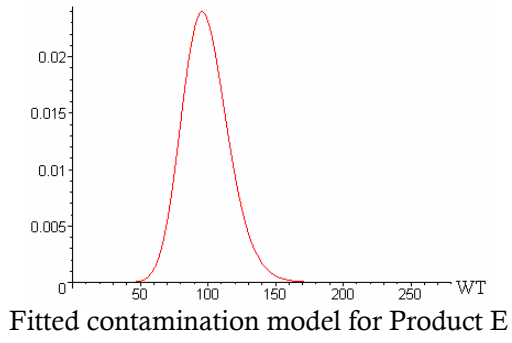
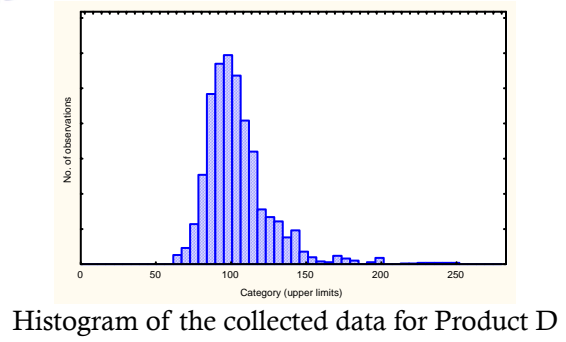
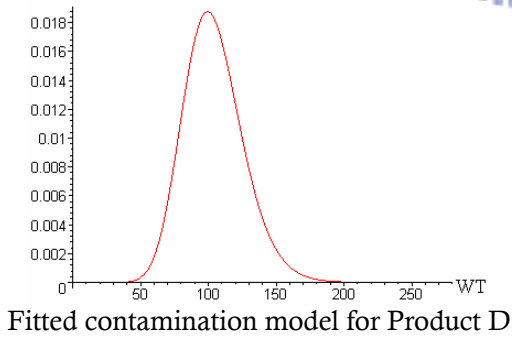
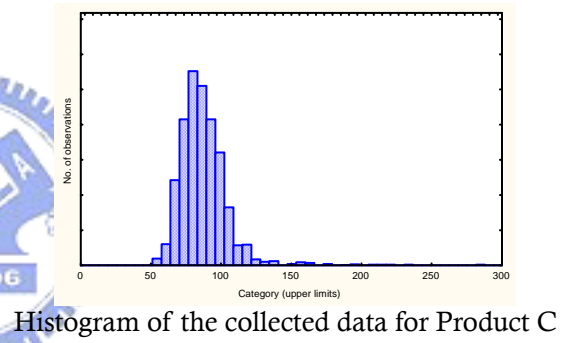
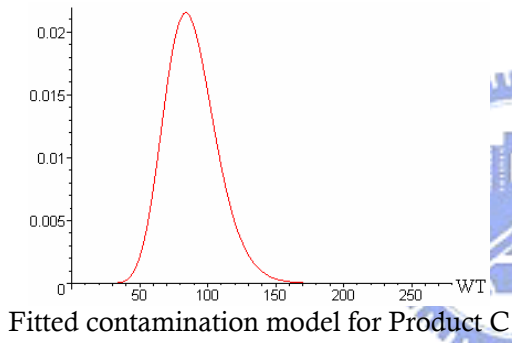
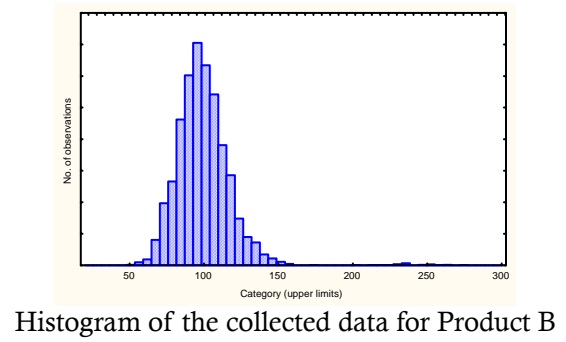
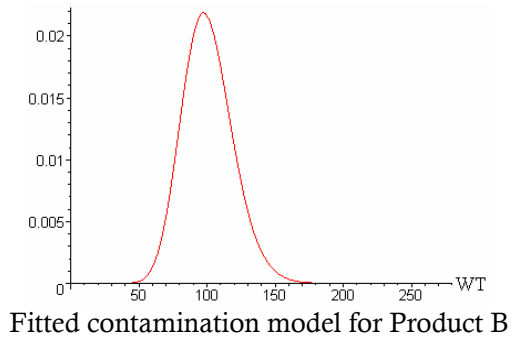
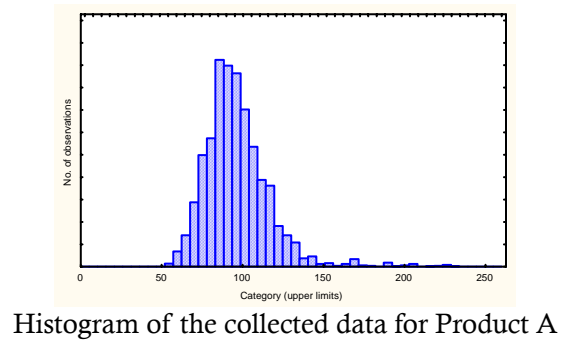
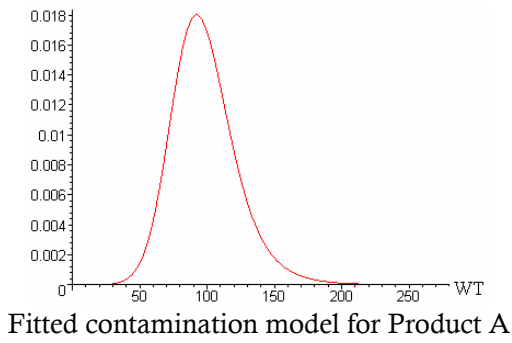


Figure 3-6. Fitted contamination model versus histogram of the collected data for experiment 2.



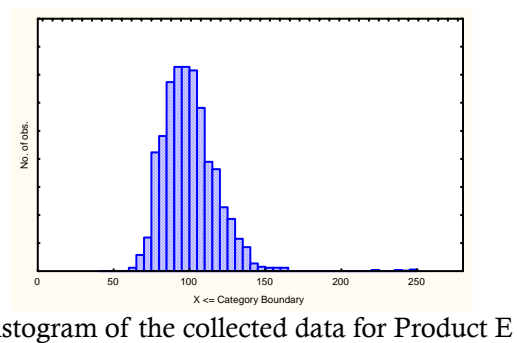
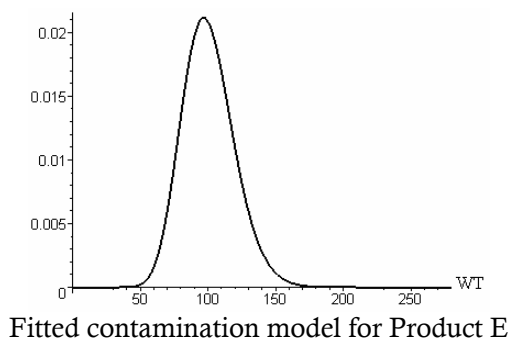
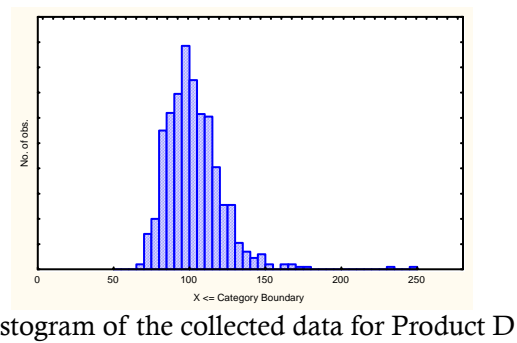
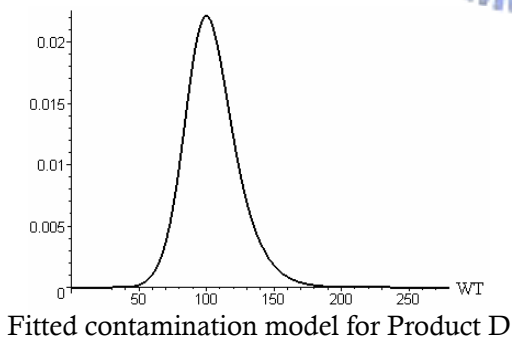
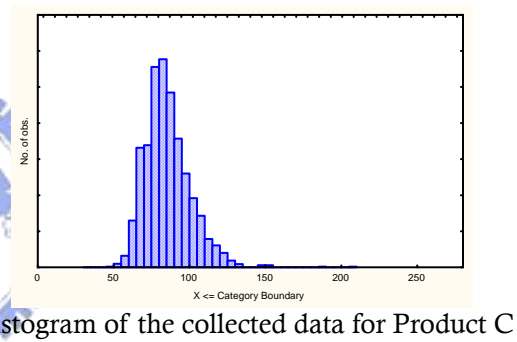
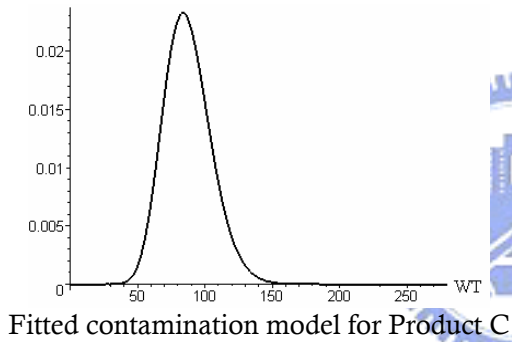
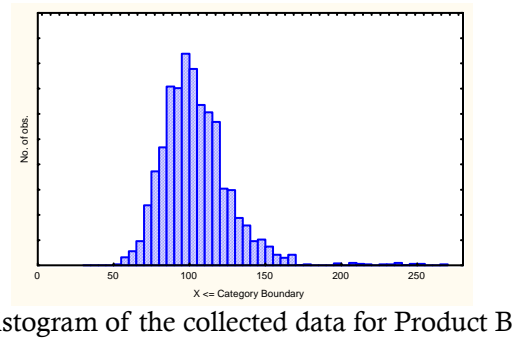
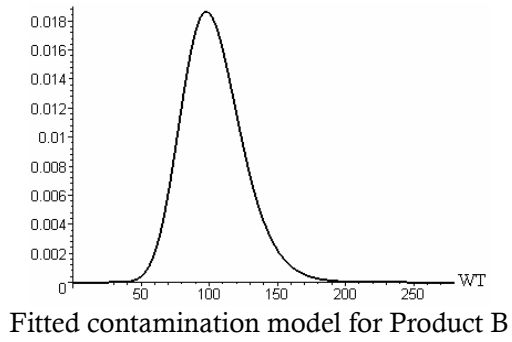
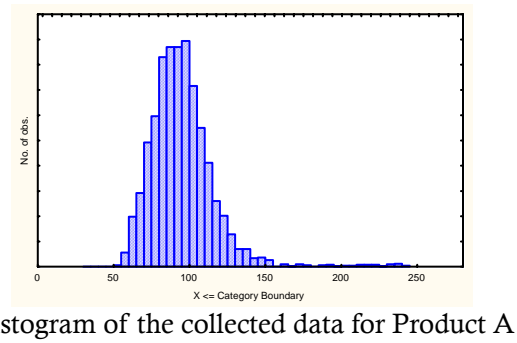
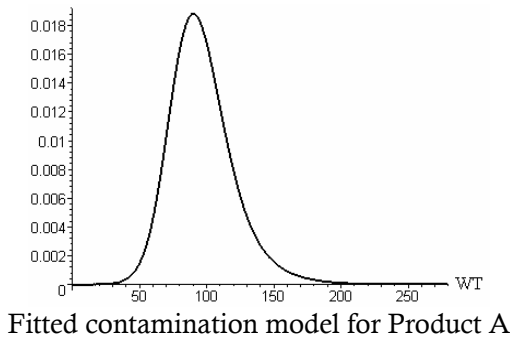


Figure 3-7. Fitted contamination model versus histogram of the collected data for experiment 3.

Table 3-8. Performance summary for the experiments

Experiment	Product type	95-percentile cycle time (hours)	%*
Experiment 1	Product A	315.45	96.75%
	Product B	338.76	96.83%
	Product C	304.33	94.79%
	Product D	352.02	95.25%
	Product E	345.63	95.42%
Experiment 2	Product A	328.53	97.12%
	Product B	334.81	96.56%
	Product C	309.04	97.37%
	Product D	358.97	94.41%
	Product E	340.40	96.71%
Experiment 3	Product A	326.69	97.39%
	Product B	344.83	94.70%
	Product C	304.89	96.78%
	Product D	355.90	96.78%
	Product E	345.79	96.68%

\* %: percentage of number of collected data  $\leq T_{95\%}$  .

## 4. Scheduling IC Assembly Operations

In order to increase a company's competition edge and profitability, the main focus of manufacturing strategies for an Integrated-Circuit (IC) manufacturer is to improve delivery time performance while minimizing production costs. Because of different product profit rates and the varied importance level of customers, there often exists more than one priority level of orders. Better on-time delivery would be the main concern of corporate level. However, ignoring setup considerations in scheduling decisions can result in loss of capacity. Therefore, any successful scheduling system needs to take the sequence-dependent nature of the setups into account [74].

In this chapter, we consider the IC assembly scheduling problem (ICASP) involves constraints on multiple job-priorities, job cluster, job-cluster dependent processing time, machine capacity, and sequentially dependent setup times. We first formulate the ICASP as an integer programming problem. The programming model considers the multiple job-priority constraint, and the processing time and the setup time in the capacity constraints. An efficient heuristic is also proposed to obtain the near-optimal solution for large scale problems.

### 4.1. The IC Assembly Process

In the IC assembly stage, materials, such as plastics and ceramic, are used to pack the good dies by forming a protective layer on electric circuits to avoid them suffering from scoring or heat punctures. Four main functions for the packaging are: to provide physical protection for each chip, to provide a barrier layer against chemical impurities and moisture, to ensure each chip connecting to electric circuit with sturdy leads, and to dissipate heat generated during chip operation. Many packaging variations exist in the industry. The IC package is selected so that the above four functions are optimized to

meet certain design constraints: performance, size, weight, reliability, and cost objectives [59].

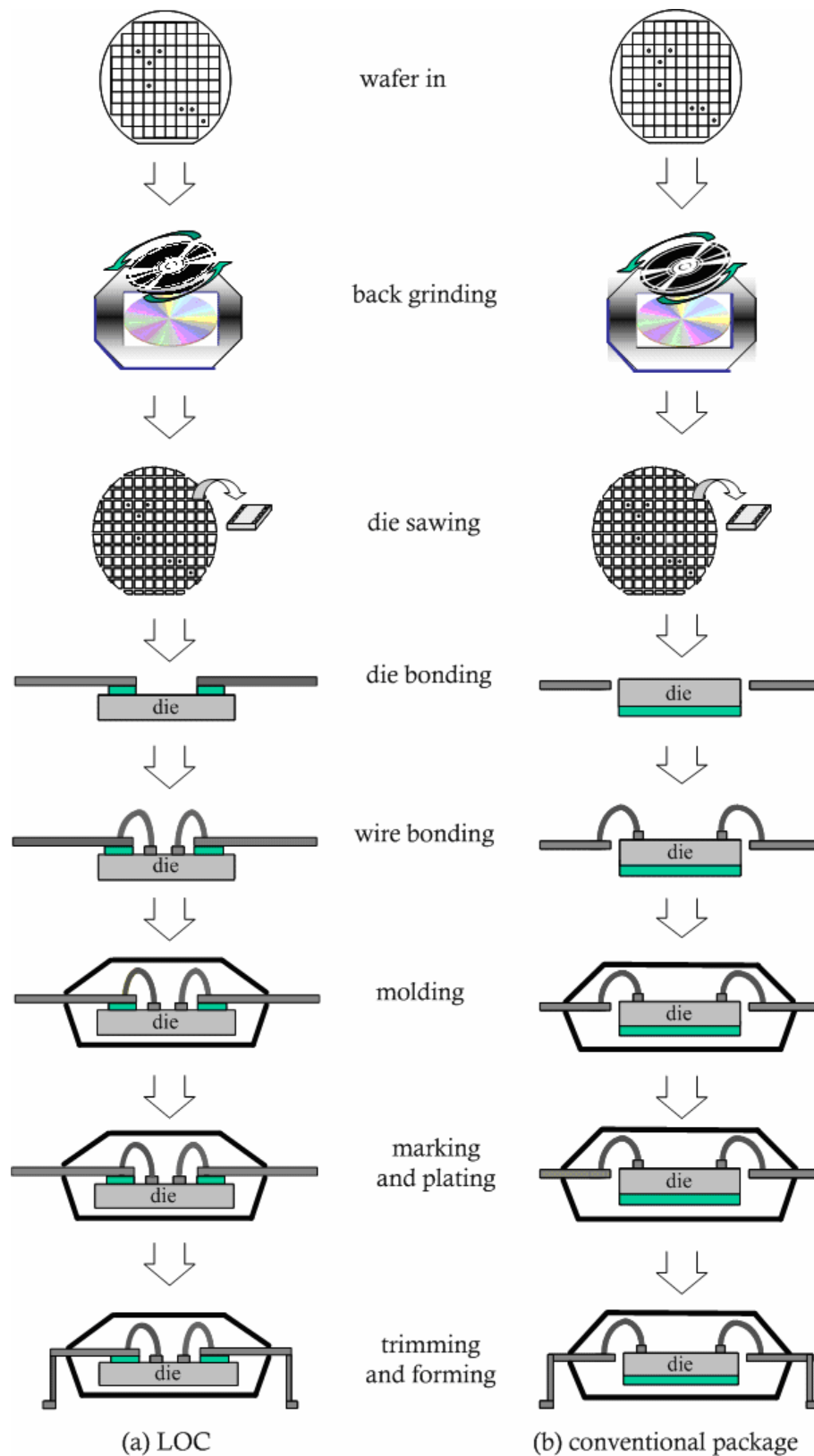


Figure 4-1. The process flows of plastic packaging products.

In this stage, wafers are first coated with a protective layer on the surface and mechanically polished on the back side to reduce thickness. Then, each wafer is taped on a solid frame from the back with a sticky, flexible material to hold it in one piece during die separation. Good chips are next picked up in the die-sorting process and attached to a lead frame in a thermal process. A wire bonding process makes the connection between bonding pads on the chip and lead pins on the lead frame with thin metal wires. Another common used technology is flip chip technology that forms metal bumps instead of bonding pads on the chip surface. Materials, such as plastics and ceramic, are used to seal the chip by forming a protective layer on electric circuits to avoid them suffering from scoring or heat punctures.

There are two types of IC packaging, namely the ceramic and the plastic. Most of the commercial IC chips use plastic packaging. For the IC assembly factory mainly producing memory product, the conventional packages and TSOP2 (Thin Small Outline Package, type 2) package dominant the production lines. The process flows of conventional package and TSOP2 package are the same. Actually, in the floor shop, the machines at each stage can process the operations for these two packages, except for at the die bonding stage. For conventional package, the die bonding process is to position the good dies on the paddle of the leadframe (using epoxy). While, for TSOP2 package, the die bonding process is lead on chip (LOC), which the device is fixed with a LOC tape underneath the leadframe, no curing needed. Therefore, due to the machine difference and cost consideration, the capacity expansion of die bonder is usually carefully evaluated. Though the critical resources in most IC assembly factories are die bonder and wire bonder [45], [58], [72], for memory products, the package lead count of each die is relatively small and the throughput of the wire bonders is satisfied. Therefore, in the assembly facility mainly producing memory products, the die bonders are treated as the

bottleneck. Developing efficient scheduling methods to minimize the total die bonder workload and enhance the utilization of the die bonder is essential.

In contrast to the front-end processes are highly reentrant, the back-end process follows a more linear, assembly-line type of flow [56], [72]. In the IC assembly scheduling problem, the bottleneck, the die-bonders, is scheduled to be utilized as efficiently as possible, and this implies the reduction of number of setups is crucial. After completing the scheduling on the bottleneck, the lot release time and the scheduling on all the non-bottlenecks facilitate the feeding of the bottleneck.

#### **4.2. The IC Assembly Scheduling Problem (ICASP)**

For the ICASP investigated in this research, the jobs are assigned processing priorities and are clustered by their product families with each family containing several product types, which must be processed on a group of identical parallel machines. Further, the job processing time may vary, depending on the product type (job cluster) of the job process on. Setup times for two consecutive jobs of different product types (job clusters) on the same machine are sequentially dependent. The objective of the ICASP is to find a schedule for the jobs, which satisfies the priority processing restrictions without violating the machine capacity constraints, while the total machine workload is minimized. Minimizing the total setup time is equivalent to the minimization of the total machine workload.

The IC assembly scheduling problem is to seek a schedule for the jobs to be processed in the time horizon, which minimizes the total die bonder workload, satisfying the job priority without violating the machine capacity constraint. In IC assembly scheduling problem, the jobs are processed on groups of identical die bonders and the

total processing time is constant. Thus, reducing the total setup time is essential to the minimization of the total machine workload. Process characteristics modeled include sequence dependent setup times, multiple job-priority consideration, and machine capacity constraint.

These integrated circuits, or dies, are formed on wafers that are typically grouped into lot sizes of 25. The size of each lot may vary which depends on the design of dies and die yield. Note that, at die bonding stage, a lot flowing into the die bonding area is in the form of complete wafer, and it flows out of this area in the form of die on leadframe.

#### ***4.2.1. Sequence Dependent Setup Times***

Since different types of dies must be operated on the LOC die bonder with some specific size of chop table, mount head and mount stage, and some parameter setting on the machines, some setup operations may be required. Figure 4-2 shows the die bonding on the LOC die bonder. The required for parameter settings can be regarded as a fixed constant. In the situation, where the current job is formed on 12-inch wafer and the next job is formed on 8-inch wafer, and vice versa, the next job would have to put on hold until the chop table is changed. Furthermore, in the situation, where the current job is performed with small mount head and mount stage and the die size of next job is large, the next job would have to put on hold until the mount stage and mount head is changed.

Thus, the setup time required for switching one product type to another depends on the size of wafer and die. As the jobs come from several product families with each family including a few product types, switching a job to another among different product type within the same product family only require the parameter setting operations on the machine. In other cases, switching a job from one to another among different product

type from different product families must consider the total corresponding setup time occurring due to changing chop table, changing mount stage and mount head, and the parameter setting operations on the machine.

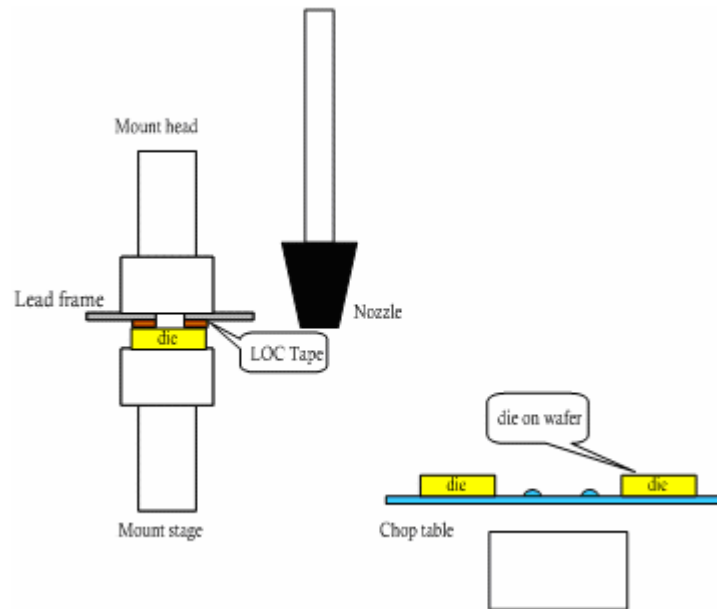


Figure 4-2. Die bonding on the LOC die-bonder.

#### ***4.2.2. Multiple job-Priorities Consideration***

Because of different product profit rates and the varied importance level of customers, there often exists more than one priority levels of orders in most semiconductor companies [26], [74]. Based on the job priority, for any two jobs scheduled on the same machine, job A with higher job priority must be completed before job B with lower job priority can be begun. Throughout this dissertation, we assume that each lot is assigned a value of job priority, which is known at the beginning of the planning horizon. The assignment of job priority method is beyond the scope of this dissertation, and we refer the interested reader to [46], [58] for approaches to assign priority value.



### 4.2.3. Machine Capacity Constraint

Normally, the lead time for the total assembly portion is 4 to 6 days. By deducting the setup times and processing times on the non-bottleneck machines, the time horizon of the bottleneck is set to 2 days. Due to the variety of lot size and the importance of the process lot in the initial status, a rolling horizon approach is used in the ICASP. In real-life applications, the capacity for each machine can be set based on the available capacity in the time horizon, and the processing time unit can be “minute” or “hour”. Throughout this dissertation, we have set the “minute” as the unit of the processing time, setup time, machine workload, and machine capacity in our investigation.

### Problem Complexity

The ICASP is NP-hard. Even without the multiple job-priority constraint, the ICASP special case which minimizes makespan on a single machine in the presence of sequence-dependent setup times is equivalent to the Traveling Salesman Problem, and it has been shown to be NP-hard [26] [44].

## 4.3. An Integer Programming Formulation for ICASP

A mathematical programming formulation is a natural way to solve machine scheduling problems [4] [61]. The IP formulations for ICASP have been investigated, but our IP formulation includes both sequentially dependent setup times and multiple job priority conditions at the same time, therefore, is considerably more complicated than those in [72].

We first define  $R = \{R_0, R_1, R_2, \dots, R_I\}$  containing  $I + 1$  clusters of jobs, each job cluster  $R_i = \{r_{ij} | j = 0, 1, 2, \dots, J_i\}$  containing  $J_i$  ( $j = 1, 2, \dots, J_i$ ) jobs to be processed and one pseudo-job  $r_{i0}$  ( $j = 0$ ) which is used as the initial status of a machine. Thus, job

cluster  $R_0 = \{r_{00}\}$  contains one pseudo-job, job cluster  $R_1 = \{r_{10}, r_{11}, r_{12}, \dots, r_{1J_1}\}$  contains  $J_1 + 1$  jobs, job cluster  $R_i = \{r_{i0}, r_{i1}, r_{i2}, \dots, r_{iJ_i}\}$  contains  $J_i + 1$  jobs, and job cluster  $R_I = \{r_{I0}, r_{I1}, r_{I2}, \dots, r_{IJ_I}\}$  contains  $J_I + 1$  jobs. We also define  $A = \{0, 1, 2, \dots, H\}$  as the set of job priority code containing  $H + 1$  priority levels. Let  $h_{ij}$  ( $h_{ij} \in A$ ) be the job priority code of job  $r_{ij}$ . This code is in the form of a non-negative integer, in such a way, a smaller priority code of job indicates that this job has a higher job priority. Thus, set  $h_{ij} < h_{i'j'}$  ( $h_{ij}, h_{i'j'} \in A$ ) if job  $r_{ij}$  has a higher priority than job  $r_{i'j'}$ .

We also define  $M = \{m_1, m_2, \dots, m_k\}$  as the group of machines containing a set of  $K$  identical machines. Let  $W_k$  be the predetermined machine capacity expressed in terms of processing time unit. Further, let  $n_{ij}$  be the lot size of job  $r_{ij}$ , and  $p_{ik}$  be the unit processing time for each job  $r_{ij}$  in cluster  $R_i$  ( $r_{ij} \in R_i$ ) on machine  $K$ . Therefore, the job processing time for job  $r_{ij}$  is  $n_{ij}p_{ik}$ . Let  $s_{ii'}$  be the sequence dependent setup time between any two consecutive jobs  $r_{ij} (\in R_i)$  and  $r_{i'j'} (\in R_{i'})$  from different job clusters ( $i \neq i'$ ). Note that, the priority codes and lot size for the job  $r_{i0}$  should be set to zero so that these pseudo-jobs should be scheduled as the first jobs on each machine, which indicates the initial status of each machine.

Let  $x_{ijk}$  be the variable indicating whether the job  $r_{ij}$  is scheduled on machine  $m_k$ , with  $x_{ijk} = 1$  if job  $r_{ij}$  is scheduled on machine  $m_k$ , and  $x_{ijk} = 0$  otherwise. Let  $y_{iji'j'k}$  be the precedence variable defined on two jobs  $r_{ij}$  and  $r_{i'j'}$  scheduled on machine  $m_k$ , with  $y_{iji'j'k} = 1$  if job  $r_{ij}$  precede job  $r_{i'j'}$  (not necessarily directly), and

$y_{ij'i'j'k} = 0$  otherwise. Let  $z_{ij'i'j'k}$  be the direct-precedence variable defined on two jobs  $r_{ij}$  and  $r_{i'j'}$  scheduled on machine  $m_k$ , with  $z_{ij'i'j'k} = 1$  if job  $r_{ij}$  direct precede job  $r_{i'j'}$ , and  $z_{ij'i'j'k} = 0$  otherwise.

To find a schedule for the jobs which minimize the total machine workload without violating the machine capacity and job priority constraints, we consider the following integer programming model. Although the first summation term (the total processing time) in the objective function of the integer programming model is constant, it is necessary to be used to provide the information of total machine workload in the solutions because managers prefer to know the total machine workload instead of only the total setup times. In addition, with the processing time included in the objective function, the integer programming model can be used to solve a more general ICASP problem, where the machines are unrelated.

$$\text{Minimize } \left\{ \sum_{k=1}^K \left\{ \sum_{i=0}^I \sum_{j=0}^{J_i} x_{ijk} n_{ij} p_{ik} + \sum_{i=0}^I \sum_{j=0}^{J_i} \left( \sum_{i'=0}^I \sum_{j'=0}^{J_{i'}} z_{ij'i'j'k} s_{i'i'} \right) \right\} \right\}$$

subject to

$$x_{i_k 0k} = 1, \text{ for all } k, \quad (4-1)$$

$$\sum_{i=0}^I x_{i0k} = 1, \text{ for all } k, \quad (4-2)$$

$$\sum_{k=1}^K x_{ijk} = 1, \text{ for all } i, j > 0, \quad (4-3)$$

$$\sum_{i=0}^I \sum_{j=0}^{J_i} x_{ijk} n_{ij} p_{ik} + \sum_{i=0}^I \sum_{j=0}^{J_i} \left( \sum_{i'=0}^I \sum_{j'=0}^{J_{i'}} z_{ij'i'j'k} s_{i'i'} \right) \leq W_k, \text{ for all } k, \quad (4-4)$$

$$(y_{ij'i'j'k} + y_{i'j'ijk}) - x_{ijk} \leq 0, \text{ for all } i, j, i', j', k, r_{ij} \neq r_{i'j'}, \quad (4-5)$$

$$(y_{ij'i'j'k} + y_{i'j'ijk}) - x_{i'j'k} \leq 0, \text{ for all } i, j, i', j', k, r_{ij} \neq r_{i'j'}, \quad (4-6)$$

$$(y_{ij'i'j'k} + y_{i'j'ijk}) - (x_{ijk} + x_{i'j'k} - 1) \geq 0, \quad \text{for all } i, j, i', j', k, r_{ij} \neq r_{i'j'}, \quad (4-7)$$

$$(h_{ij} - h_{i'j'}) \times (y_{ij'i'j'k} - y_{i'j'ijk}) \leq 0, \quad \text{for all } i, j, i', j', k, r_{ij} \neq r_{i'j'}, \quad (4-8)$$

$$y_{ij'i^*j^*k} - Q(y_{ij'i'j'k} + y_{i'j'i^*j^*k} - 2) \geq 1, \quad \text{for all } i, j, i^*, j^*, i', j', k, r_{ij} \neq r_{i^*j^*} \neq r_{i'j'}, \quad (4-9)$$

$$y_{ij'i'j'k} \geq z_{ij'i'j'k}, \quad \text{for all } i, j, i', j', k, r_{ij} \neq r_{i'j'}, \quad (4-10)$$

$$\sum_{i=0}^I \sum_{j=0}^{J_i} x_{ijk} - \sum_{r_{ij} \neq r_{i'j'}} z_{ij'i'j'k} = 1, \quad \text{for all } k, \quad (4-11)$$

$$\sum_{r_{ij} \neq r_{i'j'}} z_{ij'i'j'k} \leq 1, \quad \text{for all } i, j, i', j', k, \quad (4-12)$$

$$\sum_{r_{ij} \neq r_{i'j'}} z_{i'j'ijk} \leq 1, \quad \text{for all } i, j, i', j', k, \quad (4-13)$$

$$x_{ijk} \in \{0, 1\}, \quad \text{for all } i, j, k, \quad (4-14)$$

$$y_{ij'i'j'k} \in \{0, 1\}, \quad \text{for all } i, j, i', j', k, r_{ij} \neq r_{i'j'}, \quad (4-15)$$

$$z_{ij'i'j'k} \in \{0, 1\}, \quad \text{for all } i, j, i', j', k, r_{ij} \neq r_{i'j'}. \quad (4-16)$$

The objective function seeks to minimize the sum of the total processing time  $\sum_{i=0}^I \sum_{j=0}^{J_i} x_{ijk} n_{ij} p_{ik}$  and the total setup time  $\sum_{i=0}^I \sum_{j=0}^{J_i} (\sum_{i'=0}^I \sum_{j'=0}^{J_{i'}} z_{ij'i'j'k} s_{ii'})$  over the  $K$  identical machines. The constraints in (4-1) assigns the initial status of each machine  $m_k$ . For example, in the situation, where the initial status of machine  $m_1$  is pseudo-job  $r_{30}$ , we will assign  $i_1=3$  and  $x_{301}=1$ . The constraints in (4-2) guarantee that only one pseudo-job  $r_{i0}$  is scheduled on a machine. Constraints in (4-3) guarantee that job  $r_{ij}$  is processed by one machine exactly once. The constraints in (4-4) state that each machine workload does not exceed the machine capacity.

The constraints in (4-5), (4-6), and (4-7) ensure that one job should precede another  $(y_{ij'i'j'k} + y_{i'j'ijk} = 1)$  if two jobs  $r_{ij}$  and  $r_{i'j'}$  are scheduled on the same machine  $m_k$

( $x_{ijk} = 1$  and  $x_{i'j'k} = 1$ ), while  $y_{ij'j'k} + y_{i'j'ijk} \leq 0$  if two jobs  $r_{ij}$  and  $r_{i'j'}$  are not scheduled on the same machine  $m_k$  ( $x_{ijk} = 0$  or  $x_{i'j'k} = 0$ ). The constraints in (4-8) ensure that job with smaller or equal priority code (higher or equal job priority) should precede the other job with larger or equal priority code (lower or equal job priority) when two jobs are scheduled on the same machine  $m_k$  ( $y_{ij'j'k} = 1$  and  $y_{i'j'ijk} = 0$ , if  $h_{ij} \leq h_{i'j'}$ ) or ( $y_{ij'j'k} = 0$  and  $y_{i'j'ijk} = 1$ , if  $h_{ij} \geq h_{i'j'}$ ). The constraints in (4-9) ensure that the job  $r_{ij}$  precedes job  $r_{i^*j^*}$  ( $y_{ij^*j^*k} = 1$ ) when the job  $r_{ij}$  precedes job  $r_{i'j'}$  ( $y_{ij'j'k} = 1$ ) and the job  $r_{i'j'}$  precedes job  $r_{i^*j^*}$  ( $y_{i'j'i^*j^*k} = 1$ ).

The constraints in (4-10) ensure that job  $r_{ij}$  could precede job  $r_{i'j'}$  directly ( $z_{ij'j'k} = 1$ ) only when  $y_{ij'j'k} = 1$  and job  $r_{ij}$  could not precede job  $r_{i'j'}$  directly ( $z_{ij'j'k} = 0$ ) if job  $r_{ij}$  is scheduled after  $r_{i'j'}$  ( $y_{ij'j'k} = 0$ ). The constraints in (4-11) state that there should exist  $n - 1$  direct-precedence variables, which are set to one, on the schedule with  $n$  jobs. The constraints in (4-12) guarantee that at most one job  $r_{i'j'}$  could be scheduled after job  $r_{ij}$  directly for all the jobs scheduled on the same machine  $m_k$ . The constraints in (4-13) guarantee that at most one job  $r_{i'j'}$  could be scheduled precede job  $r_{ij}$  directly for all the jobs scheduled on the same machine  $m_k$ .

In the integer programming formulation above, the total number of variables and equations increase as the number of machines or the number of jobs increase. The computational complexity of the integer programming model is as follows. For a parallel-machine problem with  $I$  job clusters and  $K$  machines, containing a total of  $N_I = 1 + (J_1 + 1) + (J_2 + 1) + \dots + (J_I + 1)$  jobs, the integer programming model contains

$N_I K$  variables of  $x_{ijk}$ ,  $N_I K(N_I - 1)$  variables of  $y_{iji'j'k}$ , and  $N_I K(N_I - 1)$  variables of  $z_{iji'j'k}$ . Further, the constraint sets in (4-1), (4-2), (4-4) and (4-12) each contains  $K$  equations, the constraint set in (4-3) contains  $N_I - (I + 1)$  equations, constraint sets in (4-5)~(4-8), and (4-10) each contains  $N_I K(N_I - 1)$  equations, the constraint sets in (4-9) contains  $N_I K(N_I - 1)(N_I - 2)$  equations, and the constraint sets in (4-12) and (4-13) contains  $N_I K$ . Thus, the total number of variables is  $2N_I^2 K - N_I K$ , and the total number of equations is  $N_I^3 K + 2N_I^2 K - N_I K + N_I + 4K - (I + 1)$ .

To accelerate the execution in solving the integer programming problem, we use both a depth-first search strategy by choosing the most recently created node [26][36], and a strong branching rule causing variable selection based on partially solving a number of sub-problems with tentative branches to find the most promising branch [36]. By using the depth-first search strategy, when the tree size or the number of fully developed branches exceeds limitations induced by computation time or memory requirements, the program terminates and returns the best solution achieved. The implementation thus allows us to set various limits on the number of memory nodes so that feasible solutions may be obtained efficiently within reasonable amount of computer time. The node limit is set to determine the maximum number of nodes solved before the program terminates, without reaching optimality [36].

#### 4.4. An Illustrative Example

Consider the following ICASP example with two parallel machines ( $m_1$  and  $m_2$ ), two job priority levels (1 and 2, in which a job with priority 1 has a higher priority than the jobs with priority 2), and three clusters of jobs ( $R_1$ ,  $R_2$ , and  $R_3$ ) ready for processing initially, as shown in Figure 4-3. Job cluster  $R_1$  contains three jobs,  $r_{11}$  with priority 1

and both  $r_{12}$  and  $r_{13}$  with priority 2. Job cluster  $R_2$  contains four jobs, both  $r_{21}$  and  $r_{22}$  with priority 1 and both  $r_{23}$  and  $r_{24}$  with priority 2. Job cluster  $R_3$  contains three jobs,  $r_{31}$  with priority 1 and both  $r_{32}$  and  $r_{33}$  with priority 2.

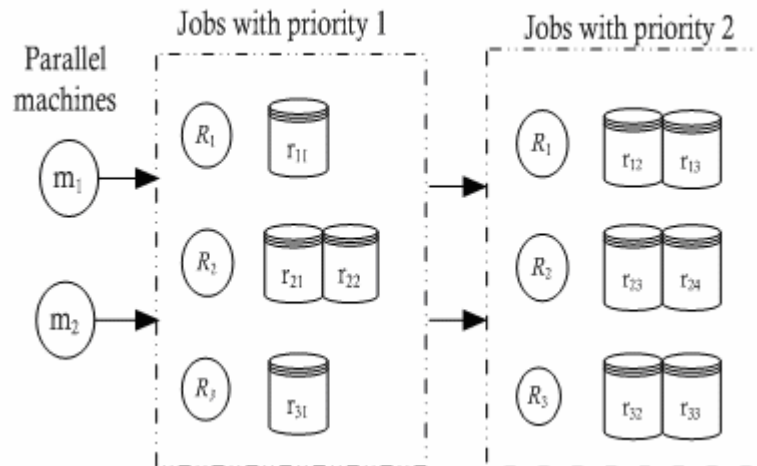


Figure 4-3. The ICASP example with two parallel machines, two priority levels, and three job clusters.

Table 4-1. Setup times required for switching one product type to another for  $R_1$ ,  $R_2$ , and  $R_3$

From	To			
	U	$R_1$	$R_2$	$R_3$
U	—	6	6	10
$R_1$	0	0	6	10
$R_2$	0	10	0	6
$R_3$	0	10	3	0

Table 4-2. Job processing times for  $R_1$ ,  $R_2$ , and  $R_3$  on the machines  $m_1$  and  $m_2$

	$R_1$	$R_2$	$R_3$
$m_1$	25	12	15
$m_2$	25	12	15

Table 4-1 displays the setup times required for switching one product type to another for the three types 1, 2, and 3. In Table 4-1 the label U denotes that the machine is in idle status. Table 4-2 displays the job processing times for job clusters  $R_1$ ,  $R_2$ , and  $R_3$  on the machines  $m_1$  and  $m_2$ . Note that the setup times and the processing times are associated with the product types, regardless of job priority levels. The capacity of each machine is set to 100 minutes in this example. The initial status of machine  $m_1$  is  $r_{10}$ , and that of machine  $m_2$  is  $r_{20}$ .

To solve the integer programming problem for the ICASP example, we adopt ILOG OPL [20] to generate the constraints and variables of the model. For the ICASP example with two machines, two job priority levels, three job clusters, and ten jobs, as shown in Figure 4-3, the model contains 756 variables and 6262 equations. We run the integer programming model using the IP software ILOG OPL 3.6 on a Pentium IV 3.0 GHz PC. The optimal solution for this example is shown in Figure 4-4. The total machine workload is 183. For machine  $m_1$ , the total machine workload is 93 with setup time 6 and processing time 87. For machine  $m_2$ , the total machine workload is 90 with setup time 9 and processing time 81. We note that the job priority constraints and the machine capacity constraints are satisfied for the solution.

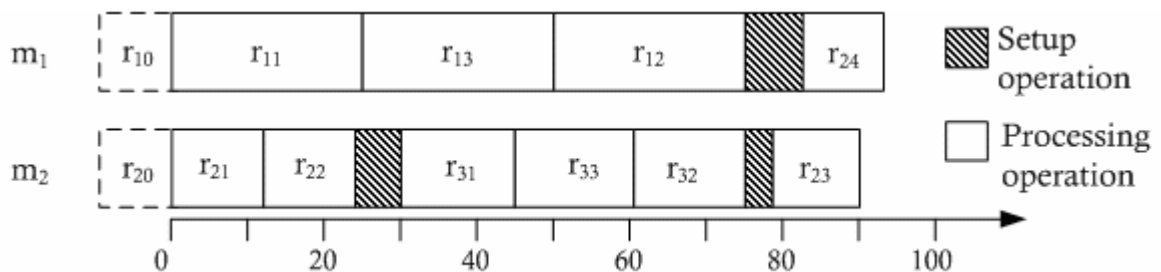


Figure 4-4. The optimal solution on the ICASP example.



Table 4-3. The integer programming solution (optimal) for the ICASP example by CPLEX 8.0

<i>The objective value and the solution time</i>							
Integer optimal solution: Objective = 183							
Solution time = 13.13 seconds    Iterations = 7390    Nodes = 145							
<i>The statistics of the model</i>							
Constraints:		6262		[Less: 1150 Greater: 5096, Equal: 16]			
Variables:		756		[Binary: 756]			
<i>The values for all variables</i>							
<i>Name</i>	<i>Value</i>	<i>Name</i>	<i>Value</i>	<i>Name</i>	<i>Value</i>	<i>Name</i>	<i>Value</i>
X101	1	Y11131	1	Z22312	1	Y21332	1
X111	1	Y11241	1	Z31332	1	Y22232	1
X121	1	Y12241	1	Z32232	1	Y22312	1
X131	1	Y13121	1	Z33322	1	Y22322	1
X241	1	Y13241	1	Y20212	1	Y22332	1
Z10111	1	X202	1	Y20222	1	Y31232	1
Z11131	1	X212	1	Y20232	1	Y31322	1
Z12241	1	X222	1	Y20312	1	Y31332	1
Z13121	1	X232	1	Y20322	1	Y32232	1
Y10111	1	X312	1	Y20332	1	Y33232	1
Y10121	1	X322	1	Y21222	1	Y33322	1
Y10131	1	X332	1	Y21232	1		
Y10241	1	Z20212	1	Y21322	1		
Y11121	1	Z21222	1	Y21322	1		
All other variables in the range 1-756 are zero							

Table 4-3 displays output solution of the integer programming model. The variables  $X_{101}=1$ ,  $X_{111}=1$ ,  $X_{121}=1$ ,  $X_{131}=1$ , and  $X_{241}=1$  indicate that the jobs  $r_{10}$ ,  $r_{11}$ ,  $r_{12}$ ,  $r_{13}$ , and  $r_{24}$  are scheduled on machine  $m_1$ . The variables  $Z_{10111}=1$ ,  $Z_{11131}=1$ ,  $Z_{12241}=1$ , and  $Z_{13121}=1$  imply that job  $r_{10}$  precedes job  $r_{11}$  directly,  $r_{11}$  precedes job  $r_{13}$  directly, job  $r_{12}$  precedes job  $r_{24}$  directly, and job  $r_{13}$  precedes job  $r_{12}$  directly. Thus, there is one product type changes, from  $R_1(r_{12})$  to  $R_2(r_{24})$ .

The variables  $X_{202}=1$ ,  $X_{212}=1$ ,  $X_{222}=1$ ,  $X_{232}=1$ ,  $X_{312}=1$ ,  $X_{322}$  and  $X_{332}=1$  indicate that the jobs  $r_{20}$ ,  $r_{21}$ ,  $r_{22}$ ,  $r_{23}$ ,  $r_{31}$ ,  $r_{32}$  and  $r_{33}$  are scheduled on machine

$m_2$ . The variables  $Z_{20212}=1$ ,  $Z_{21222}=1$ ,  $Z_{22312}=1$ ,  $Z_{31332}=1$ ,  $Z_{32232}=1$ , and  $Z_{33322}=1$  imply that job  $r_{20}$  precedes job  $r_{21}$  directly, job  $r_{21}$  precedes job  $r_{22}$  directly, job  $r_{22}$  precedes job  $r_{31}$  directly, job  $r_{31}$  precedes job  $r_{33}$  directly, job  $r_{32}$  precedes job  $r_{23}$  directly, and job  $r_{33}$  precedes job  $r_{32}$  directly. Thus, there are two product type changes, from  $R_2(r_{22})$  to  $R_3(r_{31})$  and from  $R_3(r_{32})$  to  $R_2(r_{23})$ .

Note that the solution will be different when the initial statuses of machines are different. When the initial status of machine  $m_1$  is idle ( $r_{00}$ ) and that of machine  $m_2$  is  $r_{30}$ , the optimal solution will become 189, as shown in Figure 4-5. For machine  $m_1$ , the total machine workload is 99 with setup time 12 and processing time 87. For machine  $m_2$ , the total machine workload is 90 with setup time 9 and processing time 81.

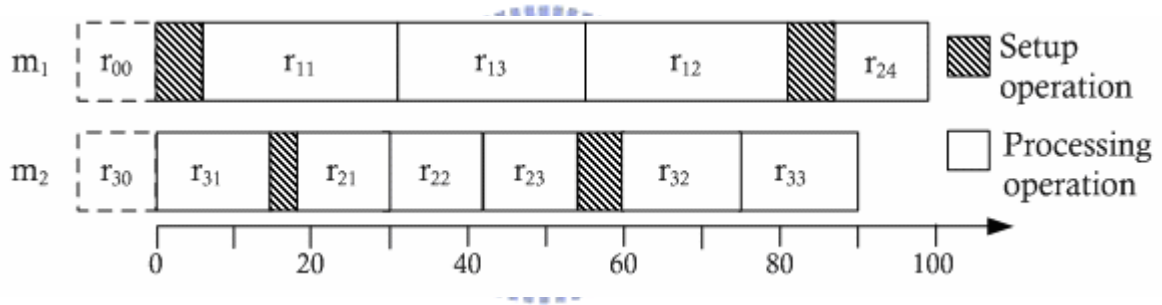


Figure 4-5. The optimal solution on the ICASP example with initial status of  $r_{00}$  and  $r_{30}$ .

#### 4.5. A heuristic algorithm

For large scale problems, the depth-first strategy can solve the problem with more computation effort. However, if the computational run time is primary concern, a heuristic algorithm may be considered.

In this section, we extend the savings algorithms investigated by Clark and Wright [20], Golden [27], Christofides et al. [17], and Altinkemer and Gavish [3] to solve the ICASP. The main concept of our algorithm is as follows.

1. Considering the initial status of each machine may differ, creates a multiple of  $K$  machine schedules simultaneously at the initial stage, where  $K$  is the number of machines.
2. Selecting a feasible job resulting the smallest setup times to extend the partial schedule  $k$ .
3. A job is feasible if it does not violate the machine capacity constraints and the priority restrictions.
4. If there is a tie for the feasible jobs, choose the job with the highest priority.

The proposed algorithm essentially consists of two phases. Phase I creates a multiple of  $K$  machine schedules simultaneously by finding the feasible job with the smallest setup times to add it to the end of partial schedule  $PS_k$ . Note that a job is feasible and is added to the machine schedule only if the capacity constraint and the job priority restrictions are not violated. After Phase I, partial schedules like  $PS_k = (r_{i_k 0}, u_1, \dots, u_{g-1}, u_g, \dots, u_{G_k})_k$  should be generated, in which  $r_{i_k 0}$  represents the initial status of machine  $m_k$ ,  $u_g$  represents the job be scheduled at position  $g$  on machine  $m_k$ , and  $G_k$  represents the total number of jobs in the schedule  $PS_k$ .

For the jobs left unscheduled in Phase I due to the job priority constraint, in Phase II, we calculate the insertion cost of every unscheduled job  $r_{ij}$  at every possible position of each partial schedule  $PS_k$  to insert the job to the lowest insertion cost position. Note that a job is inserted into the machine schedule only if the capacity constraint and the job priority restrictions are not violated. Let  $\lambda_k(u_{g-1}, r_{ij}, u_g)$  be the additional setup cost when job  $r_{ij}$  is inserted between position  $g-1$  and  $g$  in schedule  $PS_k$ . Note that the setup time  $S_{I(u_{g-1})I(u_g)}$  is determined by the product types,  $I(u_{g-1})$  and  $I(u_g)$ , of job  $u_{g-1}$  and  $u_g$ .

The procedures of the proposed heuristic algorithm are described as follows.

*Phase I- Schedule construction*

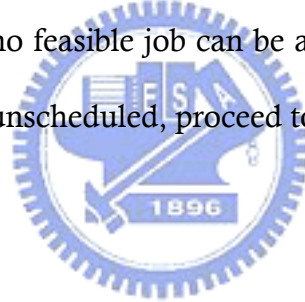
*Step 1:* For each machine  $m_k$ , let partial schedule  $PS_k = (r_{i_0})_k$  initially.

*Step 2:* Sort the setup times for all pairs of job type  $i$  and  $i'$  and create a list in ascending order of magnitude.

*Step 3:* Starting from the top of the setup time list, find the first feasible link in the list, which can be used to extend the end of  $PS_{k^*}$  without violating the machine capacity constraints and the priority restrictions. If there is a tie for the feasible jobs, choose the job with the highest priority.

*Step 4:* Repeat Step 3 until no feasible job can be added to extend the end of any  $PS_k$ .

If there are jobs left unscheduled, proceed to Phase II. Otherwise, stop.



*Phase II- Job Insertion*

*Step 1:* For each unscheduled job  $r_{ij}$ , first compute its best feasible insertion position, by

$\lambda_k^*(u_{g-1}, r_{ij}, u_g)$  in each machine's partial schedule  $PS_k$ :

$$\lambda_k(u_{g-1}, r_{ij}, u_g) = S_{I(u_{g-1})i} + S_{iI(u_g)} - S_{I(u_{g-1})I(u_g)}.$$

*Step 2:* The job  $r_{ij}$  is inserted into the lowest insertion cost position of the machine

$m_{k^*}$  determined by the lowest insertion cost  $\lambda_{k^*}^*(u_{g-1}, r_{ij}, u_g)$ .

$$\lambda_{k^*}^*(u_{g-1}, r_{ij}, u_g) = \min_{k=1, \dots, K} [\lambda_k^*(u_{g-1}, r_{ij}, u_g)].$$

*Step 3:* Repeat Step 1-2 until all jobs are scheduled.

## 4.6. Computational test

In this section, three computational tests are presented. The purpose of the first test is to show the results for the problems of small or moderate size. The second test focuses the computational efficiency for the heuristic algorithm for the problems of larger size. The third test focuses on solving the scheduling problem based on real-world applications.

### 4.6.1. Computational test 1

In this test, computational results were presented by a set of randomly generated test problems, with similar characteristics to industrial data. 12 jobs are to be completed within two days. Thus, the machine capacity is set to 2880 minutes. Table 4-4 shows the data set used to generate the test problems. We consider two values of number of job clusters ( $I=3, 6$ ), two sets of level of priority ( $H=3, 5$ ), and three values of number of machines ( $K=3, 4, 5$ ). The unit processing time for the product types are 40, 45, and 50. The lot size of each job was generated using uniform[10, 15] for 3 machines, uniform[14, 19] for 4 machines, and uniform[18, 23] for 5 machines. Thus, we have a total of 12 combinations of problem parameters. For each combination, we generate 10 instances, yielding a total of 120 problems.

Table 4-4. Data Set

Factor	Values considered	Total values
Number of job clusters ( $I$ )	3, 6	2
Levels of job priority ( $H$ )	3, 5	2
Number of machines ( $K$ )	3, 4, 5	3
Number of jobs ( $\sum J_i$ )	12	1

The IP model was tested using a computer program coded in ILOG OPL language and solved with ILOG CPLEX on a Pentium IV 3.0 GHz PC. The heuristic algorithm

was coded in Compaq Visual Fortran 6.6. For evaluating the solution quality, percentage error  $e = [(\bar{S}_h - \bar{S}_{opt}) / \bar{S}_{opt}] \times 100$  is employed, where  $\bar{S}_h$  is the average setup time of the heuristic solution and  $\bar{S}_{opt}$  is the optimal average setup time obtained from the IP model. Table 4-5 lists the results. The proposed heuristic is effective and each percentage error is less than 3%. The efficiency of the models is also reported based on the average CPU time (in seconds). For the IP model, the computation time increases with increasing the number of machines, while the heuristic algorithm is able to obtain the solutions within almost instant time for every problems in this test.

Table 4-5. Summary Results

$I$	$H$	$K$	IP model		Heuristic		$e^*$ (%)
			$\bar{S}_{opt}$	Avg. run time (sec)	$\bar{S}_h$	Avg. run time (sec)	
3	3	3	93	839.41	93	0.0015	0.00
6	3	3	240	38.23	240	0.0015	0.00
3	5	3	81	133.51	81	0.0015	0.00
6	5	3	267	39.31	273	0.0015	2.25
3	3	4	93	1974.55	93	0.0015	0.00
6	3	4	186	65.61	186	0.0015	0.00
3	5	4	120	1225.81	120	0.0015	0.00
6	5	4	201	41.85	204	0.0015	1.49
3	3	5	96	1722.96	96	0.0015	0.00
6	3	5	168	247.97	171	0.0015	1.79
3	5	5	117	2110.11	120	0.0015	2.56
6	5	5	177	65.48	177	0.0015	0.00

\*:  $e = [(\bar{S}_h - \bar{S}_{opt}) / \bar{S}_{opt}] \times 100$

Using the combination of depth-first search strategy and strong branching rule showed to be powerful. For every test problems of 3 machines and 4 machines in the data set, the optimal solution was reached within the 15,000 nodes created (within 5 minutes of execution time). For every test problem of 5 machines in the data set, the

optimal solution was reached within the 17,000 nodes created (within 10 minutes of execution time).

#### 4.6.2. Computational test 2

In this test, computational results were presented by six larger-size problems, with similar characteristics to industrial data. The jobs are to be completed within two days. Thus, the machine capacity is set to 2880 minutes. We consider two values of number of job clusters ( $I=8, 10$ ), two sets of level of priority ( $H=3, 5$ ), and three values of number of machines ( $K=8, 9, 10$ ).

The IP model with the combination of depth-first search strategy and strong branching rule (IP\_DFS) was tested using a computer program coded in ILOG OPL language and solved with ILOG CPLEX on a Pentium IV 3.0 GHz PC. The heuristic algorithm was coded in Compaq Visual Fortran 6.6. For evaluating the solution quality, percentage error  $e = [(\bar{S}_h - \bar{S}_{dfs}) / \bar{S}_{dfs}] \times 100$  is employed, where  $\bar{S}_h$  is the average setup time of the heuristic solution and  $\bar{S}_{dfs}$  is the average setup time obtained from the IP\_DFS model.

For the six problems, the solution values obtained by the heuristic algorithm were compared with those obtained by IP\_DFS. According to the computational results, the heuristic algorithm outperformed IP\_DFS both in solution quality and runtime consumed.

Table 4- 6 Summary Results for larger-size problem

$I$	$H$	$K$	IP_DFS		Heuristic		$e^*$ (%)
			$\bar{S}_{dfs}$	Run time (sec)	$\bar{S}_h$	Run time (sec)	
10	3	8	660	43953.80	360	0.0015	-45.45
8	3	9	1200	45894.81	540	0.0015	-55.00
8	5	9	1560	47395.84	660	0.0015	-57.69
10	3	9	1740	51290.69	630	0.0015	-63.79
10	5	9	1320	52408.55	750	0.0015	-43.18
8	5	10	2340	61162.38	420	0.0015	-82.05

$$*: e = [(\bar{S}_h - \bar{S}_{dfs}) / \bar{S}_{dfs}] \times 100$$

#### 4.6.3. Computational test 3

In this section, we consider the following example taken from an IC assembly shop-floor in an IC manufacturing factory located in the Science-based Industrial Park at Tainan, Taiwan. For the case we investigated, there are 20 product types of TSOP2 packaging being processed on 33 parallel LOC die bonders.

This real example contains 105 wafer lots with job priority, lot size, and unit processing time, which would be die bonding under certain size of chop table, mount stage and mount head, as shown in Table 4-6. These jobs are to be completed on the 33 parallel die bonders within two days. Therefore, the machine capacity is set to 2880 minutes.

The setup time required for switching one product type to another depends on the chop table changes, mount stage and mount head changes, and parameter settings is shown in Table 4-5. The time to change chop table is 240 minutes, the time to change mount stage and mount head is 120 minutes, and parameter settings is 30 minutes in this case.



Solving the real-world ICASP by our proposed algorithm (the program codes of the algorithm are written in Compaq Visual Fortran 6.6), the sets of machine schedule are generated. The proposed algorithm takes only 0.07 CPU seconds to obtain the solution with total machine workload 87602 with setup time 6480 and processing time 81122 on 33 die bonders, as shown in Figure 4-6.



Table 4-7. The product types, processing time, and job priority code in the real-world example

<i>Job ID</i>	<i>Product type</i>	<i>Lot size</i>	<i>Unit processing time</i>	<i>job priority code</i>	<i>Job ID</i>	<i>Product type</i>	<i>Lot size</i>	<i>Unit processing time</i>	<i>Job priority code</i>
1	01	8	42	2	31	06	8	42	3
2	01	8	42	2	32	07	20	42	1
3	01	7	42	3	33	07	20	42	1
4	02	7	42	2	34	07	19	42	2
5	02	7	42	3	35	07	19	42	2
6	03	22	40	1	36	07	19	42	3
7	03	23	40	1	37	07	18	42	3
8	03	21	40	2	38	07	18	42	3
9	03	20	40	2	39	08	20	42	2
10	03	20	40	3	40	08	20	42	2
11	03	20	40	3	41	08	19	42	2
12	03	18	40	3	42	08	19	42	3
13	03	18	40	4	43	08	19	42	3
14	04	20	40	2	44	08	18	42	3
15	04	22	40	2	45	08	18	42	3
16	04	22	40	2	46	08	18	42	4
17	04	20	40	3	47	08	18	42	4
18	04	20	40	3	48	09	19	45	2
19	04	18	40	3	49	09	19	45	2
20	04	18	40	4	50	09	18	45	3
21	04	18	40	4	51	09	18	45	3
22	05	16	40	2	52	09	18	45	3
23	05	18	40	2	53	09	18	45	3
24	05	16	40	3	54	10	20	45	2
25	05	16	40	3	55	10	20	45	3
26	05	17	40	3	56	10	20	45	3
27	05	16	40	3	57	10	20	45	3
28	06	9	42	2	58	10	19	45	3
29	06	9	42	2	59	10	19	45	3
30	06	8	42	2	60	10	19	45	3

Table 4-6. The product types, processing time, and job priority code in the real-world example (continued)

<i>Job ID</i>	<i>Product type</i>	<i>Lot size</i>	<i>Unit processing time</i>	<i>Job priority code</i>	<i>Job ID</i>	<i>Product type</i>	<i>Lot size</i>	<i>Unit processing time</i>	<i>Job priority code</i>
61	10	18	45	4	84	13	20	45	3
62	10	18	45	4	85	14	20	50	1
63	11	23	40	1	86	14	20	50	1
64	11	23	40	1	87	14	19	50	2
65	11	23	40	2	88	14	19	50	3
66	11	22	40	2	89	14	19	50	3
67	11	22	40	3	90	15	20	50	2
68	11	22	40	3	91	15	20	50	2
69	11	21	40	3	92	15	20	50	3
70	11	21	40	3	93	15	20	50	4
71	12	20	40	1	94	15	20	50	4
72	12	20	40	2	95	16	18	42	4
73	12	20	40	2	96	16	18	42	4
74	12	20	40	3	97	16	17	42	5
75	12	20	40	3	98	17	15	50	3
76	12	20	40	3	99	17	15	50	4
77	12	19	40	4	100	18	12	50	4
78	12	19	40	4	101	18	12	50	4
79	13	23	45	2	102	19	16	45	3
80	13	22	45	2	103	19	16	45	4
81	13	22	45	2	104	20	7	45	5
82	13	20	45	3	105	20	7	45	5
83	13	20	45	3					

Table 4-8. Setup times required for switching one product type to another in the real-world example

<i>From</i>	<i>T<sub>0</sub></i>																				
	U	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20
U	0	270	270	150	150	150	270	150	150	150	150	150	150	150	150	150	270	150	150	150	270
01	0	0	30	270	270	270	30	270	270	270	270	270	270	270	270	270	150	390	270	270	150
02	0	30	0	270	270	270	30	270	270	270	270	270	270	270	270	270	150	390	270	270	150
03	0	390	390	0	30	30	390	30	30	30	30	150	150	150	150	150	390	150	150	150	390
04	0	390	390	30	0	30	390	30	30	30	30	150	150	150	150	150	390	150	150	150	390
05	0	390	390	30	30	0	390	30	30	30	30	150	150	150	150	150	390	150	150	150	390
06	0	150	150	270	270	270	0	270	270	270	270	270	270	270	390	390	150	390	390	390	150
07	0	390	390	30	30	30	390	0	30	30	150	150	150	150	150	150	390	150	150	150	390
08	0	390	390	30	30	30	390	30	0	30	150	150	150	150	150	150	390	150	150	150	390
09	0	390	390	30	30	30	390	30	30	0	150	150	150	150	150	150	390	150	150	150	390
10	0	390	390	30	30	30	390	30	30	30	0	150	150	150	150	150	390	150	150	150	390
11	0	390	390	30	30	30	270	30	30	30	30	0	30	30	150	150	390	150	150	150	390
12	0	390	390	30	30	30	270	30	30	30	30	30	0	30	150	150	390	150	150	150	390
13	0	390	390	30	30	30	270	30	30	30	30	30	30	0	150	150	390	150	150	150	390
14	0	270	270	30	30	30	270	30	30	30	30	30	30	30	0	30	390	150	30	30	390
15	0	270	270	30	30	30	270	30	30	30	30	30	30	30	30	0	390	150	30	30	390
16	0	30	30	270	270	270	30	270	270	270	270	270	270	270	270	270	0	270	270	270	30
17	0	270	270	30	30	30	270	30	30	30	30	30	30	30	30	30	270	0	30	30	270
18	0	270	270	30	30	30	270	30	30	30	30	30	30	30	30	30	390	150	0	30	390
19	0	270	270	30	30	30	270	30	30	30	30	30	30	30	30	30	390	150	30	0	390
20	0	30	30	270	270	270	30	270	270	270	270	270	270	270	270	270	30	270	270	270	0

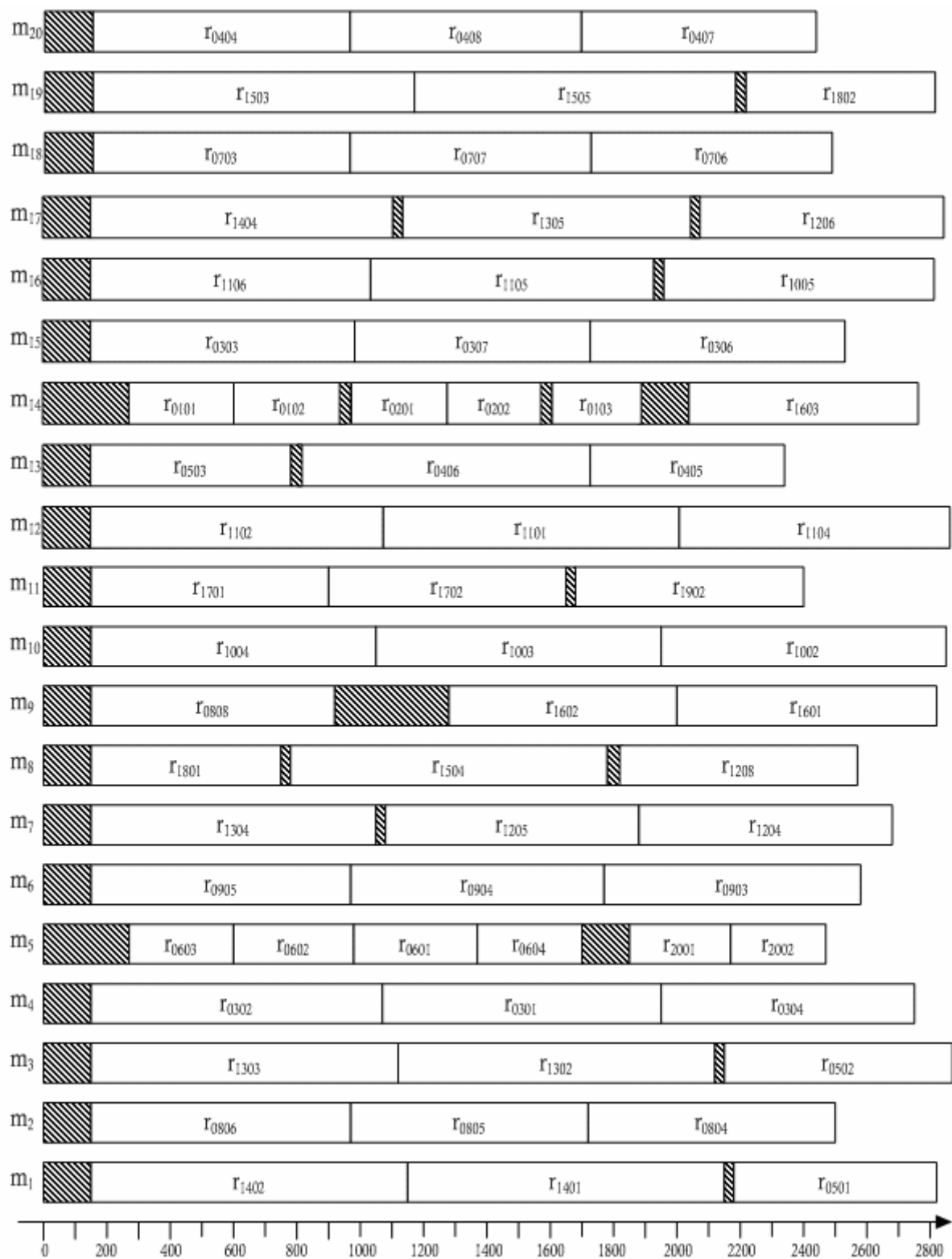


Figure 4-6. The schedule for the real-world ICASP application example.

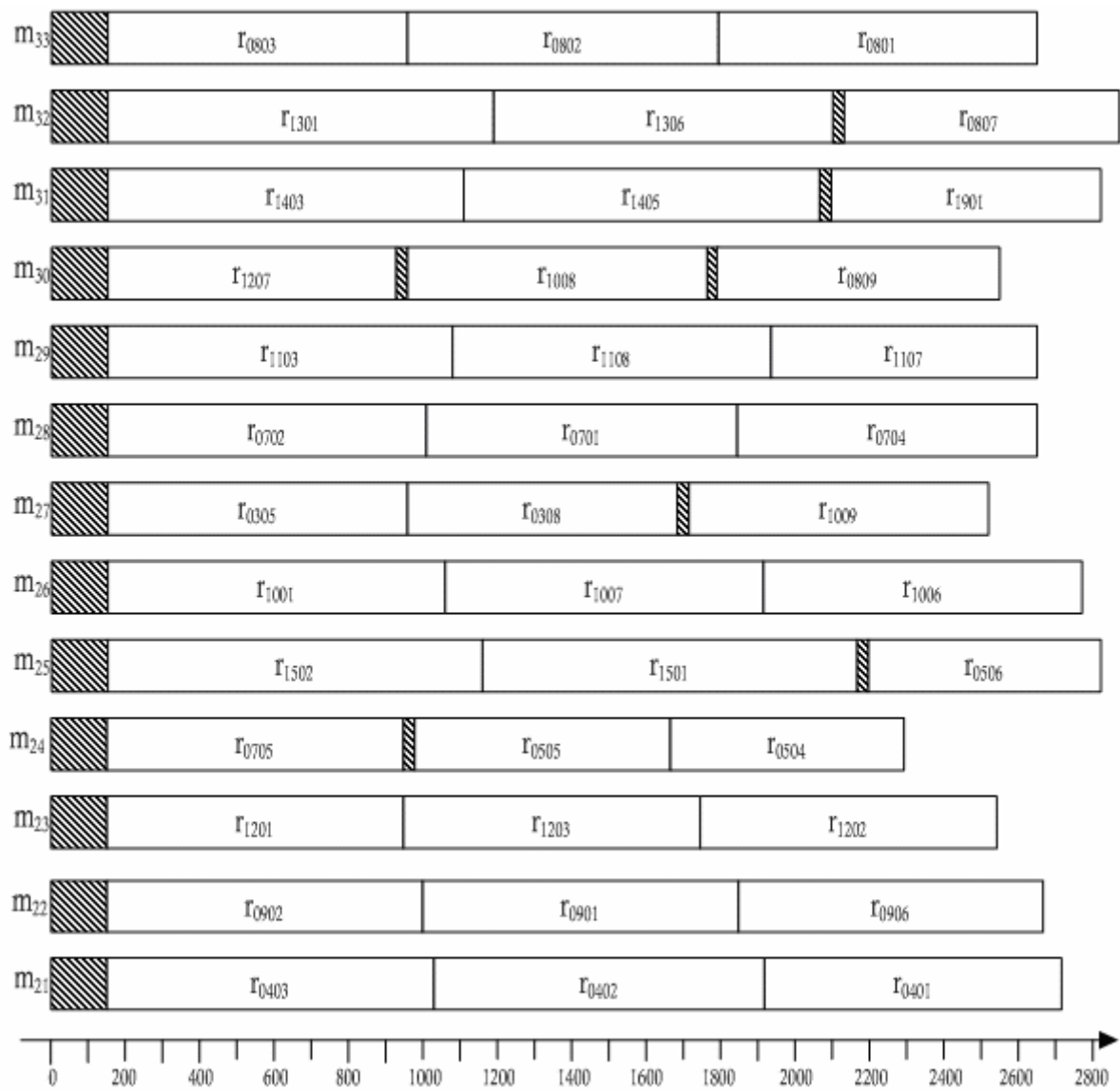


Figure 4-5. The schedule for the real-world ICASP application example (continued).

## 5. Conclusions and Future Research


### 5.1. Conclusions

Semiconductor companies must maintain high-level customer service to gain their competitive edge. Facing the environment with volatile demand, how to deliver order on time justifies the efficiency of the production planning and scheduling in semiconductor manufacturing. At the same time, minimizing production cost is the other managerial goal. Finding practical scheduling methods with the consideration of multiple processing-priorities and reducing setup costs simultaneously is a great challenge.

Wafer fabrication determines to a large extent the production plan of the whole semiconductor manufacturing due to its high complexity and long manufacturing process time. The accuracy of due-date assignment for wafer fabrication strongly influences the efficiency of the scheduling of downstream (back-end) operations. In this dissertation, we first considered the due-date assignment problem for wafer fabrication, which has many real-world applications. We modeled the due-date assignment problem for wafer fabrication under two environments. For the one with single product mix, waiting time of each product type is modeled by gamma distribution and the due dates are set to be consistent with the target on-time-delivery rate. The other is where product mix changes periodically, the contamination model is applied to tackle the effects of product mix changes and the due dates can then be set. A real-world example taken from a wafer fabrication factory is also provided to demonstrate the effectiveness and accuracy of the proposed model. The results show that the due-date assignment model provides a quite good solution.

For the back-end site, this dissertation is the first attempt to capture distinct production characteristics of the IC assembly operations in a scheduling model. The ICASP is described in detail and then formulate the ICASP as an integer programming model to minimize the total machine workload. An effective and efficient heuristic algorithm is also proposed for solving large-scale problems. From the computational tests, the performances of the proposed model and heuristic algorithm are quite satisfactory. For the problems of small or moderate size in the test problems, the proposed heuristic is effective and each percentage error is less than 3%. A real-world example taken from an IC assembly shop-floor in an IC manufacturing factory, where 105 jobs to be processed on 33 machines, is solved by the proposed algorithm to obtain the near optimal solution within 0.07 CPU seconds.

## 5.2. Future Research



There are some avenues to pursue in the future development of scheduling models for IC assembly operations. The first practical extension concerns the unrelated parallel-machine scheduling. When the capacity expansion of bottleneck-machines proceeded at different timing, the machines could be unrelated. As our integer programming model can also be used to solve the unrelated-parallel-machine ICASP, the part of this development could be to modify the proposed heuristic algorithm of Section 4.5 for solving the more general ICASP, where the machines are unrelated.

Another practical extension concerns the mixture-priority jobs in ICASP. In some cases companies have two types of orders, in which one type of orders are for specific customer and the other type of orders are for spot market. The customer orders are assigned with job priority, while the spot-market orders are not. Extension of the proposed models to the more general ICASP with the consideration of mixture-priority jobs could be our interest of research.



Finally, research can be pursued to solve the ICASP where the production system has more than one bottlenecks. A complete analysis of the relationships between those bottlenecks and the processing flow of jobs would be desirable.



## References

- [1] Ahmed, I. and Fisher, W. W. "Due date assignment, job order release, and sequencing interaction in job shop scheduling," *Decision Sciences*, 23(3), pp. 633-647, 1992.
- [2] Allahverdi, A., Gupta, J. N. D., and Aldowaisan, T. "A review of scheduling research involving setup considerations," *Omega*, 27, pp. 219-239, 1999.
- [3] Altinkemer, K. and Gavish, B., "Parallel savings based heuristics for the delivery problems," *Operations Research*, 39(3), pp. 456-469, 1991.
- [4] Blazewicz, J., Dror, M., and Weglarz, J. "Mathematical programming formulations for machine scheduling: A survey," *European Journal of Operational Research*, 51, pp. 283-300, 1991.
- [5] Chang, F. C. R. "A study of factors affecting due-date predictability in a simulated dynamic job shop," *Journal of Manufacturing System*, 13(6), pp. 393-400, 1994.
- [6] Chang, F. C. R. "Heuristics for dynamic job shop scheduling with real-time updated queueing time estimates," *International Journal of Production Research*, 35(3), pp. 651-665, 1997.
- [7] Chang, P. C. and Hsieh, J. C. "A neural networks approach for due-date assignment in a wafer fabrication factory," *International Journal of Industrial Engineering*, 10(1), pp. 55-61, 2003.
- [8] Chang, P. C., Hsieh, J. C., and Liao, T. W. "Evolving fuzzy rules for due-date assignment problem in semiconductor manufacturing factory," *Journal of Intelligent Manufacturing*, 16, pp. 549-557, 2005.
- [9] Chang, P. C. and Liao, T. W. "Combining SOM and fuzzy rule base for flow time prediction in semiconductor manufacturing factory," *Applied Soft Computing*, 6, pp. 198-206, 2006.
- [10] Cheng, T. C. E. "Optimal due-date assignment in an assembly shop," *International Journal of Operations & Production*, 14(2), pp. 31-42, 1994.
- [11] Cheng, T. C. E. and Gupta, M. "Survey of scheduling research involving due date determination decisions," *European Journal of Operational Research*, 38, pp. 156-164, 1989.
- [12] Cheng, T. C. E. and Sin, C. C. S. "A state-of-art review of parallel-machine scheduling research," *European Journal of Operational Research*, 47, pp. 271-292, 1990.

- [13] Cheng, T. C. E. and Diamond, J. E. "Scheduling two job classes on parallel machines," *IIE Transactions*, 27, pp. 689-693, 1995.
- [14] Chern, C. C. and Liu, Y. L. "Family-based scheduling rules of a sequence-dependent wafer fabrication system," *IEEE Transactions on Semiconductor Manufacturing*, 16(1), pp. 15-25, 2003.
- [15] Chiu, C., Chang, P. C., and Chiu, N. H. "A case-based expert support system for due-date assignment in a wafer fabrication factory," *Journal of Intelligent Manufacturing*, 14, 287-296, 2003.
- [16] Chou, Y. C. and Hong, I. H. "A methodology for product mix planning in semiconductor foundry manufacturing," *IEEE Transactions on Semiconductor Manufacturing*, 13(3), pp. 278-285, 2000.
- [17] Christofides, N., Mingozzi, A., and Toth, P., *Combinatorial Optimization*, John Wiley & Sons: New York, 1979.
- [18] Chung, S. H. and Huang, H. W. "The block-based cycle time estimation algorithm for wafer fabrication factories," *International Journal of Industrial Engineering*, 6(4), pp. 307-316, 1999.
- [19] Chung, S. H., Yang, M. H., and Cheng, C. M. "The design of due-date assignment model and the determination of flow time control parameter for the wafer fabrication factories," *IEEE Transaction on Components, Packaging, and Manufacturing Technology*, 20, pp. 278-287, 1997.
- [20] Clark, G. and Wright, J. "Scheduling vehicles from a central depot to a number of delivery points," *Operations Research*, 12, pp. 568-580, 1964.
- [21] Conway, R. W., Maxwell, W. L., and Miller, L. W., *Theory of Scheduling*. Massachusetts: Addison-Wesley, 1967.
- [22] Dümmler, M. A. "Analysis of the instationary behavior of a wafer fab during product mix changes," *Proceedings of the 2000 Winter Simulation Conference*, pp. 1436-1442, 2000.
- [23] Dunstall, S. and Wirth, A. "A comparison of branch-and-bound algorithms for a family scheduling problem with identical parallel machines," *European Journal of Operational Research*, 167, pp. 283-296, 2005.
- [24] Enns, S. T. "An economic approach to job shop performance analysis," *International Journal of Production Economics*, 38, pp. 117-131, 1995.
- [25] Enns, S. T. "Lead time selection and the behavior of work flow in job shops," *European Journal of Operational Research*, 109, pp. 122-136, 1998.

- [26] Freed, T. and Leachman, R. C. "Scheduling semiconductor device test operations on multihead testers," *IEEE Transactions on Semiconductor Manufacturing*, 12(4), pp. 523-530, 1999.
- [27] Golden, B. "Evaluating a sequential vehicle routing algorithm," *AIIE Transactions*, 9(2), pp. 204-208, 1977.
- [28] Gordon, V., Proth, J. M., and Chu, C., "A survey of the state-of-the-art of common due date assignment and scheduling research," *European Journal of Operational Research*, 139, pp. 1-25, 2002a.
- [29] Gordon, V., Proth, J. M., and Chu, C., "Due date assignment and scheduling: SLK, TWK, and other due date assignment models" *Production Planning and Control*, 13(2), pp. 117-132, 2002b.
- [30] Hogg, R. V. and Craig, A. T. *Introduction to Mathematical Statistics*, 4th ed., New York: Macmillan, 1978.
- [31] Hood, S. J., Bermon, S., and Barahona, F. "Capacity planning under demand uncertainty of semiconductor manufacturing," *IEEE Transactions on Semiconductor Manufacturing*, 16(2), pp. 273-280, 2003.
- [32] Hopp, W. J. and Roofsturgis, M. L. "Quoting manufacturing due dates subject to a service level constraint," *IIE Transactions*, 32, pp. 771-784, 2000.
- [33] Hsu, P.-L., Hsu, C.-I., Chang, P.-C., and Chiu, C. "Regression trees approach for flow-time prediction in wafer manufacturing processes using constraint-based genetic algorithm," *International Journal of Production Research*, 44(24), pp. 5327-5341, 2006.
- [34] Hsu, S. Y. and Sha, D. Y. "Due date assignment using artificial neural networks under different shop floor control strategies," *International Journal of Production Research*, 42(9), pp. 1727-1745, 2004.
- [35] Hung, Y. F. and Leachman, R. C. "A production planning methodology for semiconductor manufacturing based on interactive simulation and linear programming calculations," *IEEE Transaction on Semiconductor Manufacturing*, 9(2), pp. 257-269, 1996.
- [36] ILOG Ltd. ILOG OPL studio 3.6 User's Manual, ILOG S. A., France, 2002.
- [37] Kaplan, A. C. and Unal, A. T. "A probabilistic cost-based due date assignment model for job shops," *International Journal of Production Research*, 31(12), pp.2817-2834, 1993.

- [38] Lam, K. and Xing, W. "New trends in parallel machine scheduling," *International Journal of Operations & Production*, 17(3), pp. 326- , 1997.
- [39] Lawrence, S. R. "Estimating flowtimes and setting due-dates in complex production systems," *IIE Transactions*, 27, pp. 657-668, 1995.
- [40] Lee, W. C., "Inventory model involving controllable backorder rate and variable lead time demand with the mixtures of distribution," *Applied Mathematics and Computation*, 160, pp. 701-717, 2005.
- [41] Lee, Y. H. and Pinedo, M. "Scheduling jobs on parallel machines with sequence-dependent setup times," *European Journal of Operational Research*, 100, pp.464-474, 1997.
- [42] Li, C. L. and Cheng, T. C. E. "Due-date determination with resequencing," *IIE Transactions*, 31, pp. 183-188, 1999.
- [43] Little, J. D. C. "A proof for the queueing formula  $L = \lambda W$ ," *Operation Research*, 9, pp. 383-387, 1961.
- [44] Liu, C. Y. and Chang, S. C. "Scheduling flexible flow shops with sequence-dependent setup effects," *IEEE Transactions on Robotics and Automation*, 16(4), pp. 408-419, 2000.
- [45] Liu, T. H., Trappey, A. J. C. and Chan, F. W. "A scheduling system for IC packaging industry using STEP enabling technology," *IEEE Transactions on Component, Packaging, and Manufacturing Technology*, 20(4), pp. 256-267, 1997.
- [46] Liu, W., Chua, T. J., Cai, T. X., Wang, F. Y., and Yan, W. J. "Practical lot release methodology for semiconductor back-end manufacturing," *Production Planning & Control*, 16(3), pp.297-308, 2005.
- [47] Lu, S. C. H., Ramaswamy, D., and Kumar, P. R. "Efficient scheduling policies to reduce mean and variance of cycle-time in semiconductor manufacturing plants," *IEEE Transactions on Semiconductor Manufacturing*, 7(3), pp. 374-388, 1994.
- [48] Mokotoff, E. "Parallel machine scheduling problems: a survey," *Asia-Pacific Journal of Operational Research*, 18, pp. 193-242, 2001.
- [49] Omar, M. K. and Teo, S. C. "Minimizing the sum of earliness/tardiness in identical parallel machines schedule with incompatible job families: An improved MIP approach," *Applied Mathematics and Computation*, 181, pp. 1008-1017, 2006.
- [50] Ooijen, H. P. G., and Bertrand, J. W. M. "Economic due-date setting in job-shops based on routing and workload dependent flow time distribution functions," *International Journal of Production Economics*, 74, pp. 261-268, 2001.

- [51] Ouyang, L. Y. and Wu, K. S., "Mixture inventory model involving variable lead time with a service level constraint," *Computers Operations Researches*, 24(9), 875-882, 1997.
- [52] Pearn, W. L. and Chang, C. S., "An implementation of the precision index for contaminated processes," *Quality Engineering*, 11(1), pp. 101-110, 1998-1999.
- [53] Pearn, W. L., Chung, S. H., and Yang, M. H., "Minimizing the total machine workload for the wafer probing scheduling problem," *IIE Transactions*, 34, pp. 211-220, 2002.
- [54] Pearn, W. L., Chung, S. H., and Yang, M. H., "The wafer probing scheduling problem (WPSP)," *Journal of the Operational Research Society*, 53, pp. 864-874, 2002.
- [55] Pearn, W. L., Chung, S. H., Yang, M. H., and Chen, Y. H., "Algorithms for the wafer probing scheduling problem with sequence-dependent set-up time and due date restrictions," *Journal of the Operational Research Society*, 55, pp. 1194-1207, 2004.
- [56] Peter, P., and Yang, T. "Integrated facility layout and material handling system design in semiconductor fabrication facilities," *IEEE Transactions on Semiconductor Manufacturing*, 15, pp. 91-101, 2002.
- [57] Perlstein, D., Jarvis, W. H., and Mazzuchi, T. A., "Bayesian calculation of cost optimal burn-in test durations for mixed exponential populations," *Reliability Engineering & System Safety*, 72, pp. 265-273, 2001.
- [58] Potoradi, J., Boon, O. S., Mason, S. J., Fowler, J. W., and Prund, M. E. "Using simulation-based scheduling to maximize demand fulfillment in a semiconductor assembly facility," Proceedings of the 2002 Winter Simulation Conference, E. Yiicesan, C.-H. Chen, J. L. Snowdon, and J. M. Charnes, eds., pp. 1857-1861, 2002.
- [59] Quirk, M. and Serda, J. *Semiconductor Manufacturing Technology*, Prentice-Hall, Inc., New Jersey, 2001.
- [60] Raghu, T. S. and Rajendran, C. "Due-date setting methodologies based on simulated annealing – an experimental study in a real-life job shop," *International Journal of Production Research*, 33(9), pp. 2535-2554, 1995.
- [61] Rinnooy Kan, A. H. G., *Machine scheduling problems: classification, complexity and computations*. The Hague, Holland: Martinus Nijhoff, 1976.
- [62] Roman, D. B. and del Valle, A. G. "Dynamic assignment of due-dates in an assembly shop based in simulation," *International Journal of Production Research*, 34(6), pp. 1539-1554, 1996.

- [63] Schutten, J. M. J. and Leussink, R. A. M. "Parallel machine scheduling with release dates, due dates and family setup times," *International Journal of Production Economics*, 46-47, pp. 119-125, 1996.
- [64] Seidmann, A. and Smith, M. L. "Due date assignment for production systems," *Management Science*, 27(5), pp.571-581, 1981.
- [65] Sha, D. Y. and Hsu, S. Y. "Due date assignment in wafer fabrication using artificial neural networks," *The International Journal of Advanced Manufacturing Technology*, 23(9), pp. 768-775, 2004.
- [66] Sha, D. Y. and Liu, C.-H. "Development and evaluation of a tree-indexing approach to improve case-based reasoning: illustrated using the due date assignment problem," *International Journal of Production Research*, 44(15), pp. 3033-3049, 2006.
- [67] Shen, Y. and Leachman, R. C. "Stochastic wafer fabrication scheduling," *IEEE Transactions on Semiconductor Manufacturing*, 16(1), pp. 2-14, 2003.
- [68] Sivakumar, A. I. and Chong, C. S. "A simulation based analysis of cycle time distribution, and throughput in semiconductor backend manufacturing," *Computers in Industry*, 45, pp. 59-78, 2001.
- [69] Spearman, M. L., Woodruff, D. L., and Hopp, W. J. "CONWIP: a pull alternative to Kanban," *International Journal of Production Research*, 28, pp. 879-894, 1990.
- [70] Tahar, D. N., Yalaoui, F., Chu, C., and Amodeo, L. "A linear programming approach for identical parallel machine scheduling with job splitting and sequence-dependent setup times," *International Journal of Production Economics*, 99, pp. 63-73, 2006.
- [71] Tecnomatix Technologies Ltd., *eM-Plant Objects Manual*, Tecnomatix Software Company, Germany, 2000.
- [72] Tovia, F., Mason, S. J., and Ramasami, B. "A scheduling heuristic for maximizing wirebonder throughput," *IEEE Transactions on Electronics Packaging Manufacturing*, 27(2), pp. 145-150, 2004.
- [73] Uzsoy, R., Lee, C. Y., and Martin-Vega, L. A. "A review of production planning and scheduling models in the semiconductor industry, part I: system characteristics, performance evaluation and production planning," *IIE Transactions*, 24(4), pp. 47-60, 1992.
- [74] Uzsoy, R., Martin-Vege, L. A., Lee, C. Y., and Leonard, P. A. "Production scheduling algorithms for a semiconductor test facility," *IEEE Transactions on Semiconductor Manufacturing*, 4(4), pp. 270-280, 1991.

- [75] Vig, M. M., and Dooley, K. J. "Dynamic rules for due-date assignment," *International Journal of Production Research*, 29(7), pp. 1361-1377, 1991.
- [76] Vig, M. M., and Dooley, K. J. "Mixing static and dynamic flowtime estimates for due-date assignment," *Journal of Operations Management*, 11, pp. 67-79, 1993.
- [77] Wang, L. X. and Mendel, J. M. "Generating fuzzy rules by learning from examples," *IEEE Transactions on System, Man and Cybernetics*, 22(6), pp.1414-1427, 1992.
- [78] Webster, S., and Azizoglu, M. "Dynamic programming algorithms for scheduling parallel machines with family setup times," *Computers & Operations Research*, 28, pp. 127-137, 2001.
- [79] Weeks, J. K., and Fryer, J. S. "A methodology for assigning minimum cost due-dates," *Management Science*, 23(8), pp. 872-881, 1977.
- [80] Weeks, J. K. "A simulation study of predictable due-dates," *Management Science*, 25(4), pp. 363-373, 1979.
- [81] Wood, S. C. "Cost and cycle time performance of fabs based on integrated single-wafer processing," *IEEE Transactions on Semiconductor Manufacturing*, 10(1), pp. 98-111, 1997.
- [82] Xiao, H., *Introduction to Semiconductor Manufacturing Technology*. Prentice-Hall, Upper Saddle River, 2001.
- [83] Yin, X. F., Chua, T. J., Wang, F. Y., Liu, M. W., Cai, T. X., Yan, W. J., Chong, C. S., Zhu, J. P., and Lam, M. Y., "A rule-based heuristic finite capacity scheduling system for semiconductor backend assembly," *International Journal of Computer Integrated Manufacturing*, 17(8), pp. 733-749, 2004.



## Appendix

### Appendix A. Product Process Data<sup>1</sup>

Table A-1. Sequence and Processing Time for Product A (unit: minute)

Sequence	Workstation	Processing Time	Sequence	Workstation	Processing Time
1	W43	3.00	81	W26	346.00
2	W45	3.00	82	W37	3.00
3	W79	1.00	83	W04	3.00
4	W67	12.00	84	W29	346.00
5	W24	346.00	85	W37	3.00
6	W37	3.00	86	W44	2.00
7	W46	42.00	87	W35	247.00
8	W01	3.00	88	W42	3.00
9	W20	8.00	89	W81	21.00
10	W07	32.00	90	W69	10.00
61	W40	38.00	121	W46	42.00
62	W39	6.00	122	W01	3.00
63	W39	6.00	123	W22	8.00
64	W07	32.00	124	W38	77.00
65	W71	11.00	125	W07	32.00
66	W02	3.00	126	W71	11.00
67	W46	42.00	127	W02	3.00
68	W01	3.00	128	W46	42.00
69	W48	82.00	129	W01	3.00
70	W40	38.00	130	W22	8.00
71	W40	38.00	131	W38	77.00
72	W39	6.00	132	W07	32.00
73	W39	6.00	133	W71	11.00
74	W07	32.00	134	W02	3.00
75	W71	11.00	135	W46	42.00
76	W02	3.00	136	W01	3.00
77	W67	12.00	137	W22	8.00
78	W37	3.00	138	W38	77.00
79	W67	12.00	139	W07	32.00
80	W04	3.00	140	W71	11.00

<sup>1</sup> Due to the confidentiality of the product process data which is obtained from anonymous wafer

production companies, only a partial of the process data of product A is given here for reference.

Table A-1. Sequence and Processing Time for Product A (continued) (unit: minute)

Sequence	Workstation	Processing Time	Sequence	Workstation	Processing Time
141	W02	3.00	251	W59	17.00
142	W69	10.00	252	W37	3.00
143	W41	48.00	253	W21	8.00
144	W31	247.00	254	W46	42.00
145	W76	25.00	255	W57	5.00
146	W55	41.00	256	W01	3.00
147	W66	3.00	257	W52	6.00
148	W41	48.00	258	W21	8.00
149	W77	25.00	259	W08	27.00
150	W41	48.00	260	W13	120.00
191	W78	2.00	261	W37	3.00
192	W46	42.00	262	W02	3.00
193	W57	5.00	263	W08	27.00
194	W01	3.00	264	W73	13.00
195	W52	6.00	265	W02	3.00
196	W47	82.00	266	W55	41.00
197	W18	96.00	267	W37	3.00
198	W02	3.00	268	W78	2.00
199	W73	13.00	269	W63	32.00
200	W02	3.00	270	W37	3.00
201	W53	6.00	271	W62	48.00
202	W59	17.00	272	W54	36.00
203	W37	3.00	273	W37	3.00
204	W59	1.00	274	W78	2.00
205	W37	3.00	275	W46	42.00
206	W64	120.00	276	W57	5.00
207	W37	3.00	277	W01	3.00
208	W58	40.00	278	W52	6.00
209	W59	17.00	279	W47	82.00
210	W37	3.00	280	W18	96.00
211	W46	42.00	301	W73	13.00
212	W57	5.00	302	W36	346.00
213	W01	3.00	303	W82	72.00
214	W52	6.00	304	W05	24.00
215	W21	8.00	305	W83	3.00
216	W08	27.00			
217	W13	120.00			
218	W37	3.00			
219	W02	3.00			
220	W08	27.00			

## Appendix B. Workstation Data

Table B-1. Relevant data for each workstation<sup>2</sup>

Workstation Number	W01	W04	W07	W10	W13	W16	W19	W22	W25	W28
Processing Batch	1	1	2	1	2	1	1	1	6	6
Machine Number	3	2	7	1	3	2	2	2	3	5
MTBF (hr)	200	–	300	250	200	–	200	500	500	108.6
MTTR (hr)	4	–	8	4	4	–	4	4	8	12.2
MTBPM (hr)	716	–	240	330	60	–	–	168	4320	480
MTPM (hr)	4	–	1	2	6	–	–	0.5	72	24
Workstation Number	W31	W34	W37	W40	W43	W46	W49	W52	W55	W58
Processing Batch	6	6	1	4	1	1	1	1	1	1
Machine Number	3	3	5	2	1	13	2	3	4	4
MTBF (hr)	500	500	–	70	–	24	70	100	100	400
MTTR (hr)	8	8	–	6	–	1.5	2	4	5	8
MTBPM (hr)	4320	4320	–	168	–	163	162	716	96	710
MTPM (hr)	72	72	–	8	–	5	4	4	8	10
Workstation Number	W61	W64	W67	W70	W73	W76	W79	W81		
Processing Batch	1	1	2	2	2	2	1	2		
Machine Number	1	7	1	2	2	1	2	1		
MTBF (hr)	100	105	1400	1400	1400	1400	400	–		
MTTR (hr)	8	10	5.5	5.5	5.5	5.5	8	–		
MTBPM (hr)	20	10	2160	2160	2160	2160	710	–		
MTPM (hr)	5	1.5	4	4	4	4	10	–		

---

<sup>2</sup>Due to the confidentiality of the workstation data, only a partial of the workstation data is given

here for reference.