# Modeling and scheduling of multi-cluster tools in wafer fabrication system

Zhu Wang

Université de Valenciennes et du Hainaut-Cambrésis

同济大学 TONGJI UNIVERSITY

**Thesis submitted for the degree of Doctor from University of**

**VALENCIENNES and HAINAUT-CAMBRESIS**

**Specialty in Automation and Information Engineering**

**Submitted and defended by Zhu WANG.**

**November 22nd, 2017, Shanghai, CHINA**

**Doctoral School:**
Sciences Pour l'Ingénieur (SPI)

**Laboratory:**
Laboratoire d'Automatique, de Mécanique et d'Informatique Industrielles et Humaines (LAMIH)

# Modeling and Scheduling of Multi-cluster Tools in Wafer Fabrication System

## DEFENSE COMMITTEE

**President of defense committee**
LU, Zhiqiang. Professor at University of Tongji, Shanghai, China.

**Reviewers**
Kacem Imed , professeur at University of Lorraine
YIN Yaobao , professeur at Tongji University, Shanghai

**Examiners**
Chu Feng , Professor at University of Evry Val d'Essonne
YE Chunming, Professor at University of Shanghai for Science and Technology

**Invite**
Duvivier David, Professor at University of Valenciennes and Hainaut-Cambresis

**Dissertation Advisor**
Trentesaux, Damien. Professor, University of Valenciennes and Hainaut-Cambresis, Valenciennes, France.

**Dissertation Co-Advisor**
Zhou, Binghai. Professor, Tongji University, Shanghai, China

**Dissertation Co-Supervisor**
Bekrar, Abdelghani. Associate professor, University of Valenciennes and Hainaut-Cambresis, Valenciennes, France.

Université de Valenciennes et du Hainaut-Cambrésis

同濟大學 TONGJI UNIVERSITY

## Thèse de doctorat

## Pour obtenir le grade de Docteur de l'Université de

## VALENCIENNES ET DU HAINAUT-CAMBRESIS

**Spécialité Automatique et Génie Informatique**

**Présentée et soutenue par Zhu WANG.**

**Le 22/11/2017, à Shanghai, CHINE**

**Ecole doctorale :**
Sciences Pour l'Ingénieur (SPI)

**Laboratoire et équipe de recherche :**
Laboratoire d'Automatique, de Mécanique et d'Informatique Industrielles et Humaines (LAMIH)

# Modélisation et planification des outils multi-clusters dans un système de fabrication de plaquette de silicium

## JURY

**Président du jury**
LU, Zhiqiang. Professeur à Université de Tongji, Shanghai, Chine.

**Rapporteurs**
KACEM, Imed. Professeur à l'université de Lorraine, Metz, France.
YIN, Yaobao. Professeur, Tongji University, Shanghai, Chine.

**Examinateurs**
CHU, Feng. Professeur à l'université d'Evry Val d'Essonne, Evry, France.
YE, Chunming. Professeur, University of Shanghai for Science and Technology, Shanghai, Chine.

**Invité**
Duvivier, David. Professeur à l'université de Valenciennes et Hainaut Cambrésis, Valenciennes, France.

**Directeur de thèse**
TRENTESAUX, Damien. Professeur à Université de Valenciennes et du Hainaut-Cambresis, Valenciennes, France

**Co-directeur de thèse**
ZHOU, Binghai. Professeur, Tongji University, Shanghai, Chine.

**Co-encadrant**
BEKRAR, Abdelghani. Maître de conférences, Université de Valenciennes et du Hainaut-Cambresis, Valenciennes.

# ABSTRACT

Multi-cluster tool is a highly automated and costly wafer fabrication system with multi-loop coupling structure, and scheduling of such equipment directly affects the overall efficiency of semiconductor manufacturing enterprises. Multi-cluster tools scheduling problem has the features of large scale, complex wafer flow patterns, strict residency time constraints and intense resource conflict, which are significantly different from any other manufacturing system. Since the existing literatures have proved that most of the wafer fabrication systems scheduling problems are NP-hard, it's difficult to obtain the optimal solution by using exact algorithms. Thus, how to develop an efficient heuristic algorithm to solve the multi-cluster tools scheduling problem attracts considerable attention both in academia and in industry.

After reviewing the literatures, it is found that the research on the cyclic scheduling problem of multi-cluster tools rarely takes into account the characteristics of residency constraints. The scale of the object is limited to three single cluster tools, and the proposed scheduling methods are mostly mathematical programming and simple scheduling rules. For non-cyclic scheduling problem, there are only few literatures, and the optimality of the proposed algorithms are not evaluated in the literatures. Due to its complexity, the researches on scheduling of multi-cluster tools are not sufficient up to now, especially in the research domains of taking a comprehensive consideration of the features above-mentioned. Therefore, in this thesis, the multi-cluster tool is studied and our research mainly focuses on the characteristics of residency constraints, resource constraints and wafer flow patterns. Based on the descriptions of research domains, some solid models are developed for different scheduling problems and some efficient heuristic algorithms are constructed to realize the objectives.

The 1-unit cyclic production in single wafer flow pattern is the most common production method of wafer fabrication system, and it is easy to implement and control. To ensure the feasibility of schedule, this thesis uses the method of prohibited intervals to eliminate the solution space of the deadlock caused by resource constraints and residency constraints. A non-linear mixed-integer programming model with the objective of minimum fundamental period is constructed. Based on the mathematical model, a two-stage approximate-optimal scheduling algorithm is

proposed. Firstly, the feasible solution of the scheduling problem is obtained by using the bottleneck-based search method in the initial feasible solution stage. Then, based on the lower bound of the problem that is proposed in this thesis, search for the approximate-optimal solution from the feasible solutions by sliding time block. Finally, simulation experiments and analysis demonstrate the effectiveness of two-stage approximate-optimal scheduling algorithm. The experimental results show that even in the case of uneven load distribution of the equipment the proposed algorithm still obtains a satisfactory approximate-optimal solution.

In order to improve efficiency, multi-unit cyclic production is adopted for semiconductor wafer fabrication. Due to the increase of the number and variety of wafers in cycle time, the resource competition in the multi-cluster tools is more intense, thus increasing the difficulty of scheduling. In this thesis, we study the 2-unit cyclic scheduling problem of multi-cluster tools with residency constraints and put forward a chaos-based particle swarm optimization-tabu search hybrid heuristic algorithm. First, the problem domain is described and non-linear mixed integer programming model is established with objective of minimizing fundamental period of the system based on the method of prohibited intervals. Secondly, we use chaos theory and tabu list in particle swarm optimization to improve the quality of solution and the computational efficiency. Thirdly, experimental results indicate the effectiveness of the proposed model and algorithm.

With the increasing demand of ASIC, non-cyclic production in multiple wafer flow patterns are more and more adopt by semiconductor wafer fabrication enterprises. In order to enhance the productivity, we design a bottleneck-based push-pull scheduling algorithm. It starts with controlling the Takt time of bottleneck module of the multi-cluster tools, and then uses “pull” strategy for the bottleneck downstream modules while adopts “push” strategy for the bottleneck upstream modules, so as to reduce the current residency time and achieve the goal of minimum makespan. Simulation experiments and analysis are carried out to evaluate the performance of bottleneck-based push-pull algorithm. Results show the stability and efficiency of proposed algorithm.

In summary, this thesis deals with three static scheduling problems: the 1-unit cyclic scheduling problem in single wafer flow pattern, the multi-unit cyclic scheduling problem in single wafer flow pattern, and the non-cyclic scheduling problem in multi-wafer flow patterns. According to the characteristics of scheduling

problem, scheduling models are constructed, heuristic scheduling methods are developed. The research results have achieved the purpose of enhancing the performance of multi-cluster tools and improving the yield and productivity.

**Key Words:** multi-cluster tools, residency constraint, wafer flow pattern, scheduling, heuristic algorithm

# 摘 要

集束型设备群是一种多环耦合结构的半导体晶圆制造系统，自动化程度高，造价昂贵，其调度水平直接影响到半导体制造企业的整体效益。集束型设备群的调度问题具有规模庞大、晶圆流模式复杂、驻留时间约束严格、资源冲突激烈等区别于其他制造系统的调度问题的显著特征。现有文献研究证明了半导体晶圆制造系统的大多数调度问题为 NP-hard 问题，因而很难运用精确算法获得问题的最优解。如何设计高效的启发式调度算法来求解集束型设备群的调度问题已成为学术界和工程界的研究热点。

本文回顾了相关文献的研究成果后发现，针对集束型设备群的循环调度问题的研究鲜有考虑驻留约束等特征，研究对象的规模也局限在三台集束型设备以内，调度方法大多为数学规划和简单的调度规则。集束型设备群的非循环调度问题的研究成果较少，文献中没有对所提出的算法的最优性进行评价。由于其极高的复杂性，目前针对集束型设备群调度问题的研究还不很深入，尤其是在全面考虑集束型设备群调度问题特征的问题域缺乏研究成果。因此，本文以集束型设备群为研究对象，针对半导体晶圆制造特有的驻留约束、资源约束和晶圆流模式进行了问题域的研究。在此基础上，根据不同的研究对象有针对性的建立了调度模型，开发了高效的启发式调度算法实现相应的调度目标。

目前，单一晶圆流模式下的 1-级循环生产是晶圆制造系统最主要的生产模式，具有易于执行和控制的特点。为了保证调度方案的可行性，本文采用了禁止区间法来排除由资源约束和驻留约束限制引起的、可能导致集束型设备群发生死锁的解空间，构建了以最小基本周期为目标的非线性混合整数规划模型。在调度模型的基础上，本文设计了两阶段近似最优求解算法，在初始可行调度空间阶段运用基于瓶颈的搜索方法获得调度问题的可行解，然后，以本文提出的调度问题的下界为基准，通过滑动时间块在可行解中寻找近似最优调度。仿真实验和分析验证了两阶段近似最优求解算法的有效性。实验结果表明，即使是在各设备载荷分布不均匀的情况下，算法依然能够获得令人满意的近似最优解。

为了提高生产效率，在半导体晶圆制造的过程中有时会采用多级循环生产。由于单位循环时间内晶圆品种和数目的增加，集束型设备群内的资源竞争更为激烈，增加了调度的难度。本文研究了带驻留约束的集束型设备群的 2-级循环调度问题，并提出了一种基于混沌理论的粒子群-禁忌搜索混合启发式调度算法。首先，本文对该问题域进行了描述，建立了以循环时间最短为目标的基于禁止区间

法的非线性混合整数规划模型；然后，将混沌理论和禁忌表融入粒子群算法，以提高解的质量和计算效率；最后，仿真实验和分析验证了调度模型和算法的有效性。

随着专用集成电路需求的增加，多种晶圆流模式下的非循环生产越来越多的被半导体晶圆制造企业采用。为了提升集束型设备群在多种晶圆流模式下进行非循环生产的效率，本文设计了一种基于瓶颈的推拉结合式调度算法，从控制集束型设备群瓶颈的生产节拍入手，通过对瓶颈上游模块采用“拉”式策略而对瓶颈下游模块采用“推”式策略的调度优化方法，达到缩短晶圆在集束型设备群的实际驻留时间的目的，最终实现了总加工完成时间最短的调度目标。通过仿真实验和分析，验证了算法的稳定性和高效性。

本文的研究内容主要包括了单一晶圆流模式下的 1-级循环调度、单一晶圆流模式下的多级循环调度和多种晶圆流模式下的非循环调度这三个静态调度问题。针对调度问题的特点构建了调度模型，开发了启发式调度方法。研究成果达到提升集束型设备群性能、提高晶圆的良品率和生产效率的目的。

**关键词**：集束型设备群，驻留约束，晶圆流模式，调度，启发式算法

# Acknowledgement

How time flies! It has been more than 4 years since the first enrollment in the spring of 2013, and every details flashback vividly in my mind. Now, no matter laughter or tears are all gone, because I am about to graduate!

I want to extend my sincere thanks to Prof. ZHOU Binghai, who is my Chinese supervisor. Prof. ZHOU is a talented, frankly and upright professor, his merits have had a profound impact on me. Prof. ZHOU is diligent and rigorous in teaching, he teaches me how to select topic, how to do research, how to write an article, and even how to do submission. It can be said that my paper also condoned his efforts.

I would like to express my heartfelt gratitude to my French supervisor, Professor Damien TRENTESAUX, for his international vision, innovative thinking, meticulous academic style and profound knowledge. Although I only study in France for one year, Professor TRENTESAUX still invested so many efforts in helping me with my work. I so much miss and appreciate the inspiring seminars, your motivating words, and your sincere suggestions on my career.

In addition, I also want to thanks Associate Professor Abdelghani BEKRAR for his participation in this work, which has brought me a broader view of my research subject. Dr. BEKRAR is intelligent, knowledgeable and warm-hearted. He was always by my side when I encountered troubles, and encouraged me to move on and be a better researcher.

To all the teachers of the Institute of Industrial Engineering at the School of Mechanical and Energy Engineering at Tongji University for helping and caring during my studies! Thanks to my classmates and friends, they are Ph.D. QI Faqun, Ph.D. CHENG Guoqing, Ph.D. GAO Zhongshun, Ph.D. LIU Xiaobin, SUN Chao, CHEN Jinxiang, SHAO Jianyi, and ZHANG Gangzhi.

To all of the colleagues of the LAMIH at UVHC, they are Ph.D. JIMENEZ GORDILLO José-Fernando, Ph.D. PIRES Sandro, Ph.D. SAHLI Zahir, Ph.D. RAHIMI Ali, Dr. KADRI Farid, Ph.D. HAMIEH Ahmed, Ph.D. GHAZI Nawal, Assistant Prof. CHAABANE Sondès, Madame AUREGGI Corinne, Prof. BERGER Tierry, Prof. SALLEZ Yves, Prof. SENECHAL Olivier; and thanks to Associate Prof.

XU Weijiang at ENSIAME. Thanks for participate in this work, and thanks for made France feel like home to me!

Finally yet importantly, I sincerely thank my parents for your full support and understanding, family is my strongest backing. I hope that one day I could be as perfect as you are.

Thanks for all my loved ones and friends who care about me and care for me!

November 2017

# CONTENTS

**ABSTRACT** ..... **I**

**摘 要** ..... **IV**

**ACKNOWLEDGEMENT** ..... **VI**

**CONTENTS** ..... **VIII**

**LIST OF FIGURES** ..... **11**

**LIST OF TABLES** ..... **13**

**CHAPTER 1 INTRODUCTION** ..... **1**

1.1 BACKGROUNDS ..... 1

*1.1.1 Wafer fabrication system and its characteristics* ..... *2*

*1.1.2 Scheduling of multi-cluster tools* ..... *5*

1.2 LITERATURE REVIEW ..... 9

*1.2.1 Research on 1-unit cyclic scheduling problem with residency constraints* ..... *10*

*1.2.2 Research on multi-cluster tool multi-unit cyclic scheduling problem with residency constraints* ..... *12*

*1.2.3 Research on non-cyclic scheduling problem with residency constraints under multi-wafer flow patterns* ..... *14*

*1.2.4 Research on scheduling algorithms* ..... *15*

*1.2.5 Summary* ..... *20*

*1.2.6 Scientific issue* ..... *20*

1.3 SIGNIFICANCE ..... 22

1.4 RESEARCH CONTENT ..... 23

1.5 THESIS OUTLINE ..... 24

**CHAPTER 2 THE STRUCTURE AND CHARACTERISTICS OF MULTI-CLUSTER TOOLS** ..... **27**

2.1 ARCHITECTURE OF MULTI-CLUSTER TOOLS ..... 27

2.2 CHARACTERISTICS OF MULTI-CLUSTER TOOLS ..... 30

*2.2.1 Residency constraints* ..... *30*

*2.2.2 Resource constraints* ..... *32*

2.3 WAFER FLOW PATTERN ..... 32

2.4 SUMMARY ..... 33

**CHAPTER 3 RESEARCH ON ONE-UNIT CYCLIC SCHEDULING PROBLEM ..... 35**

3.1 PROBLEM DESCRIPTION ..... 35

3.2 AN MPI-BASED NON-LINEAR MIXED-INTEGER PROGRAMMING MODEL ..... 37

*3.2.1 Notations and variables ..... 37*

*3.2.2 Objective function ..... 39*

*3.2.3 Calculate the time at which the wafer leaves each PM ..... 39*

*3.2.4 Machine constraints ..... 41*

*3.2.5 TM constraints ..... 42*

*3.2.6 Residency constraints ..... 46*

3.3 LOWER-BOUND OF 1-UNIT CYCLIC SCHEDULING PROBLEM ..... 47

3.4 MPI-NLMIP-BASED TWO-STAGE APPROXIMATE-OPTIMAL SCHEDULING ALGORITHM ..... 50

*3.4.1 Core idea and process of MNB algorithm ..... 50*

*3.4.2 Steps of MNB Algorithm ..... 51*

*3.4.3 TMs scheduling ..... 54*

3.5 SIMULATION AND EXPERIMENTAL ANALYSIS ..... 55

*3.5.1 CPU time ..... 55*

*3.5.2 Performance analysis ..... 58*

*3.5.3 Case study ..... 60*

3.6 SUMMARY ..... 62

**CHAPTER 4 RESEARCH ON MULTI-UNIT CYCLIC SCHEDULING PROBLEM ..... 65**

4.1 PROBLEM DESCRIPTION ..... 65

4.2 A NON-LINEAR MIXED-INTEGER PROGRAMMING MODEL ..... 67

*4.2.1 Notations and variables ..... 67*

*4.2.2 Objective function ..... 69*

*4.2.3 Calculate the leaving time of wafer A and B on each PM ..... 70*

*4.2.4 Machine constraints ..... 71*

*4.2.5 TMs constraints ..... 73*

*4.2.6 Residency constraints ..... 77*

*4.2.7 Complexity analysis of Proposed NLMIP model ..... 78*

4.3 CASE STUDY ..... 78

4.4 A CHAOS-BASED HYBRID PSO-TS HEURISTIC ALGORITHM ..... 81

*4.4.1 Basic particle swarm optimization ..... 82*

*4.4.2 Chaotic search technology* ..... *82*

*4.4.3 Tabu list of Tabu search* ..... *83*

*4.4.4 The core idea and process of chaos-based hybrid PSO-TS algorithm* ..... *83*

*4.4.5 Algorithm design* ..... *85*

4.5 SIMULATION AND EXPERIMENTAL ANALYSIS ..... 89

*4.5.1 CPU time* ..... *89*

*4.5.2 Performance analysis* ..... *91*

4.6 SUMMARY ..... 95

**CHAPTER 5 RESEARCH ON NON-CYCLIC SCHEDULING PROBLEM ..... 97**

5.1 PROBLEM DESCRIPTION ..... 97

5.2 A NON-LINEAR PROGRAMMING MODEL ..... 99

*5.2.1 Notations and variables* ..... *99*

*5.2.2 Mathematical model* ..... *101*

5.3 LOWER-BOUND OF THE NON-CYCLIC SCHEDULING PROBLEM ..... 103

5.4 BOTTLENECK-BASED PUSH-PULL SCHEDULING ALGORITHM ..... 105

*5.4.1 Core idea and process of algorithm* ..... *105*

*5.4.2 Steps of BP Algorithm* ..... *107*

5.5 SIMULATION AND EXPERIMENTAL ANALYSIS ..... 111

*5.5.1 CPU time* ..... *113*

*5.5.2 Performance analysis* ..... *114*

*5.5.3 ANOVA* ..... *118*

5.6 SUMMARY ..... 118

**CHAPTER 6 CONCLUSIONS AND FUTURE WORKS ..... 123**

6.1 CONCLUSIONS ..... 123

6.2 INNOVATION ..... 125

6.3 FUTURE WORKS ..... 126

**REFERENCES ..... 127**

**APPENDIX A MPI-NLMIP MODEL-BASED TWO-STAGE APPROXIMATE-OPTIMAL SCHEDULING ALGORITHM ..... 135**

**APPENDIX B THE FLOW CHART OF THE CHAOS-BASED HYBRID PSO-TS HEURISTIC ALGORITHM ..... 137**

**RESUME ..... 139**

# List of Figures

Figure 1.1 Global and regional trend of monthly semiconductor product sales data (3 months moving average) (Data origin: WSTS) ..... 1
Figure 1.2 Semiconductor IC product manufacturing process flow ..... 3
Figure 1.3 A cluster tool with 8-chambers and a multi-cluster tool (Pictures origin: MVSystem LLC & Brooks Automation) ..... 5
Figure 1.4 A simplified event graph for simulation of a cluster tool (Decision-Moving-Done Cycle) ..... 10
Figure 1.5 Structural chart of this thesis ..... 26
Figure 2.1 A schematic of multi-cluster tools ..... 27
Figure 2.2 Wafer resides on parallel machines ..... 31
Figure 2.3 Wafer resides on efficient processing module ..... 31
Figure 2.4 Schematic diagrams of different wafer flows ..... 33
Figure 3.1 Schematic view of multi-cluster tool and single wafer flow ..... 35
Figure 3.2 Schematic view of thresholds division for parameters in $S_{ij}$ ..... 39
Figure 3.3 Three cases that may lead to TM resource competition when wafer p and q are in the PMs ..... 43
Figure 3.4 Three cases that may lead to TM resource competition when wafer p and q are in the PM and BM, respectively ..... 45
Figure 3.5 Diagrammatic sketch of two wafers on BMs ..... 46
Figure 3.6 Flow chart of MNB heuristic algorithm ..... 52
Figure 3.7 CPU time spend on solving MPI-NLMIP Model with CPLEX ..... 56
Figure 3.8 CPU time spend on scheduling multi-cluster tools with 2 to 30 clusters with MNB algorithm ..... 57
Figure 3.9 Comparison of CPU time spend on solving MPI-NLMIP Model with MNB algorithm and with CPLEX ..... 58
Figure 3.10 Schematic views of three-cluster tools and wafer flow ..... 60
Figure 3.11 The Gantt chart of schedule obtained by MNB algorithm before "Verification and Improvement" step ..... 63
Figure 3.12 The Gantt chart of final schedule obtained by MNB algorithm ..... 64
Figure 4.1 Schematic view of 2-degree cyclic production ..... 65

Figure 4.2 Schematic views of three-cluster tools and wafer flow of 2-degree cyclic production ..... 79
Figure 4.3 Gantt chart of schedule obtained by CPLEX ..... 81
Figure 4.4 The basic flow of Chaos-based Hybrid PSO-TS heuristic algorithm ..... 84
Figure 4.5 Influence of number of cluster tools on CPU time ..... 90
Figure 4.6 Influence of number of BMs on CPU time ..... 90
Figure 4.7 Comparison of PSO and Chaos-based hybrid PSO-TS algorithm in aspect of CPU time ..... 91
Figure 5.1 Schematic view of wafer flow pattern, bottleneck PM, fore-bottleneck PM and post-bottleneck PM of wafer $w$ ..... 101
Figure 5.2 Flow chart of BP algorithm ..... 106
Figure 5.3 Relationship between CPU time of BP algorithm and number of wafer ..... 114
Figure 5.4 Relationship between CPU time of BP algorithm and number of wafer types ..... 114
Figure 5.5 Relationship of wafer type and CPU time of BP algorithm ..... 115
Figure 5.6 Influence of number of TMs on BP algorithm ..... 116
Figure 5.7 Comparison of BP algorithm and Pull strategy ..... 117
Figure 5.8 Comparison of BP algorithm and lower bound of Makespan ..... 118

# List of Tables

Table 1.1 Classification of Multi-cluster tools Scheduling.....6
Table 3.1 The impact of relax constraints on MPI-NLMIP Model in aspect of CPU time and optimal FP .....48
Table 3.2 Comparison of NPI-NLMIP Model and R-MPI-NLMIP Model on performance .....49
Table 3.3 List of parameters for MPI-based non-linear MIP Model verification experiment.....56
Table 3.4 Simulation data for MNB algorithm performance analysis experiment .....59
Table 3.5 MNB algorithm performance analysis: compared to lower bound of 1-unit cyclic scheduling problem.....60
Table 3.6 Simulation data and the operation results of MNB algorithm before " Check and Improve" step .....61
Table 3.7 Simulation data and final operation results of MNB algorithm.....61
Table 4.1 Schedule of three-cluster tools case obtained by CPLEX.....80
Table 4.2 Influence of wafer flow and structure of multi-cluster tools on solving NLMIP Model with CPLEX .....93
Table 4.3 Comparison of PSO and Chaos-based hybrid PSO-TS algorithm in aspect of the quality of the solution .....94
Table 5.1 Parameters related to the CPU time of BP algorithm.....113
Table 5.2 Parameters for impact of structure of multi-cluster tools on CPU time experiment.....115
Table 5.3 Results comparison of BP algorithm, Pull strategy and lower bound of non-cyclic scheduling problem .....117
Table 5.4 Results of one-way ANOVA .....120
Table 5.5 Results of two-way ANOVA .....121

# Chapter 1 **Introduction**

## 1.1 Backgrounds

In recent years, novel techniques that rely on the level of integrated circuits (IC) manufacturing technology have made rapid development, e.g. industrial internet of things (IIoT), artificial intelligence (AI), virtual reality (VR), cloud computing, and so on. In pace with the maturity of novel techniques applications, a substantial increase in demand for semiconductor products emerges [1]-[4]. According to the data released by US Semiconductor Industry Association (SIA) (see Figure 1.1), global semiconductor sales reached 338.9 billion US dollars in 2016, an annual growth rate of 1.1% [5]. The semiconductor industry is booming. As a strategic industry, the technique level of semiconductor manufacturing industry is related to the national information security and national economic development, and its development has become an important criteria measure of a country's comprehensive national strength [6]-[8].

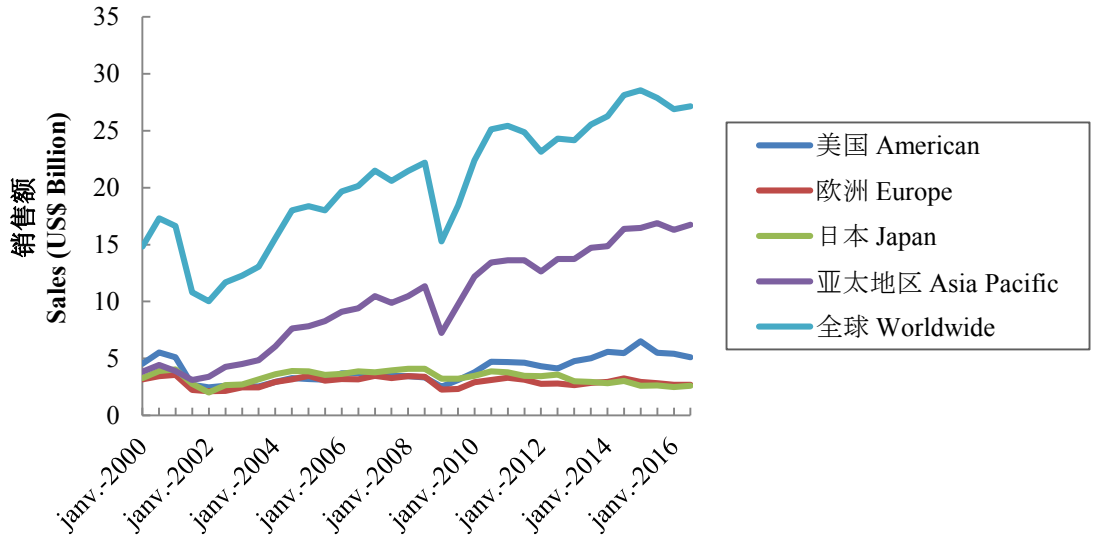



Figure 1.1 Global and regional trend of monthly semiconductor product sales data (3 months moving average) (Data origin: WSTS)

Wafer fabrication system is the most complex and expensive part of the semiconductor manufacturing process, scheduling of such system is of significant influence to economic efficiency. In general, for every 1% reduction in the cycle time,

the annual income increased by tens of millions of US dollars. Due to the huge capacity of electronic product manufacturing, China has leapt to the first place of 2016-semiconductor products sales growth ranking by an annual growth rate of 9.2%. However, the standard of production management does not match the industrial scale. As the development of scheduling technology is lagging behind, low production efficiency, low utilization of equipment, and low yield of products have always plagued the sustainable development of enterprises. Thus, advanced scheduling theory to guide the production is very necessary.

Multi-cluster tool is a new type of wafer fabrication system that widely used in 300mm wafer fabrication. Different from other manufacturing systems, multi-cluster tools has the features of large-scale, complicated wafer flow patterns, strict residency time constraints and intense resource conflict. Therefore, multi-cluster tools scheduling problems are quite complex. It is neither a typical Job shop scheduling problem, nor a Flow shop scheduling problem. The traditional flow shop, job shop, or the hybrid method of the two is no longer suitable for the scheduling of multi-cluster tools [9][10]. Currently, most of the researches on multi-cluster tools scheduling problems are concerned about performance analysis and small-scale problem. Due to the high complexity, scheduling of multi-cluster tools under various wafer flow patterns, especially large-scale multi-cluster tools scheduling problems are still very lacking.

Based on the above discussion, this thesis attempts to establish a model for the multi-cluster tools scheduling problems under various wafer flow patterns so that they can describe the characteristics of such problems. Furthermore, we try to explore targeted and efficient heuristic scheduling algorithms to enhance production efficiency and international market competitiveness of wafer fabrication enterprises.

### 1.1.1 Wafer fabrication system and its characteristics

As shown in Figure 1.2, the IC product manufacturing process consists of five main aspects: silicon wafer production, wafer fabrication, wafer probe, assembly or packaging, and final test [11][12]。Among them, wafer fabrication is the core of IC product manufacturing process for complex technology, capital-intensive and high value-added features [13][14]. The quality of IC product depends entirely on wafer fabrication process.

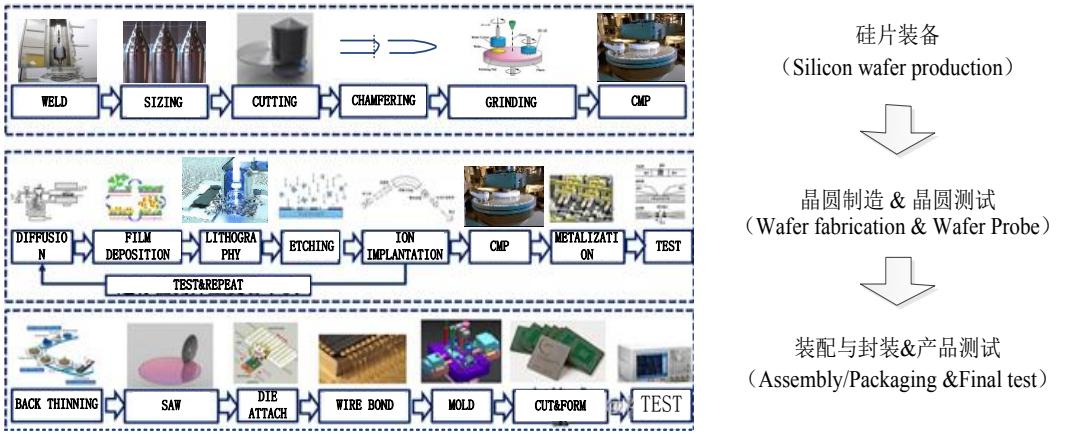



Figure 1.2 Semiconductor IC product manufacturing process flow

Wafer refers to a thin slice of silicon semiconductor material used in electronics for the fabrication of IC products. The wafer serves as the substrate for microelectronics devices built in and over the wafer and undergoes varieties of fabrication processes, so that it becomes an IC product with specific electrical function. For example, the chip is cut from a wafer [15]. Currently, the most common diameters of wafers are 200mm and 300mm, and the maximum wafer diameter is 450mm. The larger the wafer size, the more the number of chips that can be obtained by cutting a single wafer, the more complex the fabrication process. The wafer is processed layer-by-layer with a series of processes, such as oxidation, deposition, photolithography, etching, iron implantation, metallization, chemical mechanical polishing, and cleaning [16]. Ultimately, layers of circuit system are formed on a wafer. In general, wafer fabrication process has the following characteristics [17]-[19]:

(1) The fabrication process is complex and the production cycle is long. Generally, a wafer contains 15-30 layers of circuits; each layer requires 20-40 processes. Thus, a wafer need to go through more than 300 processes in total, the production cycle lasts for 3 months.

(2) High automation. In order to ensure the quality of wafer, the wafer fabrication environment is extremely clean and confined. In such circumstance, precise and automated machining equipment are widely adopted to reduce the manual intervention, which may cause air pollution. Therefore, wafer fabrication system is highly automated. [20]

(3) Multiple wafer types and large production amount. With increasing demand for Application Specific Integrated Circuits (ASICs), new products continue to emerge. Nowadays, wafer fabrication systems are capable of processing

several to dozens of different products at the same time.

(4) Fabrication environment is unstable. Because of large amount of physical and chemical reactions are involved in wafer fabrication process, the wafer fabrication environment is unstable, and thus the quality of wafer is affected by multiple factors. The precision of the wafer fabrication process is extremely high, in case of product quality defects that may be aroused by slight deviation.

(5) Equipment is of high value. Wafer fabrication integrated many key processes and related equipment resources, including lithography machine (the price of each lithography machine nearly US$ 100 million), automatic material handling robots and other bottleneck equipment. Therefore, wafer fabrication system is expensive. For example, the initial investment of a 300mm Fab is about US$ 3 billion, among which, more than 75% of the cost is spent on equipment purchase [21][22].

Due to the above characteristics, wafer fabrication system has become one of the most complex manufacturing systems. Previously, wafers were fabricated using separate processing unit, but this over-dispersed device structure is not conducive to improve productivity and yield. With the development of technique, wafer fabrication system is constantly improving. Over the past two decades, combination equipment, which is called cluster tool, is widely used in the 200mm wafer fab [23]-[25]. As shown in Figure 1.3, the cluster tool combines multiple sets of processing modules and material handling systems; it provides a flexible and efficient environment for wafer fabrication [26]. In recent years, a new integrated, automated and multi-loop coupling structured wafer fabrication system, the multi-cluster tools, has emerged. It is usually composed of two or more single cluster tools connected through buffer modules. A multi-cluster tool includes a number of robot transport modules. In order to meet the requirements for cleanliness in wafer fabrication, the multi-cluster tool is equipped with cassette module to isolate the internal vacuum environment from the outside.

Compared with the single cluster tool, multi-cluster tool integrates the previous loosely coupled discrete wafer processing flow into a direct correlated and tightly coupled discrete processing flow, achieving the purpose of improving the degree of automation and cleanliness of wafer fabrication system [27]. Besides, the multi-cluster tool is capable of integrates the required device modules according to the needs of wafer fabrication process. Therefore, the flexibility of multi-cluster tool is better than

that of single cluster tool. At present, multi-cluster tool is mainly used for 300mm wafer fabrication process.

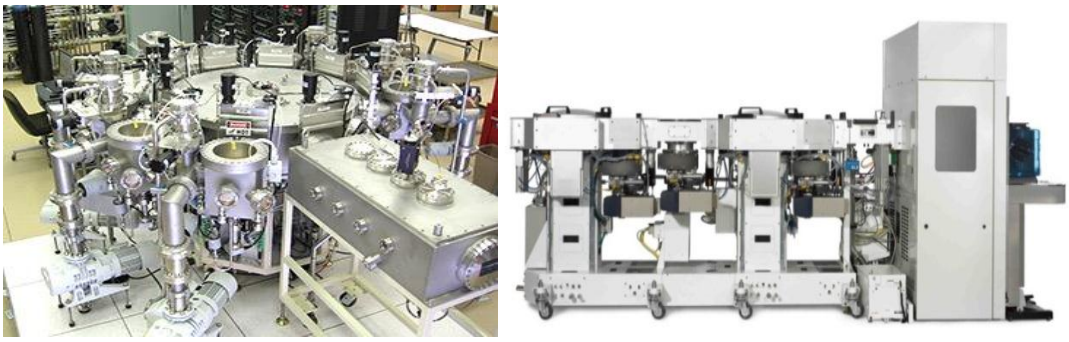


Figure 1.3 A cluster tool with 8-chambers and a multi-cluster tool (Pictures origin: MVSystem LLC & Brooks Automation)

In this thesis, we will take multi-cluster tools as object and study the scheduling problem of multi-cluster tools.

### 1.1.2 Scheduling of multi-cluster tools

Scheduling of multi-cluster tools is a model and data-based optimization and decision-making process [28][29]。In wafer fabrication process, robot transport modules and processing modules are finite resources; thus, the wafer-to-resource competition often occurs. How to arrange the sequence and time of the robot moves under the premise of satisfying a series of constraints. In other words, how to allocate the finite resources to wafers over a given period such that wafers can transport between the cassette module, processing module and buffer module, achieving a specific scheduling objectives. The above decision-making process is scheduling of multi-cluster tools.

As a new wafer fabrication system, the scheduling optimization problem of the multi-cluster tool has many new features, and the complexity of the problem is higher than ever [30]. First, there are coupling and dependency effects between single cluster tools, the effect will lead to a chain reaction. That is, if the wafer is congested or deadlocked in a cluster tool, this congestion or deadlock will be transmitted from one cluster tool to another through a buffer module, resulting in congestion or deadlock of the entire multi-cluster tool. Secondly, the capacities of cluster tools are uneven. If the gap of capacity between the two connected cluster tools is large, congestion will happen in the cluster tool with smaller capacity, and result in low utilization of the cluster tool with larger capacity due to insufficient number of wafers, and thereby reducing the output of the multi-cluster tool. In order to optimize the productivity, all

of the cluster tools in a multi-cluster tool must coordinate their operations. Third, the unstable environment increases the difficulty of scheduling. In order to avoid wafers scrapping caused by over-processing or insufficient-processing, residency constraints widely exists in wafer fabrication process. Residency constraints may restrict both the processing module and the transport module. It ensures that the current processing time of the wafer is within a reasonable range, while it also increases the difficulty of multi-cluster tool scheduling.

The scheduling of multi-cluster tools can be divided into various types from the aspects of decision-making dimension, scheduling mode, wafer flow pattern and relevant constraints (see Table 1.1).

Table 1.1 Classification of Multi-cluster tools Scheduling

| Factors | Classification of multi-cluster tool scheduling |
|---|---|
| Decision-making dimension | Robot-dominant, Process-dominant |
| Scheduling mode | Cyclic scheduling, Non-cyclic scheduling |
| Wafer flow pattern | Single wafer type, Multi-wafer types |
| Relevant constraints | Reentrant, Resource constraint, Residency constraint |

From the view of decision-making, the scheduling of multi-cluster tool is classified as robot-dominant type and process-dominant type. When the robot is dominant, material handling robots are always busy, and the Takt time is determined by the operation time of the robot. Under this circumstance, the specific scheduling objective is achieved by optimizing the sequence of robot moves. Robot-dominated situations usually occur in multi-cluster tools where the robot is tightly constrained, such as the robot handling, loading, and unloading time is long, or a robot is used for transporting wafers between multiple processing modules. On the contrary, if the process is dominant, the robot has to wait besides the processing module until the wafer process is completed; thus, the processing module determines the Takt time. It is possible to achieve the purpose of enhancing the productivity by optimizing the sequence of wafers or increasing the utilization of processing modules. This case usually occur in a multi-cluster tool where the processing module is tightly constrained, such as processing time is relatively long, the number of processing modules in each cluster tool is small, or time required for each robot move is short.

This thesis will focus on the scheduling of robot-dominate multi-cluster tools, which is an important issue in multi-cluster tools scheduling and the most direct way to improve productivity.

From the aspect of the scheduling mode, the scheduling of the multi-cluster tool can be divided into cyclic scheduling and non-cyclic scheduling. Cyclic production is the most common mode of production for wafer fabrication systems, especially in mass production [31]. It is classified as 1-unit cyclic production and k-unit cyclic production. For the sake of convenience, we define the following terms.

**Definition 1.1** [32]*: 1-unit cycle includes a series of robot moves, during which exact one wafer enters the multi-cluster tool and exact one wafer leaves the multi-cluster tool.*

**Definition 1.2** [33][34]*: 1-unit cycle time is the shortest time required for complete 1-unit cycle. It is also known as the Fundamental Period (FP) in multi-cluster tools scheduling problem.*

**Definition 1.3***: Such a robot moves sequence is referred to as an optimal cyclic schedule if the sequence of robot moves is feasible and is the minimum FP in cyclic production.*

**Definition 1.4** [35]*: k-unit cycle refers to a series of robot moves, in a k-unit cycle, exactly k pieces of wafer enter and leave the multi-cluster tools; meanwhile, each processing module in the multi-cluster tool is loaded for k times. When the robots complete above moves, the multi-cluster tool returns to the initial state.*

**Definition 1.5***: k-unit cycle time is the shortest time required for multi-cluster tools to perform a k-unit cycle.*

With cassette modules, the multi-cluster tools are able to continuously load and unload wafers in clean vacuum environment. Therefore, for most of time, multi-cluster tools stay in steady, i.e., steady state; scheduling of multi-cluster tools mainly refers to scheduling in steady state. The scheduling of the multi-cluster tool in transit state usually involves maintenance, repair, breakdown and other issues. This thesis considers the cyclic scheduling problem of multi-cluster tools. This is the most common scheduling problem in in steady state. The relevant data can be obtained in detail, which is suitable for model-based optimal or sub-optimal scheduling. Non-cyclic scheduling problem of multi-cluster tools is addressed in this thesis, too. It is a brand new issue for the study of multi-cluster tools scheduling problem in steady state. High efficient scheduling algorithms are in great need to be developed.

From the perspective of the wafer flow pattern, the scheduling of the multi-cluster tools can be divided into multi-cluster tools scheduling problem under single wafer flow pattern and that under multi-wafer flow patterns. First, for clarity, we define the following terms:

**Definition 1.6** [36]*: For a lot of wafers, the Makespan is the length of time since the first wafer enters the multi-cluster tool to the last wafer leaves the multi-cluster tool.*

In the wafer fabrication system, wafers flow from one multi-cluster tools to another in lots (or batches) according to predetermined processes. Typically, one lot of wafers consists of 25 to 50 chips. In order to protect the circuit layer from damage, the wafer is contained in a special turnover container. Wafers in one lot are normally contained in the same turnover container and transport to the cassette module of multi-cluster tool. When all of the wafers are processed, they will be packed in the turnover container again and then transport to next process along with the turnover container. Single wafer flow pattern means wafers within its brew have the same wafer flow. Single wafer flow pattern includes following two cases: 1) wafers are identical; 2) wafers are not identical but have the same processing route. Under the single wafer flow pattern, we set minimum FP as the objective of 1-unit cyclic scheduling problem and K-unit cyclic scheduling problem. Multi-wafer flow patterns means the processing route of wafers in a lot is not identical, the sequence of wafers in different lots are not same, too. Under this circumstance, we take minimum Makespan as objective of non-cyclic scheduling problem. By optimizing the sequence of robot moves, the objective can be achieved. As wafer fabrication process is very complex, the utilization of resources varies under different wafer flow patterns in the multi-cluster tool. In this thesis, we will study three typical scheduling problems: 1-unit cyclic scheduling problem with single wafer type, k-unit cyclic scheduling problem under single wafer flow pattern and non-cyclic scheduling problem under multi-wafer flow patterns.

From the view of the relevant constraints, the multi-cluster tool scheduling involves re-entrant, resource constraints, residency constraints and so on. Re-entrance in wafer fabrication is unique. Re-entrance means that wafer repeats enter the same processing module for the same processing, and the nuances of the wafer reentrant path affect the scheduling of the entire multi-cluster tools. Current scheduling researches usually use graph theory to model a single kind of reentrant path.

Therefore, it is difficult to establish a generic and abstract mathematical model to describe the scheduling of multi-cluster tool considering wafer reentrant. Resource constraints are the status that different wafers wait for the same resource in a multi-cluster tool, which leads to resource competition in multi-cluster tool. The residency constraint is a constraint that strictly controls the residency time of the wafer within the processing module in order to prevent wafer from over-processing. Resource constraints and residency constraints are prevalent in the wafer fabrication system and wafer fabrication process. They are important factors that affecting the scheduling of multi-cluster tool. The influence of resource constraints and residency constraints are considered in this thesis, which makes the research domain more practical. Meanwhile, the proposed schedule is more conductive to enhance the utilization of equipment, the wafer yield, and ultimately to improve the overall performance of wafer fabrication system.

Based on the above discussion, this thesis studies the cyclic and non-cyclic scheduling problem of multi-cluster tool considering resource constraints and residency constraints in the case of robot dominate. The research domain includes 1-unit cyclic scheduling problem, multi-unit cyclic scheduling problem and non-cyclic scheduling problem. The objective of the research is to improve the efficiency of the multi-cluster tool under different wafer flow patterns by optimizing the sequence of robot moves.

## 1.2 Literature review

In recent years, a great deal of articles has emerged on solving scheduling and optimization issues of wafer fabrication systems and related fields [37]-[50]. Among all of the published articles, researches on modeling and scheduling of hoist in Printed Circuit Board (PCB) [51]-[55] and robotic cell [56]-[64] are mature, which provide references for the scheduling of multi-cluster tools.

Multi-cluster tools are distinguished from other systems for strict residency constraints, resource constraints and complex wafer flow patterns, which are considered in this thesis. As the best of our knowledge, most of early researches focused on the throughput analysis and deadlock prevention strategy for multi-cluster tools. At present, the research on scheduling problem of multi-cluster tool has just started both in the domestic and foreign. Because of its specialty and complexity,

scheduling problems of multi-cluster tool with residency constraints and resource constraints under varies wafer flow patterns have attracted a lot of attention.

### 1.2.1 Research on 1-unit cyclic scheduling problem with residency constraints

In wafer fabrication systems, 1-unit cyclic production under single wafer flow pattern is most widely used production mode because of the characteristics of easy execution and control. The research on 1-unit cyclic scheduling problem has gained widespread attention of scholars at home and abroad.

Ding and Yi [65] presented an event graph-based simulation and scheduling analysis of multi-cluster tools. In order to describe the complex robot moves accurately, such as the moves of the dual-armed robot, Ding and Yi further decompose the "transfer" action of the robot into "place" and "pick" actions. Thus, they simplified the multi-cluster tools event graph model to a "Decision-Moving-Done" cycle. The experimental results showed that the event graph-based "Decision-Moving-Done" cycle is able to describe complex manufacturing systems, such as 4-cluster tools.

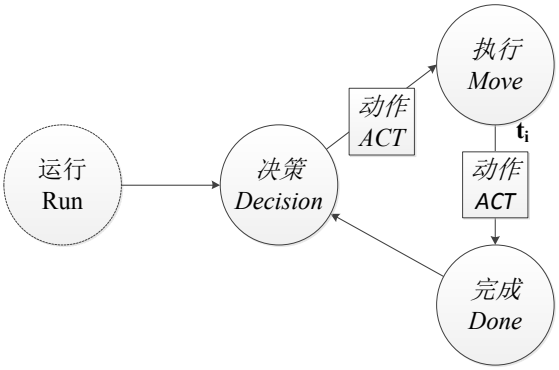



Figure 1.4 A simplified event graph for simulation of a cluster tool (Decision-Moving-Done Cycle)

Using finite capacity PN modeling technology, Zhu et al. [66] established a PN model of one-wafer cyclic scheduling problem. They proved that as long as the bottleneck cluster tool is a robot-dominant, there must be an optimal solution for one-wafer cyclic scheduling problem of multi-cluster tool.

Chan et al. [67]-[69] addressed the optimal cyclic scheduling problem for a two-cluster tool in the case where the robot transport time is constant. Firstly, they used an analytic method for establishing the lower bound of the resource-based two-cluster tools scheduling problem and proved that the optimal solution can be

found in the polynomial time. On this basis, they defined the concept of decoupling equivalence (DE), and proposed the conditions, in which the use of DE does not affect the productivity. At last, they proved that the "pull" strategy is capable of finding the optimal solution in polynomial time. However, in this literature, Chan et al. did not specify whether the proposed algorithm is feasible for larger multi-cluster tools.

The above-mentioned literature uses different methods for the modeling and cyclic scheduling problem of multi-cluster tools, but neither of them take into account the residency constraints, which is an important characteristics of wafer fabrication system. In fact, in wafer fabrication process, residency constraints and resource constraints are widely present.

Zhu et al. [70][71] used resource-oriented Petri net (ROPN) modeling technique in their research on modeling multi-cluster tools with residency constraints under single wafer flow pattern. In these two articles, Zhu et al. proposed a solution to a schedulable problem, whereas, such results are only available to the processing dominant scheduling problems.

Considering the influence of time window constraints, Zhou and Liu [72] studied the problem of two-hoist cyclic scheduling problem and proposed a heuristic algorithm that can generate combinations of sequences of hoist actions. Then, they sorted the combinations and found the optimal hoist actions sequence by means of linear programming (LP).

Chen et al. [73] studied the hoist cyclic scheduling problem with time window constraints. Chen et al. established a LP problem model with the objective of minimum cycle time. A graph-based algorithm was proposed by combining the branch-and bound algorithm with the bi-valued graph. The proposed method greatly reduced CPU time. Based on [73], Che et al. [74][75] introduced the branch and bound algorithm for solving multi-hoist cyclic scheduling problem. Due to the complexity of multi-cluster tools scheduling problem, it is difficult for the graph theory to describe such problems accurately. Therefore, above-mentioned methods are not applicable for cyclic scheduling problem of multi-cluster tools.

Che et al. [76] introduced the method of prohibited intervals (MPI) to establish a mathematical programming model of hoist cyclic scheduling problem. In the same way, literature [74] and [77] built mathematical programming models for the cyclic scheduling problems of hoist and robotic cells with time window constraints, respectively. After that, they proved that the optimal solution exists at several certain

points through analytical method, and designed heuristic algorithms to identify the feasibility of these points.

For two-cluster tools, Chan and Roeder [78] proposed the formula-based, the linear programming-based and the regression-based methods to estimate the impact of various factors on productivity. The results showed that when the time data (such as robot move time, processing time) is constant, the theoretical cycle time would be lower than the actual (random) cycle time. For the same problem, [79] and [81] proposed a decomposition method. They decomposed the two-cluster tools into two single cluster tools, and built linear programming model of each single cluster tool scheduling problem with objective of minimizing fundamental period, respectively. Zhou and Li[82] [82] set up a mixed-integer programming (MIP) model with the objective of minimum cycle time for the two-hoist scheduling problem with time window constraints. They solved the MIP model with CPLEX. In short, the above literatures used mathematical programming methods to model the 1-unit cyclic scheduling problem with residency constraints and they solved the model with CPLEX. The multi-cluster tools they studied are small scale equipment, which consist of two cluster tools; but they did not illustrate the feasibility of these methods for larger multi-cluster tools.

Based on the literature reviews, we found that there are only few studies concerned about residency constraints in the field of 1-unit cyclic scheduling problem of multi-cluster tools. In addition, most present models of multi-cluster tools with residency constraints are built by the method of mathematical programming; only small-scale scheduling problems such as 2-cluster tools are involved, and most of them are solved with CPLEX. In fact, in order to shorten the material handling distance between devices and reduce residency time out of the vacuum chamber, most of multi-cluster tools consist of more than three single cluster tools, even up to twelve single cluster tools. Therefore, in this thesis, it is assumed that the multi-cluster tool consists of three or more single cluster tools, with the aim of establishing a scheduling model and algorithm with versatility.

### 1.2.2 Research on multi-cluster tool multi-unit cyclic scheduling problem with residency constraints

Multi-cyclic production is another widely adopt production mode in wafer fabrication process, which is an efficient way to enhance the productivity of

manufacturing system. With number of wafer and number of wafer types in 1-unit cycle time increases, the resource conflicts in multi-cluster tool became fiercer. The difficulty of multi-unit cyclic scheduling problem exceeds that of 1-unit cyclic scheduling problem. Nowadays, there are only few studies on multi-cluster tools scheduling problem with residency constraints.

Che et al. [76] solved the optimal scheduling problem of hoist with multi-part types, including ordering the parts and hoist actions. They established the MPI-based model for hoist scheduling problem, and then they employed the dynamic branch and bound procedure to enumerate the prohibited intervals of decision variables. On this basis，Che and Chu[83] modeled the multi-unit cyclic scheduling problem of a flow shop with two robots. The problem converted to enumeration of sequences of robot moves, which single robot cannot execute.

For robotic cells scheduling problem with two-part types, Lei et al. [77] proposed a branch and bound algorithm to search for the optimal solution, and they proved that the productivity of robotic cells with two-part types is higher than that of robotic cells with identical parts.

Sriskandarajah et al. [84] studied the scheduling problem of the dual-armed robotic cell with multi-part types. They proved that the problem is strongly NP-hard, thus it is hard to find the optimal solution even if the sequence of the robot moves was predefined. They also proposed a heuristic algorithm to solve the two-robotic cells cyclic scheduling problem and then they extended the scale of robotic cell to M-machine.

Geismar et al. [85] established a model for robotic cells k-unit cyclic scheduling problem and proposed an algorithm to find the approximate-optimal solution of the problem. On this basis, Geismar et al. [86] proposed another approximate scheduling algorithm for robotic cell with single-gripper and dual-grippers.

In above-mentioned literatures, scholars focused on the single wafer type, constant processing time, no-wait, free-pick up, etc. Most of the objects are traditional flow shop and small-scale robotic cells. In recent years, with the upgrading of manufacturing technology, the scale of integrated manufacturing system has increased, the requirements of the workpiece processing technology has gradually increased, too. Thus, the control of processing time becomes more stringent. In the research domain of multi-unit cyclic scheduling of integrated manufacturing systems, some scholars have taken into account the processing time constraints.

Zhou et al. [87] built a MIP model for multi-unit cyclic scheduling problem of flow shop with time window constraints. The model is based on problem description and analysis and can be solved by CPLEX. Experimental results identified the feasibility and applicability of the established model. But, in their research, there is only one robot in the flow shop.

Kats and Levner [88] studied the robotic cells 2-unit cyclic scheduling problem with process time windows, and they proposed a polynomial algorithm under the assumption that there is only one robot in the robotic cell. The complexity of proposed algorithm is $\mathrm{O}\left(m^8 \log m\right)$.

For multi-unit multi-cluster tools cyclic scheduling problem, Li and Fung [89] assumed that the robot transport time is constant. They built a mixed-integer linear programming (MILP) model for the scheduling problem and used simulation method to explain how to solve the MILP model, and the optimal solution of k-unit cycle was found in their research.

Based on the above analysis, for multi-unit cyclic scheduling problem, a majority of literatures assumed that wafers are identical and the number of robots is within two. In order to involve the feature of wafer fabrication process, that is, the residency constraints, this thesis will study the multi-unit cyclic scheduling problem of multi-cluster tools with residency constraints under single wafer flow pattern but has varies of wafer types.

### 1.2.3 Research on non-cyclic scheduling problem with residency constraints under multi-wafer flow patterns

The demand of ASIC is increasing in recent years. In order to follow this trend, wafer fabrication enterprises gradually transmit the traditional cyclic production to non-cyclic production under a variety of wafer flow patterns. Non-cyclic scheduling problem with residency constraints is attracting more and more attention.

Paul et al. [90] proposed an adaptive time window heuristic algorithm to solve the scheduling problem of hoist with multi-part types. They defined two kinds of time windows for each action, one of them is for describing the feasible start time, and the other is for describing the feasible completion time. When the parts arrive, the time of each hoist action is calculated immediately. In order to avoid the occurrence of a situation in which the part has entered the production line but the hoist are not

available for the part, before the processing time starts, it must be identified that the time of robot moves is within the feasible time intervals.

Liu and Zhou [91][92] studied the scheduling problem of multi-cluster tool with residency constraints and multi-wafer types, and they put forward a heuristic online scheduling method based on time constraint set. The proposed algorithm consists of two parts, that is, the forward search of the feasible solution space and the backtracking calculation of the optimal scheduling time. The scheduling objective was the minimum makespan. Based on the above research results, Zhou et al. [93] proposed a method to convert the dual-armed robot into a single-armed robot. The post-conversion scheduling problem can be solved by using the time constraint set–based heuristic algorithm. Experimental results proved that the proposed heuristic algorithm adapt to the online scheduling of multi-cluster tools with single-armed robots and dual-armed robots. However, the optimality of the scheduling is not evaluated in above works.

Zhou and Li [94] present a two-stage idea of solving multi-cluster tools scheduling problems with multiple wafer types: 1) sequence the order of wafers; 2) schedule the robot moves. Based on Ant Colony search method and bi-directional search method, they constructed a novel heuristic algorithm to achieve the goal of minimizing makespan. Simulation experiments verified the effectiveness of proposed algorithm, but they did not prove the optimality of scheduling, neither.

The study of the non-cyclic scheduling problem with residency constraints is at its early age. The above studies considered the residency constraints in the multi-cluster tool scheduling problems. The proposed heuristic algorithms are relatively fast and adapt to the dynamic scheduling. The performance of the algorithms determines the qualities of the solutions. The above research results provide good references for the non-cyclic scheduling research of multi-cluster tools, but none of them illustrates the optimality of scheduling. Based on the theory of constraints (TOC), from the perspective of scheduling optimality, we will discuss the modeling and scheduling algorithm of multi-cluster tools non-cyclic scheduling problem with residency constraints.

### 1.2.4 Research on scheduling algorithms

From the perspective of optimality, the solution of scheduling problem is divided into two categories: the optimal solution and the approximate-optimal solution. Scheduling algorithms varies with different objectives.

The exact algorithms are used for optimal scheduling problem. The most commonly used precision algorithms are mathematical programming methods, such as analytical method, branch and bound method, mixed-integer programming, and so on [95]-[97]. The mathematical programming method has a deep theoretical basis, which can be solved by CPLEX and other commercial software. However, with the increase of the scale of the problem, the CPU time increases exponentially. Mathematical programming-based exact algorithm offers optimal solution, but it is limited to solve small-scale scheduling problem.

For example, Levner et al. studied the robotic cell cyclic scheduling problem, assuming that only one part was processed in each cycle. Using the method of prohibited interval, a model was constructed, and a polynomial algorithm with complexity of $\mathrm{O}\left(N^3 \log N\right)$ was proposed, where $N$ was the number of machines. [74], [76] and [77] also use the MPI for modeling and scheduling. Firstly, establish an MPI-based mathematical programming model, analyze the model by the means of mathematical analysis and interval analysis, and prove that the optimal solution must be in a few special points. Then, check the feasibility of the special points by designing a feasible solution check algorithm. Finally, analyze the complexity of proposed algorithm.

The MPI is good at describe the relationship between the residency constraints and the optimal solution intuitively, eliminate the solution space that may cause the deadlock of multi-cluster tools, and it is able to effectively transmit the high-dimensional problem to low-dimensional problem, which provides a way for modeling of multi-cluster tool scheduling problem. Using MPI to model the automated integrated manufacturing system with residency constraints has attracted the attention of scholars. In this thesis, MPI-based mixed integer programming models are established for 1-unit cyclic scheduling problem and multi-unit cyclic scheduling problem with residency constraints, and then the commercial software CPLEX are introduced to solve the model. In the case of small-scale scheduling problem, the experiment found high quality solutions.

Since most of multi-cluster tools scheduling problems are proved NP-hard, there are limitations for obtaining exact solutions. To make up for this deficiency, heuristic algorithm provides a good solution.

In recent years, many innovative scheduling algorithms have emerged, including constructive heuristic algorithm and meta-heuristic algorithm.

For large-scale cyclic scheduling problem with residency constraints, constructive heuristic algorithm is widely adopted. Yoon and Lee [100] discussed the online scheduling of single cluster tool with residency constraints and proposed a two-stage scheduling algorithm that can be solved in polynomial time. The algorithm is composed of two sub-algorithms, named: feasible scheduling space (FEASIBLE-SCHED-SPACE) and optimal scheduling (OPTIMAL-SCHED). As the name suggests, the feasible scheduling space algorithm is used to calculate the feasible solution space in the continuous domain. The optimal scheduling algorithm calculates the minimum makespan according to the feasible solution space. Experimental results verified that the proposed two-stage heuristic algorithm could obtain satisfied solution.

This thesis inherits the design idea of the above two-stage heuristic algorithm. A MPI-NLMIP based two-stage optimization algorithm is presented for solving multi-cluster tools 1-unit cyclic scheduling problem. The proposed algorithm is consisting of initial feasible scheduling space stage and approximate-optimal scheduling stage. In the first stage, the algorithm uses the bottleneck based searching method to find the feasible solution of the scheduling problem. In the approximate-optimal scheduling stage, we search for the approximate-optimal schedule in the feasible scheduling space based on the lower bound of the scheduling problem proposed in this thesis. At last, the objective of minimum cycle time is achieved. The simulation results show that the algorithm can obtain satisfactory approximate-optimal solution even when the load distribution of the device is extremely uneven.

For non-cyclic scheduling problem with multiple wafer types and residency constraints, constructive heuristic algorithms are widely used due to the speed of computation. According to the principle of "bottleneck machine-driven non-bottleneck machine" in theory of constraints, Zhai et al. [101] present a heuristic algorithm for job shop scheduling problem based on bottleneck process decomposition. The algorithm first identifies bottleneck device, and then decompose

the process along with the equipment, thus divide the large-scale scheduling problem into three sub-problems: the bottleneck process set scheduling, upstream non-bottleneck process set scheduling and downstream non-bottleneck process set scheduling. Finally, the solution to the original problem is obtained by solving the sub-problems.

Due to the huge gap between the structure of the multi-cluster tools and the job shop, the method of decomposing the original problem into sub-problems in the above literature cannot describe the coupling relationship of the cluster tools. Nevertheless, the method, which tries to improve the production efficiency of the manufacturing system through improve the rhythm of bottleneck equipment based on TOC, has broaden the way to solve the multi-cluster tools non-cyclic scheduling problem.

Thus, in this thesis, a bottleneck-based push-pull algorithm is proposed, aiming to solve the multi-cluster tools non-cyclic scheduling problem with residency constraints. According to the TOC, the proposed algorithm minimizes the wafer current residency time on the bottleneck module with the strategy of “pull” and “push” for the downstream and upstream module, respectively. Instead of decomposing the scheduling problem into three independent sub-problems, this thesis using the method of scheduling the three types of modules in turn and taking into account the close relationship between the robots caused by the coupling structure of cluster tools. Finally, the objective of minimum makespan is achieved.

Due to the solution of high quality, meta-heuristic attracts much attention. Meta-heuristic algorithms include genetic algorithm (GA), simulation-annealing (SA), tabu search (TS), particle swarm optimization (PSO), ant colony (AC), etc. [102].

Lim [103] proposed a GA based on the method of coding sequences of hoist move for the scheduling problem with time window constraints. The proposed algorithm requires relatively long time for large-scale scheduling problem.

Yang et al. [104] applied the simulated annealing algorithm to solve the multi-robotic cells scheduling problem. Through a large number of random simulations, they verified that the algorithm is capable of obtaining the optimal solution theoretically. However, due to the limited quantity of computations in practice, the optimal solution and the convergence speed are highly dependent on the convergence condition and the annealing time, which leads to the difficulty of obtaining the optimal solution or satisfactory solution of the large-scale multi-robotic cells scheduling problem.

For the scheduling problem of two-hoist with time window constraints, Zhou et al. [105] proposed a linear programming model-based searching algorithm. Firstly, the linear programming model is used to find the optimal schedule under the condition that the sequences of the move are given and the hoists are assigned. Then, a tabu list is introduced in searching to avoid solving the same linear programming model repeatedly. Lastly, they demonstrated the effectiveness and efficiency of the proposed algorithm in computational experiments.

Guo et al. [106] combined the ACO algorithm with decomposition method and created the decomposition-based classified ant colony optimization (D-CACO) scheduling algorithm. As the same implies, D-CACO algorithm uses the decomposition method to decompose the scheduling problem of large-scale multi-cluster tool into multiple single cluster tools scheduling sub-problems, and then use the classified ACO algorithm to group all of the operations of the sub-problems. Finally, depending on the type of machine, each sub-problem is scheduled.

The above-mentioned meta-heuristic algorithms are quit complex and the computational speed is relatively slow, but the solution quality high, which is suitable for static scheduling.

Kennedy and Eberhart [107] proposed the Particle Swarm Optimization algorithm in 1995, inspired by the results of the predation behavior of bird groups. This new heuristic algorithm has the characteristics of small number of individuals, simple calculation, and good robustness, and thus gets more and more attention. However, there is a problem that the PSO algorithm is easy to fall into the local optimum, which is similar to other meta-heuristic algorithms, and it has the disadvantages of premature convergence and large amount of computation. In order to solve this problem, Li and Che [108] introduced Chaotic search technology into the particle swarm algorithm. Using the ergodicity of chaos, they effectively avoid the algorithm into the local optimization, and achieve the purpose of optimizing the performance of PSO.

For the multi-unit cyclic scheduling problem of automated integrated manufacturing system, most of literatures use the exact algorithm to find the optimal solution or develop some simple scheduling rules; the results are not ideal. Therefore, this thesis aims to find an effective algorithm that can simultaneously account for the quality of the solution and CPU time. Based on the above literature, this thesis proposes a chaos-based hybrid PSO-TS optimization algorithm, which introduced the chaos search technique into PSO to increase the hysteresis. Meanwhile, using tabu list

to record the infeasible scheduling, and thus to reduce the CPU time and avoid to solve the same problem repeatedly.

### 1.2.5 Summary

From the above literature review, it is not hard to see that only few studies on the 1-unit cyclic scheduling problem of multi-cluster tools have considered the residency constraints, and the scale of problem is relatively small. Most of the literatures adopted exact algorithm to find the optimal solution. For multi-unit cyclic scheduling problem, a majority of literatures studied traditional flow shop and small-scale integrated manufacturing system, assuming that the part variety is single and the processing time is predetermined. Residency constraints are out of scope for most of literatures. Usually, exact algorithms are preferred for finding optimal solution of the scheduling problem, some simple scheduling rules are also developed but the results are not ideal. The researches on non-cyclic scheduling of multi-cluster tools with residency constraints are still not sufficient. Varieties of high computational speed heuristic algorithms are proposed. Nevertheless, the literature does not evaluate the optimality of the proposed scheduling model and algorithm.

### 1.2.6 Scientific issue

Multi-cluster tools are applied for the most complex section of semiconductor manufacturing process, and its characteristic determines that the multi-cluster tools is a large-scale advanced manufacturing system with complex logic relationship between the equipment resources and the products. Due to its inherent complexity, the existing literatures for the multi-cluster tools production management and advanced control methods are far less than the development of wafer processing technology and equipment. At present, the research domain is dominated by small-scale multi-cluster tools cyclic scheduling problem. Most of researches adopt mathematical programming to solve the problem and supplemented with simple scheduling rules. However, the above methods cannot meet the need of development of wafer fabrication process, which is large-scale, complex wafer flow pattern and high automation. Therefore, high efficient scheduling method is in urgent need.

Mathematical programming method has a strong theoretical basis and a wealth of tools to describe the complex logical relationship. It has been used in scheduling research for a long time. The constructive scheduling algorithm has the characteristics

of fast computation, strong pertinence and simple realization, and can obtain the approximate-optimal solution of large-scale problem in a short time. Especially when it hybrids with the meta-heuristic algorithm, the constructive heuristic algorithm would inherit excellent performance of meta-heuristic algorithm, like the robustness and high quality solution. This research has attracted the attention of many scholars in recent years. According to the literature review, the existing mathematical programming model and heuristic scheduling algorithm cannot describe and solve the multi-cluster tools scheduling problem with characteristics of large-scale, residency constraints, resource constraints and a variety of wafer hybrid production. Therefore, in this thesis, the scientific problem of the modeling and scheduling algorithm of the multi-cluster tools in the semiconductor wafer manufacturing system is put forward, and the scientific production management method suitable for the cluster equipment group is explored.

This thesis focus on two scientific problems: 1) Feature-oriented modeling and lower bounds study of multi-cluster tools scheduling problems; 2) Research on heuristic algorithm for scheduling of multi-cluster tools with complex (residency) constraints. The details are as follows:

1) Feature-oriented modeling and lower bounds study of multi-cluster tools scheduling problems;

Modelling is a method of describing scheduling problems in a formal language, which is the basis for analysing the inherent logical relations of scheduling problems. Mathematical programming is one of the most widely used modelling methods. It can display complex scheduling problems intuitively through mathematical symbols, and then use the solution tools to obtain the optimal solution of the problem conveniently. At present, there are a variety of mathematical programming method, such as analytic method, branch and bound method, mixed integer programming and so on, which are suitable for the scheduling problem of small feasible solution space. However, semiconductor wafers fabrication are characterized by complex processes, numerous wafer types, long production cycles, and strict residency time constraints, which are different from other manufacturing processes. Moreover, the multi-cluster tools in the wafer fabrication system have high automation, costly, intense competition and other characteristics, making the scheduling problem extremely complex. How to choose the appropriate mathematical programming method to establish a model that can reflect the characteristics of the scheduling problem is the focus of this thesis.

2) Research on heuristic algorithm for scheduling of multi-cluster tools with complex (residency) constraints

At present, most of the wafer fabrication enterprises in China remain adopt the basic scheduling rules, such as critical ratio (CR) + first in first out (FIFO), the earliest due date (EDD). Compared with the rapid development of wafer fabrication technology, production management skills need to be improved. The existing scheduling algorithm is mainly designed for small-scale wafer fabrication system, and most of them ignore the important feature, the residency constraints. Therefore, it is urgent to develop advanced and efficient scheduling algorithm. Most of scheduling problem of wafer fabrication systems has been proved NP-hard, so it is difficult to obtain the optimal solution for the scheduling problem of multi-cluster tools with complex (residency) constraints. Due to the existence of residency constraints and the characteristics of resource competition, it is necessary to schedule not only the occupation of the wafer to the robot, but also the occupation of the wafer to the processing equipment and the coordination of the adjacent manipulators. These problems greatly reduce the scope of feasible solutions, increase the difficulty of the search, thus, the complexity of the problem greatly improved. Develop the corresponding heuristic scheduling algorithm to deal with the above problems is in great need.

## 1.3 Significance

Wafer fabrication is a profitable industry. For example, in a 12-inch wafer fab with a monthly output of 30,000 wafers; the initial investment is about $1.6 billion, and if the utilization rate is increased by 1% for each device, the annual cost savings will be $ 2.95 million; if the monthly production capacity is increased by 1%, the monthly increase will be $ 710,000 in revenue. The multi-cluster tool is the most complex and expensive part of the wafer fabrication system, and it is the result of the intelligent, integrated and automated development of the wafer fabrication equipment in recent years [109]. Advanced multi-cluster tool scheduling and control technology can greatly improve the efficiency of wafer fabrication system, reduce costs, and shorten the fundamental period of wafer fabrication. Under the circumstance of expensive and limited equipment resources, this is of great practical significance to improve the market competitiveness of wafer fabrication industry in China.

Wafer fabrication involves the strict production process, such as residency constraints, reentrant, personalized customer requirements; and massive data processing requirements, etc. It makes scheduling problems of multi-cluster tools of high complexity. Since the multi-cluster tool is a new wafer fabrication equipment, the academia do not have a systematic understanding of its running rules in various wafer flow patterns, and the results of the corresponding scheduling optimization methods are insufficient. Based on the in-depth study of the existing results, and through the inheritance and development of key technologies, this thesis aims at improve the productivity and create higher efficiency. We will build a model that conforms to the characteristics of the multi-cluster tool, construct an efficient scheduling optimization method, use C ++ language programming algorithm to achieve the proposed algorithm, and finally design an efficient simulation experiment to evaluate the performance of the proposed algorithm. The results of this thesis are very helpful for enriching the theory system of scheduling and promoting the development of scheduling theory.

In summary, this subject not only has important theoretical research significance, but also has a significant practical value.

## 1.4 Research content

This thesis focuses on modelling and scheduling problem of multi-cluster tool considering residency constraints. Specifically, we will discuss the following scheduling problems in this thesis.

**1-unit cyclic scheduling problem under single-wafer flow pattern**: When the wafer types are identical, an MPI-based nonlinear MIP model is established for the scheduling problem. Based on this, we present and prove the lower bound of the problem. In order to reduce the CPU time, a two-stage constructive heuristic scheduling algorithm is proposed. In experimental analysis section, compare the solution of nonlinear mixed-integer programming model obtained by CPLEX with the put forward lower bound, and then analyze the applicable domain of proposed two-stage heuristic algorithm.

**Multi-unit cyclic scheduling problem under single-wafer flow pattern:** Learn from the above research, establish a non-linear MIP model of multi-unit cyclic scheduling problem of multi-cluster tools under single wafer flow pattern, and take

minimum FP as the objective function. Then, solve the established MIP model with CPLEX and analyze the complexity of the NLMIP model. Based on the above work, a chaos-based hybrid PSO-TS optimization algorithm is proposed. Simulation and experiments verify the reliability of the proposed NLMIP model and heuristic algorithm.

**Non-cyclic scheduling problem of multi-cluster tools under multi-wafer flow patterns:** With the minimum makespan as the scheduling target, a non-linear programming model of multi-cluster tool scheduling problem is established, and the lower bound of the problem is constructed and proved. For the efficiency of scheduling, a TOC-based heuristic algorithm, which is called bottleneck-based push-pull heuristic algorithm, is put forward. The validity and feasibility of the proposed algorithm are verified through simulation experiments. At last, the parameters that may influence the performance of proposed heuristic algorithm are studied by using the method of analysis of variances (ANOVA).

## 1.5 Thesis outline

The details of the chapters of this thesis are as follows.

Chapter 1 is introduction. This chapter mainly introduces the research background and significance of this subject, briefly reviews the process, characteristics of wafer fabrication and the equipment used in wafer fabrication system. In this chapter, we classified the multi-cluster tool scheduling problem, point out the problems studied in this thesis, and then reviews the relevant researches at home and abroad, and finally elaborate the research contents and the outline of this thesis.

Chapter 2 is about the structure and the characteristics of multi-cluster tool studied in this thesis. We describe in detail the multi-loop coupling structure of multi-cluster tool, highlight the characteristics and their effect to the scheduling of multi-cluster tools, including resource constraints, residency constraints and wafer flow patterns.

Chapter 3 is about the study of cyclic scheduling problem of multi-cluster tool considering the residency constraints, that is, the 1-unit cyclic scheduling problem under single-wafer flow pattern in above section.

Chapter 4 is the research of multi-unit cyclic scheduling problem of multi-cluster tool, which is the second problem studied in this thesis: multi-unit cyclic scheduling problem under single-wafer flow pattern. This chapter takes a 2-unit cyclic scheduling problem as an example.

In chapter 5, we investigate the modeling and non-cyclic scheduling problem of multi-cluster tool with residency constraints, that is, the third problem of this subject: the non-cyclic scheduling problem of multi-cluster tool under multi-wafer flow patterns.

Chapter 6 is conclusions and future works. In this chapter, we mainly summarize the thesis and prospects the future works.

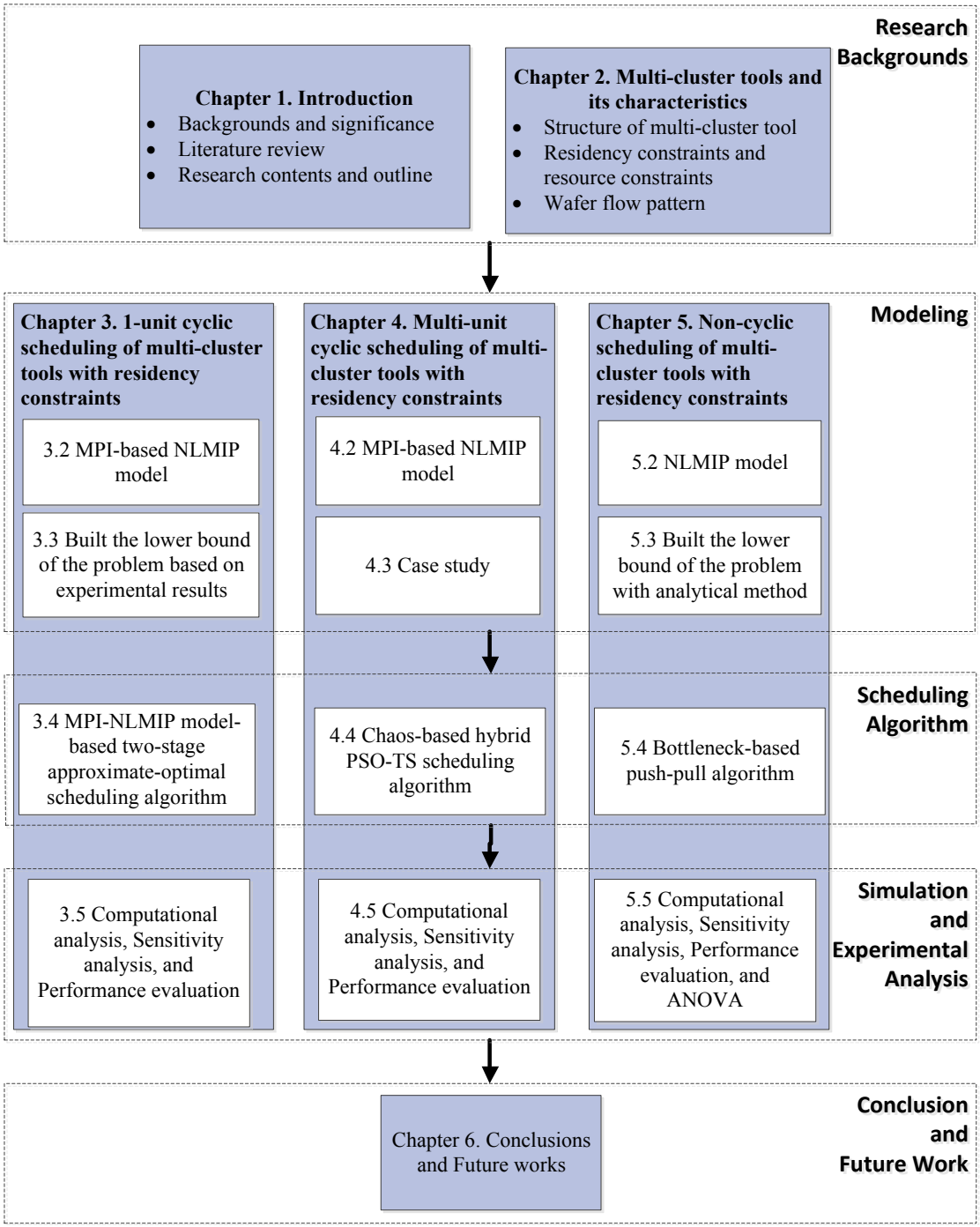



Figure 1.5 Structural chart of this thesis

# Chapter 2 The Structure and Characteristics of Multi-cluster Tools

First, this chapter will introduce the various modules that make up the multi-cluster tool in detail, including the name, function and features of the module. Then, we focus on two factors that have important influence on the scheduling of multi-cluster tool: residency constraints and resource constraints; and describe the causes of these two factors and their specific impact on the scheduling of multi-cluster tools. Finally, this chapter will introduce the definition and classification of the wafer flow pattern and illustrate its impact on the scheduling objectives.

## 2.1 Architecture of multi-cluster tools

As shown in Figure 2.1, the single cluster tool $C_1$ consists of a cassette module (CM), several processing modules (PM), and a transport module (TM), as defined in the SEMI standard E21-9 [110]. Wherein the cassette module is used to store the wafers to-be-processed and completed wafers, the processing module is responsible for wafer processing, such as lithography, etching and other processes, and the transport module is responsible for handling, loading and unloading of the wafers within the cluster tool. It is worth noting that the robot of transport module must be in a vacuum that is relatively isolated from the outside.

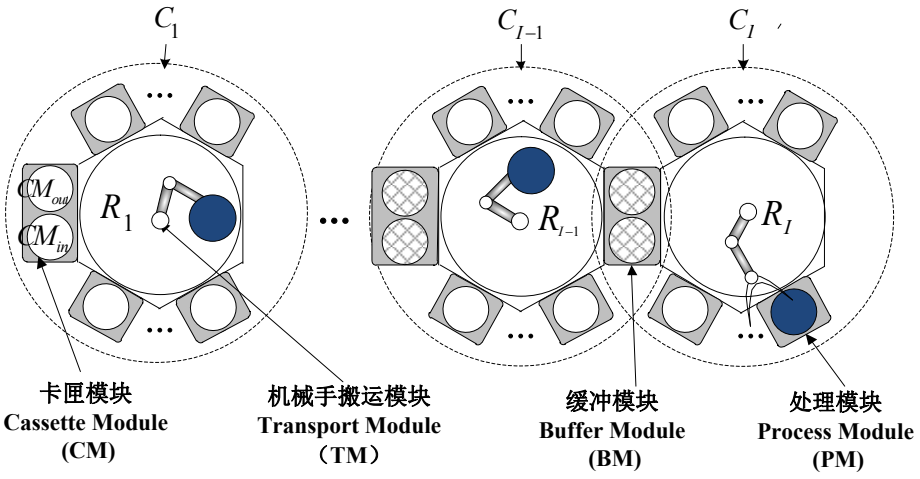



Figure 2.1 A schematic of multi-cluster tools

The multi-cluster tool is an automated integrated manufacturing unit that is connected by a number of cluster tools ($C_1, C_2, \ldots, C_I$) with the same or different processing functions via a buffer module (BM). As shown in Figure 2.1, $C_i$ ($i = 1, 2, \ldots, I$) denotes the *i*-th cluster tool. All the modules of a multi-cluster tool are installed on the ring skeleton except for the robots of transport modules. The robot is mounted in the center of a single cluster tool and can be rotated 360 degrees so that it can reach every module on the cluster tool. The wafer enters the multi-cluster tool through the cassette module first, and then enters the processing module according to the predetermined route to complete the processing, the buffer module acts as a connection channel to allow the wafer to enter the cluster tool connected to it. When wafer complete all the processes, it leaves the multi-cluster tool through another cassette module. The specific effects of these modules are described in detail below

1) Cassette module

Wafer fabrication requirements for environmental cleanliness are extremely harsh, subtle pollutants in the air will lead to poor quality of the wafer or even cause wafer scrapped. In order to prevent such phenomena from occurring, the cassette module is set up at the junction of the multi-cluster tool and the external environment to ensure that the entire process of the wafer is carried out in a vacuum environment.

The cassette module has two interfaces, one for the external environment and the other for the internal processing environment of multi-cluster tool. Wafer in batches (or lots) access to the cassete module through the interface that connects to the external environment. Then, the two interfaces of the cassette module close until the cassette is completely evacuated. At this point, the interface connected to the internal processing environment opens again and the wafers are able to get into the processing modules one by one in a predetermined order. After all wafers have been processed, the internal interface will be closed again until the material handling system arrives. Finally, the external interface will open so that the material handling equipment will transport the wafers in batch (or lot) to the next process.

Typically, a multi-cluster tool is equipped with two cassette modules, one for storing wafers waiting to be processed and the other for storing wafers that have been processed and waiting to be transported to the next process. This structure ensures that the external environment does not contaminate the interior of multi-cluster tool. When a batch (or lot) of wafers arrive at the multi-cluster tool cassette module, the processing can be started immediately, thus, the scheduling of multi-cluster tools is

not affected by the arrival rate of the wafers. The start time of the multi-cluster tool is the time at which the first piece of wafers leaves the cassette module.

2) Processing module

The processing module is one of the core components of the multi-cluster tool and can be considered as a separate wafer-processing unit. In a multi-cluster tool, different processing modules can be responsible for different processes, such as chemical vapor deposition (CVD), lithography, etching, ion implantation, chemical mechanical grinding and so on. The processing module can be combined according to the needs of the wafer process flow. Therefore, the multi-cluster tool is a flexible wafer fabrication system.

Without considering the parallel machines, a processing module can only process one wafer at a time. The wafer enters the processing module in accordance with the established wafer flow pattern, and the processed wafer needs to wait for the robot to unload it and transport it to the next module. Since the wafer fabrication process involves many chemical and physical reactions, it is necessary to keep the processing module running, and the idle processing module can cause the waste of resources and the increase of cost. Therefore, it is not only beneficial to improve the utilization rate of the processing module, but also has an important effect on reducing the cost and improving the overall efficiency of the wafer fabrication system by properly scheduling the wafer in the processing module.

3) Transport module

The transport module is the robot material handling system for multi-cluster tool, which is responsible for the transporting, unloading and loading of the wafers between the modules in the multi-cluster tool. There is only one transport module in each single cluster tool. The robots in a multi-cluster too are operated independently. Each of the robot has a limited range of motion and can only be responsible for the transportation of the wafers in the cluster tool where it is located, so two robots have to cooperate to transport wafers between adjacent cluster tools.

A robot move includes the three most basic movements of unloading, transporting and loading. In this thesis, we assume that these three actions are coherent and not-wait. According to the presently published articles, the time for a robot move is assumed constant.

4) Buffer module

The buffer module is unique to the multi-cluster tool and is mainly used to connect modules of adjacent cluster tools. Buffer modules are typically used only for

temporary storage and transit of wafers. They are not able to process wafer. Therefore, there is usually no upper bound of the residency constraints in the buffer module, but the capacity of the buffer module is limited. It is worth noting that, as a unique channel for access to adjacent cluster tools, the buffer module is a module through which the wafer must pass.

## 2.2 Characteristics of multi-cluster tools

In order to complete the complex process of wafer fabrication, the operation of the multi-cluster tool must meet the relevant constraints. This thesis will elaborate on two key operational characteristics of a multi-cluster tool: residency constraints and resource constraints.

### 2.2.1 Residency constraints

Residency constraints are an important time constraint in the wafer fabrication, and it is actually a kind of over-processing constraint. For example, prior to chemical processing, wafers often require an uninterrupted heating devices (the always-on oven) for preheating; the wafer is able to achieve the desired temperature just after 10 seconds of heating in this device (that is, the processing module in the multi-cluster too). If the wafer is heated for 15 seconds, the temperature is also within the range of available, but if the heating time exceeds 15 seconds, the wafer may be damaged or even scrapped due to overheating. Else, if the heating time is less than 10 seconds, the temperature will be too low to achieve the necessary conditions for wafer chemical treatment. In other words, the processing time is 10 seconds, when the processing is completed, wafer can still stay in this processing module for at most 5 seconds, and the upper bound of residency constraint is 15 seconds in this processing module. A similar situation is widely found in other processes of wafer fabrication.

There are two reasons for the generation of wafer over-processing: First, imbalanced capacity of adjacent processing module and second, the insufficient capacity of transport module. For the first reason that cause wafer over-processing, there are two cases. The first case is shown in figure 2.2. When there is a parallel machine in the processing module 1 (PM 1), since both the Machine 1 and the Machine 2 process the same process (step 1, denotes as $O_{w,1}$ in the figure, where w represents the number of the wafer and 1 is the step number). The cluster tools can simultaneously processing two wafers while there is only one processing module

(PM2) for the step 2. Therefore, only one of the two processed wafers in Machine 1 and Machine 2 can be processed immediately in PM 2, while the other must wait in Machine until PM 2 is available. The other case is shown in figure 2.3. If the time at which the wafer completes the step 1 (step 1 is denotes as $O_{w,1}$ in the figure, where w represents the number of the wafer and 1 is the step number) on PM 1 is much less than the time at which the wafer completes the step 2 ($O_{w,2}$) on PM 2; then, when the second wafer is processed at PM 1, PM 2 may still process the first wafer, where the second wafer has to reside on PM 1 and wait for PM 2 to be available before entering.

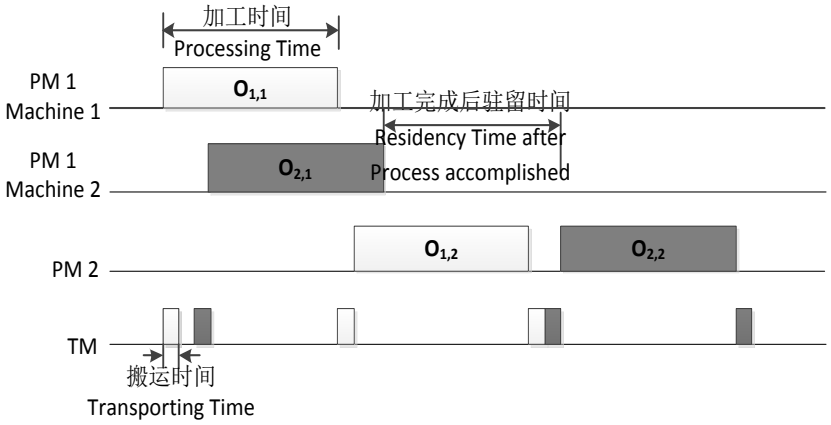



Figure 2.2 Wafer resides on parallel machines

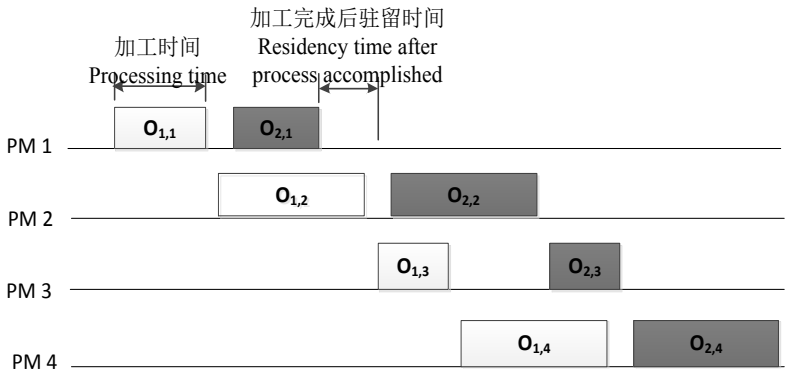



Figure 2.3 Wafer resides on efficient processing module

The lack of capacity of transport module is another reason why the wafer still resides in the processing module after processing is completed. Since a robot is responsible for the handling of the wafer between all modules of a single cluster tool and can only carry one wafer at a time, then a number of wafers will compete for the same robot. When the robot cannot respond to the need of several wafers simultaneously, it may lead to wafer over-processing.

Based on the above discussion, it can be seen that wafer reside on one or several processing modules in the multi-cluster tools often occurs during the wafer fabrication.

If the current residency time of the wafer exceeds the upper bound, it will produce bad quality wafers. Therefore, residency constraints are a very important factor that must be considered when studying multi-cluster tool scheduling problem.

### 2.2.2 Resource constraints

Resource constraint is a state in which a number of wafers in a batch (or lot) are waiting for a resource that cannot be acquired at the same time. The resource constraints of the multi-cluster tool can lead to fierce competition for resources and cause unnecessary losses. It can also reduce the utilization of equipment and even lead to the deadlock status.

Resource constraints often occur during the scheduling of multi-cluster tools. One of the causes of this situation is the residency constraints. For instance, when a wafer ($w_1$) is processed on PM 1, another wafer ($w_2$) is also ready to enter PM 1. Due to the residency constraints, $w_2$ must wait for $w_1$ to be processed in PM 1 and enters PM 1 after it leaves, resulting in resource constraints of multi-cluster tool. Besides, the tight coupling of the operation sequence between the processing module and the transport module is also one of the important causes of resource constraints. Since the only transport module in the multi-cluster tool accomplishes the handling of the wafers, so the processed wafer must be transferred to the next process via the transport module. In other words, the transportation of the wafer between the modules depends not only on whether the processing has been completed and whether the next module is available, but also on whether the transport module is available or not. If the transport module cannot respond to the wafer transporting requirements in a timely manner, it will cause the resource constraints of the multi-cluster tool.

It can be seen that the existence of resource constraints greatly improves the difficulty of scheduling problem of multi-cluster tool.

## 2.3 Wafer flow pattern

Unlike the way in which the wafers are transferred in batches (or lots) from the multi-cluster tools, the wafer flow inside the multi-cluster tools carried out one by one, without pre-emptive situation. In accordance with the pre-set path, the wafer in turn goes through the various processing modules to complete the specific processing steps. The path and order of the wafer through processing modules that have been preliminarily set in order to meet the requirements are called wafer flow pattern. The

red and blue solid lines in figure 2.4 represent two different wafer flow patterns, respectively. It is worth noting that different wafer flow patterns must represent different varieties of wafers, but different varieties of wafers are likely to have the same wafer flow pattern.

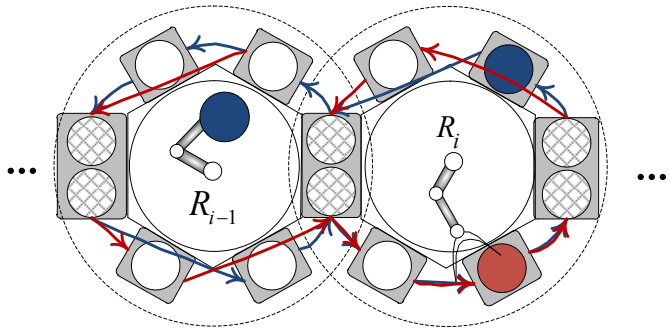



Figure 2.4 Schematic diagrams of different wafer flows

The wafer flow pattern has a very important influence on the target setting of multi-cluster tool scheduling optimization problem. In the same kind of wafer flow pattern, we usually seek the optimal cyclic scheduling, the purpose is to achieve the throughput maximize. In various wafer flow patterns, the target becomes achieving the objectives under different wafer flow patterns, such as the minimum makespan.

## 2.4 Summary

This chapter systematically introduces the multi-cluster tool and the important factors that affect its scheduling, and clarifies the object of this thesis. The multi-cluster tool is an integrated manufacturing unit composed of a cassette module, processing modules, buffer modules and a transport module. In order to complete the specific wafer fabrication process, the control of the environment and process is very strict, so there are residency constraints, resource constraints and other key factors affecting multi-cluster tool scheduling. During the operation of the multi-cluster tool, the setting of the scheduling targets is not the same depending on the wafer flow pattern. This chapter describes in detail the composition of the multi-cluster tool, and the residency constraints and resource constraints that are common to the wafer fabrication process. This chapter also analyzes the common wafer flow patterns and the objectives of multi-cluster tool scheduling problem in various wafer flow patterns.

# Chapter 3 **Research on One-unit cyclic scheduling Problem**

Due to the excellent characteristics of easy implementation and control, the 1-unit cyclic production under a single wafer flow pattern is the most important production mode of wafer fabrication system. In order to more effectively schedule the multi-cluster tools, this chapter discusses the modeling and scheduling problem of the multi-cluster tools under a single wafer flow pattern, and focuses on the characteristics of the residency constraints. This chapter establishes a MPI-based nonlinear mixed-integer programming model and builds the lower bound of the scheduling problem. Besides, a two-stage heuristic scheduling algorithm is proposed in this chapter. The effectiveness of the model and the proposed algorithm is verified by simulation experiments. This work is published in Wang et al. [116].

## 3.1 Problem description

As shown in figure 3.1, this section addresses the scheduling problem of multi-cluster tool with one-wafer type. The assumptions regarding the structure of the multi-cluster tool, the moves of the robot, the processing time and the residency constraint are as follows:

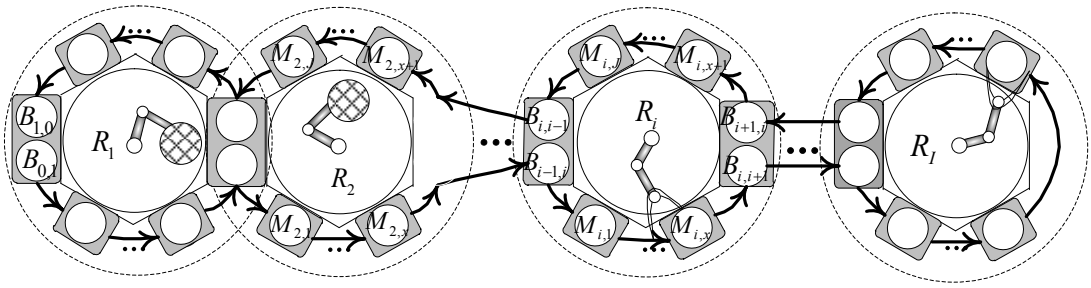



Figure 3.1 Schematic view of multi-cluster tool and single wafer flow

(1) the multi-cluster tool is symmetrically arranged;
(2) each cluster tool is connected with one or two other cluster tools;
(3) two adjacent cluster tools are connected through two buffer modules;
(4) all the transport modules are single-armed robots, for each robot, the unloading time is equal to the loading time, and the transporting time between modules is assumed to be constant;
(5) the process must begin as soon as the wafer is loaded in the processing

module;

(6) for each processing module, only one wafer can be loaded and processes at a time;

(7) each robot can handle one wafer at a time;

(8) the capacity of buffer module is one;

(9) residency constraints is considered, i.e., there is upper bound of current residency time for each processing module, after the processing is completed, the wafer would be defective or scrapped if it resides on the processing module longer than the upper bound of residency constraint.

According to the assumption (1), we can see that the number of processing modules in each cluster tool is even.

Assumption (2) defines the use of two-way connections between cluster tools, that is, the multi-cluster tool considered in this thesis is linear. Therefore, this chapter does not cover the multi-cluster tool of the tree-like divergent structure.

The buffer module is connected as an intermediary to the adjacent cluster tool. Based on assumption (3), when the wafer enters a cluster tool from the other cluster tool, it must first be transported by the robot to the buffer module, and then be removed from the buffer module by the robot of the other cluster tool. In other words, the wafer cannot skip the buffer module, and the transportation of the wafer between the cluster tools must pass through the buffer module. As mentioned above, in this chapter, a cluster tool is always connected by two corresponding buffer modules. In general, the buffer module does not have the function of processing wafers, so the buffer module does not have upper bound of residency constraints.

Assumption (4) describes the type of robot, that is, all the robots are single-arm manipulator. The processing module is circumferentially placed around the robot, and the transporting time is shorter than the processing time of the wafer, so it is feasible to assume that the robot transporting time is a small and constant.

In order to improve the utilization of the processing modules and to prevent the wafer from over-processed, it is assumed that the wafer starts processing immediately after it arrives at the processing module, without waiting (based on assumption (5)).

Depending on the situation in the actual production of the multi-cluster tool, assumption (6) to (8) in turn limit the capacity of the processing module, the transport module and the buffer module. Assumption (9) points out a very important constraint, the residency constraint, considered in this chapter. Thus, the optimal solution of the

scheduling problem addressed in this chapter must meet the following three categories of constraints:

1) Machine constraints: machine constraints include the capacity constraints of processing modules and buffer modules.
2) Transport constraints: namely, the capacity constraints of transport modules.
3) Residency constraints: wafers have residency constraints in the processing modules but do not have residency constraints in buffer modules.

To sum up, the problem studied in this chapter is how to achieve the minimum FP and maximize the throughput by efficiently scheduling the sequence and time of robot moves under the premise of meet residency constraints and resource constraints.

## 3.2 An MPI-based non-linear mixed-integer programming model

MPI is a method to find the optimal solution in the feasible solution interval by eliminating the infeasible solution interval, which can effectively transform the high dimension problem into low dimension problem. It is different from the method of time window that search for the intersection of the feasible solution interval, we establish the relationship between the constraint and the scheduling objective intuitively with MPI. Then, the union of the infeasible solution interval can be obtained and thus the set of feasible solutions is known by seeking the complement set of infeasible solution. Finally, the optimal solution is found from the feasible solution set.

In this section, we will use the method of prohibited intervals to analyze and model the multi-cluster tools scheduling problem that discussed in this chapter.

### 3.2.1 Notations and variables

In order to describe the mathematical model clearly, we define a series of notations and variables in this section. These notations and variables apply to this thesis.

In this thesis, the two-dimensional code is used for coding the cassette module, the processing module and the buffer module, that is, two subscripts are used to locate the module. For example, $M_{i,j}$ represents the $j$-th processing module of the $i$-th cluster tool, $B_{i,i+1}$ is the buffer module that a wafer pass through from the $i$-th

cluster tool to the $i+1$-th cluster tool, and $B_{0,1}$ indicates the cassette module that store unprocessed wafers.

In addition, this chapter introduces the variable $S_{i,j}$ to represents the time at which the 0-th wafer leaves $M_{i,j}$. In the steady-state of cyclic scheduling, the interval between the time at which a wafer leaves (or enters) the multi-cluster tool and the time at which the next wafer leaves (or enters) the multi-cluster tool is constant, i.e., the fundamental period, denoted as $T$. Therefore, $w \times T + S_{i,j}$ represents the time at which the $w$-th wafer leaves $M_{i,j}$.

Based on the above description, the notations and variables involved in this chapter are defined as follows:

| | |
|---|---|
| $T$ | Fundamental period; |
| $I$ | Number of cluster tools in the multi-cluster tool; |
| $J$ | Number of processing modules in the cluster tool; |
| $C_i$ | The $i$-th cluster tool; |
| $R_i$ | The robot of the $i$-th cluster tool; |
| $x$ | A half of $J$; |
| $M_{i,j}$ | The $j$-th processing module of the $i$-th cluster tool; |
| $S_{i,j}$ | The time of the 0-th wafer leaves $M_{i,j}$; |
| $\theta$ | The transporting time required for a robot to complete a transport move; |
| $B_{i,i+1}$ | The buffer module through which the wafer enters $C_{i+1}$ from $C_i$, $i \in [1, I-1]$; |
| $B_{i+1,i}$ | The buffer module through which the wafer enters $C_i$ from $C_{i+1}$, $i \in [1, I-1]$; |
| $B_{0,1}$ | The cassette module temporarily used to store unprocessed wafers; |
| $B_{1,0}$ | The cassette module temporarily used to store processed wafers; |
| $t_{B,i,i+1}$ | Wafer's current residency time on $B_{i,i+1}$; |

$t_{P,i,j}$ Wafer's current residency time on $M_{i,j}$;

$t^{L}_{P,i,j}$ Wafer processing time on $M_{i,j}$, i.e., the lower bound of residency time;

$t^{U}_{P,i,j}$ The upper bound of current residency time on $M_{i,j}$.

### 3.2.2 Objective function

As mentioned earlier, the objective of this chapter is to minimize the FP, namely:

$$\min T \tag{3-1}$$

### 3.2.3 Calculate the time at which the wafer leaves each PM

According to the definition, $S_{i,j}$ is equal to the length of time from the beginning of the 0th wafer entering the multi-cluster tool (time 0) to the time the 0th wafer leaving the PM $M_{i,j}$. That is, before the 0th wafer leaves $M_{i,j}$, $S_{i,j}$ is the sum of the current residency time and the robot handling time of all PMs and BMs that 0th wafer has passed through. It is assumed that the entering time of the 0th wafer is 0, therefore, variable $S_{i,j}$ can be expressed as $S_{i,j} = \sum_{i}\sum_{j} t_P + \sum_{i} t_B + \sum_{i}\sum_{j} \theta$.

Due to the complicated structure of multi-cluster tool, when the wafer is in a different area of the multi-cluster tools, the formula for $S_{i,j}$ is also different. As shown in figure 3.2, we divided the threshold of parameters $i$ and $j$ into five categories based on the location of wafer and use red, yellow, blue, green and grey to distinguish the five categories.

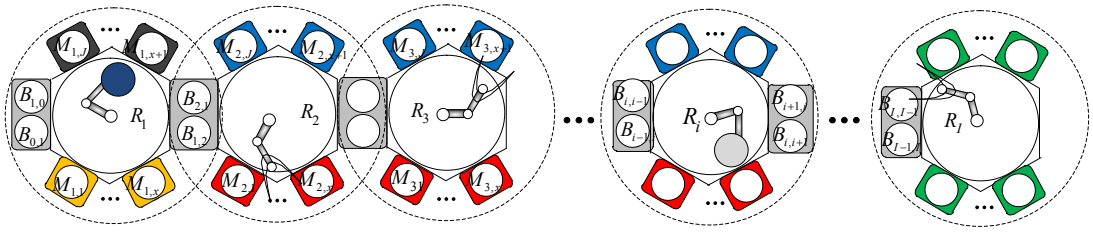



Figure 3.2 Schematic view of thresholds division for parameters in $S_{ij}$

When the wafer is on any of the red processing modules, that is, when $1<i<I$ and $j \le x$, the current total processing time for the 0-th wafer is $\sum_{m=1}^{i-1}\sum_{n=1}^{x} t_{P,m,n} + \sum_{n=1}^{j} t_{P,i,n}$, the current residency time on buffer modules is $\sum_{m=1}^{i-1} t_{B,m,m+1}$, and the current total transporting time is $\theta\left[(i-1)x+(i-1)+j\right]$. The sum of the above-mentioned time is the time at which the 0th wafer leaving the processing module:

$$S_{i,j} = \sum_{m=1}^{i-1}\sum_{n=1}^{x}\left(\theta + t_{P,m,n}\right) + \sum_{n=1}^{j}\left(\theta + t_{P,i,n}\right) + \sum_{m=1}^{i-1}\left(\theta + t_{B,m,m+1}\right); 1<i<I \ ; \ j \le x \ . \tag{3-2}$$

Similarly, when the wafer is in any of the yellow processing modules, i.e., when $i=1$ and $j \le x$, the time when the 0-th wafer leaves $M_{i,j}$ is as follows:

$$S_{i,j} = \sum_{n=1}^{j}\left(\theta + t_{P,1,n}\right) ; i=1 ; \ j \le x \ . \tag{3-3}$$

When the wafers accomplished processing on the last cluster tool, they sequentially go through the $I-1$-th cluster tool to the second cluster tool in reverse order. Namely, when wafer is on any of the blue processing modules, the total current processing time of wafer is $S_{i,j} = \sum_{n=1}^{j} t_{P,i,n} + \sum_{m=i+1}^{I}\sum_{n=1}^{J} t_{P,m,n} + \sum_{m=1}^{i-1}\sum_{n=1}^{x} t_{P,m,n}$. Thus, the time of 0-th wafer leaves $M_{i,j}$ is:

$$S_{i,j} = \sum_{n=1}^{j}\left(\theta + t_{P,i,n}\right) + \sum_{m=1}^{I-1}\left(\theta + t_{B,m,m+1}\right) + \sum_{m=i}^{I-1}\left(\theta + t_{P,m+1,m}\right) + \sum_{m=i+1}^{I}\sum_{n=1}^{J}\left(\theta + t_{P,m,n}\right) + \sum_{m=1}^{i-1}\sum_{n=1}^{x}\left(\theta + t_{P,m,n}\right);$$

$$1<i<I \ ; x<j \le J \ . \tag{3-4}$$

If the wafer is on any of the grey processing modules, that is, if $i=1$ and $x+1 \le j \le J$, the 0-th wafer leaves $M_{i,j}$ at the following time:

$$S_{i,j} = \sum_{m=2}^{I}\sum_{n=1}^{J}\left(\theta + t_{P,m,n}\right) + \sum_{n=1}^{j}\left(\theta + t_{P,1,n}\right) + \sum_{m=1}^{I-1}\left(2\theta + t_{B,m+1,m} + t_{B,m,m+1}\right);$$

$$i=1 \ ; x+1 \le j \le J \ . \tag{3-5}$$

And so on, if the wafer is on the last cluster tool, that is, in figure 3.2 on any of the green processing module, there are:

$$S_{i,j}=\sum_{m=1}^{I-1}\sum_{n=1}^{x}\left(\theta+t_{P,m,n}\right)+\sum_{n=1}^{j}\left(\theta+t_{P,I,n}\right)+\sum_{m=1}^{I-1}\left(\theta+t_{B,m,m+1}\right);i=I;1\le j\le J. \quad (3\text{-}6)$$

To sum up, $S_{i,j}$ is calculated according to equation (3-2) to (3-6).

### 3.2.4 Machine constraints

Machine constraints consist of processing module constraints and buffer module constraints.

According to assumption (6), a processing module can process one wafer at a time, which means the $p$-th wafer cannot enter before the $p-1$-th wafer leaves $M_{i,j}$. That is to say, the $p$-th wafer has to wait until the $p-1$-th wafer is unloaded and transferred to the next module. So, there is

If $1\le i\le I$ and $j=1$, then the $p-1$-th wafer is on $M_{i,j}$, and the pre-odder module of the processing module where the $p$-th wafer locates on is $B_{i-1,i}$. In order to meet the processing module constraints, the time of the $p$-th wafer leaves $B_{i-1,i}$ must not be earlier than the time when the $p-1$-th wafer is unloaded from $M_{i,j}$ and be transferred to the next processing module, that is, $S_{i-1,x}+\theta+t_{B,i-1,i}+T\ge S_{i,j}+\theta$. According to the format of MPI, the above formula can be sorted into: $T\notin\bigcup_{i=1}^{I}\left(-\infty,S_{i,1}-S_{i-1,x}-t_{B,i-1,i}\right]$.

Similarly, if $1\le i\le I-1$ and $j=x+1$, then the pre-order module of the PM where $p$-th wafer is must be $B_{i+1,i}$, therefore, $S_{i+1,J}+\theta+t_{B,i+1,i}+T\ge S_{i,j}+\theta$, to sort out: $T\notin\bigcup_{i=1}^{I-1}\left(-\infty,S_{i,x+1}-S_{i+1,J}-t_{B,i+1,i}\right]$.

If $i=I$ and $j=x+1$, the pre-order processing modules to the $p$-th wafer is $M_{I,x}$, so, $T\notin\left(-\infty,S_{I,x+1}+\theta-S_{I,x}\right]$. Else, if $1\le i\le I$ and $x+2\le j\le J$, or if $1\le i\le I$ and $2\le j\le x$, the pre-order processing module to the $p$-th wafer is

$M_{i,j-1}$ , then we can know that $T \notin \bigcup_{i=1}^{I} \bigcup_{j=x+2}^{J} \left(-\infty, S_{i,j} + \theta - S_{i,j-1}\right]$ and $T \notin \bigcup_{i=1}^{I} \bigcup_{j=2}^{x} \left(-\infty, S_{i,j} + \theta - S_{i,j-1}\right]$ .

Based on the above analysis, due to the constraints of the processing module capacity, the minimum FP needs to be met the following formula:

$$T \notin \bigcup \left\{ \begin{array}{l} \bigcup_{i=1}^{I} \left(-\infty, S_{i,1} - S_{i-1,x} - t_{B,i-1,i}\right], \\ \bigcup_{i=1}^{I-1} \left(-\infty, S_{i,x+1} - S_{i+1,J} - t_{B,i+1,i}\right], \\ \left(-\infty, S_{I,x+1} + \theta - S_{I,x}\right], \\ \bigcup_{i=1}^{I} \bigcup_{j=x+2}^{J} \left(-\infty, S_{i,j} + \theta - S_{i,j-1}\right] , \\ \bigcup_{i=1}^{I} \bigcup_{j=2}^{x} \left(-\infty, S_{i,j} + \theta - S_{i,j-1}\right] \end{array} \right\} \tag{3-7}$$

According to the assumption (8), each buffer module stores up to one wafer at a time. Then, the minimum FP must be greater than the sum of the actual residency time of the wafer in the buffer module and the robot's transporting time, thus, we have constraint (3-8).

$$T \notin \bigcup_{i=1}^{I-1} \left(-\infty, \max\left(t_{B_{i+1,j}} + \theta, t_{B_{i,i+1}} + \theta\right)\right) \tag{3-8}$$

### 3.2.5 TM constraints

It can be seen from the assumption (7) that if two wafers simultaneously send a handling command to a robot, the robot can only respond to the needs of a wafer, while the other wafer must wait until the robot is available again. In view of the particularity of the multi-cluster tool structure, the resource conflict can only occur when two wafers are on the PMs of the same cluster tool or one of them is on a buffer module connecting two adjacent cluster tools. Therefore, depending on the location of the two wafers, the modeling of the transporting module constraints can be divided into the following three cases:

1) Both wafers are on the PMs

If one wafer is on the $M_{i,p}$ and the other is on the $M_{i,q}$, where $1 \le i \le I$ and $1 \le q < p \le J$ are satisfied; they may be issued to the $R_i$ demand command at the same time, resulting in the situation of demand conflict.

As shown in figure 3.3 (1), if two wafers are on the blue processing module as shown in the figure, that is, when $1 \le q < p \le x$; then we assume that the $m$-th wafer is on the $M_{i,p}$ and the $k+m$-th wafer is on the $M_{i,q}$, where $k$ is the number of wafers that are processing between the $M_{i,p}$ and the $M_{i,q}$. In this case, the robot $R_i$ can first move the $k+m$-th wafer to the target module, and then carry the $m$-th wafer; can also transport the $m$-th wafer to the target module, and then move the $k+m$-th wafer. Therefore, the following inequality is given.

$(k+m)T + S_{i,q} + \theta \le mT + S_{i,p}$ or $(k+m)T + S_{i,q} \ge mT + S_{i,p} + \theta$, where $1 \le i \le I$ and $1 \le k \le p-q$.

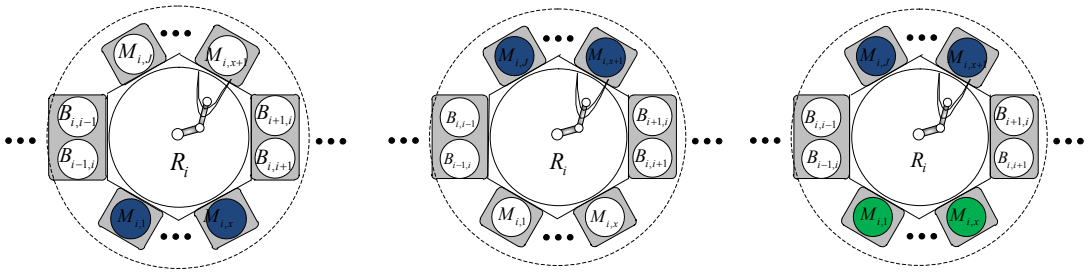



Figure 3.3 Three cases that may lead to TM resource competition when wafer p and q are in the PMs

Similarly, as shown in figure 3.3 (2) where the blue-filled processing module is located, if $x+1 \le q < p \le J$ is met, let us suppose that the $m$-th wafer is on the $M_{i,p}$ and the $k+m$-th wafer is on the $M_{i,q}$, $k$ is still the number of wafers between $M_{i,p}$ and $M_{i,q}$. Then, $T$ must satisfy the inequality $(k+m)T + S_{i,q} + \theta \le mT + S_{i,p}$ or $(k+m)T + S_{i,q} \ge mT + S_{i,p} + \theta$, where $1 \le i \le I$ and $1 \le k \le (I-i)(J+2) + p - q$.

If $1 \le q \le x$ and $x+1 \le p \le J$, we assume that the $m$-th wafer is on the $M_{i,p}$, which is filled with blue in figure 3.3 (3); and we assume that the $k+m$-th wafer is on the $M_{i,q}$, which is filed with green in the figure. Thus, the following inequality must be satisfied: $(k+m)T + S_{i,q} + \theta \le mT + S_{i,p}$ or $(k+m)T + S_{i,q} \ge mT + S_{i,p} + \theta$, where $1 \le i \le I$ and $1 \le k \le (I-i)(J+2) + (p-q)$.

In summary, the minimum fundamental period needs to satisfy the following constraint (3-9).

$$T \notin \bigcup_{i=1}^{I} \left\{ \begin{array}{l} \bigcup_{p=2}^{x} \bigcup_{q=1}^{p-1} \bigcup_{k=1}^{p-q} \left[ \frac{S_{i,p} - S_{i,q} - \theta}{k}, \frac{S_{i,p} - S_{i,q} + \theta}{k} \right], \\ \bigcup_{p=x+2}^{J} \bigcup_{q=x+1}^{p-1} \bigcup_{k=1}^{p-q} \left[ \frac{S_{i,p} - S_{i,q} - \theta}{k}, \frac{S_{i,p} - S_{i,q} + \theta}{k} \right], \\ \bigcup_{p=x+1}^{J} \bigcup_{q=1}^{x} \bigcup_{k=1}^{(I-i)(J+2)+p-q} \left[ \frac{S_{i,p} - S_{i,q} - \theta}{k}, \frac{S_{i,p} - S_{i,q} + \theta}{k} \right] \end{array} \right\} \tag{3-9}$$

2) One wafer is on the BM and the other wafer is on the PM

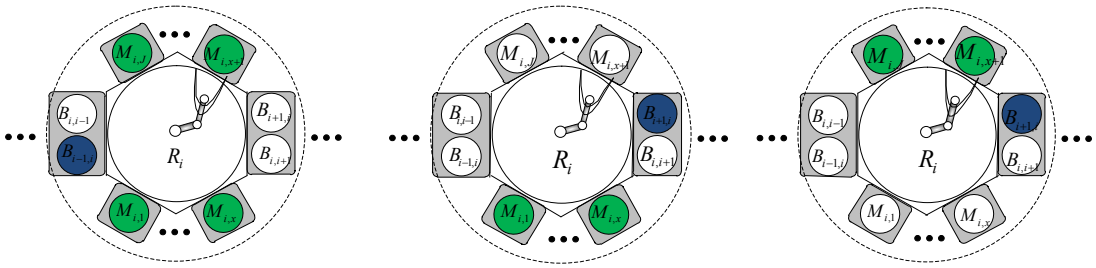
When a wafer is on a buffer module connected to $C_i$ and another wafer is on a processing module $C_i$, where $2 \le i \le I$ is satisfied, the two wafers may compete for the $R_i$. In order to avoid resource conflicts, we conducted the following analysis and established equations (3-10) and (3-11).

As shown in figure 3.4 (1), we make a hypothesis that the $k+m$-th wafer is on the $B_{i-1,i}$, i.e., the buffer module that is blue; and that the $m$-th wafer is on any of the $M_{i,p}$ $(1 \le p \le J)$, which are filled by green; $k$ is the number of wafers between $B_{i-1,i}$ and $M_{i,p}$. Due to the limitations of the robot capacity, the transport module can first carry the $k+m$-th wafer to the target module, then carry the $m$-th wafer, and vice versa. Thus, we have constraint （3-10）.

$$T \notin \bigcup_{i=2}^{I} \left\{ \begin{array}{l} \bigcup_{p=1}^{x} \bigcup_{k=1}^{p} \left[ \frac{S_{i,p} - S_{i-1,x} - t_{B,i-1,i} - 2\theta}{k}, \frac{S_{i,p} - S_{i-1,x} - t_{B,i-1,i}}{k} \right], \\ \bigcup_{p=x+1}^{J} \bigcup_{k=1}^{(I-i)(J+2)+p} \left[ \frac{S_{i,p} - S_{i-1,x} - t_{B,i-1,i} - 2\theta}{k}, \frac{S_{i,p} - S_{i-1,x} - t_{B,i-1,i}}{k} \right] \end{array} \right\} \tag{3-10}$$

As shown in figure 3.4 (2), if the $m$-th wafer is on blue-filled buffer module ($B_{i+1,i}$) and the $k+m$-th wafer is on any of the green-filled processing module ($M_{i,q}$), where $1 \le q \le x$; we assume that $k$ is the number of wafers between $B_{i+1,i}$ and $M_{i,q}$. Thus,

$$T \notin \bigcup_{i=1}^{I-1} \bigcup_{q=1}^{x} \bigcup_{k=1}^{x-q+(I-i)(J+2)} \left[ \frac{S_{i+1,J} + t_{B,i+1,i} - S_{i,q}}{k}, \frac{S_{i+1,J} + t_{B,i+1,i} - S_{i,q} + 2\theta}{k} \right].$$




(1) While one of the wafers is on the $B_{i-1,i}$, the other one is on the $M_{i,p}$

(2) While one of the wafers is on the $B_{i+1,i}$, the other one is on the $M_{i,q}$, where $1 \leqslant q \leqslant x$

(3) While one of the wafers is on the $B_{i+1,i}$, the other one is on the $M_{i,q}$, where $x+1 \leqslant q \leqslant J$

Figure 3.4 Three cases that may lead to TM resource competition when wafer p and q are in the PM and BM, respectively

If the $k+m$-th wafer is on the $B_{i+1,i}$ that is filled with blue in figure 3.4 (3); the $m$-th wafer is on the $M_{i,q}$, where $x+1 \le q \le J$, i.e. $m$-th wafer is on any of the processing modules that are filled with green; and $k$ is the number of wafers between $B_{i+1,i}$ and $M_{i,q}$; then we have the constraint as follows:

$$T \notin \bigcup_{i=1}^{I-1} \bigcup_{q=x+1}^{J} \bigcup_{k=1}^{q-x} \left[ \frac{S_{i+1,J} + t_{B,i+1,i} - S_{i,q}}{k}, \frac{S_{i+1,J} + t_{B,i+1,i} - S_{i,q} + 2\theta}{k} \right].$$

Based on MPI, we can combine the above two constraints into constraint (3-11).

$$T \notin \bigcup_{i=1}^{I-1} \left\{ \begin{array}{l} \bigcup_{q=1}^{x} \bigcup_{k=1}^{x-q+(I-i)(J+2)} \left[ \frac{S_{i+1,J} + t_{B,i+1,i} - S_{i,q}}{k}, \frac{S_{i+1,J} + t_{B,i+1,i} - S_{i,q} + 2\theta}{k} \right], \\ \bigcup_{q=x+1}^{J} \bigcup_{k=1}^{q-x} \left[ \frac{S_{i+1,J} + t_{B,i+1,i} - S_{i,q}}{k}, \frac{S_{i+1,J} + t_{B,i+1,i} - S_{i,q} + 2\theta}{k} \right] \end{array} \right\} \quad (3\text{-}11)$$

3) Two wafers are on different BMs

In addition to the cases 1) and 2) mentioned above, the demand conflict for the machine might also occur between two wafers on different buffer modules. As figure 3.5 shows, if $2 \le i \le J-1$, when the $m$-th wafer is on the $B_{i+1,i}$, that is, the

blue-filled buffer module in the figure; moreover, the $k+m$-th wafer is placed on the $B_{i-1,i}$, that is, the location of the green-filled buffer modules; similarly, $k$ is the number of wafer that are being processed between $B_{i+1,i}$ and $B_{i-1,i}$. Then, according to assumption (7), $R_i$ should transport the $k+m$-th wafer to the target module before responds to the handling commanding of the $m$-th wafer, or in reverse order response to wafer handling requirements. It can be seen that $T$ must satisfy the constraint (3-12).

$$T \notin \bigcup_{i=2}^{I-1} \bigcup_{k=1}^{(I-i)(J+2)+x+1} \left[ \frac{S_{(i+1)J} + t_{B,i+1,i} - S_{i-1,x} - t_{B,i-1,i} - \theta}{k}, \frac{S_{(i+1)J} + t_{B,i+1,i} - S_{i-1,x} - t_{B,i-1,i} + \theta}{k} \right] \quad (3\text{-}12)$$

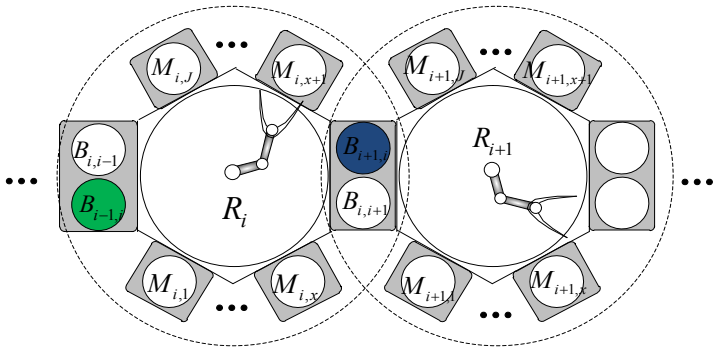



Figure 3.5 Diagrammatic sketch of two wafers on BMs

## 3.2.6 Residency constraints

In this chapter, there is residency constraint on the processing module, that is, the actual residency time of the wafer on the processing module must not be less than the processing time required for the wafer and not exceed the upper bound of residency time. Therefore, constraint (3-13) must be satisfied.

$$t_{P,i,j}^{L} \le t_{P,i,j} \le t_{P,i,j}^{U}; \quad i \in [1, I]; j \in [1, J]. \quad (3\text{-}13)$$

Since the buffer module is only used for temporary storage of wafers and does not have a processing function, there is no upper limit for the residency time of the wafer in the buffer module. Thus, constraint (3-14) and (3-15) must be satisfied.

$$t_{B,i,i+1} \ge 0; i \in [0, I-1]. \quad (3\text{-}14)$$

$$t_{B,i+1,i} \ge 0; i \in [0, I-1]. \quad (3\text{-}15)$$

To sum up, the scheduling problem studied in this chapter is a nonlinear mixed-integer programming problem with (3-1) as the objective and (3-2) to (3-15) as

constraints. In other words, in this chapter, we established a non-linear mixed-integer programming model (MPI-NLMIP model) based on MPI, which can describe the cyclic scheduling problem of multi-cluster tools with residency constraints and identical wafer flow patterns

## 3.3 Lower-bound of 1-unit cyclic scheduling problem

In this section, we try to use the experimental method to relax the partial constraints of the MPI-NLMIP model and reduce the model complexity. The lower bound of the 1-unit cyclic scheduling problem is established by solving the relaxed model by CPLEX.

The CPLEX optimization software used in this thesis is called IBM ILOG CPLEX Optimizer. It is a high performance commercial mathematical programming model developed by IBM. It has the characteristics of solving complex problems and fast response. It is suitable for solving linear programming problem, mixed-integer programming problem, quadratic programming problem, and so on. It is worth noting that IBM ILOG CPLEX Optimizer provides developers with a variety of flexible interfaces, such as the interface of C + + of Visual Studio platform adopted in this thesis. Due to these excellent performance, the IBM ILOG CPLEX Optimizer is widely used by academics and the industry, and is a commonly used programming problem solving software. This thesis uses the IBM ILOG CPLEX Optimization Studio software with version 12.2 to solve the model on a PC with Intel Core i3 (2.53GHz) CPU and 4GB memory.

The lower bound of the 1-unit scheduling problem is established as follows.

The first step is relaxing only one constraint of the MPI-NLMIP model. The experimental results are shown in table 3.1. The table shows four sets of experiments with three types of multi-cluster tools, which consist of different numbers of cluster tools. Take three-cluster tool as an example, the solution time required for solving the MPI-NLMIP model is 0.53 seconds and the FP is 26. When we relax various constraints, the CPU time reduced; the FP becomes three if we relax constraint (3-7), but the FP does not change when any other constraint is relaxed. The ideal lower bound should meet two aspects, the short CPU time and approximate to the optimal solution. According to the experimental results, when we relax the constraints (3-9) or (3-10), CPLEX requires shorter CPU time and the solution of both relaxed models are exactly same as the optimal solution.

Table 3.1 The impact of relax constraints on MPI-NLMIP Model in aspect of CPU time and optimal FP

| 3-cluster tool | | | 10-cluster tool | | | 12-cluster tool（1） | | | 12-cluser tool（2） | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Relaxed Constraint | CPU Time (Second) | FP | Relaxed Constraint | CPU Time (Second) | FP | Relaxed Constraint | CPU Time (Second) | FP | Relaxed Constraint | CPU Time (Second) | FP |
| Non | 0.53 | 26 | Non | 85.65 | 33 | Non | 298.15 | 28 | Non | 215.89 | 35 |
| 3-7 | 0.42 | 3 | 3-7 | 101.49 | 3 | 3-7 | 289.34 | 3 | 3-7 | 277.01 | 3 |
| 3-8 | 0.48 | 26 | 3-8 | 74.08 | 33 | 3-8 | 689.99 | 28 | 3-8 | 273.14 | 35 |
| 3-9 | 0.3 | 26 | 3-9 | 22.46 | 33 | 3-9 | 111.42 | 28 | 3-9 | 88.65 | 35 |
| 3-10 | 0.28 | 26 | 3-10 | 32.21 | 33 | 3-10 | 80.54 | 28 | 3-10 | 82.06 | 35 |
| 3-11 | 0.42 | 26 | 3-11 | 92.68 | 33 | 3-11 | 202.29 | 28 | 3-11 | 565.53 | 35 |
| 3-12 | 0.36 | 26 | 3-12 | 150.35 | 33 | 3-12 | 1696.43 | 28 | 3-12 | 151.94 | 35 |
| 3-cluster tool | | | 10-cluster tool | | | 12-cluster tool（1） | | | 12-cluser tool（2） | | |
| Relaxed Constraint | CPU Time (Second) | FP | Relaxed Constraint | CPU Time (Second) | FP | Relaxed Constraint | CPU Time (Second) | FP | Relaxed Constraint | CPU Time (Second) | FP |
| 3-9&3-10 | 0.08 | 26 | 3-9&3-10 | 11.58 | 33 | 3-9&3-10 | 20.34 | 28 | 3-9&3-10 | 17.91 | 35 |

The second step is based on the conclusion of the first step. In step 2, we try to approach the lower bound of the scheduling problem by relaxing the constraints (3-9) and (3-10) at the same time. The experimental results are shown in table 3.1. By comparing the MPI-NLMIP model without constraint (3-9), constraint (3-10), and the both, it can be seen that the CPU time of solving the MPI-NLMIP model without constraint (3-9), and (3-10) (the R-MPI-NLMIP model) is 0.08 seconds, which is much shorter than solving the MPI-NLMIP model (0.53 seconds). Moreover, the lower bound of 1-unit cyclic scheduling problem is equal to the optimal solution. Thus, the solution of the MPI-NLMIP model with relaxation of constraints (3-9) and (3-10) (hereinafter abbreviated as R-MPI-NLMIP model) can be used as the lower bound of the 1-unit cyclic scheduling problem studied in this chapter, and it is denoted as $T^{LB}$. In order to verify the performance of the R-MPI-NLMIP model, we compare the MPI-NLMIP model with R-MPI-NLMIP model from the two aspects of CPU time and optimality of solution, taking eight kinds of multi-cluster tools, which consist of 2 to 20 cluster tools, as examples (See table 3.2). From the point of view of CPU time, when the number of cluster tools increases, the growth rate of R-MPI-NLMIP model is much lower than that of MPI-NLMIP model, and the gap between them increases gradually. Especially in the experimental group of 20-cluster tools, CPLEX has been unable to solve the MPI-NLMIP model because of the high complexity. Besides, from the perspective of optimality, the more the number of cluster tools, the closer the lower bound of 1-unit cyclic scheduling problem is to the optimal solution; even in the two-cluster tool, the difference between the lower bound of 1-unit cyclic scheduling problem and the optimal solution is merely 5.26%.

Table 3.2 Comparison of NPI-NLMIP Model and R-MPI-NLMIP Model on performance

| | CPU time (second) | | | FP | | |
|---|---|---|---|---|---|---|
| Number of cluster tool | MPI-NLMIP model | R-MPI-NLMIP model | Gap | MPI-NLMIP model | R-MPI-NLMIP model | Gap |
| 2 | 0.16 | 0.06 | 62.50% | 19 | 18 | 5.26% |
| 3 | 0.31 | 0.08 | 74.19% | 26 | 26 | 0.00% |
| 4 | 1.33 | 0.23 | 82.71% | 26 | 26 | 0.00% |
| 6 | 15.71 | 0.44 | 97.20% | 27 | 27 | 0.00% |
| 8 | 29.98 | 1.58 | 94.73% | 33 | 33 | 0.00% |
| 10 | 131.12 | 11.58 | 91.17% | 33 | 33 | 0.00% |
| 12 | 683.52 | 19.23 | 97.19% | 35 | 35 | 0.00% |
| 20 | - | 709.46 | - | - | 29 | - |

In conclusion, the R-MPI-NLMIP model based on MPI-NLMIP model is established in this section. According to the experimental results, the R-MPI_NLMIP model is stable and the solution that obtained by CPLEX software can be used as the lower bound of 1-unit cyclic scheduling problem.

## 3.4 MPI-NLMIP-based two-stage approximate-optimal scheduling algorithm

In order to ensure the feasibility of the schedule and improve the speed of computation, we propose a two-stage approximate-optimal scheduling algorithm based on MPI-NLMIP model in this section, which is called MNB (MPI-NLMIP-based) algorithm.

**Definition 3.1**: *If* $t^L_{P,a,b}$ *satisfies the equation:* $t^L_{P,a,b} = \max_{i\in[1,I],j\in[1,J]} t^L_{P,i,j}$, *where* $a \in [1, I]$ *and* $b \in [1, J]$*; then,* $M_{a,b}$ *is the bottleneck PM of multi-cluster tool (BP), denotes as* $BP$.

The MNB algorithm uses the parameter $S_{i,j}$ to describe the complete process of wafer fabrication in the multi-cluster tool, including the current residency time of wafer in the processing module, the buffer module and the transport module. Thus, the operation status of each module of the multi-cluster tool is known. In the MNB algorithm, the key parameters $S_{i,j}$ must be in a feasible interval, thus to explore the potential to minimize the FP.

### 3.4.1 Core idea and process of MNB algorithm

The MNB algorithm is divided into the initial feasible scheduling space stage and the approximate-optimal scheduling stage. In the initial feasible scheduling space stage, the first step is to determine the schedule of bottleneck module; then search for the schedule of other modules and robots; after that, check the feasibility of schedule based on the constraints of MPI-NLMIP model. If it is not feasible, i.e., there are resource conflict, adjust the $S_{i,j}$ under the premise of satisfying residency constraints, thus changing the current residency time of wafer, and obtain a feasible schedule. The main process of MNB algorithm in the initial feasible scheduling space stage consists

of initialization, bottleneck processing module positioning, initial scheduling, inspection and adjustment. In the approximate-optimal scheduling stage, current residency time is treated as a time block. Compared the feasible solution with the lower bound present in this chapter, then slide the time block to approximate the lower bound, i.e., search for the approximate-optimal solution in the feasible solution space. The approximate-optimal scheduling stage contains two steps, the verification and improvement, and schedule output.

In multi-cluster tools, the wafer waits in the cassette module, and then enters the system one by one in a predetermined order. According to the definition of FP, in one-wafer flow pattern, the time interval between any adjacent two wafers entering the system is FP. Therefore, we only need to schedule the $0^{th}$ wafer, and then we can know the status of wafers in the multi-cluster tool at any time, and then master the status of each module in the multi-cluster tool.

## 3.4.2 Steps of MNB Algorithm

Figure 3.6 shows the steps of MNB algorithm in details.

1) Initial feasible scheduling space stage

Step 1: In the initialization phase, according to the wafer flow, we code the cassette module, processing module and buffer module uniformly. For example, in a three-cluster tool, if there are four processing modules in each cluster tool. Start from the cassette module, the wafer passes through module as follows: $\{ B_{0,1}, M_{1,1}, M_{1,2}, B_{1,2}, M_{2,1}, M_{2,2}, B_{2,3}, M_{3,1}, M_{3,2}, M_{3,3}, M_{3,4}, B_{3,2}, M_{2,3}, M_{2,4}, B_{2,1}, M_{1,3}, M_{1,4}, B_{1,0} \}$; the corresponding position is marked as follows: $\{ P_{0,1}, P_{1,1}, P_{1,2}, P_{0,2}, P_{2,1}, P_{2,2}, P_{0,3}, P_{3,1}, P_{3,2}, P_{3,3}, P_{3,4}, P_{3,0}, P_{2,3}, P_{2,4}, P_{2,0}, P_{1,3}, P_{1,4}, P_{1,0} \}$; and the corresponding number for each position is: $[0,17]$.

Besides, the count parameter $Count$ and the optimal FP are also initialized.

Step 2: search and position the BP. According to the definition of the BP, search all the processing time ($t^L_{P,i,j}$) to find out the processing time of BP, denote as $temp$, i.e., $temp = \max t^L_{P,i,j}$, and the position of BP is denoted by $BP \leftarrow P_{i,j}$.

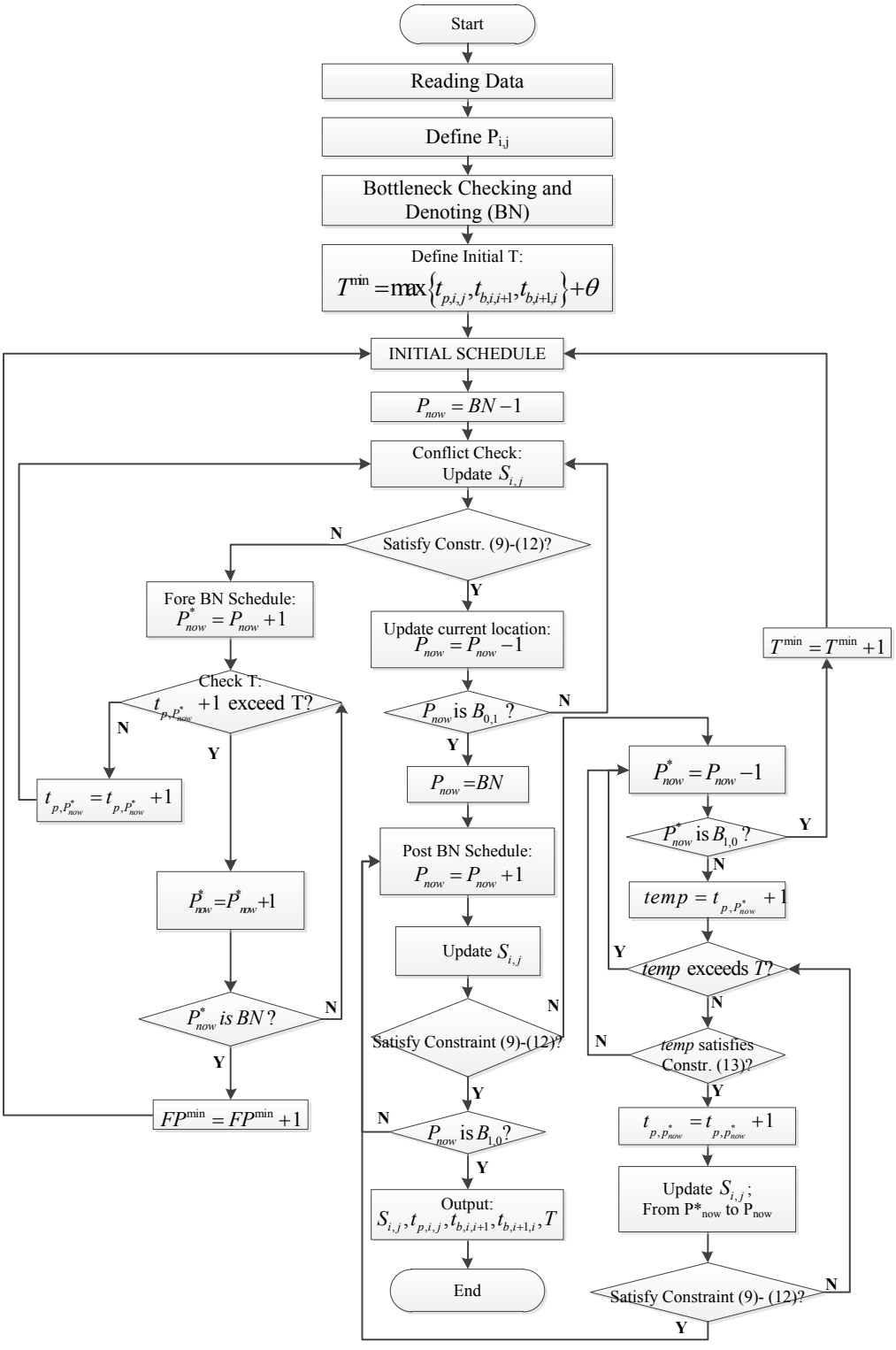



Figure 3.6 Flow chart of MNB heuristic algorithm

Step 3: initial schedule of the $0^{th}$ wafer. Since the scheduling objective in this chapter is to minimize FP, shortening the residency time has a direct effect on minimizing FP, so we assume that the current residency time of the wafer in the

processing module and the buffer module is minimal, i.e., $t_{P,i,j} \leftarrow t^{L}_{P,i,j}$, $t_{B,i,i+1} \leftarrow 0$, $t_{B,i+1,i} \leftarrow 0$. It can be inferred that the minimum FP is $T^0 \leftarrow \max_{i\in[1,I];j\in[1,J]} t^{L}_{p,i,j} + \theta$. According to the constraints (3-2) to (3-6) and (3-13) to (3-15), the feasible interval of $S_{i,j}$, denoted as $\overline{S}_{i,j} \leftarrow \left[ S^{\min}_{i,j}, S^{\max}_{i,j} \right]$, and the initial schedule of 0-th wafer is denoted as $S^0 \leftarrow \left\{ S_{i,j} \middle| i \in [1,I]; j \in [1,J] \right\}$.

Step 4: check the feasibility of initial schedule. MNB algorithm is based on MPI-NLMIP model, so a feasible schedule needs to satisfy all the constraints of the MPI-NLMIP model. In order to test the feasibility of the initial schedule obtained in the previous step, we introduce constraints (3-9) to (3-12). When any constraint is not satisfied, we first adjust $S_{i,j}$ in the feasible interval $\overline{S}_{ij}$, that is, increase one unit of time ($S_{i,j} \leftarrow S_{i,j} + 1$); if $S_{i,j}$ exceeds the feasible interval, the current minimum FP is increased by one unit time ($T \leftarrow T + 1$). This adjustment phase is divided into two parts, the fore-bottleneck part adjustment and the post-bottleneck part adjustment. Both parts start from the smallest position number to the maximum. After each time of adjustment, we return to the adjustment phase to test the feasibility of the new schedule until all the constraints are met, to obtain the feasible schedule (denoted as $S^{'}$).

At this point, the initial feasible scheduling space stage is completed.

2) Approximate-optimal scheduling stage

Step 5: evaluate the difference of the feasible schedule and lower bound proposed in this chapter, and find the approximate-optimal schedule. In order to check verify whether $S^{'}$ is satisfied, we introduce the lower bound of 1-unit cyclic scheduling problem as benchmark (denoted as $T^*$). If the ration of $T^*$ to the minimum FP (denoted as $T^{'}$) corresponding to $S^{'}$ is less than 95% (can be set according to the needs), then let $T^* \leftarrow T$ and return to initial schedule (step 3), so as to adjust the schedule. In order to prevent deadlock, we will count the number of verification and improvement phase with $Count$. The verification and improvement phase will be conducted only if $Count < 10$ (can be set according to the needs).

Step 6: output the approximate-optimal schedule obtained by MNB algorithm, including $S_{i,j}$ and $T$.

### 3.4.3 TMs scheduling

By using the MNB algorithm, we have obtained the wafer's departure time at each module ($S_{i,j}$) and the approximate-optimal solution of FP ($T$). It is now not hard to complete the scheduling of the robots in transport modules. It is worth noting that in the use of MNB algorithm scheduling multi-cluster tool, we take all the constraints associated with the robot into account, Therefore, constraint (3-16) to (3-21) must satisfy the constraint. In other words, the schedule of robot moves must be feasible.

The $R_i$ unloads the $w$-th wafer from $M_{i,j}$ at the following time:

$$t_{R,i,j}^{w,u} = w \times T + S_{i,j};\ 1 \le i \le I;\ 1 \le j \le J\ . \tag{3-16}$$

The $R_i$ loads the $w$-th wafer to $M_{i,j+1}$ at the following time:

$$t_{R,i,j+1}^{w,s} = w \times T + S_{i,j} + \theta;\ 1 \le i \le I;\ 1 \le j \le J-1\ . \tag{3-17}$$

The $R_i$ loads the $w$-th wafer to $B_{i,i+1}$ and $B_{i+1,i}$ at the following time, respectively:

$$t_{BR,i,i+1}^{w,s} = w \times T + S_{i,x} + \theta;\ 1 \le i \le I-1\ . \tag{3-18}$$

$$t_{BR,i+1,i}^{w,s} = w \times T + S_{i+1,J} + \theta;\ 1 \le i \le I-1\ . \tag{3-19}$$

The $R_i$ unloads the $w$-th wafer from $B_{i,i+1}$ and $B_{i+1,i}$ at the following time, respectively:

$$t_{BR,i,i+1}^{w,u} = w \times T + S_{i,x} + \theta + t_{B,i,i+1};\ 1 \le i \le I-1\ . \tag{3-20}$$

$$t_{BR,i+1,i}^{w,u} = w \times T + S_{i+1,J} + \theta + t_{B,i+1,i} \quad 1 \le i \le I-1\ . \tag{3-21}$$

This completes the scheduling of wafers and all modules of multi-cluster tools.

## 3.5 Simulation and experimental analysis

The goal in this chapter is to minimize the FP under the premise that the schedule is feasible. By definition, the FP is the time interval at which the two adjacent wafers arrive at the cassette module. In 1-unit cyclic schedule, the time interval at which any two wafers arrive is the same. FP is a very important indicator of the throughput of a multi-cluster tool. The comparison of FP is a comparison of throughput, because low FP is a necessary condition for achieving high throughput. In order to evaluate the model and algorithm established in this chapter effectively, we will analyze the influence of two key factors on the performance from the two aspects of the CPU time and FP. The two key factors are the multi-cluster tools of different structures and the processing time of the wafers with different distributions. The experiments presented in this section aim to verify the effectiveness of the MPI-NLMIP model and evaluate the performance of the MNB algorithm.

The MPI-NLMIP model, the R-MPI-NLMIP model and the MNB algorithm are programmed in C ++ in Microsoft Visual Studio 2010 on a PC on a PC with a 2.53 GHz Intel Core TM i3 CPU. The CPLEX Optimizer is embedded into the program. The following experimental results are the average of ten identical experiments.

### 3.5.1 CPU time

1) The MPI-NLMIP model

In order to verify the feasibility of the MPI-NLMIP model proposed in this chapter, we model the multi-cluster tools of seven different structures and solve the models with CPLEX software respectively. In this experiment, the number of cluster tool in multi-cluster tools ranges from 2 to 12, and each cluster tool consist of 4 processing modules. Wafer processing time and upper bound of current residency time are subject to normal distribution, the specific parameters are shown in table 3.3

The experimental results are shown in figure 3.7. When the scale of the multi-cluster tool is less than 10 cluster tools, the CPU time is short; when the number of cluster tools in a multi-cluster tool is between 10 and 12, the CPU time increases significantly.

If the number of cluster tools continues to increase to more than 12, CPU time increases rapidly, resulting to that the CPLEX cannot find optimal solution in the polynomial time. That is, the complexity of the corresponding MPI-NLMIP model increases and it becomes hard to solve the model with CPLEX in a reasonable time.

Therefore, the method of solve the MPI-NLMIP model with CPLEX is applicable to the number of cluster tool between 2 to 12.

Table 3.3 List of parameters for MPI-based non-linear MIP Model verification experiment

| Group No. | $I$ | $J$ | $x$ | $t^{L}_{P,i,j}$ | $t^{U}_{P,i,j}$ | $\theta$ |
|---|---|---|---|---|---|---|
| 1 | 2 | 4 | 2 | N(15,5) | N(30,5) | 3 |
| 2 | 3 | 4 | 2 | N(30,2) | N(40,2) | 3 |
| 3 | 4 | 4 | 2 | N(30,2) | N(40,5) | 3 |
| 4 | 6 | 4 | 2 | N(20,1) | N(30,5) | 3 |
| 5 | 8 | 4 | 2 | N(20,1) | N(30,10) | 3 |
| 6 | 10 | 4 | 2 | N(20,5) | N(30,5) | 3 |
| 7 | 12 | 4 | 2 | N(20,1) | N(30,5) | 3 |

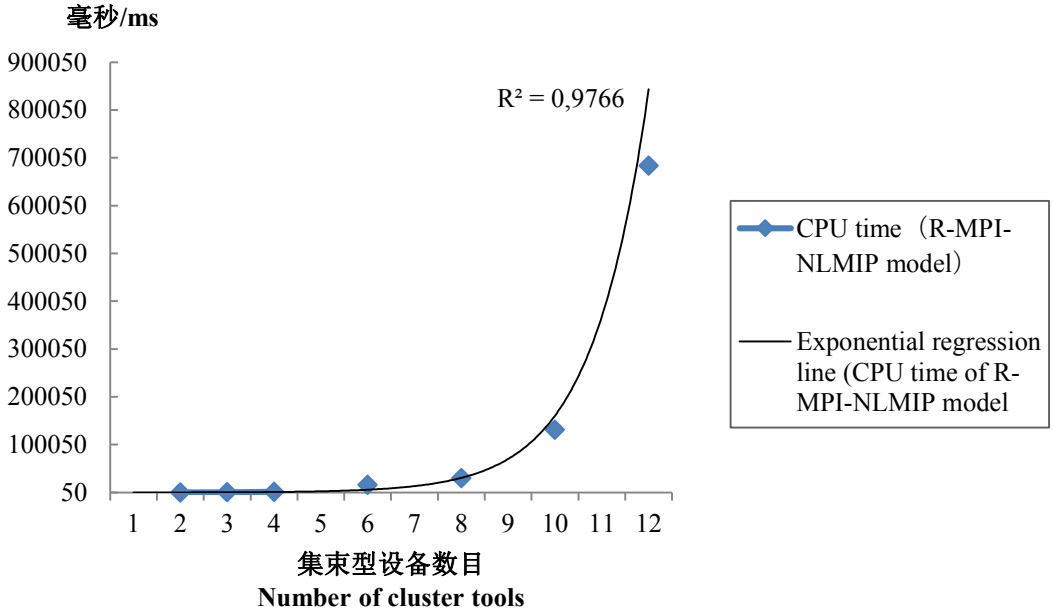



Figure 3.7 CPU time spend on solving MPI-NLMIP Model with CPLEX

2) MNB algorithm

The purpose of this experiment is to test the CPU time that MNB algorithm required to solve the scheduling problem in general cases. The test object is multi-cluster tools of varying scales. In detail, the number of cluster tools in a multi-cluster tool ranges from 2 to 30, and there are 4 processing modules in each cluster tool. The processing time of the wafer in all the test groups is subject to the normal distribution. The experimental results are shown in figure 3.8. As the number of cluster tools in the multi-cluster tool increases, the CPU time increases in a quartic polynomial regression. Especially in the multi-cluster tool consists of more than 25

cluster tools, the CPU time notably increases. However, the MNB algorithm can still solve the scheduling problem of the multi-cluster tool, which consists of less than 30 cluster tools, in the relatively short CPU time.

In contrast to the results of 1) and 2), the computation time of the MNB algorithm is much less than that required by CPLEX to solve the MPI-NLMIP model in the same simulation environment. For a clearer comparison of the differences between the two, we introduce the following variables.

$D_{CPU} = (t_{MPI} - t_{MNB}) / t_{MPI} \times 100\%$, the ratio of CPU time difference, which represents the percentage of the difference between the CPU time of the MNB algorithm ($t_{MNB}$) and the time required for the CPLEX to solve the MPI-NLMIP model ($t_{MPI}$). The larger the value, the smaller the $t_{MNB}$ compared to $t_{MPI}$, which means that the performance of the MNB algorithm is better.

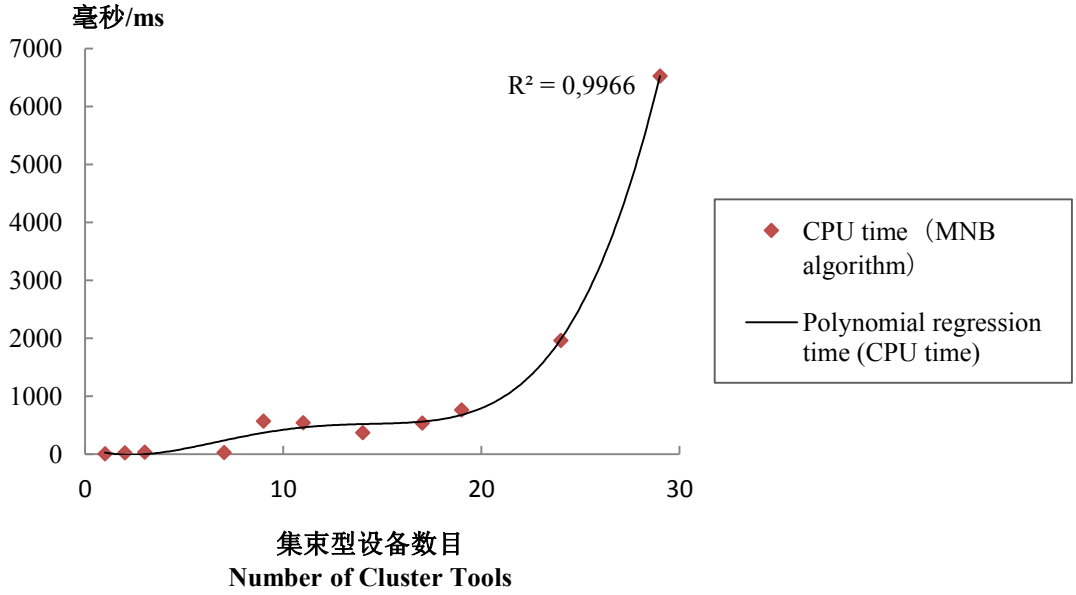



Figure 3.8 CPU time spend on scheduling multi-cluster tools with 2 to 30 clusters with MNB algorithm

As shown in figure 3.9, in the multi-cluster tools, which consist of 2 to 12 cluster tools, $D_{CPU}$ ranges from 92.58% to 99.92%. Then we can deduce that $t_{MNB}$ is much less than $t_{MPI}$. That is to say, the MNB algorithm can make a quick response to meet the practical needs.

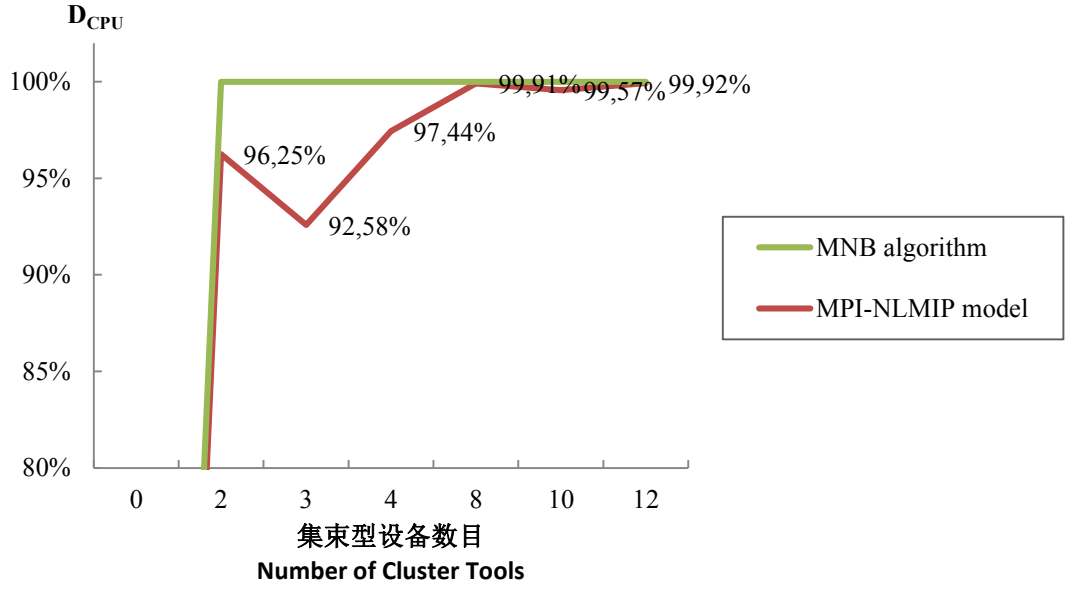



Figure 3.9 Comparison of CPU time spend on solving MPI-NLMIP Model with MNB algorithm and with CPLEX

## 3.5.2 Performance analysis

The purpose of this experiment is to verify the performance of the MNB algorithm. In this experiment, we simulate three types of multi-cluster tools: 6-cluster tool, 12-cluster tool and 20-cluster tool. The CPU time required for find the minimum FP with MNB algorithm is test when the processing time of the wafer satisfies the normal distribution or even distribution. Then, under the same experimental environment, the experimental results are compared with the CPU time that required using CPLEX to solve the R-MPI-NLMIP model and the lower bound of 1-unit cyclic scheduling problem. In order to evaluate the algorithm effectively, we introduce the following evaluation criteria.

$D_{FP} = (T_{MNB} - T_{R-MPI}) / T_{MNB} \times 100\%$, the ration of FP difference, which stands for the percentage of the difference between the minimum FP obtained by MNB algorithm ($T_{MNB}$) and the lower bound of 1-unit cyclic scheduling problem obtained by solving the R-MPI-NLMIP model with CPLEX ($T_{R-MPI}$). The smaller the value, the smaller the difference between $T_{MNB}$ and $T_{R-MPI}$, and the better the performance of MNB algorithm.

The data for the experiment are shown in table 3.4.

Table 3.4 Simulation data for MNB algorithm performance analysis experiment

| Parameter | Value |
|---|---|
| Number of cluster tool | 6,12,20 |
| Number of PM in a cluster tool | 4 |
| Transporting time (second) | 3 |
| Wafer processing time | Normal distribution, even distribution |
| Upper bound of wafer residency time | Normal distribution, even distribution |

As shown in Table 3.5, when the wafer processing time follows a normal distribution, the $D_{CPU}$ is more than 85% and the $D_{FP}$ is within 12%. In particular, in the case where the number of cluster tools is 12, the $D_{CPU}$ is 99% or more, and the $D_{FP}$ reaches 0%. With the increase of the number of cluster tools, the $D_{CPU}$ increases first and then decreases, but the $D_{FP}$ decreases first and then increases, indicating that the MNB algorithm has the best performance when the number of cluster tools is about 12. When the wafer processing time is uniformly distributed, the $D_{CPU}$ is between 70% and 98%, and the $D_{FP}$ is between 3% and 19%. As in the case of the normal distribution, the $D_{CPU}$ increases and then decreases as the number of cluster tools increases, while the $D_{FP}$ decreases then increases as the number of cluster tools increases, and the MNB algorithm performs the best when the multi-cluster tool consists of 12 cluster tools.

Then, we compare the results horizontally. When the number of cluster tools is the same, the $D_{FP}$ is lower in the case that wafer processing time is normally distributed, compared with the case that the wafer processing time is uniformly distributed. Only the last case is an exception. The $D_{CPU}$ is higher in the case that wafer processing time obeys the normal distribution than in the case that wafer processing time obeys uniform distribution, and only the second case is an exception. Therefore, we can deduce that the MNB algorithm performs well in both experimental conditions, the wafer processing time obeys normal distribution and the wafer processing time obeys uniform distribution; and the performance of MNB algorithm is better under the experimental conditions of the wafer processing time is normally distributed.

Table 3.5 MNB algorithm performance analysis: compared to lower bound of 1-unit cyclic scheduling problem

| CT | Wafer processing time | Upper bound of wafer processing time | CPU time (Second) | | | FP | | |
|---|---|---|---|---|---|---|---|---|
| | | | $T_{R-MPI}$ | $T_{MNB}$ | $D_{CPU}$ | $T_{R-MPI}$ | $T_{MNB}$ | $D_{FP}$ |
| 6 | N(20,1) | N(30,5) | 0.48 | 0.046 | 89.58% | 27 | 30 | 10% |
| | N(20,5) | N(40,10) | 0.53 | 0.054 | 90.57% | 33 | 34 | 3% |
| 12 | N(20,5) | N(30,2) | 20.16 | 0.173 | 99.16% | 39 | 39 | 0% |
| | N(20,10) | N(35,15) | 19.44 | 0.08 | 99.59% | 53 | 53 | 0% |
| 20 | N(15,5) | N(20,5) | 114.16 | 16.046 | 85.94% | 33 | 35 | 6% |
| | N(20,1) | N(30,5) | 566.66 | 13.437 | 97.63% | 29 | 33 | 12% |
| 6 | U(5,15) | U(5,30) | 3.54 | 0.592 | 83.33% | 25 | 31 | 19% |
| | U(5,15) | U(5,60) | 13.87 | 0.297 | 97.83% | 25 | 30 | 17% |
| 12 | U(5,30) | U(5,40) | 11.58 | 0.321 | 97.24% | 36 | 37 | 3% |
| | U(20,35) | U(20,70) | 21.98 | 2.536 | 88.44% | 41 | 43 | 5% |
| 20 | U(5,30) | U(5,40) | 76.72 | 19.72 | 74.30% | 36 | 40 | 10% |
| | U(20,35) | U(20,70) | 100.5 | 29.23 | 70.89% | 36 | 40 | 10% |

## 3.5.3 Case study

In this section, a three-cluster tool in the lithography area of wafer fabrication is taken as an example. The purpose of case study is to use MNB algorithm to find the minimum FP and the corresponding optimal schedule of robots.

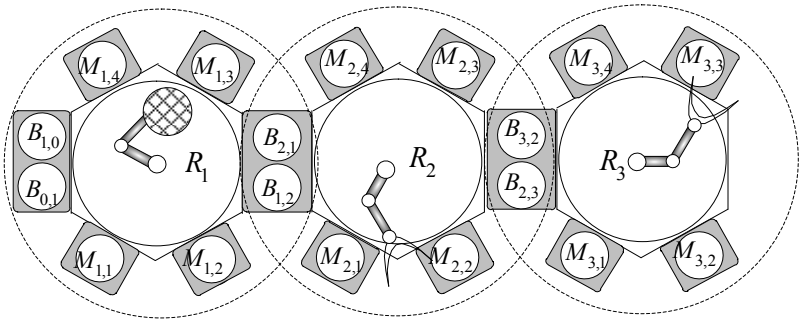



Figure 3.10 Schematic views of three-cluster tools and wafer flow

As shown in figure 3.10, the multi-cluster tool consists of three cluster tools, each of which has four processing modules, and the adjacent cluster tools are connected by two buffer modules. The relevant experimental data are shown in table

3.6 and table 3.7. Here, "$a+b\left(a\in N^{*},b\in N^{*}\right)$" indicates that the processing time for the wafer is $a$, and $b$ is the time that the wafer continues to reside on the module after the process is completed. So, $a+b$ is equal to the current residency time for the wafer.

Table 3.6 Simulation data and the operation results of MNB algorithm before " Check and Improve" step

| Current residency time | $M_{i,1}$ | $M_{i,2}$ | $M_{i,3}$ | $M_{i,4}$ | Current residency time | $B_{i,i+1}$ | $B_{i+1,i}$ |
|---|---|---|---|---|---|---|---|
| $C_1$ | 6 | 20 | 14 + 7 | 19 | $C_1$ | 13 | - |
| $C_2$ | 16 | 13 | 6 + 5 | 14 | $C_2$ | 13 | 15 |
| $C_3$ | 16 | 13 + 1 | 6 + 5 | 14 + 4 | $C_3$ | - | 25 |

In order to reflect the necessity of the "Verification and improvement" phase in the MNB algorithm, the experimental results shown in table 3.6 are the results before this phase is executed, and table 3.7 is the result finally output.

Table 3.7 Simulation data and final operation results of MNB algorithm

| Current residency time | $M_{i,1}$ | $M_{i,2}$ | $M_{i,3}$ | $M_{i,4}$ | Current residency time | $B_{i,i+1}$ | $B_{i+1,i}$ |
|---|---|---|---|---|---|---|---|
| $C_1$ | 6 | 20 | 14 + 6 | 19 | $C_1$ | 0 | - |
| $C_2$ | 16 | 13 + 3 | 6 + 4 | 14 | $C_2$ | 23 | 0 |
| $C_3$ | 16 + 3 | 13 | 6 + 4 | 14 + 3 | $C_3$ | - | 23 |

It is known that the lower bound of 1-unit cyclic scheduling problem is 26 and the CPU time is 14 milliseconds. Without the "Verification and improvement" step, the minimum FP obtained by the MNB algorithm is 28; while, the minimum FP of the MNB algorithm with "Verification and improvement" step is 26, which is the same as the lower bound of 1-unit cyclic scheduling problem. In other words, with the "Verification and improvement" step, the minimum FP is 7.14% less. Thus, the "Verification and improvement" step dose have effect on improving the performance of the MNB algorithm.

According to the schedule obtained by the MNB algorithm with and without the "Verification and improvement" step, we draw the Gantt chart separately. Figure 3.14 shows the schedule obtained by the MNB algorithm without the "Verification and improvement" step, and figure 3.15 is the schedule obtained by the MNB algorithm. As can be seen from the two figures, both schedules are conflict-free and satisfy all the constraints of the MPI-NLMIP model. Therefore, both schedules are feasible.

## 3.6 Summary

This chapter addresses the 1-unit cyclic scheduling problem of multi-cluster tools with residency constraints. With objective of minimum FP, MPI is introduced for describing the infeasible solution space, which is caused by residency constraints and resource conflicts. Thus, a nonlinear mixed-integer programming model based on MPI is proposed and solved with CPLEX. Based on this, by experimental method, we establish the lower bound of 1-unit cyclic scheduling problem that discussed in this chapter. In order to solve large-scale problem, this chapter also designs a heuristic algorithm based on MPI-NLMIP model, the MNB algorithm. The proposed algorithm use MPI-NLMIP model to eliminate the infeasible solution space, and uses the bottleneck-based search method to find the approximate-optimal solution of the scheduling problem. The approximate-optimal scheduling stage is designed to improve the quality of the solution.

The experimental results verify the feasibility and efficiency of the proposed model and algorithm in the following aspects. First, the MPI-NLMIP model can accurately describe the scheduling problems studied in this chapter, the use of CPLEX can be solved in a reasonable CPU time. Secondly, MNB algorithm is fast, the difference between the minimum FP and the lower bound that established in this chapter is less than 19%, thus, MNB algorithm is applicable to practical production. Thirdly, although the equipment load distribution is extremely uneven, MNB algorithm still can get a satisfactory approximate-optimal solution, and the performance of the MNB algorithm is optimal in the case of 12 cluster tools. Fourthly, the approximate-optimal scheduling stage of the MNB algorithm does have some help to improve the quality of solution. The schedule obtained by the MNB algorithm is feasible and without resource conflict.

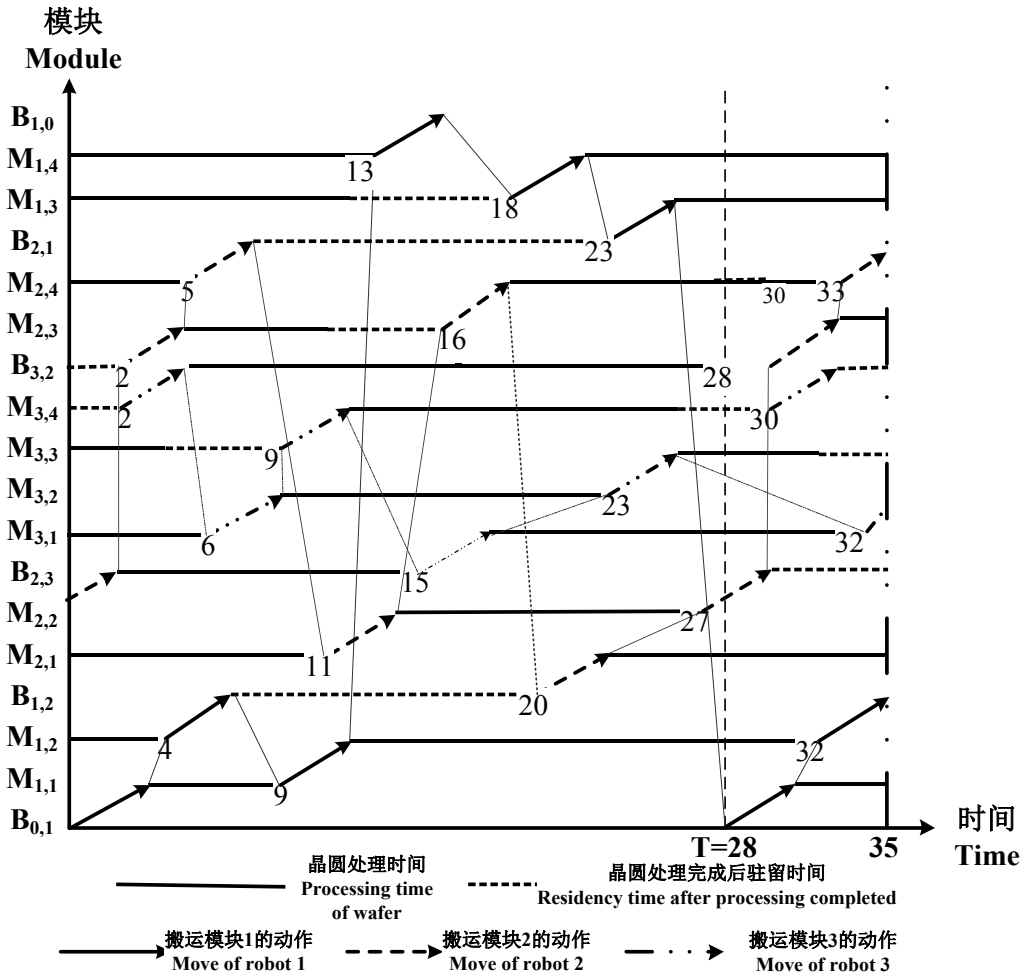



Figure 3.11 The Gantt chart of schedule obtained by MNB algorithm before "Verification and Improvement" step

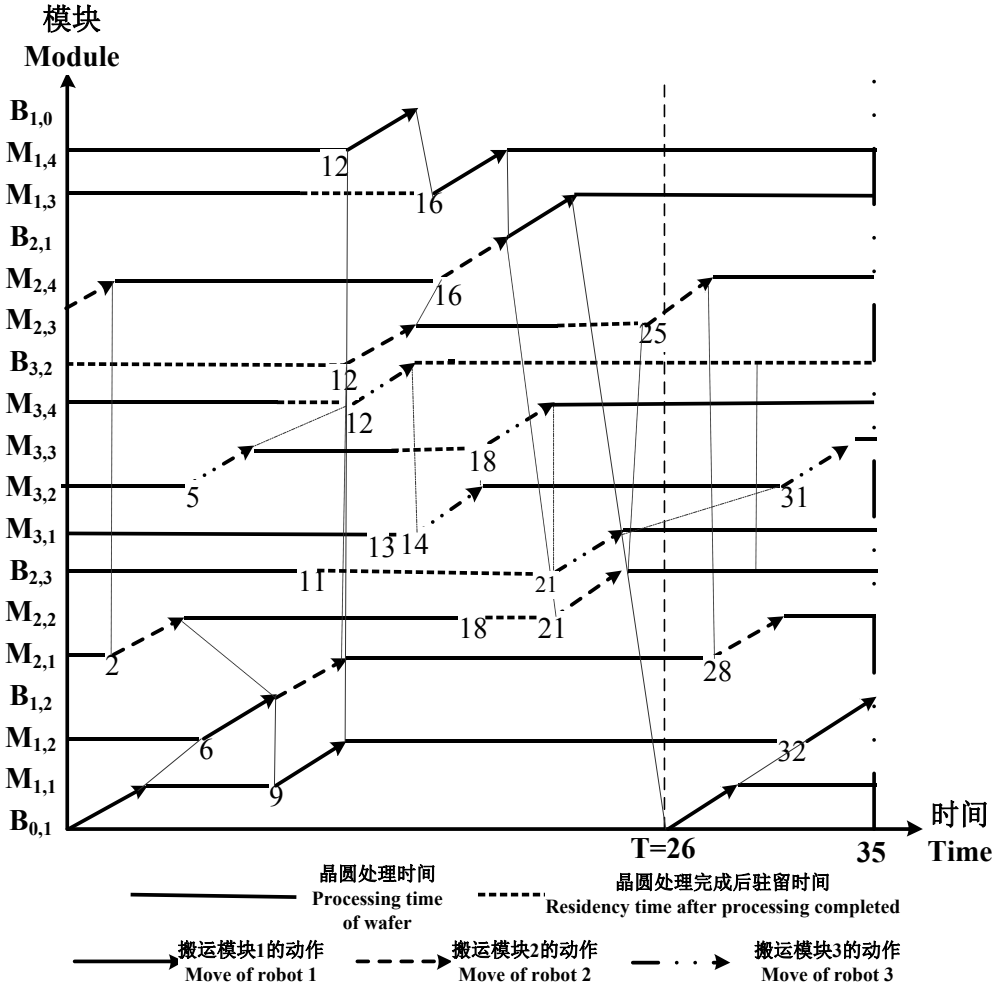



Figure 3.12 The Gantt chart of final schedule obtained by MNB algorithm

# Chapter 4 Research on Multi-unit cyclic scheduling problem

Multi-unit cyclic production are gradually used in wafer fabrication system for improve the efficiency of cyclic production. This chapter discusses the multi-unit cyclic scheduling problem of multi-cluster tools with residency constraints. This chapter establishes a 2-unit cyclic scheduling model with objective of minimum FP and solves the proposed model with CPLEX. Based on Chaos theory, a chaos-based Hybrid PSO-TS optimization algorithm is put forward. Finally, the feasibility of the model and algorithm are verified and the performance of model and algorithm are analyzed by simulation experiments. This work is published in Wang et al., 2015 [117]

## 4.1 Problem description

This chapter focuses on the 2-unit cyclic scheduling problem of multi-cluster tools. As shown in figure 4.1, red and blue are used to distinguish between two varieties of wafers in a FP, which have same wafer flow pattern. The assumptions regarding the structure of the multi-cluster tool, the moves of the robot transport module, the processing time and the residency constraint are as follows:

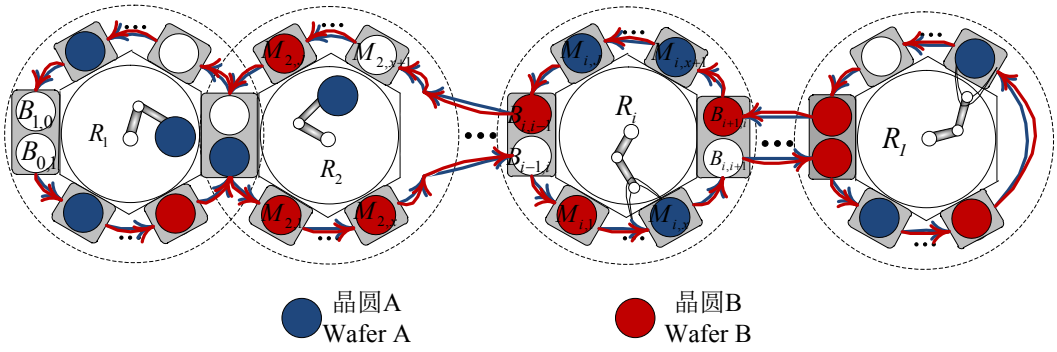



Figure 4.1 Schematic view of 2-degree cyclic production

(1) the multi-cluster tool is symmetrically arranged;
(2) each cluster tool is connected with one or two other cluster tools;
(3) two adjacent cluster tools are connected through two buffer modules;
(4) all the transport modules are single-armed robots, for each robot, the unloading time is equal to the loading time, and the transporting time between modules is assumed to be constant;

(5) the process must begin as soon as the wafer is loaded in the processing module;
(6) for each processing module, only one wafer can be loaded and processes at a time;
(7) each robot can handle one wafer at a time;
(8) the capacity of buffer module is one;
(9) residency constraints is considered, i.e., there is upper bound of current residency time for each processing module, after the processing is completed, the wafer would be defective or scrapped if it resides on the processing module longer than the upper bound of residency constraint;
(10) 2-unit cyclic production is considered in this chapter, and the two wafers are not identical but have the same wafer flow pattern. Wafers arrive at the cassette module in batches (or lots), and enter the multi-cluster tool for processing one by one according to predetermined order. The wafer cannot skip any module.

Assumptions (1) to (9) are the same as chapter 3. It can be seen from the above assumptions that the multi-cluster tool studied in this chapter is still symmetrical linear structure, regardless of the tree structure of the multi-cluster tool. Buffer module is connected to the adjacent cluster tool as a channel. The wafer cannot skip the buffer module, and the transportation of the wafer between the cluster tools must pass through the buffer module. The buffer module does not have the function of processing wafers, so the buffer module does not have upper bound of residency constraints. All the robots are single-arm manipulator. The robot transporting time is a small and constant. Wafer starts processing immediately after it arrives at the processing module without waiting. The capacity of PM, TM and BM is one, i.e., each of which can only handle one wafer at a time. It is worth noting that the residency constraints are considered in this chapter, too.

In addition to the above assumptions, we make a new assumption (10), which is that this chapter addresses the scheduling problem of 2-unit cyclic production. By definition, the two-unit cyclic production refers to the fact that exact two wafers enter the multi-cluster tool in one FP, and that exact two wafers leave the multi-cluster tool in one FP. We record the two wafers in each FP as wafer A and wafer B, respectively. The processing times of wafer A and wafer B may be different. When the wafers arrived at CM in batches, they have been arranged in the established order. For example, if the total number of wafers in CM is $2W$, start from the first wafer A (denoted as $A^1$), wafers leave CM and enter the multi-cluster tool for processing

according to the following order: $\left\{A^1, B^1, A^2, B^2, \cdots, A^w, B^w, \cdots, A^W, B^W\right\}$, among which, $A^w$ represents the wafer A of the $w$-th batch, where $1 \le w \le W$. Similarly, $B^w$ is denoted as the wafer B of the $w$-th batch, where $1 \le w \le W$. Without loss of generality, in 2-unit cyclic production, the 0-th batch of wafers is assumed as the first batch. Thus, the wafer A and wafer B of 0-th batch is denoted as $A^0$ and $B^0$, respectively. If the time when $A^0$ leaves the cassette module is counted as 0, the time when $B^0$ leaves the cassette module is recorded as $T_B$, then the time when $A^w$ from the cassette module is $wT$ and the time that $B^w$ leaves the cassette module is $wT + T_B$. And so on, according to the order of wafers above-mentioned, the corresponding time when each wafer leaves cassette module is as follows: $\{T, T+T_B, \cdots, wT, wT+T_B, \cdots, WT, WT+T_B\}$, where $0 < T_B < T$ [111].

According to the assumptions (6) to (8), the optimal solution of proposed problem in this chapter must satisfy the following three categories of constraints:

1) Machine constraints: each PM can process one wafer at a time, each BM can store one wafer at a time.
2) Transport constraints: each TM can handle one wafer at a time.
3) Residency constraints: wafers must satisfy the residency constraints in the PMs, but there are no residency constraints to wafers in BMs.

In summary, the problem studied in this chapter is how to coordinate the sequence and time of the moves of each robot while satisfying the various constraints, to find the optimal 2-unit cyclic schedule, ultimately reaching the objective of minimizing the FP and maximizing the throughput of the multi-cluster tool.

## 4.2 A non-linear mixed-integer programming model

### 4.2.1 Notations and variables

In order to be able to describe the mathematical model clearly, first, we define a series of notations and variables. This chapter uses the naming rules for variables and

notations that are similar to the previous chapter. In the following paragraphs, we will illustrate the notations and variables from easy to complex.

As in previous chapter, we use the two-dimensional code to define the relevant notations of CMs, PMs and BMs. Due to the change of the wafer variety, this chapter introduces the superscript and subscript to define the variables that is relevant to the wafer processing time and residency time. The superscript is used to distinguish the wafer type and domain of time. The subscript is used to locate the wafer location. For instance, $t_{p,i,j}^{A,L}$ indicates the lower bound of residency time of wafer A that required on $M_{i,j}$, where $L$ represents the lower bound of time. In other word, $t_{p,i,j}^{A,L}$ represents the processing time of wafer A on $M_{i,j}$.

This chapter also refers to the variable $S_{i,j}$, but in order to distinguish between the wafer A and the wafer B, we have also used the superscript and subscript to define the variable. The superscript is used to represent the type of wafer, and the subscript is used to locate the wafer position, such as $S_{i,j}^{A}$ is the time for $A^0$ to leave the $M_{i,j}$. In steady-state，the time interval between the time at which a wafer leaves (or enters) the multi-cluster tool and the time of the same wafer of next batch leaving (or entering) the multi-cluster tool if constant, i.e., the constant time interval is the fundamental period. Therefore, $wT + S_{i,j}^{A}$ is denoted as the time at which the wafer A of $w$-th batch leaves $M_{i,j}$.

This chapter defines some new variables and notations as follows:

$S_{i,j}^{A}$ The time at which wafer A of 0-th batch leaves $M_{i,j}$;

$S_{i,j}^{B}$ The time at which wafer B of 0-th batch leaves $M_{i,j}$;

$A^w$ The wafer A of $w$-batch, where $w \in [0, W]$;

$B^w$ The wafer B of $w$-batch, where $w \in [0, W]$;

$t_{p,i,j}^{A}$ The current residency time of $A^0$ in $M_{i,j}$;

$t_{p,i,j}^{B}$ The current residency time of $B^0$ in $M_{i,j}$;

$t_{B,i,i+1}^{A}$ The current residency time of $A^0$ in $B_{i,i+1}$;

$t^{B}_{B,i,i+1}$ The current residency time of $B^0$ in $B_{i,i+1}$;

$t^{A}_{B,i+1,i}$ The current residency time of $A^0$ in $B_{i+1,i}$;

$t^{B}_{B,i+1,i}$ The current residency time of $B^0$ in $B_{i+1,i}$;

$t^{A,L}_{p,i,j}$ The processing time of $A^0$ in $M_{i,j}$;

$t^{B,L}_{p,i,j}$ The processing time of $B^0$ in $M_{i,j}$;

$t^{A,U}_{p,i,j}$ The upper bound of residency time of $A^0$ in $M_{i,j}$;

$t^{B,U}_{p,i,j}$ The upper bound of residency time of $B^0$ in $M_{i,j}$;

$T$ The FP of a batch of wafers;

$T_B$ The time at which $B^0$ leaves $B_{0,1}$;

### 4.2.2 Objective function

As mentioned earlier, the objective function of this chapter is to minimize the FP, namely:

$$\min T \tag{4-1}$$

### 4.2.3 Calculate the leaving time of wafer A and B on each PM

$$S_{i,j}^{A}=\begin{cases}\sum_{m=1}^{i-1}\sum_{n=1}^{x}\left(\theta+t_{P,m,n}^{A}\right)+\sum_{n=1}^{j}\left(\theta+t_{P,i,n}^{A}\right)+\sum_{m=1}^{i-1}\left(\theta+t_{B,m,m+1}^{A}\right); & 1<i<I;j\le x \quad (4-2)\\ \sum_{n=1}^{j}\left(\theta+t_{P,1,n}^{A}\right); & i=1;j\le x \quad (4-3)\\ \sum_{n=1}^{j}\left(\theta+t_{P,i,n}^{A}\right)+\sum_{m=1}^{I-1}\left(\theta+t_{B,m,m+1}^{A}\right)+\sum_{m=i}^{I-1}\left(\theta+t_{B,m+1,m}^{A}\right)+\sum_{m=i+1}^{I}\sum_{n=1}^{J}\left(\theta+t_{P,m,n}^{A}\right)+\sum_{m=1}^{i-1}\sum_{n=1}^{x}\left(\theta+t_{P,m,n}^{A}\right); & 1<i<I;j>x \quad (4-4)\\ \sum_{m=2}^{I}\sum_{n=1}^{J}\left(\theta+t_{P,m,n}^{A}\right)+\sum_{n=1}^{j}\left(\theta+t_{P,1,n}^{A}\right)+\sum_{m=1}^{I-1}\left(2\theta+t_{B,m+1,m}^{A}+t_{B,m,m+1}^{A}\right); & i=1;j>x \quad (4-5)\\ \sum_{m=1}^{I-1}\sum_{n=1}^{x}\left(\theta+t_{P,m,n}^{A}\right)+\sum_{n=1}^{j}\left(\theta+t_{P,I,n}^{A}\right)+\sum_{m=1}^{I-1}\left(\theta+t_{B,m.m+1}^{A}\right); & i=I \quad (4-6)\end{cases}$$

$$S_{i,j}^{B}=T_B+\begin{cases}\sum_{m=1}^{i-1}\sum_{n=1}^{x}\left(\theta+t_{P,m,n}^{B}\right)+\sum_{n=1}^{j}\left(\theta+t_{P,i,n}^{B}\right)+\sum_{m=1}^{i-1}\left(\theta+t_{B,m,m+1}^{B}\right); & 1<i<I;j\le x \quad (4-7)\\ \sum_{n=1}^{j}\left(\theta+t_{P,1,n}^{B}\right); & i=1;j\le x \quad (4-8)\\ \sum_{n=1}^{j}\left(\theta+t_{P,i,n}^{B}\right)+\sum_{m=1}^{I-1}\left(\theta+t_{B,m,m+1}^{B}\right)+\sum_{m=i}^{I-1}\left(\theta+t_{B,m+1,m}^{B}\right)+\sum_{m=i+1}^{I}\sum_{n=1}^{J}\left(\theta+t_{P,m,n}^{B}\right)+\sum_{m=1}^{i-1}\sum_{n=1}^{x}\left(\theta+t_{P,m,n}^{B}\right); & 1<i<I;j>x \quad (4-9)\\ \sum_{m=2}^{I}\sum_{n=1}^{J}\left(\theta+t_{P,m,n}^{B}\right)+\sum_{n=1}^{j}\left(\theta+t_{P,1,n}^{B}\right)+\sum_{m=1}^{I-1}\left(2\theta+t_{B,m+1,m}^{B}+t_{B,m,m+1}^{B}\right); & i=1;j>x \quad (4-10)\\ \sum_{m=1}^{I-1}\sum_{n=1}^{x}\left(\theta+t_{P,m,n}^{B}\right)+\sum_{n=1}^{j}\left(\theta+t_{P,I,n}^{B}\right)+\sum_{m=1}^{I-1}\left(\theta+t_{B,m,m+1}^{B}\right); & i=I \quad (4-11)\end{cases}$$

Since the $S_{i,j}^{A}$ and $S_{i,j}^{B}$ in this chapter are defined based on the previous chapter, the calculation method is the same as $S_{i,j}$ in the previous chapter. It is worth noting that compared with $A^0$, $B^0$ was $T_B$ late than the $B_{0,1}$, so unlike $S_{i,j}^{A}$, the calculation of $S_{i,j}^{B}$ must take $T_B$ into account rather than assume the time at which $B^0$ leaves CM is zero. The specific calculation method of $S_{i,j}^{A}$ and $S_{i,j}^{B}$ are shown in constraint (4-2) to (4-6) and constraint (4-7) to (4-11).

### 4.2.4 Machine constraints

According to assumption (6) and (8), each PM can process one wafer at a time, and each BM can temporarily store one wafer at a time. Since the wafer cannot skip any module, the resource conflict caused by the demand for processing module or buffer module can only occur on two adjacent wafers. Thus, in 2-unit cyclic production, the order of the wafers is $\{A^0, B^0, A^1, B^1, \cdots, A^w, B^w\}$. And thus, two wafers that may simultaneously have a demand for a processing module or buffer module are: wafer $A^w$ and wafer $B^w$, or wafer $A^{w+1}$ and wafer $B^w$. In the following, we discuss these two cases separately.

1) Wafer $A^w$ and $B^w$

Wafer $A^w$ and $B^w$ are the same batch of wafers, they enter the PM of multi-cluster tool according to the predetermined sequence, which is $\{A^w, B^w\}$. In this case, the resource conflict may occur in the module where the $A^w$ is located, that is, when the $B^w$ has completed the processing process, waiting to be unloaded and transported to the module where $A^w$ is located, but the $A^w$ has not yet left the module. Based on the different locations of $A^w$ and $B^w$, the case which may cause resource conflict are divided into following six categories.

First, if the $A^w$ is on the $M_{i,j}$ and $B^w$ is on the $M_{i,j-1}$, in order to prevent the occurrence of resource conflicts, the $B^w$ must wait on the $M_{i,j-1}$ before the $A^w$ has finished processing and leaving the $M_{i,j}$. In other words, the $B^w$ must be unloaded and transported to the $M_{i,j}$ after the $A^w$ has left. Then the following constraints can be obtained:

$$S_{i,j}^{A}+\theta \le S_{i,j-1}^{B};i\in[1,I];\quad j\in[2,x]\cup[x+2,J]. \tag{4-12}$$

$$S_{I,x+1}^{A}+\theta \le S_{I,x}^{B}. \tag{4-13}$$

Second, if the $A^w$ is on the $M_{i,1}$ and the $B^w$ is on the $B_{i-1,i}$, as mentioned above, the $B^w$ has to wait on the $B_{i-1,i}$ before the $A^w$ has finished processing and leaving the $M_{i,1}$. There is:

$$S_{i,j}^{A}\le S_{i-1,x}^{B}+t_{B,i-1,i}^{B};i\in[2,I]. \tag{4-14}$$

Third, if the $A^w$ is on the $M_{i,J}$ and the $B^w$ is on the $B_{i+1,i}$, in order to avoid resource conflict, the following constraint must be met.

$$S_{i,j}^{A}\le S_{i+1,J}^{B}+t_{B,i+1,i}^{B};i\in[1,I-1]. \tag{4-15}$$

Fourth, if $A^w$ is on the $M_{1,1}$ and the $B^w$ is on the $B_{0,1}$, i.e., the $A^w$ has just entered the $M_{1,1}$ and the $B^w$ is still waiting on the $B_{0,1}$; then, the time the $B^w$ leaves the $B_{0,1}$ must not be earlier than the time the $A^w$ leaves the $M_{1,1}$. Thus, the following constraint must be satisfied.

$$S_{1,1}^{A}+\theta \le T_B. \tag{4-16}$$

Fifth, if $A^w$ is on the $B_{i-1,i}$ and the $B^w$ is on the $M_{i,x}$, similarly, there is:

$$S_{i-1,x}^{A}+2\theta+t_{B,i-1,i}^{A}\le S_{i-1,x}^{B};i\in[2,I]. \tag{4-17}$$

Sixth, if $A^w$ is on the $B_{(i+1)i}$ and the $B^w$ is on the $M_{i+1,J}$, the following constraint can be obtained.

$$S_{i+1,J}^{A}+2\theta+t_{B,i+1,i}^{A}\le S_{i+1,J}^{B};i\in[1,I-1]. \tag{4-18}$$

2) Wafer $A^{w+1}$ and $B^{w}$

The $A^{w+1}$ and the $B^{w}$ are different batches of wafers, which enter the multi-cluster tool for processing in order of $\left\{B^{w}, A^{w+1}\right\}$. In this case, it is possible that the $A^{w+1}$ and the $B^{w}$ compete for the resource of the module where the $B^{w}$ is located, that is, when the $A^{w+1}$ is waiting for the module where the $B^{w}$ is located and the $B^{w}$ has not yet left, the $A^{w+1}$ and the $B^{w}$ simultaneously issue a demand command to the module where the $B^{w}$ is located. In order to prevent the resource competition, we set up the following seven inequality constraints according to the position of the module where the $B^{w}$ is located.

$$S_{i,j}^{B}+\theta \le S_{i,j-1}^{A}+T;i\in[1,I];\quad j\in[2,x]\cup[x+2,J] \tag{4-19}$$

$$S_{I,x+1}^{B}+\theta \le S_{I,x}^{A}+T. \tag{4-20}$$

$$S_{i,1}^{B} \le S_{i-1,x}^{A}+t_{B,i-1,i}^{A}+T;i\in[2,I]. \tag{4-21}$$

$$S_{i,J}^{B} \le S_{i+1,J}^{A}+t_{B,i+1,i}^{A}+T;i\in[1,I-1]. \tag{4-22}$$

$$S_{1,1}^{B}+\theta \le T. \tag{4-23}$$

$$S_{i-1,x}^{B}+2\theta+t_{B,i-1,i}^{B} \le S_{i-1,x}^{A}+T;i\in[2,I]. \tag{4-24}$$

$$S_{i+1,J}^{B}+2\theta+t_{B,i+1,i}^{B} \le S_{i+1,J}^{A}+T;i\in[1,I-1]. \tag{4-25}$$

## 4.2.5 TMs constraints

Assumptions (7) is about the transport module constraints, which limits the capacity of the robot, that is, a robot can only carry a wafer at a time.

**Theorem 4.1** *In the multi-cluster tool that produces two types of wafers in 2-unit cycle production way, the number of cluster tools is known to be* $I$*, and the number of processing modules in each cluster tool is* $J$*, and the adjacent cluster tool is composed of two buffer modules. Thus, the cluster tool can handle up to* $(x+1)I-1$ *batches of wafers at the same time.*

**Proof.** *In a multi-cluster tool with given values of* $I$ *and* $J$*, there are* $IJ$ *processing modules in total. It is also known that the adjacent cluster tools are connected by two buffer modules, so the total number of buffer modules is* $2(I-1)$*. Thus, the sum of the total number of processing modules and buffer module is* $IJ+2(I-1)$*. And it is known that each processing module, buffer module can only handle one wafer at a time, then this multi-cluster tool can handle up to* $IJ+2(I-1)$ *wafers at the same time. In a two-degree cyclic schedule, two parts enter and leave the system in a cycle, that is, a batch consists of two parts. Therefore, the cluster tool can handle up to* $(x+1)I-1$ *batches of wafers at the same time.*

Based on theorem 4.1, the inequality (4-26) to (4-65) should meet that there is $w\in[1,W]$, where $W=(x+1)I-1$. In addition, in the following inequalities, we assume that variable $j\in[1,J]$ and $k\in[1,J]$, and $j\neq k$.

Depending on the type and batch of two wafers that may cause robot conflicts, we will discuss the following five categories: wafer $A^0$ and $A^w$, wafer $A^0$ and $B^w$, wafer $B^0$ and $A^w$, wafer $B^0$ and $B^w$, wafer $A^0$ and $B^0$.

1) Wafer $A^0$ and $A^w$

Because of the limited robot capacity, the robot can handle only one wafer at a time. In order to avoid the collision of the robot, the time at which the two wafers leave the module is at least not short than the time required for the robot to do a complete move ($\theta$). Thus, constraints (4-26) to (4-33) are established based on the location of the $A^0$ and $A^w$.

If $A^0$ is processed on $M_{i,j}$, and $A^w$ is processed on $M_{i,k}$, then, the $T$ should satisfy the following constraints.

$$\left|S_{i,j}^{A}-S_{i,k}^{A}-wT\right|\geq\theta;\ i\in[1,I];\ k\in[1,J-2];\ j\in[k+2,J]. \tag{4-26}$$

$$\left|S_{i,x+1}^{A}-S_{i,x}^{A}-wT\right|\geq\theta;\ i\in[1,I-1]. \tag{4-27}$$

If $A^0$ is on $M_{i,j}$ and $A^w$ is on $B_{i-1,i}$, then

$$\left|S_{i,j}^{A}-S_{i-1,x}^{A}-t_{B,i-1,i}^{A}-\theta-wT\right|\geq\theta;\ i\in[2,I];\ j\in[2,J]. \tag{4-28}$$

If $A^0$ is on $M_{i,j}$ and $A^w$ is on $B_{i+1,i}$, then

$$\left|S^A_{i,j} - S^A_{I+1,\ J} - t^A_{B,i+1,i} - \theta - wT\right| \geq \theta;\ i \in [1, I-1];\ j \in [x+2, J]. \tag{4-29}$$

If $A^0$ is on $B_{i+1,i}$ and $A^w$ is on $B_{i-1,i}$, then

$$\left|S^A_{i-1,x} + t^A_{B,i-1,i} + wT - S^A_{i+1,J} - t^A_{B,i+1,i}\right| \geq \theta;\ i \in [2, I-1]. \tag{4-30}$$

If $A^0$ is on $B_{i+1,i}$ and $A^w$ is on $M_{i,k}$, then

$$\left|S^A_{i,k} + wT - S^A_{i+1,J} - t^A_{B,i+1,i} - \theta\right| \geq \theta;\ i \in [1, I-1];\ k \in [1, x]. \tag{4-31}$$

If $A^0$ is on $M_{1,j}$ and $A^w$ is on $B_{0,1}$, then

$$\left|S^A_{1,j} - wT\right| \geq \theta;\ j \in [2, J]. \tag{4-32}$$

If $A^0$ is on $B_{2,1}$ and $A^w$ is on $B_{0,1}$, then

$$\left|S^A_{2,J} + \theta + t^A_{B,2,1} - wT\right| \geq \theta. \tag{4-33}$$

2) Wafer $A^0$ and $B^w$

Similarly, in order to avoid robot conflict, we established the constraints (4-34) to (4-43).

$$\left|S^A_{i,j} - S^B_{i,k} - wT\right| \geq \theta;\ i \in [1, I-1];\ k \in [1, x];\ j \in [x+1, J]. \tag{4-34}$$

$$\left|S^A_{i,j} - S^B_{i,k} - wT\right| \geq \theta;\ x \in [4, +\infty);\ i \in [1, I-1];\ k \in [1, x-3];\ j \in [k+3, x]. \tag{4-35}$$

$$\left|S^A_{i,j} - S^B_{i,k} - wT\right| \geq \theta;\ x \in [4, +\infty);\ i \in [1, I-1];\ k \in [x+1, J-3];\ j \in [k+3, J]. \tag{4-36}$$

$$\left|S^A_{i,j} - S^B_{i,k} - wT\right| \geq \theta;\ i = I;\ k \in [1, J-3];\ j \in [k+3, J]. \tag{4-37}$$

$$\left|S^A_{i,j} - S^B_{i-1,x} - t^B_{B,i-1,i} - \theta - wT\right| \geq \theta;\ i \in [2, I];\ j \in [3, J]. \tag{4-38}$$

$$\left|S^A_{i,j} - S^B_{i+1,J} - t^B_{B,i+1,i} - \theta - wT\right| \geq \theta;\ x \in [3, +\infty);\ i \in [1, I-1];\ j \in [x+3, J]. \tag{4-39}$$

$$\left|S^A_{i+1,J} + t^A_{B,i+1,i} - S^B_{i-1,x} - t^B_{B,i-1,i} - wT\right| \geq \theta;\ i \in [2, I-1]. \tag{4-40}$$

$$\left|S^A_{i+1,J} + t^A_{B,i+1,i} + \theta - S^B_{i,k} - wT\right| \geq \theta;\ i \in [1, I-1];\ k \in [1, x]. \tag{4-41}$$

$$\left|S^A_{1,j} - T_B - wT\right| \geq \theta;\ j \in [3, J]. \tag{4-42}$$

$$\left|S^A_{2,J} + \theta + t^A_{B,2,1} - T_B - wT\right| \geq \theta. \tag{4-43}$$

3) Wafer $B^0$ and $A^w$

For wafer $B^0$ and $A^w$, constraints (4-44) to (4-50) are set up as follows.

$$\left|S^B_{i,j} - S^A_{i,k} - wT\right| \geq \theta;\ i \in [1, I];\ k \in [1, J-1];\ j \in [k+1, J]. \tag{4-44}$$

$$\left|S^B_{i,j} - S^A_{i-1,x} - t^A_{B,i-1,i} - \theta - wT\right| \geq \theta;\ i \in [2, I];\ j \in [1, J]. \tag{4-45}$$

$$\left|S^B_{i,j} - S^A_{i+1,J} - t^A_{B,i+1,i} - \theta - wT\right| \geq \theta;\ i \in [1, I-1];\ j \in [x+1, J]. \tag{4-46}$$

$$\left|S^B_{i+1,J} + t^B_{B,i+1,i} - S^A_{i-1,x} - t^A_{B,i-1,i} - wT\right| \geq \theta;\ i \in [2, I-1]. \tag{4-47}$$

$$\left|S^B_{i+1,J} + t^B_{B,i+1,i} + \theta - S^A_{i,k} - wT\right| \geq \theta;\ i \in [1, I-1];\ k \in [1, x]. \tag{4-48}$$

$$\left|S^B_{1,j} - wT\right| \geq \theta;\ j \in [1, J]. \tag{4-49}$$

$$\left|S^B_{2,J} + \theta + t^B_{B,2,1} - wT\right| \geq \theta. \tag{4-50}$$

4) Wafer $B^0$ and $B^w$

In order to prevent the robot conflict that may caused by wafer $B^0$ and $B^w$, the following constraints are proposed.

$$\left|S^B_{i,j} - S^B_{i,k} - wT\right| \geq \theta;\ i \in [1, I];\ k \in [1, J-2];\ j \in [k+2, J]. \tag{4-51}$$

$$\left|S^B_{i,x1} - S^B_{i,x} - wT\right| \geq \theta;\ i \in [1, I-1]. \tag{4-52}$$

$$\left|S^B_{i,j} - S^B_{i-1,x} - t^B_{B,i-1,i} - \theta - wT\right| \geq \theta;\ i \in [2, I];\ j \in [2, J]. \tag{4-53}$$

$$\left|S^B_{i,j} - S^B_{i+1,J} - t^B_{B,i+1,i} - \theta - wT\right| \geq \theta;\ i \in [1, I-1];\ j \in [x+2, J]. \tag{4-54}$$

$$\left|S^B_{i-1,x} + t^B_{B,i-1,i} + wT - S^B_{i+1,J} - t^B_{B,i+1,i}\right| \geq \theta;\ i \in [2, I-1]. \tag{4-55}$$

$$\left|S^B_{i,k} + wT - S^B_{i+1,J} - t^B_{B,i+1,i} - \theta\right| \geq \theta;\ i \in [1, I-1];\ k \in [1, x]. \tag{4-56}$$

$$\left|S^B_{1,j} - T_B - wT\right| \geq \theta;\ j \in [2, J]. \tag{4-57}$$

$$\left|S^B_{2,J} + \theta + t^B_{B,2,1} - T_B - wT\right| \geq \theta. \tag{4-58}$$

5) Wafer $A^0$ and $B^0$

Constraints (4-59) to (4-65) are built for avoiding the competition for robot resources, which may happen between $A^0$ and $B^0$.

$$\left|S_{i,j}^{A}-S_{i,k}^{B}\right|\geq\theta;i\in[1,I];k\in[1,J-1];j\in[k+1,J]. \tag{4-59}$$

$$\left|S_{i,j}^{A}-S_{i-1,x}^{B}-t_{B,i-1,i}^{B}-\theta\right|\geq\theta;i\in[2,I];j\in[1,J]. \tag{4-60}$$

$$\left|S_{i,j}^{A}-S_{i+1,J}^{B}-t_{B,i+1,i}^{B}-\theta\right|\geq\theta;i\in[1,I-1];j\in[x+1,J]. \tag{4-61}$$

$$\left|S_{i-1,x}^{B}+t_{B,i-1,i}^{B}-S_{i+1,J}^{A}-t_{B,i+1,i}^{A}\right|\geq\theta;i\in[2,I-1]. \tag{4-62}$$

$$\left|S_{i,k}^{B}-S_{i+1,J}^{A}-t_{B,i+1,i}^{A}-\theta\right|\geq\theta;i\in[1,I-1];k\in[1,x]. \tag{4-63}$$

$$\left|S_{1,j}^{A}-T_{B}\right|\geq\theta;j\in[1,J]. \tag{4-64}$$

$$\left|S_{2,J}^{A}+\theta+t_{B,2,1}^{A}-T_{B}\right|\geq\theta. \tag{4-65}$$

### 4.2.6 Residency constraints

The residency constraint limits the wafer to stay on the processing module for a sufficient period to complete the process while preventing damage to the wafer due to excessive residency. The scheduling problem of the multi-cluster tool studied in this chapter takes into account the important characteristics of the wafer fabrication process of residency constraints. This is a prerequisite for accurately describing a multi-cluster tool with a mathematical model. As a module for connecting adjacent cluster tools, the buffer module does not process wafers. It is used only for temporary storage and transfer of wafers. Therefore, the buffer module is not constrained by the residency constraint. From the above description, we established the constraints (4-66) to (4-71).

$$t_{P,i,j}^{A}\in\left[t_{P,i,j}^{A,L},t_{P,i,j}^{A,U}\right];i\in[1,I];j\in[1,J]. \tag{4-66}$$

$$t_{P,i,j}^{B}\in\left[t_{P,i,j}^{B,L},t_{P,i,j}^{B,U}\right];i\in[1,I];j\in[1,J]. \tag{4-67}$$

$$t_{B,i-1,i}^{A}\in[0,+\infty);i\in[2,I]. \tag{4-68}$$

$$t_{B,i+1,i}^{A}\in[0,+\infty);i\in[1,I-1]. \tag{4-69}$$

$$t_{B,i-1,i}^{B}\in[0,+\infty);i\in[2,I]. \tag{4-70}$$

$$t_{B,i+1,i}^{B}\in[0,+\infty);i\in[1,I-1]. \tag{4-71}$$

To summarize, the NLMIP model is established with objective function (4-1) and constraints (4-2) to (4-71) in this section. The proposed NLMIP model is used to

describe the scheduling problem addressed in this chapter, which is the 2-unit cyclic scheduling problem of multi-cluster tools with residency constraints.

### 4.2.7 Complexity analysis of Proposed NLMIP model

An analysis of the complexity of the proposed NLMIP model in terms of the number of variables and the number of constraints is given in this section. Based on the definition of the decision variables, the established model consists of $K\left[I(J+2)-1\right]$ variables, where $K$ the number of unit. In this chapter, $K$ is equal to 2. Therefore, the number of variables in the MIP model established in this chapter is a quadratic function of $I$ and $J$, where $I$ represents the number of cluster tools and $J$ represents the number of processing modules in each cluster tool.

Then, we analyse the MIP model from the perspective of the number of constraints. There are 70 constraints in the MIP model, where the constraints (4-2) to (4-11) are definitions of $S_{i,j}^{A}$ and $S_{i,j}^{B}$, with a total of $2IJ$. The constraints (4-12) to (4-25) represent the machine constraints, the total number of which is $2I(J+2)-4$. The constraints (4-26) to (4-65) are based on the constraints of transport module, it contains $16IJ+(x-16)I-8J-(x+4)$ constraints. Residency constraints of a total of $2IJ+4(I-1)$, involving constraints (4-66) to (4-71). Therefore, the MIP model has $22IJ+(x-8)I-8J-(x+3)$ constraints in total, i.e., the total number of constraints of the proposed MIP model is also a quadratic function of $I$ and $J$.

## 4.3 Case study

In order to verify the validity of the NLMIP model established in section 4.2 of this chapter, we use the 3-cluster tool in lithography area of wafer fabrication as a case and solve the model using IBM ILOG CPLEX Optimization Studio 12.2 software. As shown in figure 4.2, the multi-cluster tool consists of three single cluster tools, and each single cluster tool consists of four processing modules. The adjacent cluster tools are connected with two buffer modules. Wafers arrive at $B_{0,1}$ in batches, and wait for enter the multi-cluster tool to processing.

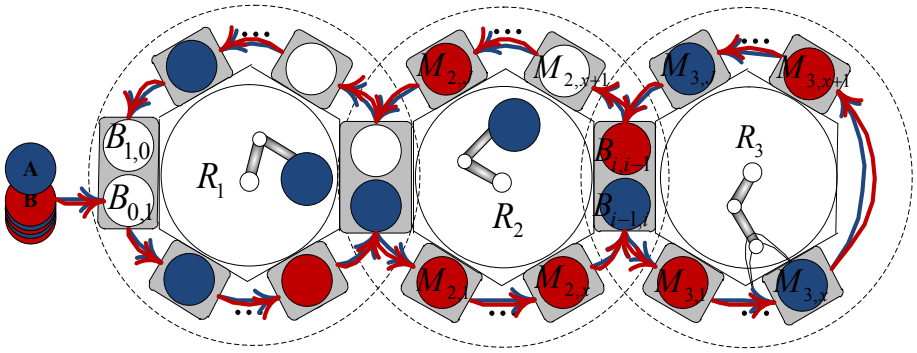



Figure 4.2 Schematic views of three-cluster tools and wafer flow of 2-degree cyclic production

In this experiment, the processing time and the upper bound of residency constraint of wafer A are respectively subject to normal distributions $t_{P,i,j}^{A,L} \sim N(15,5)$ and $t_{P,i,j}^{A,U} \sim N(20,5)$; the processing time and the upper bound of residency constraint of wafer B are respectively subject to normal distribution $t_{P,i,j}^{B,L} \sim N(10,5)$ and $t_{P,i,j}^{B,U} \sim N(16,5)$. The computations were performed on a PC with a 2.53 GHz Intel Core TM i3 processor. CPLEX software uses branch and cut algorithm to solve the MIP model used for 2.34 seconds, wafer B enters the first processing module at $T_B = 27$, the minimum FP is $T = 57$. According to the experimental results, the minimum FP is exactly the same as the lower bound of FP that obtained by the branch and cut algorithm. The specific schedule is shown in table 4.1.

To better illustrate the feasibility of the MIP model, we show the schedule in table 4.1 as a Gantt chart. As shown in figure 4.3, the vertical axis of the Gantt chart represents the processing module and the buffer module through which the wafer passes, and the horizontal axis represents the time. The thick lines and thick dashed lines in the figure represent the processing times of wafers A and B. The solid line with an arrowhead represents the moves of $R_1$, the broken line with an arrowhead is the moves of $R_2$, and the double-dashed line with an arrowhead indicates the moves of $R_3$. The solid line shows the current residency time after the processing is completed.

Table 4.1 Schedule of three-cluster tools case obtained by CPLEX

| Wafer | Module | Unloading time | Loading time | Processing time | Current residency time |
|---|---|---|---|---|---|
| A | $M_{1,1}$ | 0 | 3 | 13 | 0 |
| | $M_{1,2}$ | 16 | 19 | 9 | 5 |
| | $B_{1,2}$ | 33 | 36 | 0 | 0 |
| | $M_{2,1}$ | 36 | 39 | 21 | 0 |
| | $M_{2,2}$ | 60 | 63 | 24 | 0 |
| | $B_{2,3}$ | 87 | 90 | 0 | 0 |
| | $M_{3,1}$ | 90 | 93 | 20 | 0 |
| | $M_{3,2}$ | 113 | 116 | 10 | 0 |
| | $M_{3,3}$ | 126 | 129 | 12 | 0 |
| | $M_{3,4}$ | 141 | 144 | 7 | 0 |
| | $B_{3,2}$ | 151 | 154 | 0 | 0 |
| | $M_{2,3}$ | 154 | 157 | 4 | 0 |
| | $M_{2,4}$ | 161 | 164 | 14 | 0 |
| | $B_{2,1}$ | 178 | 181 | 0 | 1 |
| | $M_{1,3}$ | 182 | 185 | 16 | 0 |
| | $M_{1,4}$ | 201 | 204 | 21 | 6 |
| | $B_{1,0}$ | 231 | 234 | | |
| B | $M_{1,1}$ | 27 | 30 | 6 | 0 |
| | $M_{1,2}$ | 36 | 39 | 10 | 0 |
| | $B_{1,2}$ | 49 | 52 | 0 | 28 |
| | $M_{2,1}$ | 80 | 83 | 7 | 0 |
| | $M_{2,2}$ | 90 | 93 | 15 | 0 |
| | $B_{2,3}$ | 108 | 111 | 0 | 5 |
| | $M_{3,1}$ | 116 | 119 | 12 | 0 |
| | $M_{3,2}$ | 131 | 134 | 10 | 0 |
| | $M_{3,3}$ | 144 | 147 | 8 | 0 |
| | $M_{3,4}$ | 155 | 158 | 7 | 0 |
| | $B_{3,2}$ | 165 | 168 | 0 | 1 |
| | $M_{2,3}$ | 169 | 172 | 9 | 0 |
| | $M_{2,4}$ | 181 | 184 | 5 | 8 |
| | $B_{2,1}$ | 197 | 200 | 0 | 25 |
| | $M_{1,3}$ | 225 | 228 | 7 | 0 |
| | $M_{1,4}$ | 235 | 238 | 11 | 0 |
| | $B_{1,0}$ | 249 | 252 | | |

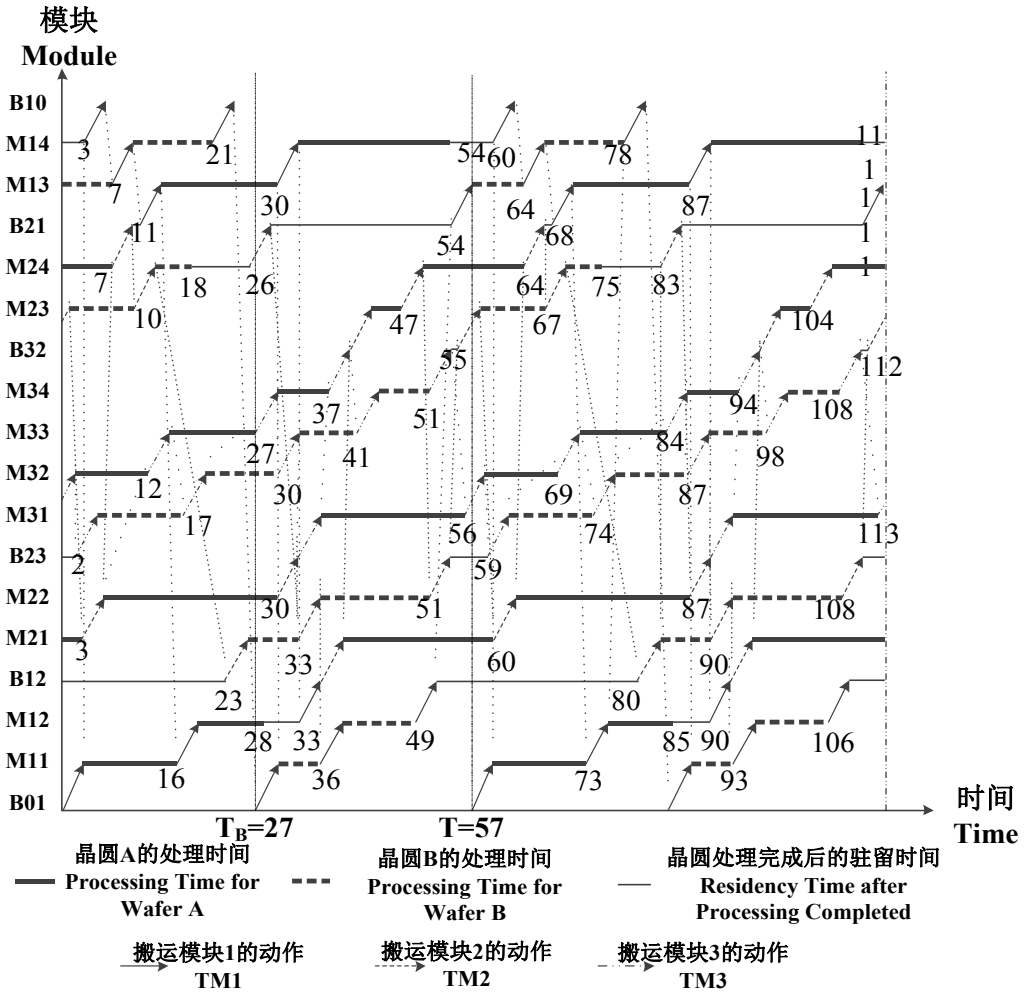



Figure 4.3 Gantt chart of schedule obtained by CPLEX

It can be seen from figure 4.3 that the schedule has no resource conflict and satisfies the residency constraint, which is a feasible schedule. Based on the above analysis, we can conclude that the MIP model can accurately describe the problems studied in this chapter and the model is effective.

## 4.4 A chaos-based hybrid PSO-TS heuristic algorithm

The basic particle swarm optimization (PSO) has the characteristics of strong searching ability and short convergence time. The core idea is to use the self-information, the individual extreme information and the global extreme value information to determine the iterative position of the next step of the particle. In the process of iteration, the particle approaches the optimal direction of the global history, so as to achieve the purpose of optimization. If the self-information and individual extreme information are dominant in the iterative process, the particle swarm will

move closer together. Therefore, the PSO algorithm is easy to fall into a local optimum solution.

In order to improve the ability of PSO to get rid of the extreme points effectively, Chaotic search technology is introduced in this chapter, aiming at improve the accuracy of algorithm by using the characteristics of easy to jump out of the local optimal solution. At the same time, this chapter also introduces a tabu list with memory ability. By recording the local optimal points of the searched region, it avoids the circuitous search and improves the convergence speed of the algorithm.

### 4.4.1 Basic particle swarm optimization

The particle swarm optimization is a meta-heuristic algorithm based on group intelligence. The particles in the algorithm are described by the position $X_k$ and velocity $V_k$. Each particle represents a possible solution to the problem. The velocity of the particle determines the direction and distance of its motion. The velocity is dynamically adjusted according to its own position and the motion trajectory of the other particles. The position of the particle changes with the velocity of the particle, so as to realize the search of the particle in the solution space.

First, initializes a group of random particles; then, the optimal solution is searched by iteration, and the particle is updated in each iteration by tracking the local optimal solution $p_i$ found by the particle itself and the global optimal solution currently found by the whole population $g$. The specific formula is as follows:

$$V_{k+1} = V_k + C_1 random()(p_k - X_k) + C_2 random()(g - X_k) \tag{4-72}$$

$$X_{k+1} = X_k + V_{k+1} \tag{4-73}$$

Where $C_1$ and $C_2$ represent the cognitive coefficients of the population and $k$ represents the number of iterations.

### 4.4.2 Chaotic search technology

Chaos is a stochastic state of motion obtained from deterministic equations [108]. Chaotic state is a common phenomenon in nonlinear systems. Chaos has the characteristics of randomness, ergodicity and regularity. Chaotic search technology

uses the above characteristics to optimize the search in the solution space. Taking the Logistics mapping as an example, the control parameter $\mu$ is expressed in equation (4-74), and when $\mu = 4$ and $0 \le X_0 \le 1$ are satisfied, the chaotic system is in a completely chaotic state.

$$Z_{k+1} = \mu Z_0 (1 - Z_k), \quad i = 1, 2, \cdots, \mu \in [2, 4] \tag{4-74}$$

Through a carrier-like approach, the chaos search technique introduces chaos into the optimization variables to present the chaotic state, and then searches the particles in the local area by adding a small amount of disturbance until the termination rule is satisfied.

### 4.4.3 Tabu list of Tabu search

The tabu list is a flexible memory technique used in tabu search algorithms that can record the optimization process that has been performed to guide the next search direction. In the process of chaotic disturbances, the neighborhood of the approximate-optimal solution may overlap with the searched region, leading to the roundabout search of the particles in the same region. In order to avoid the occurrence of this phenomenon, the tabu list is introduced in this chapter. The tabu list records the path of the particles in the last several iterations. If the particles in the chaotic state are in the tabu list, then the current iteration process is rejected.

### 4.4.4 The core idea and process of chaos-based hybrid PSO-TS algorithm

The algorithm proposed in this chapter is to introduce chaotic initialization, chaotic disturbance and tabu list on the basis of particle swarm optimization. The core idea of the proposed algorithm is to use the ergodicity of the chaotic motion to produce a large number of groups, which are the initial groups of the algorithm. In the iterative process, the chaotic perturbation is added to jump out of the local optimal solution and the infeasible solution is recorded by the tabu list. The specific algorithm flow chart is shown in Figure 4.4.

1) Chaos initialization refers to the use of chaotic sequence to initialize the particle position and velocity. At the beginning of the initialization process, a set of chaotic variables with the same number of optimization variables are generated. Then, by using the chaos technique, the chaotic variables are adjusted in the appropriate range of the optimization variables, so as to

improve the diversity of the population and the ergodicity of the particle search under the premise of preserving the randomness of the initial particle swarm. The particle group produced by chaotic initialization has properties of approximate-optimal solution.

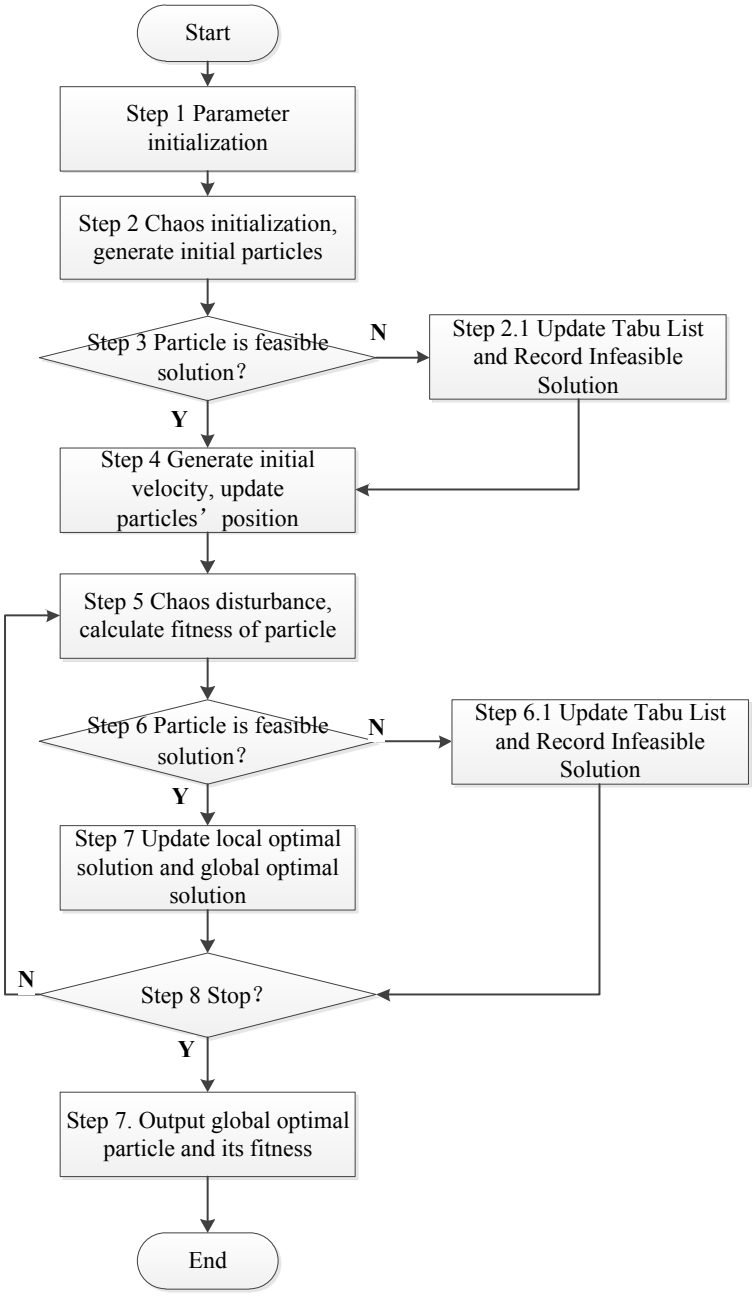



Figure 4.4 The basic flow of Chaos-based Hybrid PSO-TS heuristic algorithm

2) The chaotic disturbance is to determine the disturbance quantity according to the relevant parameters of each particle after the completion of the chaotic initialization, and to disturb the particle swarm to search a local optimum

solution in the neighborhood of the approximate-optimal solution. After several iterations, the particles are gradually close to an optimal solution.

3) The tabu list records the infeasible domain of solutions. When the particle is judged to be infeasible, the tabu list will record it; or in the search process, the particles fall into the infeasible domain of the search that has been completed, the tabu list will help to make a quick decision based on the record and refuse to repeat the search for the same area.

### 4.4.5 Algorithm design

1) The encoding of particles

In this chapter, we study the multi-unit cyclic scheduling problem of multi-cluster tools with residency constraints. One of the main difficulties is that the processing time of the wafer can be arbitrarily changed under the premise of satisfying the residency constraint. In view of this feature, this chapter chooses the current residency time of the wafer as the optimization variable in the process of modeling. The hybrid PSO-TS algorithm based on chaos search technique is used to search the approximate-optimal residency time of the wafer in each processing module and buffer module to obtain a satisfactory solution to the problem.

Based on the above discussion, this chapter uses the optimization variable (the current residency time of the wafer) as the position vector of the particle. The contents of this chapter are two-unit cyclic scheduling problem of multi-cluster tools. The residency time of the wafer includes the residency time of wafer A and wafer B in the processing module and buffer module, and the residency time of wafer B in the cassette module since wafer A enters the first processing module. Thus, the optimization variable after the iteration can be expressed as $X_k = (x_{k0}, x_{k1}, \cdots, x_{kN}) = \left(T^B, t^{A,k}_{p,1,1}, \cdots, t^{A,k}_{p,i,j}, \cdots, t^{A,k}_{p,I,J}, t^{B,k}_{p,1,1}, \cdots, t^{B,k}_{p,i,j}, \cdots, t^{B,k}_{p,I,J}\right)$, and the dimension of the particle is equal to one plus two times of the total number of PMs and BMs. The optimization variable must satisfy the dwell time constraint.

According to the above analysis we can see that the motion of particles in N-dimensional target search space can be regarded as the optimal search of N-dimensional solution space.

2) Chaos initialization

Using the ergodicity of chaotic motion, chaos initialization is adopted to generate the initial particles in a wider space, so as to improve the quality of the individual and

the efficiency of the algorithm. The basic process of chaotic initialization is divided into four steps.

First, an $N$-dimensional vector is randomly generated as an initial value, where the value range of each component of the vector falls between 0 and 1.

Secondly, on the basis of the initial value, $H-1$ $N$-dimensional chaotic variables are generated by the formula (4-74), then the above $H$ chaotic variables make $H\times N$-dimensional chaotic sequences.

Thirdly, the respective components of the chaotic variables are carried out within the range of the optimization variables according to the following equations (4-75) to (4-77) and according to the actual significance represented by the components. It is worth noting that when the actual meaning of the component is the current residency time of wafer on BMs, the component is carrier to the optimization variable using equation (4-77) since there is no upper bound of residency constraint for BMs. In equation (4-77), $B$ represents a large constant number. Through this step, we obtain $H$ $N$-dimensional initial particle swarm satisfying the residency constraint.

$$x_{0,n}=t_{p,i,j}^{A,L}+\left(t_{p,i,j}^{A,U}-t_{p,i,j}^{A,L}\right)\times Z_n \tag{4-75}$$

$$x_{0,n}=t_{p,i,j}^{B,L}+\left(t_{p,i,j}^{B,U}-t_{p,i,j}^{B,L}\right)\times Z_n \tag{4-76}$$

$$x_{0,n}=B\times Z_n \tag{4-77}$$

Fourthly, the fitness function of each particle in the initial particle swarm is calculated and arranged in descending order. The first $Q$ particles are taken as the initial particle swarm of the iteration; the initial value is taken as the initial velocity value. The position and velocity of the particles are updated according to equations (4-72) and (4-73).

3) Chaotic disturbance

In order to broaden the scope of optimization, help the particle jumps out of the local optimal solution and fly into region near the optimal solution, we introduce the chaotic perturbation in the search for the optimal solution. The chaotic process involves three steps.

First, randomly generate an $N$-dimensional vector with a component value between 0 and 1 as the initial value $u_0=\left(u_{0,1},u_{0,2},\cdots,u_{0,N}\right)$, which is the same as the chaotic initialization, and then generate the initial chaotic sequence $U=\left(u_0,u_1,\cdots,u_{H-1}\right)$ according to Eq. (4-74).

Then, determine the appropriate range of chaotic disturbances $[-\beta,\beta]$. The range of disturbance must not be too small, because small range of disturbance is not conducive to help particle jump out of the local optimal solution. The disturbance range must not be too large for reducing the accuracy of the search. The specific disturbance $\Delta X=(\Delta x_{k,1},\Delta x_{k,2},\cdots,\Delta x_{k,N})$ is calculated from the following formula:

$$\Delta x_{k,n}=-\beta+2\beta u_{k,n} \tag{4-78}$$

Third, update the particle position, calculate the fitness function and contrast. Assuming $X=(x_{k,1},x_{k,2},\cdots,x_{k,N})$ is the current position of the particle, the new position of the particle is $X'=(x_{k,1}+\Delta x_{k,1},x_{k,2}+\Delta x_{k,2},\cdots,x_{k,N}+\Delta x_{k,N})$ after the chaotic disturbance is added to the particle. Comparing the fitness before and after the particle update, if the fitness of the particles after adding the disturbance is better than that of the original particles, the particles in the original position are replaced with the new position particles.

4) Record with tabu list

The effect of the tabu list is to record the searched infeasible solution space, denoted as $\text{infeasible\_list}\equiv\{X_a,X_b,\cdots,X_c\}$, when the particles $X_d$ are judged to be infeasible and deposit into $\text{infeasible\_list}$, update the tabu list and get $\text{infeasible\_list}\equiv\{X_a,X_b,\cdots,X_c,X_d\}$, of which $a\neq b\neq c\neq d$ and $a,b,c,d\in[1,K]$.

5) Calculation of fitness function

As an index to evaluate the performance of individual particles, the fitness function must have the ability to accurately reflect the advantages and disadvantages. The goal of this chapter is to minimize FP, so we directly choose $T$ as the fitness function.

For a given particle $X=(x_{k,1},x_{k,2},\cdots,x_{k,N})$, each component value of particle is fixed, and the meaning of each components is determined. In other word, the current residency time of the wafer in each processing module and the buffer module is determined. For 2-unit cyclic scheduling problem of robotic cells with constant processing time, Che et. al. [111] proved that it can be solve in polynomial time and proposed heuristic algorithm. They used MPI to establish the scheduling problem as a series of prohibited intervals of $T$.

$$T \notin (-\infty, 2\theta) \cup \left\{ \bigcup_{k=1}^{MAX} \bigcup_{p=1}^{N} \bigcup_{q=p+1}^{N+1} \left( \left( t_{c,p-1} - t_{c,q-1} - \theta \right) / k, \left( t_{c,p-1} - t_{c,q-1} + \theta \right) / k \right) \right\} \quad (4\text{-}79)$$

Where $t_{c,n}$ represents the time at which the wafer leaves in the module corresponding to the $n^{\text{th}}$ component, it can be calculated from equations (4-2) to (4-11), and then the upper bound and lower bound of FP can be calculated, too. It should be noted that there may be an intersection between the prohibited intervals in Eq. (4-79), so it is necessary to combine all the prohibited sections into a complete set of prohibited intervals. The fitness is the upper bound of the first prohibited interval for the complete prohibition interval.

6) Feasibility judgment

After calculating the fitness of a particle, each component in the particle is known, and the FP represented by the fitness is known. In order to verify the feasibility of the solution, the formula (4-80) is used to verify the $T_B$. If the $T_B$ satisfies the formula, it is judged as feasible, otherwise it is not feasible and record into the tabu list.

$$\begin{aligned} T_B \notin (-\infty, \theta) \cup & \left\{ \bigcup_{k=0}^{MAX-1} \bigcup_{p=1}^{N} \bigcup_{q=p+1}^{N+1} \left( t_{c,p-1} + \theta - t_{c,q-1} - kT, t_{c,q-1} + \theta - t_{c,p-1} - kT \right) \right\} \\ & \cup \left\{ \bigcup_{k=1}^{MAX} \bigcup_{p=1}^{N} \bigcup_{q=p+1}^{N+1} \left( t_{c,p-1} - \theta - t_{c,q-1} + kT, t_{c,q-1} - \theta - t_{c,p-1} + kT \right) \right\} \cup (T - \theta, +\infty) \end{aligned} \quad (4\text{-}80)$$

7) Algorithm termination condition

In order to prevent the algorithm into the infinite loop, resulting in poor algorithm performance, we need to set the algorithm termination conditions. In this chapter, we use the improvement rate of the global optimal solution in the two iterations and the maximum number of iterations as the termination condition of the algorithm, that is, the algorithm is stopped if anyone of above two conditions is satisfied. In other words, the algorithm terminates immediately if the rate of improvement of the global optimal solution (denoted $g_k, g_{k-1}$) in the two iterations is less than 0.1% twice, or if the number of iterations of the algorithm exceeds the maximum number of iterations (denoted $MAX$).

## 4.5 Simulation and experimental analysis

In order to evaluate the performance of the NLMIP model and algorithm that are proposed in this chapter, we proceed to compare the influence of the structure of the multi-cluster tool, the distribution of wafer processing time and the upper-bound of the residency constraint on the NLMIP model from the two aspects: the CPU time and the optimality of solution. In addition, in this section, we also compare the proposed algorithm with basic PSO, aiming at evaluate the difference between them.

We implemented the algorithm in C ++ in Microsoft Visual Studio 2010 and solve the NLMIP model with CPLEX. As with the case analysis, the simulation environment in this experiment is a personal computer with 320G hard drive, 4GB memory and 2.53GHz frequency Core i3 processor. The following experimental results are the average of nine experiments.

### 4.5.1 CPU time

1) NLMIP model

The experiment on the CPU time required for CPLEX to solve the NLMIP model is divided into two parts: the influence of the number of cluster tools on the CPU time and the influence of the number of processing modules on the CPU time.

First, in the multi-cluster tools as shown in Figure 4.5, the CPU time increases rapidly as the number of cluster tools increases. This is because when the number of cluster tools increases, the number of variables and constraints in the NLMIP model increases, and the complexity of the model is a quadratic function about the number of variables and constraints. Therefore, the increase in the number of cluster tools will cause the rapid increase in the complexity of the NLMIP model, so that the difficulty of CPLEX solving the model increases, the CPU time increases.

Then, we analyze the influence of the number of processing modules on the CPU time. In figure 4.6, this experiment compares five multi-cluster tools of different structures, which are: (1) $I=2, J=2$; (2) $I=2$, $J=4$; (3) $I=3$, $J=4$; (4) $I=3$, $J=6$; (5) $I=4$, $J=6$. In addition, the corresponding numbers of processing modules are 4, 8, 12, 18 and 24. As can be seen from figure 4.5, when the number of processing modules increases, the CPU time increases rapidly. Through the fitting of the data, we can find that the CPU time is a quadratic equation that relates to the number of processing modules.

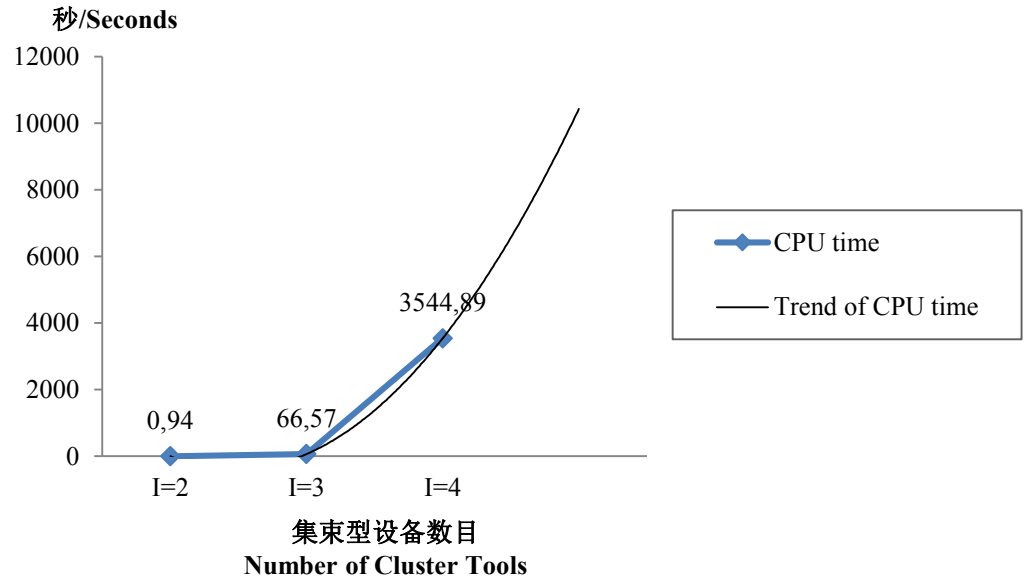



Figure 4.5 Influence of number of cluster tools on CPU time

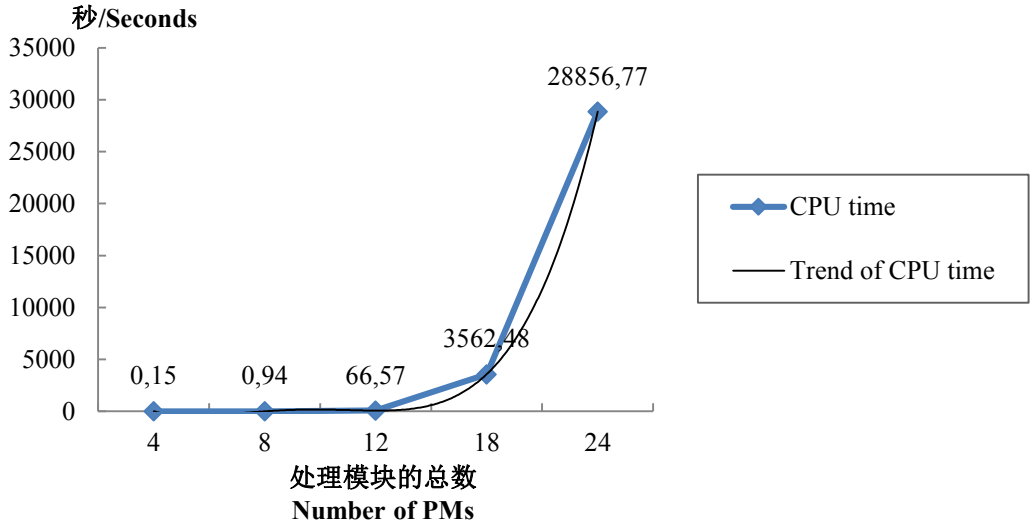



Figure 4.6 Influence of number of BMs on CPU time

The above two experiments are to study the effect of the structure of the multi-cluster tool on the time required for CPLEX to solve the NLMIP model. It can be seen from the experimental results that as the complexity of the structure of multi-cluster tools increases, the CPU time increases and the growth rate increases. This shows that the structure of the multi-cluster tool has a significant effect on the CPU time of the CPLEX solving the MIP model.

2) Chaos-based hybrid PSO-TS heuristic algorithm

The purpose of the simulation experiment is to test the differences between the proposed algorithm and the basic particle swarm algorithm in terms of computation

time. This section takes multi-cluster tools, which consists of 2 to 12 single cluster tools, as examples. In each of single cluster tool, there are 4 PMs. The processing time and upper bound of residency time of the wafers of all test groups were uniformly distributed $t_{P,i,j}^{A,L} \sim U(5,15)$, $t_{P,i,j}^{A,U} \sim U(10,20)$ $t_{P,i,j}^{B,L} \sim U(3,10)$ $t_{P,i,j}^{B,U} \sim U(6,16)$. The total number of initial particle groups is $H = 400$, and the number of excellent groups is $Q = 40$. The population cognitive coefficients are assumed to be $C_0 = 0.85$, $C_1 = 1.59$, and $C_2 = 1.59$. The maximum velocity is $V_{\max} = 1.05$. The maximum number of iterations is set to $MAX = 1000$. The simulation results are shown in Figure 4.7.

As can be seen from Figure 4.7, with the increase of the size of the multi-cluster tools, the computation time of the proposed algorithm is smaller than that of the basic particle swarm algorithm, and this advantage is more significant as the size of the multi-cluster tools is increased.

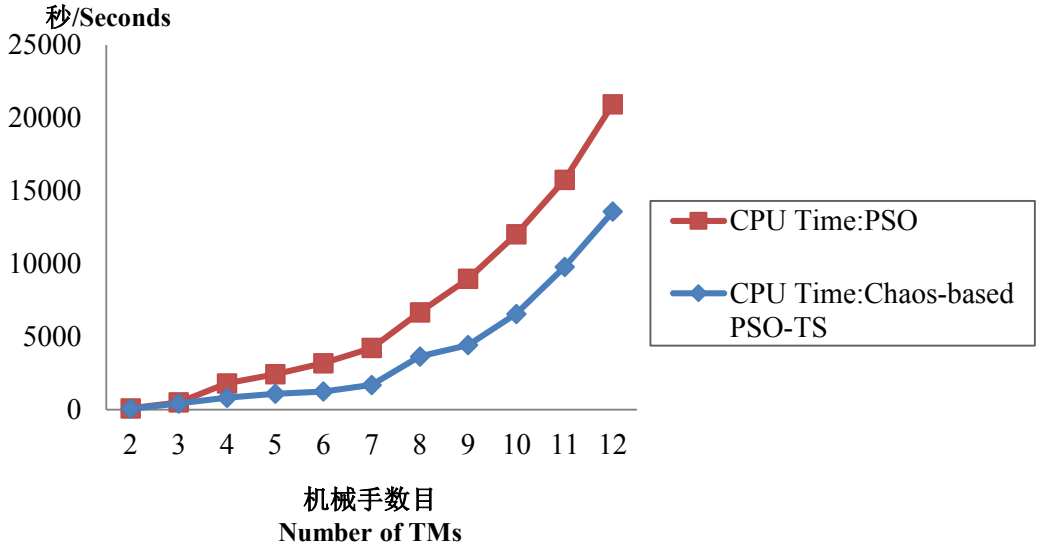



Figure 4.7 Comparison of PSO and Chaos-based hybrid PSO-TS algorithm in aspect of CPU time

## 4.5.2 Performance analysis

1) NLMIP model

In this experiment, we randomly generate multiple types of wafers, and examine the performance of the NLMIP model by comparing the CPU time and minimum FP to analyse the effect of the distribution of the wafer processing time and the residence time upper bound of residency constraints on the NLMIP model.。

In order to evaluate the performance of the NLMIP model better, we introduced the following indicators.

$T_{gap} = (T - T_{LB}) / T_{LB} \times 100\%$, the ratio of the difference of FP, which indicates the percentage of the difference between the minimum FP ( $T$ ) and the lower bound of the non-cyclic scheduling problem ($T_{LB}$). $T$ is the solution that obtained by CPLEX solving the NLMIP model. $T_{LB}$ is the solution that obtained by using branch and cut algorithm. The larger the value, the smaller the difference between $T$ and $T_{LB}$, the better the performance of the NLMIP model.

As shown in table 4.2, we will be divided simulation experiments into four groups based on the distributions of wafer processing time and the upper bounds of residency time; each group corresponds to a distribution. Such as, in the fourth group, The processing time and the upper bound of residency time of the wafer A are subject to a normal distribution, i.e., $t_{P,i,j}^{A,L} \sim N(15,3)$ and $t_{P,i,j}^{A,U} \sim N(20,3)2$, and that of the wafer B follows a uniform distribution, i.e., $t_{P,i,j}^{B,L} \sim U(3,50)$ and $t_{P,i,j}^{B,U} \sim U(10,80)$. The specific data for the wafer processing time and the upper bound of residency time for each group are randomly generated according to the distribution and are related to the structure of the multi-cluster tool used for wafer fabrication. In this section, we consider six different types of multi-cluster tools and establish the NLMIP models corresponding.

As can be seen from table 4.2, the CPU time increases with the complexity of the structure of multi-cluster tools, regardless of whether the wafer process data follows a normal distribution or a uniform distribution.

Table 4.2 Influence of wafer flow and structure of multi-cluster tools on solving NLMIP Model with CPLEX

| Group | Wafer | Distribution | $I$ | 2 | | 3 | | 4 | |
|---|---|---|---|---|---|---|---|---|---|
| | | | $J$ | 2 | 4 | 4 | 6 | 4 | 6 |
| 1 | A | $t_{P,i,j}^{A,L} \sim N(15,5), t_{P,i,j}^{A,U} \sim N(20,5)$ | CPU time (second) | 0.26 | 0.55 | 2.34 | 162.5 | 13714 | 43245 |
| | | | $T_{gap}(\%)$ | 0 | 0 | 0 | 1.54 | 2.78 | 3.42 |
| | B | $t_{P,i,j}^{B,L} \sim N(10,5), t_{P,i,j}^{B,U} \sim N(16,5)$ | $T$ | 46 | 57 | 57 | 65 | 66 | 71 |
| 2 | A | $t_{P,i,j}^{A,L} \sim U(5,15), t_{P,i,j}^{A,U} \sim U(10,20)$ | CPU time (second) | 0.23 | 1.06 | 8.67 | 67.3 | 119 | 12610 |
| | | | $T_{gap}(\%)$ | 0 | 0 | 4.88 | 2.6 | 2.7 | 4.24 |
| | B | $t_{P,i,j}^{B,L} \sim U(3,10), t_{P,i,j}^{B,U} \sim U(6,16)$ | $T$ | 39 | 40 | 41 | 38 | 37 | 40 |
| 3 | A | $t_{P,i,j}^{A,L} \sim U(5,50), t_{P,i,j}^{A,U} \sim U(15,90)$ | CPU time (second) | 0.14 | 0.83 | 2.15 | 153.2 | 13968 | 38627 |
| | | | $T_{gap}(\%)$ | 0 | 0 | 0 | 0 | 0.85 | 1.24 |
| | B | $t_{P,i,j}^{B,L} \sim U(3,50), t_{P,i,j}^{B,U} \sim U(10,80)$ | $T$ | 84 | 90 | 107 | 108 | 106 | 110 |
| Group | Wafer | Distribution | $I$ | 2 | | 3 | | 4 | |
| | | | $J$ | 2 | 4 | 4 | 6 | 4 | 6 |
| 4 | A | $t_{P,i,j}^{A,L} \sim N(15,3), t_{P,i,j}^{A,U} \sim N(20,3)$ | CPU time (second) | 0.17 | 0.87 | 7.5 | 11.8 | 14.49 | 170.3 |
| | | | $T_{gap}(\%)$ | 0 | 0 | 0 | 0 | 0 | 0 |
| | B | $t_{P,i,j}^{B,L} \sim U(3,50), t_{P,i,j}^{B,U} \sim U(10,80)$ | $T$ | 64 | 106 | 82 | 99 | 93 | 109 |

In contrast to the experimental results of the first and second groups in table 4.2, the value of $T_{gap}$ is smaller when the wafer processing data is subject to a normal distribution that is when the data distribution is uniform, indicating that the quality of $T$ is better when the wafer processing data is normally distributed compared to uniformly distributed. From the second and third groups of experiments in table 4.2, it can be concluded that in a cluster tool, greater the difference between the maximum and minimum values of the wafer processing time, the smaller the $T_{gap}$. In other words, the bigger the gap of the uniformly distributed processing data, the higher the quality of $T$. At last, in contrast to the first, third and fourth groups of experiments, we found that the $T_{gap}$ is smaller when wafers A and B were subject to different distributions than they were subject to the same distribution. That is, for the wafer processing data with more complex distribution, the $T$ is closer to the $T_{gap}$, and the NLMIP model performs better.

2) Chaos-based hybrid PSO-TS algorithm

In order to compare the performance of the proposed algorithm and the basic particle swarm algorithm, the simulation experiment is carried out with the solution quality as the measurement. The parameters of basic particle swarm algorithm are the same as those in Section 4.5.1, and the processing time of the wafers the upper bound of residency time are uniformly distributed. The results are shown in the table 4.3, which are average of 20 experiments.

Table 4.3 Comparison of PSO and Chaos-based hybrid PSO-TS algorithm in aspect of the quality of the solution

| Number of TMs | PSO | Chaos-based hybrid PSO-TS algorithm |
|---|---|---|
| 2 | 40 | 40 |
| 3 | 41 | 41 |
| 4 | 41 | 37 |
| 6 | 44 | 42 |
| 10 | 52 | 47 |
| 12 | 60 | 51 |

As can be seen from Table 4.3, the proposed algorithm runs better than the basic particle swarm algorithm. In particular, the advantage of chaos-based hybrid PSO-TS

algorithm in performance is more prominent for the large-scale scheduling problem of multi-cluster tools.

## 4.6 Summary

In this chapter, a non-linear mixed-integer programming model is proposed to minimize the FP, and the complexity of the NLMIP model is established for the modeling and multi-unit cyclic scheduling problem of the multi-cluster tool with multi-wafer types and residency constraints. Based on this, the CPLEX software is used to solve the model, and the validity of the solution and the feasibility of the schedule are verified by case study. This chapter also proposes a hybrid PSO-TS algorithm based on Chaotic search technology, which introduces Chaotic search technology and tabu list into basic particle swarm algorithm to prevent the algorithm from falling into local optimal and circuitous search. The proposed algorithm provides a method for solving the approximate-optimal solution of large-scale problem. The simulation results show that the proposed model and algorithm are well performed, which are embodied in the following aspects. Firstly, the influence of the number of cluster tools and the number of processing modules on the CPU time and the solution is analyzed, and it is found that the NLMIP model is suitable for the multi-cluster tools with the number of single cluster devices not exceeding 20 and the number of robots is not more than 4. Secondly, Secondly, if the multi-cluster tools are small-scale and the wafer processing time is normal distributed or uniform distributed, then a NLMIP scheduling model of 2-unit non-cyclic scheduling problem can be established and solved by CPLEX in a reasonable time. The quality of solution is high, feasible, and resource conflict-free. Thirdly, compared with the basic particle swarm algorithm, the algorithm proposed in this chapter has advantages in terms of computation time and quality of solution. This advantage is more obvious as the problem scale expands.

# Chapter 5 **Research on Non-cyclic Scheduling Problem**

Although the mass cyclic production can achieve the goal of maximizing throughput, but with the popularity of intelligent manufacturing, the increasing demand for ASIC, non-cyclic production under multiple wafer flow patterns also increases in the wafer fabrication. In order to improve the productivity, this chapter discusses the modeling and non-cyclic scheduling of multi-cluster tools that take into account the residency constraints. A mathematical model is built for the above scheduling problem with the objective of minimizing the makespan. The lower bound of the non-cyclic scheduling problem is put forward and proved. Because of the difficulty in find exact solution, based on TOC, we design a bottleneck-based push-pull heuristic scheduling algorithm. Lastly, it is expected to verify the effectiveness of the proposed algorithm by simulation. This work is published in Wang and Zhou 2015 [115].

## 5.1 Problem description

The structure of the multi-cluster tool studied in this chapter is basically the same as that of the previous two chapters. The difference is that the wafer flow pattern in this chapter can be various, and the objective is to minimize the makespan. The assumptions regarding the structure of the multi-cluster tool, the moves of the transport module, the processing time and the residency constraint are as follows:

(1) each cluster tool $i$ ($i = 1,...,I$) is connected with one or two other cluster tools; two adjacent cluster tools are connected through two buffer modules ($B_{i(i+1)}$ and $B_{(i+1)i}$);

(2) all the transport modules are single-armed robots, for each robot, the unloading time is equal to the loading time, and the transporting time between modules is assumed to be constant;

(3) the process must begin as soon as the wafer is loaded in the processing module;

(4) for each processing module, only one wafer can be loaded and processes at a time;

(5) each robot can handle one wafer at a time;

(6) the capacity of buffer module is one;

(7) residency constraints is considered, i.e., there is upper bound of current residency time for each processing module, after the processing is completed, the wafer would be defective or scrapped if it resides on the processing module longer than the upper bound of residency constraint;

(8) The wafer flow patterns of different types of wafers are not exactly the same, and the processing times of different types of wafers on the same processing module can be different.

According to the assumptions (1) to (8), this chapter still considers the linear multi-cluster tool; as the transmission channel, the buffer module is connected with the adjacent cluster tools. Since the buffer module has no processing function, there is not restriction of residency time on buffer modules. The transport modules of the multi-cluster tool considered in this chapter are single-armed robots; the handling time is short and is assumed to be constant. When the wafer enters the target processing module, it must start processing immediately without waiting. According to the practice, the capacity of processing module, transport module and buffer module are one. After the wafer has been processed on the processing module, it needs to wait until the target module is available. Due to the particularity of the wafer, in order to ensure quality, the wafer has an upper limit of residency time on the processing module, i.e., there are residency constraints on the processing module.

Based on chapter 3 and 4, this chapter addresses the scheduling problem of multi-cluster tools with multi-wafer types. In this chapter, wafers arrive at CM in lot, and then wafers enter the multi-cluster tool one by one. Wafer can skip processing modules but cannot skip the preorder wafer.

From the above, it can be seen that the problem studied in this chapter is to coordinate the moves of multiple robots in a multi-cluster tool that fabricates multiple types of wafers and to achieve an objective that minimizes the makespan while meeting various constraints, thereby maximizing the yield of the multi-cluster tool.

## 5.2 A non-linear programming model

### 5.2.1 Notations and variables

In order to describe the mathematical model clearly and accurately, this chapter adds a series of related notations and variables. The notations and variables referred to in this chapter will be described as follows.

First, as in previous two chapters, we use the one-dimensional code to define the relevant notations of the number of cluster tools and the TMs, that is, to locate the number of cluster tool with a subscript. For example, $C_i$ represents the $i$-th cluster tool, $R_i$ represents the transport module in $C_i$. We adopt two-dimensional code to define the CMs, PMs and BMs, i.e., to locate the location with double subscripts.

Then, the wafers are numbered. According to the order in which wafers are entered into the multi-cluster tools, $w$ represents the $w$-th wafer, assuming that the number of wafers in one batch is $W$.

Lastly, because of the multiple types of wafers, we employ the same method as chapter 4 to define the variables that is relevant to the unloading time and loading time. The superscript is used to distinguish the wafer type. The subscript is used to locate the wafer location. Such as, $t_{rs,i,j}^{w}$ indicates the time ( $s$ ) at which $w$-th wafer that is unloaded from $M_{i,j}$ by robot $r$.

Based on the above description, the notations and variables added in this chapter are defined as follows:

| | |
|---|---|
| $\pi$ | A schedule; |
| $\pi^*$ | The optimal schedule; |
| $\mathrm{t}_{makespan}(\pi)$ | The makespan that corresponds to $\pi$; |
| $w$ | The $w$-th wafer of a lot; |
| $W$ | The total number of wafers in a lot; |
| $t_{P,i,j}^{w,L}$ | The processing time of the $w$-th wafer in $M_{i,j}$; |
| $t_{P,i,j}^{w,U}$ | The upper bound of residency time of the $w$-th wafer in $M_{i,j}$; |
| $t_{rs,i,j}^{w}$ | The unloading time of wafer $w$ from $M_{i,j}$; |

| | |
|---|---|
| $t^{w}_{rl,i,j}$ | The loading time of wafer $w$ to $M_{i,j}$; |
| $t^{w}_{ms,i,j}$ | The time at which wafer $w$ starts to process on $M_{i,j}$; |
| $t^{w}_{ml,i,j}$ | The time at which wafer $w$ completes the process on $M_{i,j}$; |
| $t^{w}_{res,i,j}$ | The current residency time of wafer $w$ on $M_{i,j}$ since the process is completed; |
| $T^{w}_{rs,i,j}$ | The time interval sets of unloading time at which wafer $w$ is unloaded from $M_{i,j}$. |

Before we build the mathematical model in this section, we define the following concepts.

**Definition 5.1**: *If* $t^{w,L}_{P,a,b} = \max\{t^{w,L}_{P,i,j}\}$ *is satisfied, where* $i = 1,\cdots,I$ *and* $j = 1,\cdots,J$ *; then,* $M_{ab}$ *is called the bottleneck PM of wafer* $w$ *(BPw), and denoted by* $BP^{w}$.

**Definition 5.2**: *The fore-bottleneck module is a general term for all processing modules and buffer modules in the upstream direction of* $BP^{w}$*, according to the wafer flow pattern; the post-bottleneck module is a general term for all processing modules and buffer modules in the downstream direction of* $BP^{w}$*, according to the wafer flow pattern.*

For example, in figure 5.1, if the BPw of wafer $w$ is $M_{i,1}$ (modules that colored in yellow); then, the fore-bottleneck modules are the modules that colored in red, i.e., $\{M_{1,1}, M_{1,2},\cdots, M_{1,x}, B_{1,2}, M_{2,1},\cdots, B_{i-1,i}\}$; and the post-bottleneck modules are the one colored in blue, that is, $\{M_{i,2},\cdots, M_{i,x}, B_{i,x+1},\cdots, M_{I,J}, B_{I,I-1}, M_{I-1,x+1},\cdots, M_{1,J}\}$.

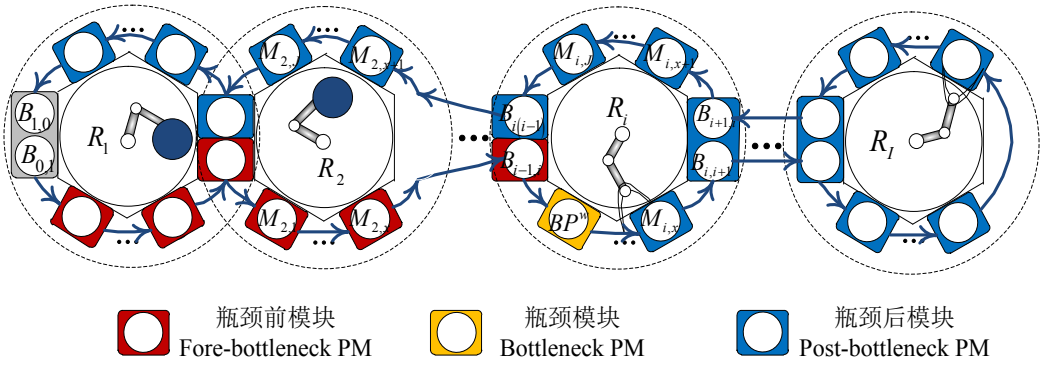



Figure 5.1 Schematic view of wafer flow pattern, bottleneck PM, fore-bottleneck PM and post-bottleneck PM of wafer $w$

## 5.2.2 Mathematical model

As mentioned earlier, the objective of this chapter is the minimum makespan for a lot of wafers, namely:

$$t_{makespan}\left(\pi^{*}\right)=Min\left(t_{rs,I,J}^{W}+\theta\right). \tag{5-1}$$

According to the assumptions (3) and (6), the robot can only carry one wafer at a time. The handling time of the robot is shorter than the current residency time and is constant. Then,

$$t_{ml,i,j}^{w}-t_{ms,i,j}^{w}>\theta;i\in[1,I];\ j\in[1,J];w\in[1,W]. \tag{5-2}$$

It can be seen from the assumption (5) that a processing module can only process one wafer at a time, thus, the time interval for the robot to load twice in succession must be less than the processing time of the wafer in this processing module:

$$t_{rl,i,j}^{w}-t_{rl,i,j}^{w-1}>t_{P,i,j}^{w-1,L};i\in[1,I];\ j\in[1,J];w\in[1,W]. \tag{5-3}$$

Wafer $w+1$ must be loaded to the buffer module after the wafer $w$ leaves, then there are:

$$t_{rs,i,j}^{w+1}-t_{rl,i,j-1}^{w}\geq 0;i\in[1,I];\ j\in[2,x];w\in[1,W-1]. \tag{5-4}$$

$$t_{rs,i,j}^{w+1}-t_{rl,i,j-1}^{w}\geq 0;i\in[1,I];\ j\in[x+2,J];w\in[1,W-1]. \tag{5-5}$$

$$t_{rs,I,j}^{w+1}-t_{rl,I,j-1}^{w}\geq 0;\ j\in[2,J];w\in[1,W-1]. \tag{5-6}$$

Based on the capacity constraints of the buffer module, that is, according to the assumption (7), the buffer module can only temporarily store one wafer at a time.

Therefore, the wafer $w$ must be loaded after the wafer $w-1$ leaves the buffer module.

$$t_{rs,i,x}^{w} - t_{rl,i+1,1}^{w-1} + 2\theta \geq 0 \; ; i \in [1, I-1]; w \in [2, W]. \tag{57}$$

$$t_{rs,i+1,J}^{w} - t_{rl,i,x+1}^{w-1} + 2\theta \geq 0 \; ; i \in [1, I-1]; w \in [2, W]. \tag{5-8}$$

Similarly, for wafer $w$, there are:

$$t_{rl,i+1,1}^{w} - t_{rs,i,x}^{w} \geq 2\theta \; ; i \in [1, I-1]; w \in [1, W]. \tag{5-9}$$

$$t_{rl,i,x+1}^{w} - t_{rs,i+1,J}^{w} \geq 2\theta \; ; i \in [1, I-1]; w \in [1, W]. \tag{5-10}$$

$$t_{rl,i+1,J}^{w} = t_{B,i,i+1}^{w} + t_{rs,i,x}^{w} + 2\theta \; ; i \in [1, I-1]; w \in [1, W]. \tag{5-11}$$

$$t_{rl,i,x+1}^{w} = t_{B,i+1,i}^{w} + t_{rs,i+1,J}^{w} + 2\theta \; ; i \in [1, I-1]; w \in [1, W]. \tag{5-12}$$

The time interval at which the robot unloads the wafer twice must meet the following inequalities:

$$t_{rs,i,x}^{w} \geq t_{rs,i,x}^{w-1} + t_{B,i,i+1}^{w-1} \; ; i \in [1, I-1]; w \in [2, W]. \tag{5-13}$$

$$t_{rs,i+1,J}^{w} \geq t_{rs,i+1,J}^{w-1} + t_{B,i+1,i}^{w-1} \; ; i \in [1, I-1]; w \in [2, W]. \tag{5-14}$$

According to assumption (8), the current residency time of the wafer on the processing module needs to satisfy the residency constraint, that is, the current residency time must be greater than the processing time and less than the upper bound of residency time. Then,

$$t_{P,i,j}^{w,L} \leq t_{rs,i,j}^{w} - t_{ms,i,j}^{w} \leq t_{P,i,j}^{w,U} \; ; i \in [1, I]; \; j \in [1, J]; w \in [1, W]. \tag{5-15}$$

$$t_{rs,i,j}^{w} - t_{ml,i,j}^{w} \leq t_{P,i,j}^{w,U} - t_{P,i,j}^{w,L} \; ; i \in [1, I]; \; j \in [1, J]; w \in [1, W]. \tag{5-16}$$

$$t_{rs,i,j}^{w} = t_{ml,i,j}^{w} + t_{res,i,j}^{w} \; ; i \in [1, I]; \; j \in [1, J]; w \in [1, W]. \tag{5-17}$$

$$t_{ml,i,j}^{w} = t_{ms,i,j}^{w} + t_{P,i,j}^{w,L} \; ; i \in [1, I]; \; j \in [1, J]; w \in [1, W]. \tag{5-18}$$

Since the moves of the robot are coherent, the following equation must be satisfied:

$$t_{rl,i,j+1}^{w} = t_{rs,i,j}^{w} + \theta \; ; i \in [1, I-1]; \; j \in [1, x-1]; w \in [1, W]. \tag{5-19}$$

$$t_{rl,i,j+1}^{w} = t_{rs,i,j}^{w} + \theta \; ; i \in [1, I-1]; \; j \in [x+1, J-1]; w \in [1, W]. \tag{5-20}$$

$$t^{w}_{rl,I,j+1} = t^{w}_{rs,I,j} + \theta ; i \in [1, I-1]; w \in [1, W]. \quad (5\text{-}21)$$

Assumption (4) specifies that the wafer start to processing immediately after arriving at the processing module without waiting, that is,

$$t^{w}_{rl,i,j} = t^{w}_{ms,i,j} ; i \in [1, I]; \ j \in [1, J]; w \in [1, W]. \quad (5\text{-}22)$$

Finally, on the basis of the analysis of the scheduling problem, we establish a mathematical model of the scheduling problem of multi-cluster tool considering the residency constraints in the case of multiple wafer flow patterns. The model is a nonlinear programming model with function (5-1) as the objective and subjects to constraint (5-2) to (5-22).

## 5.3 Lower-bound of the non-cyclic scheduling problem

The scheduling problem studied in this chapter is NP-hard problem, so it is hard to find optimal solution in polynomial time. In order to establish the lower bound of the non-cyclic scheduling problem that is discussed in this chapter, this section presents following theorems and definitions that are intended to provide a reference for the evaluation of the performance of the scheduling algorithm proposed in ths next section.

**Theorem 5.1** *The cyclic scheduling problem of single cluster tool with residency constraints is strongly NP-hard* [112].

Compared with the cyclic scheduling problem mentioned in theorem 5.1, the scale of the scheduling problem of the multi-cluster tool studied in this chapter is much larger than that. Thus, the difficulty of find optimal solution is higher, too. So we established the lemma 5.1.

**Lemma 5.1** *The scheduling problem of multi-cluster tools considering residency constraints in multi-wafer flow patterns is NP-hard.*

Minimizing the makespan is one of the basic objectives for multi-cluster tool scheduling problem. Based on lemma 5.1, the optimal solution of the scheduling problem of multi-cluster tools considering residency constraints in multi-wafer flow patterns is hard to find in polynomial time. Thus, we try to establish the lower bound of the non-cyclic scheduling problem.

For the problem studied in this chapter, the makespan of a lot of wafers is the length of time from the first wafer leaves the CM and enters the multi-cluster tool to

the last piece of the lot of wafers leaving the other CM of multi-cluster tool, i.e., $C_{makespan}(\pi)=\max\left(C_t^1,...,C_t^W\right)$.

**Definition 5.3**: *if there is optimal solution of multi-cluster tool scheduling problem ( $\pi$ ), then there must be $LB(\pi)$ and $k$, $k\in N^+$, and satisfy $\Delta(1,k+W)-\Delta(0,k)=LB(\pi)$. In $\Delta(0/1,w)$, 0 means that wafer $w$ leaves the CM and enters the multi-cluster tool,，1 means that wafer $w$ is processed and leaves the multi-cluster tool through the other CM. For instance, $\Delta(1,w)$ is the time that the wafer $w$ leaves the multi-cluster tool through CM, $\Delta(0,w)$ is the time that wafer $w$ enters the multi-cluster tool. $LB(\pi)$ represents the lower bound of the makespan according to schedule $\pi$.*

According to the above definition, we establish the lower bound of the non-cyclic scheduling problem of multi-cluster tools with residency constraints and multi-wafer types.

**Theorem 5.2** *In multiple wafer flow patterns, if the objective of scheduling problem of $I$ -cluster tools with residency constraints is minimum makespan of $W$ wafers in a lot, then the makespan that corresponds to schedule $\pi$ is:*

$$LB(\pi)=(2I+1)\theta+\sum_{i=1}^{I}\sum_{j=1}^{J}\left(t_{P,i,j}^{1,L}+\theta\right)+\sum_{w=2}^{W}\max_{i\in[1,I];j\in[1,J]}\left\{t_{P,i,j}^{w,L}+2\theta\right\}$$

,

$LB(\pi)$ is the lower-bound of the scheduling problem that studied in this chapter.

**Proof：** *At the beginning, all modules of the multi-cluster tool are in an idle state, that is, all modules within the multi-cluster tool are available when the first wafer enters the multi-cluster tool. Thus, the first wafer can be unloaded immediately from the processing module after the completion of processing, and its current residency time at the buffer module is zero. During the entire processing, the total of current residency time on the processing modules and the robot handling time of the first wafer is $\sum_{i=1}^{I}\sum_{j=1}^{J}\left(t_{P,i,j}^{1,L}+\theta\right)$, and the sum of current residency time on buffer modules and the robot handling time of the first wafer is $(2I+1)\theta$.*

For wafer 2 to wafer $W$, ignoring the first wafers for the occupation of modules, and assuming that the schedule can meet the premise of the current residency time of wafer is not less than wafer processing time, then the difference of time between wafer $w-1$ leaves the multi-cluster tool and wafer $w$ leaves the multi-cluster tool is $\max_{i\in[1,I];j\in[1,J]}\left\{t_{P,i,j}^{w,L}+2\theta\right\}$. Therefore, the difference of time between wafer $W$ and the first wafer leaves the multi-cluster tool is $\sum_{w=2}^{W}\max_{i\in[1,I];j\in[1,J]}\left\{t_{P,i,j}^{w,L}+2\theta\right\}$.

Based on the definition of makespan, the makespan that corresponds to the schedule $\pi$ is as follows. In other words, the lower-bound of the non-cyclic scheduling problem is as follow.

$$LB(\pi)=(2I+1)\theta+\sum_{i=1}^{I}\sum_{j=1}^{J}\left(t_{P,i,j}^{1,L}+\theta\right)+\sum_{w=2}^{W}\max_{i\in[1,I];j\in[1,J]}\left\{t_{P,i,j}^{w,L}+2\theta\right\}.$$

## 5.4 Bottleneck-based push-pull scheduling algorithm

In this section, we are going to propose an efficient scheduling algorithm based on the bottleneck module.

### 5.4.1 Core idea and process of algorithm

In order to schedule efficiently, and to achieve the goal of minimizing the makespan, based on the mathematical model established in this chapter and the principle of "bottleneck machine dominate other machines" in TOC, a bottleneck-based push-pull scheduling method called BP algorithm is proposed by control the Takt of bottleneck equipment. The BP algorithm is designed to solve three types of problems: robot resource conflict, processing module resource conflict and residency constraint.

The flow chart of BP algorithm is shown in figure 5.2. According to the assumptions on wafer flow patterns, after wafers arrive at the cassette module in lot, they enter the PMs according to the established order. Therefore, when the current wafer waits to enter the first processing module, we begin to calculate the scheduling time point. First of all, the current bottleneck module of the multi-cluster tool is calculated based on the definition, and thus the multi-cluster tool is divided into fore-bottleneck modules and post-bottleneck modules. Then, for the fore-bottleneck modules, a pull strategy is adopted. Under the premise of satisfying all the constraints, the optimal time point of robot moves is found in the order of step-by-step

backtracking by sliding the time block. After that, we use the push strategy to schedule the post-bottleneck module. In this step, we calculate the optimal scheduling time of the post-bottleneck modules in turn until the cassette module. If the feasible time interval cannot be found by sliding the time block within the range of satisfying the residency constraints, then to delay the time block of the bottleneck module by a unit of time, and thus the feasible schedule is searched again. Finally, output the optimal sequences of robot moves and the corresponding minimum makespan of a lot of wafers.

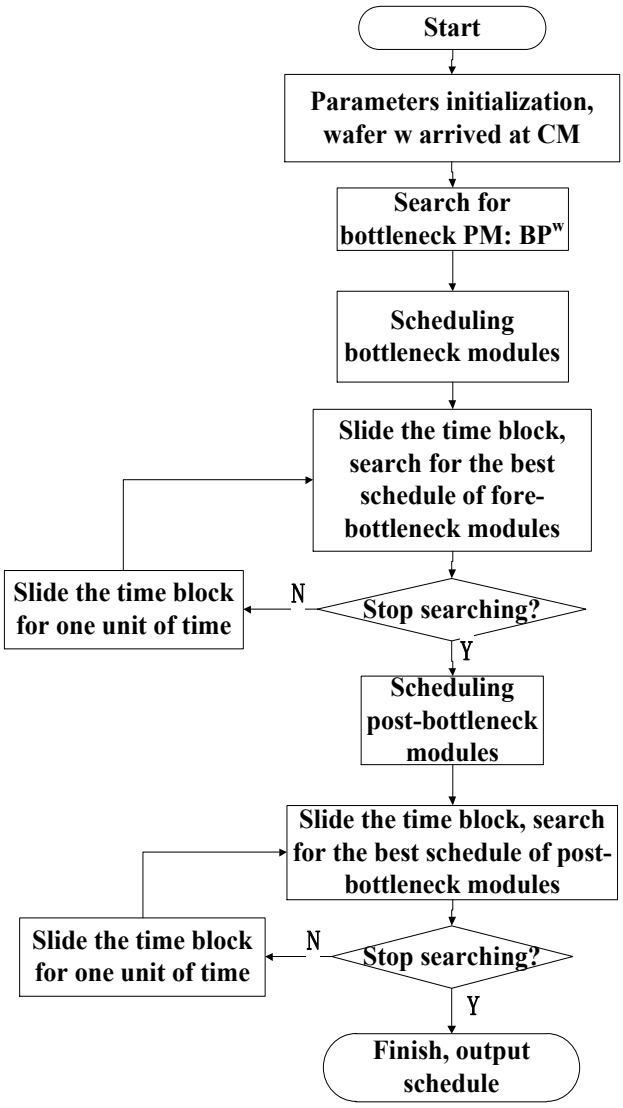



Figure 5.2 Flow chart of BP algorithm

## 5.4.2 Steps of BP Algorithm

BP algorithm consists of three phases: 1) initialization, bottleneck identification and scheduling; 2) scheduling fore-bottleneck modules with pull strategy; 3) scheduling post-bottleneck modules with push strategy. The detailed flow of the BP algorithm is as follows.

1) Initialization, bottleneck identification and scheduling

Step 1. Parameters initialization, and waiting for wafer $w$ entering the multi-cluster tools.

Step 2. Identify the bottleneck module of the pre-order wafer $w-1$, i.e., $BP^{w-1}=M_{\alpha\beta}$。

Step 3. Calculate the time at which the current wafer $w$ is unloaded from the bottleneck upstream module $BP^{w-1}$ by the robot.

$$t^{w}_{rs,\alpha-1,x}=t^{w}_{rl,\alpha,\beta}-t^{w}_{B,\alpha,\alpha+1}-2\theta;\alpha\in[1,I-1];\beta=1;w\in[2,W]. \tag{5-23}$$

$$t^{w}_{rs,\alpha+1,J}=t^{w}_{rl,\alpha,\beta}-t^{w}_{B,\alpha+1,\alpha}-2\theta;\alpha\in[1,I-1];\beta=x+1;w\in[2,W]. \tag{5-24}$$

$$t^{w}_{rs,\alpha,\beta-1}=t^{w}_{ml,\alpha,\beta-1};\alpha\in[1,I-1];\beta\in\{[2,x]\cup[x+2,J]\};w\in[2,W]. \tag{5-25}$$

$$t^{w}_{rs,\alpha,\beta-1}=t^{w}_{ml,\alpha,\beta-1};\alpha=I;\beta\in[2,J];w\in[2,W]. \tag{5-26}$$

Step 4. According to inequalities (5-27) to (5-31), it is judged whether the lower bound of the residency constraint is satisfied, that is, whether or not the wafer can accomplish the processing. If not, slide the time block according to equations (5-32) to (5-35) until all the inequalities (5-36) to (5-40) are satisfied.

$$t^{w}_{rs,\alpha-1,x}<\sum_{i=1}^{\alpha-1}\sum_{j=1}^{x}\left(t^{w,L}_{P,i,j}+\theta\right);\alpha\in[1,I-1];\beta=1;w\in[2,W]. \tag{5-27}$$

$$t^{w}_{rs,\alpha+1,J}<\sum_{i=1}^{\alpha}\sum_{j=1}^{x}\left(t^{w,L}_{P,i,j}+\theta\right)+\sum_{i=\alpha+1}^{I}\sum_{j=1}^{J}\left(t^{w,L}_{P,i,j}+\theta\right);\alpha\in[1,I-1];\beta=x+1;w\in[2,W]. \tag{5-28}$$

$$t^{w}_{rs,\alpha,\beta-1}<\sum_{i=1}^{\alpha-1}\sum_{j=1}^{x}\left(t^{w,L}_{P,i,j}+\theta\right)+\sum_{j=1}^{\beta-1}\left(t^{w,L}_{P,i,j}+\theta\right);\alpha\in[1,I-1];\beta\in[2,x];w\in[2,W]. \tag{5-29}$$

$$t^{w}_{rs,\alpha,\beta-1}<\sum_{i=1}^{\alpha-1}\sum_{j=1}^{x}\left(t^{w,L}_{P,i,j}+\theta\right)+\sum_{i=\alpha+1}^{I}\sum_{j=1}^{J}\left(t^{w,L}_{P,i,j}+\theta\right)+\sum_{j=1}^{\beta-1}\left(t^{w,L}_{P,\alpha,j}+\theta\right);$$

$$\alpha\in[1,I-1];\beta\in[x+2,J];w\in[2,W]. \tag{5-30}$$

$$t^{w}_{rs,\alpha,\beta-1} < \sum_{i=1}^{\alpha-1}\sum_{j=1}^{x}\left(t^{w,L}_{P,i,j}+\theta\right)+\sum_{j=1}^{\beta-1}\left(t^{w,L}_{P,\alpha,j}+\theta\right);\alpha=I;\beta\in[x+2,J];w\in[2,W]. \tag{5-31}$$

$$t^{w}_{rs,\alpha-1,x}=t^{w}_{rs,\alpha-1,x}+1;\alpha\in[1,I];\beta=1;w\in[2,W]. \tag{5-32}$$

$$t^{w}_{rs,\alpha+1,J}=t^{w}_{rs,\alpha+1,J}+1;\alpha\in[1,I-1];\beta=x+1;w\in[2,W]. \tag{5-33}$$

$$t^{w}_{rs,\alpha,\beta-1}=t^{w}_{rs,\alpha,\beta-1}+1;\alpha\in[1,I-1];\beta\in\left\{[2,x]\bigcup[x+2,J]\right\};w\in[2,W]. \tag{5-34}$$

$$t^{w}_{rs,\alpha,\beta-1}=t^{w}_{rs,\alpha,\beta-1}+1;\alpha=I;\beta\in[2,J];w\in[2,W]. \tag{5-35}$$

$$t^{w}_{rs,\alpha-1,x}\geq\sum_{i=1}^{\alpha-1}\sum_{j=1}^{x}\left(t^{w,L}_{P,i,j}+\theta\right);\alpha\in[1,I-1];\beta=1;w\in[2,W]. \tag{5-36}$$

$$t^{w}_{rs,\alpha+1,J}\geq\sum_{i=1}^{\alpha}\sum_{j=1}^{x}\left(t^{w,L}_{P,i,j}+\theta\right)+\sum_{i=\alpha+1}^{I}\sum_{j=1}^{J}\left(t^{w,L}_{P,i,j}+\theta\right);\alpha\in[1,I-1];\beta=x+1;w\in[2,W]. \tag{5-37}$$

$$t^{w}_{rs,\alpha,\beta-1}\geq\sum_{i=1}^{\alpha-1}\sum_{j=1}^{x}\left(t^{w,L}_{P,i,j}+\theta\right)+\sum_{j=1}^{\beta-1}\left(t^{w,L}_{P,i,j}+\theta\right);\alpha\in[1,I-1];\beta\in[2,x];w\in[2,W]. \tag{5-38}$$

$$t^{w}_{rs,\alpha,\beta-1}\geq\sum_{i=1}^{\alpha-1}\sum_{j=1}^{x}\left(t^{w,L}_{P,i,j}+\theta\right)+\sum_{i=\alpha+1}^{I}\sum_{j=1}^{J}\left(t^{w,L}_{P,i,j}+\theta\right)+\sum_{j=1}^{\beta-1}\left(t^{w,L}_{P,\alpha,j}+\theta\right);$$

$$\alpha\in[1,I-1];\beta\in[2,J];w\in[2,W]. \tag{5-39}$$

$$t^{w}_{rs,\alpha,\beta-1}\geq\sum_{i=1}^{\alpha-1}\sum_{j=1}^{x}\left(t^{w,L}_{P,i,j}+\theta\right)+\sum_{j=1}^{\beta-1}\left(t^{w,L}_{p,\alpha,j}+\theta\right);\alpha=I;\beta\in[2,J];w\in[2,W]. \tag{5-40}$$

Step 5. Determine whether the inequalities (5-41) to (5-44) are all true. If all are set up, read the data; otherwise, go back to step 4, update equalities (5-32) to (5-35).

$$\left|t^{w}_{rs,\alpha-1,x}-t^{w}_{rs,\alpha,\beta}\right|<\theta;\alpha\in[1,I-1];\beta=1;w\in[2,W]. \tag{5-41}$$

$$\left|t^{w}_{rs,\alpha+1,J}-t^{w}_{rs,\alpha,\beta}\right|<\theta;\alpha\in[1,I-1];\beta=x+1;w\in[2,W]. \tag{5-42}$$

$$\left|t^{w}_{rs,\alpha,\beta-1}-t^{w}_{rs,\alpha,\beta}\right|<\theta;\alpha\in[1,I-1];\beta\in\left\{[2,x]\bigcup[x+2,J]\right\};w\in[2,W]. \tag{5-43}$$

$$\left|t^{w}_{rs,\alpha,\beta-1}-t^{w}_{rs,\alpha,\beta}\right|<\theta;\alpha=I;\beta\in[2,J];w\in[2,W]. \tag{5-44}$$

The phase of initialization, bottleneck identification and scheduling ends here.

2) Scheduling fore-bottleneck modules with pull strategy

Step 6. Parameters initialization. Define auxiliary variables $m$ and $n$. If $\beta\neq x+1$ and $\beta\neq 1$, then let $m=\alpha$, $n=\beta-1$ and $t^{w}_{res,m,n}=0$. If $\alpha=I$ and

$\beta = x+1$, let $m = \alpha$, $n = \beta - 1$ and $t^{w}_{res,m,n} = 0$. Otherwise, if $\alpha = 1$ and $\beta = 1$, go to step 12.

Step 7. If $t^{w}_{res,m,n} \le t^{w,U}_{P,m,n} - t^{w,L}_{P,m,n}$, then go to step 8. Otherwise, update $t^{w}_{rs,\alpha,\beta-1}$ based on equations (5-34) and (5—35). After that, if the current module is buffer module, go back to step 4; if the current module is processing module, let $t^{w}_{res,m,n} = t^{w}_{res,m,n} + 1$.

Step 8. Calculate the parameters as follows.

If the current module under scheduling is processing module, then

$$t^{w}_{ml,m,n} = t^{w}_{rs,m,n} - t^{w}_{res,m,n}. \tag{5-45}$$

$$t^{w}_{ms,m,n} = t^{w}_{ml,m,n} - t^{w,L}_{P,m,n}. \tag{5-46}$$

$$t^{w}_{rs,m,n-1} = t^{w}_{rl,m,n} - \theta. \tag{5-47}$$

$$t^{w}_{rl,m,n} = t^{w}_{ms,m,n}. \tag{5-48}$$

If the current module under scheduling is $B_{m+1,m}$ or $B_{m-1,m}$, then

$$t^{w}_{rs,m+1,J} = t^{w}_{rl,m,n} - t^{w}_{B,m+1,m} - 2\theta. \tag{5-49}$$

$$t^{w}_{rs,m-1,x} = t^{w}_{rl,m,n} - t^{w}_{B,m-1,m} - 2\theta. \tag{5-50}$$

Step 9. Check whether the scheduling time of the robot moves is feasible. If it is, proceed to step 10; otherwise, let $t^{w}_{res,m,n} = t^{w}_{res,m,n} + 1$, and go back to step 7.

Step 10. When the current module is $B_{m+1,m}$ or $B_{m-1,m}$, if the formula (5-51) or (5-52) cannot be satisfied respectively, increase the residence time of one unit; otherwise, record the parameter value. When the current module is the processing module, if the formula (5-53) cannot be satisfied, increase the residence time of one unit, otherwise, record the parameter value.

$$t^{w}_{rs,\alpha+1,J} - t^{w-1}_{rl,\alpha,\beta} + \theta \ge 0. \tag{5-51}$$

$$t^{w}_{rs,\alpha-1,x} - t^{w-1}_{rl,\alpha,\beta} + \theta \ge 0. \tag{5-52}$$

$$t^{w}_{rs,m,n} - t^{w-1}_{rs,m,n} \ge \theta. \tag{5-53}$$

Step 11. Update parameters. If the current module is processing module but not $M_{1,1}$ and the pre-order module of current module is a processing module, let $n = n-1$ and $t^{w}_{res,m,n} = 0$. When the current module is $B_{m+1,m}$, let $t^{w}_{B,m+1,m} = 0$; if the pre-order module of current module is $B_{m-1,m}$, let $t^{w}_{B,m-1,m} = 0$. If current module is $B_{m+1,m}$, let $m = m+1$, $n = J$ and $t^{w}_{res,m,n} = 0$. If the current module is $B_{m-1,m}$, let $m = m-1$, $n = x$ and $t^{w}_{res,m,n} = 0$. If $m = 1$ and $n = 1$, proceed to step 12.

After completing the scheduling for all fore-bottleneck modules, the algorithm goes to the next phase.

3) Scheduling post-bottleneck modules with push strategy

Step 12. Parameters initialization. Define auxiliary parameters $m$ and $n$. If $\beta \neq x$ and $\beta \neq J$, let $m = \alpha$, $n = \beta + 1$ and $t^{w}_{res,m,n} = 0$. If $\alpha = I$ and $\beta = x$, let $m = \alpha$, $n = \beta + 1$ and $t^{w}_{res,m,n} = 0$. If $\alpha = I$ and $\beta = J$, proceed to step 17.

Step 13. If the current module satisfies all the residency constraints, proceeds to step 14. Otherwise, if the current module is a bottleneck module, return to the pre-order module of the bottleneck module, update $t^{w}_{rs,\alpha,\beta-1}$ and return to step 4; in other cases, add a unit of the current residency time to the current module and return to step 13.

Step 14. Calculate the following parameters.

If the current module is processing module, then

$$t^{w}_{rs,m,n} = t^{w}_{ml,m,n} + t^{w}_{res,m,n}. \tag{5-54}$$

$$t^{w}_{rl,m,n} = t^{w}_{rs,m,n} + \theta. \tag{5-55}$$

$$t^{w}_{ml,m,n} = t^{w}_{ms,m,n} + t^{w,L}_{P,m,n}. \tag{5-56}$$

If the current module is $B_{m,m+1}$, then

$$t^{w}_{rs,m+1,x} = t^{w}_{ml,m+1,x} + t^{w}_{res,m,x}. \tag{5-57}$$

$$t^{w}_{rl,m+1,1} = t^{w}_{rs,m,x} + t^{w}_{B,m,m+1} + 2\theta. \tag{5-58}$$

If the current module is $B_{m+1,m}$, then

$$t^{w}_{rs,m+1,J} = t^{w}_{ml,m+1,J} + t^{w}_{res,m+1,J}\,. \tag{5-59}$$

$$t^{w}_{rl,m,x+1} = t^{w}_{rs,m+1,J} + t^{w}_{B,m+1,m} + 2\theta\,. \tag{5-60}$$

Step 15. If the currently scheduled module is a buffer module and satisfies inequality (5-61) or (5-62), when the robot has an available time interval, add a unit of current residency time to the current module, record the parameters; when the robot has no available intervals return to step 13. If the currently scheduled module is a processing module and satisfies inequality, (5-63), add a unit of current residency time to the current module and return to step 13; otherwise, record the parameters.

$$t^{w}_{rs,m,x} \geq t^{w-1}_{rl,m+1,1} - \theta\,. \tag{5-61}$$

$$t^{w}_{rs,m+1,J} \geq t^{w-1}_{rl,m,x+1} - \theta\,. \tag{5-62}$$

$$t^{w}_{rl,m,n} \geq t^{w-1}_{rs,m,n} + 2\theta\,. \tag{5-63}$$

Step 16. Update parameters. When the current module is processing module but not $M_{1,J}$, if the post-order module is processing module, let $n = n+1$ and $t^{w}_{res,m,n} = 0$; if the post-order module of current module is $B_{m+1,m}$, let $t^{w}_{B,m+1,m} = 0$; if the post-order module of current module is $B_{m-1,m}$, let $t^{w}_{B,m-1,m} = 0$. If the current module is $B_{m+1,m}$, let $n = x+1$ and $t^{w}_{res,m,n} = 0$. If the current module is $B_{m-1,m}$, let $n = 1$ and $t^{w}_{res,m,n} = 0$; if $m = 1$ and $n = J$, proceed to step 17.

Step 17. Scheduling the next wafer, and let $w = w+1$. If $w \leq W$, return to step 1; otherwise, output schedule.

The BP algorithm flow ends here.

According to the above detailed steps, the complexity of BP algorithm is $\mathrm{O}\left(I^2\left(J+2\right)^2 W^2\right)$.

## 5.5 Simulation and experimental analysis

Unlike the previous two chapters, the objective in this chapter is to minimize the makespan of wafers in a lot. By definition, the makespan is the length of the time interval from the time the first wafer in a lot enters the system until the last wafer in the lot has left the system. The makespan is an important measure of the throughput of

the multi-cluster tool in the multiple wafer flow patterns, and its size directly reflects the production efficiency and yield level. In order to evaluate the BP algorithm effectively, a series of experiments were carried out, including the comparison of BP algorithm and the ordinary pull algorithm. The experiment in this section aims to verify the effectiveness of the BP algorithm and evaluate its performance from different perspectives.

In order to compare the differences between the BP algorithms and ordinary pull algorithm accurately, we introduce the following indicators

$r_{makespan} = \frac{t^{BP}_{makespan} - t^{Pull}_{makespan}}{t^{Pull}_{makespan}} \times 100\%$, ration of difference of makespan or difference rate of makespan, indicates the percentage of difference between the makespan obtained by BP algorithm ($t^{BP}_{makespan}$) and that by ordinary pull algorithm ($t^{Pull}_{makespan}$). The smaller the value is, the smaller the difference is. In other words, the shorter the residency time is, the better the BP algorithm performs.

$r_{TM} = \frac{W\theta}{\left(\max\ \mathrm{t}^{w}_{P,i,j} + \theta\right) \times \varphi} \times 100\%$, the utilization rate of robot, indicates the frequency of the utilization of robot. The higher the value, the more busy the robot.

$r_{LB} = 1 - \frac{t^{BP}_{makespan} - t^{LB}_{makespan}}{t^{LB}_{makespan}} \times 100\%$, the ratio of difference of the makespan, represents the percentage of the difference between the makespan obtained by the BP algorithm and the lower bound of the scheduling problem. The smaller the value, the better the performance of BP algorithm, that is, the closer the makespan obtained by the BP algorithm to the lower bound of makespan.

In the simulations and experiments below, we assume that the processing time of the wafer is subject to a normal distribution $N(\mu, \delta)$, and the time is counted in seconds.

In the definitions mentioned above, $\varphi$ is the coefficient used to adjust the $r_{TM}$, $t^{LB}_{makespan}$ is the lower bound of makespan based on theorem 2.

In order to measure the significant effect of the main effect and interaction between the factors, the ANOVA method was used. ANOVA mainly involves three parameters: F value test, p value and the critical value. In the case of large value of F value, the original hypothesis can be rejected, indicating that the significance of this factor is large. The p value represents the probability of occurrence of error type 1, that is, the probability that the “reject” event occurred. The smaller the p value, the

less the probability of the "reject" event happens [113]. When the p value is less than the critical value of 0.05, the original hypothesis is rejected to prove that the factor is significant [114].

We implement the BP algorithm and the general pull algorithm with C ++ in Microsoft Visual Studio 2010 programming software. Simulation environment is 320G hard drive, 4GB memory and 2.53GHz frequency Core i3 processor personal computer. The following experimental results are the average of ten experiments.

### 5.5.1 CPU time

The purpose of this experiment is to test the CPU time of the BP algorithm in general circumstance. In the case of three-cluster tools, there are four processing modules in each cluster tool. In the practical production, multi-cluster tools like this belong to the large-scale equipment, such as the wafer fabrication equipment in etching process area. Table 5.1 shows the parameters related to this experiment. Experimental results are shown in figure 5.3.

Table 5.1 Parameters related to the CPU time of BP algorithm

| Parameter | Value |
|---|---|
| Number of cluster tools | 3 |
| Number of PMs in each cluster tool | 4 |
| Robot handling time | 4 |
| $\mu$ | $[20,80]$ |
| $\delta$ | $1/8\,\mu, 1/4\,\mu, 1/2\,\mu$ |
| The upper bounds of residency time after finishing processing | $[0,20]$ |
| Number of wafers in a lot | 5,10,…,30, 40,…,80 |

As shown in figure 5.3, the CPU time of the BP algorithm increases as the number of wafers increases. When the number of wafers is 5 to 10, the CPU time is quite short. Even when the number of wafers increases to 80, the CPU time is only 6 milliseconds. Overall, as the number of wafers increases, the CPU time increases linearly. As can be seen from the above, the BP algorithm can be applied to the non-cyclic scheduling of multi-cluster tool.

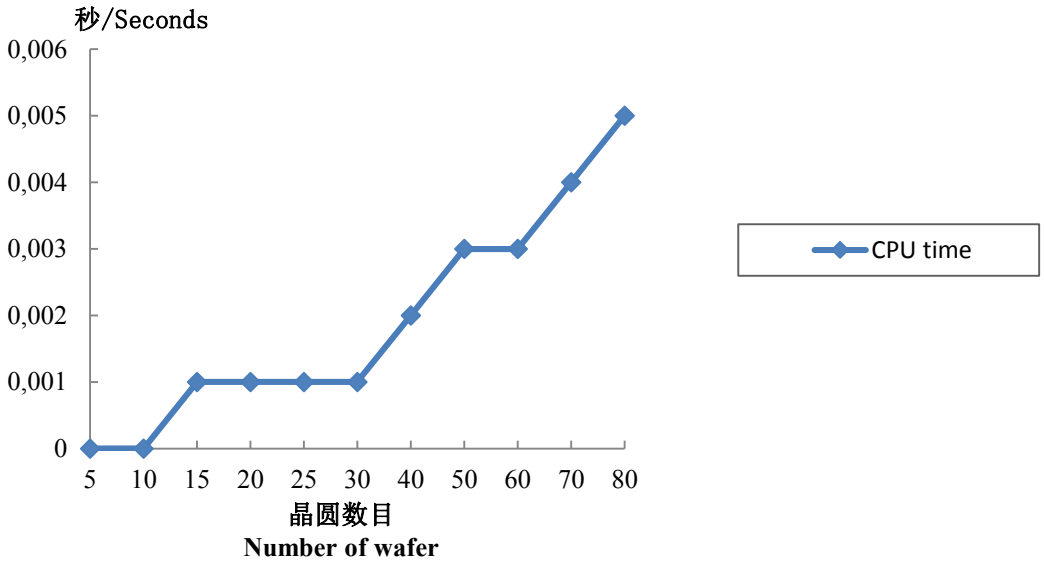



Figure 5.3 Relationship between CPU time of BP algorithm and number of wafer

## 5.5.2 Performance analysis

The performance analysis of BP algorithm mainly includes three aspects: wafer types, the structure of multi-cluster tools, and a comparison with the ordinary pull algorithm.

1) Effect of wafer types to CPU time

This experiment analyzes the relationship between the wafer type and the CPU time. Take the two-cluster tool as an example, we assume that the upper bound of residency constraint is constant in this experiment. The simulation results are shown in figure 5.4.

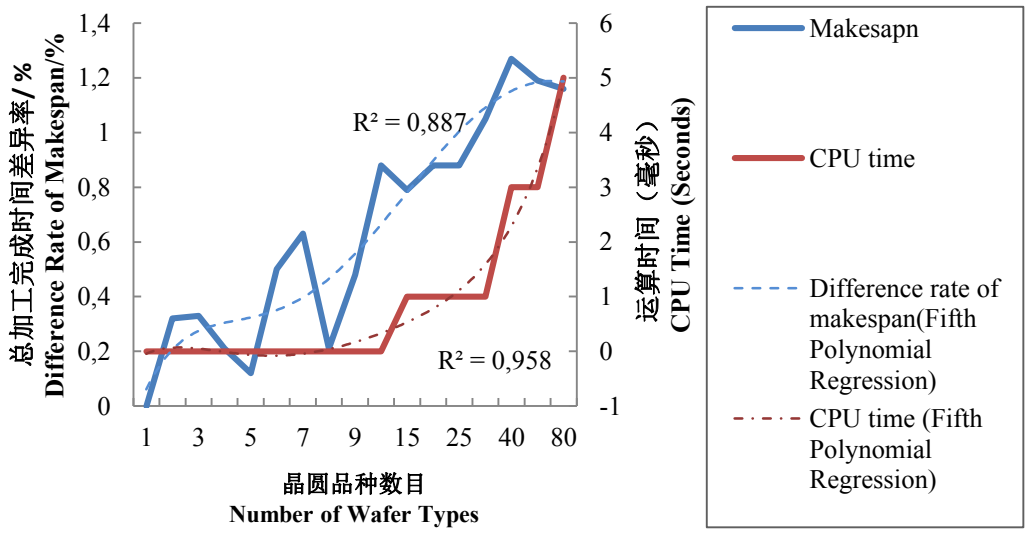



Figure 5.4 Relationship between CPU time of BP algorithm and number of wafer types

As can be seen from figure 5.4, the CPU time of the BP algorithm is very short, and the $r_{makespan}$ gradually increases with the increase of the number of wafer types. When the number of wafer types increases to 80, the $r_{makespan}$ remains at 1.4% or less. Thus, BP algorithm can adapt to the uncertainty of non-cyclic scheduling. At the same time, the BP algorithm can effectively shorten the time that the wafer resides on the processing module after processing is completed, which effectively improves the utilization of the equipment.

2) Effect of structure of multi-cluster tools to performance of BP algorithm

This experiment investigates the influence of the structure of the multi-cluster tools on the CPU time of BP algorithm. The experimental data are shown in table 5.2. The experimental results are shown in figure 5.5.

Table 5.2 Parameters for impact of structure of multi-cluster tools on CPU time experiment

| Parameter | Group 1 | Group 2 | Group 3 | Group 4 |
|---|---|---|---|---|
| Number of PMs in each cluster tool | 2 | 5 | 2 | 4 |
| Number of cluster tools | 2 | 2 | 4 | 3 |
| Number of wafers | 30 | 30 | 30 | 30 |
| $\mu$ | [20,80] | [20,80] | [20,80] | [20,80] |
| $\delta$ | | $1/8\,\mu$, $1/4\,\mu$, $1/2\,\mu$ | | |
| $\theta$ | | 5 | | |
| Upper bound of residency time after finishing wafer processing (second) | | 1 | | |

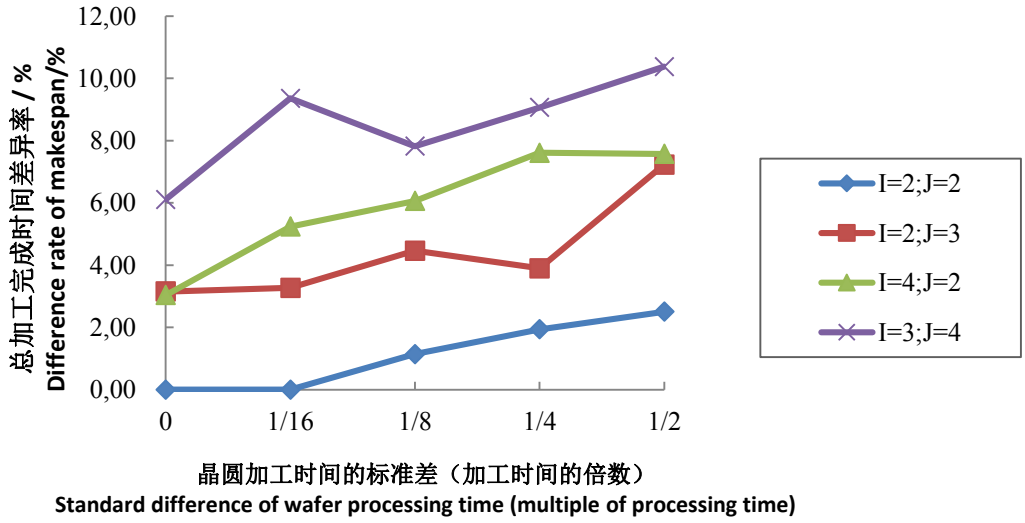



Figure 5.5 Relationship of wafer type and CPU time of BP algorithm

According to the difference between the different types of wafers and the $r_{makespan}$ shown in 5.5, the $r_{makespan}$ increases with the increase of the standard deviation of the wafer processing time, and the $r_{makespan}$ is slowly increasing. Therefore, the BP algorithm can adapt to the schedule problem in varieties of wafers flow patterns.

Another experiment is to analyze the relationship between the number of robots and the CPU time. As shown in figure 5.6, the difference between the CPU time and the $r_{makespan}$ is very slow in the multi-cluster tools with 3 to 6 cluster tools. However, as the number of robot increases, the $r_{TM}$ decreases linearly. Therefore, the performance of the BP algorithm is stable.

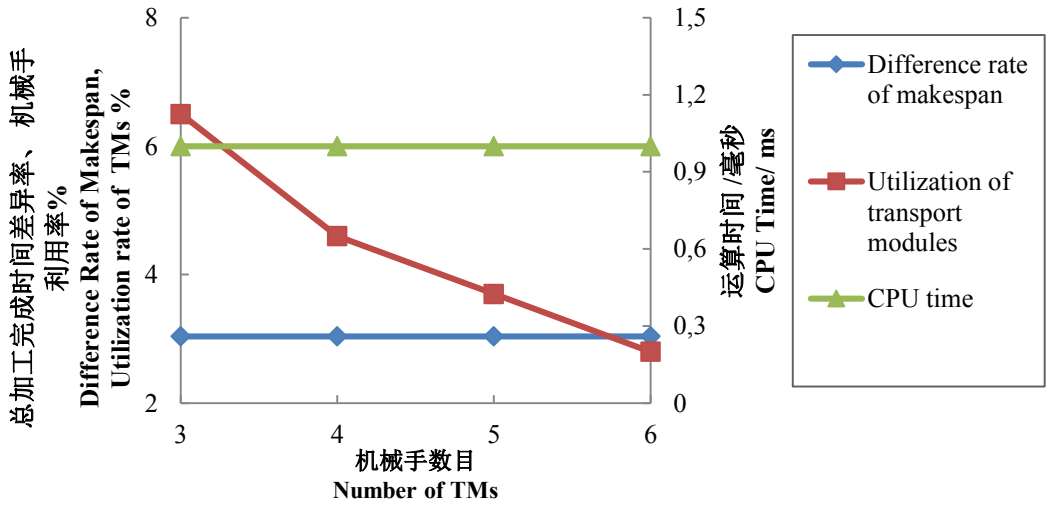



Figure 5.6 Influence of number of TMs on BP algorithm

3) Comparison between BP algorithm and ordinary pull strategy

This experiment compares the makespan obtained by BP algorithm with that by ordinary pull strategy and with the lower bound of non-cyclic scheduling problem. In order to ensure the reliability of the experiment, we experimented with four different types of multi-cluster tools. For each multi-cluster tool, we tested nine groups of different experimental parameters, and each experiment was carried out 50 times, the experimental results are shown in table 5.3.

Figure 5.7 shows the experimental results of the nine groups. With the increase of the experimental group, the cumulative value of the $r_{makespan}$ is gradually increased. When the scale of the multi-cluster tool increases, the increasing rate of the $r_{makespan}$

is also increased. In other words, the experimental results show that the BP algorithm is superior to the ordinary pull algorithm when the multi-cluster tool is complex and larger in scale.

Table 5.3 Results comparison of BP algorithm, Pull strategy and lower bound of non-cyclic scheduling problem

| Parameter | 1 | 2 | 3 | 4 |
|---|---|---|---|---|
| Number of PMs in each cluster tool | 2 | 4 | 2 | 4 |
| Number of cluster tools | 2 | 2 | 4 | 3 |
| Number of wafers in a lot | 25 | 25 | 25 | 25 |
| $\mu$ | $[20,80]$ | $[20,80]$ | $[20,80]$ | $[20,80]$ |
| $\delta$ | | $1/8\ \mu, 1/4\ \mu, 1/2\ \mu$ | | |
| $\theta$ | | 7 | | |

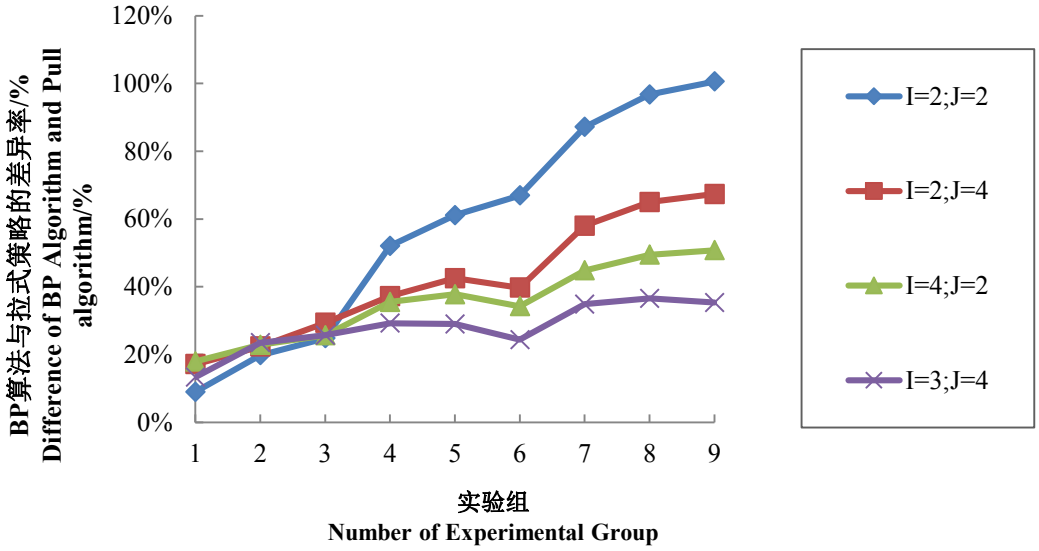



Figure 5.7 Comparison of BP algorithm and Pull strategy

Compared with the lower bound of the scheduling problem that is discussed in this chapter, the $r_{LB}$ decreases with the increase of the complexity of the structure of multi-cluster tool, but still maintain exceeds 85%. In other words, the performance of BP algorithm is superior in the case of the complexity of the structure of multi-cluster tools is higher. The experimental results are shown in figure 5.8.

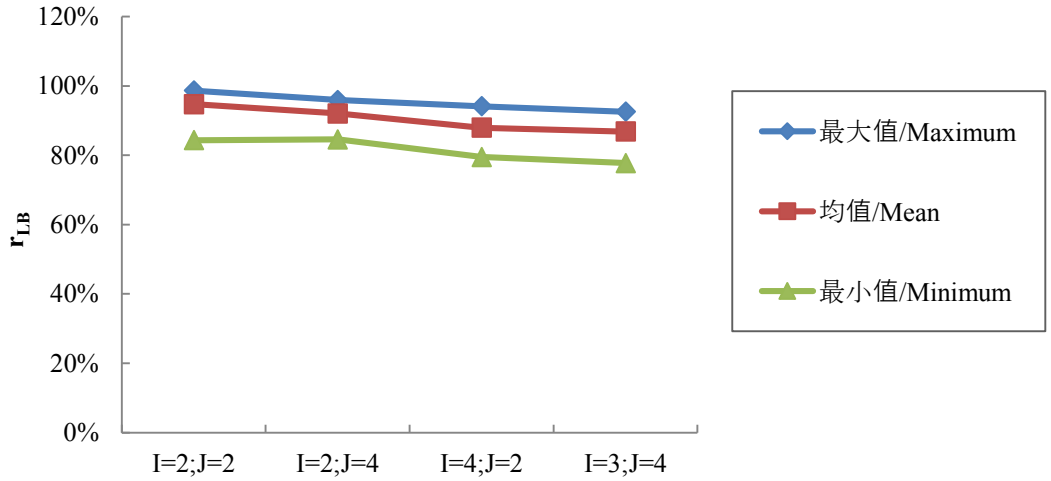



Figure 5.8 Comparison of BP algorithm and lower bound of the scheduling problem

### 5.5.3 ANOVA

In this section, we use one-way ANOVA and two-way ANOVA to analyze and validate important parameters that may affect BP algorithm. The results are shown in table 5.4 and table 5.5.

First, as shown in table 5.4, we analyzed the effects of the number of wafers ($W$), the number of cluster tools ($I$), and the variety of wafer types ($W^*$) on the BP algorithm. The experimental results show that these three factors have no significant effect on the performance of BP algorithm.

In this work, the two-factor analysis of variance is used to analyze the influence of the number of cluster tools and the number of processing modules ($I$ & $J$) and the standard difference of the number of wafer types ($\sigma_p$) to the $r_{makespan}$ and the accumulated improvement of the $r_{makespan}$ ($\sum r_{makespan}$). As shown in Table 5.5, the experimental results show that the number of cluster tools and the number of processing modules in each cluster tool ($I$ & $J$) and the standard difference of the number of wafer types ($\sigma_p$) have a significant effect on the performance of BP algorithm.

## 5.6 Summary

In this chapter, we study the modeling and non-cyclic scheduling problem of multi-cluster tool with residency constraints. Aims at minimizes the makespan, a

nonlinear programming model is established in this chapter. Using the analytical method, the lower bound of the non-cyclic scheduling problem is set up. Since the problem studied in this chapter is a NP-hard, we can hardly find a polynomial algorithm to solve it. Thus, based on TOC, we propose a bottleneck-based push-pull scheduling algorithm, which is called BP algorithm. The algorithm adjusts the Takt of the multi-cluster tools from the scheduling and control of bottleneck module, and improves the scheduling of robot moves with the strategy of "push" and "pull".

In order to verify the effectiveness of the algorithm, a series of experiments were carried out. The experimental results show that the BP algorithm is efficient and can schedule 80 different kinds of wafers in a short CPU time. In most cases, the BP algorithm can obtain the approximate-optimal solution of the scheduling problem. The structure of the multi-cluster tool, neither the number of wafers in a lot nor the varieties of wafer types has significant effect on the performance of BP algorithm. Then, from the perspective of optimality, compare the BP algorithm and the ordinary pull strategy. The results show that the BP algorithm is more flexible, and there is small difference between the minimum makespan obtained by the BP algorithm and the lower bound of the scheduling problem that is discussed in this chapter. Thirdly, the ANOVA is used to verify the experimental results, and the influence of the parameters on the performance of BP algorithm is analysed. In conclusion, the BP algorithm proposed in this chapter can effectively solve the non-cyclic scheduling problem of multi-cluster tool considering residency constraints. The schedule is feasible and the performance of BP algorithm is very stable.

Table 5.4 Results of one-way ANOVA

| Parameter VS Object | Source of variation | Sum of squares | Degrees of freedom | Mean sum of squares | F ratio | P-value | F crit |
|---|---|---|---|---|---|---|---|
| $W$ VS CPU time | Inter group | 3182.225 | 10 | 318.2225 | 0.329103 | 0.954614 | 2.853625 |
| | Intra group | 10636.32 | 11 | 966.9382 | | | |
| | Total | 13818.55 | 21 | | | | |
| $W^*$ VS CPU time | Inter group | 3599.6 | 16 | 224.975 | 0.586885 | 0.853535 | 2.2888 |
| | Intra group | 6516.74 | 17 | 383.3376 | | | |
| | Total | 10116.34 | 33 | | | | |
| $W^*$ VS $r_{makespan}$ | Inter group | 3600.211 | 16 | 225.0132 | 0.587196 | 0.853297 | 2.2888 |
| | Intra group | 6514.387 | 17 | 383.1992 | | | |
| | Total | 10114.6 | 33 | | | | |
| $I$ VS $r_{makespan}$ | Inter group | 2.5 | 3 | 0.833333 | 0.078515 | 0.968308 | 6.591382 |
| | Intra group | 42.45465 | 4 | 10.61366 | | | |
| | Total | 44.95465 | 7 | | | | |
| $I$ VS $r_{makespan}$ | Inter group | 2.440375 | 3 | 0.813458 | 0.076973 | 0.969167 | 6.591382 |
| | Intra group | 42.27225 | 4 | 10.56806 | | | |
| | Total | 44.71262 | 7 | | | | |
| $I$ VS CPU time | Inter group | 2.5 | 3 | 0.833333 | 0.077552 | 0.968845 | 6.591382 |
| | Intra group | 42.982 | 4 | 10.7455 | | | |
| | Total | 45.482 | 7 | | | | |

Table 5.5 Results of two-way ANOVA

| Parameter VS Object | Source of variation | Sum of squares | Degrees of freedom | Mean sum of squares | F ratio | P-value | F crit |
|---|---|---|---|---|---|---|---|
| $(I \& J) \& \sigma_p$ VS $\sum r_{makespan}$ | $\sigma_p$ | 0.9414576 | 2 | 0.4707288 | 178.1929153 | 3.98E-15 | 3.4028261 |
| | $I \& J$ | 0.4282659 | 3 | 0.1427553 | 54.03956535 | 7.994E-11 | 3.0087866 |
| | Interaction | 0.3340717 | 6 | 0.0556786 | 21.07696604 | 1.796E-08 | 2.5081888 |
| | Intra | 0.0634003 | 24 | 0.0026417 | | | |
| | Total | 1.7671956 | 35 | | | | |
| $(I \& J) \& \sigma_p$ VS $r_{makespan}$ | $I \& J$ | 0.0144184 | 3 | 0.0048061 | 64.47320878 | 1.136E-07 | 3.4902948 |
| | pin | 0.0032371 | 4 | 0.0008093 | 10.85635625 | 0.0005872 | 3.2591667 |
| | Error | 0.0008945 | 12 | 7.454E-05 | | | |
| | Total | 0.0185501 | 19 | | | | |

# Chapter 6 **Conclusions and Future Works**

## 6.1 Conclusions

The wafer fabrication system is one of the most complex manufacturing systems, and multi-cluster tools is a brand new kind of multi-loop coupling structured and automated 300mm wafer fabrication system. The multi-cluster tools have the characteristics of strict time constraints, intense resource conflicts, costly and so on, which is different from other manufacturing systems. Solving the scheduling problem of multi-cluster tools effectively not only concerns the development of scheduling theory but also relates to the improvement of production efficiency of wafer fabrication system. Therefore, this research has important theoretical research significance as well as significant practical application value.

After reviewing the literatures, it is found that the scheduling problem of multi-cluster tool is NP-hard in the strong sense, which means the problem is extremely complicated. The scheduling problem of multi-cluster tools considering residency constraints, resource constraints and wafer flow patterns is a hot topic in academia and industry currently. Based on the national science foundation of China, this thesis studies the modeling and scheduling of multi-cluster tools with residency constraints and resource constraints under varies wafer flow patterns in wafer fabrication. On the foundation of studying the hypothesis of other articles, we put forward some innovative viewpoints in the problem domain. From the aspects of modeling and scheduling algorithm, this paper further improves the existing researches, and develops efficient heuristic algorithms. The experimental results show that the proposed algorithms are satisfied.

Specifically, the main results of this work are as follows.

1) Two MPI-based models of multi-cluster tools cyclic scheduling problems are established. Because of the features of residency constraints and resource constraints, the state space of multi-cluster tools scheduling problem is quite complicated, leading to the difficulty of modeling. In order to solve this problem, this thesis introduces the method of prohibited intervals. Based on the analysis of cases that deadlock occurs because of the resource constraints and residency constraints, the infeasible state space is excluded effectively. The relationship

between the intermediate variable $S_{i,j}$ and the fundamental period is established, and a nonlinear mixed-integer programming model of 1-unit cyclic scheduling problem with the objective of minimum FP is thus constructed. On this basis, the 1-unit cyclic scheduling model is extended to the 2-unit cyclic scheduling model using the same method. The mathematical programming models are solved by CPLEX, and it is found that the scheduling problem of multi-cluster tools within three robots can be solved.

2) A MPI-NLMIP based two-stage approximate-optimal scheduling algorithm is proposed. Currently, literatures have presented a variety of scheduling methods for 1-unit cyclic scheduling problem, but most of them are limited to the multi-cluster tools composed of three single cluster tools. In fact, the multi-cluster tools are large-scale and tight coupled, which is easy to cause deadlock. In this thesis, a two-stage approximate-optimal scheduling algorithm based on MPI-NLMIP is proposed. In the complex solution space, the searching process is divided into two stages. In detail, the initial feasible scheduling space stage is based on the MPI-NLMIP model, in this stage, we aim to find feasible solutions; the approximate-optimal scheduling stage is to search for an approximate-optimal solution. The searching process effectively eliminates the solution space that triggers deadlock and ensures high quality of the solution. The MNB algorithm reduces the CPU time and obtains a satisfactory approximate-optimal solution even the workload is uneven.

3) A chaos-based hybrid PSO-TS scheduling algorithm is put forward. Compared with the 1-unit cyclic scheduling problem, the solution space of the 2-unit cyclic scheduling problem is more complicated, and the results obtained by the heuristic scheduling rules in the current literatures are not ideal. To deal with the insufficiency of researches on multi-unit cyclic scheduling of multi-cluster tools with residency constraints, this thesis presents a chaos-based hybrid PSO-TS algorithm. The algorithm undertakes the chaos search technique to expand the search space, thus it effectively avoids the shortcomings of basic PSO, which is easily stuck to a local optimum solution. The introduction of the Tabu list prevents the roundabout search and improves the computing speed of the algorithm. Contrast with basic PSO, the hybrid PSO-TS algorithm based on chaos search technique performs better in the aspect of CPU time and the quality of the solution.

4) A bottleneck-based push-pull algorithm is present. Due to the characteristics of the complicate wafer flow pattern, a bottleneck-based push-pull heuristic scheduling algorithm is proposed to solve multi-cluster tools non-cyclic scheduling problem, which is called BP algorithm. The BP algorithm treats the multi-loop coupling structured wafer fabrication system as a whole, which is different from the existing literatures. According to TOC, "the production efficiency of the system is determined by its bottleneck equipment", BP algorithm focus on the control of bottleneck equipment production. For fore-bottleneck and post-bottleneck equipment, it uses pull and push strategy, respectively. It is for reducing the current residency time of the wafer, thereby achieving the goal of minimizing the makespan. Simulation experiments and analysis show that the algorithm is fast and stable.

## 6.2 Innovation

1) In order to describe the characteristics of the multi-cluster tools in the form of formalized language and to visually show the complex logical relationship between the equipment resource and the robot resource, the FP and the wafer processing time; meanwhile, this thesis introduce the MPI for the first time to highlight the characteristics of scheduling problem of multi-cluster tools. The MPI-based mixed-integer programming models for 1-unit cyclic scheduling problem and 2-unit cyclic scheduling problem with residency constraints are constructed individually.
2) This thesis studies the 1-unit cyclic scheduling problem of multi-cluster tools composed of three or more single cluster tools for the first time. With residency constraints, the problem domain is more realistic. A two-stage heuristic algorithm based on MPI-NLMIP is presented. The proposed heuristic algorithm is able to solve large-scale 1-unit cyclic scheduling problem, which is different from the current literatures, which use mathematical programming in general.
3) In view of characteristic of large scale, in this thesis, we consider the problem of 2-unit cyclic scheduling problem with multiple wafer types under single wafer flow pattern. To solve the above problems, this thesis proposes a hybrid PSO-TS scheduling algorithm based on Chaotic search technology. The proposed algorithm overcomes the shortcoming of basic PSO, which is easy to fall into the local optimal solution. The chaos initialization and chaotic disturbance enhance

the ergodicity of the search. Besides, tabu list is introduced to improve the computation speed. In conclusion, the proposed algorithm can obtain the approximate-optimal solution of large-scale multi-cluster tools scheduling problem quickly and efficiently.

4) Based on TOC, a bottleneck-based push - pull scheduling algorithm is proposed from the viewpoint of scheduling optimality for the first time. By controlling the Takt of the bottleneck equipment, the approximate-optimality of solution is obtained surely. In addition, this thesis creatively combines the "pull" and "push" strategies to reduce the current residency time and improve the utilization of the multi-cluster tools.

## 6.3 Future works

As a highly complicate wafer fabrication system, multi-cluster tools production management issues closely related to the enterprise's production planning, workshop level control, inventory management, supply chain management and many other aspects. Due to the limited time, this thesis focuses on the inter-warehouse scheduling control problem of multi-cluster tools. Other related research can be carried out with the support of the research results.

The research on the modeling and scheduling of tree-like multi-cluster tool is still at the initial stage. The existing research focuses on the lower bound analysis of FP, and how to solve the problem is worthy of further study. Especially, if residency constraints and reentrant were considered in the study simultaneously, the complexity of the scheduling problem would be huge. On the issue of non-cyclic scheduling problem, this thesis does not combine the external random disturbance events of multi-cluster tools, such as downtime and emergency insertion. The scheduling problem of multi-cluster tool in uncertain environment is worthy of more deeply and extensive research.

It is a challenging task to study the modeling and scheduling problem of multi-cluster tool considering residency constraints. This thesis has carried out the beneficial exploration to several representative problems in this field, and put forward my own opinions.

# References


[1] Zhang, T., 2014. “Development strategy research for the foundary industry in China.” Southwestern University of Finance and Economics.

[2] Halim, Z., Rizwana, K., Shariq, B., Ghulam, A., 2016. “Artificial intelligence techniques for driving safety and vehicle crash prediction.” *Artificial Intelligence Review* 46(3):1-37.

[3] Turner, C.J., Hutabarat, W., Oyekan, J., Tiwari, A., 2016. “Discrete event simulation and virtual reality use in industry: new opportunities and future trends.” *IEEE Transactions on Human-Machine Systems* 46 (6): 882-894.

[4] Dustdar, S., 2016. “Cloud computing.” *Computer* 49 (2):12-13.

[5] Rosso, D., Global Semiconductor Sales Reaches $339 Billion in 2016. https://www.semiconductors.org/news/2017/02/02/global_sales_report_2017/global_semiconductor_sales_reach_339_billion_in_2016/.

[6] Yang, L., Wang, Y. N., 2016. “Physical hardware trojan failure analysis and detection method.” *Acta Physica Sinica* 65(11):49-57.

[7] Green, E.M., 1996. “Economic security and high technology competition in an age of transition: the case of the semiconductor industry.” *International Journal of Urban & Regional Research* 22 (1):166-168.

[8] Chu, M.C., 2009. “Globalisation and security : the migration of the Taiwanese semiconductor industry to China and its implications for the US-China-Taiwan security relations.” University of Cambridge.

[9] Li, L., Lu, R., Zang, J., 2016. “Scheduling model of cluster tools for concurrent processing of multiple wafer types.” *Mathematics in Practice and Theory* 46(16):152-161.

[10] Qiao, Y., 2015. “Modeling, scheduling and control of cluster tools in semiconductor manufacturing.” Guangdong University of Technology.

[11] Xiao, H., 2012. Introduction to Semiconductor Manufacturing Technology, Second Edition. Bellingham: SPIE Press.

[12] Wu, Q., Qiao, F., Li, L., Wang, Z., 2006. “Semiconductor manufacturing system scheduling.” Peking: Publishing House of Electronic Industry.

[13] Guo, C., 2012. “The research on scheduling wafer fabrication system with decomposition method and ant colony optimization algorithm.” Shanghai Jiaotong University.

[14] Zhang, J., Wu, L., Zhai, W., 2009. “Control of reusable manufacturing system.” Peking: Science Press.

[15] Van Zant P., 2000. “Microchip fabrication : a practical guide to semiconductor processing (6 edition).” New York: McGraw-Hill Education.

[16] Pan, C., Wu, N., 2009. “Scheduling of cluster tools in wafer fabrication.” *Computer Integrated Manufacturing Systems* 15 (3): 522-528.

[17] Gong, Q., 2012. “Scheduling and simulation of single-arm cluster tools with wafer reentrant process.” Guangdong University of Technology.

References


[18] Hill, R., 2005. “Characteristics and opportunities of semiconductor manufacturing.” *Electronic Engineering & Product World* 4 (A):112-112.

[19] Hu, H., Jiang, Z., Chen, K., 2009. “Composite scheduling method in semiconductor wafer fabrication.” *Journal of Shanghai Jiaotong University* 43(3):460-464.

[20] Shang, R., 2010. “An application of toc in semiconductor manufacturing.” Peking University.

[21] Burggraaf, P., 1995. “Coping with the high cost of wafer fabs.” *Semiconductor International* 18 (3): 45-54.

[22] Gupta, J.N.D., Ruiz, R., Fowler, J.W., Mason, S.J., 2006. “Operational planning and control of semiconductor wafer production.” *Production Planning & Control* 17(17):639-647.

[23] Bader, E.M., Hall, R., Strasser, G., 1990. “Integrated processing equipment.” *Solid State Technology* 33 (5): 149-154.

[24] Li, X., 2010. “Dynamic scheduling of cluster tool with residency constraints.” Shanghai Jiao Tong University.

[25] Quirk, M., Serda, J., Han, Z., 2004. “Semiconductor manufacturing technology.” Peking: Publishing House of Electronic Industry.

[26] Zheng, X., 2011. “Research on modeling and simulation methods for cluster tools in semiconductor manufacturing.” University of Chinese Academy of Sciences.

[27] Liu, S., 2012. “Cyclic scheduling of multi-cluster tools with temporal constraints.” Shanghai Jiao Tong University.

[28] Xu, J., Dai, G., Wang, H., 2004. “An overview of theories and methods of production scheduling.” *Journal of Computer Research and Developmen*t 41(2):257-267.

[29] Conway, R.W., Maxwell, W.L., Miller, L.W., 1967. “Theory of scheduling addison-wesley.” *Arte y parte: revista de arte - España, Portugal y América*.

[30] Zhu, Q., 2013. “Petri net modeling and optimal scheduling of multi-cluster tools.” Guangdong University of Technology.

[31] Dawande, M.W., Geismar, H.N., Sethi, S.P., Sriskandarajah, C., 2007. “Throughput optimization in robotic cells.” New York: Springer US.

[32] Dawande, M.W., Sriskandarajah, C., Sethi, S.P., 2002. “On throughput maximization in constant travel-time robotic cells.” *Manufacturing & Service Operations Management* 4(4):296-312.

[33] Perkinson, T.L., Mclarty, P.K., Gyurcsik, R.S., Cavin, R.K., 1994. “Single-wafer cluster tool performance: an analysis of throughput.” *IEEE Transactions on Semiconductor Manufacturing* 7(3):369-373.

[34] Venkatesh, S., Davenport, R., Foxhoven, P., Nulman, J., 1997. “A steady-state throughput analysis of cluster tools: dual-blade versus single-blade robots.” *Semiconductor Manufacturing IEEE Transactions on* 10(4):418-424.

[35] Geismar, H.N., Dawande, M., Sriskandarajah, C., 2004. “Robotic cells with parallel machines: throughput maximization in constant travel-time cells.” *Journal of Scheduling* 7(5):375-395.

[36] Pinedo, M., Hadavi, K., 1992. “Scheduling: theory, algorithms and systems development.” Berlin Heidelberg: Springer.

[37] Chen, J., Zhou, B., 2012. “Scheduling algorithm for cluster tools with residency and reentrant constraints” *Computer Integrated Manufacturing Systems* 18(12):2667-2673.

[38] Gao, Z., Zhou, B., 2016. “The scheduling and performance analysis of cluster tools with buffers based on branch searching.” *Acta Automatica Sinica* 42(1):81-88.

[39] Zhou, B., Li, M., 2016. “Scheduling method for double-cluster tools with parallel chambers based on capacity constraint resource.” *Journal of Beijing University of Aeronautics and Astronautics* 42 (7).

[40] Li, X., Zhou, B., Lu, Z., 2009. “Scheduling algorithm for cluster tools of wafer fabrications based on events-driven.” *Journal of Shanghai Jiaotong Univerisity* 43 (6):898-901.

[41] Gao, Z., Zhou, B., 2013. “Disjunctive graph-based modeling and scheduling for cluster tools.” *Journal of Shanghai Jiaotong University* 47(8):1227-1233.

[42] Lu, R., Li, L., 2014. “Research on Scheduling Problem of Cluster Tools with Residency Time Constraints.” *Journal of System Simulation* 26(8):1775-1780.

[43] Li, L., Hu, J., 2011. “Online scheduling problem of cluster tools with residency time constraints.” *Control and Decision* 26(1):37-43.

[44] Lu, R., 2015. “Framework model of scheduling system of cluster tool controller in semiconductor manufactory.” *Manufacturing Automation* (2):103-107.

[45] Kim, J.H., Lee, T.E., Lee, H.Y., Park, D.B., 2003. “Scheduling analysis of time-constrained dual-armed cluster tools.” *IEEE Transactions on Semiconductor Manufacturing* 16(3):521-534.

[46] Perkinson, T.L., Gyurcsik, R.S., Mclarty, P.K., 1996. “Single-wafer cluster tool performance: an analysis of the effects of redundant chambers and revisitation sequences on throughput.” *IEEE Transactions on Semiconductor Manufacturing* 9(3):384-400.

[47] Lee, H.Y., Lee, T.E., 2006. “Scheduling single-armed cluster tools with reentrant wafer flows.” *IEEE Transactions on Semiconductor Manufacturing* 19(2):226-240.

[48] Srinivasan, R.S., 1998. “Modeling and performance analysis of cluster tools using Petri nets.” *IEEE Transactions on Semiconductor Manufacturing* 11(3):394-403.

[49] Jung, C., Lee, T.E., 2012. “An efficient mixed integer programming model based on timed petri nets for diverse complex cluster tool scheduling problems.” *IEEE Transactions on Semiconductor Manufacturing* 25(2):186-199.

[50] Kim, D.K., Jung, C., Lee, T.E., Jung Y.J., 2012. “Cyclic scheduling of cluster tools with non-identical chamber access times.” *IEEE Transactions on Semiconductor Manufacturing* 25(25):2068-2079.

[51] Li, X., Fung, R.Y.K., 2015. “Optimal multi-degree cyclic solution of multi-hoist scheduling without overlapping.” *IEEE Transactions on Automation Science & Engineering* 14(2):1064-1074.

[52] Amraoui, A.E., Manier, M.A., Moudni, A.E., Benrejeb, M., 2010. “Genetic algorithm for a cyclic Hoist Scheduling Problem with time-window constraints and heterogeneous part jobs.” Marrakech: Control and Automation 20 (1):351-356.

[53] Ng, W.C., 1995. “A branch and bound algorithm for hoist scheduling of a circuit board production line.” *International Journal of Flexible Manufacturing Systems* 8(1):45-65.

[54] Che, A., Feng, J., Chen, H., Chu, C., 2015. “Robust optimization for the cyclic hoist scheduling problem.” *European Journal of Operational Research* 240(3):627–636.

[55] Che, A., Chu, C., 2008. “Optimal scheduling of material handling devices in a pcb production line: problem formulation and a polynomial algorithm.” *Mathematical Problems in Engineering*, 2008(4):267-290.

[56] Levner, E., Kats, V., Levit, V.E., 1997. “An improved algorithm for cyclic flowshop scheduling in a robotic cell.” *European Journal of Operational Research* 97(3):500-508.

[57] Sethi, S., Sriskandarajah, C., Sorger, G., Blazewicz, J., Kubiak, W., 2008. “Sequencing of parts and robot moves in a robotic cell.” *International Journal of Flexible Manufacturing Systems* 4(3-4):331-358.

[58] Dawande, M., Geismar, H.N., Sethi, S.P., Sriskandarajah, C., 2005. “Sequencing and Scheduling in Robotic Cells: Recent Developments.” *Journal of Scheduling* 8(5):387-426.

[59] Yan, P., Chu, C., Yang, N., Che, A., 2010. “A branch and bound algorithm for optimal cyclic scheduling in a robotic cell with processing time windows.” *International Journal of Production Research* 48(21):6461-6480.

[60] Qiao, L.H., Zhu, Y.X., Yang, J.J., Li, Y., 2009. “A petri net and genetic algorithm based method for flexible manufacturing cells modeling and scheduling.” *Key Engineering Materials* 407-408:268-272.

[61] Fathian, M., 2012. “Developing petri net model and meta-heuristic algorithms for cyclic scheduling in 2- machine robotic cells.” *African Journal of Business Management* 6(15):5456-5466.

[62] Al-Ahmari, A., 2015. “Optimal robotic cell scheduling with controllers using mathematically based timed Petri nets.” *Information Sciences* 329:638-648.

[63] Geismar, H.N., Sriskandarajah, C., Ramanan, N., 2004. “Increasing throughput for robotic cells with parallel machines and multiple robots.” *IEEE Transactions on Automation Science & Engineering* 1(1):84-89.

[64] Che, A., Chu, C., 2009. “Multi-degree cyclic scheduling of a no-wait robotic cell with multiple robots.” *European Journal of Operational Research* 199(1):77-88.

[65] Ding, S., Yi, J., 2004. “An event graph based simulation and scheduling analysis of multi-cluster tools.” Washington DC: *Proceedings of 2004 Winter Simulation Conference* 2:1915-1924.

[66] Zhu, Q.H., Wu, N.Q., Qiao, Y., Zhou, M.C., 2013. “Petri net-based optimal one-wafer scheduling of single-arm multi-cluster tools in semiconductor manufacturing.” I*EEE Transactions on Semiconductor Manufacturing* 26(4):578-591.

[67] Chan, W.K.V., Yi, J., Ding, S., Song, D., 2008. “Optimal scheduling of k-unit production of cluster tools with single-blade robots.” Arlington: IEEE International Conference on Automation Science and Engineering. IEEE, pp. 335-340.

[68] Chan, W.K.V., Ding, S., Yi, J., Song, D., 2011.” Optimal scheduling of multi-cluster tools with constant robot moving times, Part II: Tree-like topology configurations.” *IEEE Transactions on Automation Science Engineering* 8(1):17–28.

[69] Chan, W.K.V., Yi, J., Ding, S., 2011. “Optimal scheduling of multicluster tools with constant robot moving times, part i: two-cluster analysis.” *IEEE Transactions on Automation Science & Engineering* 8(1):5-16.

[70] Zhu, Q.H., Wu, N.Q., Qiao, Y., Zhou, M.C., 2014. “Modeling and schedulability analysis of single-arm multi-cluster tools with residency time constraints via Petri nets.” Taipei: *2014 IEEE International Conference on Automation Science and Engineering (CASE)* 2014:81-86.

[71] Zhu, Q.H., Wu, N.Q., Qiao, Y., Zhou M., 2015. “Scheduling of single-arm multi-cluster tools with wafer residency time constraints in semiconductor manufacturing.” *IEEE Transactions on Semiconductor Manufacturing* 28(1):117-125.

[72] Zhou, Z., Liu, J., 2008. “A heuristic algorithm for the two-hoist cyclic scheduling problem with overlapping hoist coverage ranges.” *Iie Transactions* 40(8):782-794.

[73] Chen, H., Chu, C., Proth, J.M., 1998. “Cyclic scheduling of a hoist with time window constraints.” *IEEE Transactions on Robotics & Automation* 14(1):144-152.

[74] Che, A., Chu, C., 2004. “Single-track multi-hoist scheduling problem: a collision-free resolution based on a branch-and-bound approach.” *International Journal of Production Research* 42(12):2435-2456.

[75] Che, A., Chu, C., 2007. “Cyclic hoist scheduling in large real-life electroplating lines.” *Or Spectrum* 29(3):445-470.

[76] Che, A., Yan, P., Yang, N., Chu, C., 2010. “Optimal cyclic scheduling of a hoist and multi-type parts with fixed processing times.” *International Journal of Production Research* 48(5):1225-1243.

[77] Lei, L., Liu, Q., 2001. “Optimal cyclic scheduling of a robotic processing line with two-product and time-window constraints.” *Infor Information Systems & Operational Research* 39(2):185-199.

[78] Chan, W.K.V., Roeder, T.M., 2010. “On gradient estimation of scheduling for multi-cluster tools with general robot moving times.” Toronto: IEEE Conference on Automation Science and Engineering, pp:112-117.

[79] Li, L., Hu, J., 2010. “Scheduling model and its algorithm for two-cluster tools with single-blade robot.” *Journal of System Simulation* 22(8): 1942-1946.

[80] Li, L., Hu, J., 2010. “K wafer cycle sequence problem in multi-cluster tools scheduling.” *Computer Integrated Manufacturing Systems* 16(1): 109-114, 126.

[81] Hu, J., Li, L., 2015. “Scheduling method for single-arm multi-cluster tools with residency time constraints.” Shenyang: IEEE International Conference on Cyber Technology in Automation, Control, and Intelligent Systems. IEEE, 2015:1574-1578.

[82] Zhou, Z., Li, L., 2009. “A solution for cyclic scheduling of multi-hoists without overlapping.” *Annals of Operations Research* 168(1):5-21.

[83] Che, A., Chu, C., 2005. “Multi-degree cyclic scheduling of two robots in a no-wait flowshop.” *IEEE Transactions on Automation Science & Engineering* 2(2):173-183.

[84] Sriskandarajah, C., Drobouchevitch, I., Sethi, S.P., Chandrasekaran, T., 2004. “Scheduling Multiple Parts in a Robotic Cell Served by a Dual-Gripper Robot.” *Operations Research* 52(1):65-82.

[85] Geismar, H.N., Dawande, M., Sriskandarajah, C., 2005. “Approximation algorithms for k -unit cyclic solutions in robotic cells.” *European Journal of Operational Research* 162(2):291-309.

[86] Geismar H N, Chan L M A, Dawande M, Sriskandarajah C., 2008. “Approximations to Optimal k -Unit Cycles for Single-Gripper and Dual-Gripper Robotic Cells.” *Production & Operations Management* 17(5):551–563.

[87] Zhou, Z., Che, A., Yan, P., 2012. “A mixed integer programming approach for multi-cyclic robotic flowshop scheduling with time window constraints.” *Applied Mathematical Modelling* 36(8):3621-3629.

[88] Kats V, Levner E, 2011. “Parametric algorithms for 2-cyclic robot scheduling with interval processing times.” *Journal of Scheduling* 14(3):267-279.

[89] Li X, Fung R Y, 2016. “Optimal K-unit cycle scheduling of two-cluster tools with residency constraints and general robot moving times.” *Journal of Scheduling* 19(2):165-176.

[90] Paul, H.J., Bierwirth, C., Kopfer, H., 2007. “A heuristic scheduling procedure for multi-item hoist production lines.” *International Journal of Production Economics* 105(1):54-69.

[91] Liu, M.X., Zhou, B.H., 2012. “Scheduling algorithm of multi-cluster tools based on time constraint sets.” *Acta Automatica Sinica* 38(3):479-485.

[92] Liu, M.X., Zhou, B.H., 2013. “Modelling and scheduling analysis of multi-cluster tools with residency constraints based on time constraint sets.” *International Journal of Production Research* 51(16):4835-4852.

[93] Zhou B, Liu M, Zhou S, 2014. “Scheduling method for dual-blade multi-cluster tools with residency constraints.” *Journal of Harbin Institute of Technology* 46(1):83-89.

[94] Zhou B.H., Li M., 2016. “Scheduling Method for Multi-cluster Tools with Diverse Wafer Flow Patterns.” *Journal of Northeastern University* 37(5):697-701.

[95] Leung, J., Zhang, G., 2003. “Optimal cyclic scheduling for printed circuit board production lines with multiple hoists and general processing sequence.” *Robotics & Automation IEEE Transactions on* 19(3):480 - 484.

[96] Leung, J.M.Y., Zhang, G., Yang, X., Lam, K., 2004. “Optimal Cyclic Multi-Hoist Scheduling: A Mixed Integer Programming Approach.” *Operations Research* 52(6):965-976.

[97] Zhou, Z., Li, H., 2002. “A heuristic method for one hoist dynamic scheduling.” *Systems Engineering —Theory Methodology Applications* 11(2):136-140.

[98] Chan, W.K.V., Yi, J., Ding, S., 2007. “On the Optimality of one-unit cycle scheduling of multi-cluster tools with single-blade robots.” Scottsdale: IEEE International Conference on Automation Science and Engineering. IEEE, 2007:392-397.

[99] Yi, J., Ding, S., Song, D., Zhang, M.T., 2008. “Steady-state throughput and scheduling analysis of multi-cluster tools: a decomposition approach”. *IEEE Transactions on Automation Science and Engineering* 5(2): 321-336.

[100] Yoon, H.J., Lee, D.Y., 2005. “Online scheduling of integrated single-wafer processing tools with temporal constraints.” *Semiconductor Manufacturing IEEE Transactions on* 18(3):390-398.

[101] Zhai, Y.N., Sun, S.D., Wang, J.Q., Guo, S.H., 2011. “Scheduling algorithm based on bottleneck operations decomposition for large-scale job shop scheduling problems.” *Computer Integrated Manufacturing Systems* 17(4): 826-831.

[102] Cong, M.Y., Wang, L.P., 2003. “Survey on the theory of meta-heuristic algorithms.” *Chinese High Technology Letters* 13(5):105-110.

[103] Lim, J.M., 1997. “A genetic algorithm for a single hoist scheduling in the printed-circuit-board electroplating line.” *Computers & Industrial Engineering* 33(3):789-792.

[104] Yang, G.W., Ju, D.P., Zheng, W.M., Lam, K., 2001. “Solving multiple hoist scheduling problems by use of simulated annealing.” *Journal of Software* 12(1):11-17.

[105] Zhou, Z., Wang, Y., 2007. “A search algorithm for cyclic scheduling of two hoists without overlapping partition.” *Systems Engineering* 25(4):104-109.

[106] Guo, C., Jiang, Z., Zhang, H., Li, N., 2012. “Decomposition-based classified ant colony optimization algorithm for scheduling semiconductor wafer fabrication system.” *Computers & Industrial Engineering* 62(1):141-151.

[107] Kennedy J., Eberhart R., 2002. “Particle swarm optimization.” *Perth: IEEE International Conference on Neural Networks, 1995 Proceedings;* 4(8):1942-1948.

[108] Li, P., & Che, A. D. 2009. “Chaos particle swarm optimization approach to robotic cells scheduling.” *Industrial Engineering Journal* 12(6): 90-95.

[109] Park, K., Morrison, J.R., 2010. “Control of wafer release in multi cluster tools.” Xiamen: IEEE International Conference on Control and Automation 2010:1481-1487.

[110] “SEMI E21, 1996. “Cluster tool module interface: mechanical interface and wafer transport standard”. Semiconductor Equipment and Materials International (SEMI), [Online]. Available: http://www.semi.org.

[111] Che, A., Hu, H., Chabrol, M., Gourgand, M., 2011. “A polynomial algorithm for multi-robot 2-cyclic scheduling in a no-wait robotic cell.” *Computers & Operations Research* 38(9):1275-1285.

[112] Dawande, M., Geismar, N., Pinedo, M., Sriskandarajah, C., 2010. “Throughput optimization in dual-gripper interval robotic cells.” *Iie Transactions* 42(1):1-15.

[113] Fahmy, S.A., Elmekkawy, T.Y., Balakrishnan, S., 2007. “Analysis of reactive deadlock-free scheduling in flexible job shops.” *International Journal of Flexible Manufacturing Systems* 19(3):264-285.

[114] Montgomery, D.C., 2013. “Design and analysis of experiments, 8th edition.” *Environmental Progress & Sustainable Energy* 32(1):8–10.

[115] **Wang, Z.**, & Zhou, B. H. (2015). Bottleneck-based scheduling method of multi-robot cells with residency constraints. *International Journal of Computer Integrated Manufacturing*, 28(12), 1237-1251.

[116] **Wang, Z.**, Zhou, B. H., Trenteseaux, D., Bekrar, A., (2017). Approximate optimal method for cyclic solutions in multi-robotic cell with processing time window. *Robotics and Autonomous Systems*, 98, 307-316.

[117] **Wang, Z.**, Zhou, B. H., Trenteseaux, D., Bekrar, A., (2015). An MIP approach to optimize the fundamental period of multi-cluster tools system with residency constraints. 15th IFAC Symposium on Information Control Problems in Manufacturing: INCOM 2015, *IFAC Papersonline, 48(3), 1732-1737*.

# Appendix A MPI-NLMIP model-based two-stage approximate-optimal scheduling algorithm

**MNB algorithm**. Scheduling problem of I-cluster tools with residency constraints and objective of minimize FP

1. Initialization

$P_{01} \leftarrow 0$, $P_{I0} \leftarrow (I-1)\times(x+1)+J+1$,

**for** $i \in [1, I-1]$ **do**

$P_{i0} \leftarrow (2I-i-1)\times(x+1)+J+1$;

$P_{0(i+1)} \leftarrow i\times(x+1)$;

**for** $j \in [1, x]$ **do**

$P_{ij} \leftarrow (i-1)\times(x+1)+j$;

**end**

**for** $j \in [x+1, J]$ **do**

$P_{ij} \leftarrow (2I-i-1)\times(x+1)+j$;

**end**

**end**

**for** $j \in [1, J]$ **do**

$P_{Ij} \leftarrow (I-1)\times(x+1)+j$;

**end**

$Count \leftarrow 0$, $T^* \leftarrow 0$;

2. Locate bottleneck module

**If** $temp = \max t^L_{p_{ij}}$, $BP \leftarrow P_{ij}$;

3. Initial schedule

$t_{p_{ij}} \leftarrow t^L_{p_{ij}}$, $T^0 \leftarrow \max t^L_{p_{ij}} + \theta$;

3.2 **While** $Count \le 1$ **do**

$t_{b_{i(i+1)}} \leftarrow 0$, $t_{b_{(i+1)i}} \leftarrow 0$;

**end**

**While** $Count \in [2, 10)$ **do**

$t_{b_{i(i+1)}} \leftarrow T^* - k\times(Count-1)\times\theta$

$t_{b_{(i+1)i}} \leftarrow T^* - k\times(Count-1)\times\theta$;

**end**

Feasible intervals of $S_{ij}$ is $\overline{S}_{ij} \leftarrow \left[S^{\min}_{ij}, S^{\max}_{ij}\right]$;

$S^0 \leftarrow \left\{S_{ij} \middle| i \in [1, I]; j \in [1, J]\right\}$;

4. Check and adjustment

**While** constr. (3-10) to (3-13) can't be satisfied simultaneously d**o**

**for** $P_{ij} \in [1, BP-1]$ **do**

**if** $S_{ij}+1 \le S_{ij}^{\max}$, $S_{ij} \leftarrow S_{ij}+1$;

**if** $S_{ij}+1 > S_{ij}^{\max}$, $T \leftarrow T+1$;

$S_{ij} \leftarrow S^0$;

**end**

**for** $P_{ij} \in [BP, 2\times(I-1)+I\times J]$ **do**

**if** $S_{ij}+1 \le S_{ij}^{\max}$, $S_{ij} \leftarrow S_{ij}+1$;

**if** $S_{ij}+1 > S_{ij}^{\max}$, $T \leftarrow T+1$;

$S_{ij} \leftarrow S^0$;

**end**

**end**

5. Verification and improve

$Count \leftarrow Count+1$

**While** $Count < 10$ **do**

**if** $LB \div T - 0.95 < 0$,

**while** $Count = 1$ **do**

$T^* \leftarrow T$

**end**

**while** $Count \in [2,10)$ **do**

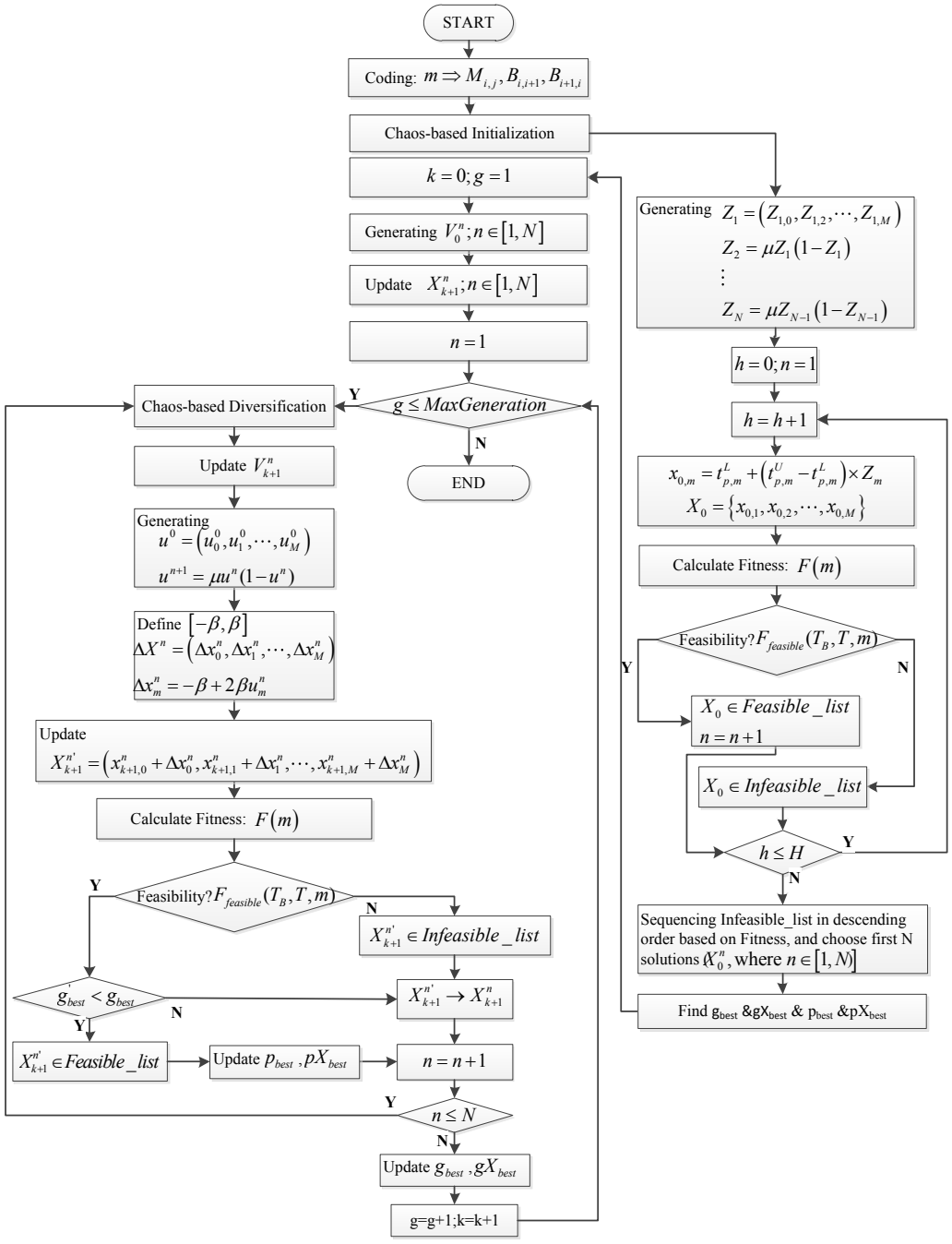
**if** $T^* > T$, $T^* \leftarrow T$;

**end**

**end**

6. Output $T$, $S$

# Appendix B The flow chart of the Chaos-based hybrid PSO-TS heuristic algorithm



# Résumé


**Chapitre 1. Introduction.**

En tant qu'industrie stratégique, le niveau technique de la fabrication des semi-conducteurs contraint le développement de l'économie nationale. Son niveau de développement est devenu un critère important pour mesurer le pouvoir national global d'un pays. Le système de fabrication des plaquettes est la partie la plus complexe et la plus coûteuse du processus de fabrication des semi-conducteurs. Son niveau d'ordonnancement de production a un impact significatif sur la rentabilité économique. Le Multi-cluster tools est un système de fabrication de plaquettes semi-conductrices du type couplage multi-boucles, largement utilisé dans la fabrication de plaquettes de 300 mm et 450 mm. Le problème d'ordonnancement du Multi-cluster tools présente les caractéristiques des modèles de flux de plaquettes complexe à grande échelle, des contraintes de résidences strictes et d'importants conflits de ressources, ce qui rend ce problème différent des autres problèmes d'ordonnancement d'un système de fabrication. Les articles existants montrent que la plupart des problèmes d'ordonnancement des systèmes de fabrication de plaquettes semi-conductrices sont des problèmes NP-hard, et il est difficile d'obtenir une solution optimale en utilisant un algorithme exact. Comment concevoir un algorithme d'ordonnancement heuristique efficace pour résoudre le problème d'ordonnancement du Multi-cluster tools est ainsi d'une grande importance pour promouvoir le développement de la théorie de l'ordonnancement et améliorer le niveau d'ordonnancement de production de l'industrie des semi-conducteurs. Il est devenu un sujet d'actualité dans les milieux universitaires et concerne également les services d'ingénierie dans les entreprises productrices.

Ce chapitre passe en revue les articles relevant de cette problématique. Nous aboutissons à la conclusion que la recherche actuelle sur le problème d'ordonnancement du Multi-cluster tools tient rarement compte des caractéristiques des contraintes de résidences, et la taille des systèmes étudiés est limitée à trois cluster tools. En plus, les méthodes d'ordonnancement sont principalement élaborées à partir de programmes mathématiques et de règles d'ordonnancement simples. Les résultats au problème d'ordonnancement non cyclique du Multi-cluster tools sont rares et

l'optimalité des algorithmes proposés n'est que rarement évaluée. En raison de sa grande complexité, la recherche sur le problème d'ordonnancement du Multi-cluster tools reste insuffisante. Par conséquent, cette thèse s'est focalisée sur le Multi-cluster tools comme objet de recherche. Notre travail prend en compte les contraintes de résidences, les contraintes de ressources et les modèles de flux de plaquettes. Sur cette base, des modèles d'ordonnancement seront établis, et des algorithmes d'ordonnancement heuristique efficace seront développés pour atteindre un ensemble d'objectifs de production.

**Chapitre 2.**

Ce chapitre présente une formalisation de la structure d'un Multi-cluster tools et des facteurs importants qui affectent son ordonnancement. Le Multi-cluster tools est une unité de fabrication intégrée composée d'un module de cassette, d'un module de traitement, d'un module tampon et d'un module de transport par robot. Afin de réaliser le processus de fabrication de plaquettes, le Multi-cluster tools présente des exigences très strictes sur l'environnement et les opérations à mener, notamment les contraintes de résidence et les contraintes de ressources. De plus, les indicateurs de performance des ordonnancement varient selon le modèle de flux de plaquettes. Ce chapitre décrit en détail la configuration du modèle de flux de plaquettes du Multi-cluster tools selon différents modèles de flux de plaquettes. Les indicateurs modélisés sont le temps de cycle le plus court pour une production cyclique, et dans le problème d'ordonnancement non cyclique, le Makespan le plus petit.

**Chapitre 3.**

La production de cycle à 1-unité en modèle de flux de plaquette unique est actuellement le mode de production le plus répandu. Il est relativement facile à mettre en œuvre et à contrôler. Afin de garantir la viabilité d'un ordonnancement, ce chapitre traite du problème de l'ordonnancement cyclique du type 1-unité du Multi-cluster tools en tenant compte des contraintes de, avec pour objectif de minimiser la période fondamentale (FP). Pour ce faire, la méthode par interdiction d'intervalle est utilisée pour éliminer efficacement l'espace de non-solution suite aux contraintes de résidence et de ressources ce qui peut entraîner un deadlock du Multi-cluster tools si elles ne sont pas correctement gérées. Ce chapitre propose également un NLMIP (programmation non linéaire à variables mixtes) avec l'objectif de minimiser la période fondamentale. La solution exacte des problèmes de petite taille est construite en utilisant CPLEX. Sur cette base, une borne inférieure est calculée. Pour la solution des problèmes de grande taille, le présent chapitre a conçu sur la base du modèle MPI-NLMIP, un algorithme d'ordonnancement approximatif à deux étages -

l'algorithme MNB. Dans la première étape, une solution réalisable est calculée en utilisant la méthode de recherche basée sur le principe du goulot; ensuite, en considérant la borne inférieure comme référence, un ordonnancement optimal approximatif est construit en exploitant une approche par bloc de temps.

Les résultats de simulation ont montré la faisabilité du modèle et de l'algorithme proposés. Ce dernier possède de bonnes performances: premièrement, le modèle MPI-NLMIP est capable de modéliser précisément les problèmes étudiés dans ce chapitre, et pour les problèmes de petite taille on peut utiliser CPLEX pour trouver la solution dans un temps raisonnable de calcul; deuxièmement, l'algorithme MNB présente une vitesse de calcul très rapide, la différence entre un minimum de FP et la limite inférieure de FP ne dépasse pas 19%, ce qui peut satisfaire le besoin d'ordonnancement d'une production réelle; troisièmement, même dans le cas où la distribution de la charge d'un équipement est extrêmement inégale, la MNB obtient encore une solution proche satisfaisante. Il atteint une performance optimale dans un Multi-cluster tools composé de 12 dispositifs; finalement, la phase d'ordonnancement optimal approximatif de l'algorithme MNB aide à améliorer la qualité de la solution, la solution d'ordonnancement finale est réalisable et ce, sans conflit de ressources.

**Chapitre 4.**

La production cyclique du type multi-unité est l'un des moyens les plus communs d'améliorer l'efficacité du système de fabrication des plaquettes mais ce type de système est extrêmement complexe à ordonnancer. En raison de l'augmentation du nombre et de la variété des plaquettes dans un temps de cycle, la concurrence de ressources dans le Multi-cluster tools est en effet encore plus forte, ce qui rend l'ordonnancement encore plus difficile. Ce chapitre se focalise sur le problème d'ordonnancement cyclique du type 2-unité du Multi-cluster tools avec les contraintes de résidences. Tout d'abord, le problème est décrit puis un NLMIP basé sur le MPI est présenté, avec l'objectif de minimiser le temps de cycle. Sur cette base, on trouve la solution en utilisant le logiciel CPLEX, et vérifie la validité de la solution et la faisabilité de ce programme d'ordonnancement. Ce chapitre présente également un algorithme heuristique de PSO-TS sur la base de la théorie du chaos. L'approche suggérée empêche l'algorithme de tomber dans un optimum local et permet de trouver une solution optimale approximative pour les problèmes à grande échelle.

Les résultats de simulation présentés mettent en évidence les bonnes performances du modèle et de l'algorithme proposés. Une analyse de l'impact du nombre de Multi-cluster tools et celui des modules de traitement sur le temps de calcul et le FP

minimal est réalisée. Dans le même temps, on a déterminé les limites d'application du modèle NLMIP qui est capable de traiter jusqu'à 20 Multi-cluster tools et 4 robots pour chacun. Ensuite, pour le problème d'ordonnancement cyclique du type 2-unité du Multi-cluster tools de petite taille dans lequel le temps de traitement de plaquettes suit la distribution normale ou la distribution uniforme, un modèle d'ordonnancement de NLMIP a été établi afin de trouver la solution en utilisant CPLEX dans un délai raisonnable. La solution est de très bonne qualité et les ordonnancement obtenus sont faisables et sans conflit de ressources. Troisièmement, par comparaison avec le PSO, l'algorithme proposé présente l'avantage du temps de calcul et de la qualité des solutions. Cet avantage s'amplifie avec la taille du problème.

**Chapitre 5.**

Avec l'augmentation de la demande d'ASIC, le mode de production non cyclique avec modèles multiples de flux de plaquettes est de plus en plus utilisé par des entreprises de fabrication de plaquettes semi-conductrices. Ce chapitre contient une étude du problème de la modélisation et l'ordonnancement non-cyclique du Multi-cluster tools avec des contraintes de résidences. L'objectif est maintenant de minimiser le Makespan de l'ordonnancement. Pour ce faire, un modèle de programmation non linéaire a été établi. Après analyse, on trouve et prouve une borne inférieure des problèmes d'ordonnancement. Étant donné que l'on étudie dans ce chapitre des problèmes NP-hard, il est difficile d'obtenir la solution optimale. Par conséquent, sur la base de la théorie des contraintes, un algorithme d'ordonnancement du type pression-traction sur le base de goulot, dénommé algorithme BP est proposé. Cet algorithme débute par le contrôle de la cadence de production du goulot du Multi-cluster tools. Par la méthode d'application de la stratégie de traction dans le module amont de goulot et pression dans le module en aval, on réduit le temps de résidence des plaquettes dans le Multi-cluster tools.

Afin de vérifier la validité de l'algorithme, une série d'expériences de simulation est effectuée. Les résultats montrent que l'algorithme de BP est plus rapide, et peut finir l'ordonnancement de 80 différentes variétés de plaquettes dans un temps relativement court. Dans la plupart des cas, l'algorithme BP permet d'obtenir un ordonnancement quasi-optimal. La structure du Multi-cluster tools, le nombre des plaquettes ainsi que les variétés de plaquette n'ont pas d'influence significative sur l'algorithme de BP. Deuxièmement, du point de vue de l'optimalité, en comparant BP avec la stratégie normale de traction, les résultats ont montré que l'algorithme de BP est plus flexible. Il y a peu de différence entre les résultats de l'algorithme de BP et la borne inférieure du problème d'ordonnancement. Troisièmement, la méthode d'analyse de la variance

(Anova) nous a permis de vérifier les résultats de ces expériences et analyser l'impact de paramètres sur l'algorithme. Il peut donc être conclu que l'algorithme BP est capable de résoudre efficacement le problème d'ordonnancement non cyclique du Multi-cluster tools en considérant les contraintes de résidences. L'ordonnancement obtenu est réalisable, et sa performance est très stable.

**Chapitre 6.Conclusions et travaux futurs.**

Dans le cadre d'un financement accordé par la Fondation nationale des sciences naturelles de la Chine, nos travaux ont porté sur la modélisation et l'algorithme d'ordonnancement du Multi-cluster tools en considérant les contraintes de résidence et les contraintes de ressources tenant en compte différents modèles de flux des plaquettes. Dans cette thèse, après avoir étudié les contributions de la littérature, un ensemble d'idées innovantes en recherche ont été présentées et défendues. La recherche aura été menée sur trois problèmes d'ordonnancement statiques: l'ordonnancement cyclique du type 1-unité avec flux unique de plaquettes, l'ordonnancement cyclique du type multi-unité avec flux unique de plaquettes, et l'ordonnancement non cyclique avec de multiples flux de plaquettes.

Tout d'abord, pour décrire formellement les caractéristiques qui distinguent le Multi-cluster tools des autres systèmes de fabrication, et pour montrer visuellement les relations logiques complexes entre les différentes ressources, en soulignant les caractéristiques du problème d'ordonnancement du Multi-cluster tools, la méthode par intervalle interdit est introduite de manière originale pour modéliser le problème d'ordonnancement du Multi-cluster tools. On a construit respectivement des modèles NLMIP des problèmes d'ordonnancement cycliques du type 1-unité et 2-unité sous contraintes de résidence sur la base de MPI. Deuxièmement, on a étudié de manière originale le problème d'ordonnancement cyclique du type 1-unité du Multi-cluster tools composé de plus de trois dispositifs. En tenant compte des contraintes de résidence, nous nous sommes approché des conditions réelle de production. Par rapport aux méthodes, on a proposé l'algorithme d'ordonnancement optimal et approximatif à deux étages basé sur MPI-NLMIP，et fournit un programme efficace pour résoudre le problème d'ordonnancement cyclique du type 1-unité à grande échelle. Troisièmement, on a étudié de manière originale le problème d'ordonnancement cyclique du type multi-unité du Multi-cluster tools composé de plus de trois dispositifs, et proposé l' algorithme heuristique de PSO-TS sur la base de la théorie du chaos. Il permet de surmonter les inconvénients des optimums locaux. L'utilisation de la technique de recherche chaotique a amélioré la qualité de la recherche. L'introduction de la liste tabou a amélioré la vitesse de calcul. Enfin, du

point de vue de l'optimalité d'ordonnancement et sur la base de la théorie des contraintes, cette thèse a proposé pour la première fois l'algorithme d'ordonnancement du type pression-traction sur le base du principe du goulot. Par le contrôle des cadences des équipements du goulot, cet algorithme garantit l'obtention d'une solution optimale approximative. Cet algorithme original combine les stratégies de traction et de pression, ce qui permet de réduire le temps de résidence tout en améliorant le taux d'utilisation du Multi-cluster tools.

La recherche sur la modélisation et l'ordonnancement du Multi-cluster tools est récente, et la plupart des recherches existantes se concentrent sur l'analyse d'une borne inférieure du cycle élémentaire. C'est pourquoi il est intéressant de faire une étude plus approfondie sur la résolution optimale du problème d'ordonnancement. En particulier, en tenant compte des contraintes de résidence et de ré-entrance, la complexité du problème d'ordonnancement s'accroit très fortement. Dans cette thèse, nous n'avons pas pris en compte les événements aléatoires et les perturbations externes du Multi-tools en compte, par exemple une insertion urgente. Le problème d'ordonnancement du Multi-cluster tools dans un environnement incertain mérite une étude plus approfondie et plus large. Cette thèse a néanmoins exploré plusieurs questions scientifiques représentatives dans ce domaine et a proposé un ensemble de modèles originaux pour résoudre les problèmes correspondants. Nous avons identifié par la même occasion un ensemble très intéressant de perspectives à aborder dans un futur proche, le monde des semi-conducteurs devenant au fur et à mesure que les années passent, un domaine stratégique pour un très grand nombre de pays les produisant.