基于时间 Petri 网与蚁群算法的晶圆制造系统调度问题研究

作者姓名 董扬平

指导教师姓名、职称 周孟初 教授

申请学位类别 工学硕士

学校代码 10701　　学　号 1101110071
分 类 号 TN82　　密　级 秘密

西安电子科技大学

硕士学位论文

基于时间 Petri 网与蚁群算法的晶圆制造系统调度问题研究

作者姓名：董扬平

一级学科：控制科学与工程

二级学科（研究方向）：控制理论与控制工程

学位类别：工学硕士

指导教师姓名、职称：周孟初 教授

学　　院：机电工程学院

提交日期：20xx 年 x 月

Research on the scheduling problem of wafer manufacturing system based on temporal Petri network and ant colony algorithm

A thesis submitted to

XIDIAN UNIVERSITY

in partial fulfillment of the requirements

for the degree of Master

in Control Science and Engineering

By

Dong Yangping

Supervisor: Zhou Mengchu Title: Professor

February 2015

西安电子科技大学
学位论文独创性（或创新性）声明

秉承学校严谨的学风和优良的科学道德，本人声明所呈交的论文是我个人在导师指导下进行的研究工作及取得的研究成果。尽我所知，除了文中特别加以标注和致谢中所罗列的内容以外，论文中不包含其他人已经发表或撰写过的研究成果；也不包含为获得西安电子科技大学或其它教育机构的学位或证书而使用过的材料。与我一同工作的同事对本研究所做的任何贡献均已在论文中作了明确的说明并表示了谢意。

学位论文若有不实之处，本人承担一切法律责任。

本人签名：_____　　日　　期：_____

西安电子科技大学
关于论文使用授权的说明

本人完全了解西安电子科技大学有关保留和使用学位论文的规定，即：研究生在校攻读学位期间论文工作的知识产权属于西安电子科技大学。学校有权保留送交论文的复印件，允许查阅、借阅论文；学校可以公布论文的全部或部分内容，允许采用影印、缩印或其它复制手段保存论文。同时本人保证，结合学位论文研究成果完成的论文、发明专利等成果，署名单位为西安电子科技大学。

保密的学位论文在_____年解密后适用本授权书。

本人签名：_____　　导师签名：_____

日　　期：_____　　日　　期：_____

摘要

本文针对晶圆制造系统，提出了一种新的时间 Petri 网子类，变迁库所时间网对其进行建模，并设计了一套变迁发射时 Petri 网时间更新机制，并使用蚁群算法对其进行调度。在蚁群算法中，将每一只蚂蚁看作一个调度器，通过模拟蚂蚁在 Petri 网可达图上的移动和信息交流，实现了对系统的调度优化。通过实验比较，发现蚁群算法无法得到最优解。于是提出超级蚂蚁机制、贪心选取初始信息素、使用贪心算法预添加信息素、复活蚂蚁、蚁群算法与模拟退火算法结合、蚂蚁回溯这 6 种优化思路，经过实验比较分析，最终蚂蚁回溯算法有不错的效果，证明了该方法在优化系统调度效率和降低制造成本方面的有效性。本文的研究为晶圆制造系统的调度问题提供了一种新的优化思路和方法。

关键词：晶圆制造系统，Petri 网，时间 Petri 网，蚁群算法，回溯算法

ABSTRACT

In this paper, a new temporal Petri network subclass is proposed for the wafer fabrication system, which is modeled by the time network of the transition library, and a set of Petri net time update mechanism at the time of transition launch is designed, and it is scheduled by ant colony algorithm. In the ant colony algorithm, each ant is regarded as a scheduler, and the scheduling optimization of the system is realized by simulating the movement and information exchange of ants on the Petri network reachability map. Through experimental comparison, it is found that the ant colony algorithm cannot obtain the optimal solution.

Therefore, six optimization ideas are proposed: super ant mechanism, greedy selection of initial pheromones, pre-addition of pheromones using greedy algorithm, resurrection of ants, combination of ant colony algorithm and simulated annealing algorithm, and ant backtracking. After experimental comparison and analysis, the ant backtracking algorithm finally has a good effect.

The effectiveness of this method in optimizing system scheduling efficiency and reducing manufacturing costs is proved.

The research in this paper provides a new optimization idea and method for the scheduling problem of wafer fabrication system.

Keywords: Wafer fabrication systems，Petri Net，Time Petri Net，Ant colony algorithm，Backtracking algorithm

插图索引

表格索引

符号对照表

符号	符号名称
$[N]$	关联矩阵
$[N]^+$	输出关联矩阵
$[N]^-$	输入关联矩阵
$\mathbb{N}$	自然数集合
$\mathbb{N}^+$	正整数集合
N	Petri 网
P	库所集合
p	一个库所
$p^\bullet$	库所 p 的后置
${}^\bullet p$	库所 p 的前置
$\mathbb{Q}_0^+$	正有理数集合
$R(N, M_0)$	网 (N, M_0) 的状态空间
$RG(N, M_0)$	网 (N, M_0) 的可达图
$\mathbb{R}^+$	正实数集合
T	变迁集合
t	一个变迁
$t^\bullet$	变迁 t 的后置
${}^\bullet t$	变迁 t 的前置

缩略语对照表

缩略语	英文全称	中文对照
TPN	Time Petri Net	时间 Petri 网
TdPN	Timed Petri Net	时延 Petri 网
TTPN	Time Transition Petri Net	带时间变迁的 Petri 网
TPPN	Time Place Petri Net	带时间库所的 Petri 网
TAPN	Time Arc Petri Net	带时间弧的 Petri 网
TTPPN	Time Transition Place Petri Net	变迁库所时间网

目录

第一章　研究生学位论文撰写的总体要求

研究生学位论文必须是学位申请者本人在导师的指导下独立完成的学术成果，要求立论正确，推理严谨，数据可靠，层次分明，文字通畅，不得抄袭和剽窃他人成果。研究生学位论文的撰写语言为中文和英文两种，硕士学位论文篇幅一般不低于 3 万字，博士学位论文篇幅一般不低于 5 万字。

研究生学位论文中使用的术语、符号、代号必须全文统一并符合规范化要求，计量单位一律采用国务院 1984 年 2 月 27 日发布的《中华人民共和国法定计量单位》标准。

第二章　研究生学位论文撰写的内容要求

我校研究生学位论文包括以下几个部分：

2.1　封面

（1）题目：题目是以最恰当、最简明的词语反映论文中最重要的特定内容的逻辑组合，力求简短切题。中文题目（包括副标题和标点符号）一般不超过 20 个字，英文题目一般不超过 10 个实词。

题目位于确定位置的文本框中，文本框格式为水平位置：相对于右侧页边距绝对位置 0.5 厘米；垂直位置：相对于下侧页边距绝对位置 15 厘米；文字环绕方式为浮于文字上方；文本框大小：高度为绝对值 3.2 厘米，宽度为绝对值 15 厘米。文字格式为中文宋体、英文 Times New Roman，二号加粗，居中对齐，左右不缩进，段前段后不留空，行距为固定值 30 磅。

（2）责任者姓名：包括论文作者姓名、指导教师姓名及职称（博士学位论文、学术型和同等学力硕士学位论文）以及学校、企业导师姓名及职称（专业学位硕士学位论文）。没有企业导师的专业学位类别请将“企业导师姓名及职称”栏目删除。

（3）申请学位类别：按照学科门类和学位层次填写，如工学博士、工学硕士、工程硕士、工商管理硕士等。

作者姓名、指导教师姓名职称、申请学位类别信息位于确定位置的文本框中，文本框格式为水平位置：相对于右侧页边距绝对位置 3.5 厘米；垂直位置：相对于下侧页边距绝对位置 20 厘米；文字环绕方式为浮于文字上方；文本框大小：高度为绝对值 3.5 厘米，宽度为绝对值 9 厘米。标题字体为黑体四号加粗，具体内容的文字格式为中文宋体、英文 Times New Roman，四号加粗，左对齐，左右不缩进，段前段后不留空，行距为固定值 30 磅。

2.2　题名页

题名页包括中文题名页和英文题名页，主要由学校代码、分类号、学号、密级、论文题目、作者姓名、一级学科、二级学科（博士学位论文、学术型和同等学力硕士学位论文）、领域（专业学位硕士学位论文）、学位类别、指导教师姓名、职称（博士学位论文、学术型和同等学力硕士学位论文）、学校、企业导师姓名、职称（专业学位硕士学位论文）、提交日期等部分组成。没有企业导师的专业学位类别请将“企业导师姓名及职称”栏目删除。

（1）学校代码：指本单位编号，我校代码是“10701”。

（2）分类号：指在《中国图书资料分类法》中的分类号（填写前四位即可）。

（3）学号：按照入学时研究生院编制的统一编号填写。

（4）密级：密级由导师确定，分为公开和秘密两种。

学校代码和分类号位于确定位置的文本框中，文本框格式为水平位置：相对于右侧页边距绝对位置 0.2 厘米；垂直位置：相对于下侧页边距绝对位置 0.3 厘米；文字环绕方式为浮于文字上方；文本框大小：高度为绝对值 1.1 厘米，宽度为绝对值 4.5 厘米。学号和密级位于确定位置的文本框中，文本框格式为水平位置：相对于右侧页边距绝对位置 10.9 厘米；垂直位置：相对于下侧页边距绝对位置 0.3 厘米；文字环绕方式为浮于文字上方；文本框大小：高度为绝对值 1.1 厘米，宽度为绝对值 4.5 厘米。中文题名页中的学校代码、分类号、学号和密级的字体为宋体，字号为五号加粗，行距为多倍行距 1.2，段落间距为段前 0 磅，段后 0 磅；

学位论文题目位于确定位置的文本框中，文本框格式为水平位置：相对于右侧页边距绝对位置 0 厘米；垂直位置：相对于下侧页边距绝对位置 11 厘米；文字环绕方式为浮于文字上方；文本框大小：高度为绝对值 3.2 厘米，宽度为绝对值 15.5 厘米。字体为宋体，字号为二号加粗，行距为固定值 30 磅，段落间距为段前 0 磅，段后 0 磅；

作者姓名、指导教师姓名职称、一级学科、二级学科、领域、学位类别、提交日期位于确定位置的文本框中，文本框格式为水平位置：相对于右侧页边距绝对位置 4.5 厘米；垂直位置：相对于下侧页边距绝对位置 16 厘米；文字环绕方式为浮于文字上方；文本框大小：高度为绝对值 8.6 厘米，宽度为绝对值 10.5 厘米。标题和具体内容的字体为宋体，标题字号为四号加粗，具体内容的字号为四号不加粗，行距为固定值 32 磅，段落间距为段前 0 磅，段后 0 磅。

英文题名页中的学科填写一级学科（专业学位填写类别），学位论文题目字体为 Times New Roman，字号二号加粗，行距为固定值 30 磅，段落间距为段前 0 磅，段后 0 磅，其他内容的字体为 Times New Roman，字号三号，行距为固定值 30 磅，段落间距为段前 0 磅，段后 0 磅。学位论文题目位于确定位置的文本框中，文本框格式为水平位置：相对于右侧页边距绝对位置 0 厘米；垂直位置：相对于下侧页边距绝对位置 0 厘米；文字环绕方式为浮于文字上方；文本框大小：高度为绝对值 3.5 厘米，宽度为绝对值 15.5 厘米。学科信息文本框格式为水平位置：相对于右侧页边距绝对位置 0 厘米；垂直位置：相对于下侧页边距绝对位置 6 厘米；文字环绕方式为浮于文字上方；文本框大小：高度为绝对值 5.5 厘米，宽度为绝对值 15.5 厘米。作者信息文本框格式为水平位置：相对于右侧页边距绝对位置 0 厘米；垂直位置：相对于下侧页边距绝对位置 18.7 厘米；文字环绕方式为浮于文字上方；文本框大小：高度为绝对值 4.5

厘米，宽度为绝对值 15.5 厘米。

2.3　声明

声明是对学位论文创新性和使用授权的声明和说明，论文提交图书馆和存档时作者本人和指导教师必须签名确认。

声明部分标题字体为宋体，字号为四号加粗，居中排列，行距为固定值 20 磅，段落间距为段前 0 磅，段后 0 磅；正文字体为宋体，字号为小四号，行距为固定值 20 磅，段落间距为段前 0 磅，段后 0 磅；标题与正文之间空一行，签名行与正文之间空一行，日期行与签名行之间空一行。

2.4　摘要

摘要是学位论文的内容不加注释和评论的简短陈述，简明扼要陈述学位论文的研究目的、内容、方法、成果和结论，重点突出学位论文的创造性成果和观点。摘要包括中文摘要和英文摘要，硕士学位论文中文摘要字数一般为 1000 字左右，博士学位论文中文摘要字数一般为 1500 字左右。英文摘要内容与中文摘要内容保持一致，翻译力求简明精准。摘要的正文下方需注明论文的关键词，关键词一般为 3 ～8 个，关键词和关键词之间用逗号并空一格。

中文摘要标题字体为黑体，字号为三号，居中排列，行距为固定值 20 磅，段落间距为段前 24 磅，段后 18 磅；正文字体为宋体，字号为小四号，行距为固定值 20 磅，段落间距为段前 0 磅，段后 0 磅；关键词和正文之间空一行，关键词字体为宋体，字号为小四号，标题加粗。英文摘要标题字体为 Times New Roman，字号为三号，居中排列，行距为固定值 20 磅，段落间距为段前 24 磅，段后 18 磅；正文的每一段落首行不空格，段落与段落之间空一行；正文字体为 Times New Roman，字号为小四号，行距为固定值 20 磅，段落间距为段前 0 磅，段后 0 磅；关键词字体为 Times New Roman，字号为小四号，标题加粗。

2.5　插图索引

学位论文中插图的目录索引。插图索引标题字体为黑体，字号为三号，居中排列，行距为固定值 20 磅，段落间距为段前 24 磅，段后 18 磅；正文内容字体为宋体，字号为小四号，行距为固定值 20 磅，段落间距为段前 0 磅，段后 0 磅。

2.6 表格索引

学位论文中表格的目录索引。表格索引标题字体为黑体，字号为三号，居中排列，行距为固定值 20 磅，段落间距为段前 24 磅，段后 18 磅；正文内容字体为宋体，字号为小四号，行距为固定值 20 磅，段落间距为段前 0 磅，段后 0 磅。

2.7 符号对照表

学位论文中符号代表的意义及单位（或量纲）的说明。符号对照表标题字体为黑体，字号为三号，居中排列，行距为固定值 20 磅，段落间距为段前 24 磅，段后 18 磅；正文内容字体为宋体，字号为小四号，行距为固定值 20 磅。

2.8 缩略语对照表

学位论文中缩略语代表意义的说明。缩略语按照英文单词首字母顺序排列，对照表标题字体为黑体，字号为三号，居中排列，行距为固定值 20 磅，段落间距为段前 24 磅，段后 18 磅；正文内容中文字体为宋体，字号为小四号，英文字体为 Times New Roman，字号为小四号，行距为固定值 20 磅。

2.9 目录

目录是学位论文的提纲，是论文各组成部分的小标题，应分别依次列出并注明页码。各级标题分别以第一章、1.1、1.1.1 等数字依次标出，目录中最多列出三级标题，正文中如果确需四级标题，用（1）、（2）形式标出。学位论文的前置部分（摘要、插图索引、表格索引、符号对照表、缩略语对照表）和学位论文的主体部分（正文、参考文献、致谢、作者简介）都要在目录中列出。

目录标题字体为黑体，字号为三号，居中排列，行距为固定值 20 磅，段落间距为段前 24 磅，段后 18 磅；目录内容中一级标题字体为黑体，字号为小四号，其余标题字体为宋体，字号为小四号。

2.10 正文

正文是学位论文的主体和核心部分。正文的一级标题居中排列，字体为黑体，字号为三号，行距为固定值 20 磅，段落间距为段前 24 磅，段后 18 磅；二级标题不缩进，字体为宋体加粗，字号为小三号，行距为固定值 20 磅，段落间距为段前 18 磅，段后 12 磅；三级标题缩进 2 字符，字体为宋体，字号为四号加粗，行距为固定值 20

磅，段落间距为段前 12 磅，段后 6 磅；正文内容字体为宋体，字号为小四号，行距为固定值 20 磅，段落间距为段前 0 磅，段后 0 磅。正文一般包括以下几个方面：

2.10.1　绪论

绪论是学位论文主体部分的开端，切忌与摘要雷同或成为摘要的注解。绪论除了要说明论文的研究目的、研究方法和研究结果外，还应评述与论文研究内容相关的国内外研究现状和相关领域中已有的研究成果；其次还要介绍本项研究工作的前提和任务、理论依据、实验基础、涉及范围、预期结果以及该论文在已有基础上所要解决的问题。

2.10.2　各章节

各章节一般由标题、文字叙述、图、表、公式等构成，章节内容总体要求立论正确，逻辑清晰，数据可靠，层次分明，文字通畅，编排规范。论文中若有与指导教师或他人共同研究的成果，必须明确标注；如果引用他人的结论，必须明确注明出处，并与参考文献保持一致。

（1）图：包括曲线图、示意图、流程图、框图等。图序号一律用阿拉伯数字分章依序编码，如：图 1.3、图 2.11。每一个图应有简短确切的图名，连同图序号置于图的正下方。图名称、图中的内容字号为五号，中文字体为宋体，英文字体为 Times New Roman，行距一般为单倍行距。图中坐标上标注的符号和缩略词必须与正文保持一致。引用图应在图题右上角标出文献来源；曲线图的纵横坐标必须标注“量、标准规定符号、单位”，这三者只有在不必要标明（如无量纲等）的情况下方可省略。

（2）表：包括分类项目和数据，一般要求分类项目由左至右横排，数据从上到下竖列。分类项目横排中必须标明符号或单位，竖列的数据栏中不要出现“同上”、“同左”等词语，一律要填写具体的数字或文字。表序号一律用阿拉伯数字分章依序编码，如：表 2.5、表 10.3。每一个表格应有简短确切的题名，连同表序号置于表的正上方。表名称、表中的内容字号为五号，中文字体为宋体，英文字体为 Times New Roman，行距一般与正文保持一致。

（3）公式：正文中的公式、算式、方程式等必须编排序号，序号一律用阿拉伯数字分章依序编码，如：(3-32)、(6-21)。对于较长的公式，另起行居中横排，只可在符号处（如：+、-、*、/、<> 等）转行。公式序号标注于该式所在行（当有续行时，应标注于最后一行）的最右边。连续性的公式在“=”处排列整齐。大于 999 的整数或多于三位的小数，一律用半个阿拉伯数字符的小间隔分开；小于 1 的数应将 0 置于小数点之前。公式的行距一般为单倍行距。

（4）计量单位：学位论文中出现的计量单位一律采用国务院 1984 年 2 月 27 日发

布的《中华人民共和国法定计量单位》标准。

2.10.3 结论

结论是学位论文最终和总体的结论，不是正文中各段的小结的简单重复，应准确、精炼、完整，其中要着重阐述作者研究的创造性成果以及在本研究领域中的重大意义，还可提出有待进一步研究和探讨的问题。

2.11 参考文献

参考文献是文中引用的有具体文字来源的文献集合，博士学位论文参考文献一般不少于 80 篇，其中近 5 年的参考文献不少于 20 篇，硕士学位论文参考文献一般不少于 30 篇，其中近 5 年的参考文献不少于 5 篇。参考文献标题字体为黑体，字号为三号，居中排列，段落间距为段前 24 磅，段后 18 磅；参考文献若是中文文献，字体为宋体，字号为五号，若是英文文献，字体为 Times New Roman，字号为五号。学位论文的撰写要本着严谨求实的科学态度，凡有引用他人成果之处，引用处右上角用方括号标注阿拉伯数字编排的序号（必须与参考文献一致），同时所有引用的文献必须用全称，不能缩写，并按论文中所引用的顺序列于文末。引用文献的作者不超过 3 位时全部列出，超过时列前 3 位，后加“等”字或“et al.”。参考文献的著录要符合《文后参考文献著录规则》（GB/T7714-2005）要求：

（1）期刊 (报纸) 参考文献：[序号] 主要责任者. 文献名称 [文献类别代码]. 期刊 (报纸) 名, 年份, 卷 (期): 引文页码.

（2）专著参考文献：[序号] 主要责任者. 专著名称 [文献类别代码]. 其他责任者. 出版地: 出版单位, 出版年份.

（3）专利参考文献：[序号] 主要责任者. 专利名称: 国别, 专利号 [文献类别代码]. 出版日期.

（4）技术标准参考文献：[序号] 起草责任者. 标准代号-标准顺序号-发布年. 标准名称 [文献类别代码]. 出版地: 出版单位, 出版年份.

（5）电子参考文献：[序号] 主要责任者. 题名 [文献类别代码]. 获取和访问路径. [引用日期].

（6）会议论文集参考文献：[序号] 编者. 论文集名. (供选择项：会议名, 会址, 开会年) 出版地: 出版者, 出版年份.

（7）学位论文参考文献：[序号] 主要责任者. 文献题名 [文献类别代码]. 保存地: 保存单位, 年份.

（8）国际、国家标准参考文献：[序号] 标准代号. 标准名称 [文献类别代码]. 出版地: 出版者, 出版年.

（9）报告类参考文献：[序号] 主要责任者. 文献题名 [文献类别代码]. 报告地: 报告会主办单位, 年份.

参考文献著录中的文献类别代码：

（1）普通图书：M

（2）会议录：C

（3）汇编：G

（4）报纸：N

（5）期刊：J

（6）学位论文：D

（7）报告：R

（8）标准：S

（9）专利：P

（10）数据库：DB

（11）计算机程序：CP

（12）电子公告：EB

载体类型：

网络：OL

磁带：MT

磁盘：MK

光盘：CD

2.12　致谢

作者对完成论文提供帮助和支持的组织和个人表示感谢的文字记载。致谢标题字体为黑体，字号为三号，居中排列，行距为固定值 20 磅，段落间距为段前 24 磅，段后 18 磅；正文内容字体为宋体，字号为小四号，行距为固定值 20 磅，段落间距为段前 0 磅，段后 0 磅。

2.13　作者简介

对作者的简要介绍，主要包括个人基本情况、教育背景、攻读博士/硕士学位期间的研究成果等三个部分内容。攻读博士/硕士学位期间的研究成果是指本人攻读博士/硕士学位期间发表（或录用）的学术论文，申请（授权）专利、参与的科研项目及科研获奖等情况，分别按时间顺序列出。其中，发表论文、申请（授权）专利、科研获奖只列出作者排名前 3 名的，参与的科研项目按重要程度最多列出 5 项。作者简介

标题字体为黑体，字号为三号，居中排列，行距为固定值 20 磅，段落间距为段前 24 磅，段后 18 磅。作者简介的正文内容严格按照本模板中的范例书写。

2.14 其他

学位论文中如果需要注释，可作为脚注在页下分别著录，切忌在文中注释；如果有附录部分，可编写在正文之后，与正文连续编页码，每一附录均另页起，附录依次用大写英文字母 A、B、C……编序号，如：附录 A、附录 B 等。

第三章　研究生学位论文的编辑、打印、装订要求

3.1　学位论文封面的编辑和打印要求

学位论文的封面由研究生院按国家规定统一制定印刷，封面内容必须打印，不得手写。

3.2　学位论文的版面设置要求

（1）行间距：固定值 20 磅（题名页除外）。

（2）字符间距：标准。

（3）页眉设置：单面页码页眉标题为章节题目，每一章节的起始页必须在单面页码，双面页码页眉标题统一为“西安电子科技大学博/硕士学位论文”，页眉标题居中排列，字体为宋体，字号为五号。页眉文字下添加双横线，双横线宽度为 0.5 磅，距正文距离为：上下各 1 磅，左右各 4 磅。

（4）页码设置：学位论文的前置部分和主体部分分开设置页码，前置部分的页码用罗马数字标识，字体为 Times New Roman，字号为小五号；主体部分的页码用阿拉伯数字标识，字体为宋体，字号为小五号。页码统一居于页面底端中部，不加任何修饰。

（5）页面设置：为了便于装订，要求每页纸的四周留有足够的空白边缘，其中页边距为上 3 厘米、下 2 厘米；内侧 2.5 厘米、外侧 2.5 厘米；装订线为 0.5 厘米；页眉 2 厘米，页脚 1.75 厘米。

3.3　学位论文的打印、装订要求

（1）打印：学位论文必须用 A4 纸页面排版，双面打印；

（2）装订：依次按照中文题名页、英文题名页、声明、摘要、插图索引、表格索引、符号对照表、缩略语对照表、目录、正文、附录（可选）、参考文献、致谢、作者简介的顺序，用学校统一印制的学位论文封面装订成册。盲审论文必须删除致谢部分的文字内容（致谢标题须保留）以及封面和研究成果中的作者和指导教师姓名，研究成果列表中应体现作者的排序，如第一作者、第一发明人等。

3.4 其他说明

本规定由研究生院负责解释，从申请 2015 年 9 月毕业和授位的研究生开始执行，其它有关规定同时废止。研究生毕业论文撰写要求参照学位论文撰写要求执行。

第四章　图、表、公式示例

图：包括曲线图、示意图、流程图、框图等。图序号一律用阿拉伯数字分章依序编码，如：图 1.3、图 2.11。

每一个图应有简短确切的图名，连同图序号置于图的正下方。图名称、图中的内容字号为五号，中文字体为宋体，英文字体为 Times New Roman，行距一般为单倍行距。图中坐标上标注的符号和缩略词必须与正文保持一致。引用图应在图题右上角标出文献来源；曲线图的纵横坐标必须标注“量、标准规定符号、单位”，这三者只有在不必要标明（如无量纲等）的情况下方可省略。

图与正文之间一般应空一行。

公式：正文中的公式、算式、方程式等必须编排序号，序号一律用阿拉伯数字分章依序编码，如：(3-32)、(6-21)。

对于较长的公式，另起行居中横排，只可在符号处（如：+、-、*、/、<> 等）转行。公式序号标注于该式所在行（当有续行时，应标注于最后一行）的最右边。连续性的公式在“=”处排列整齐。大于 999 的整数或多于三位的小数，一律用半个阿拉伯数字符的小间隔分开；小于 1 的数应将 0 置于小数点之前。公式的行距一般为单倍行距。

公式与正文之间一般应空一行。

$$X_{e1}(s,n_1,k_1)=\begin{pmatrix}k_1\\ s\end{pmatrix}\frac{n_1!}{(n_1-s)!}\sum_{v=0}^{\min(n_1-s,k_1-s)}(-1)^v\begin{pmatrix}k_1-s\\ v\end{pmatrix}\times\frac{(n_1-s)!}{(n_1-s-v)!}(n_1-s-v)^{k_1-s-v} \tag{4-1}$$

表：包括分类项目和数据，一般要求分类项目由左至右横排，数据从上到下竖列。

分类项目横排中必须标明符号或单位，竖列的数据栏中不要出现“同上”、“同左”等词语，一律要填写具体的数字或文字。表序号一律用阿拉伯数字分章依序编码，如：表 2.5、表 10.3。

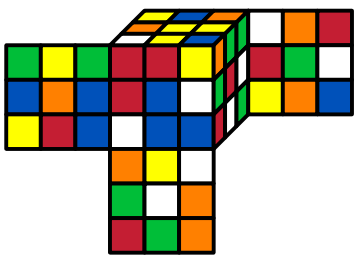


图 4.1　插图示例

表 4.1　表格示例

馈电方式 / 电性能参数	探针	环形缝隙	探针和缝隙		缝隙和 CPW	
			探针	缝隙	缝隙	CPW
谐振频率	9.5 GHz	8.8 GHz	9.4 GHz	9.8 GHz	9.2 GHz	9.3 GHz
带宽 （$\|S_{11}\|$ <-10 dB）	7.3%	4.5%	6.9%	6.8%	4.9%	5.3%
隔离度 （带内最差）	-16.5 dB	-17 dB	-31 dB		-22 dB	
方向图	不对称	对称	不对称	对称	对称	对称
交叉极化电平	高	低	高	低	低	低

每一个表格应有简短确切的题名，连同表序号置于表的正上方。表名称、表中的内容居中排列，字号为五号，中文字体为宋体，英文字体为 Times New Roman，行距一般与正文保持一致。表格线统一用单线条，磅值为 0.5 磅。

表格与正文之间一般应空一行。

计量单位：学位论文中出现的计量单位一律采用国务院 1984 年 2 月 27 日发布的《中华人民共和国法定计量单位》标准。

第五章　绪论

5.1　研究背景及意义

自从我国 80 年代改革开放起，工业发展的速度就非常的迅速且稳定。回顾近代工业史，我国在制造业方面的成就尤为亮眼，无论是工业化的科技理论水平还是实际成果产业都在世界舞台上有着令人瞩目的表现。从官方数据总结中我们可以清晰的发现，在 2017 年我国 GDP 的构成中，有近乎三分之一的数据量来源自工业经济。从全球化的角度去对比，通过当前世界工业标准分类我们可以清晰的发现，我国在 22 个总体大类中的煤炭，生铁等等七个传统大类中稳居制造生产的榜首，甚至在某些类别中具有压倒性地位；同时针对于互联网时代下的新工业制造，我国也依旧具有很先进的科技手段和成熟的发展模式，例如无论是工业机器人还是当今有着非常广泛应用的新能源汽车，我国都具有极高的市场占有率，并呈现出非常强劲的长期竞争力。但是这并不代表着我国的工业发展是完美的，没有任何问题的。我国工业前期的迅猛发展离不开人口红利和优势，仔细分析还会发现整体工业发展不平衡，虽然科技理论非常先进，但是很多传统的工业流水线并不能利用被他们认为成“空中楼阁”的相关科学理论，而是滞后的一直利用劳动力的廉价和数量的巨大进行重复的作业。这一现状随着整体人口结构的转型可预见的将会暴露越来越多的问题[?]。

传统的工业制造不再能满足 21 世纪以来新的产品需求：我们需要更加短的生产周期和更加复杂的产品性质，同时也不能忽略更加灵活的产品更新需求。在这一现状的催生下，一个非常有创造性的理论被提出并逐渐完善应用。那就是柔性制造系统 (Flexible Manufacturing System, FMS)。这一系统着眼于现代工业对于多样和更新的需求，具有优秀的柔性化和智能化性质。典型的 FMS 需要几个基本模块：数控机床，物料传递系统，计算机总控系统。由于其集成性和网络化的特点，它为企业提供了更加高效、灵活的制造方式, 从而提高了企业的竞争力。较为典型的是它在生产时针对共享资源的处理，在实际工业生产中为相关的产线提供了极大的效率提升；在无论是针对产品更新度还是多样化，柔性制造系统都为现代工业交出了一份令人惊艳的答卷[?][?]。而在柔性制造系统中，有一个模块发挥着非常关键的作用，这一模块就是生产调度。这一模块的质量和效率直接关系到整体生产的质量和效率，无论从经济角度还是社会层面都具有非常战略性的意义。但是这一问题经常是难以找到非常好的方式进行解决的：因为极其大量的实际产线调度问题都是多项式复杂程度的非确定性问题，在学术界我们称之为 NP(Non-deterministicPolynomial) 完全问题[?][?]，这一特点使得想要找到一个明确固定的规律是完全不可能的，从而导致了整体制造的调

度非常的复杂。不止如此，随着工业发展，对于某些特定的产品，整体产线需要很多约束条件，例如精确的时间要求，极端的生产环境等。这一实际情况无疑又加大了生产调度问题寻求更优方法的难度。综上我们不难发现，学术界和企业界都对调度问题有着非常大的关注度。

在新工业中，电子工业无疑是非常重要的一个方向。由于大多数电子组件的化学成分都为硅或部分含硅化合物，我们习惯性的直接将电子产业等同于半导体产业。半导体产业与我们现代的生活息息相关，其中集成电路技术几乎出现在先进可以看到的所有电子产品中。从大家每天使用的手机到机构企业需要的大型计算机，都不能离开集成电路，这也代表着半导体产业的强应用性。而在整体半导体产业中，对半导体的加工无疑是最具有经济效益和社会意义的产业。具体对其进行分类，当今我国应用的主要技术包括晶圆制造加工，薄膜沉积、光刻、蚀刻、掺杂等等。其中，晶圆制造这一步骤主要通过化学手段将硅精纯并制作成硅片达成；而晶圆加工往往需要更多的工业控制生产步骤加入，通过包括融化、晶体成长、裁切检测、切片清洗等等复杂而精确的加工步骤产生出最终的芯片。

想要解决晶圆制造问题，第一步就需要对一个现实复杂的模型进行精确又高效的建模。只有数学模型贴合性好，我们才能正确的进行算法的模拟和运算，从而适应不同的加工需求和给出更优秀的调度结果。一般在学术研究中，我们常用 Petri 网、自动机等对柔性制造系统进行建模分析；自动机无法很好的表述出系统并发关系，使针对晶圆制造这一典型的复杂离散系统，我们选择 petri 网进行建模[?][?][?][?]。

5.2 国内外研究现状

1962 年，CarlAdam Petri 在他的博士论文《与自动机通信》中首次提出了 Petri 网的概念。后来，该模型被命名为 Petri 网，逐渐成为了理论计算机科学中的一个很有创造性的方向。由于当时自动机理论中缺乏重要的并发概念，不适合描述和研究狭义相对论、测不准原理等现代物理学中的典型问题，而 Petri 网模型能以自然、直观、易懂的方式分析并行系统的各种状态行为，这一非常优异的理论特色使得相关学术界中很多研究人员产生了浓厚的兴趣[?][?]。

1975 年 7 月，麻省理工学院举办了第一届 Petri 网及相关理论研讨会。此后，国际上每年都会定期举办有关 Petri 网相关理论的研讨会，关于 Petri 网相关理论及其应用的研究成果不断涌现。经过多年的深入研究，Petri 网理论在垂直和水平两个方向上都得到了很大的发展和完善。垂直发展体现在 Petri 网的建模理论上，从最基本的条件/事件网 (C/E)、位置/过渡网 (P/T) 逐渐发展到谓词/过渡网和彩色网等高级网。横向发展体现在 Petri 网的时间属性上，从非参数网发展到时间 Petri 网和随机 Petri 网。目前，Petri 网理论已广泛应用于柔性制造系统、离散事件系统、故障诊断系统、工作

流建模与管理等多个领域[?][?][?]。

国内对于 petri 网的研究也在近 30 年中不断进步和完善。1988 年，南京航空航天大学的陈浩发表了第一篇关于 Petri 网的硕士论文，而随着国内学术界的不断耕耘和发展，相关的论文和成果达到近 5000 篇。目前，仍有大量研究人员在该领域不断进行更深入的探索，相信未来会发现更多有用的研究成果。

5.3　论文结构

本文主要是研究基于 Petri 网和蚁群算法的柔性制造系统的调度问题，基于库所时间网与变迁时间网提出一种新的时间网子类，并使用此时间网子类对实际制造系统进行建模，最后使用蚁群算法求解模型调度策略，并结合模型实际情况，对蚁群算法设计了 6 种优化方案。本论文包含五章，各章节主要内容如下：

第一章概括性地阐述了本文研究的背景与意义，简述了 Petri 网、各调度算法在国内外的现阶段的研究情况，然后总结了一系列求解调度问题的方法，最后给出了本文的组织结构。

第二章是本文的基础知识部分，这部分主要是详细介绍了 Petri 网的基本理论，包过 Petri 网的基本概念、定义和特性等，然后介绍了 Petri 网在晶圆制造领域的一些应用并且介绍了基于 Petri 网的 FMS 建模相关理论与方法，通过一个简单的例子，简述了 Petri 网建模的过程。最后介绍了一系列时间网。

第三章在第二章的基础上，将变迁时间网与库所时间网进行结合，提出变迁库所时间网这种新的时间网子类。并基于变迁库所时间网设计了一种供调度算法使用的变迁发射流程，并设计了一种用于计算各种 Petri 网模型调度策略的程序架构。并通过一个简单的例子，描述了此发射流程的全过程。在此基础上，使用变迁库所时间网对一个实际的晶圆制造系统进行建模。

第四章设计并使用蚁群算法对第四章建立的变迁库所时间网模型求解调度策略。并对求解出的调度策略进行分析，对蚁群算法提出了一系列优化方案。最后分别对这些优化方案进行实验分析。

最后一章为总结与展望，对本文所研究的内容进行了回顾，并提出了研究存在的不足以及接下来需要进一步研究的地方。

第六章　Petri 网基本理论

本章为本文的理论研究的基础，主要介绍了 Petri 网概念与特性，包过：Petri 网定义、Petri 网模型结构、Petri 网可达图和 Petri 网特性；介绍了 Petri 网在不同领域的应用；介绍了一系列 Petri 网的调度方法；最后介绍了一系列时间网。为后续章节提供理论支持。

6.1　Petri 网概念与特性

Petri 网是一种描述离散事件系统的数学模型，它既可以用数学定义形式化地表示，也可以用图形形象化地表示。使用 Petri 网建立的模型具有简单、易懂、可操作性强等特点，能够很直观地表述出各类实际系统模型中并发、依赖、冲突、死锁等情况。Petri 网既可用于结构分析又可用于行为分析，因此在各个领域均有广泛应用[?][?][?]。

6.1.1　Petri 网定义

一个 Petri 网中通常是由两种节点组成的图，这两种节点分别为：库所 (Place)，变迁 (Transition)，每个元素含义与功能各不相同：库所中存放托肯 (Token)，托肯表示系统的资源；变迁一般代表能够改变系统状态的某种事件，变迁执行一种称为发射的操作，变迁发射后会改变库所中的资源的状态，从而改变系统的状态；变迁和库所间通过有向弧 (Arc) 连接，如果有向弧从库所指向变迁，意味着变迁发射后会从库所中取走一定数目的托肯，如果有向弧从变迁指向库所，意味着变迁发射后会放入一定数目的托肯进库所；Petri 网也可以图形表示，在图形上分别用矩形（□）表示变迁、圆圈（◯）表示库所，某些变迁和托肯间会通过箭头连接，箭头 (←) 表示有向弧。

定义 2.1[?]：Petri 网可以用一个四元组进行数学表示，四元组的元素分别为：库所集合、变迁集合、有向弧集合、有向弧上到权值的映射关系，即 $N=(P,T,F,W)$，$P=\{p_0,p_1,p_2,...,p_i\}$，$i\in\mathbb{N}^+$，是一个限非空的库所集合；$T=\{t_0,t_1,t_2,...,t_j\}$，$j\in\mathbb{N}^+$，也是一个限非空的变迁集合；这两个集合不存在相交部分，因为 Petri 网中的一个元素不能即是库所又是变迁，即 $P\neq\emptyset, T\neq\emptyset, P\cap T=\emptyset$。库所和变迁间通过有向弧连接，即 $F\subseteq(P\times T)\cup(T\times P)$。弧上带有权值，即 i $W:(P\times T)\cup(T\times P)\to\mathbb{N}$，对于任意 $w\in W$，$w=1$，如果存在某个权值 w 不为 1，则称这个网为一般网。i **例 2.1**　如图 2.1 就是某 Petri 网的图形表示，他的数学表示为：$N=(P,T,F,W)$，$P=\{p_1,p_2,p_3,p_4,p_5,p_6\}$ 为此 Petri 网库所的集合，此 Petri 网有 6 个库所；$T=\{t_1,t_2,t_{j3},t_4\}$ 为此 Petri 网变迁

的集合，此 Petri 网有 4 个变迁；

$$F = \{(p_1, t_2), (t_1, p_1), (p_2, t_3), (t_2, p_2), (p_3, t_4),$$
$$(t_3, p_3), (p_4, t_1), (t_2, p_4), (p_5, t_3), (t_4, p_5), (p_6, t_1), (t_4, p_6)\}$$

为此 Petri 网有向弧集合，有向弧的权值为：$W(p_1, t_2) = W(t_1, p_1) = W(p_2, t_3) = W(t_2, p_2) = W(p_3, t_4) = W(t_3, p_3) = W(p_4, t_1) = W(t_2, p_4) = W(p_5, t_3) = W(t_4, p_5) = 1$，$W(t_4, p_6) = W(p_6, t_1) = 2$。因为这个网中的有向弧上的权值并不全为 1，而是存在 2 个有向弧上权值为 2，所以是一个一般网。

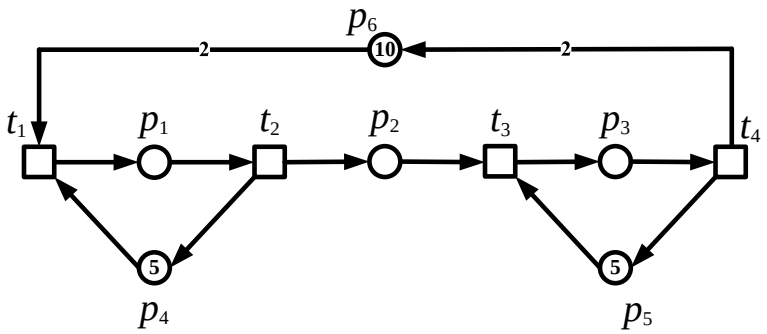


图 6.1　一个一般 Petri 网模型

定义 2.2[?]：对于某 Petri 网 $N = (P, T, F, W)$，在此网中的任意一个节点 $m \in P \cup T$，分别在其前后打点，表示此节点的前置节点集合和后置节点集合，记 ${}^\bullet m$ 为节点 m 的前置集合，${}^\bullet m = \{n \in P \cup T | (n, m) \in F\}$；记 $m^\bullet$ 为节点 m 的后置集合，$m^\bullet = \{n \in P \cup T | (m, n) \in F\}$，如果节点 $m \in P$，那么 ${}^\bullet m \subseteq T$，$m^\bullet \subseteq T$，如果节点 $m \in T$，那么 ${}^\bullet m \subseteq P$，$m^\bullet \subseteq P$，相应地，将全部节点的集合记作 $M \subseteq (P \cup T)$，则节点集合满足 ${}^\bullet M = \cup_{m \in M} {}^\bullet m$，$M^\bullet = \cup_{m \in M} m^\bullet$。若满足 $\forall i \in \mathbb{N}^+ = \{1, 2, ..., n-1\}$，$y_{i+1} \in y_i^\bullet$。此节点序列 $m_1, m_2, ..., m_i, ..., m_j$ 即为 Petri 网 N 的路径，其中 $m_i \in P \cup T$。如果在这条路径中除了 m_1 和 m_j 以外，所有的节点都不相同，并且 $m_1 = m_j$，即此节点序列的首尾相同，遍历此序列最终会回到原点，因此此序列构成一条回路，此 Petri 网被称为一条简单回路。

在定义 2.2 中，如果 m 表示库所，此库所的前置集 ${}^\bullet m$ 中的变迁称为输入变迁，此库所的后置集 $m^\bullet$ 中的变迁称为输出变迁集。如果 m 表示变迁，此变迁的前置集 ${}^\bullet m$ 中的库所称为输入库所，此变迁的后置集 ${}^\bullet m$ 中的库所称为输出库所。

例 2.2　如图 2.2 所示，是一个简单的 Petri 网模型，此网 P 中所有库所 $(p_1, p_2, p_3, p_4, p_5)$ 的前置变迁集合与后置变迁集合为：${}^\bullet p_1 = \{t_1\}$，$p_1{}^\bullet = \{t_2\}$，${}^\bullet p_2 = \{t_2\}$，$p_2{}^\bullet = \{t_3\}$，${}^\bullet p_3 = \{t_3\}$，$p_3{}^\bullet = \{t_4\}$，${}^\bullet p_4 = \{t_2, t_4\}$，$p_4{}^\bullet = \{t_1, t_3\}$，${}^\bullet p_5 = \{t_4\}$，$p_5{}^\bullet = \{t_1\}$。网模型中

所有变迁 (t_1, t_2, t_3, t_4) 的前置库所集合和后置库所集合为：${}^\bullet t_1 = \{p_4, p_5\}$，$t_1{}^\bullet = \{p_1\}$，${}^\bullet t_2 = \{p_1\}$，$t_2{}^\bullet = \{p_2, p_4\}$，${}^\bullet t_3 = \{p_2, p_4\}$，$t_3{}^\bullet = \{p_3\}$，${}^\bullet t_4 = \{p_3\}$，$t_4{}^\bullet = \{p_4, p_5\}$。对于一组节点序列：$p_1, t_2, p_2, t_3, p_3, t_4, p_4, t_1, p_1$ 就是此 Petri 网的一条路径，因为此路径的首尾相同，其他节点均不相同，因此这是一条简单回路。

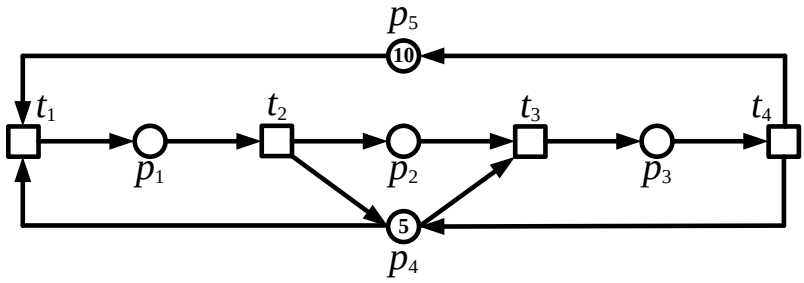


图 6.2　一个 Petri 网模型

定义 2.3[?]：一个 Petri 网 $N = (P, T, F, W)$，如果满足条件 $\forall x, y \in (P \cup T)$，$W(x, y) > 0$，$W(y, x) = 0$，则将此 Petri 网称为一个纯网 (pure net)。

要判断一个 Petri 网是否是纯网，可以使用关联矩阵，以及前置矩阵和后置矩阵。$[N]$ 是 $|P| \times |T|$ 的一个整数矩阵。$[N]$ 可以进一步进行拆分为 $[N]^+$ 和 $[N]^-$，对其进行矩阵运算得到 $[N]$，运算公式为 $[N] = [N]^+ - [N]^-$。其中 $\forall t \in T$，$\forall p \in P$，$[N]^+(p, t) = W(t, p)$，$[N]^-(p, t) = W(p, t)$。

例 2.3　图 2.2 所示的 Petri 网模型，所有的有向弧只有一个方向，因此此 Petri 网没有自环，是一个纯网。

对于图 2.2 中所示的 Petri 网中，将其库所变迁的关系分解为如下三个矩阵，其中 $[N]^+$ 为输入关联矩阵，$[N]^-$ 为输出关联矩阵，$[N]$ 为关联矩阵。

$$[N^+] = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix} \quad [N^-] = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad [N] = \begin{pmatrix} -1 & 1 & 0 & 0 \\ 0 & -1 & 1 & 0 \\ 0 & 0 & -1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 0 & 0 & -1 \end{pmatrix} \tag{6-1}$$

在上述所示的关联矩阵中，$[N^+]_{(p_1, t_2)} = 1$，意味这如果此 Petri 网发射 t_2 变迁，会往库所 p_1 中放入 1 个托肯，$[N^-]_{(p_1, t_2)} = 0$，意味着发射 t_2 变迁，不会往 p_1 取走托肯，前两个关联矩阵复合后得到的 $[N](p_1, t_2) = 1$ 意味着如果此 Petri 网发射 t_2 变迁，库所 p_1 中的托肯数量会加 1。

定义 2.4[?]：Petri 网 $N = (P, T, F, W)$ 中任意标识 M 都是一个维数与库所数相同的自然数向量。(P, T, F, W, M_0) 也可以直接记为 (N, M_0)，N 表示 Petri 网的结构，M_0 称为网系统的初始标识，表示 Petri 网的初始状态。

例 2.4 图 2.2 所示的 Petri 网，此 Petri 网的初始标识向量为：$M_0 = (0,0,0,5,10)^T$，也可以使用与库所有关的多项式表示，即当表示向量对应库所的数值作为系数，乘上对应库所，再求和组成的多项式对于初始标识 M_0，此标识向量为：$M_0 = 5p_4 + 10p_5$。

定义 2.5[?]：对于某 Petri 网，如果在当前标识 M 下，如果某个变迁 t 满足 $\forall p \in {}^\bullet t, M(p) \geq W(p,t)$，则称变迁 t 在标识 M 下是使能的，并将其记作 $M[t\rangle$。一般来说，Petri 网库所是有容量限制的，如果变迁 t 在标识 M 下是使能的，并且变迁 t 发射后，不会超过库所容量的限制，即满足 $\forall p \in t^\bullet, M(p) + W(t,p) \leq MAX(p)$，则此变迁能被安全发射，其中 $MAX(p)$ 表示库所 p 中能够存放的最多托肯的数量，也就是库所容量。变迁 t 发射后会改变 Petri 网模型中各库所中托肯的状态，因此 Petri 网的状态也就被改变了，从而产生一个新的 Petri 网的标识，在可达图中新标识是原标识的后继节点，两标识通过变迁连接，新标识的计算公式为 $M'(p) = M(p) - W(p,t) + W(t,p)$，原标识与新标识的关系记为 $M[t\rangle M'$，表示可达状态 M 可以通过发射变迁 t 到达可达状态 M'。

例 2.5 在图 2.2 所示的 Petri 网模型中，初始标识为 $M_0 = 5p_4 + 10p_5$，表示库所 p_4 中有 5 个托肯，库所 p_5 中有 10 个托肯，在此标识下变迁 t_1 是可以安全发射的，记作 $M_0[t_1\rangle$，表示变迁 t_1 在状态 M_0 下发射后既不会使库所中的托肯数为负数，也不会超过库所容量。变迁 t_1 的前置库所集合为 ${}^\bullet t_1 = \{p_4, p_5\}$，分别判定 p_4，p_5 两个库所是否能安全地取走托肯：$M_0(p_4) = 5 > W(p_4,t_1) = 1$，$M_0(p_5) = 10 > W(p_5,t_1) = 1$，这两个库所中都有足够多的托肯数发射变迁 t_1，所以变迁 t_1 能够发射，在发射后，库所 p_1 的托肯数更新为 $M_1(p_1) = M_0(p_1) - W(p_1,t_1) + W(t_1,p_1) = 1$，库所 p_4 的托肯数更新为 $M_1(p_4) = M_0(p_4) - W(p_4,t_1) + W(t_1,p_4) = 4$, 库所 p_5 的托肯数更新为 $M_1(p_5) = M_0(p_5) - W(p_5,t_1) + W(t_1,p_5) = 9$。此 Petri 网的标识从 $M_0 = 5p_4 + 10p_5$ 更新为 $M_1 = p_1 + 4p_4 + 9p_5$，M_0 可以通过发射 t_1 到达 M_1，记作 $M_0[t_1\rangle M_1$。

定义 2.6[?]：如果对于 Petri 网 N，初始标识 M 发射某变迁后能够到达新标识 M'，则称 M 到 M' 是可达的，记为 $M[\delta\rangle M_n$，如果 $M[t_1\rangle M_1[t_2\rangle M_2 \ldots M_{n-1}[t_n\rangle M_n$，意味着 t_1，t_2 可以被连续发射，则变迁序列 $\delta = t_1 t_2 \ldots t_{n-1} t_n$ 为网中的一个可发射的序列，$M_1, M_2, \ldots, M_{n-1}, M_n$ 为 Petri 网 N 的可达标识。因为这两个变迁均可发射，所以可以使用下述状态方程直接计算发射这两个变迁后的标识，$M_n = M + [N]\overrightarrow{\delta}$，其中将 $\overrightarrow{\delta}: T \to \mathbb{N}$ 称作是计数型整数向量，在发射序列 δ 中变迁 t 发射的次数和可以用 $\overrightarrow{\delta}(t)$ 来表示。

例 2.6 如图 2.2 所示，在此 Petri 网 N 中发射一条变迁序列 $\delta = t_1 t_2 t_3 t_4$，其中 $\overrightarrow{\delta} = [1 \quad 1 \quad 1 \quad 1]^T$ 称为 Petri 网 N 的发射向量。通过发射变迁序列 δ 产生了新的可达标识 M'，此过程可以使用下述状态方程表示：

$$M_n = M_0 + [N]\overrightarrow{\delta} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 5 \\ 10 \end{pmatrix} + \begin{pmatrix} -1 & 1 & 0 & 0 \\ 0 & -1 & 1 & 0 \\ 0 & 0 & -1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 0 & 0 & -1 \end{pmatrix} \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 5 \\ 10 \end{pmatrix} = M_0 \tag{6-2}$$

此变迁序列发射完后，又回到了初始标识 M_0，这是因为这个发射序列正好是此 Petri 网的一条完整的回路，并且所有的弧上权值均为 1。

定义 2.7[?]：对于一个 Petri 网 N，从某标识 M 出发，能够通过变迁序列到达的所有标识的集合记为 $M[\rangle$。如果是从初始标识 M_0 出发，其可达标识集为 $M_0[\rangle$，也可以记为 $R(N, M_0)$。在标识 M_0 下，如果 $\exists M' \in R(N, M)$ 使得 $M'[t\rangle$，当且仅当对 $\forall M \in R(N, M_0)$ 成立；对于变迁集中的所有变迁，即 $\forall t \in T$，则变迁 t 是活的，如果存在某变迁在初始标识 M_0 下是活的，则网 (N, M_0) 是活的 (live)；对于初始标识下所有的可达标识 $\forall M \in R(N, M_0)$，如果 $\nexists t \in T$，$M'[t\rangle$，则此 Petri 网 (N, M_0) 是无死锁的 (deadlock-free)。

例 2.7　判定 2.3 所示的三个 Petri 网的活性。

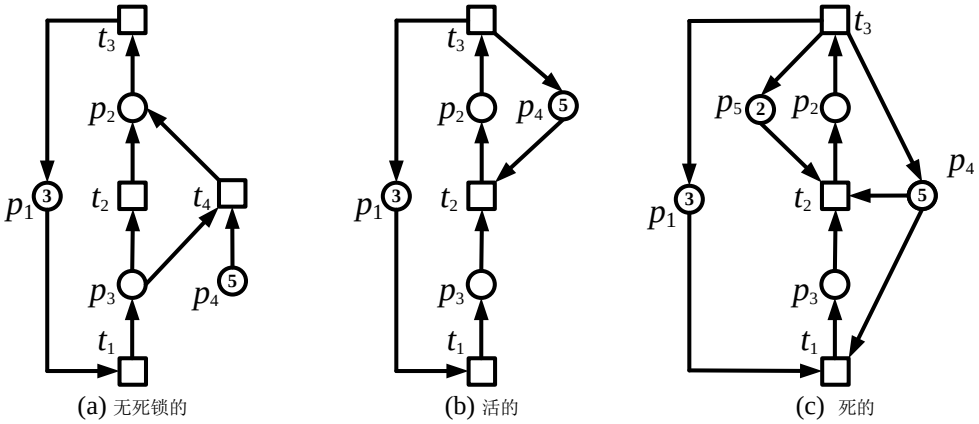


图 6.3　一个 Petri 网活性示例

图 2.3(a) 中的 Petri 网没有死锁标识，变迁变迁 t_4 不是活的，因此称此 Petri 网是无死锁的；图 2.3(b) 中的 Petri 网所有变迁都是活的，因此称此 Petri 网活的；图 2.3(c) 中的 Petri 网存在死锁标识，因此称此 Petri 网是死的。

6.1.2　Petri 网模型结构

使用 Petri 网建模，可以很直观地描述出模型结构[?]，本小节将对 Petri 网中的四种结构模型：顺序、并发、冲突、混淆进行详细介绍。

（1）顺序关系

在某标识下，如果某变迁发射，可以时原本不能使能的变迁使能，也就是说变迁可以顺序发射下去，则称这两个变迁在这个标识下是顺序关系，如图 2.4（a）所示。

定义 2.8[?]：对于标识 M，$\exists$ 变迁 t_1、t_2，$M[t_1\rangle M'$，$\neg M[t_2\rangle$,$M'[t_2\rangle$，变迁 t_1、t_2 在标识 M 下为顺序关系。

（2）并发关系

在某标识下，如果两个变迁都可以使能，并且其中任何一个变迁发射都不会使另一个变迁不使能，就称这两个变迁为并发关系。并发关系下的两个变迁，是可以独立发射的，如图 2.4（b）所示。

定义 2.9[?]：对于标识 M，$\exists$ 变迁 t_1、t_2，$M[t_1\rangle$，$M[t_2\rangle$，并且 $M[t_1\rangle M_1 \Rightarrow M_1[t_2\rangle$，$M[t_2\rangle M_2 \Rightarrow M_2[t_1\rangle$，变迁 t_1、t_2 在标识 M 下为并发关系。

（3）冲突关系

与并发关系正好相反，如果在某标识下，两个使能变迁发射任何一个，都会使另一个无法使能，则称这两个变迁在这个标识下是冲突关系，如图 2.4（c）所示。

定义 2.10[?]：对于标识 M，$\exists$ 变迁 t_1、t_2，$M[t_1\rangle$，$M[t_2\rangle$，并且 $M[t_1\rangle M_1 \Rightarrow M_1[t_2\rangle$，$M[t_2\rangle M_2 \Rightarrow M_2[t_1\rangle$，变迁 t_1、t_2 在标识 M 下为并发关系。

（4）混淆关系

如果一个 Petri 网中同时存在并发和冲突两种关系的变迁，并且并发关系的变迁发射后会时冲突关系消失。在这种情况下是无法判断出冲突关系是否出现过的，因此称变迁间的这个关系为混淆关系，如图 2.4（d）所示。

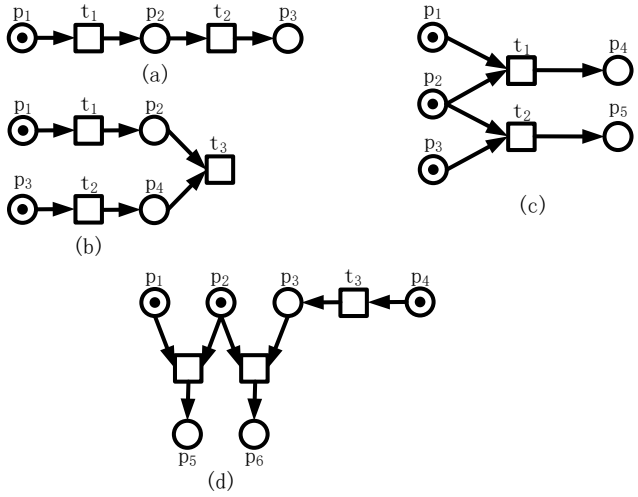


图 6.4　Petri 网模型结构

6.1.3　Petri 网可达图

本文所研究的调度算法需要在 Petri 网可达图中搜索一条路径，所以本节将要介绍 Petri 网可达图 (reachability graph)。可达图是分析 Petri 网的重要工具，而且不论是何种 Petri 网都有可达图。在可达图中可以清晰直观地看出不同标识是如何通过变迁转换的。给出 Petri 网以及此 Petri 网的初始标识，便可确定这个 Petri 网的可达图。利用 Petri 网可达图可以分析此 Petri 网所描述的离散事件系统的可达性、有界性、活性、安全性等一些列重要性质，是 Petri 网理论中十分有用的分析工具。

定义 2.11[?]：记一个 Petri 网 (N, M_0) 的可达图为 $RG(N, M_0) = (U, S)$，可达图中包含该网模型的所有可达标识以及可达标识之间的变迁关系，可达图是一个有向图。其中，可达图中用圆圈表示节点，用 $U = R(N, M_0)$ 代表网的全部可达标识；可达图中节点之间的有向连接弧用集合 $S = \{(M, t, M') \mid M' \in R(N, M_0)，M[t\rangle M'\}$ 来表示，弧上标注了对应的变迁，用来说明从一个可达标识到达另一个可达标识所需要发射的变迁，也就是可达标识之间的映射关系。

可达图算法与传统的图遍历算法并无本质上的区别。通过算法 2.1[?]可以得到一个 Petri 网的可达图：

算法 6.1 可达图算法

procedure RG
Require: 一个标记的 Petri 网 (N, M_0)。
Ensure: 网模型的可达图。
　　可达图 $RG(N, M_0)$ 的开始点为初始标识 M_0
　　一开始 Petri 网的全部可达标识没有被标记
　　while{有可达标识未被标记}**do**
　　对未被标记的可达标识 M
　　考虑可达标识 M 下每一个使能的变迁 t，通过 $M' = M + [N](\cdot, t)$ 计算新的可达标识 M'
　　if{可达图 RG 中还没有添加新的可达标识 M'} **then**
　　将新的可达标识 M' 添加到可达图中
　　在 M 到 M' 之间添加有向弧并标记为 t
　　end if
　　标记可达标识 M
　　end while
end procedure

对图 2.2 中的 Petri 网使用此算法求取可达图，为了减少可达标识数，将初始标识改为 $(0, 0, 0, 1, 2)$，将得到如下可达标识：

可达标识	使能变迁
$M_0:(0,0,0,1,2)$	t_1
$M_1:(1,0,0,0,1)$	t_2
$M_2:(0,0,1,0,1)$	t_4
$M_3:(0,2,0,1,0)$	t_3
$M_4:(0,1,1,0,0)$	t_4
$M_5:(0,1,0,1,1)$	t_1 t_3
$M_6:(1,1,0,0,0)$	t_2

表 6.1　图 2.2 中的 Petri 网可达图

6.1.4　Petri 网特性

Petri 网的特性包过可达性、有界性、活性、可逆性、可覆盖性和持续性[?]，可使用上一节提到的可达图分析 Petri 网的这些特性。本节将简略介绍上述特性的含义。

（1）可达性

可达性将是本文调度算法部分最为关注的特性。如果存在某条变迁序列 δ，使得初始标识 M_0 按此变迁序列发射，能够到达标识 M'，就说 M' 是从 M_0 可达的，记作 $M_0[\delta\rangle M'$。如果 δ 中只有一个变迁，则说 M' 是从 M_0 立即可达的。实际上调度算法所要做的就是找到一条最优的变迁序列。一个 Petri 网 N 的所有可达标识组成一个可达标识集，记作 $R(N,M_0)$ 或 $R(M_0)$。表 2.1 就是图 2.2 中的 Petri 网的可达标识集，这里面所有标识都是可达的。

（2）有界性

有界指的是 Petri 网的可达标识数是有限的，也可以描述成 Petri 网从初始标识 M_0 开始任何一个可达标识中任意一个库所的托肯数都有界。实际的物理模型中许多是有界的，一般情况下库所指的是存放资源的场所，是有容量的。实际建模时一般会直接给出库所容量的约束。表 2.1 中只有 7 个可达标识，因此图 2.2 中的 Petri 网是有界的。

（3）活性

Petri 网的活性和 Petri 网中是否有死锁有相关性。如果可达图的任何子图中都存在一个标识能使变迁 t 使能，则说明变迁 t 是活的，如果所有变迁都是活的，则称 Petri 网是活的。如果 Petri 网是活的，则它一定没有死锁标识。图 2.2 中的 Petri 网就是活的。

（4）可逆性

可逆性也和 Petri 网中是否有死锁有相关性。如果可达图中任意一个可达标识都可以通过发射某条变迁序列回到初始标识，则称 Petri 是可逆的。因此如果 Petri 网是可逆的，则它一定没有死锁标识。图 2.2 中的 Petri 网就是可逆的。

（5）可覆盖性

对于标识 M，如果 Petri 网可达图中存在标识 M'，M' 中每个库所的托肯数都比 M 的大，则说明 M 是可覆盖的。

（6）持续性

具有持续性的 Petri 网，一旦某变迁使能了，那它会一直使能，直到发射此变迁为止。图 2.2 中的 Petri 网的 t_2、t_3、t_4 变迁都是可持续的。

6.2　Petri 网的应用

在解决柔性制造系统生成调度问题时，Petri 网能够充分反应实际模型的各种特性，因此 Petri 网在此方面应用十分广泛。Petri 网在柔性制造系统中的各种应用如图 2.5 所示。

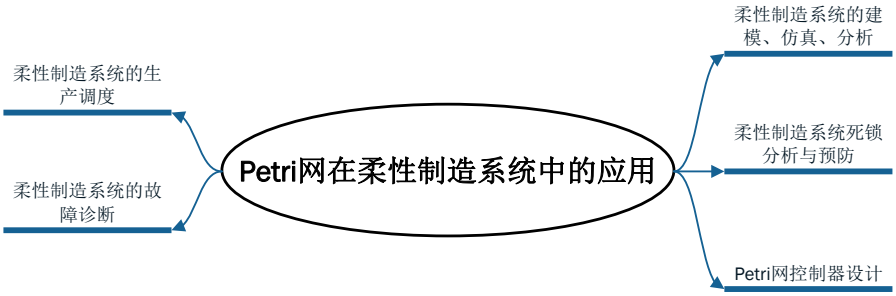


图 6.5　Petri 网在柔性制造系统中的各种应用

6.3　Petri 网的调度方法

本文将着重研究基于蚁群算法的 Petri 网调度问题。这是一个群体智能的图搜索算法，在 Petri 网背景下，搜索的空间是可达图。因此本节将对各种图搜索算法进行介绍。本节介绍的算法分为两个大类：传统的图搜索算法、群体智能算法。

6.3.1　传统的图搜索算法

传统的图搜索算法具有很相似的结构。本类算法有两个集合，一个集合用于存放待扩展的节点，另一个集合用于存放已经扩展过的节点。每次从待扩展的节点集合中取出节点，基于此扩展新节点，如果新节点不在存放已经扩展过的节点集合中存在，则说明它有扩展的价值，将其放入待扩展的节点集合中。不断重复上述操作，直到找到终点，或者待扩展的节点集合为空。

算法 6.2 图搜索算法

```
procedure GRAPHSEARCH
    将起点存入待扩展的节点集合
    while 待扩展的节点集合不为空 do
        curr ← 从待扩展的节点集合取出节点
        while curr 未完成扩展 do
            next ← 扩展一个 curr 的后继节点
            if next 不在存放已经扩展过的节点集合中存在 then
                将 next 放入待扩展的节点集合中
            end if
        end while
        将 curr 放入已经扩展过的节点集合中
    end while
end procedure
```

此后的一系列算法都是对从待扩展的节点集合取出节点这一步进行更精细的设计，从而改变图搜索的顺序。

（1）深度优先搜索

深度优先搜索算法如果不发生回溯，每次扩展的节点都是相邻的，这意味着当前节点完成扩展后必须从它扩展出的新节点选择下一次要扩展的节点。因此因该选择先入后出的堆栈来实现待扩展的节点集合[?][?][?][?][?]。

（2）基于 Petri 网的启发式调度

此算法是深度优先搜索算法的一个子类，通过设计对当前节点更为精细的扩展顺序来实现。比如每次都以单步最优的策略进行扩展，则可将当前节点的后继节点按距离排序后放入待扩展的节点集合中。体现在 Petri 网中，即为每次都发生最短完工的变迁，此策略称为最早完工策略[?][?][?][?][?]。

（3）广度优先搜索

广度优先搜索会一层层地扩展搜索树，因此需要按顺序依次扩展当前节点的后继节点。因此待扩展的节点集合应选取先进后出的队列进行实现[?][?][?][?][?]。

（4）迪杰斯特拉算法

迪杰斯特拉算法与广度优先搜索算法类似，在大方向上也是一层层地扩展可达树。不同的是，迪杰斯特拉算法每次扩展的节点都是待扩展的节点集合中离原点距离最短的。因此迪杰斯特拉算法一定能求出全局的最短路径[?][?][?][?][?]。

（5）启发式搜索算法

此算法又称 A 星算法，是对迪杰斯特拉算法的进一步优化。此算法的待扩展的节点集合会对每一个进入集合的节点进行估计，估计的是经过此节点的最短路径的

长度。此长度值 f 分为两部分：起点到当前点的最短路径长度 g、当前节点到终点的最短路径长度 h，

$$f = g + h$$

如果估计合理，g 的值是确定的，因为每次扩展都会往最优解上走，g 的值就是当前路径当前节点离原点的距离。因此 h 的值将是影响算法的关键。

因为 Petri 网的启发式函数需要结合具体情况进行设计，而本文研究的蚁群算法是一种通用的图搜索算法，所以本文在算法章节并不对 A 星算法进行测试。而深度优先搜索和广度优先搜索策略太过简单，效果是不如其他更为精细的搜索机制的，因此本文将主要使用 Petri 网的启发式调度、迪杰斯特拉算法为传统搜索算法的代表进行测试[?][?][?][?][?]。

6.3.2　群体智能算法

群体智能算法之间流程差异巨大，但本质思想是类似的。群体智能算法会并发地开启多条搜索路径，并提供某种正反馈机制把解往优的解上引导。

（1）遗传算法

遗传算法大体流程为：

1、将问题的解编码为染色体；

2、对一系列染色体评价其适应度；

3、按适应度大小淘汰一批染色体；

4、从幸存的染色体中使用交叉操作生成新的染色体；

5、重复进行上述操作一定轮次后对适应度最高的染色体进行解码，得到并输出解[?][?][?][?][?]。

算法 6.3 遗传算法

```
procedure GA
    随机生成一批染色体
    while 未到达迭代轮数或解未收敛 do
        对一系列染色体评价其适应度
        按适应度大小淘汰一批染色体
        从幸存的染色体中使用交叉操作生成新的染色体
    end while
    对适应度最高的染色体进行解码，得到并输出解
end procedure
```

本文研究的蚁群算法无需考虑 Petri 网的具体结构，因此基因算法也应具备通用性。所以本基因算法训练的染色体为变迁的优先级，再使用传统图搜索算法按此优先级进行搜索求解。具体流程会在算法章节详细描述。

（2）蚁群算法

蚁群算法与基因算法不同，它无需设计染色体编码方式，可直接应用于图搜索。蚁群算法的大体流程为：

1、各蚂蚁根据图中信息素浓度并发搜索；

2、所有蚂蚁搜索完成后按规则在图上更新信息素；

3、重复进行上述操作一定轮次后输出探索到的最优解[?][?][?][?][?]。

算法 6.4 蚁群算法

```
procedure AntClonyOptimization
    while 未到达迭代轮数或解未收敛 do
        各蚂蚁根据图中信息素浓度并发搜索
        所有蚂蚁搜索完成后按规则在图上更新信息素
    end while
    输出探索到的最优解
end procedure
```

蚂蚁会在更优的路径上添加更多的信息素，并且有更大的概率走上信息素浓度高的路径。在此正反馈机制下，更优路径的信息素浓度会随算法运行逐步升高。

传统蚁群算法在求解 Petri 网调度问题时，其性能会受网结构影响，本文对此提出了一系列优化思路。

6.4 一系列时间网

实际制造系统中，时间也是重要的因素。如机械臂的运动、加工腔加工均需要消耗不同的时间，产率是衡量调度策略优劣的重要指标，要求取产率，则需要知道系统完成特定任务时的时间，为了实现上述功能，将时间因素加入 Petri 网中。

Petri 网由库所、变迁以及连接库所变迁的有向弧构成，库所中存有托肯。常规的为 Petri 网添加时间因素的方式便是在以上四种组成部分上添加时间限制，时间限制会影响到 Petri 网的使能逻辑。时间限制添加在 Petri 网不同的组成部分上便形成了四种不同的时间网，分别为：时间变迁网、时间库所网、时间弧网、时间托肯网。

6.4.1 时间变迁网

将时间限制以区间的形式添加到普通 Petri 网的变迁上，便形成了时间变迁网 (Time Transition Petri Net，TTPN)。每个时间区间由两个值组成，分别为区间的左右端点。区间的左端点，称为静态最早发射时间 (Static Earliest Firing Time，Static EFT)，区间的右端点为静态最晚发射时间 (Static Latest Firing Time，Static LFT)。当 Petri 网的某个变迁使能之后，它最少需要消耗静态最早发射时间，最多需要消耗静态最迟发

射时间才能完成发射。

定义 2.12[?]：$I=[a,b]$ 是一个时间区间 (time interval)，其中：

1. $a\in\mathbb{R}^+, b\in\mathbb{R}\cup\infty$

2. $a\leq b$

记 TI 为时间区间的集合，设 $I_1,I_2\in TI, I_1=[m,n], I_2=[p,q], c\in\mathbb{R}^+$，

则时间区间与时间区间的运算方式为：$I_1+I_2=[m+p,n+q]$,

时间区间与常数的运算方式为 $I_1+c=[m+c,n+c], I_1-c=[max(m-c,0),max(n-c,0)]$,

时间区间的最值运算为：$max(I_1,I_2)=[max(m,p),max(n,q)]$,

$min(I_1,I_2)=[min(m,p),min(n,q)]$。

表明时间区间经过一元和二元运算后，其结果仍为时间区间。

定义 2.13[?]：一个 TTPN 是一个六元组 $Z=(P,T,F,W,M_0,I)$，其中：

1. 五元组 (P,T,F,W,M_0) 是一个基本 Petri 网；

2. $I:T\rightarrow(\mathbb{Q}_0^+\cup 0)\times(\mathbb{Q}_0^+\cup\infty)$，并且 T 中的每个变迁 $I(t)=(I_1(t),I_2(t))$，$0\leq I_1(t)\leq I_2(t)$。

例 2.8　如图 2.6 所示，此 Petri 网即为 TTPN。在基本的 Petri 网中每个变迁上添加时间区间，可以得到如图的 TTPN。变迁 t_3 处的时间区间为 [3, 5] 则意味着 t_3 使能后，至少需要 3 个时间单位，才可能发射，但必须在 5 个时间单位之内进行发射。其他使能变迁上的时钟也在计时，因此逻辑与变迁 t_3 类似，如果变迁 t_1 也是使能变迁，则 t_1 必须在 1 个时间单位之前进行发射，但是此时还未到使能变迁 t_3 的最早发射时间，因此 t_1 必然先于 t_3 发射，综合来看 t_3 是不可能发射的。变迁的前置库所中的托肯被取走过，此变迁上的时钟才会被重置，否则会一直保持计时，因此当 t_1 发射后，t_3 的时钟也会计时，如果 t_1 发射后下一个标识中 t_3 依然能使能，其等待的时间应减去一部分，而不需要再重新等 3 个时间单位才能发射。

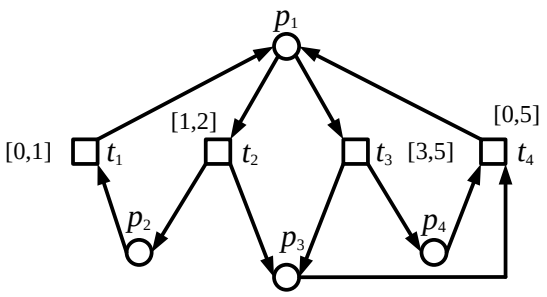


图 6.6　一个 TTPN 模型

Petri 网带有时间之后，原来的标识向量是无法表示此 Petri 网状态的所有信息。所

以 TTPN 的标识除了包含库所中的托肯数，还包含了当前 Petri 网上每个变迁上的时钟信息。因此 TTPN 的标识需要包含两个元素，第一个元素描述库所状态，称作 place-marking(简记作 p-marking)，第二个元素描述变迁状态，称为 transition-marking(简记作 t-marking)。p-marking 即为普通 Petri 网的标识。t-marking 表示了每个变迁上时钟的当前时间，如果此变迁不使能，则用符号 ν 表示。

定义 2.14[?]：记 P 为时间网 Z 所有库所的集合, 网 Z 的一个 p-marking 是一个从 P 到 $\mathbb{N}$ 的映射：$P \to \mathbb{N}$。显然，时间网 Z 中的 p-marking 也是原网中的标识。

定义 2.15[?]：记 T 为时间网 Z 所有变迁的集合，时间网 Z 中的任意一个映射：$T \to \mathbb{R}_0^+ \cup \{\nu\}$ 就是一个 t-marking。

定义 2.16[?]：记 $Z = (P, T, F, W, M_0, I)$ 为一个时间 Petri 网，m 为 Z 的一个 p-marking，h 为 Z 的一个 t-marking，Z 的一个状态为一个二元组 $z = (m, h)$ 且：

1. $\forall t((t \in T \wedge t^- \not\leq m) \to h(t) = \nu)$;

2. $\forall t((t \in T \wedge t^- \leq m) \to (h(t) \in \mathbb{R}_0^+ \wedge h(t) \leq LFT(t)))$;

状态 $z_0 = (m_0, h_0)$ 为时间网 Z 的初始状态，其中：

$$h_0(t) = \begin{cases} 0 & if \quad t^- \leq m_0 \\ \nu & if \quad t^- \not\leq m_0 \end{cases}.$$

以上是时间网的静态特性。正如普通 Petri 网的动态特性是由发射规则所决定的，时间网的状态也与发射规则有关，时间网的当前状态会因为当前的 p-marking 或者 t-marking 的变化而改变。普通 Petri 网的 p-marking 会随着变迁的发射而改变，在时间网中，变迁的发射通常不仅改变当前的 p-marking，也会改变 t-marking。除了变迁的发射，t-marking 也会随着时间的流逝而改变。

定义 2.17[?]：在时间网 $Z = (P, T, F, W, M_0, I)$ 中，变迁 t 在状态 $z = (m, h)$ 下准备好发射的条件是：

1. t 在原网的标识 m 下是使能的，且 $t^- \leq m$;

2. $h(t) \geq EFT(t)$。

定义 2.18[?]：记 $\hat{t}$ 和 $z = (m, h)$ 分别为时间网 $Z = (P, T, F, W, M_0, I)$ 的一个变迁和一个状态，变迁 $\hat{t}$ 能够在状态 z 下发射的条件是 $\hat{t}$ 已经满足准备好发射的条件，记作 $z \xrightarrow{\hat{t}}$。$\hat{t}$ 发射之后，网 Z 的状态从 z 改变到 $z' = (m', h')$，记作 $z \xrightarrow{\hat{t}} z'$，其中：

1. $m' = m + \Delta\hat{t}$;

2. $\forall t(t \in T \longrightarrow h'(t) = \begin{cases} \nu & if \quad t^- \not\leq m' \\ h(t) & if \quad t^- \leq m \wedge t^- \leq m' \wedge {}^\bullet t \cap {}^\bullet \hat{t} = \emptyset \wedge t \neq \hat{t} \\ 0 & otherwise \end{cases})$.

根据上述的定义，TTPN 的每个变迁上都有一个时钟。此变迁使能以后，它上面的时钟开始计时，达到此变迁的最早发射时间后，此变迁允许被发射，但时钟上的时间不允许超过最迟发射时间。一旦变迁被发射、变得不使能或变迁前置库所中的托肯被改变，此变迁的时钟会重新计时。

TTPN 是由普通 Petri 网上添加时间区间得到的，但普通 Petri 网也可看做一种 TTPN。即将普通 Petri 网所有的变迁的最早发射时间设为 0，最迟发射时间设为无穷大，即时间区间为 $[0,\infty]$。

6.4.2　时间库所网

时间库所网 (Time Pace Petri Net，TPPN) 即为在库所上添加时间限制的 Petri 网，每个进入库所的托肯，需要满足库所上的时间限制，才能被取走，这意味着每个托肯上都有一个时钟。

定义 2.19[?][?][?]：一个 TPPN 是一个六元组 $Z=(P,T,F,W,M_0,I)$，其中：

1. 五元组 (P,T,F,W,M_0) 是一个基本 Petri 网；

2. $I:P\rightarrow(\mathbb{Q}_0^+\cup 0)\times(\mathbb{Q}_0^+\cup\infty)$，$p_i\rightarrow I(p_i)=[a_i,b_i],0\leq a_i\leq b_i$。

与 TTPN 的标识类似 TPPN 中也需要涵盖系统的时间信息。TPPN 的时间区间加在库所上，但由于进入库所的托肯有先后顺序，因此同一个库所中的一系列托肯在时间上并不等价。要涵盖系统的完整信息，需要记录每一个托肯上的时钟。

TPPN 所有的托肯都是同步倒计时的，因此每进行一次发射，都要对所有的托肯重新计算其时钟。但是在本文第三章建立的实际晶圆制造系统的 Petri 网中，大部分库所是不存在时间约束的，为了提高之后算法效率，降低程序运行时间，本文的发射逻辑中会跳过无时间约束的库所中托肯的计时。

例 2.9　如图 2.7 所示，是一个 TPPN 模型。其时间区间均添加在库所上，表示进入此库所的托肯如果要被取出，需要满足的时间限制。库所 p_2 中有两个托肯，假设这两个托肯是刚刚一起被放入这个库所中的，即这两个托肯上的时钟的计时为 0。当变迁 t_1 发射后，会取走库所 p_2 中的一个托肯。这意味着 p_2 中至少有一个托肯的时钟计时超过了一个时间单位。TPPN 中所有的托肯是同步计时的，当变迁 t_1 发射后，取走库所 p_2 中的一个托肯，如果要继续发射变迁 t_1，是不需要消耗时间的，因为另一个托肯上的时钟也一并计时超过了一个时间单位

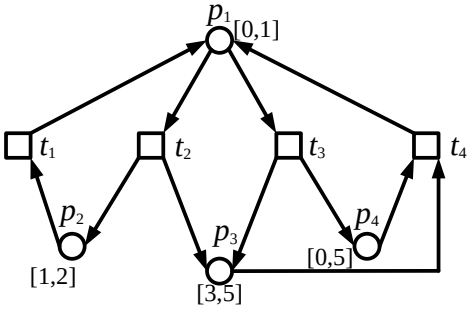


图 6.7　一个 TPPN 模型

TTPN 和 TPPN 的差异不仅仅是时间区间加在变迁或库所上这一点，他们时钟计时方式也有很大的区别。TTPN 时钟加在变迁上，当变迁使能时，时钟开始计时，此变迁的前置库所中的托肯被更新过就会重置计时。而 TPPN 的时钟加在托肯上，会一直计时下去，托肯被移动才会重置时钟。

基于以上的描述，TTPN 和 TPPN 的计时方式并没有实质上的冲突，可以兼容起来。只需要将 TTPN 的变迁计时前判断库所使能的逻辑扩展为 TPPN 的使能逻辑就行了。本文为解决实际问题按此思路将 TTPN 和 TPPN 结合起来，提出了一种新的时间网子类。

6.5 本章小结

本章介绍了 Petri 网概念与特性、Petri 网的应用、Petri 网的调度方法和一系列时间网。在接下来的章节中，将使用这些基础知识为实际系统进行建模，并使用调度算法对模型进行调度，并求解调度方案。

第七章　时间网与实际制造系统结合

7.1　变迁和库所上均有时延的时间 Petri 网

假设存在一个场景，机械臂需要从加工腔中取走一个工件放入另一个加工腔，工件在加工腔中加工需要时间，机械臂移动也需要时间。如果将加工腔看作库所，工件看作托肯，机械臂的动作看作变迁，则意味着托肯进入某库所后等待一段时间，之后变迁开始计时一段时间后发射，将此托肯移入另一个库所中。

因此需要一种新的时间 Petri 网模型，将 TTPN 和 TPPN 结合起来，本文提出了一种新的时间 Petri 网子类，时间变迁库所网 (Time Transition Place Petri Net，TTPPN)，解决了这种场景下的建模问题。

TTPPN 需要在变迁和库所上都添加时间区间。库所上的时间区间表示进入此库所的托肯，至少需要等待一段时间才能被取走，但不能驻留过长的时间。普通 Petri 网的使能逻辑中某个库所是否使能只需要判断此库所中的托肯数是否大于其尝试发射后置变迁输入弧上的权值就行了，也就是要保证库所中有足够的托肯能被取走。但 TTPPN 库所的使能还需要看托肯上的时钟，因为在 TPPN 中，托肯能否被取走，还需要判断托肯的停留时间是否满足了库所上的时间限制。因此 TTPPN 库所能否使能，需要看此库所中满足此库所时间限制的托肯的个数是否不低于其尝试发射后置变迁输入弧上的权值。

定义 3.1：一个 TTPPN 是一个七元组 $Z=(P,T,F,W,M_0,I_p,I_t)$，其中：

1. 五元组 (P,T,F,W,M_0) 是一个基本 Petri 网；

2. $I_p:P\to(\mathbb{Q}_0^+\cup 0)\times(\mathbb{Q}_0^+\cup\infty)$，$p_i\to I(p_i)=[a_i,b_i],0\le a_i\le b_i$；

3. $I_t:P\to(\mathbb{Q}_0^+\cup 0)\times(\mathbb{Q}_0^+\cup\infty)$，$t_i\to I(t_i)=[a_i,b_i],0\le a_i\le b_i$。

在 TTPN 中，变迁一旦使能，其上的时钟便会开始计时。在 TTPPN 中也是如此。当此变迁的所有前置库所都使能的那一瞬间，此变迁的时钟开始计时。

在实际的制造系统中，完成工序的耗时是一个衡量控制策略的重要指标。Petri 网的变迁表示的是此系统能执行的动作。为了让系统完成任务所执行的一系列的动作的耗时尽可能短，则要避免不必要的时间开销。这意味着，变迁的发射应该尽可能早，变迁上的时钟不允许有无意义的计时。

为了实现这个需求，发射逻辑应该分为 3 部分：变迁发射前的逻辑、变迁发射的逻辑、变迁发射后的逻辑。某变迁如果满足库所使能，并且需要被发射，须要按顺序执行完这 3 个逻辑。

算法 7.1 变迁发射逻辑

输入: 变迁库所时间网 $TTPPN$，变迁库所时间网的标识 $Marking$，准备发射的变迁 t

输出: 变迁 t 发射后到达的新标识 $next$

1: $next \leftarrow$ 将 $Marking$ 克隆一份
2: $beforeTlanuch(TTPPN, next, t)$ //变迁发射前的逻辑
3: $tlanuch(TTPPN, next, t)$ //变迁发射的逻辑
4: $afterTlanuch(TTPPN, next, t)$ //变迁发射后的逻辑
5: 返回 $next$

变迁发射前的逻辑要实现两个目标：1、计算出使此变迁使能的最短时间。2、从对应库所中删除被此变迁取走的托肯。变迁如果要使能，则其前置库所中需要有足够多完成计时的托肯。因此变迁使能的最短时间为此变迁所有前置库所中，所有被取走托肯中计时离时间限制最长的那段时间。如果将时钟改为倒计时，此时间应该为被取走的托肯中倒计时最长的那个时间。为了避免消耗不必要的时间，对于特定库所，取走的托肯应该是最先完成倒计时的那一批。

算法 7.2 变迁发射前的逻辑

procedure BEFORETLANUCH($TTPPN$,$next$,t)
  for all $p \in \bullet t$ **do**
    $needGetCount \leftarrow Pre(p, t)$
    $minTime \leftarrow$ 标识 $next$ 库所 p 中第 $needGetCount$ 的时钟计时
    $time \leftarrow max(time, minTime)$// $time$ 为使变迁 t 使能的最短时间
    for all $token \in TOKEN(p)$ **do** //$TOKEN(p)$ 为库所 p 中的托肯的集合
      if $TIME(token) \leq minTime$ **then** //$TIME(token)$ 为此 token 上时钟的计时
        从 $TOKEN(p)$ 中删除 $token$
      end if
    end for
  end for
end procedure

按上述流程变迁 t 取托肯时总是从托肯序列的头部取，之后的流程会在托肯序列的尾部放入托肯。而托肯在库所中停留会导致托肯时钟倒计时。因此托肯序列必然是升序的。

此处托肯序列选取何种数据结构实现会影响到算法效率。基于上述分析对托肯序列会有两种操作：从序列头部删除托肯、从序列尾部添加托肯。上述需求有两种实现方式：链表、循环数组。

如果使用链表，即需要定义节点结构。节点由两部分组成：托肯倒计时时钟、下一个节点的地址。这将带来以下两个缺陷。

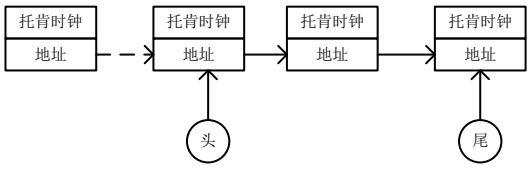


图 7.1　托肯序列的链表实现

为了将托肯连接成串，额外存储了大量地址信息。如果地址和托肯时钟选取相同的数据类型，那么整个数据结构只有一半的有效信息。

当托肯被移除后，此节点便失去引用成为内存垃圾。清理内存垃圾会带来时间开销。添加新的托肯需申请新的托肯节点，申请内存亦会带来时间开销。因此从时间和空间来看，链表这种数据结构实现托肯序列的功能并不高效。

本算法使用循环数组来实现此功能。预先申请固定长度的数组，并保存头尾两个指针。当删除托肯时头指针向前移动，如果已经移动到数组尾部，则从头开始。当添加托肯时，尾指针向前移动，如果已经移动到数组尾部，则从头开始。当尾指针追上头指针时，意味着数组已满。需重新申请更大的数组，并将数据移入完成扩容。

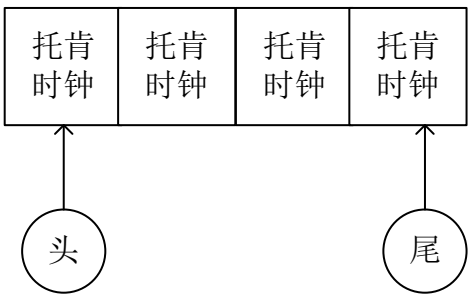


图 7.2　托肯序列的循环数组实现

循环数组中保留的信息只有托肯时钟，并且添加、删除托肯时如不发生扩容，均不会申请新内存。如果发生扩容，也是一次申请连续内存，效率高于链表的每次申请单个节点。

变迁发射的逻辑要实现两个目标：1、计算出此变迁发射的总耗时。2、更新其他使能变迁的时钟。变迁发射的总耗时即为之前求出的变迁使能的最短时间加上此变迁上的时钟。但是其他变迁有可能在此变迁发射的整个过程中（包过为了让此变迁使能，之前托肯的倒计时过程）使能计时了，因此需要在此环节一并更新他们的时钟。这意味着需要对此时库所使能的变迁计算其使能的最短时间，其计算逻辑与变迁发射前的逻辑中的类似。

变迁发射的逻辑是三段逻辑中最为繁琐的，库所时间网和变迁时间网的特点都会在这段逻辑上体现出来。在变迁时间网中，变迁发射后所有使能变迁是同步开始继续计时的，但结合上库所时间网变迁计时存在先后差异。因此需要计算出其他变迁提前计时的情况，并更新变迁上时钟。

算法 7.3 变迁发射时的逻辑

procedure TLANUCH($TTPPN$,$next$,t)
  $timer \leftarrow TIME(t)$ //$TIME(t)$ 为标识 $next$ 变迁 t 上时钟的倒计时的时间数值
  $time = time + timer$
  for all $t_{other} \in T$ **do**
    if $t_{other}^{-} \leq m$ **then**
      $needGetCount \leftarrow Pre(p,t)$
      $minTime \leftarrow pneedGetCount$
      if $minTime \leq time$ **then**
        $TIME(t_{other}) = TIME(t_{other}) - time + minTime$
        if $TIME(t_{other}) \leq 0$ **then** $TIME(t_{other}) = 0$
        end if
      end if
    end if
  end for
end procedure

变迁发射后的逻辑要实现三个目标：1、对此 Petri 网所有托肯进行计时.2、对需要重置时钟的变迁，重置其时钟。3、对此变迁的后置库所放入托肯。TTPPN 托肯上的时钟是始终在计时的，因此需要对目前 Petri 网中的托肯上的倒计时时钟减去变迁发射的总耗时。当某变迁的前置库所被别的变迁取走过托肯，此变迁上的时钟需要被重置。当前发射的变迁上的时钟也需要被重置。变迁发射后，其后置库所会被放入托肯，这些新放入的托肯上的倒计时时钟为库所时延。

在实际的制造系统中，并非所有库所上都有时间约束，因此并不需要对 Petri 网中的所有托肯重新计算时钟。当某库所上没有时间约束时，应该跳过计时逻辑。这段优化在实际情况下有显著效果，因为在建模时会加入额外的控制库所，这类库所中的托肯数往往会远高于其他库所，如果不跳过这些托肯的时钟计算，会浪费大量算力。

算法 7.4 变迁发射后的逻辑

procedure AFTERTLANUCH($TTPPN$,$next$,t)
  for all $p \in P$ **do**
    if 标识 $next$ 库所 p 上没有时间约束 **then**
      跳过计时
    end if
    for all $token \in TOKEN(p)$ **do**
      $TIME(token) = TIME(token) - time$
      if $TIME(token) < 0$ **then**
        $TIME(token) = 0$
      end if
    end for
  end for
  for all $p \in {}^\bullet t$ **do**
    for all $t_{other} \in p^\bullet$ **do**
      $TIME(t_{other}) = 0$
    end for
  end for
  for all $p \in t^\bullet$ **do**
    $needPutCount \leftarrow Post(p, t)$
    for $i \leftarrow 1, needPutCount$ **do**
      将 $token$ 添加进 $TOKEN(p)$ 中
      $[a, b] \leftarrow I_p(p)$
      $TIME(token) = a$
    end for
  end for
end procedure

上述逻辑发射解决了时间区间有下界无上界的情况。例如某托肯在库所中最早需要停留 3 个时间单位，当不能超过 5 个时间单位，使用上述逻辑还不能实现。本文对于这种情况的实现思路为：直接将超过上界约束的标识设置为死锁标识，令其所有的变迁均无法使能。

之后调度算法在发射变迁 t 时会先判断此变迁能否使能。如果能使能，则发射此变迁。

算法 7.5 使能判断逻辑

输入: 变迁库所时间网 $TTPPN$，变迁库所时间网的标识 $Marking$，准备发射的变迁 t

输出: 布尔值

1: **if** 此标识 $Markintg$ 超过时间约束 **then**
2: return false
3: **end if**
4: return 发射变迁 t 标识 $Marking$ 是否满足库所使能要求

如下图所示，这是一个库所变迁时延网的模型。一共有四个变迁和四个库所。库所 p_2 中有两个托肯，其余的库所中均没有托肯。所有的库所和变迁上均有时间约束。库所上的时间约束既有上界又有下届。而变迁上的时间约束只有上界。

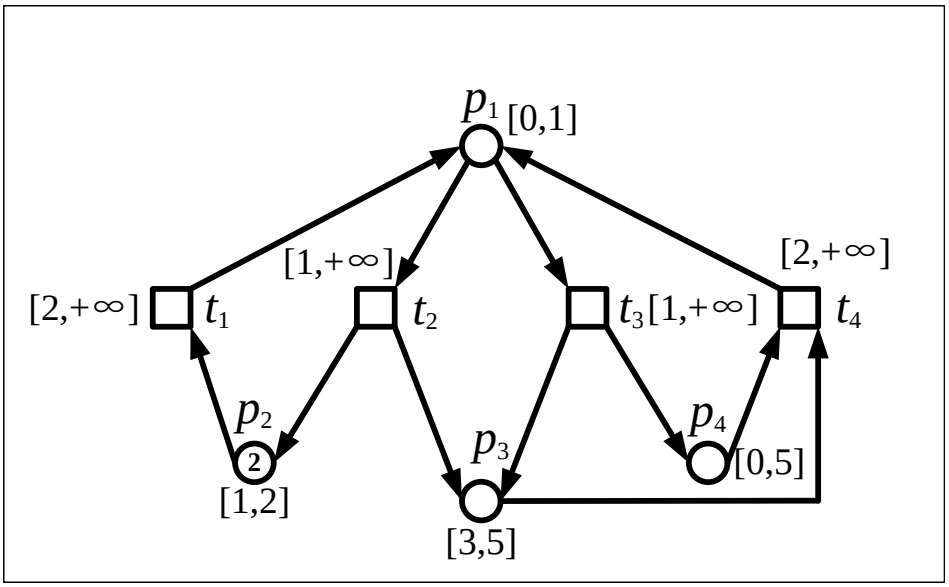


图 7.3　一个库所变迁时间网的例子

初始情况下库所 p_4 中没有托肯，当 p_4 中被移入 1 个托肯作为目标。有以下的调度策略：$t_1 -> t_3$。其中标识的序列为：$M_1 : (0,2,0,0)$ 全局时间:0，变迁使能时间:0；$M_2 : (1,1,0,0)$ 全局时间:3，变迁使能时间:1；$M_3 : (0,1,1,1)$ 全局时间:4，变迁使能时间:3。

标识 M_2 是变迁 t_1 发射后生成的。t_1 需要从 p_2 中取走一个托肯，如果要使能，必须等待其唯一的前置库所 p_2 中的一个托肯完成倒计时。因此 t_1 的最小使能时间是 1 个时间单位。同时 Petri 网中的其他托肯会同时倒计时，这意味着 p_2 中所有的托肯都完成了倒计时。t_1 发射后，如果 p_2 还需要被取走一个托肯，那么取走它的变迁是可以直接使能的。

t_1 使能后，至少需要等待 2 个时间单位才能发射。因此 M_2 的全局时间为 3 个时间单位。

t_3 需要从 p_1 中取走一个托肯。而 p_1 中的托肯不需要等待就可以被取走，因此 t_3 可以立刻使能。本程序会尽可能的减少无意义的驻留，所以 t_3 的使能时间即为 M_2 生成的最早的全局时间，也就是 3 个时间单位即可使能。使能后至少需要 1 个时间单位才可以发射，因此 M_3 的全局时间为 4 个时间单位。

7.2　Petri 网调度算法项目的架构设计

本文为我研究生期间参与某半导体企业设计晶圆制造的预研项目的研究。此项目的目标为对实际的晶圆制造系统进行调度。因此需要按要求对系统进行建模，并使用调度算法对模型求解调度策略。

我负责算法的设计与开发。对此类系统进行建模的方式有许多种，在项目初期，本项目组负责建模的同学尝试了变迁时间网、库所时间网等多种时间网进行建模。同样的，求解 Petri 网调度策略的算法也有很多种，有经典的图搜索算法也有群体智能算法。因此我设计了一个求解各种 Petri 网模型的调度算法集合的程序架构。

此架构使用 Java 语言开发，整体分为 3 个接口：Marking、PetriNet、Search。

Marking 表示系统状态，比如库所向量，变迁、库所上的时钟会在这个接口的实现类中声明。因为调度算法中频繁使用哈希表，所以根据系统状态求取哈希值的方法也在此接口实现类中声明。

PetriNet 存有系统的结构，是用于实现系统状态转移的。结合 Petri 网的实际背景，此接口主要对外提供两个方法，发射和判断使能。调度算法一般会遍历所有变迁，使用此接口判断变迁是否使能，如果能够使能再根据实际情况，选择是否发射此变迁。使用发射方法时，会传入一个变迁，此接口的实现类内部会有当前系统的状态，发射方法完成时会返回发射变迁后系统的下一个状态，也就是返回一个 Marking 接口的对象。调度算法得到此对象后，可以更新系统的当前状态，并进行接下来的循迹。

Search 接口对外提供一个 search 方法，使用此方法后会返回一个解对象，包含标识序列和变迁序列。变迁序列即为算法得到的调度策略，标识序列为系统按调度策略运行时各个阶段的状态。

本架构的优点为使用接口改变了算法和模型的依赖关系，实现了解耦。通常情况下，模型的底层环节，算法是顶层环节，算法依赖于模型。使用本架构后，算法依赖于模型的接口，具体模型去实现模型接口，更换模型时不需要修改算法。

之后本人一直使用此程序架构进行开发，在模型层面先后编写了普通 Petri 网、变迁时间网、库所时间网、变迁库所时间网等代码；在算法层面开发了 A 星算法、蚁群算法、遗传算法、贪心算法等算法。

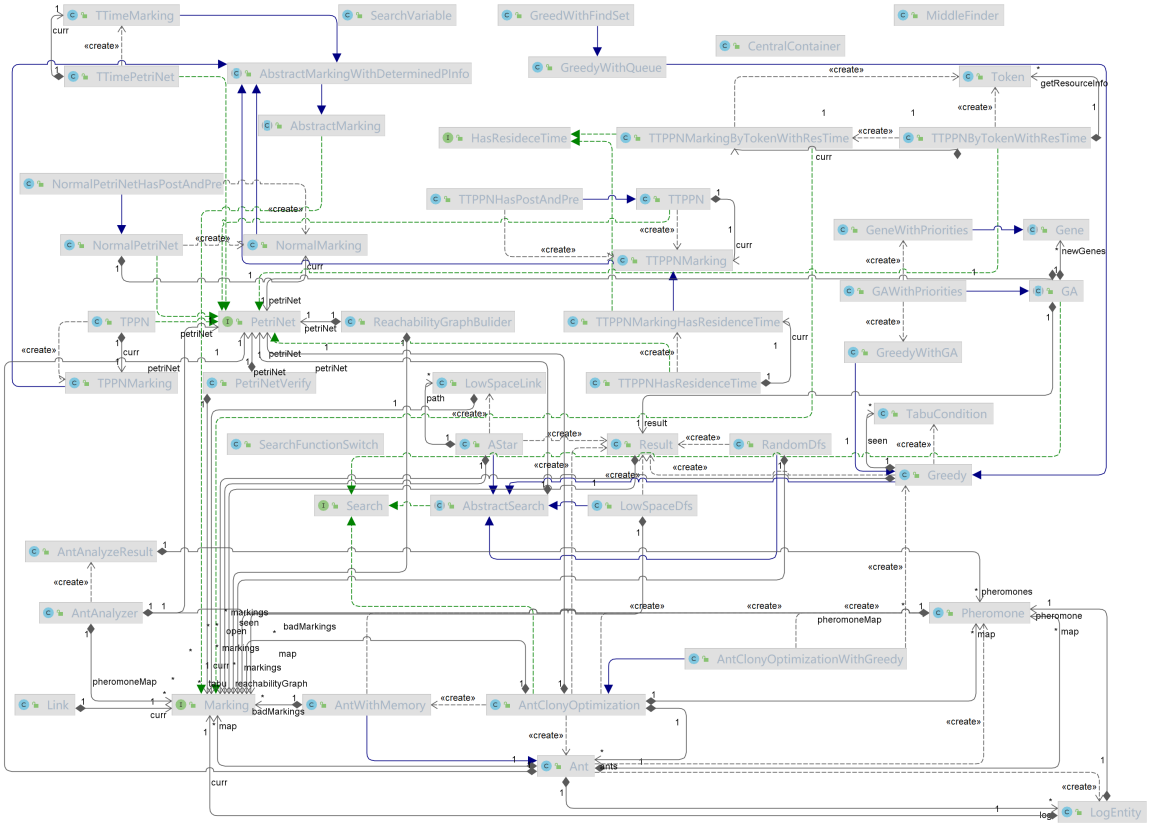


图 7.4　算法程序架构图

目前程序架构如图所示。

7.3　使用库所变迁时间网对实际制造系统进行建模

前文介绍了各种时间网，并结合变迁时间网和库所时间网提出一种新的时间网子类：变迁库所时间网（TTPPN），以及设计了一套 TTPPN 变迁发射流程，保证了发射变迁后系统全局时间尽可能早，供之后的调度算法使用。

本节将使用上一节提出的 TTPPN 对一个实际的晶圆制造系统进行建模。模型来源于 2022 年第六届全国大学生集成电路创新创业大赛 (北方华创杯)。

7.3.1　使用 Petri 网对晶圆制造系统建模的特点

晶圆的加工制造包含多种工艺，需要多种设备协同完成，这些设备的组合运行导致了集束生产设备的高度复杂性。另外，相比传统的仅能完成单一工艺流程的固定生产线，晶圆制造需要灵活的工艺方案以及多种工艺流程同时进行。因此组合设备的调度方案需要不断更换，这导致设备调试以及调度方案成本过高。本文将使用 Petri 网工具建立数学模型反应晶圆加工中双臂组合设备运行情况，基于此研究满足约束的调度方案，并将 Petri 网相关的建模、调度理论应用于晶圆加工制造设备的运行控制。

Petri 网是一种适合于柔性制造系统的形式化方法：对非形式化需求做形式化处理时有助于发现歧义、矛盾；系统的形式化模型可以帮助得到半自动甚至全自动的系统开发方法；可以用数学方法验证形式化模型的正确性而避免对每种情况逐一测试；经过形式验证的子系统可以并入更大系统；形式化模型允许不同设计方法相比较。

一个 Petri 网包含两种节点，称为库所和变迁，分别由圆圈和矩形表示。库所和变迁通过有向弧连接，指定出 Petri 网运行的动态规律。用 Petri 网建模制造系统时，通常用库所表示资源和工序的状态，用变迁表示工序事件的发生或起止，库所中的小黑点称为托肯，记录对应资源的数量。

时间是生产调度的根本参考，因此本文将使用时延 Petri 网对晶圆制造设备建模，包括 T-时间 Petri 网（TPN），每个变迁将被指定一个时间区间，该变迁仅在给定时间区间内才可以发生，和变迁库所时延网（TTPPN），变迁仅在给定时间区间内才可以发生且库所中的托肯在给定时间区间才被视作可使用的。一旦变迁到达时间区间的上限，该变迁将强制发射（强语义）。若一个变迁被多重允许，即当前资源可以保证一项工序执行两次及以上，每次变迁发射后时间约束将重新计算（单服务器）。

7.3.2　基于工序的建模方法进行建模

晶圆的加工流程可以视为由四个典型工序组成：校准、进料、加工工序、以及冷却并出料。此处以图 4.1 所示设备为例，加工配方为从 LP1 中取得晶圆放入校准模块 AL，校准完成后放入真空锁 LL 的 S2 槽位，在 PM3 或 PM4 中执行第一道工序，在 PM1、2、5、6 中执行第二道工序，放入 LL 中的 S1 槽位冷却完成后取出放回 LP1。

该配方要求每个晶圆按顺序进行 5 个工序，因此使用序号 0-5 依次表示晶圆状态。其中校准、进料工序由机械手 TM1 调度，工序 1、工序 2 由机械手 TM2 调度，冷却并出料由 TM1、TM2 协作完成。加工过程中，机械手 TM1、TM2 的移动，真空锁的转换，加工模块的加工是相对独立的子模块，机械手对晶圆的取放调度是主要的加工流程，因此按照工序划分后描述并给出相应的关联矩阵。

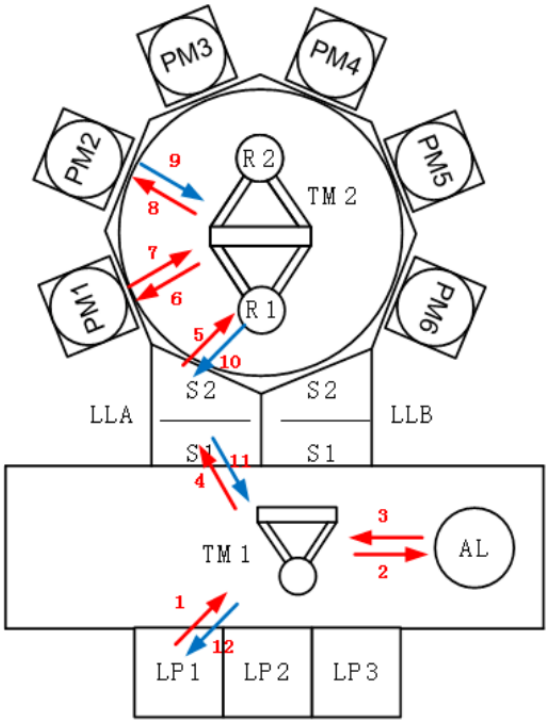


图 7.5　晶圆制造系统

（1）库所与变迁的命名

首先对文中将要出现的库所和变迁的命名规则加以说明：

1. 库所表示系统中的各种资源，可以分为机械手 TM1 和 TM2 的位置，处于不同位置不同状态的晶圆，各加工模块的物理容量，真空锁状态四类：

   (a) 机械手 TM1 和 TM2 的位置：TM1 的可能位置集合 A =LP1, AL, LLA, LLB, 将对应库所的标签命名为 TM1ata，其中 a ∈ A，意为机械手 TM1 处于 a 位置; TM2 中 R1 机械手所处位置集合 B =PM1, PM2, PM3, PM4, PM5, PM6, LLA, LLB，R1atb，其中 b ∈ B，意为机械手 R1 处于 b 位置，同时 R2 与 R1 处在相对位置。

   (b) 处于不同位置不同状态的晶圆：其中晶圆可能被放置的位置集合 C =LP1, AL, PM1, PM2, PM3, PM4, PM5, PM6, AS1, AS2, BS1, BS2，机械手集合 D =TM1, R1, R2，晶圆状态包括 E =0, 1, 2, 3, 4, 5，因此我们用库所 eatc、eind 分别表示 e 状态晶圆处于 c 位置以及处于 d 机械手，库所中托肯的数量代表了相应资源的个数。注意，并非所有组合都有对应库所，例如 0atAS1，因为加工过程中不会把未校准的晶圆放入真空锁，具体使用到的库所和变迁会在后文分工序描述。

   (c) 各加工模块的物理容量：晶圆可能被放置在位置集合 C =LP1, AL, PM1, PM2, PM3, PM4, PM5, PM6, AS1, AS2, BS1, BS2 或机械手集合 D =TM1,

R1, R2，这些位置的物理容量用库所 ccap 或者 dcap 来表示，库所中的托肯数代表了这些位置的容量。以及一个特殊的库所 motor 代表机械手 TM2 取放操作的电机是否空闲。

(d) 真空锁状态：真空锁集合 F =LLA, LLB，真空锁的状态集合 G =v, nv，用 fg 表示真空锁 f 处于状态 g, 其中 v 表示真空状态，nv 表示大气状态。

2. 变迁同样分为四类，主要包括机械手取放各模块上的晶圆、机械手的移动、真空锁状态的切换、各加工模块的加工：

(a) 机械手取放各模块上的晶圆: 机械手集合 D =TM1, R1, R2，晶圆可能被放置的位置集合 C =LP1, AL, PM1, PM2, PM3, PM4, PM5, PM6, AS1, AS2, BS1, BS2，用 dpickc，ddropc 分别表示机械手 d 将晶圆从位置 c 拿起或放入位置 c。

(b) 机械手的移动：TM1 的可能位置集合 A =LP1, AL, LLA, LLB，TM2 中 R1 机械手位置集合 B =PM1, PM2, PM3, PM4, PM5, PM6, LLA, LLB，用 a − a′，b − b′ 分别表示 TM1 从位置 a 移动到 a′，R1 从位置 b 移动到 b′，其中 a, a′ ∈ A，b, b′ ∈ B。

(c) 真空锁状态的切换：真空锁集合 F =LLA，LLB，真空锁的状态集合 G =v，nv，用 Ag − g′，Bg − g′ 分别表示真空锁 A，B 由状态 g 切换到 g′，其中 g, g′ ∈ G。

(d) 各加工模块的加工: 加工模块集合 H =AL, PM1, PM2, PM3, PM4, PM5，PM6，用 hwork 表示加工模块 h 执行加工程序。另外，用 LLAcd，LLBcd 分别表示晶圆在 LLA，LLB 执行冷却工序。

7.3.3　主要功能模块

在晶圆加工需要的工序之外，存在部分子模块独立于主系统运行，他们的运行主要取决于自身的运行规律，同时对主系统中的操作其限制作用。在 Petri 网中表现为独立子网，其中所有变迁的前置库所都包含在子网内部。案例中的机械手移动，真空锁的状态切换，加工模块 PM 对晶圆的加工操作都是独立于主系统运行的功能模块。例如机械手移动，仅与当前位置相关，而机械手的位置是主系统一些操作的必要条件。因此可以对各功能模块建立的子 Petri 网可以直接用来并入总 Petri 网。

（1）机械手移动模块

样例模型包含机械臂 TM1 和机械臂 TM2，其中 TM2 包含两个机械手 R1 和 R2，相对布置。TM1 的可能位置集合 A =LP1, AL, LLA, LLB，共 4 个，将对应库所的标签

命名为 TM1ata，其中 a ∈ A，意为机械手 TM1 处于 a 位置。同样的，TM2 的位置用 R1 所处位置来表示，共 8 个，位置集合 B =PM1, PM2, PM3, PM4, PM5，PM6，LLA，LLB，将对应库所的标签命名为 R1atb，其中 b ∈ B，意为机械手 R1 处于 b 位置。系统初始状态时，R1 处于 LLA 处，TM1 处于 LP1 处。

根据加工顺序要求，单个晶圆的移动包括从仓储 LP1 中取出放入校准模块 AL，从 AL 放入真空锁 LLA 或 LLB，最后从 LLA 或 LLB 取出放回 LP1。考虑到多个晶圆同时加工的情况，提到的 3 个步骤会乱序发生，因此需要加入必要的移动变迁保证 TM1 的连续移动。移动变迁的发射仅仅与机械手位置有关，以 LP1-AL 为例，它的发生需要 TM1 处于当前位置 LP1，发生后使得 TM1 位置处于 AL。

TM2 的移动包括机械手 R1、R2 分别将晶圆从真空锁 LLA 或 LLB 取出放到第一道工序的加工模块 PM3 或 PM4，再从 PM3 或 PM4 取出放入第二道工序的加工模块 PM1、PM2、PM5 或 PM6，最后从第二道工序模块取出放入真空锁 LLA 或 LLB，以及各变迁乱序发生时必要的移动。为了简洁，此处仅列出单个晶圆由 R1 执行各工序需要的移动变迁，实际建模可以考虑列出所有 7×6 个变迁。

TM1 与 TM2 位置库所

编号	库所	释义	初始托肯数
p_1	R1atPM1	机械手 R1 处于 PM1, 机械手 R2 处于 PM5	0
p_2	R1atPM2	机械手 R1 处于 PM2, 机械手 R2 处于 PM6	0
p_3	R1atPM3	机械手 R1 处于 PM3, 机械手 R2 处于 LLB	0
p_4	R1atPM4	机械手 R1 处于 PM4, 机械手 R2 处于 LLA	0
p_5	R1atPM5	机械手 R1 处于 PM5, 机械手 R2 处于 PM1	0
p_6	R1atPM6	机械手 R1 处于 PM6, 机械手 R2 处于 PM2	0
p_7	R1atLLA	机械手 R1 处于 LLA, 机械手 R2 处于 PM3	1
p_8	R1atLLB	机械手 R1 处于 LLB, 机械手 R2 处于 PM4	0
p_9	TM1atLP1	机械手 TM1 处于 LP1	1
p_{10}	TM1atAL	机械手 TM1 处于 AL	0
p_{11}	TM1atLLA	机械手 TM1 处于 LLA	0
p_{12}	TM1atLLB	机械手 TM1 处于 LLB	0

TM1 移动变迁

编号	变迁	释义	前置库所	后置库所
t_1	LP1-AL	从 LP1 运动到 AL	TM1atLP1	TM1atAL
t_2	AL-LLA	从 AL 运动到 LLA	TM1atAL	TM1atLLA
t_3	AL-LLB	从 AL 运动到 LLB	TM1atAL	TM1atLLB
t_4	LLA-LLB	从 LLA 运动到 LLB	TM1atLLA	TM1atLLB
t_5	LLB-LLA	从 LLB 运动到 LLA	TM1atLLB	TM1atLLA
t_6	LLA-LP1	从 LLA 运动到 LP1	TM1atLLA	TM1atLP1
t_7	LLB-LP1	TM1 从 LLB 运动到 LP1	TM1atLLB	TM1atLP1

TM2 移动变迁

编号	变迁	释义	前置库所	后置库所
t_8	LLA-PM3	R1 从 LLA 运动到 PM3	R1atLLA	R1atPM3
t_9	LLA-PM4	R1 从 LLA 运动到 PM4	R1atLLA	R1atPM4
t_{10}	LLB-PM3	从 R1 从 LLB 运动到 PM3	R1atLLB	R1atPM3
t_{11}	LLB-PM4	R1 从 LLB 运动到 PM4	R1atLLB	R1atPM4
t_{12}	PM3-PM1	R1 从 PM3 运动到 PM1	R1atPM3	R1atPM1
t_{13}	PM3-PM2	R1 从 PM3 运动到 PM2	R1atPM3	R1atPM2
t_{14}	PM3-PM5	R1 从 PM3 运动到 PM5	R1atPM3	R1atPM5
t_{15}	PM3-PM6	R1 从 PM3 运动到 PM6	R1atPM3	R1atPM6
t_{16}	PM4-PM1	R1 从 PM4 运动到 PM1	R1atPM4	R1atPM1
t_{17}	PM4-PM2	R1 从 PM4 运动到 PM2	R1atPM4	R1atPM2
t_{18}	PM4-PM5	R1 从 PM4 运动到 PM5	R1atPM4	R1atPM5
t_{19}	PM4-PM6	R1 从 PM4 运动到 PM6	R1atPM4	R1atPM6
t_{20}	PM1-LLA	R1 从 PM1 运动到 LLA	R1atPM1	R1atLLA
t_{21}	PM1-LLB	R1 从 PM1 运动到 LLB	R1atPM1	R1atLLB
t_{22}	PM2-LLA	R1 从 PM2 运动到 LLA	R1atPM2	R1atLLA
t_{23}	PM2-LLB	R1 从 PM2 运动到 LLB	R1atPM2	R1atLLB
t_{24}	PM5-LLA	R1 从 PM5 运动到 LLA	R1atPM5	R1atLLA
t_{25}	PM5-LLB	R1 从 PM5 运动到 LLB	R1atPM5	R1atLLB
t_{26}	PM6-LLA	R1 从 PM6 运动到 LLA	R1atPM6	R1atLLA
t_{27}	PM6-LLB	R1 从 PM7 运动到 LLB	R1atPM6	R1atLLB

（2）真空锁模块

真空锁抽气与充气的动作是独立进行的。此处共有两个真空锁 LLA 和 LLB，LLA 与 LLB 的抽气与充气动作互相独立，动作一旦开始就不会中止：

TM2 移动变迁

编号	库所	释义	初始托肯数
p_{13}	Av	LLA 处于真空状态	0
p_{14}	Anv	LLA 处于大气状态	1
p_{15}	Bv	LLB 处于真空状态	0
p_{16}	Bnv	LLB 处于大气状态	1

真空锁变迁

编号	变迁	释义
t_{28}	Av-nv	LLA 由真空转为大气态
t_{29}	Anv-v	LLA 由大气转为真空态
t_{30}	Bv-nv	LLB 由真空转为大气态
t_{31}	Bv-v	LLB 由大气转为真空态

真空锁关联矩阵

Pre/Post	Av-nv	Anv-v	Bv-nv	Bnv-v
Av	1/0	0/1		
Anv	0/1	1/0		
Bv		1/0	0/1	
Bnv		0/1	1/0	

（3）独立加工模块

加工模块与系统的其他工作互不冲突，一旦原料进入，加工模块开始加工直到工序完成，罗列如下：

7.3.4 主要加工工序

（1）校准工序

校准工序包含机械手 TM1 取出 LP1 中未加工晶圆，放入校准模块 AL，校准完成后取出校准后晶圆，这里认为校准过程需要 TM1 参与。根据加工要求，前置关联矩阵与后置关联矩阵如下：

校准工序库所

编号	库所	释义	初始托肯数
p_{17}	2atPM3	待加工晶圆在加工模块 PM3	0
p_{18}	2atPM4	待加工晶圆在加工模块 PM4	0
p_{19}	3atPM3	已加工晶圆在加工模块 PM3	0
p_{20}	3atPM4	已加工晶圆在加工模块 PM4	0
p_{21}	3atPM1	待加工晶圆在加工模块 PM1	0
p_{22}	3atPM2	待加工晶圆在加工模块 PM2	0
p_{23}	3atPM5	待加工晶圆在加工模块 PM5	0
p_{24}	3atPM6	待加工晶圆在加工模块 PM6	0
p_{25}	4atPM1	已加工晶圆在加工模块 PM1	0
p_{26}	4atPM2	已加工晶圆在加工模块 PM2	0
p_{27}	4atPM5	已加工晶圆在加工模块 PM5	0
p_{28}	4atPM6	已加工晶圆在加工模块 PM6	0
p_{29}	4atAS1	待冷却晶圆在出料口 AS1	0
p_{30}	4atBS1	待冷却晶圆在出料口 BS1	0
p_{31}	5atAS1	冷却完成晶圆在出料口 AS1	0

校准工序变迁

编号	变迁	释义	前置库所	后置库所
t_{32}	PM3work	加工模块 PM3 工作	2atPM3	3atPM3
t_{33}	PM4work	加工模块 PM3 工作	2atPM4	3atPM4
t_{34}	PM1work	加工模块 PM1 工作	3atPM1	4atPM1
t_{35}	PM2work	加工模块 PM2 工作	3atPM2	4atPM2
t_{36}	PM5work	加工模块 PM5 工作	3atPM5	4atPM5
t_{37}	PM6work	加工模块 PM6 工作	3atPM6	4atPM6
t_{38}	LLAcd	晶圆在真空锁 LLA 冷却	4atAS1	5atAS1
t_{39}	LLBcd	晶圆在真空锁 LLB 冷却	4atBS1	5atBS1

校准工序库所

编号	库所	释义	初始托肯数
p_{32}	5atBS1	冷却完成晶圆在出料口 BS1	0
p_{33}	0atLP1	LP1 包含未校正晶圆数量	25
p_{34}	0inTM1	未加工晶圆处于机械手 TM1	0
p_{35}	1inTM1	已校正晶圆处于机械手 TM1	0
p_{36}	0atAL	未加工晶圆处于校准模块 AL	0
p_{37}	1atAL	已校正晶圆处于校准模块 AL	0
p_{9}	TM1atLP1	机械手 TM1 处于 LP1	1
p_{10}	TM1atAL	机械手 TM1 处于 AL	0
p_{38}	TM1cap	机械手 TM1 空闲	1
p_{39}	ALcap	校准模块 AL 空闲	1

校准工序变迁

编号	变迁	释义
t_{40}	TM1pickLP1	机械手 TM1 从 LP1 中取出晶圆
t_{41}	TM1dropAL	机械手 TM1 将晶圆放入 AL
t_{42}	ALwork	执行校准程序
t_{43}	TM1pickAL	机械手 TM1 取出校准后晶圆

校准工序关联矩阵

Pre/Post	TM1pickLP1	TM1dropAL	ALwork	TM1pickAL
0atLP1	1/0			
0inTM1	0/1	1/0		
1inTM1				0/1
0atAL	0/1	1/0		
1atAL			0/1	1/0
TM1atLP1 1/1				
TM1atAL		1/1		1/1
TM1cap	1/0	0/1	1/1	1/0
ALcap		1/0		0/1

（2）进料工序

在大气状态下机械手 TM1 将待进料晶圆从 AL 放入进料口 AS2 或 BS2，再在真空状态下由机械手 R1 或 R2 取出，其中 R1、R2 相对布置，不能同时取放晶圆。根据加工要求，进料过程中涉及库所、变迁以及变迁的关联矩阵如下，其中 TM1 在真空锁中取放晶圆 TM1(pick/drop)(AS2/BS2) 需要相应真空锁保持大气状态,TM2 在真空锁中取放晶圆 (R1/R2)(pick/drop)(AS1/BS1) 需要相应真空锁保持真空状态。这里认为机械手拿取真空锁中晶圆时真空锁不进行真空切换：

表 7.1　进料工序库所

编号	库所	释义	初始托肯数
p_{35}	1inTM1	待进料晶圆处于机械手 TM1	0
p_{40}	2inR1	已进料晶圆处于机械手 R1	0
p_{41}	2inR1	已进料晶圆处于机械手 R2	0
p_{42}	1atAS2	待进料晶圆处于进料口 AS2	0
p_{43}	1atBS2	待进料晶圆处于进料口 BS2	0
p_{11}	TM1atLLA	机械手 TM1 处于进料口 AS2	0
p_{12}	TM1atLLB	机械手 TM1 处于进料口 BS2	0
p_{7}	R1atLLA	机械手 R1 处于 LLA	1
p_{8}	R1atLLB	机械手 R1 处于 LLB	0
p_{7}	R2atLLA	机械手 R2 处于 LLA	0
p_{8}	R2atLLB	机械手 R2 处于 LLB	0
p_{38}	TM1cap	机械手 TM1 空闲	1
p_{44}	R1cap	机械手 R1 空闲	1
p_{45}	R2cap	机械手 R2 空闲	1
p_{46}	motor	电机空闲	1
p_{47}	AS2cap	进料口 AS2 空闲	1
p_{48}	BS2cap	进料口 BS2 空闲	1

表 7.2　进料工序变迁

编号	变迁	释义
t_{44}	TM1dropAS2	机械手 TM1 将晶圆放入进料口 AS2
t_{45}	TM1dropBS2	机械手 TM1 将晶圆放入进料口 BS2
t_{46}	R1pickAS2	机械手 R1 取出 AS2 处晶圆
t_{47}	R1pickBS2	机械手 R1 取出 BS2 处晶圆
t_{48}	R2pickAS2	机械手 R2 取出 AS2 处晶圆
t_{49}	R2pickBS2	机械手 R2 取出 BS2 处晶圆

表 7.3　进料工序关联矩阵

Pre/Post	TM1dropAS2	TM1dropBS2	R1pickAS2	R1pickBS2	R2pickAS2	R2pickBS2
1inTM1	1/0	1/0				
2inR1			0/1	0/1		
2inR2					0/1	0/1
1atAS2	0/1		1/0		1/0	
1atBS2		0/1		1/0		1/0
TM1atLLA	1/1					
TM1atLLB		1/1				
R1atLLA			1/1			
R1atLLB				1/1		
R2atLLA					1/1	
R2atLLB						1/1
TM1cap	0/1	0/1				
R1cap			1/0	1/0		
R2cap					1/0	1/0
motor	1/1	1/1	1/1	1/1	1/1	1/1
AS2cap	1/0		0/1		0/1	
BS2cap		1/0		0/1		0/1
Av			1/1		1/1	
Anv	1/1					
Bv				1/1		1/1
Bnv		1/1				

（3）加工工序 1

机械手 R1 或 R2 将待加工工件放进加工模块 PM3 或 PM4，加工完成后夹出，加工期间不需要机械手参与。

表 7.4　加工工序 1 库所

编号	库所	释义	初始托肯数
p_{40}	2inR1	待加工晶圆在机械手 R1	0
p_{41}	2inR2	待加工晶圆在机械手 R2	0
p_{17}	2atPM3	待加工晶圆在加工模块 PM3	0
p_{18}	2atPM4	待加工晶圆在加工模块 PM4	0
p_{19}	3atPM3	已加工晶圆在加工模块 PM3	0
p_{20}	3atPM4	已加工晶圆在加工模块 PM4	0
p_{49}	3inR1	已加工晶圆在机械手 R1	0
p_{50}	3inR2	已加工晶圆在机械手 R2	0
p_{51}	PM3cap	加工模块 PM3 空闲	1
p_{52}	PM4cap	加工模块 PM4 空闲	1
p_{44}	R1cap	机械手 R1 空闲	1
p_{45}	R2cap	机械手 R2 空闲	1
p_{46}	motor	电机空闲	1
p_3	R1atPM3	机械手 R1 处于 PM3	0
p_4	R1atPM4	机械手 R1 处于 PM4	0
p_3	R1atLLB	机械手 R2 处于 PM3	0
p_4	R1atLLA	机械手 R2 处于 PM4	1

表 7.5　加工工序 1 变迁

编号	变迁	释义
t_{50}	R1dropPM3	R1 将晶圆放进加工模块 PM3
t_{51}	R1dropPM4	R1 将晶圆放进加工模块 PM4
t_{52}	R2dropPM3	R2 将晶圆放进加工模块 PM3
t_{53}	R2dropPM4	R2 将晶圆放进加工模块 PM4
t_{54}	R1pickPM3	R1 取出 PM3 处晶圆
t_{55}	R1pickPM4	R1 取出 PM4 处晶圆
t_{56}	R2pickPM3	R2 取出 PM3 处晶圆
t_{57}	R2pickPM4	R2 取出 PM4 处晶圆

表 7.6　加工工序 1 关联矩阵

Pre/Post	R1dropPM3	R1dropPM4	R2dropPM3	R2dropPM4	R1pickPM3	R1pickPM4	R2pickPM3	R2pickPM4
2inR1	1/0	1/0						
2inR2			1/0	1/0				
2atPM3 0/1		0/1						
2atPM4		0/1		0/1				
3atPM3					0/1		0/1	
3atPM4						0/1		0/1
3inR1					0/1	0/1		
3inR2							0/1	0/1
PM3cap	1/0		1/0		0/1		0/1	
PM4cap		1/0		1/0		0/1		0/1
R1cap	0/1	0/1			1/0	1/0		
R2cap			0/1	0/1			1/0	1/0
motor	1/1	1/1	1/1	1/1	1/1	1/1	1/1	1/1
R1atPM3	1/1				1/1			
R1atPM4		1/1				1/1		
R1atLLB			1/1				1/1	
R1atLLA				1/1				1/1

（4）加工工序 2

与工序 1 类似，机械手 R1 或 R2 将待加工工件放进加工模块 PM1、2、5、6 之一，加工完成后夹出。

表 7.7　加工工序 2 库所

编号	库所	释义	初始托肯数
p_{49}	3inR1	待加工晶圆在机械手 R1	0
p_{50}	3inR2	待加工晶圆在机械手 R2	0
p_{21}	3atPM1	待加工晶圆在加工模块 PM1	0
p_{22}	3atPM2	待加工晶圆在加工模块 PM2	0
p_{23}	3atPM5	待加工晶圆在加工模块 PM5	0
p_{24}	3atPM6	待加工晶圆在加工模块 PM6	0
p_{25}	4atPM1	已加工晶圆在加工模块 PM1	0
p_{26}	4atPM2	已加工晶圆在加工模块 PM2	0
p_{27}	4atPM5	已加工晶圆在加工模块 PM5	0
p_{28}	4atPM6	已加工晶圆在加工模块 PM6	0
p_{53}	4inR1	已加工晶圆在机械手 R1	0
p_{54}	4inR2	已加工晶圆在机械手 R2	0
p_{55}	PM1cap	加工模块 PM1 空闲	1
p_{56}	PM2cap	加工模块 PM2 空闲	1
p_{57}	PM5cap	加工模块 PM5 空闲	1
p_{58}	PM6cap	加工模块 PM6 空闲	1
p_{44}	R1cap	机械手 R1 空闲	1
p_{45}	R2cap	机械手 R1 空闲	1
p_{46}	motor	机械手电机空闲	1
p_{1}	R1atPM1	机械手 R1 处于 PM1, 机械手 R2 处于 PM5	1
p_{2}	R1atPM2	机械手 R1 处于 PM2, 机械手 R2 处于 PM6	0
p_{5}	R1atPM5	机械手 R1 处于 PM5, 机械手 R2 处于 PM1	0
p_{6}	R1atPM6	机械手 R1 处于 PM6, 机械手 R2 处于 PM2	0

表 7.8　加工工序 2 变迁

编号	变迁	释义
t_{58}	R1dropPM1	R1 将晶圆放进加工模块 PM1
t_{59}	R1dropPM2	R1 将晶圆放进加工模块 PM2
t_{60}	R1dropPM5	R2 将晶圆放进加工模块 PM5
t_{61}	R1dropPM6	R2 将晶圆放进加工模块 PM6
t_{62}	R2dropPM1	R2 将晶圆放进加工模块 PM1
t_{63}	R2dropPM2	R2 将晶圆放进加工模块 PM2
t_{64}	R2dropPM5	R2 将晶圆放进加工模块 PM5
t_{65}	R2dropPM6	R2 将晶圆放进加工模块 PM6
t_{66}	R1pickPM1	R1 取出 PM1 处晶圆
t_{67}	R1pickPM2	R1 取出 PM2 处晶圆
t_{68}	R1pickPM5	R1 取出 PM5 处晶圆
t_{69}	R1pickPM6	R1 取出 PM6 处晶圆
t_{70}	R2pickPM1	R2 取出 PM1 处晶圆
t_{71}	R2pickPM2	R2 取出 PM2 处晶圆
t_{72}	R2pickPM5	R2 取出 PM5 处晶圆
t_{73}	R2pickPM6	R2 取出 PM6 处晶圆

表 7.9　R1 加工工序 2 关联矩阵

Pre/Post	R1dropPM1	R1dropPM2	R1dropPM5	R1dropPM6	R1pickPM1	R1pickPM2	R1pickPM5	R1pickPM6
3inR1	1/0	1/0	1/0	1/0				
3atPM1	0/1							
3atPM2		0/1						
3atPM5			0/1					
3atPM6				0/1				
4atPM1					1/0			
4atPM2						1/0		
4atPM5							1/0	
4atPM6								1/0
4inR1					0/1	0/1	0/1	0/1
PM1cap	1/0				0/1			
PM2cap		1/0				0/1		
PM5cap			1/0				0/1	
PM6cap				1/0				0/1
R1cap	0/1	0/1	0/1	0/1	1/0	1/0	1/0	1/0
motor	1/1	1/1	1/1	1/1	1/1	1/1	1/1	1/1
R1atPM1	1/1				1/1			
R1atPM2		1/1				1/1		
R1atPM5			1/1				1/1	
R1atPM6				1/1				1/1

表 7.10　R2 加工工序 2 关联矩阵

Pre/Post	R1dropPM1	R1dropPM2	R1dropPM5	R1dropPM6	R1pickPM1	R1pickPM2	R1pickPM5	R1pickPM6
3inR2	1/0	1/0	1/0	1/0				
3atPM1	0/1							
3atPM2		0/1						
3atPM5			0/1					
3atPM6				0/1				
4atPM1					1/0			
4atPM2						1/0		
4atPM5							1/0	
4atPM6								1/0
4inR2					0/1	0/1	0/1	0/1
PM1cap	1/0				0/1			
PM2cap		1/0				0/1		
PM5cap			1/0				0/1	
PM6cap				1/0				0/1
R2cap	0/1	0/1	0/1	0/1	1/0	1/0	1/0	1/0
motor	1/1	1/1	1/1	1/1	1/1	1/1	1/1	1/1
R1atPM1	1/1				1/1			
R1atPM2		1/1				1/1		
R1atPM5			1/1				1/1	
R1atPM6				1/1				1/1

7.3.5　冷却与出料工序

在真空状态下，机械手 R1 或 R2 将加工完成的工件放进真空锁出料口 AS1，BS1 之一，冷却完成后等待。在大气状态下，机械手 TM1 将 AS1 或 BS1 中冷却完成的晶圆取出并放回 LP1。这里认为机械手拿取真空锁中晶圆时真空锁不进行真空切换，晶圆冷却的同时可以切换真空锁状态。

表 7.11　冷却与出料库所

编号	库所	释义	初始托肯数
p_{53}	4inR1	待冷却晶圆在机械手 R1	0
p_{54}	4inR2	待冷却晶圆在机械手 R2	0
p_{29}	4atAS1	待冷却晶圆在出料口 AS1	0
p_{30}	4atBS1	待冷却晶圆在出料口 BS1	0
p_{31}	5atAS1	冷却完成晶圆在出料口 AS1	0
p_{32}	5atBS1	冷却完成晶圆在出料口 BS1	0
p_{59}	5inTM1	冷却完成晶圆在机械手 TM1	0
p_{60}	5atLP1	待冷却晶圆在出料口 LP1	0
p_{61}	AS1cap	出料口 AS1 空闲	1
p_{62}	BS1cap	出料口 BS1 空闲	1
p_{44}	R1cap	机械手 R1 空闲	1
p_{45}	R2cap	机械手 R2 空闲	1
p_{38}	TM1cap	机械手 TM1 空闲	1
p_{46}	motor	机械手电机空闲	1
p_{7}	R1atLLA	机械手 R1 处于 LLA	1
p_{8}	R1atLLB	机械手 R1 处于 LLB	0
p_{7}	R2atLLA	机械手 R2 处于 LLA	0
p_{8}	R2atLLB	机械手 R2 处于 LLB	0
p_{11}	TM1atLLA	机械手 TM1 处于 LLA	0
p_{12}	TM1atLLB	机械手 TM1 处于 LLB	0
p_{9}	TM1atLP1	机械手 TM1 处于 LP1	1
p_{13}	Av	LLA 处于真空状态	0
p_{14}	Anv	LLA 处于大气状态	1
p_{15}	Bv	LLB 处于真空状态	0
p_{16}	Bnv	LLB 处于大气状态	1

表 7.12　冷却与出料变迁

编号	变迁	释义
t_{74}	R1dropAS1	R1 将晶圆放进出料口 AS1
t_{75}	R1dropBS1	R1 将晶圆放进出料口 BS1
t_{76}	R2dropAS1	R2 将晶圆放进出料口 AS1
t_{77}	R2dropBS1	R2 将晶圆放进出料口 BS1
t_{78}	TM1pickAS1	TM1 将晶圆从 AS1 取出
t_{79}	TM1pickBS1	TM1 将晶圆从 BS1 取出
t_{80}	TM1dropLP1	TM1 将晶圆从放入 LP1

表 7.13　冷却出料关联矩阵

Pre/Post	R1dropAS1	R1dropBS1	R2dropAS1	R2dropBS1	TM1pickAS1	TM1pickBS1	TM1dropLP1
4inR1	1/0	1/0					
4inR2			1/0	1/0			
4atAS1	0/1		0/1				
4atBS1		0/1		0/1			
5atAS1					1/0		
5atBS1						1/0	
5inTM1				0/1	0/1	1/0	
5atLP1							0/1
AS1cap	1/0		1/0		0/1		
BS1cap		1/0		1/0		0/1	
R1cap	0/1	0/1					
R2cap			0/1	0/1			
TM1cap					1/0	1/0	0/1
motor	1/1	1/1	1/1	1/1			
R1atLLA	1/1						
R1atLLB		1/1					
R2atLLA			1/1				
R2atLLB				1/1			
TM1atLLA					1/1		
TM1atLLB						1/1	
TM1atLP1							1/1
Av	1/1		1/1				
Bv		1/1		1/1			
Anv					1/1		
Bnv						1/1	

此晶圆制造系统 Petri 网模型如图所示。

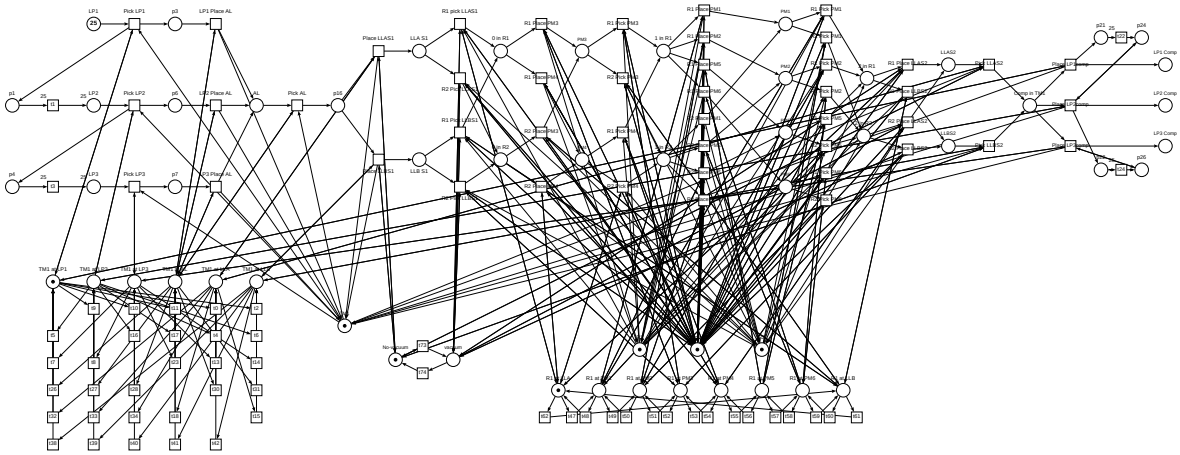


图 7.6　实际晶圆制造系统

本文第四章将使用蚁群算法对此模型进行调度。

7.4　本章小结

本章在上一章的基础上，将库所时间网和变迁时间网结合起来，提出了一种新的时间网子类：库所变迁时间网（TTPPN）；设计了一套 TTPPN 发射变迁的流程，供下一

章的调度算法使用；设计了一种计算各种 Petri 网模型的程序架构；最后使用 TTPPN 对一个实际的晶圆制造系统进行建模。

第八章　基于库所变迁时延网的蚁群算法研究

第二章介绍了一系列的时间 Petri 网，第三章提出了一种称为 TTPPN 的时间网子类，并基于 TTPPN 对一个实际的半导体晶圆生产系统进行了建模。

本章将基于上章建立的 TTPPN 模型，使用算法计算出一条变迁序列。TTPPN 发射此变迁序列，会到达一个规定好的终止标识。在实际的半导体晶圆生产系统中，最开始，晶圆会堆放在一个腔体中。随着加工过程的运行，晶圆会被带离初始的腔体，经过一系列的环节，最终被放入到另一个腔体中。在 TTPPN 中，这些腔体可被建模为库所。因此算法的任务即为寻找一条能将一个库所中所有托肯转移到另一个库所中的变迁序列。

另外，按上一章的变迁发射逻辑，每个变迁发射都会有一个具体的时刻，将此时刻作为一个属性存入 TTPPN 的标识中，称为全局时刻。TTPPN 发射变迁序列生成的最终标识的全局时刻即为此系统完成晶圆加工的总耗时，称为完工时间。实际情况下，需要完工时间越短越好。这个值将用来衡量算法的解质量。算法从开始到找到终止标识所消耗的时间称为程序运行时间。这个值将用来衡量算法的速度。对于启发式算法、群体智能算法来说，时间复杂度的分析十分困难。另外程序运行时间还会受操作系统进程调度策略的影响，未必能如实反映算法速度。因此本章将记录 Petri 网的发射次数，用以作为分析算法速度的补充数据。当 Petri 网的库所、变迁数确定时，发射变迁产生新标识的时间开销也是确定的。而对于蚁群算法，其大部分时间开销是发射变迁带来的，因此用发射次数来表征算法的速度。

8.1　基于 TTPPN 的基本蚁群算法

蚁群算法（Ant Clony Optimization，ACO）是一种群智能算法，它是由一群无智能或有轻微智能的个体（Agent）通过相互协作而表现出智能行为，从而为求解复杂问题提供了一个新的可能性。蚁群算法最早是由意大利学者 Colorni A., Dorigo M. 等于 1991 年提出。算法运行时蚂蚁会以更高的概率选择信息素浓度高的路，而蚂蚁会为长度短的路留下更多的信息素。因此选路机制与信息素添加机制构成了一个正反馈的逻辑，随着程序的持续运行，正反馈会将解往更优的方向引导[?][?][?][?][?]。

TTPPN 为时间网，其托肯和变迁上均有时钟，如果要完整的表示 TTPPN，需要记录每一个托肯和变迁上的时钟。即便两标识中所有库所中的托肯数都相同，但只要托肯上的时钟计时不同，依然要视为不同的标识。因此 TTPPN 与其不加时间的网相比，可达图会异常庞大。

但是在蚁群算法中，即便不存储 TTPPN 的完整信息，依然可以保证上述正反馈逻辑。当区分标识仅考虑标识的库所向量时，可达图便可看作是一个边上权值可变的有向图。

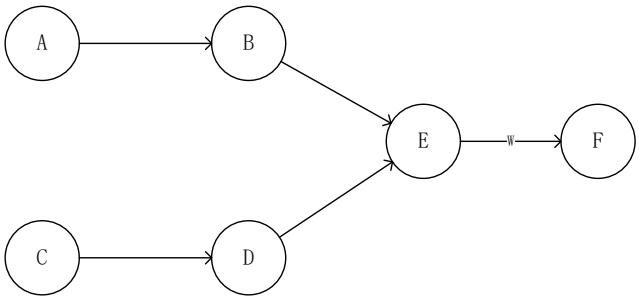


图 8.1　可变权值图

边上权值会受到初始标识到此标识路径的影响。选路机制与信息素添加机制构成的正反馈逻辑并不会被破坏，因此即便不区分时钟计时不同的标识，依然可以使用蚁群算法。

基于 TTPPN 的基本蚁群算法可分为三个阶段：蚂蚁循迹阶段、信息素稀释阶段和信息素添加阶段。

算法 8.1 基于 TTPPN 的基本蚁群算法

输入: 变迁库所时间网 $TTPPN$，变迁库所时间网的初始标识 $initMarking$，蚂蚁只数 $antCount$，算法迭代轮数 $round$

输出: 变迁序列 $trans$

1: **for** $i \leftarrow 1, round$ **do**
2: 　$Ants \leftarrow$ 初始化 $antCount$ 只蚂蚁放入蚂蚁集合
3: 　$RG \leftarrow$ 初始化一个空的可达图
4: 　$antsTravel(Ants, RG, TTPPN, initMarking)$ //蚂蚁循迹阶段
5: 　$dilute(RG)$ //信息素稀释阶段
6: 　$putPheromone(RG, Ants)$ //信息素添加阶段
7: **end for**
8: 返回找到的最优变迁序列 $trans$

蚂蚁循迹阶段是算法的主要流程，本阶段中每只蚂蚁都会独立地在可达图中进行搜索。

算法 8.2 蚂蚁循迹阶段

procedure ANTSTRAVEL($Ants$,RG,$TTPPN$,$initMarking$)
　for all $ant \in Ants$ **do** //$Ants$ 为蚂蚁的集合
　　$antTravel(ant, RG, TTPPN, initMarking)$ //单只蚂蚁循迹
　end for
end procedure

每只蚂蚁都有一个禁忌表，蚂蚁每次会选择发射一个变迁，并把发射变迁生成的标识存入此禁忌表中，之后的搜索过程会禁止发射到禁忌表中已经有的标识，避免蚂蚁走重复的路。此蚂蚁共有两种结局：1、蚂蚁走到终点标识；2、蚂蚁无路可走。

算法 8.3 单只蚂蚁循迹

procedure ANTTRAVEL(ant,RG)
　$tabu \leftarrow$ 初始化蚁群禁忌表
　$log \leftarrow$ 初始化蚂蚁日志
　while 蚂蚁未到达终点并且未走到死路 **do**
　　$next(ant, RG, TTPPN, initMarking, tabu, log)$ //为蚂蚁选择变迁并发射的逻辑
　end while
end procedure

蚂蚁无路可走既可能是因为蚂蚁走上了死锁标识，没有使能变迁可以发射，也可能是此标识的所有使能变迁发射后的标识均在禁忌表中出现过了。蚂蚁会一直搜索下去直到走到终点，蚂蚁在搜索过程中会记录发射的变迁序列，用以更新可达图上的信息素和输出最后的解。因为每只蚂蚁的搜索过程相互独立，所以每只蚂蚁搜索过程可由独立的线程并发执行，以提升效率。蚁群算法的可达图是不带有时间信息的，仅由标识与标识通过使能变迁连接，标识为图的节点，变迁为图的有向弧。变迁上有一个权值，称为信息素浓度，蚂蚁会根据信息素浓度，并综合考虑到禁忌表，随机选择一个使能变迁发射。如果蚂蚁到达的标识之前未被添加进可达图，则默认这个标识的所有使能变迁上的信息素浓度是相同的。

算法 8.4 蚂蚁选择变迁并发射的逻辑

procedure NEXT($Ants$,RG,$TTPPN$,$initMarking$,ant,RG)
　$t = chose(RG, initMarking)$
　$initMarking = lanuch(TTPPN, initMarking, t)$
　将当前标识添加进禁忌表 $tabu$ 中
　将发射的变迁 t 记录在蚂蚁日志 log 中
end procedure

定义 4.1：记 P_{Mt} 为可达图中标识 M 的变迁 t 上的信息素浓度，$P_{Mt} \in \mathbb{R}$。处于标识 M 的蚂蚁选择发射变迁 t 的概率为 $\frac{P_{Mt}}{\sum_{i \in T} P_{Mi}}$。蚂蚁随机选择变迁发射的算法为

轮盘赌算法。如果有 n 个待选择的元素，此算法会生成 n 个彼此相邻的区间，区间的长度与此元素被选择的概率成正比。算法会随机生成一个大于 0，小于区间长度的数值。这个数值落在哪个区间内，算法就会选择哪个数。

算法 8.5 蚁群算法中的轮盘赌算法

procedure CHOSE(RG,$initMarking$)
  $total \leftarrow 0$
  $probability \leftarrow$ 用于存储变迁和变迁上信息素浓度的键值对集合

  for all $t \in T$ **do**
    if $t^- \leq m$ **then**
      $next \leftarrow lanuch(TTPPN, initMarking, t)$
      if $next \in tabu$ **then**
        continue
      end if
      $value \leftarrow$ 变迁 t 上的信息素浓度
      $total = total + value$
      $put(probability, t, value)$ //$put(probability, t, value)$ 指在键值对集合 probability 中添加键值对 $(t, value)$
    end if
  end for
  $random \leftarrow$ 随机生成一个大于 0，小于 $total$ 的数
  $total = 0$
  for all $t \in keys(probability)$ **do** //$keys(probability)$ 表示键值对集合 $probability$ 中键的集合
    $value \leftarrow get(probability, t)$ //$get(probability, t)$ 表示从键值对集合 $probability$ 中取键 t 的值
    $total = total + value$
    if $total \geq random$ **then**
      return true
    end if
  end for
end procedure

蚂蚁会根据可达图变迁上的信息素浓度的大小，随机选择一个变迁发射，并在走到终点后，为可达图增加信息素。随着蚁群算法不断运行下去，可达图中的信息素浓度会越来越高，导致之后蚂蚁添加的信息素占总信息素浓度的比重越来越低。在这种情况下，后期蚂蚁对解的影响将会减少。而前期可达图中信息素尚少，蚂蚁循迹几乎是随机的，所以后期的蚂蚁比前期的蚂蚁更为重要。因此应该加入信息素稀释机制。

定义 4.2：记 α 为信息素稀释率，$\alpha \in \mathbb{R}$，P_{Mt}^i 为蚁群算法第 i 轮中 P_{Mt} 的值。记 A_{Mt}^{ji} 为第 j 只蚂蚁第 i 轮要在标识 m 上添加的信息素浓度。$P_{Mt}^{i+1} = \alpha P_{Mt}^i + A_{Mt}^{ji}$

每轮可达图上的信息素都会按比例被稀释掉。

算法 8.6 信息素稀释阶段

procedure DILUTE(RG)
  for all $M \in RG$ **do**
    for all $t \in T$ **do**
      $P_{Mt} = \alpha P_{Mt}$
    end for
  end for
end procedure

蚁群在循迹过程中，会将到达终点标识的变迁标识序列保存起来，称为蚂蚁日志（log），并在添加信息素阶段根据蚂蚁日志，更新信息素。如果蚂蚁日志中的标识已经在可达图中存在，则直接更新其上的信息素。如果此标识之前从未被放入可达图，则在此时放入可达图。因此蚁群算法的可达图是随着算法运行，逐渐扩展出来的，并不需要一开始就提供。信息素的添加量要和路径长度成反比，在我们时间网的场景下，蚂蚁日志中的变迁标识序列中最后一个标识记录的时刻是完成调度的总耗时。因此信息素的添加量应与这一时刻成反比。

定义 4.3：记 T_{ji} 为第 j 只蚂蚁第 i 日志的变迁标识序列中最后一个标识记录的时刻，此蚂蚁要在标识 m 上添加的信息素浓度 $A_{Mt}^{ji} = \frac{C}{T_{ji}}$。

算法 8.7 信息素添加阶段

procedure PUTPHEROMONE(RG,$Ants$)
  for all $ant \in Ants$ **do**
    for all $M \in Markings(log)$ **do**
$Markings(log)$ 为蚂蚁日志 log 中的标识序列
      if $M \in RG$ **then**
        $P_{Mt} = A_{Mt}$
      else
        将 M 放入 RG 中
        $value \leftarrow$
        $P_{Mt} = value$
      end if
    end for
  end for
end procedure

上述流程如果以单只蚂蚁为粒度，进行并发计算，无线程安全问题容易解决。因为每只蚂蚁所需要的资源仅有两种：可达图、蚂蚁日志。其中日志为每只蚂蚁私有资源，其他蚂蚁无权操作，因此无线程冲突。而蚂蚁对于可达图有两种操作：读、写。写操作只发生在稀释和更新信息素阶段。而读操作发生在蚂蚁循迹阶段。读写操作是分离的。因此读操作不存在线程冲突。

对于写操作，有线程安全问题，因此需要加锁，并发度不如其他流程。但通过合理选择可达图的数据结构，可以尽可能提升并发度。

8.2 蚁群可达图使用的数据结构

蚂蚁在循迹阶段，需从可达图上得出当前标识上各个使能变迁的信息素浓度。因此蚂蚁对可达图有查询操作。

在信息素浓度稀释阶段，可达图上各个变迁的信息素浓度需按比例稀释。因此可达图需支持修改数据的操作。

在添加信息素阶段，蚂蚁有可能探索到从未被发现的标识。需要将这个标识的信息加入可达图中。因此蚂蚁对可达图有添加信息的操作。

随着算法的一轮轮迭代，某些标识上的使能变迁的信息素已经被稀释到很低了。蚂蚁走上此标识的概率极小，即便走上了，其使能变迁上信息素浓度也近乎均匀。因此应该从可达图中删除这类标识，以节约内存。

基于以上的分析，蚁群算法的可达图需支持增、删、改、查四种操作。所以应该选择增、删、改、查复杂度均为 $O(1)$ 的哈希表作为数据结构实现。

可达图在算法流程中需写入数据，此操作有线程安全问题，因此需要对其加锁。一下两种为哈希表的常规加锁方式。

8.2.1 对操作方法加锁

当算法中有线程要对可达图进行写入操作时，需尝试取得写入操作的锁，并在写入完成后释放锁。锁的数量只有一个。对于取得锁和释放锁的逻辑，均调用操作系统自带的原语以保证原子性。如果某流程具有原子性，则此流程只有两种状态：未开始状态、已完成状态。因此可以使用 CAS 原语实现加锁。CAS 的功能是将某地址的值与给定值比较并交换，并根据比较结果返回特定值。CAS 方法具有原子性，利用 CAS 方法的特点可以设计互斥锁。互斥锁实现方法如下：

1、判断 CAS 操作是否成功，如果不是则重复此步；

2、执行写可达图的操作；

3、使用 CAS 将特定地址的值置回初值。

使用此逻辑，保证了不可能有两个线程同时修改可达图，从根本上解决了线程安全问题。但是这种加锁逻辑，过于粗糙，大大降低了并发度。

使用 CAS 锁，如果某线程未取得锁，它会一直尝试取得锁。可以使用信号量机制，直接让处理机暂停执行此线程，从而执行其他线程，来提高效率。

8.2.2　对数据加锁

即便是对可达图进行修改，如果修改的不是同一个标识的数据，也是不会有线程安全问题的。而同时对同一个标识进行写操作，完全是小概率事件。所以对具体标识进行加锁是更为高效的选择。

基于哈希表的底层原理，实际上是对哈希表内部数组的各个地址进行加锁，锁的数量最多为内部数组长度。当写操作需要处理某地址的数据，先尝试取得此地址上的锁，并在写入完成后释放锁。

锁的数量增加会带来额外的内存开销。如果需减少这部分的内存开销可以将锁加在地址的区间上，以减少锁的数量。

8.3　备份可达图提高算法并发度

本算法分为三个阶段：蚂蚁循迹阶段、信息素稀释阶段、信息素添加阶段。原算法中三阶段是串行的。蚂蚁需要先循迹，得到解后才能添加信息素，因此第一阶段和第三阶段存在依赖关系，必然是串行的。然而第二阶段和一、三阶段的依赖关系并不明确。理论上第二阶段和第三阶段可看作是一个大阶段，即更新信息素阶段。而第一阶段和第三阶段本质上并没有依赖关系，只因为第一阶段蚂蚁循迹需要读可达图，第二阶段可达图稀释需要写可达图，因此为保证线程安全，将其分为两个阶段。

如果将可达图备份一份，这两个阶段是可以并发执行的。

现将可达图分为两份，主可达图（RG）与副可达图（$RGCopy$）。这两份可达图互为副本，在之后蚁群算法一轮轮迭代中会交替充当主可达图供蚁群使用。

在蚂蚁循迹阶段另外加入一个线程，专门用于完成可达图备份与稀释。

主可达图因为上一轮被修改了信息，这部分信息未被加入副可达图中，副可达图也需要同步这部分修改。

修改的内容分为三部分：对原标识的信息素更新、添加新标识、删除信息素浓度过低的标识。因此需遍历原可达图，并基于此修改副可达图的信息。再遍历副可达图，删除原可达图中不存在的标识。

完成蚂蚁循迹流程后，交换原副可达图。

算法 8.8 备份可达图

```
procedure COPY(RG,RGCopy)
    for all m ∈ RG do
        if m ∈ RGCopy then
            同步 m 上的信息素浓度
        else
            将 m 添加入 RGCopy 中
        end if
    end for
    for all m ∉ RGCopy do
        if m ∈ RG then
            从 RGCopy 中删除 m
        end if
    end for
end procedure
```

8.4 蚁群求解实际问题

本节将使用蚁群算法求解第四章对实际晶圆生成系统建立的模型。为衡量蚁群算法解的质量，需要知道模型的最优调度策略。本节将使用迪杰斯特拉算法求解 Petri 网模型的最优变迁序列。

8.4.1 基于迪杰斯特拉算法求解 Petri 网最优变迁序列

迪杰斯特拉算法用于求取图的最短路径。如果图的节点数为 n，此算法可以在 $O(nlog(n))$ 的时间复杂度下得出各节点到起点的最短路径。

基于时间网的背景，设计迪杰斯特拉算法。

（1）存放将要扩展的标识的集合

设计一个集合，此集合专门用于存放接下来将要扩展的标识，命名为 $OPEN$ 集合。Petri 网的可达图是随算法运行动态添加进内存中的，所以 $OPEN$ 中一开始只有初始标识，随着程序的运行，如果遇到有扩展价值的标识，此标识会被放入 $OPEN$ 中。

$OPEN$ 会被取出标识进行扩展，也会被放入标识。但基于迪杰斯特拉算法的机制，取出的标识应该是集合中离原点最近的标识，也就是全局时间最短的标识。

如果遍历这个集合寻找全局时间最短的标识，当集合大小为 n 时，其时间复杂度为 $O(n)$。但是如果使用小根堆去实现，其时间复杂度为 $O(log(n))$。因此本算法将使用小根堆作为此集合的实现。

（2）已经扩展的标识的集合

此集合专门用于存放已经扩展的标识，命名为 $CLOSE$ 集合。当一个标识从 $OPEN$ 取出，并且它的所有变迁均被处理过，就会放入此集合中。因为每次处理的标识都是 $OPEN$ 中全局时间最短的标识，所以当此标识进入 $CLOSE$ 中，如果之后再扩展到它，如果时间网可达图路径上的时间间隔是确定的，其全局时间必然更长。所以当标识已经在 $CLOSE$ 中了，是没有扩展的意义的。

但实际上时间网可达图路径上的权值是可变的，所以 $CLOSE$ 表中存储的为键值对，键为标识，值为目前到达此标识的最短时间。此后判断标识是否值得扩展，应该判断到达此标识的全局时间是否比 $CLOSE$ 表中记录的最短时间短，如果此时间确实更短，才去扩展此标识。

$CLOSE$ 涉及到两种操作：将已完成扩展的标识存入此集合中、查询标识是否在此集合中。因此本算法选取哈希表作为此集合的数据结构实现。哈希表存与查的时间复杂度均为 $O(1)$。使用哈希表实现的 $CLOSE$ 其底层是一个固定长度的数组。标识需要得到 $CLOSE$ 底层数组的一个元素索引，才能进入 $CLOSE$ 表。进入 $CLOSE$ 表的标识会被存放在特定索引上。此索引是根据哈希值得出的，哈希值是使用哈希函数得出的，哈希函数的输入是标识向量。对于哈希值相同的标识，本 $CLOSE$ 表会将其统一存放在一棵红黑树中。如果红黑树的节点数为 n，其增删改查的时间复杂度是 $log(n)$。

8.4.2　算法流程

迪杰斯特拉算法每次会从 $CLOSE$ 中取一个全局时间最短的标识 $curr$，如果此标识就是终点标识，则找到终点。如果 $CLOSE$ 集合取空都未找到终点，则说明终点不可达。对 $curr$ 的所有使能变迁均发射一次，如果发射产生的新标识 $next$ 已经在 $OPEN$ 中存在，并且其全局时间更短，则将其取出，更新其全局时间后再放入。在 $OPEN$ 中查询标识的时间复杂度为 $O(log(n))$，往其放入标识的时间复杂度为 $O(log(n))$。如果 $next$ 的全局时间已经高于 $CLOSE$ 中对于标识处记录的最短时间，则没有必要扩展此标识，如果不存在，之前查找时也没在 $OPEN$ 中找到就将此标识放入 $OPEN$ 中。当 $curr$ 的所有使能变迁均处理完成，将其放入 $CLOSE$ 中。

迪杰斯特拉算法流程如下所示：

算法 8.9 基于时间网的迪杰斯特拉算法

输入: 变迁库所时间网 $TTPPN$，变迁库所时间网的初始标识 $initMarking$，终点标识 $endMarking$

输出: 变迁序列 $trans$

```
for i ← 1, round do
    往 OPEN 放入初始标识 initMarking
    while OPEN 不为空 do
        curr ← 从 OPEN 中取出全局时间最小的标识
        for all t ∈ curr 的使能变迁 do
            next ← launch(TTPPN, curr, t)
            if next 在 OPEN 中且全局时间更短 then
                更新 OPEN 中 next 的全局时间
            end if
            if next 值得扩展 then
                将 next 放入 OPEN
            end if
        end for
        将 curr 放入 CLOSE 中
    end while
end for
返回找到的最优变迁序列 trans
```

8.4.3 实验和结果分析

在处理器为 i5-1130G7，最大内存为 8GB 的设备上，排除文件读入、结果输出的耗时，仅在调用算法前后记录系统时间戳，分别使用迪杰斯特拉算法和蚁群算法求解第五章建立的晶圆制造系统的模型。

（1）迪杰斯特拉算法的实验结果分析

以下是使用 6GB 内存，迪杰斯特拉算法的结果。

变迁序列为:t_1->t_7->t_1->t_2->t_1->t_{17}->t_3->t_7->t_{16}->t_1->t_{23}->t_2->t_1->t_{17}->t_{27}->t_7->t_{16}-> t_1->t_2->t_1->t_4->t_{14}->t_{24}->t_{12}->t_{22}->t_1-> t_{13}->t_{29}->t_{11}->t_{22}->t_1->t_{14}->t_{12}->t_{22}->t_{20}->t_{18}->t_1->t_4->t_{14}->t_{24}->t_{12}->t_{22}->t_{20}->t_{18}->t_{13}->t_1->t_{29}->t_{11}->t_{22}->t_{20}->t_{18}->t_{14}->t_1->t_{12}->t_{22}->t_{20}->t_{18}->t_1->t_4->t_{14}->t_{24}->t_{12}->t_{22}->t_{20}->t_{18}->t_{13}->t_{20}->t_{18}->t_8->t_{29}->t_{22}->t_{14}->t_{20}->t_{18}->t_9->t_{22}->t_{20}->t_{18}-> t_{20}->t_{18}->t_4->t_{10}->t_9->t_{24}->t_{22}->t_6->t_8->t_{22}->t_{28}->t_{10}->t_{20}-> t_{18}->t_9->t_{22}->t_{20}->t_{18}->t_{20}->t_{18}->t_4->t_{10}->t_9->t_{22}->t_{20}->t_{18}

完工时间为：1600.8s

程序运行时间为：8.972s

迪杰斯特拉算法每次都会选取离起点最近的节点进行扩展，因此扩展过程为一层层推进，其扩展过的可达标识数与可达图成比例。本算法在使用变迁发射方法 $launch$ 时会使用到标识的时钟信息，而时间网的标识数巨大，为节约内存开销，本算法仅将

标识向量存入 $CLOSE$ 中。因此在扩展过程中会产生大量的垃圾信息，需要被垃圾回收。内存大小影响垃圾回收的次数，内存越大，垃圾回收次数越少。

分别测试从 1GB 内存到 6GB 内存迪杰斯特拉算法的程序运行时间。

表 8.1　内存与程序运行时间关系

内存（GB）	1	2	3	4	5	6
程序运行时间（s）	12.695	10.930	9.031	8.752	8.803	8.972

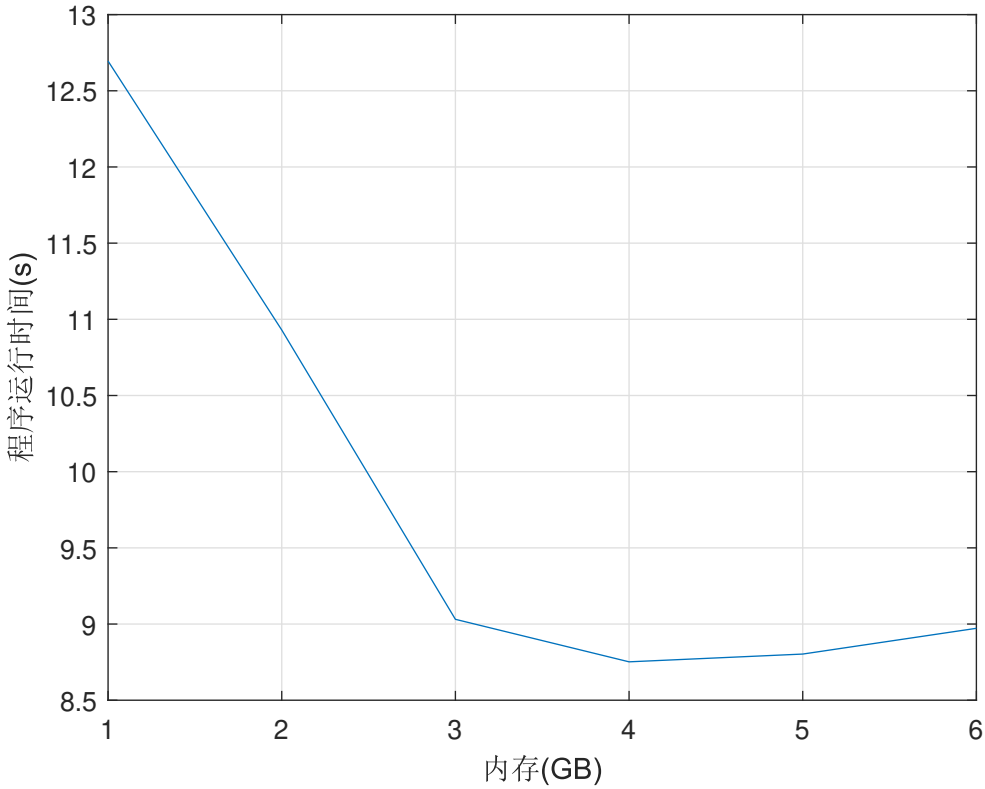


图 8.2　迪杰斯特拉算法内存与程序运行时间关系

如上图所示，随着内存的增加，程序运行时间有下降的趋势，当内存增加到 3GB 时，程序运行时间稳定在 9s 左右。

（2）蚁群算法的实验结果分析

以下是使用 6GB 内存，蚁群算法 100 只蚂蚁迭代 100 轮的结果。

变迁序列：t_1->t_2->t_1->t_7->t_{17}->t_3->t_{16}->t_{23}->t_1->t_2->t_1->t_4->t_{13}->t_{24}->t_7->t_{14}->t_3->t_4->t_9->t_{10}->t_{24}->t_{22}->t_1->t_{11}->t_{13}->t_{23}->t_{21}->t_{19}->t_{12}->t_{21}->t_{18}->t_{17}->t_{20}->t_{18}->t_1->t_{27}->t_{24}->t_{10}->t_{28}->t_{12}->t_{14}->t_{23}->t_{22}->t_8->t_{21}->t_{24}->t_6->t_{19}->t_{12}->t_{22}->t_{17}->t_8->t_{27}->t_{22}->t_1->t_7->t_1->t_{17}->t_7->t_{20}->t_{18}->t_{23}->t_1->t_2->t_1->t_{17}->t_7->t_{20}->t_{19}->t_3->t_{28}->t_{14}->t_{27}->t_{12}->t_{21}->t_{16}->t_{18}-

$>t_{24}->t_{14}->t_4->t_9->t_{21}->t_{18}->t_6->t_{28}->t_8->t_6->t_{24}->t_{10}->t_{22}->t_4->t_9->t_{10}->t_{21}->t_{18}->t_9->t_{22}->t_{20}->t_{18}->t_8->t_{21}->t_{18}->t_{20}->t_{18}->t_{20}->t_{18}$

完工时间：2554.7s

程序运行时间：23.374s

此结果无论是完工时间还是程序运行时间，均要长于迪杰斯特拉算法的解。

导致此结果的可能原因有 3 个：算法不能收敛、算法收敛到较差解、算法收敛慢。

基于以上的分析，设计实验。依次提高迭代轮数，观察完工时间，如果随迭代轮数增加，完工时间分布随机，则说明算法未收敛。如果随迭代轮数增加，完工时间区域稳定值，并且稳定值接近迪杰斯特拉算法的解，说明算法收敛慢，如果稳定值显著高于迪杰斯特拉的解，说明算法收敛到较差解。

另外，在计算机 CPU 有限的情况下，提高蚂蚁只数会时程序频繁进行线程切换，增大程序运行时间。使用需要探寻蚂蚁只数与算法效率的关系。

将迭代轮数从 10 轮每次递增 10 轮直到 100 轮，统计完工时间，观察蚁群算法是否有收敛趋势。

表 8.2　蚁群算法迭代轮数与完工时间的关系

轮数	10	20	30	40	50	60	70	80	90	100
完工时间（s）	2554.7	2254.1	1956.1	1921.7	2223.8	2494.7	2513.5	2487.9	2393.8	2554.7

如图所示，蚁群算法在本场景下并没有收敛，迭代轮数在 30 轮之前，完工时间有下降趋势，但随着轮数的增加，完工时间也增加，最终有稳定在 2500.0s 的趋势。

从这个结果可以看出，此蚁群算法收敛到一个较差的解上，而且并不是收敛到局部最优解，因为随着迭代轮数增加，算法曾到达了较优的解，但依然有向差解接近的趋势。

为证明前 30 轮确实有下降趋势，而非是偶然现象，将迭代轮数从 1 轮每次递增 10 轮直到 10 轮，统计完工时间。

表 8.3　蚁群算法迭代轮数与完工时间的关系（小轮数）

轮数	1	2	3	4	5	6	7	8	9	10
完工时间（s）	3274.6	3136.1	3153.8	3024.6	2952.2	2840.0	2708.3	2630.0	2600.5	2554.7

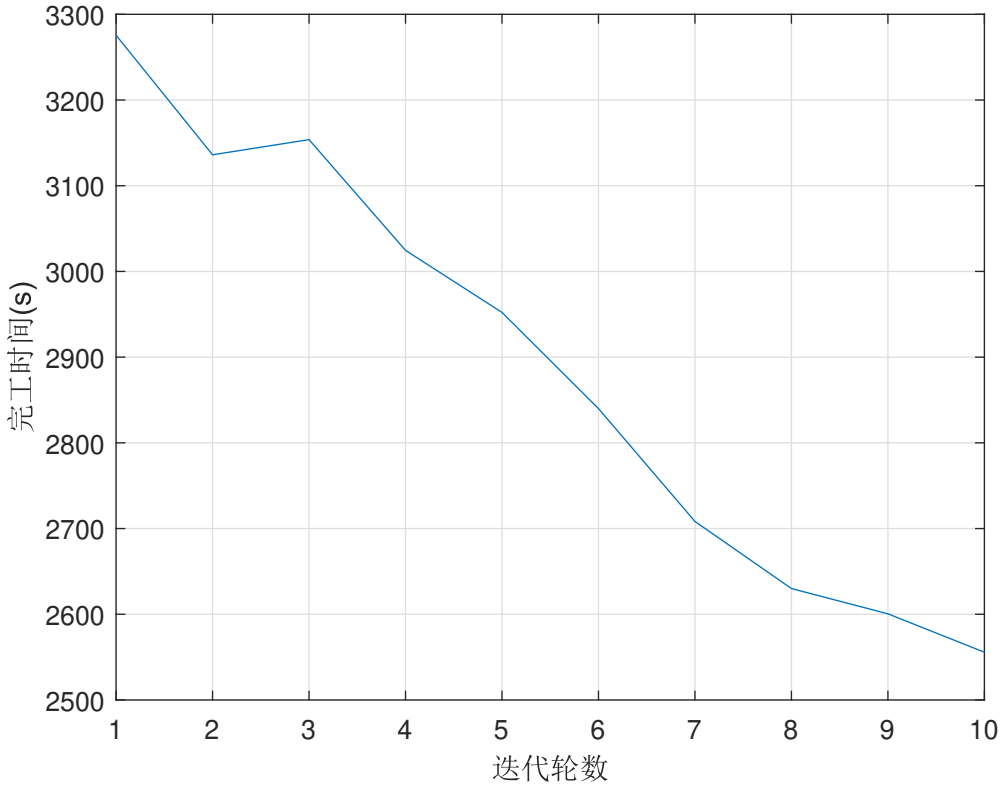


图 8.3　蚁群算法迭代轮数与完工时间关系（1 到 10 轮）

如上图所示，低轮数时蚁群算法确实有收敛趋势。

本程序中每只蚂蚁都是一个线程，如果线程数量小于 CPU 数量，各个线程能并行指向，彼此互不影响。而如果线程数量高于 CPU 数量，CPU 就需要不断第切换到不同的线程，以实现程序并发地执行。如果两线程竞争一个 CPU，必然会导致其中一个线程等待。因此随蚂蚁只数增加，程序运行时间会增加。此外线程切换本身也是有开销的。所以需要探寻蚁群算法蚂蚁只数对完工时间的影响，得出一个恰当的蚂蚁只数，并以此合理配置 CPU 数量。

为了观察蚂蚁只数对算法收敛性的影响，固定迭代轮数为 10 轮，从 10 只蚂蚁每次递增 10 只直到 100 只对此模型进行求解，观察完工时间。

表 8.4　蚁群算法蚂蚁只数与完工时间的关系

蚂蚁只数	10	20	30	40	50	60	70	80	90	100
完工时间（s）	2799.9	2404.6	3184.9	2314.9	2637.2	2718.6	2599.7	2596.0	2894.7	2554.7

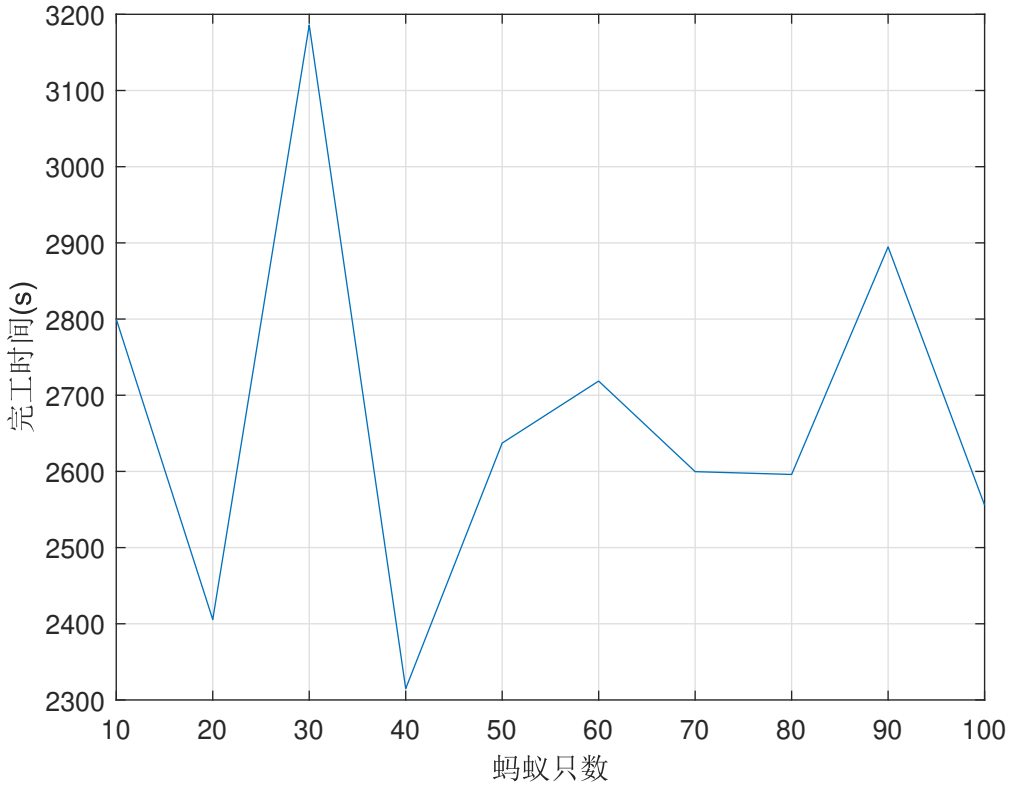


图 8.4　蚁群算法迭代轮数与完工时间关系（1 到 10 轮）

如上图所示，随着蚂蚁只数的增加，完工时间是随机的，这意味着可能 10 只蚂蚁以后，蚂蚁只数就对算法收敛性影响不大了。

为表明低蚂蚁只数时，蚂蚁只数对算法收敛是有影响的，固定迭代轮数为 10 轮，从 1 只蚂蚁每次递增 1 只直到 10 只对此模型进行求解，观察完工时间。

如果蚂蚁只数在增加到一定值后，对解质量影响不大，则应该让蚁群算法运行在 CPU 数量高于此蚂蚁只数的机器上，以避免线程切换与等待的开销，提升算法运行效率。

但是如果蚂蚁只数较高，应该选择处理器更多的 GPU 来运行蚁群算法。本蚁群算法的可达图是动态扩展的，因此存在大量内存申请与清理的操作，此操作不适合在 GPU 上使用，所以本文的代码是为 CPU 编写的。

表 8.5　蚁群算法蚂蚁只数与完工时间的关系（小只数）

蚂蚁只数	1	2	3	4	5	6	7	8	9	10
完工时间（s）	3640.5	3760.5	3262.0	3362.1	3256.0	3008.7	2651.0	2684.5	2752.9	2799.9

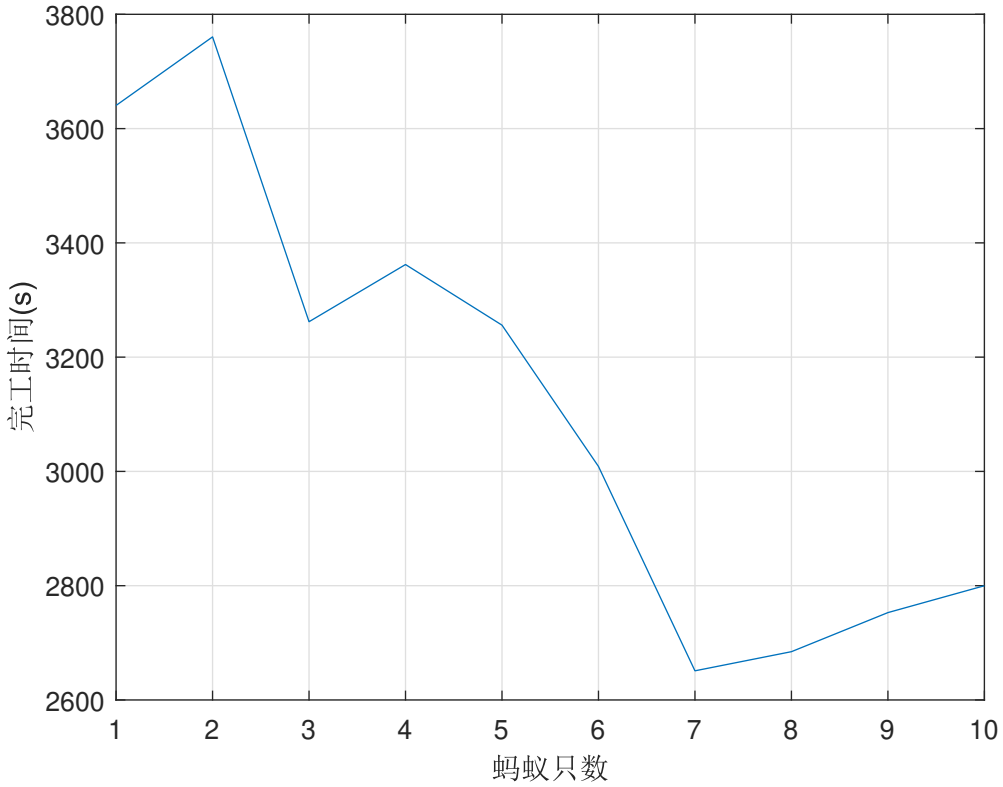


图 8.5　蚁群算法迭代轮数与完工时间关系（1 到 10 轮）

如上图所示，在蚂蚁数量较少时，蚂蚁只数越大，完工时间越小。

综上，提升蚁群算法蚂蚁只数、和算法迭代次数，在一开始会对算法收敛性有较大贡献。随着蚂蚁只数增加到 10 只后，再增加蚂蚁只数对算法影响不大。当迭代轮数增加到 30 轮以后，蚁群算法的解有恶化的趋势，再继续增加迭代轮数会使算法稳定在一个较差的解。

但是无论如何配置蚂蚁只数和迭代轮数，都无法使解向迪杰斯特拉算法求出的解靠近。下一节将继续分析蚁群算法解的情况，并且尝试了一系列优化方案。

8.5　改进的蚁群算法

基于上一小节的数据进行分析，原始的蚁群算法存在求解特定时间网的变迁序列时存在缺陷。如果某时间 Petri 网的最优变迁序列上的使能变迁很少，而完工时间较差的变迁序列彼此又互相连通，原始蚁群算法便无法收敛。其原因为走上最优解的蚂蚁有更高的概率走入死锁标识，失去添加信息素的资格。而走向较差解的蚂蚁会有更高的概率走向终点标识，并在可达图上留下信息素。原始蚁群算法的正反馈机制被打破，使算法无法收敛。为修复这一异常，本文对原始蚁群算法进行了如下改进。

8.5.1 超级蚂蚁机制

如果此蚂蚁的解比目前的最优解还优，则赋予它添加更多信息素的权利，能添加的额外的信息素和对解优化的贡献相关。这么做是为了削弱之前非最优解的蚂蚁对流程的影响。当某蚂蚁运气好，得出的解远好于目前的解，便可迅速提升这个解上的信息素，从而将误入歧途的蚂蚁重新引入正轨。

定义 4.5：记 $MinTime$ 为蚁群算法运行时，当前得到的最短完工时间。$CurrTime$ 为当前蚂蚁完成循迹后得出的完工时间。β 为此蚂蚁添加信息素的增益。

$$\beta = \begin{cases} 1 & MinTime - CurrTime \leqslant 0 \\ MinTime - CurrTime & MinTime - CurrTime > 0 \end{cases}$$

此蚂蚁添加的信息素应乘上此增益。因此第 j 只蚂蚁第 i 轮要在标识 m 上添加的信息素浓度 A_{Mt}^{ji} 应修改为

$$A_{Mt}^{ji} = \beta A_{Mt}^{ji}$$

当此蚂蚁循迹得出的解好于目前找到的最优解，它添加的信息素会被增益 β 放大。当得出的解的完工时间越短，增益越大。蚁群算法出现局部收敛时，使用此机制，蚂蚁只要以小概率探索到了局部以外的更优解，此解便会被添加大量信息素，引导其他蚂蚁探索新解，跳出局部收敛。

此蚂蚁需将找到的解更新为新的最优解。因此在每只蚂蚁的添加信息素逻辑中需添加一段更新最优解流程。

算法 8.10 更新最优解

```
procedure RENNEWMINTIME
    if β > 1 then
        MinTime = CurrTime
    end if
end procedure
```

此流程中需修改 $MinTime$ 这一全局变量。而添加信息素逻辑是并发执行的，因此此处 $MinTime$ 有可能同时被多个线程写入。但因为此处即便有线程安全问题，也不影响算法执行，因此为提高并发度，不进行加锁。

8.5.2 贪心选取初始信息素

原算法蚂蚁遇到从未走过的地点（没有一只蚂蚁去过的地点），会以平均的概率选取下一步。如果蚂蚁所在标识的各使能变迁是等价的，并不知道哪个变迁更优，蚂蚁在首次经过此标识时确实应该以平均的概率选择一条变迁。但结合实际背景，这些

变迁并不等价。加工腔中的晶圆如果加工完毕，应该尽早取走。基于此特点，现将蚂蚁初次选路的策略修改为以更大的概率选择近的路。

将原算法中遇到新标识时构建新信息素对象的逻辑修改为以下逻辑。

算法 8.11 贪心选取初始信息素

procedure GREEDYINITPHEROMONE
  for all $t \in T$ **do**
    if $t^{-} \leq m$ **then**
      $nextMarking \leftarrow lanuch(currMarking, t)$
      将变迁 t 上的信息素浓度设置为 $time(nextMarking) - time(currMarking)$
$time(marking)$ 为取得标识 $marking$ 全局时间的方法
    end if
  end for
end procedure

对于此标识的所有使能变迁，均发射一次得出此标识的后继标识。按这些后继标识的全局时间对这些变迁分配初始信息素。时间越短初始信息素浓度越高。

这是一种贪心的思想。发射使后继标识全局时间最短的变迁，意味着寻求单步最优的变迁发射。实际生产系统中单步最优构成的解往往质量不差，因此这种逻辑在某些场景下，效果不错。但如果模型最优解远离贪心解，此机制探寻新路径时依然会使用贪心的逻辑，会影响算法的收敛效率。因此下一节将提出一种约束较弱的贪心策略。

8.5.3　使用贪心算法预添加信息素

与上一节使用的思想类似，使用贪心的思想优化算法。具体逻辑为：先使用贪心算法求解出一个解，在此解的变迁标识序列上添加一部分信息素。再使用蚁群算法在此基础上继续求解。

贪心算法会不断地发射使后继标识全局时间最短的变迁，直到找到终点。因为可达图中存在死锁标识，需为算法加入回溯机制，遇到死锁标识时，回溯，并发射使后继标识的全局时间次大的变迁。

因此定义一个函数 $find(marking)$，此函数的功能为判断以标识 $marking$ 为初始标识，判断是否能探索到与终点标识连通的路径。

算法 8.12 贪心算法

```
procedure FIND(marking)
    if marking 为终点标识 then
        return true
    end if
    对 marking 的使能变迁按全局时间进行排序，并装入 enableTrans 中
    for all t ∈ enableTrans do
        nextMarking ← lanuch(t, currMarking)
        if nextMarking 被扩展过 then
            continue
        end if
        if find(nextMarking) then
            return true
        end if
    end for
    return false
end procedure
```

此函数使用递归实现了回溯功能。

在函数运行时一并记录发射的变迁，如果发生回溯，删除回溯的变迁，便得到一条变迁序列。基于此序列预先为可达图分配信息素。

8.5.4 复活蚂蚁

当蚂蚁走到死锁标识时循迹结束，此时得到的变迁序列并不能到达终点标识，因此无法基于此序列添加信息素。如果最优解附近存在大量死锁，那么大部分最优解附近的蚂蚁将失去添加信息素的资格，反而使得差解信息素浓度过高。本节设计了一种机制，可以为使走上死锁的蚂蚁复活。

定义 4.6：记 $similarity(s_1, s_2)$ 为标识序列 s_1, s_2 的相似度。$sameMarkingCount(s_1, s_2)$ 为标识序列 s_1, s_2 中相同标识的数量。$length(s_1)$ 为 s_1 中标识的数量。

计算方法为：

$$similarity(s_1, s_2) = \frac{sameMarkingCount(s_1, s_2)}{length(s_1)}$$

如果此蚂蚁死亡，则将其替换为活蚂蚁中与其相似度最高的蚂蚁。

8.5.5 蚁群算法与模拟退火算法结合

如果蚁群算法真的如预期，应该是会随着迭代次数的增加，收敛到一个最优解。这意味着最优解实际上是出现在后期的。因此前期解的重要性不如后期。尽管本蚁群算法每轮都会去稀释信息素，但是是对信息素浓度总体进行成比例稀释。而信息素浓

度分为两部分：初始信息素 + 添加信息素。在前期解的重要性较弱的情况下，应该让蚂蚁更为随机地选择路径。所以前期应该降低添加信息素所占比例。基于此思路，设计模拟退火机制：对添加信息素添加限制，随着轮数增加，逐步放开此限制。

定义 4.6：记 ε 为蚂蚁添加信息素的增益，i 为蚁群算法当前迭代轮数，$round$ 为蚁群算法的总轮数。

$$\varepsilon = \frac{i}{round}$$

将第 j 只蚂蚁第 i 轮要在标识 m 上添加的信息素浓度 A_{Mt}^{ji} 应修改为

$$A_{Mt}^{ji} = \varepsilon A_{Mt}^{ji}$$

随着轮数增加，增益 ε 会逐渐增加，加大了后期蚂蚁的重要性。

先前代码中为了加速收敛，提升了对解有贡献蚂蚁的地位。这种蚂蚁出现在前期会让期待解变得很苛刻，以至于后期难有蚂蚁成为对解有贡献蚂蚁。而模拟退火机制会削弱前期蚂蚁的地位。两种机制实际上是矛盾的。

8.5.6　蚂蚁回溯

基于对本测试用例的解的分析，未避免走上最优解的蚂蚁重新走上岔路而死亡，最为直观的策略为使蚂蚁拥有回溯的功能。直接添加回溯功能会使蚂蚁不能死去，以至于所有蚂蚁都要等待最后一只蚂蚁走到终点才能进行添加信息素的流程。这无疑会极大地增加算法的时间开销。所以应该设置蚂蚁寿命上限，来提前结束对解优化意义不大的蚂蚁流程。基于以上的分析，此寿命上限理应是当前最优解的全局时间。但如果使用全局时间，意味着其他蚂蚁依然会搜索以此全局时间为半径的部分可达图，时间复杂度过大。所以本算法倾向于使用最长跳数为蚂蚁寿命上限。

在原蚁群算法当蚂蚁循迹时，所有的后继标识均已在禁忌表中存在，以为着蚂蚁死亡。在蚂蚁死亡时，将蚂蚁的当前标识替换为蚂蚁日志中倒数第二个标识，再删除日志的最后一个标识，便完成了回溯。

算法 8.13 蚂蚁回溯

```
procedure antTraceBack
    currMarking ← log[length(log) − 2]
    删除 log 中最后一个标识
end procedure
```

在执行回溯操作时，不能从禁忌表中删除回溯的标识。回溯完成后，因为回溯掉的标识依然存在于禁忌表中，因此蚂蚁再次出发时，不会走向被回溯的标识。

使用蚂蚁跳数作为寿命，需要在单只蚂蚁循迹流程执行 $next$ 方法后对此蚂蚁寿命加 1。将这种策略定义为最小步数策略（Min Step Strategy）。

算法 8.14 最小步数策略

```
procedure MINSTEPSTRATEGY
    while 蚂蚁未到达终点且蚂蚁未到达寿命上限 do
        while 蚂蚁死亡 do
            antTraceBack()
        end while
        next()
        蚂蚁寿命加 1
    end while
end procedure
```

此策略中蚂蚁寿命将由两部分组成：解的长度、蚂蚁回溯次数。

定义 4.7：记蚂蚁寿命为 $life$, 解的长度为 $length$, 蚂蚁回溯次数为 $back$。

$$life = length + back$$

本质上蚂蚁寿命上限的大小表示着蚂蚁的循迹能力。寿命上限越大，循迹能力越强。因此如果要保证最终解的质量需要较大的蚂蚁寿命上限。而在此机制下，蚂蚁寿命上限会受到解长度的影响。如果将解分为优解与差解，比较其长度，有两种区别的形式：优解长差解短、差解长优解短。

当差解长优解短时，一旦寿命上限收敛到优解时，蚂蚁将失去探寻差解的能力。这对于解质量无不良影响。但是当情况为优解长差解短时，一旦寿命上限收敛到差解时，蚂蚁将失去探寻优解的能力。会使解质量变差。

因此应该将解长度对蚂蚁寿命的影响剔除，这样蚂蚁寿命只与回溯次数有关。

$$life = back$$

为实现此机制，需要将蚂蚁寿命加 1 的逻辑移动至蚂蚁回溯之后。将这种机制定义为最小失败策略（Min Fall Strategy）。

算法 8.15 最小失败策略

```
procedure MINFALLSTRATEGY
    while 蚂蚁未到达终点且蚂蚁未到达寿命上限 do
        while 蚂蚁死亡 do
            antTraceBack()
            蚂蚁寿命加 1
        end while
        next()
    end while
end procedure
```

回溯蚂蚁有两种终止条件：找到终点标识、寿命超过上限。因此蚂蚁寿命的配置会影响到算法效率。

（1）静态蚂蚁寿命

外部设置一个值作为蚂蚁寿命，在程序运行时，此值不会发生改变。在此策略下，程序的最大耗时是容易估计的。悲观的情况下，每一轮所有蚂蚁均无法找到终点，走完了最大寿命。记蚂蚁的个数为 a，迭代轮数为 b，寿命上限为 c，此策略下时间复杂度为 $O(abc)$

但是静态配置寿命上限不能很契合实际情况。寿命上限过低会使得大量蚂蚁无法找到终点，相当于减少了种群数量，使收敛性大大下降。而寿命上限配置过高，所有的蚂蚁都需等待最后一只蚂蚁完成循迹，延长程序运行时间。

（2）最小蚂蚁寿命

此机制下，蚂蚁有权刷新寿命上限，当此蚂蚁走到终点时，寿命比寿命上限还低，可更新寿命上限为自己的寿命。

蚂蚁的寿命上限会随着程序运行不断降低。后期蚁群循迹速度会越来越快。而且只有蚂蚁走到终点才会更新寿命上限，因此总会有可行解上被添加信息素。之后的蚂蚁会依靠这些信息素，相对于静态寿命上限，此策略获得可行解的概率更高。

但是此策略有两个缺陷。1、变迁序列短的解未必完工时间短。更新寿命上限后，可能会排除最优解。2、此策略会使局部收敛更为严重。在最优解周围存在大量死锁标识时，蚂蚁一旦走上最优解，意味着大量的回溯，这消耗了蚂蚁寿命。因此使用此机制，不能很好地发挥出回溯蚁群的优化效果。

（3）统计平均寿命

如果蚂蚁的寿命服从正态分布，那么极少数蚂蚁将拥有极长的寿命。如果放弃这部分蚂蚁将大大提高程序运行速度，而这部分蚂蚁所占的比例很少，即便放弃也不会对解造成很大的影响。

定义 4.8：记 μ 为蚂蚁平均寿命，记 σ^2 为蚂蚁寿命的方差，蚂蚁寿命上限 $maxLife$ 为

$$maxLife = \mu + 3\sigma$$

蚂蚁的平均寿命无法直接得出，需要设计统计量估计。

定义 4.9：记 $\widehat{\mu}$ 为蚂蚁平均寿命 μ 的统计量，μ_i 表示第 i 只蚂蚁的寿命，n 为蚂蚁数量。一般来说为蚂蚁平均寿命的统计量计算公式为：

$$\widehat{\mu} = \frac{1}{n}\sum_{i=1}^{n}\mu_i$$

如果使用此方式估计蚂蚁平均寿命，需要等待一轮蚁群循迹完全结束。如果初始值配置不合理，首轮蚁群循迹效果将不理想。如果初始值配置过大，首轮蚁群中最后一只蚂蚁迟迟无法结束循迹，其余蚂蚁将等待最后一只蚂蚁，大大增加程序运行时

间。如果初始值配置过小，首轮循迹的蚂蚁容易提前结束循迹，样本数量将大大减少，以至于蚂蚁平均寿命将长时间无法被有效更新，之后的蚁群循迹将重复首轮的情况。因此需要设计一种新的统计量，统计蚂蚁平均寿命。

新的统计量将使用迭代的定义方式。

定义 4.10：记 $\widehat{\mu}_i$ 为蚂蚁平均寿命 μ 第 i 代的统计量，μ_i 表示第 i 只蚂蚁的寿命,$\rho \in (0,1)$ 为系数蚂蚁平均寿命的统计量计算公式为：

$$\widehat{\mu}_{i+1} = \rho\widehat{\mu}_i + (1-\rho)\mu_i$$

并且 $\widehat{\mu}_1 = \mu_1$。

此统计量是无偏的，可用数学归纳法证明。

证明. $E(\widehat{\mu}_1) = E(\mu_1) = \mu$
假设 $E(\widehat{\mu}_i) = \mu$
$E(\widehat{\mu}_{i+1}) = E(\rho\widehat{\mu}_i + (1-\rho)\mu_i) = \rho E(\widehat{\mu}_i) + (1-\rho)E(\mu_i) = \mu$
因此此统计量是无偏的。 □

对 $\widehat{\mu}_i$ 求方差，并令 $i \to \infty$，以 ρ 为变量求最小值。得出 $\rho = 0.5707$。

使用此统计量使，只要蚂蚁一旦走到终点，即可更新平均寿命从而解决了上述问题。使用相同的方式设计蚂蚁寿命的方差的统计量为：

$$\widehat{\sigma}^2_{i+1} = \rho\widehat{\sigma}^2_i + (1-\rho)(\widehat{\mu}_i - \mu_i)^2$$

使用此估计方式，尽可能准确地估计蚂蚁寿命上限。

8.6 实验数据及其分析

上一节为了修复在求解第四章给出的晶圆制造系统模型时蚁群算法无法收敛到最优解的问题，提出了超级蚂蚁、贪心选取初始信息素、贪心预添加信息素、复活蚂蚁、蚁群与模拟退火结合、回溯蚁群 6 种优化思路。本节将对上述思路进行实验并分析结果。

分别对上述 6 种优化思路，使用 100 只蚂蚁的蚁群算法分别从 10 轮每次递增 10 轮直到 80 轮进行测试，观察完工时间。

表 8.6　各优化思路完工时间与迭代轮数关系

完工时间（s）　迭代轮数 / 优化思路	10	20	30	40	50	60	70	80
超级蚂蚁	2023.5	1931.2	1924.3	1911.5	1830.2	1911.5	1973.7	2024.1
贪心选取初始信息素	1629.0	1600.8	1612.8	1657.3	1612.5	1600.8	1788.9	1600.8
贪心预添加信息素	1600.8	1600.8	1600.8	1600.8	1600.8	1600.8	1600.8	1600.8
复活蚂蚁	2124.5	2000.5	2024.5	1911.5	1880.5	1920.7	1920.5	2040.5
蚁群与模拟退火结合	2554.7	1921.7	1956.1	1990.1	2393.8	2554.7	2494.7	2023.5
回溯蚁群	1763.5	1662.4	1600.8	1606.2	1600.8	1600.8	1600.8	1600.8

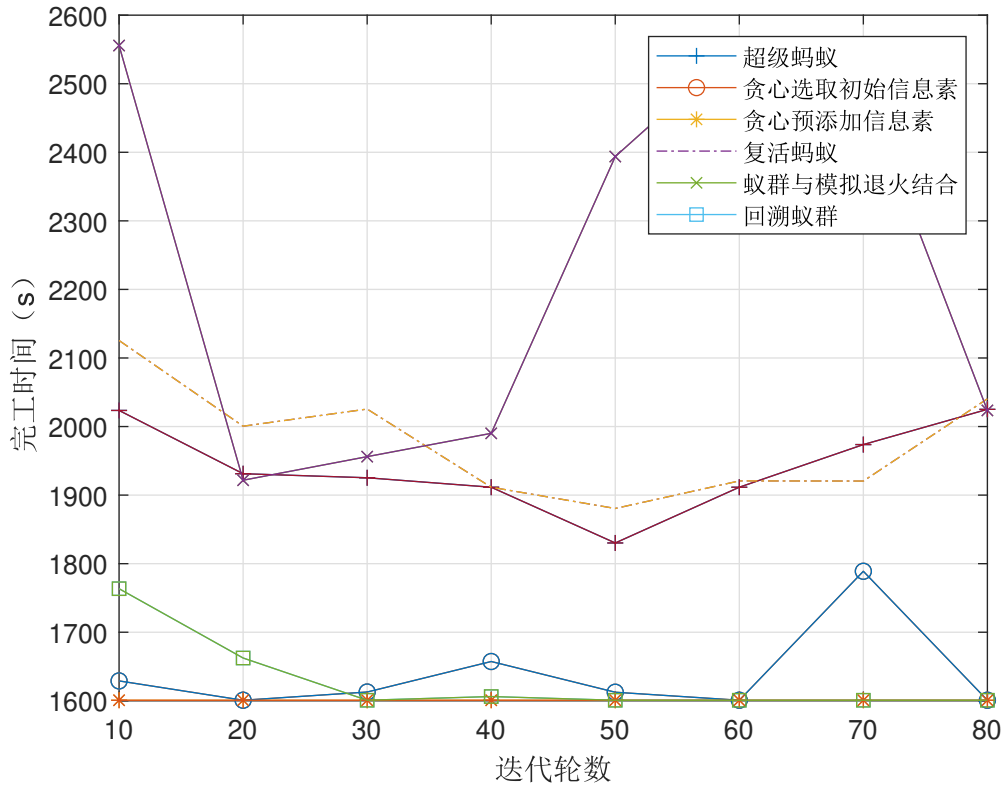


图 8.6　蚁群算法各个优化思路迭代轮数与完工时间关系（10 到 80 轮）

如上图所示，按迭代轮数与完工时间关系曲线的运动趋势可将 6 种优化思路分为 3 类：随迭代轮数增加，完工时间先递减后在较高值处振荡、完工时间始终在迪杰斯特拉解附近、随迭代轮数增加完工时间有收敛到迪杰斯特拉解的趋势。

关系曲线符合随迭代轮数增加，完工时间先递减后在较高值处振荡的优化思路有：超级蚂蚁、复活蚂蚁、蚁群与模拟退火结合

关系曲线符合完工时间始终在迪杰斯特拉解附近的优化思路有：贪心选取初始信息素、贪心预添加信息素

关系曲线符合随迭代轮数增加完工时间有收敛到迪杰斯特拉解的趋势的优化思路有：回溯蚁群

各种优化思路均对解产生了效果，单纯从解质量来看贪心选取初始信息素、贪心预添加信息素、回溯蚁群均能求解出接近迪杰斯特拉算法的解。接下来将对各优化思路的类别进行分析。

8.6.1 对随迭代轮数增加，完工时间先递减后在较高值处振荡的优化思路进行分析

超级蚂蚁、复活蚂蚁、蚁群与模拟退火结合这三种思路随迭代轮数增加，完工时间先递减后在较高值处振荡。将其与原始蚁群相比较，可以发现这三种优化思路都能够加快低迭代轮数时完工时间下降的速度。复活蚂蚁机制的完工时间下降速度低于超级蚂蚁机制。但是随着迭代轮数的继续增加，完工时间下降的趋势明显减缓，最终会在一个较高值附近振荡。加入模拟退火机制的算法的振荡幅度显著高于其他两种机制，并且前期完工时间是从较高值快数下降的。

对于超级蚂蚁来说，如果发现更优的解，更优解会被添加上远高于其他解的信息素，前期几乎所有蚂蚁都会被引导上某条更优解。但是基于上一节对第四章模型解分布的分析，最优解附近存在大量死锁，有走向最优解的蚂蚁很容易遇到死锁提前结束循迹。因此即便使用超级蚂蚁，最优解上积累的信息素也会很快被其他互通的较差解超过。算法无法收敛到最优解。

对于复活蚂蚁来说，前期完工时间下降快是因为提前结束循迹的蚂蚁被其他蚂蚁替代复活后，有效蚂蚁的数量大大增加，实际上相当于增加了蚂蚁数量。但是因为走向最优解并遇到死锁提前结束循迹蚂蚁会被走向较差解的蚂蚁替代，算法依然无法收敛到最优解。

对于模拟退火来说，在还未探索到较优解时，逐步提高后期蚂蚁，将避免算法局部收敛，因此算法前期波动会更大。后期受第四章模型特殊的解分布的影响，拥有更高地位的蚂蚁反而会更剧烈地把解引导到互通的差解区域。因此后期的解依然波动很大。

综上这三种机制并不适合求解类似第四章模型的时间 Petri 网。

8.6.2 对完工时间始终在迪杰斯特拉解附近的优化思路进行分析

贪心选取初始信息素、贪心预添加信息素这两种优化思路完工时间始终在迪杰斯特拉解附近。这两种思路都和贪心的思想紧密相关。有可能第四章模型本身适合使用贪心算法求解，因此需要排除贪心算法本身对解的影响才能分析蚁群算法效果。

使用贪心算法对第四章模型进行求解，结果如下：

变迁序列为:t_1->t_7->t_1->t_2->t_1->t_{17}->t_3->t_7->t_{16}->t_1->t_{23}->t_2->t_1->t_{17}->t_{27}->t_7->t_{16}->t_1->t_2->t_1->t_4->t_{14}->t_{24}->t_{12}->t_{22}->t_1->t_{13}->t_{29}->t_{11}->t_{22}->t_1->t_{14}->t_{12}->t_{22}->t_{20}->t_{18}->t_1-

$>t_4->t_{14}->t_{24}->t_{12}->t_{22}->t_{20}->t_{18}->t_{13}->t_1->t_{29}->t_{11}->t_{22}->t_{20}->t_{18}->t_{14}->t_1->t_{12}->t_{22}->t_{20}->t_{18}->t_1->t_4->t_{14}->t_{24}->t_{12}->t_{22}->t_{20}->t_{18}->t_{13}->t_{20}->t_{18}->t_8->t_{29}->t_{22}->t_{14}->t_{20}->t_{18}->t_9->t_{22}->t_{20}->t_{18}->t_{20}->t_{18}->t_4->t_{10}->t_9->t_{24}->t_{22}->t_6->t_8->t_{22}->t_{28}->t_{10}->t_{20}->t_{18}->t_9->t_{22}->t_{20}->t_{18}->t_{20}->t_{18}->t_4->t_{10}->t_9->t_{22}->t_{20}->t_{18}$

完工时间为：1600.8s

程序运行时间为：0.065s

变迁序列和完工时间均与迪杰斯特拉解一样，因此此网的贪心解就是最优解。

原模型为 S3PR 模型，本就适合最早加工策略的调度，因此使用贪心算法就能求出最优解。破坏原模型的顺序结构，使其加工腔的晶圆可以重入得到网 1 。

基于上文的分析，算法收敛到差解的主要原因是此模型存在死锁，因此需要研究算法在无死锁模型下的情况。网 1 存在大量死锁，对网 1 进行死锁控制形成网 2 。对网 1、网 2 使用 100 只蚂蚁的蚁群算法分别从 10 轮每次递增 10 轮直到 80 轮进行测试两种优化思路，观察完工时间。

网 1 的测试结果如下：

表 8.7　网 1 各优化思路完工时间与迭代轮数关系

完工时间（s）　迭代轮数 / 优化思路	10	20	30	40	50	60	70	80
贪心选取初始信息素	2554.7	2124.5	2024.1	1973.7	1921.1	1973.7	1990.0	1973.7
贪心预添加信息素	2393.8	2024.5	2024.5	2024.1	1973.7	1920.7	1973.1	1921.1

网 2 的测试结果如下：

表 8.8　网 2 各优化思路完工时间与迭代轮数关系

完工时间（s）　迭代轮数 / 优化思路	10	20	30	40	50	60	70	80
贪心选取初始信息素	2040.5	2034.5	1911.5	1820.3	1788.9	1763.5	1662.4	1600.8
贪心预添加信息素	2554.7	2120.1	1973.7	1900.5	1820.3	1788.9	1700.5	1606.2

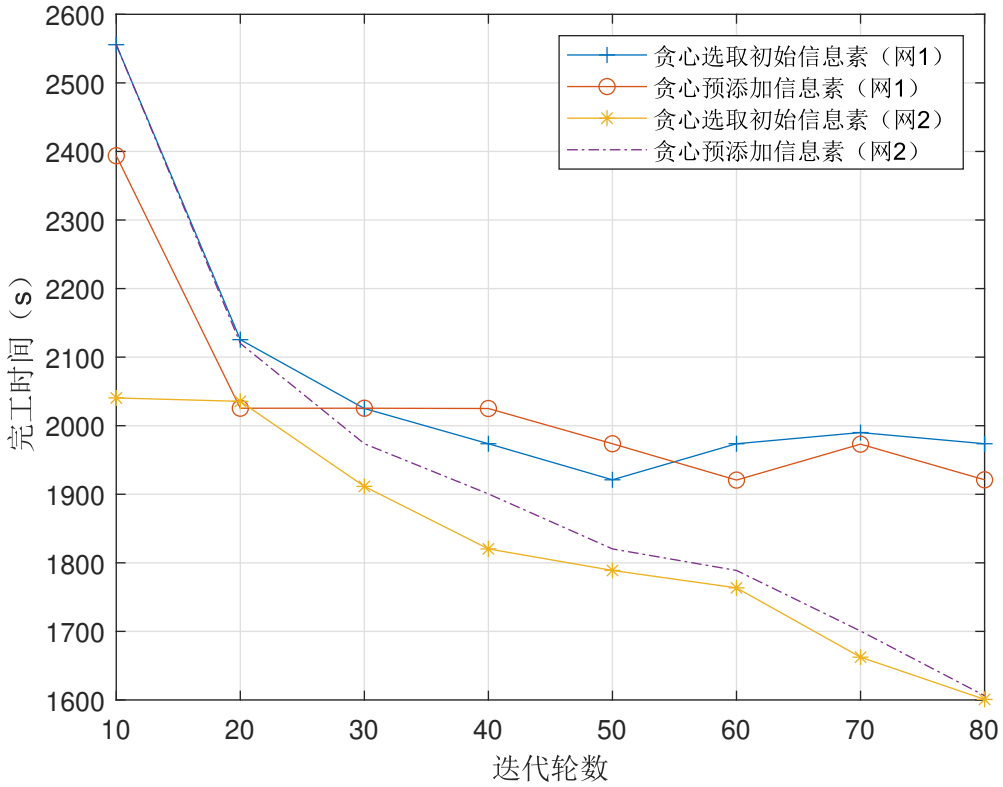


图 8.7　贪心优化思路迭代轮数与完工时间关系（10 到 80 轮）

如上图所示，如果不做死锁控制，加入贪心机制的蚁群算法依然无法收敛到最优解。因此对于最优解附近有大量死锁的时间 Petri 网，这两种优化思路效果有限。但是如果在使用蚁群求解前，预先消除 Petri 网死锁，算法将会有很好的收敛效果。

8.6.3　对随迭代轮数增加完工时间有收敛到迪杰斯特拉解的趋势的优化思路进行分析

只有回溯蚁群机制一种优化思路能让算法随迭代轮数增加完工时间有收敛到迪杰斯特拉解的趋势。

本次测试回溯蚁群时蚂蚁寿命上限是使用统计平均寿命的方法实现的。对静态蚂蚁寿命和最小蚂蚁寿命也在相同条件下进行实验，静态蚂蚁寿命设置为 100 。

表 8.9　各寿命上限机制与完工时间与迭代轮数关系

完工时间（s）＼迭代轮数 蚂蚁寿命上限机制	10	20	30	40	50	60	70	80
静态蚂蚁寿命	2359.2	2120.1	1983.1	1900.5	1880.4	1780.5	1730.5	1616.2
最小蚂蚁寿命	2492.3	2020.4	2123.1	2124.7	2374.2	2522.5	2473.3	2554.7

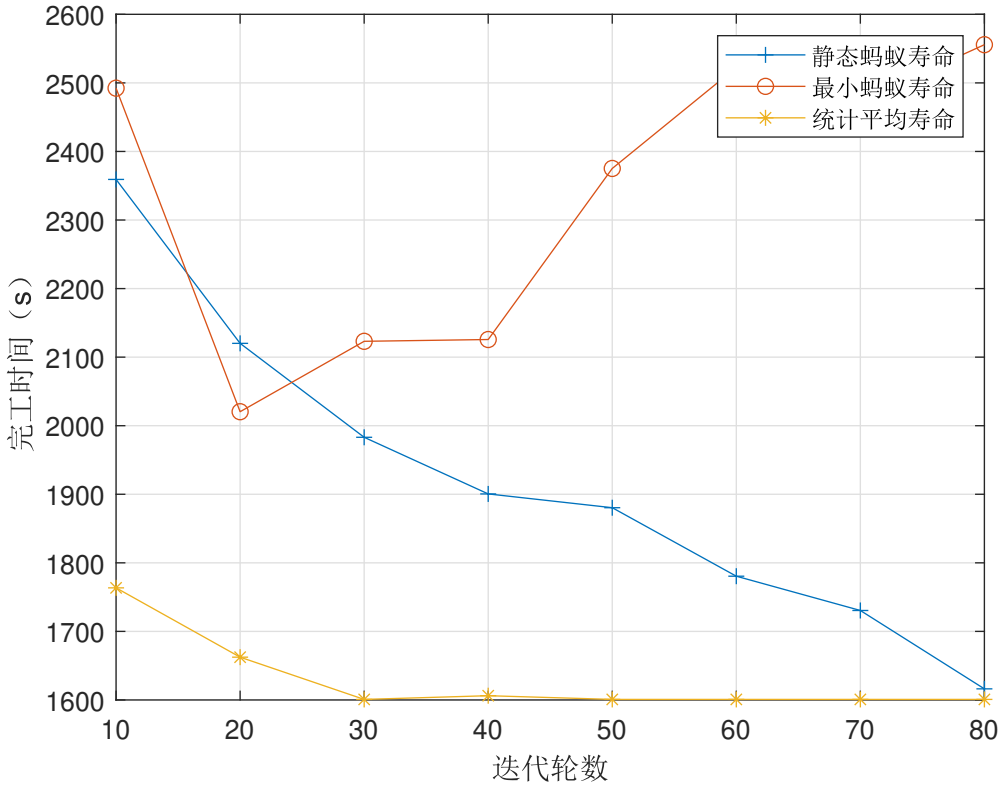


图 8.8　不同蚂蚁寿命上限机制与完工时间关系（10 到 80 轮）

如上图所示，静态蚂蚁寿命与统计平均寿命能够使算法收敛到最优解，但是最小蚂蚁寿命机制随着迭代轮数的增加，解的质量会恶化。

其原因可能是使用最小蚂蚁寿命机制时，算法后期的蚂蚁寿命上限过低，以至于蚂蚁失去回溯能力，无法走上死锁过多的路径。

统计使用最小蚂蚁寿命机制与统计平均寿命时，蚂蚁寿命上限与迭代轮数的关系。

表 8.10　各寿命上限机制与寿命上限与迭代轮数关系

寿命上限 / 迭代轮数 / 寿命机制	10	20	30	40	50	60	70	80
静态蚂蚁寿命	83	24	10	1	1	1	1	1
统计平均寿命	167	201	123	25	12	15	10	13

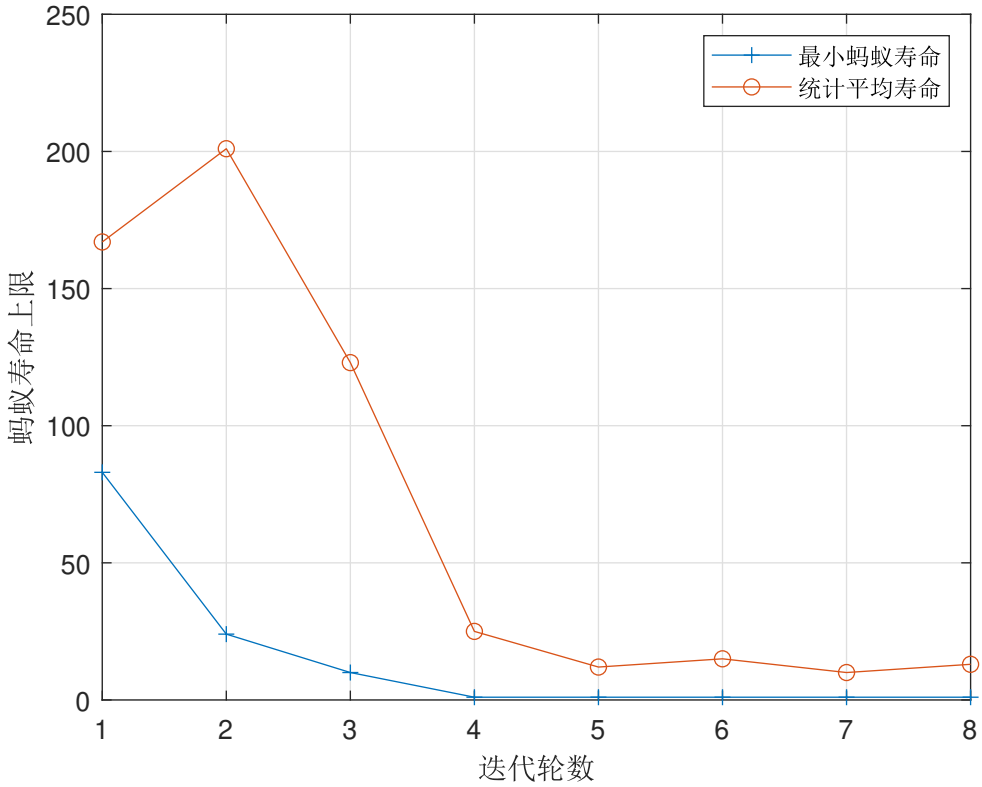


图 8.9　不同蚂蚁寿命上限机制蚂蚁寿命上限与迭代轮数的关系（10 到 80 轮）

如上图所示，使用最小蚂蚁寿命时，蚂蚁寿命上限很快降低到最小值，以至于后期蚂蚁几乎没有回溯能力。但使用统计平均寿命时蚂蚁寿命上限最终收敛于 12 左右，依然具备一定的回溯能力，并且远小于直接配置静态寿命 100，大大节约了程序运行时间。

8.6.4　CPU 数量对回溯蚁群算法速度的影响

基于本章第 1 节对本蚁群算法流程的描述，本文所使用的蚁群算法为多线程算法。所有蚂蚁均可并行地探索可达图，蚂蚁数量越多，在 CPU 数量不做限制时，算法将寻找到更多的解。因此不论从解质量还是程序运行时间的角度看蚂蚁数量越多越好。

但是根据 4.4.3.2 节的实验数据，当蚂蚁数量在 10 以上时，再增加蚂蚁数量对解贡献就不大了。而实际情况下 CPU 数量是有限的，线程切换时会有额外的 CPU 资源的开销，因此不宜设置过高的蚂蚁数量。

迪杰斯特拉算法是单线程算法，不会受到 CPU 数量影响，因此提升 CPU 数量无法减小程序运行时间。在使用了 m1 ultra 的 Mac Studio 上使用迪杰斯特拉算法计算第四章建立的时间 Petri 网模型，程序运行时间为 7.232s。以此为基准，使用 100 只蚂蚁的蚁群算法迭代 100 轮，统计不同 CPU 数量下程序运行时间。

表 8.11　各寿命上限机制与寿命上限与迭代轮数关系

CPU 数量	11	12	13	14	15	16	17	18	19	20
程序运行时间（s）	23.525	14.312	8.753	6.624	4.231	4.441	3.001	2.121	1.361	1.214

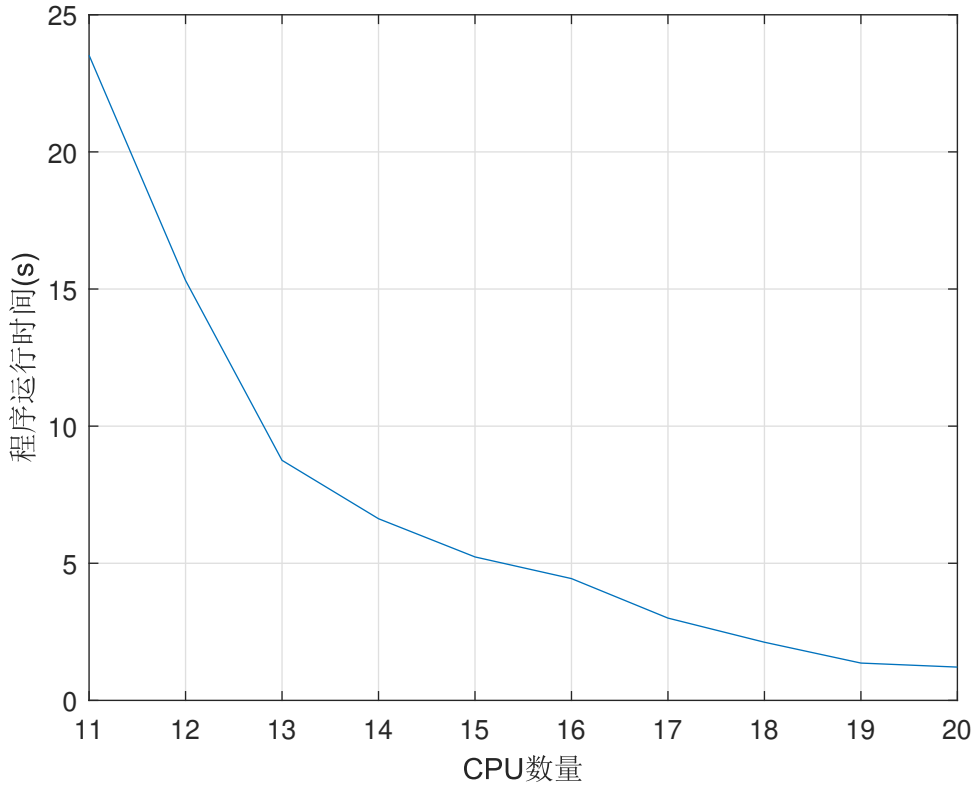


图 8.10　蚁群算法不同 CPU 数量下程序运行时间

如上图所示，当开启 14 个 CPU 时，程序运行时间就已经优于迪杰斯特拉算法了，继续增加 CPU 数量能显著提升算法效果。

8.7　本章小结

本章提出了一种基于时间 Petri 网的蚁群算法，使用此算法求解上一章模型的调度策略，并与迪杰斯特拉算法相比较。发现蚁群算法的解明显劣于迪杰斯特拉算法，而且随迭代轮数增加，算法不能收敛。于是基于实际情况，设计了 6 种优化方案，最终回溯蚁群策略效果明显。

第九章　总结与展望

9.1　总结

晶圆制造系统是用于生产半导体芯片的设备和工艺的集合体。它由多个工艺步骤组成，包括晶圆清洗、切割、涂覆、曝光、蚀刻、离子注入、金属沉积等。晶圆制造系统通常由多个设备组成，包括晶圆清洗机、曝光机、蚀刻机、离子注入机、金属沉积机等。这些设备通常由自动化系统控制，以确保工艺步骤的准确性和一致性。本文使用 Petri 网对实际的晶圆制造系统进行建模，并设计蚁群算法求此系统进行调度。

本文第二章介绍了 Petri 网的基本理论，总结了各种调度算法。并且梳理了一系列时间 Petri 网的相关知识。

第三章根据晶圆制造系统的特点，结合了第二章介绍的变迁时间网可库所时间网，提出了一种新的时间网子类，变迁库所时间网。并且设计了一种此时间网变迁发射的时间计算流程，以保证变迁发射后系统时间尽可能短，以及一个用于计算各种 Petri 网模型的程序架构，供后续调度算法使用。之后使用变迁库所时间网对一个实际的晶圆制造系统进行建模。

第四章使用蚁群算法对第三章建立的模型进行调度。但发现调度结果不能收敛到最优解，通过对解的分析，从不同角度提出了 6 种优化思路，最终回溯蚁群算法效果显著。

9.2　展望

9.2.1　变迁库所时间网的进一步优化思路

如果晶圆需要进入一系列的加工腔加工，并且规定此晶圆从进入第一个加工腔开始，必须在规定时间内完成加工，最直观的做法是有某套机制能够跟踪晶圆。晶圆由托肯表示，各托肯在 Petri 网中是无法区分的。因此之后可以设计并开发一套托肯编号机制。

实际的 Petri 网系统中，既有托肯表示晶圆，也有表示逻辑值，这类托肯在被变迁移动时所进入的库所是不同的，并且实际系统中存在多个晶圆交换操作。因此应该设计一套托肯移动的规则，以保证上述逻辑的正确。

9.2.2　蚁群算法的进一步优化思路

对于蚁群算法的进一步优化，可以从时间和空间两个方面入手。减少算法的时间开销可以通过阻止蚂蚁探索无意义的标识实现；减少空间开销可以设计更高效的数

据结构来实现。

（1）全局时间禁忌表

本文结合 Petri 网的实际情况，提出了回溯蚁群算法并取得了不错的成果。蚂蚁能够回溯后便拥有了发现不可行解的能力。如果将这批不可行解记录下来，下一轮蚁群循迹时提前禁止蚂蚁走上不可行解，将大大提高蚁群算法收敛速度。不可行解由标识组成，因此可以使用一个新的集合存储这些标识。此集合用于避免蚂蚁走上无意义的标识，功能与禁忌表相似，因此将其命名为全局禁忌表。

而本算法是基于时间 Petri 网的，此 Petri 网超过时间约束也会引发死锁，此类死锁受时间因素影响不能简单的判定为不可行解。

综上全局禁忌表存储的应该为一系列的键值对，键为不可行解中的标识向量，值应为一系列的充分条件。如果蚂蚁探索到的标识满足这一系列的条件，则意味着继续探索不可能到达终点标识。这一系列的充分条件会随着蚁群的探索一步步细化，逼近充要条件。

（2）实现禁忌表新的数据结构

如果将 Petri 网的标识向量编码为二进制串，有一种比哈希表更高效的数据结构，二元决策图可以以更低的空间开销存储二进制信息。蚁群算法为多线程算法，更为适合在 GPU 上运行，GPU 的显存带宽远高于内存带宽，但通常来说显存大小不如内存，并且显存与内存通信需要消耗时间。因此可以使用布隆过滤器在显存和内存间做一层中间层，把显存中长期未处理过的标识转移入内存中。内存与硬盘间也可以使用相似的逻辑，这样可以极大提升存储标识的数量。

附录 A　插图示例

西安电子科技大学是以信息与电子学科为主，工、理、管、文多学科协调发展的全国重点大学，直属教育部，是国家“优势学科创新平台”项目和“211 工程”项目重点建设高校之一、国家双创示范基地之一、首批 35 所示范性软件学院、首批 9 所示范性微电子学院、首批 9 所获批设立集成电路人才培养基地和首批一流网络安全学院建设示范项目的高校之一。2017 年学校信息与通信工程、计算机科学与技术入选国家“双一流”建设学科。

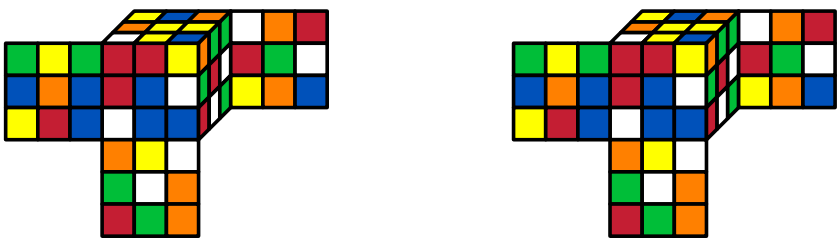
A.1　子图

学校前身是 1931 年诞生于江西瑞金的中央军委无线电学校，是毛泽东等老一辈革命家亲手创建的第一所工程技术学校。1958 年学校迁址西安，1966 年转为地方建制，1988 年定为现名。

建校 90 年来，学校始终得到了党和国家的高度重视，是我国“一五”重点建设的项目之一，也是 1959 年中央批准的全国 20 所重点大学之一。20 世纪 60 年代，学校就以“西军电”之称蜚声海内外。毛泽东同志曾先后两次为学校题词：“全心全意为人民服务”、“艰苦朴素”。

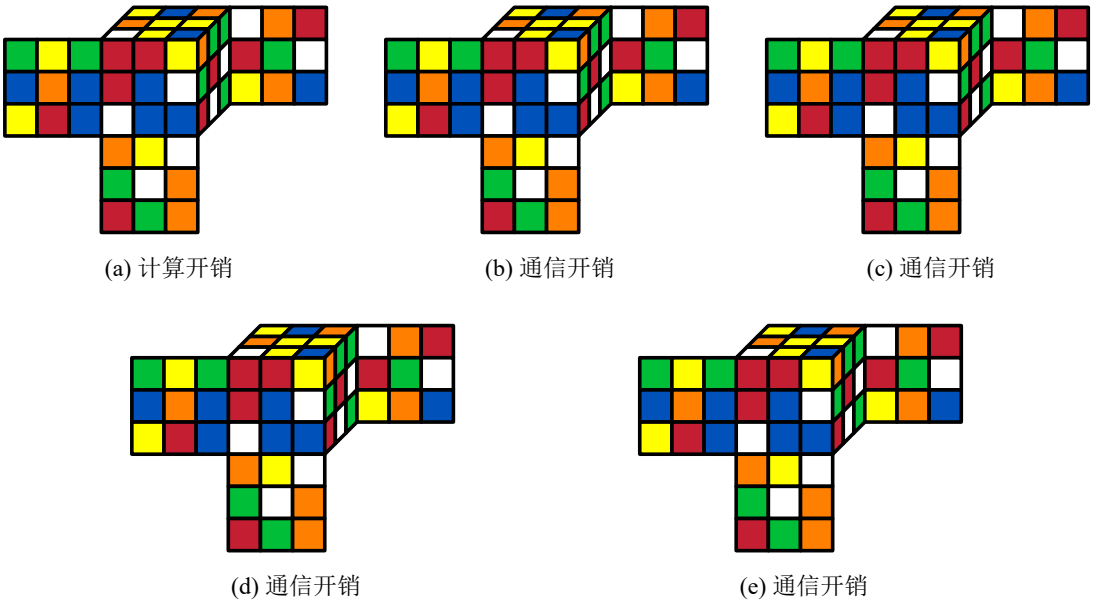
学校现建设有南北两个校区，总占地面积约 270 公顷，校舍建筑面积 130 多万平方米。图书馆馆藏文献约 1817 万册，其中纸质文献约 304 万册，电子文献约 1513 万册，内容覆盖了学校各个学科或专业。

截至 2020 年 11 月底，学校共有全日制在校生 36543 人，其中本科生 22439 人，硕士生 11448 人，博士生 2407 人；有在籍网络和函授教育本科生 43207 人，网络和函授教育专科生 53959 人。设有研究生院。设有通信工程学院、电子工程学院、计算机科学与技术学院（示范性软件学院）、机电工程学院、物理与光电工程学院、经济



(a) 计算开销　　(b) 通信开销

图 A.1　方案开销



(a) 计算开销　(b) 通信开销　(c) 通信开销

(d) 通信开销　(e) 通信开销

图 A.2　方案开销

与管理学院、数学与统计学院、人文学院、外国语学院、微电子学院、生命科学技术学院、空间科学与技术学院、先进材料与纳米科技学院、网络与信息安全学院、马克思主义学院、人工智能学院、网络与继续教育学院等 17 个学院。

学校是国内最早建立信息论、信息系统工程、雷达、微波天线、电子机械、电子对抗等专业的高校之一，开辟了我国 IT 学科的先河，形成了鲜明的电子与信息学科特色与优势。“十三五”期间，学校获批 8 个国防特色学科。学校现有 2 个国家“双一流”重点建设学科群（包含信息与通信工程、电子科学与技术、计算机科学与技术、网络空间安全、控制科学与工程 5 个一级学科），2 个国家一级重点学科（覆盖 6 个二级学科），1 个国家二级重点学科，34 个省部级重点学科，14 个博士学位授权一级学科，26 个硕士学位授权一级学科，10 个博士后科研流动站，65 个本科专业。全国第四轮一级学科评估结果中，3 个学科获评 A 类：电子科学与技术学科评估结果为 A+档，并列全国第 1；信息与通信工程学科位于 A 档；计算机科学与技术学科评估结果为 A-档，学校电子信息类学科继续保持国内领先水平。根据 ESI 公布数据，学校工程学和计算机科学均位列全球排名前 1‰。

学校树立了以人为本、教师是大学核心竞争力的理念，锻造了一支结构合理、富有创新精神的教师队伍。现有专任教师 2300 余名，其中，博士生导师 700 余人，硕士生导师 1500 余人。学校有院士 3 人，“万人计划”入选者 28 人，长江学者 36 人，国家自然科学基金创新研究群体 2 个，科技部重点创新团队 5 个，教育部创新团队 6 个，国家级教学名师 4 人，国家级教学团队 6 个，973 项目首席科学家 3 人，教育部新世纪优秀人才 51 人，“何梁何利”科学与技术奖获得者 4 人，教育部教学指导委员

会委员 19 人，享受政府特殊津贴 165 人。

学校不断地创新教育理念，深化教学内容、课程体系与实践教学改革，大力推进素质教育，取得了显着成果。现有国家级特色专业 14 个，国家级精品课程 13 门，国家级精品资源共享课 11 门，国家级视频公开课 3 门，国家精品在线开放课程 9 门，国家级一流本科课程 13 门，建设有 3 个国家人才培养及教学基地、6 个国家级实验教学示范中心、3 个国家级虚拟仿真实验中心，以及 3 个国家级人才培养模式创新实验区。学校人才培养素以理论基础扎实、工程实践能力突出、创新意识强等特色在全国高校中形成了“品牌”。学校坚持“因材施教、分类培养”的教育理念，积极探索实施“卓越工程师教育培养计划”、“钱学森空间科学实验班”和“科教结合协同育人行动计划”等一系列创新型人才培养模式改革。近五年来，学校本科生参与课外科技活动的普及率高，获得各类省级、国家级学科和科技竞赛奖 3600 余项。研究生和本科毕业生总体就业率一直保持在 96% 以上，位居全国高校前列。2006 年，学校顺利通过教育部本科教学工作水平评估并获得“优秀”；2020 年，学校获中国高校“就业最佳典范奖”。

A.2　单个图片

多年来，学校致力于电子信息技术领域的系统研制、科技攻关、工程研发等，创造了我国电子与信息技术领域等多项第一，包括第一台气象雷达、第一套流星余迹通讯系统、第一台可编程雷达信号处理机、第一台毫米波通讯机，以及我军通信装备史上第一部“塞绳电报互换机”、第一台“塔型管空腔振荡器”、第一套“三坐标相控阵雷达”等，为我国信息化、国防现代化做出了重要的贡献。学校现有 9 个国家级科技创新基地、1 个科工局科技创新基地，10 个教育部科技创新基地、29 个陕西省科技创新基地，2013 年入选国家级创新人才培养示范基地。先后牵头承担了“973”、“863”、重大专项、国家重点研发计划、国家自然科学重大项目、国家重大项目科研仪器研制项目等重大、重点项目，产生了一批标志性的研究成果。2013 年以来，学校科研指标稳步提升，在认知雷达、移动通讯、网络信息安全、高功率微波集成器件、智能计算、大型天线机电耦合等方面取得了卓有成效的成果，2012 年以来学校获国家科技奖励 21 项。2014 年，学校牵头的“信息感知技术协同创新中心”通过国家“2011 计划”认定，位列行业产业类第一，进一步奠定了学校在全国高校中突出的国防科研特色优势地位。

学校大力加强产学研相结合，不断增强科技创新能力。建设有中国西部军民融合创新谷暨西安电子谷、陕西工业研究院、国家大学科技园，同时与国内大型知名企事业单位联合建立股份制公司，成立战略联盟、设立企业基金、建立联合实验室及研究生实习基地，有力促进了科技成果的转化。学校积极开展国际国内的交流与合作，拓

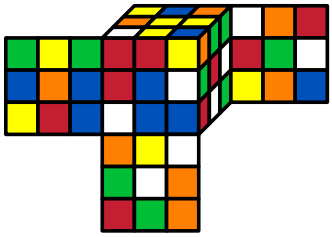


图 A.3　方案开销

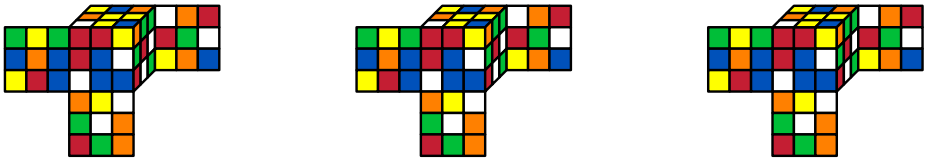


图 A.4　方案开销

展外部发展空间。学校先后与 35 个国家和地区的 155 所大学及研究机构建立友好关系，与 10 余个研究所、研究中心、企业集团建立了长期战略合作伙伴关系，与西安、广州、青岛、重庆等地方政府开展深入合作，共建研究院所、研究中心、新型研发机构，与跨国公司建立 66 个联合实验室，基本形成多方位、多层次、宽领域的对外合作创新发展格局。

A.3　多个图片非子图

建校 90 年来，学校先后为国家输送了 31 万余名电子信息领域的高级人才，产生了 120 多位解放军将领，成长起了 24 位院士（1977 年恢复高考以后院士校友 20 位，位列全国前茅），10 余位国家副部级以上领导，培养了联想创始人柳传志，国际 GSM 奖获得者李默芳，欧洲科学院院士、著名的纳米技术专家王中林，“天宫一号”目标飞行器总设计师杨宏等一大批 IT 行业领军人物和技术骨干、科研院所所长和大学校长等，为国家建设和社会进步做出了重要贡献。

在全面建设社会主义现代化国家新征程中，西安电子科技大学将继续坚持走内涵式发展道路，秉承“全心全意为人民服务”的办学宗旨，坚持“立足西部、育人育才、强军拓民、服务引领、团结实干”的发展思路，坚持立德树人根本任务，全面提升教育质量，为把学校建设成为电子信息特色鲜明的一流大学而不懈奋斗！

附录 B　表格示例

B.1　tabular

tabular 示例如表 **??** 和表 **??** 所示。

西安电子科技大学是以信息与电子学科为主，工、理、管、文多学科协调发展的全国重点大学，直属教育部，是国家“优势学科创新平台”项目和“211 工程”项目重点建设高校之一、国家双创示范基地之一、首批 35 所示范性软件学院、首批 9 所示范性微电子学院、首批 9 所获批设立集成电路人才培养基地和首批一流网络安全学院建设示范项目的高校之一。2017 年学校信息与通信工程、计算机科学与技术入选国家“双一流”建设学科。

学校前身是 1931 年诞生于江西瑞金的中央军委无线电学校，是毛泽东等老一辈革命家亲手创建的第一所工程技术学校。1958 年学校迁址西安，1966 年转为地方建制，1988 年定为现名。

建校 90 年来，学校始终得到了党和国家的高度重视，是我国“一五”重点建设的项目之一，也是 1959 年中央批准的全国 20 所重点大学之一。20 世纪 60 年代，学校就以“西军电”之称蜚声海内外。毛泽东同志曾先后两次为学校题词：“全心全意为人民服务”、“艰苦朴素”。

学校现建设有南北两个校区，总占地面积约 270 公顷，校舍建筑面积 130 多万平方米。图书馆馆藏文献约 1817 万册，其中纸质文献约 304 万册，电子文献约 1513 万册，内容覆盖了学校各个学科或专业。

截至 2020 年 11 月底，学校共有全日制在校生 36543 人，其中本科生 22439 人，硕士生 11448 人，博士生 2407 人；有在籍网络和函授教育本科生 43207 人，网络和函授教育专科生 53959 人。设有研究生院。设有通信工程学院、电子工程学院、计算机科学与技术学院（示范性软件学院）、机电工程学院、物理与光电工程学院、经济与管理学院、数学与统计学院、人文学院、外国语学院、微电子学院、生命科学技术

表 B.1　表格示例 1

	Numbers		
	1	2	3
Alphabet	A	B	C
Roman	I	II	III

表 B.2　表格示例 2

<table>
<tr><td>a</td><td colspan="2">b</td><td>c</td></tr>
<tr><td rowspan="2">a</td><td>e</td><td>f</td><td rowspan="2">c</td></tr>
<tr><td>e</td><td>f</td></tr>
<tr><td>a</td><td colspan="2">b</td><td>c</td></tr>
</table>

学院、空间科学与技术学院、先进材料与纳米科技学院、网络与信息安全学院、马克思主义学院、人工智能学院、网络与继续教育学院等 17 个学院。

学校是国内最早建立信息论、信息系统工程、雷达、微波天线、电子机械、电子对抗等专业的高校之一，开辟了我国 IT 学科的先河，形成了鲜明的电子与信息学科特色与优势。“十三五”期间，学校获批 8 个国防特色学科。学校现有 2 个国家“双一流”重点建设学科群（包含信息与通信工程、电子科学与技术、计算机科学与技术、网络空间安全、控制科学与工程 5 个一级学科），2 个国家一级重点学科（覆盖 6 个二级学科），1 个国家二级重点学科，34 个省部级重点学科，14 个博士学位授权一级学科，26 个硕士学位授权一级学科，10 个博士后科研流动站，65 个本科专业。全国第四轮一级学科评估结果中，3 个学科获评 A 类：电子科学与技术学科评估结果为 A+档，并列全国第 1；信息与通信工程学科位于 A 档；计算机科学与技术学科评估结果为 A-档，学校电子信息类学科继续保持国内领先水平。根据 ESI 公布数据，学校工程学和计算机科学均位列全球排名前 1‰。

学校树立了以人为本、教师是大学核心竞争力的理念，锻造了一支结构合理、富有创新精神的教师队伍。现有专任教师 2300 余名，其中，博士生导师 700 余人，硕士生导师 1500 余人。学校有院士 3 人，“万人计划”入选者 28 人，长江学者 36 人，国家自然科学基金创新研究群体 2 个，科技部重点创新团队 5 个，教育部创新团队 6 个，国家级教学名师 4 人，国家级教学团队 6 个，973 项目首席科学家 3 人，教育部新世纪优秀人才 51 人，“何梁何利”科学与技术奖获得者 4 人，教育部教学指导委员会委员 19 人，享受政府特殊津贴 165 人。

B.2　tabularx

tabularx 示例如表 **??** 所示。

学校不断地创新教育理念，深化教学内容、课程体系与实践教学改革，大力推进素质教育，取得了显着成果。现有国家级特色专业 14 个，国家级精品课程 13 门，国家级精品资源共享课 11 门，国家级视频公开课 3 门，国家精品在线开放课程 9 门，国家级一流本科课程 13 门，建设有 3 个国家人才培养及教学基地、6 个国家级实验教学示范中心、3 个国家级虚拟仿真实验中心，以及 3 个国家级人才培养模式创新实验

表 B.3　表格示例 3

表格测试数据	表格测试数据表格测试数据表格测试数据	表格测试数据	表格测试数据表格测试数据表格测试数据
表格测试	表格测试数据表格测试数据表格测试数据	表格测试数据	表格测试数据表格测试数据表格测试数据
表格测试数据	表格测试数据表格测试数据	表格测试	表格测试数据表格测试数据表格测试数据

区。学校人才培养素以理论基础扎实、工程实践能力突出、创新意识强等特色在全国高校中形成了“品牌”。学校坚持“因材施教、分类培养”的教育理念，积极探索实施“卓越工程师教育培养计划”、“钱学森空间科学实验班”和“科教结合协同育人行动计划”等一系列创新型人才培养模式改革。近五年来，学校本科生参与课外科技活动的普及率高，获得各类省级、国家级学科和科技竞赛奖 3600 余项。研究生和本科毕业生总体就业率一直保持在 96% 以上，位居全国高校前列。2006 年，学校顺利通过教育部本科教学工作水平评估并获得“优秀”；2020 年，学校获中国高校“就业最佳典范奖”。

多年来，学校致力于电子信息技术领域的系统研制、科技攻关、工程研发等，创造了我国电子与信息技术领域等多项第一，包括第一台气象雷达、第一套流星余迹通讯系统、第一台可编程雷达信号处理机、第一台毫米波通讯机，以及我军通信装备史上第一部“塞绳电报互换机”、第一台“塔型管空腔振荡器”、第一套“三坐标相控阵雷达”等，为我国信息化、国防现代化做出了重要的贡献。学校现有 9 个国家级科技创新基地、1 个科工局科技创新基地，10 个教育部科技创新基地、29 个陕西省科技创新基地，2013 年入选国家级创新人才培养示范基地。先后牵头承担了“973”、“863”、重大专项、国家重点研发计划、国家自然科学重大项目、国家重大项目科研仪器研制项目等重大、重点项目，产生了一批标志性的研究成果。2013 年以来，学校科研指标稳步提升，在认知雷达、移动通讯、网络信息安全、高功率微波集成器件、智能计算、大型天线机电耦合等方面取得了卓有成效的成果，2012 年以来学校获国家科技奖励 21 项。2014 年，学校牵头的“信息感知技术协同创新中心”通过国家“2011 计划”认定，位列行业产业类第一，进一步奠定了学校在全国高校中突出的国防科研特色优势地位。

学校大力加强产学研相结合，不断增强科技创新能力。建设有中国西部军民融合创新谷暨西安电子谷、陕西工业研究院、国家大学科技园，同时与国内大型知名企事业单位联合建立股份制公司，成立战略联盟、设立企业基金、建立联合实验室及研究生实习基地，有力促进了科技成果的转化。学校积极开展国际国内的交流与合作，拓

表 B.5　表格示例 4

表格测试数据	表格测试数据表格测试数据表格测试数据	表格测试数据	表格测试数据表格测试数据表格测试数据
表格测试	表格测试数据表格测试数据表格测试数据	表格测试数据	表格测试数据表格测试数据表格测试数据
表格测试数据	表格测试数据表格测试数据	表格测试	表格测试数据表格测试数据表格测试数据

展外部发展空间。学校先后与 35 个国家和地区的 155 所大学及研究机构建立友好关系，与 10 余个研究所、研究中心、企业集团建立了长期战略合作伙伴关系，与西安、广州、青岛、重庆等地方政府开展深入合作，共建研究院所、研究中心、新型研发机构，与跨国公司建立 66 个联合实验室，基本形成多方位、多层次、宽领域的对外合作创新发展格局。

B.3　tabulary

tabulary 示例如表 **??** 所示。

建校 90 年来，学校先后为国家输送了 31 万余名电子信息领域的高级人才，产生了 120 多位解放军将领，成长起了 24 位院士（1977 年恢复高考以后院士校友 20 位，位列全国前茅），10 余位国家副部级以上领导，培养了联想创始人柳传志，国际 GSM 奖获得者李默芳，欧洲科学院院士、著名的纳米技术专家王中林，“天宫一号”目标飞行器总设计师杨宏等一大批 IT 行业领军人物和技术骨干、科研院所所长和大学校长等，为国家建设和社会进步做出了重要贡献。

在全面建设社会主义现代化国家新征程中，西安电子科技大学将继续坚持走内涵式发展道路，秉承“全心全意为人民服务”的办学宗旨，坚持“立足西部、育人育才、强军拓民、服务引领、团结实干”的发展思路，坚持立德树人根本任务，全面提升教育质量，为把学校建设成为电子信息特色鲜明的一流大学而不懈奋斗！

附录 C　算法示例

西安电子科技大学是以信息与电子学科为主，工、理、管、文多学科协调发展的全国重点大学，直属教育部，是国家“优势学科创新平台”项目和“211 工程”项目重点建设高校之一、国家双创示范基地之一、首批 35 所示范性软件学院、首批 9 所示范性微电子学院、首批 9 所获批设立集成电路人才培养基地和首批一流网络安全学院建设示范项目的高校之一。2017 年学校信息与通信工程、计算机科学与技术入选国家“双一流”建设学科。

C.1　示例 1

学校前身是 1931 年诞生于江西瑞金的中央军委无线电学校，是毛泽东等老一辈革命家亲手创建的第一所工程技术学校。1958 年学校迁址西安，1966 年转为地方建制，1988 年定为现名。

建校 90 年来，学校始终得到了党和国家的高度重视，是我国“一五”重点建设的项目之一，也是 1959 年中央批准的全国 20 所重点大学之一。20 世纪 60 年代，学校就以“西军电”之称蜚声海内外。毛泽东同志曾先后两次为学校题词：“全心全意为人民服务”、“艰苦朴素”。

学校现建设有南北两个校区，总占地面积约 270 公顷，校舍建筑面积 130 多万平方米。图书馆馆藏文献约 1817 万册，其中纸质文献约 304 万册，电子文献约 1513 万册，内容覆盖了学校各个学科或专业。

截至 2020 年 11 月底，学校共有全日制在校生 36543 人，其中本科生 22439 人，硕士生 11448 人，博士生 2407 人；有在籍网络和函授教育本科生 43207 人，网络和函授教育专科生 53959 人。设有研究生院。设有通信工程学院、电子工程学院、计算机科学与技术学院（示范性软件学院）、机电工程学院、物理与光电工程学院、经济与管理学院、数学与统计学院、人文学院、外国语学院、微电子学院、生命科学技术学院、空间科学与技术学院、先进材料与纳米科技学院、网络与信息安全学院、马克思主义学院、人工智能学院、网络与继续教育学院等 17 个学院。

具体的内容如算法 **??** 中第 **??** 行所示。具体的内容如算法 **??** 中第 **??** 行至第 **??** 行所示。

学校是国内最早建立信息论、信息系统工程、雷达、微波天线、电子机械、电子对抗等专业的高校之一，开辟了我国 IT 学科的先河，形成了鲜明的电子与信息学科特色与优势。“十三五”期间，学校获批 8 个国防特色学科。学校现有 2 个国家“双一

算法 C.1 The Bellman-Kalaba algorithm

```
1: procedure BellmanKalaba(G, u, l, p)
2:     for all v ∈ V(G) do
3:         l(v) ← ∞
4:     end for
5:     l(u) ← 0
6:     repeat
7:         for i ← 1, n do
8:             min ← l(v_i)
9:             for j ← 1, n do
10:                if min > e(v_i, v_j) + l(v_j) then
11:                    min ← e(v_i, v_j) + l(v_j)
12:                    p(i) ← v_j
13:                end if
14:            end for
15:            l'(i) ← min
16:        end for
17:        changed ← l ≠ l'
18:        l ← l'
19:    until ¬changed
20: end procedure

21: procedure FindPathBK(v, u, p)
22:    if v = u then
23:        Write v
24:    else
25:        w ← v
26:        while w ≠ u do
27:            Write w
28:            w ← p(w)
29:        end while
30:    end if
31: end procedure
```

流”重点建设学科群（包含信息与通信工程、电子科学与技术、计算机科学与技术、网络空间安全、控制科学与工程 5 个一级学科），2 个国家一级重点学科（覆盖 6 个二级学科），1 个国家二级重点学科，34 个省部级重点学科，14 个博士学位授权一级学科，26 个硕士学位授权一级学科，10 个博士后科研流动站，65 个本科专业。全国第四轮一级学科评估结果中，3 个学科获评 A 类：电子科学与技术学科评估结果为 A+档，并列全国第 1；信息与通信工程学科位于 A 档；计算机科学与技术学科评估结果为 A-档，学校电子信息类学科继续保持国内领先水平。根据 ESI 公布数据，学校工程学和计算机科学均位列全球排名前 1‰。

学校树立了以人为本、教师是大学核心竞争力的理念，锻造了一支结构合理、富

有创新精神的教师队伍。现有专任教师 2300 余名，其中，博士生导师 700 余人，硕士生导师 1500 余人。学校有院士 3 人，“万人计划”入选者 28 人，长江学者 36 人，国家自然科学基金创新研究群体 2 个，科技部重点创新团队 5 个，教育部创新团队 6 个，国家级教学名师 4 人，国家级教学团队 6 个，973 项目首席科学家 3 人，教育部新世纪优秀人才 51 人，“何梁何利”科学与技术奖获得者 4 人，教育部教学指导委员会委员 19 人，享受政府特殊津贴 165 人。

算法 C.2

输入:
输出: $\rho(S)$.

学校不断地创新教育理念，深化教学内容、课程体系与实践教学改革，大力推进素质教育，取得了显着成果。现有国家级特色专业 14 个，国家级精品课程 13 门，国家级精品资源共享课 11 门，国家级视频公开课 3 门，国家精品在线开放课程 9 门，国家级一流本科课程 13 门，建设有 3 个国家人才培养及教学基地、6 个国家级实验教学示范中心、3 个国家级虚拟仿真实验中心，以及 3 个国家级人才培养模式创新实验区。学校人才培养素以理论基础扎实、工程实践能力突出、创新意识强等特色在全国高校中形成了“品牌”。学校坚持“因材施教、分类培养”的教育理念，积极探索实施“卓越工程师教育培养计划”、“钱学森空间科学实验班”和“科教结合协同育人行动计划”等一系列创新型人才培养模式改革。近五年来，学校本科生参与课外科技活动的普及率高，获得各类省级、国家级学科和科技竞赛奖 3600 余项。研究生和本科毕业生总体就业率一直保持在 96% 以上，位居全国高校前列。2006 年，学校顺利通过教育部本科教学工作水平评估并获得“优秀”；2020 年，学校获中国高校“就业最佳典范奖”。

C.2　示例 2

多年来，学校致力于电子信息技术领域的系统研制、科技攻关、工程研发等，创造了我国电子与信息技术领域等多项第一，包括第一台气象雷达、第一套流星余迹通讯系统、第一台可编程雷达信号处理机、第一台毫米波通讯机，以及我军通信装备史上第一部“塞绳电报互换机”、第一台“塔型管空腔振荡器”、第一套“三坐标相控阵雷达”等，为我国信息化、国防现代化做出了重要的贡献。学校现有 9 个国家级科技创新基地、1 个科工局科技创新基地，10 个教育部科技创新基地、29 个陕西省科技创新基地，2013 年入选国家级创新人才培养示范基地。先后牵头承担了“973”、“863”、重大专项、国家重点研发计划、国家自然科学重大项目、国家重大项目科研仪器研制项目等重大、重点项目，产生了一批标志性的研究成果。2013 年以来，学校科研指

标稳步提升，在认知雷达、移动通讯、网络信息安全、高功率微波集成器件、智能计算、大型天线机电耦合等方面取得了卓有成效的成果，2012 年以来学校获国家科技奖励 21 项。2014 年，学校牵头的“信息感知技术协同创新中心”通过国家“2011 计划”认定，位列行业产业类第一，进一步奠定了学校在全国高校中突出的国防科研特色优势地位。

算法 C.3 Euclid's algorithm

```
1: procedure EUCLID(a, b)                          ▷ The g.c.d. of a and b
2:     r ← a mod b
3:     while r ≠ 0 do                              ▷ We have the answer if r is 0
4:         a ← b
5:         b ← r
6:         r ← a mod b
7:     end while
8:     return b                                    ▷ The gcd is b
9: end procedure
```

学校大力加强产学研相结合，不断增强科技创新能力。建设有中国西部军民融合创新谷暨西安电子谷、陕西工业研究院、国家大学科技园，同时与国内大型知名企事业单位联合建立股份制公司，成立战略联盟、设立企业基金、建立联合实验室及研究生实习基地，有力促进了科技成果的转化。学校积极开展国际国内的交流与合作，拓展外部发展空间。学校先后与 35 个国家和地区的 155 所大学及研究机构建立友好关系，与 10 余个研究所、研究中心、企业集团建立了长期战略合作伙伴关系，与西安、广州、青岛、重庆等地方政府开展深入合作，共建研究院所、研究中心、新型研发机构，与跨国公司建立 66 个联合实验室，基本形成多方位、多层次、宽领域的对外合作创新发展格局。

建校 90 年来，学校先后为国家输送了 31 万余名电子信息领域的高级人才，产生了 120 多位解放军将领，成长起了 24 位院士（1977 年恢复高考以后院士校友 20 位，位列全国前茅），10 余位国家副部级以上领导，培养了联想创始人柳传志，国际 GSM 奖获得者李默芳，欧洲科学院院士、著名的纳米技术专家王中林，“天宫一号”目标飞行器总设计师杨宏等一大批 IT 行业领军人物和技术骨干、科研院所所长和大学校长等，为国家建设和社会进步做出了重要贡献。

在全面建设社会主义现代化国家新征程中，西安电子科技大学将继续坚持走内涵式发展道路，秉承“全心全意为人民服务”的办学宗旨，坚持“立足西部、育人育才、强军拓民、服务引领、团结实干”的发展思路，坚持立德树人根本任务，全面提升教育质量，为把学校建设成为电子信息特色鲜明的一流大学而不懈奋斗！

@ARTICLE24143, author=Murata, T., journal=Proceedings of the IEEE, title=Petri nets: Properties, analysis and applications, year=1989, volume=77, number=4, pages=541-580, doi=10.1109/5.24143

致谢

本论文是在导师的悉心指导下完成的，从论文的选题到论文的撰写，无不渗透着导师的心血，……值此论文完稿之际，谨对导师的辛勤培育以及谆谆教诲表示最衷心的感谢!

作者简介

1. 基本情况

张三，男，陕西西安人，1982 年 8 月出生，西安电子科技大学电子工程学院电磁场与微波技术专业 2008 级硕士研究生。

2. 教育背景

2001.08～2005.07，西安电子科技大学，本科，专业：电子信息工程

2008.08～　　，西安电子科技大学，硕士研究生，专业：电磁场与微波技术

3. 攻读硕士学位期间的研究成果

3.1 发表学术论文

[1] XXX, XXX, XXX . Rapid development technique for drip irrigation emitters[J].RP Journal,UK.,2003,9(2): 104-110.(SCI: 672CZ, EI: 03187452127)

[2] XXX, XXX, XXX . 基于快速成型制造的滴管快速制造技术研究 [J]. 西安交通大学学报, 2001, 15(9): 935-939. (EI: 02226959521)

[3] …

3.2 申请（授权）专利

[1] XXX, XXX, XXX 等. 专利名称: 国别, 专利号 [P]. 出版日期.

[2] …

3.3 参与科研项目及获奖

[1] XXX 项目, 项目名称, 起止时间, 完成情况, 作者贡献。

[2] XXX, XXX, XXX 等. 科研项目名称. 陕西省科技进步三等奖, 获奖日期.

[3] …