

# 作业车间调度理论及其 优化方法研究

王永明 尹红丽 秦开大 著



科学出版社

# 作业车间调度理论及其 优化方法研究

王永明 尹红丽 秦开大 著

科学出版社

北京

## 内 容 简 介

作业车间调度问题是现代生产系统的关键问题之一。因此,作业车间调度理论及其优化方法的研究得到了学术界和企业界的广泛关注。本书是一本研究生产调度原理和算法的专业著作,其内容涵盖作业车间调度的基本理论及关键问题,主题集中于:具有复合邻域结构的禁忌搜索方法研究和设计、两个特定机器和工作环境下的调度问题研究及其解决方法、适合车间调度遗传算法的三维染色体编码研究、遗传算法控制参数的优化选取、标准调度问题及其复杂度等。

本书可作为计算机、自动化、应用数学、运筹管理等领域的研究生及教学科研的人员的参考用书,也可以作为从事生产调度相关工作的人员的参考资料。

### 图书在版编目(CIP)数据

---

作业车间调度理论及其优化方法研究/王永明,尹红丽,秦开大著.—北京:科学出版社,2013

ISBN 978-7-03-037353-3

I. ①作… II. ①王…②尹…③秦… III. ①企业管理-生产调度-系统最优化 IV. ①F273

中国版本图书馆 CIP 数据核字 (2013) 第 081520 号

---

责任编辑:李晓迎/责任校对:纪振红  
责任印制:徐晓晨/封面设计:蓝正设计

**科学出版社**出版

北京东黄城根北街16号

邮政编码:100717

<http://www.sciencep.com>

**中国科学院印刷厂**印刷

科学出版社发行 各地新华书店经销

\*

2013年4月第 一 版 开本:720×1000 B5

2013年4月第一次印刷 印张:10

字数:210 000

**定价:48.00元**

(如有印装质量问题,我社负责调换)

## 前 言

调度是决策的一种形式，它在制造业和服务业中扮演着关键角色。生产调度是实现制造业运筹、管理与优化技术的核心，它在时间上对一组可用的制造资源进行加工任务的安排，将工件分配至相应的机器上，确定各机器上操作的加工次序和开工时间，从而使得某一性能指标达到最优。生产调度问题可以抽象地描述为在一些等式或不等式约束构成的离散解空间中寻找目标函数的最优解。它是一类重要的组合优化问题，运筹学称之为排序问题。除少数特殊问题外，大部分生产调度问题的计算复杂度为 NP-hard。有效的优化调度能使生产和商业领域增加产出、减少周转时间、减少库存，最终减少生产费用、增加利润和提升客户满意度。

作业车间调度问题在所有生产调度中是最复杂、最困难也是更具一般性的问题。本书在对作业车间调度问题进行细致分析并建立数学模型的基础上，研究了作业车间调度的若干关键问题。其相关工作主要集中在：具有复合邻域结构的禁忌搜索方法研究和设计、两个特定机器和工作环境下的调度问题研究及其解决方法、符合车间调度遗传算法的三维染色体编码研究、遗传算法控制参数的优化选取等。旨在夯实本领域的研究基础，为求解作业车间调度问题提供新的有效解决思路，开辟新的研究方向。所完成的工作和取得的成果具体如下：

(1) 对生产调度问题的定义与描述作了重点探讨，结合 Graham 等人提出的生产调度问题的分类和描述法，给出了生产调度问题的三元组  $\alpha | \beta | \gamma$  描述方法，这种方法使复杂的生产调度问题的描述变得条理化、简约化。

(2) Van Laarhoven 等人提出的邻域结构具有连通性，但解的数量非常大，而 Nowichi 等人设计的邻域结构具有邻域规模小的特点，但不具有连通性。本书结合以上两种邻域结构的优势，同时避免其不足，提出了一种基于具有复合邻域结构的禁忌搜索算法，并在证明了复合邻域结构具有连通性的基础上，对算法的有效性作了多方面的验证。

(3) 对于无等待作业车间调度问题  $J | \text{no-wait} | C_{\max}$ ，首先利用三划分问题的强 NP-hard 特性间接证明了本问题的复杂性，然后提出了一种正反排时算法来解决无等待作业车间调度问题的两个子问题之一，并结合解决子问题二的  $K$ -邻域禁忌搜索来使问题  $J | \text{no-wait} | C_{\max}$  得到解决。其中的正反排时算法开辟了新的解空间，是非延时的排时算法无法达到的。

(4) 在通常的作业车间调度问题中引入了加工顺序决定的准备时间和每个操

作都有速度不同的多个并行机器可以备选两个约束后，虽然调度目标也是最小化最大制造期，但却没有可行的求解方法。在建立了此问题的混合整数规划模型并研究其调度解编码方法后，本书提出了一种两阶段禁忌搜索算法来求解多约束作业车间调度，该方法可以同时搜索最好的加工顺序安排和加工每个操作的最佳机器。这是一个很接近现实生产环境的调度问题，但目前对其的研究罕见。

(5) 在使用迭代进化算法解决调度问题时，编码方法是关键，不同的编码需要设计不同的进化算子与之相适应，它们对整个算法的有效性和效率有重大影响。结合已有的编码方法和作业车间调度的特点，本书提出了一种三维染色体编码方法及与之相对应的遗传进化算子，并与传统的算法进行比较，验证了其有效性和正确性。

(6) 对于遗传算法 (genetic algorithm, GA)、模拟退火 (simulated annealing)、禁忌搜索 (tabu search) 等具有通用性的算法，在具体运用于实际问题时怎样根据问题的情境设置最优的算法以控制参数，一直是个公开的问题。当使用遗传算法来解决大规模作业车间调度问题时，本书提出了基于量优化分配算法的比较方法来对候选的算法以控制参数进行优化决策，选出针对具体问题的最佳算法以控制参数，解决算法参数设置的盲目性问题。

感谢陈剑教授、汪万良教授、王凌教授、刘民教授、梁樑教授等，在与他们及其团队成员交流的过程中学到了很多。他们在优化调度领域的多年研究经验给了我巨大的帮助和启示。

感谢纽约大学的 Michael Pinedo 教授、那不勒斯大学的 Roberto Mcchiaroli 教授、卡内基梅隆大学的 Matt Streeter 教授、普渡大学的 Reha Uzsoy 教授、香港理工大学的 T. C. E Cheng 教授等，是他们帮我找到了许多珍贵的资料，并不远万里给我邮寄了很多硬拷贝材料。

本书的出版得到了国家自然科学基金 (编号: 70961003, 71062006, 71162019, 71262029)、中国博士后科学研究基金 (编号: 20110491761)、省级应用基础研究基金 (编号: 2008ZC018M) 及教育厅研究基金重点项目 (编号: 2001z061) 的资助。同时也得到民族教育信息化教育部重点实验室的支持。

王永明 尹红丽 秦开大

2013年1月

# 目 录

## 前言

<b>第 1 章 导论</b> .....	1
1.1 概述 .....	1
1.2 调度问题的描述 .....	1
1.3 调度问题及解的分类 .....	6
1.4 本书主要内容 .....	9
<b>第 2 章 作业车间调度问题研究</b> .....	11
2.1 作业车间调度问题的定义及数学描述.....	11
2.2 作业车间调度问题的复杂性.....	15
2.3 作业车间调度问题的求解方法.....	16
2.4 作业车间调度的研究现状.....	22
<b>第 3 章 禁忌搜索算法研究</b> .....	26
3.1 局部搜索算法.....	26
3.2 禁忌搜索算法研究.....	32
<b>第 4 章 作业车间调度与禁忌搜索算法</b> .....	41
4.1 调度解中的关键路径及块.....	41
4.2 邻域函数研究.....	43
4.3 禁忌搜索中的初始解.....	46
4.4 禁忌搜索算法实例分析.....	47
<b>第 5 章 基于复合邻域结构的禁忌搜索算法</b> .....	51
5.1 禁忌搜索复合邻域结构研究.....	51
5.2 基于复合邻域结构的禁忌搜索算法的基本流程.....	52
5.3 算例实验分析.....	53
<b>第 6 章 基于禁忌搜索的无等待作业车间调度</b> .....	57
6.1 无等待作业车间调度建模.....	57
6.2 无等待作业车间调度问题的分解及复杂度.....	58
6.3 基于禁忌搜索的无等待作业车间调度问题.....	61
6.4 算例实验分析.....	66
<b>第 7 章 基于禁忌搜索的多约束作业车间调度</b> .....	69
7.1 多约束作业车间调度.....	69

---

7.2	多约束作业车间调度建模	69
7.3	多约束作业车间调度编码与算法研究	71
7.4	算例实验分析	74
<b>第 8 章</b>	<b>基于遗传进化算法的作业车间调度</b>	<b>77</b>
8.1	遗传进化算法	77
8.2	作业车间调度与遗传算法	84
8.3	基于三维染色体编码的作业车间调度	89
8.4	算例实验分析	93
<b>第 9 章</b>	<b>基于计算量优化分配的算法参数决策</b>	<b>97</b>
9.1	问题描述和建模	97
9.2	遗传算法比较和选择方法	98
9.3	基于计算量优化分配的遗传算法的比较方法	100
9.4	算例实验分析	101
<b>第 10 章</b>	<b>作业车间调度标准问题及其复杂度</b>	<b>105</b>
10.1	作业车间调度标准问题	105
10.2	作业车间调度问题的复杂度	136
	<b>参考文献</b>	<b>139</b>

# 第 1 章 导 论

## 1.1 概 述

随着我国从粗放型生产向集约型生产的转变,最近几年,生产调度问题在国内的研究得到了前所未有的重视。所谓调度,就是为了实现某一目的而对共同使用的资源实行时间分配<sup>[1, 2]</sup>。它是一个决策的过程,其目的是优化一个或多个目标。调度问题来源于不同的领域,如医院护理、生产计划、计算机设计、军队作战、交通运输、后勤及通信等。一个组织中的资源和任务会呈现多种形式。车间里的机器、机场的跑道、建筑场所的工作团队、计算环境中的处理单元都可以看成资源。任务则可能是生产过程中的操作、机场飞机的起飞和着陆、建筑项目的各个阶段、计算机程序的执行等。每个任务可能有某种优先级水平、最早可能开始的时间和工期。目标也会呈现出多种形式。一个目标可能是最小化最后一项任务的完成时间,而另一个目标可能是最小化超过工期才完成的任务数量。

生产调度是实现制造业运筹、管理与优化技术的核心。它是在时间上对一组可用的制造资源,即机器进行加工任务的安排,将工件分配至相应的机器上,确定各机器上加工操作的加工次序和开工时间,使得某一性能指标最优。在执行这些加工任务时需要满足某些物理约束,如工件的加工顺序约束、机器加工能力限制等。同时也要满足某些非物理约束,如工件到达时间约束、客户交货期限需求等。因此,生产调度问题可表述为在一些等式或不等式约束构成的离散解空间中,寻找目标函数值的最优解。它是一类重要的组合优化问题,运筹学称之为排序问题<sup>[3, 4]</sup>。除少数特殊问题,大部分生产调度问题的计算复杂度为 NP-hard,存在优化求解的困难<sup>[5]</sup>。有效的调度优化算法能使现代商业领域增加产出、减少周转时间、减少库存,最终减少生产费用、增加利润和提升客户满意度。因此,调度优化算法性能的好坏对这些行业的高效运作有重要影响,其研究也具有重要的理论意义和实用价值。

## 1.2 调度问题的描述

生产调度问题是在可用机器资源上对工件进行时间安排,使得某一或某些调度优化准则最优。因此,机器、工件和优化准则是生产调度问题的三个基本要素。Graham 等在 1979 年提出生产调度问题分类和描述的三参数表述法<sup>[6, 7]</sup>,即

一个调度问题用三元组  $\alpha | \beta | \gamma$  来描述, 其中,  $\alpha$  域描述机器环境, 它只包含单一的一项;  $\beta$  域提供加工特征和约束的细节, 可能不包含任何一项, 也可能包含多个选项;  $\gamma$  域描述调度的优化目标, 通常情况下只包含一项。

(1) 参数域  $\alpha$  表示机器加工环境, 主要的机器环境描述如下。

单机 (1), 即 single machine。单机情况是所有机器环境中最简单的一种情况, 也是所有复杂机器环境中的一种特殊情况。

并行同速机 (Pm), 即 identical machines in parallel, 是指  $m$  台具有相同加工速度的同型号并行机。工作  $j$  需要单一操作, 可能在  $m$  个机器的任何一台上加工, 也可能在属于某个给定子集的任何一台机器上进行加工。如果不允许工作  $j$  仅简单地在一台机器上加工, 而必须在属于某个特定子集  $M_j$  的一台机器上加工, 那么  $M_j$  就出现在  $\beta$  域中。

并行异速机 (Qm), 即 machines in parallel with different speeds, 是指  $m$  台具有不同加工速度的机器。机器  $i$  的速度记作  $v_i$ 。假设工作  $j$  所有的加工都在机器  $i$  上进行, 那么工作  $j$  在机器  $i$  上耗费的时间  $p_{ij} = p_j / v_i$ 。如果所有的机器具有相同的速度, 即对所有的  $i$  有  $v_i = 1$  及  $p_{ij} = p_j$ , 那么这种环境和并行同速机环境相同。

并行无关联异速机 (Rm), 即 unrelated machines in parallel, 是指  $m$  台随工作不同加工速度也不同的异速机。这种环境是并行异速机环境的一般化。机器  $i$  的加工工作  $j$  的速度记作  $v_{ij}$ 。假设工作  $j$  所有的加工都在机器  $i$  上进行, 那么工作  $j$  在机器  $i$  上耗费的时间  $p_{ij} = p_j / v_{ij}$ 。如果机器的速度独立于工作, 即对所有的  $j$  有  $v_{ij} = v_i$ , 那么这种环境和并行异速机环境相同。

流水作业车间 (Fm), 即 flow shop。有  $m$  台串联机, 每项工作必须按相同的工艺路径依次在每台机器上加工一次。如果机器前工件队列按先进先出 (first in first out, FIFO) 规则依次通过机器, 该流水作业车间称为排列流水作业车间 (permutation flow shop), 在参数域  $\beta$  中包含排列 (prmu)。

柔性流水作业车间 (FFc), 即 flexible flow shop。它是流水作业车间和并行机环境的综合和一般化, 即  $c$  个串联的加工中心, 工件按相同的工艺路径依次在每个加工中心上加工一次, 每个加工中心有一组同型并行机。

作业车间 (Jm), 即 job shop。在有  $m$  台机器的加工车间里, 每项工作都有预先确定的加工路径, 每项工作最多访问一台机器一次。如果每项工作可以在同一机器上加工多次, 此时在参数域  $\beta$  中包含再循环 (recrc)。

柔性作业车间 (FJc), 即 flexible job shop, 柔性作业车间是作业车间和并行机环境的综合和一般化。代替  $m$  台机器的是  $c$  个加工中心, 每个加工中心有许多并行同速机。每项工作按各自给定的工艺路径通过车间, 工作  $j$  只需要在每个工作中心的一台机器上加工, 而每台机器都可以完成该工作。如果一项工作需

要在它经过车间的加工路径上访问一个工作中心多次，那么在参数域  $\beta$  中包含  $\text{recre}$ 。

开放车间 (Om)，即 open shop，开放车间有  $m$  台机器，其中工件可按任意路径在每台机器上加工一次。

(2) 参数域  $\beta$  表示加工约束和特定的限制，可能包含多个选项。下面给出参数域  $\beta$  中一些主要的约束描述。

到达时间  $r_i$ ，即 release dates。如果这个符号出现在  $\beta$  域中，表示工件  $i$  不能在到达时间  $r_i$  之前开始加工。如果  $r_i$  没有出现在  $\beta$  域中，则工作  $j$  的加工可以在任何时间开始。与到达时间不同，工作的工期不在该域中规定。目标函数的类型给出了是否有工期的要求。

顺序决定的准备时间  $s_{jk}$ ，即 sequence dependent setup times。 $s_{jk}$  表示工作  $j$  和工作  $k$  之间的准备时间，该时间长短与工作的加工顺序有关。如果工作  $k$  是第一个，则  $s_{0k}$  表示工作  $k$  的准备时间；如果工作  $j$  是最后一个，则  $s_{j0}$  表示工作  $j$  之后的清理时间。如果工作  $j$  和工作  $k$  之间的准备时间依赖于机器，那么将包含下标  $i$ ，即  $s_{ijk}$ 。如果  $\beta$  域中没有出现  $s_{jk}$ ，所有的准备时间都假设为零或者与顺序无关，这时，准备时间包含在加工时间中。

抢占 (prmp)，即 preemption。它表示一个工件若在一台机器上开始加工，在它完工之前可以允许停止加工此工件，而加工另一工件。被抢占的工件已经进行的加工不会丢失，当它重新返回到机器上时，只需要加工剩下的操作。当允许抢占时，prmp 出现在  $\beta$  域中；当 prmp 不在  $\beta$  域中时，不允许抢占。

排列 (prmu)，即 permutaion。该约束可能出现在 flow shop 环境，它表示 flow shop 上所有机器上工件的加工顺序相同。

重入 (reentr)，即 reentrant。重入情况一般出现在作业车间调度环境或柔性作业车间调度环境，它表示一台机器在同一机器或加工中心上加工多次。

优先约束 (prec)，即 precedence constraints。它可能出现在单机或并行机环境中，在一项工作开始加工前，另一项工作必须先完成。有几种优先约束的特殊形式，即如果每项工作最多有一个前驱，最多有一个后继，该约束称为链式；如果每项工作最多有一个后继，该约束称为入树；如果每项工作最多包含一个前驱，那么该约束称为出树。如果  $\beta$  域中没有出现 prec，则不受优先限制。

故障 (brkdwn)，即 breakdowns。机器故障意味着机器不能连续使用。例如，在并行机环境中，在任何时间点，可用机器的数量是时间的函数。

机器适用限制  $M_j$ ，即 machine eligibility restrictions。当加工环境是  $m$  台并行机器的时候，符号  $M_j$  会出现在  $\beta$  域中，当  $M_j$  出现时，意味着并不是所有  $m$  台机器都有能力加工工作  $j$ 。集合  $M_j$  表示可以加工几个工作  $j$  的机器集合。如果  $\beta$  域中不包含  $M_j$ ，则工作  $j$  可以在任何一台机器上加工。

阻塞 (block)，即 blocking。阻塞是可能出现在流水车间中的现象。如果一个流水车间在两台连续的机器之间只有有限的缓冲区，那么可能会出现当缓冲区满的时候，上游的机器不允许释放已经加工好的工作。这时加工好的工作不得不保留在上游机器上，从而阻塞那台机器加工别的工作。

无等待 (nwt)，即 no-wait。无等待需求是在流水车间发生最多的一种情况。工作不允许在两台连续的机器间等待。这意味着一项工作在第一台机器上的开始时间不得不推迟，以保证该工作可以不必等待任何机器而通过生产车间。这种操作的一个例子是在轧钢厂中，不允许钢板等待，因为在等待的过程中它会冷却。

再循环 (recrc)，即 recirculation。在作业加工车间或者柔性作业加工车间中，当一项工作可能访问一台机器或加工中心超过一次的时候，recrc 就出现在  $\beta$  域中。

其他任何出现在  $\beta$  域中的项都是很容易理解的。例如， $P_j = p$  意味着所有的加工时间都是相等的，而  $d_j = a$  意味着所有的工期都是相同的。

(3) 参数域  $\gamma$  表示优化准则。设工作  $j$  在机器  $i$  上的操作完成时间为  $C_{ij}$ ，工作  $j$  离开系统的时间为  $C_j$ 。工作  $j$  的延迟定义为

$$L_j = C_j - d_j$$

当工作完成滞后时是正，当工作提前完成时为负。

工作  $j$  的滞后定义为

$$T_j = \max(C_j - d_j, 0) = \max(L_j, 0)$$

由此可以看出滞后和延迟的不同在于滞后永远不会是负的。另外，单位惩罚定义为

$$U_j = \begin{cases} 1, & C_j > d_j \\ 0, & \text{其他情况} \end{cases}$$

延迟、滞后、单位惩罚是经常用到的三个与工期有关的惩罚函数，其形状一般如图 1-1 所示。

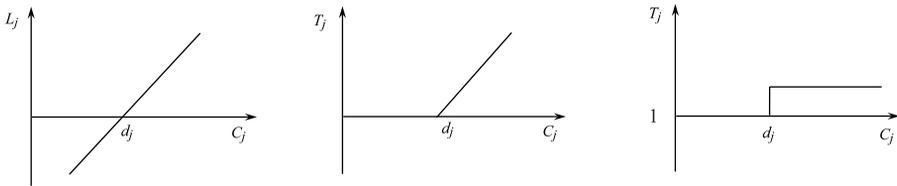


图 1-1 常用的三个与工期相关的惩罚函数

有了上述定义，下面列举出常用的最小化优化目标函数。

最大完成时间（制造期  $C_{\max}$ ）。制造期（makespan）定义为  $\max(C_1, C_2, \dots, C_n)$ ，它和最后一项离开系统的工作完成时间相等。最小化制造期意味着最高效率地利用机器。

最大延迟  $L_{\max}$ ，定义为  $\max(L_1, L_2, \dots, L_n)$ ，它度量违反工期的最坏情况。

加权完成时间和  $\sum w_j C_j$ 。 $n$  项工作完成时间的加权给出了一个调度引起的持有或库存成本指标。完成时间和在文献中通常被称为流水时间，加权完成时间和则称为加权流水时间。

折扣加权完成时间和  $\sum w_j (1 - e^{-rC_j})$ 。这是一个比前一个更一般的成本函数，此时成本以每单位时间  $r$  的比率产生折扣， $0 < r < 1$ 。也就是说，如果工作  $j$  没有在时间  $t$  内完成，则在  $[t, t + dt]$  时间段产生的附加成本为  $w_j r e^{-rt} dt$ 。如果工作  $j$  在时间  $t$  内完成，则在  $[0, t]$  产生的总成本是  $w_j (1 - e^{-rt})$ 。 $r$  的值通常接近于零。

加权滞后和  $\sum w_j T_j$ 。这给出了由工作滞后完成所引起的成本指标，它也是一个比加权完成时间和更一般的成本函数。该性能指标可用于评价不同重要程度客户交货需求的满意度。

加权滞后工作数量  $\sum w_j U_j$ 。这是实际生活中常用的优化目标，在这种优化目标下，显然先加工重要程度高的工作可以得到较好的性能指标。

有了  $\alpha | \beta | \gamma$  三元组表示方法，我们可以用  $Fm | p_{ij} = p_j | \sum w_j C_j$  来表示一个有  $m$  台机器的流水车间。工作  $j$  在所有  $m$  台机器上的加工时间相同且等于  $p_j$ 。目标是找出  $n$  项工作通过系统的顺序，使得加权完成时间总和最小化。

$FJc | r_j, s_{ijk} | \sum w_j T_j$  指的是有  $c$  个加工中心的柔性加工车间。工作具有不同的提交日期，且受制于与机器相关的、由加工顺序决定的准备时间，没有再循环。因此，一项工作最多访问一个加工中心一次。目标是最小化加权滞后时间的总和。这个特殊的问题在实际生活中有很多应用。 $Jm || C_{\max}$  表示有  $m$  台机器的作业车间的调度问题，没有再循环。因此，一项工作最多访问一台机器一次。目标是最小化制造时间。它是本书研究的主要问题类型。

当然，还有很多没有在这个框架中描述的调度模型<sup>[8-10]</sup>。例如，一个更一般的柔性作业车间，它的每个加工中心由许多加工速度与工作相关的并行异速机器组成。当一项工作在通过系统的加工路线上的机器组时，它可以在任何一台机器上加工，但它的加工时间取决于加工它的那台机器。人们也可以定义一个作业车间和开放车间的混合模型。有些工作的加工路线是固定的，而其他工作的路线是

自由的。还有许多这里没有提到的调度特征，这些特征包括循环调度、人员调度、资源约束调度、成批调度。但是有了  $\alpha | \beta | \gamma$  三元组表示方法作为基本语言，调度问题的描述趋于规范化和简洁化，极大地方便了我们对复杂问题的理解和交流。

区别于旅行商、四着色等 NP-hard 问题，车间调度问题具有以下特点<sup>[11-15]</sup>：

(1) 多重约束性。在通常情况下，工件的处理受到严格的工艺路线约束，各道工序的先后关系不能颠倒。同时，由于设备的处理能力有限，工件必须按照顺序在机床上处理。

(2) 离散性。车间生产系统是个典型的离散系统，其中，工件的加工发生在不同的时间和资源上，并且任务的到达、订单的更改、设备的增添和故障等都是离散事件。因此，我们可以利用数学规划、离散系统建模与仿真的方式，通过排序理论研究车间调度问题。

(3) 计算复杂性。车间中的工件、机床、操作人员、物料传送系统和缓冲区之间相互影响、相互制约。工件的加工除上述双重约束外，还要考虑操作人员的熟练程度、刀具数量、系统缓冲能力和各类动态事件的影响。在这些条件的综合影响下，车间调度实质上是一个在若干等式和不等式约束下的组合优化问题，从计算时间复杂度看，它是一个 NP-hard 问题，随着调度问题规模的增大，问题可行解的数量呈指数级增加，因而求解非常困难。

(4) 随机性。车间调度中有很多随机和不确定的因素。例如，工件到达时间的不确定性，实际工件的加工时间也有一定的随机性，以及系统中常有突发事件，如新任务到达、紧急任务插入、交货期改变、订单被取消、设备发生故障等。

(5) 多目标。车间生产中，针对不同的加工任务，相应地有其特定的调度目标，而且这些目标之间往往是有冲突的。例如，生产周期最小化、设备总负荷最小化等。如何使车间调度系统适应不同的任务类型和规模，一直是车间调度中面临的难题。

### 1.3 调度问题及解的分类

根据研究的侧重点不同，车间调度问题有多种分类方式。

#### 1. 资源约束种类和数量

单资源车间调度 (single resource constrained)，即只有一种资源制约着车间的生产能力。在绝大多数的相关科技文献中，单资源一般指车间生产环境中，机床设备的数量不能同时满足所有可加工工序立即被加工的要求。

双资源车间调度 (dual resource constrained), 即同时有两种资源制约着车间的生产能力。机床设备往往是制约资源的因素之一, 车间有时会缺乏有经验或一技之长的工人, 也可能某种类型的刀具数量有限。因此, 这两种资源可以是机床设备和工人或刀具。这种情况的表现形式之一就是工人数量少于机床设备的数量。车间中也常常会发生一些辅助资源有限的情况, 如一个车间只有一辆或两辆自动物料运送车 (automated guided vehicle, AGV), 然而需要同时传送的零件数量很可能较多, 在这种情况下, 自动物料运送车就会成为制约车间提高生产能力的的一个重要因素。同理, 奇缺的刀具、夹具及运送零件的叉车、吊车和货盘等都可能成为第二种制约资源。

多资源车间调度 (multiple resource constrained), 即同时有两种以上的生产所需资源制约着车间的生产能力。这些资源包括员工、机床设备、机器人、物料运送系统和辅助资源, 如货盘、夹具和刀具等。

单资源车间调度是双资源车间调度的特例, 双资源车间调度又是多资源车间调度的特例。因此, 多资源车间调度问题是最复杂的一种, 迄今为止, 很少有制约资源数量为两种以上的文献发表。

## 2. 零件和车间构成

作业车间。在这种车间中, 机床设备的布局可以是任意的, 零件的加工路径是事先确定的, 但工序内容和数量是任意的。这是一种最一般的车间调度形式。

流水车间。在这种车间中, 每个零件都有相同的加工路径。这样, 机床设备的布局就如流水线一样, 零件依次从流水线的一端进入, 最后从另一端流出。

开放车间。每个零件的工序之间的加工顺序是任意的。零件的加工可以从任何一道工序开始, 在任何一道工序结束。

单车间 (single shop)。在这种车间中, 每个零件只能有一道工序。

## 3. 车间调度的类型

静态车间调度 (static scheduling)。所有的零件在开始调度时刻已经准备就绪。车间的调度不考虑零件在加工过程中出现的意外情况, 如机床突然损坏、零件的交货期提前、有更紧迫的零件要求被加工等。

动态车间调度 (dynamic scheduling)。车间的调度要求考虑零件在加工过程中出现的各种意外情况。这种调度方式要求调度能随时响应车间加工能力的变化, 在有突发事件出现后, 能立即根据当时的车间加工能力, 对加工的零件重新展开调度, 以确保在任何时候都能保持车间的加工性能指标处于最优或次优状态。

#### 4. 解的分类

调度中的顺序通常是指  $n$  项工作的排列或在一台给定机器上工作加工的次序。调度通常指的是在更加复杂的机器环境中分配工作。调度策略的概念常常用于随机环境中：策略为系统可能处在的任何一种状态规定了合适的行为。调度者在制定调度规则时，对什么是允许的、什么是不允许的必须作出假定。例如，一个调度可能不允许在任何机器上有任何非强迫的空闲。于是根据解的特性对解空间进行分类是必要和有益的。

**定义 1-1** (可行调度) 若同一机器上既没有任意两个时间区间重叠，也没有分配给同一个工件的任意两个时间区间重叠，并且满足调度问题的一些特殊工艺约束，则称一个调度为可行调度。进而，称使得调度准则或指标最优的可行调度为最优调度。

**定义 1-2** (非延迟调度) 如果一项操作等待加工时没有一台机器是空闲的，那么这个可行的调度称为非延迟调度。一个调度是非延迟的要求等价于禁止非强迫的空闲。

**定义 1-3** (活动调度) 如果在不推迟其他操作或破坏优先顺序的条件下，没有一个操作可以提前加工，则称一个调度为活动调度。

**定义 1-4** (半活动调度) 如果在不改变机器上加工顺序的条件下，没有操作可以提前加工，则称一个调度为半活动调度。

**定义 1-5** (性能指标的正规性) 对于一个调度问题，设  $C_i$  为工件  $J_i$  的加工完毕时间，若对于满足不等式关系  $C_1 \leq C_1'$ ,  $\dots$ ,  $C_n \leq C_n'$  的任意两组加工完成时间  $\{C_i\}$  和  $\{C_i'\}$ ，对  $f$  必须满足  $f(C_1, \dots, C_n) \leq f(C_1', \dots, C_n')$ ，则称性能指标函数  $f$  是正规的。

容易验证，最大完成时间为正规性能指标，对于正规性能指标下的调度问题，其求解过程可以简化。

**定义 1-6** (性能指标的等价性) 对于作业调度问题，若性能指标 A 下的最优调度对应性能指标 B 下的最优调度，那么称两个性能指标 A 和 B 是等价的，反之亦然，即 B 的最优性蕴含了 A 的最优性。

容易验证，对于作业车间调度问题，最大完成时间、平均正在加工工件数和平均机器空闲是等价的。

上述定义表明，若作业车间处于非活动调度下，则一定可以找到某些操作，使其可以更早地加工。当然，有时只能通过改变机器上工件的加工次序来做到。譬如当一个操作在前道工序完成后，可将其插入到同一机器中操作时间比它长而出现时刻比它早的另一个操作之前，即在那个操作还未开始加工前插入到机器的空闲时间内。显然，通过将非活动化调度转化为活动化调度，正规性能指标必然

有所改善。对于正规化指标最大完成时间,已证明最优调度必为活动调度。因此,如果将搜索空间限于活动调度集,不仅能保证最优调度的存在,还能够提高优化效率。

调度解的包含关系如图 1-2 所示。其中,方框为整个调度解空间,feasible 指可行调度空间,semi-active 指半活动调度集,active 指活动调度集,non-delay 指非延迟调度集,optimal 指问题的最优解空间。

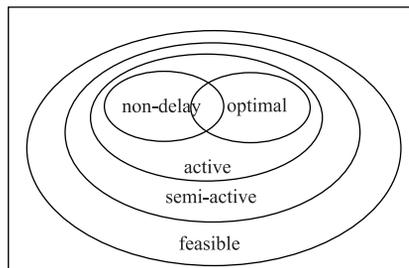


图 1-2 调度解的类型调度及其空间关系图

## 1.4 本书主要内容

随着社会不断向前发展,竞争日益加剧,制造业中普遍面临的许多问题不断地凸显出来。如何在有限的生产资料条件下,以最短的生产周期、最小的成本将产品制造出来以满足客户的需求是每一个制造厂商不断追求的目标。如何运用有限的资源,降低产品的生产成本,缩短产品的制造周期,保证按时交货,提高企业信誉,赢得更多的客户,成为制造厂商在竞争中生存的一个重要条件。有效地利用现有的资源,合理地制定企业和车间生产计划,是达到这个目标的关键。

生产调度的任务就是如何安排加工作业的顺序并合理地分配目前有限的制造资源从而使加工作业在合理的时间内完成。有效的调度方法和优化技术的研究与应用,是实现先进制造和提高生产效益的基础和关键。改善生产调度方案可大大提高企业生产效益和资源利用率,进而增强企业的竞争能力。

生产调度在整个生产流程中的位置和功能如图 1-3 所示。生产调度系统从 ERP 或者 MRPII 提取设备信息、作业任务信息及相关的信息;生产调度系统通过对调度环境的分析,建立工序约束和设备能力约束等调度问题的基本信息;通过分析各任务的重要度,确定各个任务的权重系数,根据调度的优化目标,进行调度优化;调度人员通过对调度结果进行编辑,确认调度结果,这些调度方案通过生产执行系统变成调度指令下达到作业现场进行生产作业<sup>[16-23]</sup>。

本书在对作业车间调度问题进行深入剖析的基础上,对禁忌搜索、进化算法运用于作业车间调度的若干新问题,以及两种特殊的作业车间调度问题进行多方面的探索性研究,旨在夯实本领域的研究基础,为求解作业车间调度问题提供新的有效解决思路,开辟新的研究方向等。各章具体安排如图 1-4 所示。

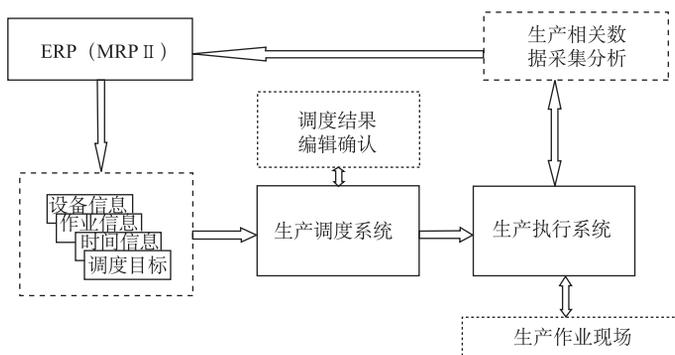


图 1-3 生产调度系统与其他系统的关系

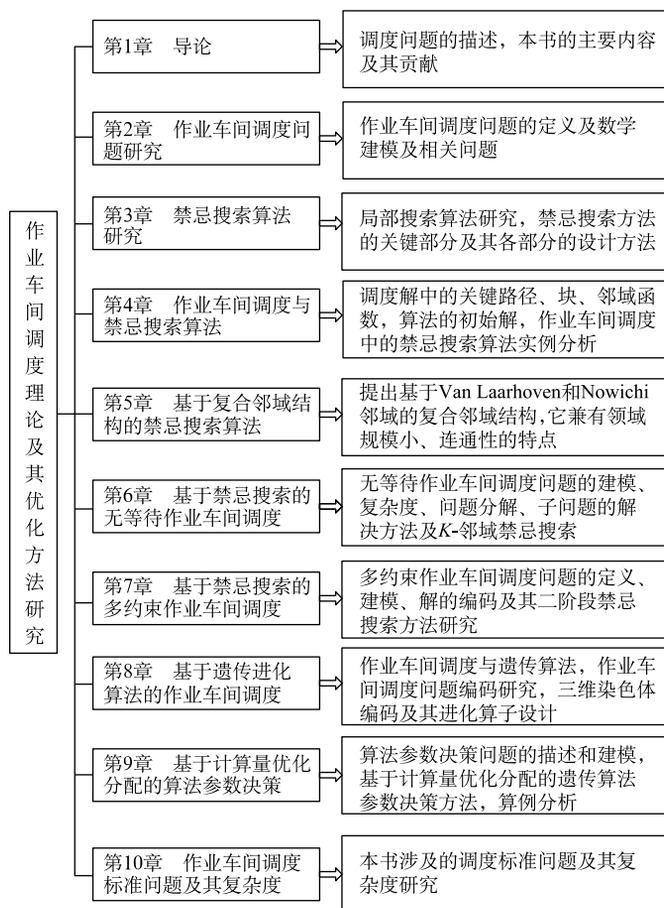


图 1-4 本书的内容结构

## 第 2 章 作业车间调度问题研究

### 2.1 作业车间调度问题的定义及数学描述

车间生产过程中的调度问题是制造系统、运筹技术、管理技术与优化技术发展的核心，有效的调度方法与优化技术的研究和应用已成为先进制造技术实践的基础和关键。作业车间调度问题是所有生产调度中最复杂、最困难，也是更具一般性的问题之一。现代经济日益强化的竞争趋势和不断变化的用户需求要求生产者重新估价生产制造策略，如更短的产品生产周期和零库存系统等，而作业车间调度的生产环境最适宜满足现有经济和用户的需求。利用有限的资源满足被加工任务的各种约束，并确定工件在相关设备上的加工顺序和时间，以保证所选择的性能指标最优，能够潜在地提高企业的经济效益。作业车间调度问题具有很多实际应用背景，开发有效而精确的调度算法是调度和优化领域重要的课题。然而，作业车间调度问题是最困难的约束组合优化问题和典型的 NP-hard 问题，其特点是没有一个有效的算法能在多项式时间内求出其最优解。正是由于作业调度问题的复杂性，即使在规模较小时，当前要获得最优解仍是非常困难的<sup>[24-29]</sup>。

作业车间调度问题是具有特殊工件特性和加工环境的最典型和最重要的调度问题。作业车间调度问题研究  $n$  个工件在  $m$  台机器上的加工过程， $O_{ij}$  表示第  $i$  个工件在第  $j$  台机器上的操作，相应的操作时间  $p_{ij}$  为已知，事先给定各工件在各机器上的加工次序，即技术约束条件，要求确定与技术约束条件相容的各机器上所有工件的加工次序，使加工性能指标达到最优。若各工件的技术约束条件相同，一个作业车间调度问题就转化为较为简单的流水车间调度问题<sup>[30, 31]</sup>。进而，若各机器上各工件的加工次序也相同，则问题可进一步转化为置换流水车间调度问题。

在典型的作业车间调度问题中，除技术约束外，通常还假定以下条件：

- 各工件经过其准备时间后即可开始加工；
- 每一时刻每台机器只能加工一个工件，且每个工件只能被一台机器所加工；
- 一个操作一旦开始则中间不允许间断，整个加工过程中机器均有效；
- 整个加工过程中，每个工件不能在同一台机器上加工多次；
- 各工件必须按工艺路线以指定的次序在机器上加工；
- 不考虑工件加工的优先权；

- 操作允许等待，即前一个操作未完成，则后面的操作需要等待；
- 所有机器处理的加工类型均不同；
- 除非特殊说明，工件的加工时间事先给定，且在整个加工过程中保持不变；
- 除非特殊说明，工件加工时间内包含加工的准备时间；
- 除非特殊说明，缓冲区容量为无穷大。

### 2.1.1 数学规划法

假设  $n$  个工件在  $m$  台机器上加工，已知各操作的加工时间和各工件在各机器上的加工次序约束，要求确定与工艺约束条件相容的各机器上所有工件的加工开始时间或完成时间或加工次序，使加工性能指标达到最优。下面是调度问题  $J_m \|C_{\max}$  的一个整数规划模型描述：

$$\text{Min max}\{\max_{1 \leq k \leq m, 1 \leq i \leq n} c_{ik}\} \quad (2-1)$$

$$\begin{aligned} \text{约束于 } c_{ik} - p_{jk} + M(1 - a_{ihk}) &\geq c_{ih}, \quad i = 1, 2, \dots, n; \quad h, k = 1, 2, \dots, m \\ c_{jk} - p_{ik} + M(1 - x_{ijk}) &\geq p_{jk}, \quad i, j = 1, 2, \dots, n; \quad k = 1, 2, \dots, m \\ c_{ik} &\geq 0, \quad i = 1, 2, \dots, n; \quad k = 1, 2, \dots, m \end{aligned} \quad (2-2)$$

其中，式 (2-1) 表示目标函数，即最小化最大完成时间；式 (2-2) 表示工艺约束条件决定的每个工件的各个操作的加工先后顺序，以及加工各个工件的各机器的先后顺序，同时也保证在同一时刻一个工件不会分配到两个机器上和两台机器不会同时加工一个工件。式中，符号  $c_{ik}$  和  $p_{ik}$  分别为  $i$  工件在机器  $k$  上的完成时间和加工时间； $M$  是一个足够大的正数； $a_{ihk}$  和  $x_{ijk}$  分别为指示系数和指示变量，其意义为

$$\begin{aligned} a_{ihk} &= \begin{cases} 1, & \text{若机器 } h \text{ 先于机器 } k \text{ 加工工件 } i \\ 0, & \text{其他情况} \end{cases} \\ x_{ijk} &= \begin{cases} 1, & \text{如果工件 } i \text{ 先于工件 } j \text{ 在机器 } k \text{ 上加工} \\ 0, & \text{其他情况} \end{cases} \end{aligned}$$

### 2.1.2 析取图描述法

除了整数规划描述外，析取图(disjunctive graph)也是描述作业车间调度的有效工具。典型的  $n$  个工件， $m$  台机器，最小化最大完成时间的问题  $J_m \|C_{\max}$  可以用析取图  $G = (V, A, e)$  来表示(图 2-1)，设总操作数为  $N$ ， $V$  为所有操作  $O_{ij}$  构成的顶点集， $0$  和  $\Theta$  两个虚操作表示加工的开始和结束。

- $A$  为约定工序加工先后顺序的有向弧集。
- $e$  为连接在同一机器上加工的操作的无向弧，无向弧  $e$  形成  $m$  个团，每台机器对应于图中的一个团（团是图论中的术语，指任意两个节点均相互连接的图），属于一个团内的所有操作（节点）需要在相同的机器上加工。
- 连接一个节点的非入射弧都具有节点所代表的加工时间长度。
- 起点  $0$  含有  $n$  条有向弧，其长度为零，指向  $n$  项工作的第一个操作，终点  $\Theta$  含有  $n$  条入射弧，它们来自  $n$  项工作的最后操作。

表 2-1 为  $J3 \parallel C_{\max}$  的一个实例，其析取图如图 2-1 所示。

表 2-1 一个  $J3 \parallel C_{\max}$  实例的加工顺序及时间

工件		操作序号		
		1	2	3
加工时间	J1	10	7	6
	J2	5	8	—
	J3	2	12	4
加工机器	J1	M1	M2	M3
	J2	M3	M1	—
	J3	M1	M2	M3

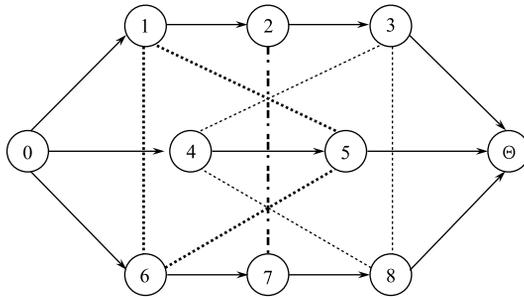


图 2-1 表 2-1 实例的析取图建模

通过图 2-1 的析取图建模，则对作业车间调度的求解就归结为确定各无向弧的方向使最终的图满足如下条件：

- 团中确定弧方向后不能行成回路，并且其所有节点最多有两条入射弧，这样就决定了某台机器上的所有操作顺序；
  - 选定所有无向弧的方向后，图中不能有回路，因为这将对应不可行解。
- 可行解的最大完成时间取决于图中从起点  $0$  到  $\Theta$  的最长路径（关键路径），

路径上的每一个操作紧接着同一机器上的下一个操作或者同一工件在不同机器上的下一个操作。若以最小化最大完成时间为指标，则要求找到一种无向弧的选取方法，使其关键路径长度最小化。图 2-2 为实例的一个可行解，其甘特图如图 2-3 所示。图 2-4 为实例的一个最优解，其甘特图如图 2-5 所示。

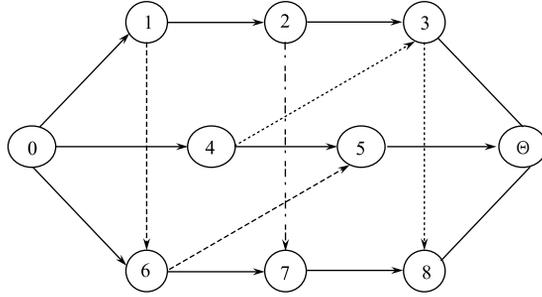


图 2-2 图 2-1 实例的一个可行解

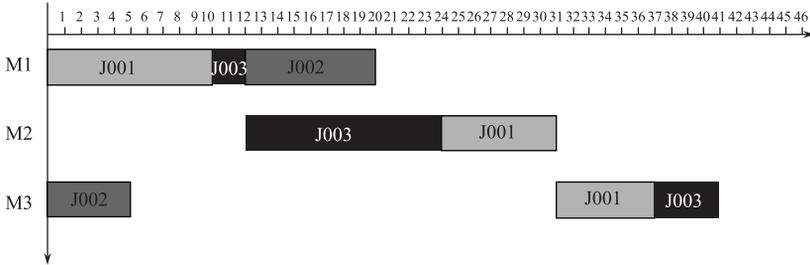


图 2-3 可行解的甘特图

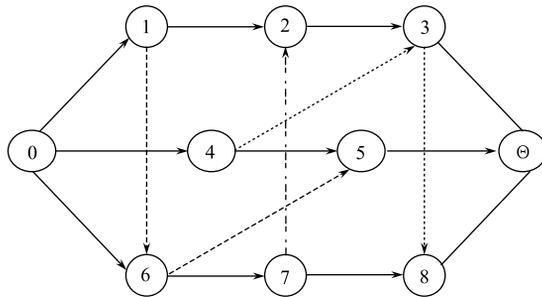


图 2-4 图 2-1 实例的一个最优解

作业车间调度问题的求解远复杂于旅行商问题和流水线作业调度问题，其原因在于：

- (1) 解空间容量巨大。如果算上非可行解， $n$  个工件、 $m$  台机器的问题包含

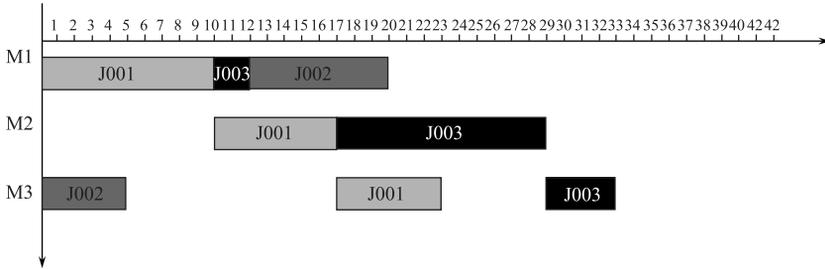


图 2-5 最优解的甘特图

$(n!)^m$  种排列；

- (2) 调度解的编码复杂且多样性，算法的搜索操作多样化；
- (3) 存在工艺技术约束条件的限制，必须考虑解的可行性；
- (4) 调度指标的计算相对算法的搜索比较费时；
- (5) 优化超曲面缺少结构信息，通常复杂且存在多个分布无规则甚至彼此相邻的极小值。

## 2.2 作业车间调度问题的复杂性

问题的复杂性是指在所有的解决该问题的算法中寻找最好算法的复杂性。在一些大规模组合优化问题中，解空间庞大，要用穷举法搜索到问题的最优解是不切实际的。例如，对于对称旅行商问题，所有可行路径有  $(n-1)/2$  条，若以路径比较为基本操作，则需要  $(n-1)/2-1$  次基本操作才能够保证得到绝对最优解。对于每秒执行一百万次操作的计算机，当  $n=20$  时就需要 1929 年才能找到最优解。因此，有些问题存在算法可解，但实际上由于时间或者存储空间的需求难以承受，其在实践中并不一定可解。在算法分析和设计中，一般把求解问题的关键操作（如加、减、乘、比较等运算）指定为基本操作，而把算法执行某个操作的次数定义为算法的时间复杂度。算法执行期间占用的存储单元则定义为算法的空间复杂度<sup>[32-35]</sup>。下面将给出与问题复杂度相关的几个定义，定义中的四类问题的关系如图 2-6 所示。

**P (polynomial) 类问题：**如果一个判定性问题的复杂度是该问题的一个实例的规模  $n$  的多项式函数，则我们说这种可以在多项式时间内解决的判定性问题属于 P 类问题。P 类问题是所有复杂度为多项式时间的问题的集合，记为 P。

**NP (nondeterministic polynomial) 类问题：**对于一个问题的解或相关线索，它的正确与否可以在多项式时间内得到验证。这类问题的集合称为 NP 类判定问题集合，记为 NP。

**NP-C(NP-complete) 类问题：**如果所有的 NP 类问题都可以以多项式归结

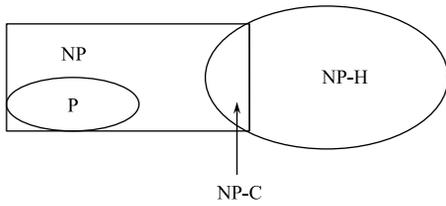


图 2-6 P、NP、NP-C、NP-H 类问题的关系图

到问题 A，则称 A 为 NP-C 类问题。

NP-H (NP-hard) 类问题：对于问题 A，若 NP 中的任何一个问题可在多项式时间归结为判定问题 A，则称 A 为 NP-H 类问题。

NP-C 和 NP-H 两者的区别是：验证一个问题 A 是否为 NP-H 无须判断 A 是否属于 NP，而验证其是否为 NP-C 则需要。

对于生产调度问题，除少数小规模问题存在多项式算法外，大量的调度问题属于 NP-H 类问题，至今仍没找到可以精确求得最优解的多项式时间算法。对于作业车间调度问题来说，两个工件以上的问题都是 NP 类问题，即作业车间调度问题没有找到多项式  $P(n)$  时间算法来求解。一个规模为  $n \times m$  的作业车间调度问题有  $(n!)^m$  个可能的解（包括不可行解在内）。对于一个  $6 \times 6$  的问题，其可能解的个数为  $1.4 \times 10^{17}$ ，如果计算机每秒能试探  $10^6$  个解，则遍历所有解大约需要 4400 年的时间。表 2-2 是一些问题的实例及其复杂性。

表 2-2 几个不同规模问题的枚举求解时间

调度问题规模	可能解的个数	列举需要的时间
$6 \times 6$	$1.4 \times 10^{17}$	$4.4 \times 10^3$
$10 \times 10$	$4.0 \times 10^{65}$	$1.3 \times 10^{52}$
$20 \times 20$	$5.3 \times 10^{367}$	$1.7 \times 10^{354}$
$50 \times 50$	$1.4 \times 10^{3224}$	$4.5 \times 10^{3210}$

### 2.3 作业车间调度问题的求解方法

对于作业车间调度问题的研究，已有几十年的历史了。早在 20 世纪 60 年代，人们就开始了对这一问题的研究，Giffler 和 Thompson 在 1960 年提出了用于生产调度的优先分派规则方法<sup>[36]</sup>。Gere 在 1966 年提出了用于作业车间调度问题一组基于优先分派规则的启发式算法<sup>[37]</sup>。Balas 在 1969 年第一个应用基于析取图的列举方法来处理机器的调度问题<sup>[38]</sup>。这些是可见到的较早的对作业车间调度问题进行研究的文献报道。由于作业车间调度问题的复杂性和困难性，时至今日，人们为了得到一个理想的解决这一问题的方法，一直对该问题进行研究，从而产生了各种各样的用于求解该问题的方法和各种流派，图 2-7 归纳了应用于求解这一问题的主要的技术和分类。

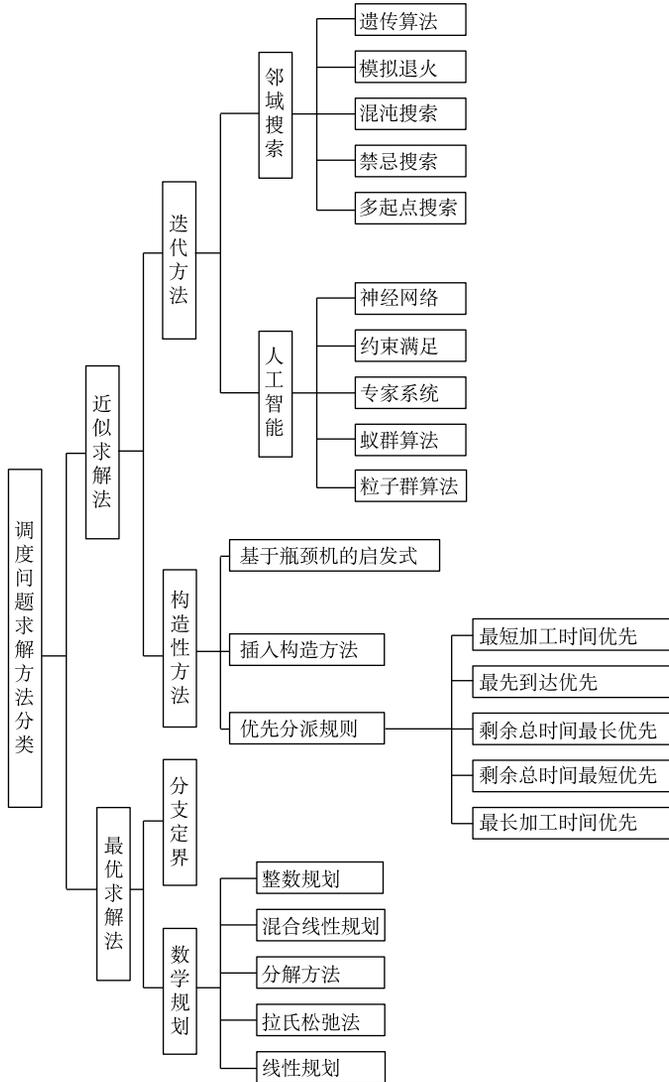


图 2-7 求解作业车间调度的技术与方法分类

从图 2-7 可以看出，用于作业车间调度问题的技术与方法主要分为两类：一类是近似方法；一类是最优化方法。用近似方法求解作业车间调度问题时，可以很快地得到问题的解，但它们不能保证所得到的解是最优的；用最优化方法求解这一问题时，可以得到全局最优解，但它们只能解决小规模作业车间调度问题，而且速度很慢。

### 2.3.1 最优求解方法

最优求解方法主要是通过对作业车间调度问题建立一个整数规划模型,采用基于枚举思想的分支定界法或线性规划法等来解决调度最优化或近优化问题。

#### 1. 分枝定界法

采用动态树结构来描述所有可行解排序的解空间,树的分枝隐含需要被搜索的可行解,某个深度的树的路径表示部分工序的顺序。算法从根节点开始搜索,在对最优解搜索过程中,允许把大部分分枝从搜索过程中去掉。但是分枝定界法方法对初始上界很敏感。在进行分枝定界法之前通常要用启发式方法得到一个上界,然后以此上界为标准进行分支定界。在定界过程中如果树的分枝大于认定的上界,就将此分枝丢掉,从而缩小了搜索空间。如果初始上界设置不当,则可能得不到最优解。目前,对分枝定界法的研究重点是如何改进分枝和定界的策略,从而产生一个更强大的排除规则,以便在最初的搜索阶段从被考虑的问题中去掉大量的节点<sup>[38]</sup>。分枝定界法只适合解决小规模调度问题,并且其对实际问题比较敏感。因此,限制了它在调度问题上的应用。

#### 2. 线性规划法

线性规划法是一种解决在线性约束条件下追求最大或最小的线性目标函数的方法<sup>[39]</sup>。它将调度优化目标和各种约束条件表示为线性等式或不等式,通过求解线性规划问题达到求解调度的目的。理论上,只要问题的最优解存在,就能找到最优解。但由于调度问题所涉及的约束关系数量过大,实际运用中受到“组合爆炸”问题的困扰,因而不能获得真正的实用。

#### 3. 拉格朗日松弛法

拉格朗日松弛法用非负拉格朗日乘子将工艺约束和资源约束进行松弛,最后将惩罚函数引入目标函数中。分枝定界法的主要问题是整数约束,为了解决这个问题,产生了拉格朗日松弛法,它删除了整数约束而加入了相应的代价。Lun等<sup>[40]</sup>用拉格朗日松弛法解决了单机调度和多台并行机调度问题。Chen等<sup>[41]</sup>通过引入更多的拉氏算子松弛了工序之间的顺序约束,并将这种方法用于生产车间调度问题。但是,与分枝定界法相比,拉格朗日松弛法更加耗时。

#### 4. 分解方法

分解方法将原问题分解为多个小的易于解决的子问题,对子问题求出最优,最后求得原问题的最优解。Davis和Jones<sup>[42]</sup>提出了一套基于数学规划问题分解

的方法。这套方法将车间控制分成两层。上层详细说明了每个零件加工的最早开始时间和最迟结束时间，考虑各子问题之间的约束关系；下层通过对所有工序的细致排序来细化上述的时间限制，考虑各个子问题内的约束。Gershwin<sup>[43]</sup>使用临时分解的概念提出了一个数学规划框架，用这个框架来分析生产计划和调度。这个框架的特点是分级和多层。较高的层次处理共性问题，较低的层次处理个性问题，实际的调度发生在最下面三层。

### 2.3.2 近似求解方法

20 世纪 70 年代到 80 年代中期，人们的研究重点是对计算复杂和可计算理论的证明，无数的研究成果都证明了作业车间调度问题属于 NP-H 类问题。于是，该领域的学者不再奢望计算出精确解，而纷纷转向求得近似解，一些著名的近似解算法相应地被提出。近似求解方法包括构造方法和迭代方法。

#### 1. 构造方法

##### 1) 优先分派规则

优先分派规则是最早的近似算法，它给所有的被加工工序分配优先权，然后选出优先权最高的加工工序最先排序，接下来按优先权次序依次进行排序。这种方法非常容易实施，并且计算负担较低。优先规则有很多种，Panwalkar 等人在 1977 年归纳了 100 多个优先规则，其中最主要的有优先选择加工时间最短工序的 SPT (shortest processing time) 规则、先来先服务 FCFS (first come first serve) 规则、优先选择剩余加工时间最长作业的 MWR (most work remaining) 规则、优先选择剩余加工时间最短作业的 LWR (least work remaining) 规则和随机挑选 RANDOM (random selection) 规则等<sup>[44]</sup>。表 2-3 包含了实际中常用的一些分派规则。

表 2-3 常用优先分派规则列表

规则	全称	描述
SPT	shortest processing time	先选择最短加工时间的工序
LPT	longest processing time	优先选择最长加工时间的工序
FCFS	first come first serve	选择同一机床上零件队列中最先到达的工序
MWR	most work remaining	优先选择剩余总加工时间最长的零件的工序
LWR	least work remaining	优先选择剩余总加工时间最短的零件的工序
WSPT	weighted shortest processing time	优先选择最短加工时间的工序 (考虑权重)
EDD	earliest due date	优先选择具有最早交货期的零件
MS	minimum slack	优先选择具有最少迟缓时间的工序
MOR	most operation remaining	优先选择剩余工序数最多的零件的工序
LOR	least operation remaining	优先选择剩余工序数最少的零件的工序

续表

规则	全称	描述
SOT	shortest operation time	优先选择加工时间最短的工序
LOT	longest operation time	优先选择加工时间最长的工序
RANDOM	random selection	随机选择加工工序

优先分派规则对于不同的调度目标产生的效果不同，我们应该按照目标函数的要求选用不同的规则。Vollman 在 1994 年通过大量的仿真实验发现 SPT 对大多数性能指标来说是非常好的分派规则<sup>[45, 46]</sup>。这个结论也被其他大量仿真研究所证实。由于优先分派规则容易实现，计算量小，进而有人提出把几个优先级线性随机地结合起来使用。

## 2) 移动瓶颈算法

Adams 等在 1988 年提出的 SBP 是第一个解决 FT10 标准问题的启发式算法<sup>[47, 48]</sup>。这个构造算法之所以能取得好结果的主要应归功于彻底解决了单一机器排序问题的算法和软件。移动瓶颈算法工作的特点是子问题的识别、瓶颈选择、子问题的解、排序重新最优化<sup>[49]</sup>。具体地说，是把问题化为  $m$  个单一机器问题，即  $m$  个子问题，每次解决一个子问题，把每个子问题的解与所有其他子问题的解比较，每个机器依它们的解的好坏排列，具有最大下界的机器被认为是瓶颈机器。移动瓶颈对瓶颈机器排序，留下被忽视的未被排序的机器，固定已经排序的机器。当每次瓶颈机器排序后，每个先前被排定的有接受改进能力的机器，通过解决单一机器问题方法，再次被局部重新最优化。而单一机器问题的排序是用 Carlier<sup>[50]</sup>的方法通过迭代来解决的，这个方法可以快速给出一个精确的解。移动瓶颈方法的主要贡献是提供了一种用不断转换的单一机器排序来确定一个优化调度的方法。

## 2. 迭代方法

迭代方法分为局部搜索和人工智能 (artificial intelligence, AI) 等方法。

### 1) 局部搜索

局部搜索包括遗传算法、模拟退火、禁忌搜索、混沌搜索、多起点局部搜索等。

### 2) 人工智能

AI 利用知识表达技术把人的知识包括进去，同时使用各种搜索技术求得一个令人满意的调度问题解。它把调度过程描述成一个在满足约束的解空间中搜索的过程。人工智能包括神经网络 (neural network)、约束满足 (constraint satisfaction)、专家系统 (expert systems)、蚁群算法 (ant colony algorithms) 和

模糊逻辑 (fuzzy logic) 等。

(1) 神经网络。它是在对人脑组织结构和运行机制认识理解的基础上, 模拟其结构和智能行为的一种工程系统。神经网络应用于调度问题已有十几年的历史, 虽然神经网络有着分布处理问题的能力, 但对于调度问题来说, 约束条件和变量个数多, 致使网络庞大, 神经元和连接数目非常多, 存在较大的计算负担, 且容易陷入局部解。神经网络模仿了人类学习和对事物进行预测的能力, 是一种并行处理模型。这种模型根据网络拓扑结构、节点特征和训练或学习规则的不同而变化。Rabelo 首先采用反向传播神经网络来处理具有各种类型的生产车间调度问题<sup>[51,52]</sup>。通过训练使得网络能够正确地根据生产特征来选择合适的调度策略和评价指标。网络的训练分别采用 3、4、5、8、10 和 20 台机床的加工情况。训练的输入信息有零件特征, 如零件种类、加工路径、到期时间和加工时间等; 还有车间特征, 如机床台数、加工性能等。训练的输出信息是对特定问题所选分派规则和评价指标的评判等级。Foo 和 Takefuji<sup>[53, 54]</sup>采用 Hopfield 网络来求解一些小型的调度问题。Zhou<sup>[55]</sup>采用双层 Hopfield 网络求解  $4 \times 3$  和  $10 \times 10$  的调度问题。由于产生一个可行调度涉及的变量太多, 所以这些方法计算效率低, 而且常常会产生不可行调度。因而, 它们不能解决实际调度问题。

(2) 约束满足。约束满足技术是应用人工智能技术的迭代近似方法之一, 它应用了很多在分枝定界方法中用到的规则和策略, 通过运用约束来减少搜索空间的有效规模。这些约束限制了选择变量的次序和分配到每个变量可能值的排序。当一个值被分配给一个变量后, 对其进行一致性检查, 将不一致的情况剔除。但是这需要进行回访跟踪。当所有的变量都得到分配的值并且不与约束条件冲突时, 约束满足问题就得到了解决<sup>[55-62]</sup>。虽然这些技术只关心找一个可行解, 但很多情况下存在着表述约束困难的问题, 并且需要大量的回访跟踪。因此, 通常无法达到预期目的。

(3) 专家系统。专家系统和基于知识的系统均由两部分组成, 即知识库和推理机制。知识库包括一些规则、过程和启发式信息等; 推理机制用来选择一种策略处理知识库中的知识, 以便随时解决问题。推理机制分为数据驱动和目标驱动两种。ISIS 是最早提出针对车间调度的专家系统<sup>[63]</sup>。ISIS 采用了面向约束推理的方法, 把约束分成三种类型: 组织目标约束、物理限制约束和临时约束。ISIS 利用这些约束知识维护调度的一致性和验证最低限度满足约束的调度决策。Wu 和 Wysk 提出了一个将专家系统和仿真集成的调度系统, 专家系统采用数据驱动和目标驱动两种方式, 从知识库的众多分派规则和启发式规则中选择小部分适用的规则<sup>[64]</sup>。利用这些规则来优化一个单独的性能指标, 在不同的调度阶段, 性能指标是可以不同的。通过对这些规则的仿真评价, 最终选择一个最好的规则。规则的表现性能可以整理成数据, 在离线状态下可以更新知识库。其他关于专家

系统和基于知识的系统的调度还有 OPIS<sup>[65]</sup> 和 SONIA<sup>[66, 67]</sup>。

(4) 蚁群算法。蚁群算法是 Dorigo 等在 1996 年提出的一种模拟进化算法<sup>[68, 69]</sup>，通过模拟自然界中真实的蚂蚁搜索食物源的行为来求解组合优化问题，并取得了一定的效果。蚁群算法的主要缺点是收敛速度慢。实践证明，信息素初始阶段的积累是影响蚁群算法收敛速度的关键因素之一<sup>[70]</sup>。

(5) 模糊逻辑。模糊逻辑理论一般用在混合调度方法中。它主要用来解决车间调度问题中不确定的加工时间、约束和辅助时间<sup>[71]</sup>。这些不确定性可以用模糊数据表示。Slany 提出了一个在基于知识的调度系统中集成模糊约束松弛的方法<sup>[72]</sup>。Grabot 和 Geneste 用模糊逻辑原理将分派规则进行组合来解决多目标调度问题<sup>[45]</sup>。Tsumimura 等提出了一个用模糊理论为流水车间中加工时间建模的混合调度系统，采用三角模糊数来表示加工时间<sup>[73]</sup>。每一个零件用两个三角模糊数来定义，一个是下界，另一个是上界。用分枝定界过程来优化生产调度的最大完成时间指标。

## 2.4 作业车间调度的研究现状

在求解作业车间调度问题时，强调通过列举算法去求解。用列举算法在求解作业车间调度这一问题时，需要有对求解过程的精细完备的数学表达。虽然列举算法的策略对这一问题有着巨大的理论价值，但从很多的研究工作可以看出，应用列举算法来求解这一问题的结果是极其令人失望的，这种算法主要的问题是对很多调度问题不能得到可行解。因此，它在实际使用中受到了限制<sup>[74]</sup>。

分枝定界方法是主要的列举策略之一<sup>[74]</sup>。它用动态结构分枝来描述所有的可行解排序的解空间，这些分枝隐含有要被搜索的可行解。这个方法可以用数学式和规则来描述，在对最优解搜索过程中，它允许把大部分的分枝从搜索过程中去掉。这种方法流行了很多年。它对解决总工序数  $N < 250$  的问题非常适合，但对于解决大规模问题，由于它需要巨大的计算时间，所以限制了它的使用。另外，分枝定界方法对初始上界值相当敏感，如果初始上界值设置不当，则不能得到最优的可行解。目前，对这种方法研究的重心是如何改进分枝定界的策略，从而产生一个更强大的排除规则，以便在最初的搜索阶段从被考虑的问题中去掉大量的节点。

20 世纪 70 年代到 80 年代，人们研究的重点转移到了如何证明作业车间调度问题的复杂性<sup>[75]</sup>。从对极其棘手的作业车间调度问题的复杂性的大量研究工作中，人们发现仅有极少数特殊实例可以在多项式时间内解决。由于作业车间调度问题的复杂性及其精确列举方法所存在的上述问题，近似方法变成了一种可行的选择。近似方法为了获得速度而放弃了保证最优解。由于它求解速度快，因而

可用于解决较大规模的作业车间调度问题。

最早的近似算法是优先分派规则方法。这种方法是分配一个优先权给所有的被加工工序，然后选出优先权最高的加工工序最先排序，接下来按优先权次序依次进行排序。这种方法非常容易实施，并且计算负担较低。Panwalkar 和 Iskander<sup>[76]</sup>对各种不同规则进行了归纳和总结，并指出最好的优先分派规则算法是几种优先分派规则的组合方法。两种更加新颖的优先分派规则算法包含了模糊逻辑<sup>[77]</sup>和遗传局部搜索<sup>[46]</sup>。总之，大量优先分派规则方法研究工作表明：对于大规模作业车间调度问题，非单一的优先分派规则展示了优势；另外，优先分派规则方法具有短视的天性，比如它只考虑机器的当前状态和解的质量等级等问题，而不能全面的考虑问题。

20 世纪 80 年代末，对作业车间调度问题的研究工作重点从对其复杂性的研究转移到了开创最优化技术上来。由于优先分派规则的缺陷，人们有着对更加适当的最优化技术的追求。80 年代末和 90 年代初是最优化技术最繁荣的时期，在这个时期涌现出了大量的解决作业车间调度问题的新方法。例如，Nowicki 和 Smutnicki<sup>[78]</sup>用约束满足技术和 Foo 和 Takefuji<sup>[79]</sup>用神经网络技术求解这一问题；Storer 等<sup>[80]</sup>的多起点局部搜索；Van Laarhoven 等<sup>[81]</sup>和 Kolonko<sup>[82]</sup>的模拟退火；Glover<sup>[83]</sup>和 Taillard<sup>[84]</sup>的禁忌搜索；Nakano 和 Yamada<sup>[85]</sup>的遗传算法等。虽然在这个繁荣期研究工作主要集中于对近似方法的研究，但一些研究成果被应用于最优方法。例如，Carrier 和 Pinson<sup>[86]</sup>在 1989 年提出了一种分枝定界方法，用这一方法第一次验证了 FT10 标准问题的最优解。在此之前，从 FT10 标准问题给出的 20 多年里，研究者一直不知 FT10 标准问题的最优解是多少。一些研究者用分枝定界方法甚至求解了比 FT10 实例大的问题<sup>[87]</sup>。在 1988 年以前，只有优先分派规则方法可以求解大于 100 个加工工序的实例。

Admas 在 1988 年提出的转换瓶颈算法 (shifting bottleneck procedure, SBP) 是第一个解决 FT10 标准问题的启发式算法<sup>[47, 88]</sup>。这个构造方法能取得好的结果的主要原因之一应归功于彻底解决了单一机器排序问题的算法和软件。转换瓶颈算法工作的特点是：子问题的识别；瓶颈选择；子问题的解；排序重新最优化。具体来说是把问题化为  $m$  个单一机器问题，即  $m$  个子问题，每次解决一个子问题，把这个子问题的解与所有的其他子问题的解比较，每个机器依它们的解的好坏排列，有着最大下界的机器被认为是瓶颈机器。转换瓶颈对瓶颈机器排序，留下被忽视的未被排序的机器，固定已排序的机器。当每次瓶颈机器排序后，每个先前被排定的有接受改进能力的机器，通过解决单一机器问题方法，再次被局部重新最优化。而单一机器问题的排序是用 Carrier 的方法通过迭代来解决，这个方法可以快速给出一个精确的解。转换瓶颈算法的主要贡献是提供了用