

摘 要

排序问题是一类经典的组合优化问题，从上世纪 50 年代至今受到了许多行业的从业人员与理论研究者的密切关注。本文主要研究排序问题在供应链管理中的应用。众所周知，供应链是由多个环节构成的，因此我们不能孤立地研究排序问题，而要把排序问题与其它过程综合考虑。全文共分四章。

第一章主要介绍了供应链与排序问题的一些知识和概念，并且总结了近些年来在综合考虑排序与运输的问题研究方面取得的一些成果。

第二章研究工件占用运输工具空间不同的综合考虑运输和排序的问题。在这类问题中，工件在机器上完成加工后，需要由唯一的一辆运输工具运送到相应的顾客处。运输工具的空间是有限的，每个工件占用运输工具的空间各不相同。目标函数是极小化最后一个到达顾客的工件的到达时间。当机器环境是单台机，所有工件的顾客相同时，我们设计了最坏情况界为 $3/2 + \epsilon$ (ϵ 是任意正常数) 的渐近最优算法。当机器环境是两台平行机，所有工件的顾客相同时，我们给出了最坏情况界为 $5/3$ 的近似算法。

第三章讨论了允许在两个加工工厂之间运送原材料或成品的排序问题。每个工厂可以加工所有属于本身的工件，也可以把一些工件运送到另一个工厂去加工。这样的运送需要一定的时间。若某个工厂需要另一家工厂加工一些工件，根据工厂和工件的不同要求，有以下三种情形：（1）在工件加工之前，原材料不需要运送，工件完成加工后，需要被运送回有需求的工厂；（2）在工件加工之前，需要先运送原材料，工件完成加工后不需要被送回有需求的工厂；（3）在工件加工之前，需要先运送原材料，工件完成加工后，需要被运送回有需求的工厂。问题的目标函数都是极小化最后一个完工工件的完工时间。对这三个问题，我们分别设计了最坏情况界为 $4/3$ ， $4/3$ 和 $3/2$ 的线性时间近似算法，并且给出了动态规划算法。

第四章研究了带承诺到货时间的排序问题。在该问题中，企业根据顾客的订单来加工产品，并把完成的订单交由第三方物流公司运送给顾客。每个订单包含不同的产品数量，而且必须在顾客要求的时间之前送达。订单产生的运输费用与其中的产品数量以及运输需要的时间有关。我们的目标是安排一个加工订单的排序并为每个订单选择运输时间，使得所有订单都能在承诺的最迟到货时间之前到达其顾客处，并且产生的总运输费用尽量地少。我们给出了此问题

的最坏情况界为 2 的近似算法，并且证明了这个界是紧的。

关键词：排序问题，供应链管理，近似算法，最坏情况界，动态规划。

Abstract

Scheduling problem, one of the classical combinatorial optimization problems, has gained great attention from both manufacturers and academic researchers. This thesis mainly concerns some scheduling problems in supply chain management. As we all know, supply chain is composed of several steps, thus the integrated investigation of scheduling and other steps of supply chain is necessary. We considered some scheduling problems with transportation. The thesis is split into four chapters.

We first introduce some related preliminary concepts of supply chain and scheduling and then summarize recent research results of scheduling problems with transportation in chapter 1.

In chapter 2, we investigate scheduling problems with transportation, in which every job needs to be delivered to its customer after production and the completion time is defined as the time when the job arrives at its customer. There is one vehicle with limited capacity, and each job occupies different size in the vehicle during transportation. The goal is to minimize the maximum job completion time. When there is a single machine for processing jobs and only one customer, we present a asymptotical best possible polynomial-time approximation algorithm with worst case ratio of $3/2+\epsilon$ ($\epsilon \rightarrow 0$). When there are two parallel machines for processing jobs and only one customer, we propose a polynomial-time approximation algorithm with worst case ratio of $5/3$.

In chapter 3, we consider scheduling problems with raw material and completed product delivery between two different processing centers. Each processing center is allowed to process jobs belonging to the other one, and in doing this, transportation cost occurs. If a processing center processes jobs belonging to the other one, according to different characteristics of the processing centers and jobs, three cases are studied: (1) Before processing, raw material do not need to be transported to the other machine and after processing, the completed jobs must be transported back to the original machine. (2) Before processing, raw material must be transported to the other machine and after processing, the completed jobs do not need to be transported back to the original machine. (3) Before processing, raw material must be transported to the other machine and after processing, the completed jobs must be transported back to the original ma-

chine. The goal is to minimize the maximum job completion time. For the above three problems, we present linear time approximation algorithms with worst case ratio of $4/3$, $4/3$, and $3/2$, respectively. Besides, an dynamic programming algorithm is given for each problem.

In chapter 4, we study an integrated production and distribution scheduling problem with committed delivery dates. In this problem, the manufacturing company uses a third-party logistics service provider for shipping the completed order with different committed delivery dates to customers. The shipping cost of an order is relative to the order size and the delivery time requested. The goal is to determine a production schedule for accepted orders and a shipping time for delivering each completed order so that the total shipping cost is minimum subject to the constraint that all the orders are completed and delivered to their customers on or before the respective committed delivery dates. For this strongly NP-hard problem, we propose a polynomial-time heuristic algorithm and show that the worst-case performance ratio is bound by 2 and this bound is tight.

Key Words: Scheduling, Supply chain management, Approximation algorithm, Worst-case analysis, Dynamic programming.

第一章 绪论

1.1 排序问题与供应链管理

排序 (Scheduling) 问题是组合优化中一类有着重要理论意义和广泛实际背景的问题, 其实质是研究如何在满足一定要求下, 对需求完成任务的合理安排以得到某种意义下的最优结果。排序理论与理论计算机科学和离散组合数学存在着密切的联系, 并广泛应用到生产计划调度, 信息处理, 物流管理, 服务行业等领域。近几十年来, 排序问题得到了运筹学、工程学、管理学和计算机科学界的极大关注, 并且随着对经典问题研究的日趋深入, 大量具有实际背景的新问题不断涌现。自上世纪 50 年代人们开始研究排序问题至今, 已有大量的排序文献发表在国内外的学术期刊上, 可以这样说, 排序研究已经成为组合优化领域最活跃的分支之一。

按照学术界多年来形成的惯例, 我们把需要完成的任务称为工件 (job), 把完成任务需要的资源称为机器 (machine), 我们希望找到一个可行的排序 (feasible schedule), 使得某个给定的目标函数达到最小 (大)。这里可行一般指在同一时刻, 一台机器至多加工一个工件, 一个工件也只在一台机器上加工, 并且该排序满足问题特定的约束要求。描述一个排序问题可以用一种所谓“三参数表示法” (three-field representation) $\alpha/\beta/\gamma$ [18], 其中 α , β , γ 分别代表特定的机器环境、工件特征和最优准则。它们是排序问题的三个组成部分。

机器环境用来描述机器的数量、不同机器之间的关系等与机器有关的性质。常见的机器系统包括单台机 (single machine)、平行机 (parallel machines)、流水作业 (flow shop)、有序作业 (job shop) 和自由作业 (open shop) 等。其中, 平行机根据其性能的不同还可分为三类: 同型平行机 (identical parallel machines): 系统中所有机器的功能、效率完全一样; 同类平行机 (uniform parallel machines): 系统中机器有各自不同的加工速度, 但任意工件在不同机器上的加工时间有相同的比例关系; 不同类平行机 (unrelated parallel machines): 系统中机器各不相同, 工件在不同机器上的加工时间比不全相同。在三参数表示法中, 它们分别用 P , Q , R 表示。

工件特征一般包括工件的加工时间 (通常也称为工件长度), 工件的释放

时间, 工件相互之间的依赖关系, 工件加工时是否允许中断以及中断恢复后再加工时是否要受惩罚等等。根据排序者对工件信息的了解程度, 又可将排序问题分为离线 (offline)、在线 (online) 和半在线 (semi-online) 等。如果排序者在排序开始前就已经知道工件的全部信息, 例如工件数、每个工件的加工时间等, 则称该问题是离线的。而如果工件的信息是随着排序过程逐个释放的, 即只有在位于某个工件前的全部工件均被安排完毕后, 排序者才能知道该工件的有关信息, 而且工件一旦被安排就不能改变。这样的排序问题称为在线的。但在实际问题中, 大量的问题是介于两者之间的, 即我们或者知道该问题的一些整体信息, 或者知道后续工件的部分信息。我们把这样的问题称为是半在线的。

所谓最优准则, 通俗地讲, 也就是以什么为目标函数。如果记 C_j 为某个排序问题的可行排序中工件 J_j 的完工时间, 则称 $C_{\max} = \max_j C_j$ 为该排序的工件最大完工时间 (makespan), 它即为某种最优准则下的目标函数, 其最优准则为: 找一个可行排序, 使得工件的最大完工时间 C_{\max} 在所有的可行排序中取得最小值, 即最小化目标函数。文献中常见的其他目标函数还有 (赋权) 总完工时间、最大延误时间、最大误工时间、最小误工个数等。有关排序问题的综述, 可参看 [3; 41]。

企业从原材料和零部件采购, 运输, 加工制造, 分销直至最终送到顾客手中的这一过程被看成是一个环环相扣的链条, 这就是供应链。供应链的概念是从扩大的生产 (extended production) 概念发展来的, 它将企业的生产活动进行了前伸和后延。譬如日本丰田公司的精益协作方式就将供应商的活动视为生产活动的有机组成部分而加以控制和协调, 这就是向前延伸。后延是指将生产活动延伸至产品的销售和服务阶段。因此, 供应链就是通过计划、获得、存储、分销、服务等这样一些活动而在顾客和供应商之间形成的一种衔接, 从而使企业能满足内外部顾客的要求。供应链对上游的供应者 (供应活动)、中间的生产者 (制造活动) 和运输商 (储存活动)、以及下游的消费者 (分销活动) 同样重视。因此, 供应链管理就是对整个供应链系统进行计划、协调、操作、控制和优化的各种活动和过程, 其目标是要将顾客所需要的正确的产品 (right product) 能够在正确的时间 (right time), 按照正确的数量 (right quantity), 正确的质量 (right quality) 和正确的状态 (right status) 送到正确的地点 (right place) - 即 “6R”, 并使总成本最小。

在世界经济全球化的今天, 供应链管理已经列为企业一类重要的战略

竞争资源，尤其我国是个制造大国，从供应链管理的角度来考虑企业的经营活
动，形成这方面的核心能力，将对经济发展变得越来越重要。

产品的加工与运输是供应链中的关键两步，把这两步工作有机结合，综
合考虑将对节约成本，提高工作效率与客户满意度有明显的促进作用。已经
有许多学者从整体构建供应链架构的角度对综合考虑加工与运输过程作
了大量研究，这方面的文献综述有 Bilgen 和 Ozkarahan[1]，Chen[5]，Erenguc
等 [10]，Goetschalckx 等 [15]，Sarmiento 和 Nagi [45]，以及 Thomas 和 Griffin
[48]。另外，最近几年来，有许多学者从具体的排序角度来综合研究产品的加
工和运输问题，目标是在考虑相关的效益，费用和顾客满意度的基础上，找到
关于加工与运输产品的最优排序。这类问题是对研究整体供应链构建的深入。

这类问题根据运输之后是否有后续加工可以分为两类，一类是成品的运输
(Integrated production and outbound distribution scheduling, IPODS[4])，也就
是把产品运送到相应的顾客处，排序任务就结束了。另一类是原材料或半成品
的运输 (Integrated production and inbound distribution scheduling, IPIDS[4])，也
就是在运输之后还需要加工工件。在关于 IPODS 的文献中，Hall 和 Potts[19]
[20]，Chen 和 Vairaktarakis[8]，Wang 和 Lee[49]，Pundoor 和 Chen[42]，Chen 和
Pundoor[7]，Li 和 Ou[35]，Li 和 Vairaktarakis[36] 等人研究了综合考虑运输费用
与顾客满意程度的问题。Herrmann 和 Lee [22]，Cheng 等 [9]，Ji 等 [24] 考虑了
总库存成本与总运输费用相结合的问题。Lee 和 Chen[30]，Geismar 等 [14]，Li
等 [37]，Li 和 Ou[34]，Wang 和 Cheng[50]，Pan 等 [40] 研究了运输工具数量有
限的情形下最大化客户满意度的问题。Chen 和 Pundoor[6] 考虑了工件带约束
的条件下，如何最小化运输费用。Garcia 等 [12]，Garcia 和 Lozano[11]，Stecke
和 Zhao[47] 则研究了在保证运输及时性的基础上，最大化总效益的问题。Qi
[43] 和 [44] 研究的问题中，一个工厂可以把顾客要求的工件交由另一家工
厂加工，在不同的工厂之间需要时间来运送工件。Chang 和 Lee[2]，Pan 等
[40] 研究了工件占用运输工具空间大小不同的问题，目标是最大化顾客的满
意度。关于 IPIDS 类问题的文献有 Langston[27]，Hurink 和 Knust [23]，Lee 和
Chen[30]，Lee 和 Strusevich [31]。

下面我们对以上提到的部分研究作更具体的介绍，还有一些文献研究的问
题在后面的章节中会涉及到。文献 [8] 研究了下面的问题：工件在机器上完成加
工后，要由开始位于机器处的容量限制相同的运输工具运送到各自的顾客处，
在这里，有足够多的运输工具可以在机器与顾客间以及不同的顾客间运送工

件,所需的时间各不相同。运输工具每运送一趟,都会产生与该趟运送路线相关的运输费用。问题的目标是极小化最后一个到达顾客的工件的到达时间与总运输费用的凸和或者所有工件到达相应顾客的时间之和与总运输费用的凸和。针对机器环境,顾客的数目以及目标函数不同的情形,作者讨论了计算复杂性,给出了最优算法或者近似算法,并且进行了数值模拟。在文献 [19] 研究的问题中,工件在供应商处完成第一步加工后,被送到工厂处进行第二步加工,然后被运送到相应的顾客处。只有一个供应商,有多个工厂,机器加工环境都是单台机器,运输工具的数目与容量没有限制,目标是极小化总完工时间,最大延误时间或者(加权)总误工工件个数。作者讨论了问题的计算复杂性,给出了动态规划算法。文献 [24] 的问题中,工件在单台机上完成加工后,被分批送往顾客处。运输工具的数目与容量没有限制,每运送一批工件就会产生一笔运输费用,工件完工时间定义为在机器上的完工时间。目标是极小化赋权工件完工时间之和与总运输费用之和。作者讨论了问题的计算复杂性,对一些特殊情形给出了动态规划算法或多项式时间最优算法。文献 [30] 研究了两类带运输的排序问题,第一类问题 (Type 1) 中,机器系统是流水作业,所有工件必须按相同序从在第一台机器到最后一台机器上完成每道工序。工件从一台机器到下一台机器的运输工作由有限个有容量限制,开始时在第一台机器处的运输工具来完成,不同的工件可以在同一批被运送,而运输工具在不同的机器间运送需要的时间各不相同。目标是极小化在最后一台机器上的最大工件完工时间。当运输工具数目,运输工具容量以及工件的一些性质各不相同,作者讨论了问题的计算复杂性,给出了动态规划算法。在第二类问题 (Type 2) 中,工件在机器上完成加工后,需要由有限个有容量限制,开始时在机器处的运输工具运送到同一个顾客处,运输工具从机器到顾客与从顾客回到机器的时间不同。目标是极小化最后一个到达顾客的工件的到达时间。当机器环境,运输工具数目,运输工具容量各不相同,作者讨论了问题的计算复杂性,给出了动态规划算法。文献 [34] 研究了原材料与完工工件都需要运输的问题:在开工之初,仅有的一个有容量限制的运输工具要把未加工的工件从仓库运送到一台单台机上去加工,这辆运输工具还要把完成加工的工件运送回这个仓库。运输工具从仓库到机器与从机器回到仓库的容量限制不同,而所需时间相同。问题的目标是极小化最后一个到达仓库的完工工件的到达时间。作者讨论了问题的计算复杂性,提出了一个多项式时间近似算法,并且进行了数值模拟。文献 [37] 研究了如下的带运输排序问题:工件在单台机上完成加工后,需要由仅有的一台有

容量限制的运输工具运送到各自的顾客处，机器与不同的顾客之间的运输时间以及不同的顾客间的运输时间各不相同，工件的完工时间定义为其到达相应顾客的时间，问题的目标是极小化工件总完工时间。进一步的，作者还讨论了运输工具在顾客内部不能运输，即同一批的工件只能属于同一个顾客的情形。作者讨论了问题的计算复杂性，给出了问题的动态规划算法，并对一些特殊情形给出了最优算法。文献 [39] 考虑了带工件释放时间的问题。在每个工件的释放时间之前，不能加工该工件。工件在单台机上完成加工后，要由唯一的一个运输工具送到同一顾客处。每个工件占用运输工具的空间相同。目标是极小化最后一个到达顾客的工件的到达时间。当工件的加工是可中断时，问题是多项式时间可解的。当工件加工不可中断时，作者提出了一个近似算法，分析了最坏情况界。文献 [40] 中，工件在两台流水作业机器上完成加工后，由仅有的一台有容量限制的运输工具送到同一个顾客处，每个工件占用运输工具空间大小不同，工件的完工时间定义为到达顾客的时间，目标是极小化工件总完工时间或最大工件完工时间。作者讨论了问题的计算复杂性，对一些问题给出了多项式时间的近似算法。文献 [43] 研究了这样一类问题：一个加工工厂可以自己加工接收的订单中的工件，也可以把一些工件交由另一个工厂加工，并把完成的工件送回原来的工厂。针对目的地相同的工件只可以一批运送或者一个一个地运送以及目标函数不同的情形，作者给出了最优算法或者动态规划算法。文献 [50] 研究了带机器维护时段的问题：机器并不总是可运作的，在某个需要维护的时间段内不能加工任何工件。工件在机器上完成加工后，需要由唯一的有容量限制的运输工具送到同一个顾客处，运输工具在机器与顾客间往返的时间相同。问题的目标是极小化最后一个到达顾客的工件的到达时间。对机器环境是单台机器和两台平行机的情形，作者分别给出了问题的多项式时间最优算法或者近似算法。

1.2 算法和计算复杂性

定义 1.2.1: 算法是指一步步求解问题的通用程序，它是解决问题的程序步骤的一个清晰描述。确定性算法从前一步到后一步的运行由当前状态唯一确定。

定义 1.2.2: 对于一个排序问题 π ，如果给定任意一个实例 I ，算法 A 总能找到一个可行解 σ ，并且 σ 的目标值总等于最优解值，则称 A 为最优算法 (Optimal Algorithm)。

将实例通过某种规则编码后输入计算机所用的字节数称为该实例的输入长度。通常，算法所用的时间是指算法中所含的加、减、乘、除、比较等基本运算次数。而算法所用的时间与实例的输入长度有关。

定义 1.2.3: 算法的时间复杂性是指关于实例输入长度 n 的函数 $f(n)$ ，用来表示算法的时间需求，对每一个可能的输入长度，它是指在最坏情况下该算法解此输入长度的实例时所需时间（基本运算步数）。即对于输入长度相同的所有实例，把算法对这些实例的最坏情况作为时间复杂性的度量。

如果存在一个多项式函数 $p(n)$ ，使得算法的时间复杂性为 $O(p(n))$ ，那么称该算法为多项式时间算法，否则称为指数时间算法。还有一类算法叫伪多项式时间算法，它的时间复杂性是关于实例输入长度 n 和实例中最大数的二元多项式函数。在二进制编码下，伪多项式时间算法并不是多项式时间算法，而是指数时间算法。由此出发，可以导出计算复杂性理论的一系列重要概念和结论 [13]。

计算复杂性理论兴起于二十世纪六十年代，和算法的设计与分析密切相关。通过几十年来人们在计算复杂性方面的研究，现今 $P \neq NP$ 的猜想已被基本接受。在此前提下，所谓的 $NP-hard$ 问题就不可能在多项式时间内找到最优解。同时 $NP-hard$ 问题中有一类更难的问题，称为强 $NP-hard$ (Strongly NP-hard) 问题，这类问题甚至都不存在伪多项式时间最优算法 [13]。从而，人们在解决方法的有效性、精确性和时间可行性上寻求平衡。这样，一个自然的想法就是放弃对最优解的寻找，而把研究的重点转向寻找能在较短时间（多项式时间）内得到接近于最优解的可行解，称为近似解。

定义 1.2.4: 对于一个排序问题 π ，如果给定任意一个实例 I ，算法 A 总能找到一个可行解 σ ，那么就称 A 为 π 的近似算法 (Approximation Algorithm)。

LS (List Scheduling) 算法 [16] 和 LPT (Largest Processing Time) 算法 [17] 是经典平行机排序问题的近似算法。 LS 算法将工件按序安排在能使其最早完工的机器上加工。 LPT 算法先把所有工件按加工时间的非增序排列，然后依次将它们安排在能使其最早完工的机器上加工。

我们衡量近似算法的优劣可从两个方面来看，一是算法的时间复杂性，二是算法得到的解值与最优解值的接近程度。

定义 1.2.5: 如果 π 是一个极小 (大) 化问题, I 是任意实例, 设 A 是 π 的一个近似算法. 用 $C_A(I)$ 和 $C^*(I)$ 分别表示算法 A 解实例 I 所得的目标函数值和实例 I 的最优目标函数值. 记 $\rho_A(I) = \frac{C_A(I)}{C^*(I)}$ ($\rho_A(I) = \frac{C^*(I)}{C_A(I)}$), 则近似算法 A 的最坏情况界 (worst-case ratio) 定义为

$$\rho_A(\pi) = \inf\{\rho \geq 1 \mid \rho_A(I) \leq \rho, \forall I\}.$$

如果对问题 π , 任何近似算法的最坏情况界都比 ξ 大, 则称 ξ 是问题 π 的下界.

在不引起混淆的前提下, 我们通常将上述定义中的 $C_A(I)$ 、 $C^*(I)$ 和 $\rho_A(\pi)$ 分别简记为 C_A 、 C^* 和 ρ_A .

定义 1.2.6: 如果某问题有一系列近似算法 $\{A_\epsilon\}$, 对于任意给定的 $\epsilon > 0$, $\{A_\epsilon\}$ 是一个多项式时间算法, 而且 $\rho_{A_\epsilon} \leq 1 + \epsilon$, 则称它为多项式时间近似方案 (Polynomial Time Approximation Scheme), 简记为 PTAS. 进一步的, 如果 $\{A_\epsilon\}$ 的时间复杂性是关于输入长度以及 $\frac{1}{\epsilon}$ 的某个二元多项式, 则称它为完全多项式时间近似方案 (Fully Polynomial Time Approximation Scheme), 简记为 FPTAS.

对于一个强 NP- 难的问题 π , 若存在两个变量的多项式 q 使得对于任何 π 的实例 I , 有

$$OPT(I) < q(\text{Length}[I], \text{Max}[I]),$$

其中, $OPT(I)$ 是实例 I 的最优解的目标函数值, $\text{Length}[I]$ 是实例 I 的输入长度, $\text{Max}[I]$ 是实例 I 中的最大数, 那么不存在问题 π 的 FPTAS.

由于某些组合优化问题本身固有的难度以及最坏情况界估计需要很强的数学基础和技巧, 导致很多情况下无法证明某些算法的最坏情况界, 而实际问题又迫切需要给出一些方案来求解, 在这样的前提下, 我们给出了启发式算法的概念.

定义 1.2.7: 启发式算法是一个基于直观或经验构造的算法. 在可以接受的花费 (指计算时间、占用空间等) 下给出待解决组合优化问题每一个实例的一个可行解, 该可行解与最优解的偏离程度不一定事先可以预计.

评价启发式算法的性能主要分为两类，一是看算法占用的时间、空间等，二是分析算法的效率。算法效率往往采用大规模的随机数据进行实验验证，从平均的角度以及最坏的角度来进行衡量。

1.3 动态规划

在组合优化领域中，有一类问题，由于它的特殊性，可将求解的过程分成若干个互相联系的阶段，在它的每一阶段都需要作出决策，决策的选取不能任意确定，它依赖于当前面临的状态，又影响以后的发展。当各个阶段决策确定后，就组成一个决策序列，因而也就确定了整个求解的过程。这种把一个问题看做是一个前后关联具有链状结构的多阶段过程就称为多阶段决策过程，这种问题称为多阶段决策最优化问题。

动态规划是解决多阶段决策最优化问题的一种思想方法，阶段的确定是动态规划算法的关键。我们首先介绍能用动态规划算法的求解的问题必须满足的条件：最优化原理和无后效性。

定义 1.3.1: 最优化原理可这样阐述：一个最优化策略具有这样的性质，不论过去状态和决策如何，对前面的决策所形成的状态而言，余下的诸决策必须构成最优策略。简而言之，一个最优化策略的子策略总是最优的。

我们用一个简单的例子来说明最优化原理。在下图中，每个点代表一个城市，两个城市之间的边上的数字代表两个城市之间的距离。我们想求出从城市 A 到城市 D 的最短路。可以看到，从城市 A 到城市 D 的最短路一定要经过城市 B，而且这条路必定是从城市 B 到城市 D 的最短路。

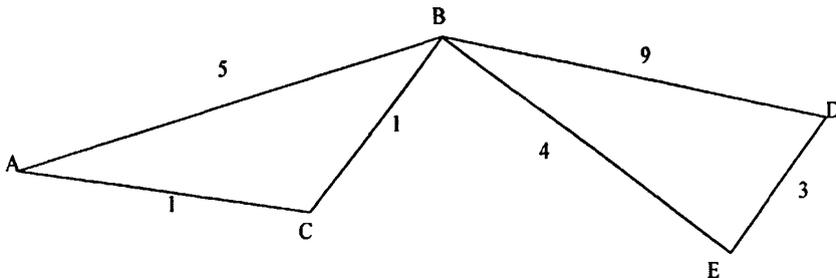


图 1.1 最短路的例子

定义 1.3.2: 将各阶段按照一定的次序排列好之后, 对于某个给定的阶段状态, 它以前各阶段的状态无法直接影响它未来的决策, 而只能通过当前的这个状态。换句话说, 每个状态都是过去历史的一个完整总结。这就是无后向性, 又称为无后效性。

我们还用上面那个例子来说明。假设我们已经确定了从城市 B 到城市 D 的最短路, 那么这一步对我们找从城市 A 到城市 B 的最短路是没有任何影响的。

设计一个标准的动态规划算法, 通常可按以下几个步骤进行:

1. 划分阶段: 按照问题的时间或空间特征, 把问题分为若干个阶段, 这若干个阶段一定要是有序的或者是可排序的 (即无后向性)。
2. 选择状态: 将问题发展到各个阶段时所处于的各种客观情况用不同的状态表示出来, 状态的选择要满足无后效性。
3. 确定决策并写出状态转移方程: 状态转移就是根据上一阶段的状态和决策来导出本阶段的状态。如果我们确定了决策, 状态转移方程也就写出来了。但事实上, 我们常常是反过来做, 根据相邻两段的各状态之间的关系来确定决策。
4. 写出规划方程 (包括边界条件): 一般说来, 只要阶段、状态、决策和状态转移确定了, 这一步比较容易做到。

动态规划的相关知识见 [38]。

1.4 装箱问题与背包问题

在分析本文中的问题时, 我们会应用到组合优化领域两个经典问题: 装箱问题 (Bin packing) 与背包问题 (Knapsack)。

定义 1.4.1: 装箱问题 假设有 n 个物品的集合 $\{J_1, J_2, \dots, J_n\}$, 各个物品的尺寸为 s_j 。现提供若干个箱子, 每个箱子的容量为 $C (C > s_j)$, 试问应该如何把物品放到箱子中, 使得各个箱子中所装物品的尺寸总和不超过 C 并且所用的箱子数尽可能的少。

关于装箱问题, 已经能够证明, 除非 $P = NP$, 否则不存在最坏情况界 $< \frac{3}{2}$ 的多项式时间近似算法 [13]。装箱问题的两个著名的算法是 FF (First Fit) 与 FFD (First Fit Decreasing)。

算法 FF

将物品一个一个装入箱子中，总是使得当前需装箱的物品装入最早被使用的箱子中。

算法 FFD

1. 将物品按照尺寸大小非增方式重新编号。
2. 将物品按照从小到大的编号装入箱子，总是使得当前需装箱的物品装入最早被使用的箱子中。

引理 1.4.1: 对装箱问题的实例 I ，记 $OPT(I)$, $FF(I)$, $FFD(I)$ 分别为最优解，由算法 FF 和 FFD 得到的解中用到的箱子数。我们有以下结论：

$$(1)([33]) FF(I) \leq \frac{7}{4}OPT(I);$$

$$(2)([51]) FFD(I) \leq \frac{11}{9}OPT(I) + 1.$$

引理 1.4.2: 不存在装箱问题的实例，使得 $OPT(I) = 5$ 并且 $FFD(I) = 7$ 。

证明: 见文献 [52]。

定义 1.4.2: 背包问题 假设有 n 个物品的集合 $\{J_1, J_2, \dots, J_n\}$ ，各个物品的尺寸大小为 s_j ，价值为 v_j 。现有一个背包，其容量为 $C(C > s_j)$ ，试问应该选取哪些物品放入背包中，使得背包中所装物品的尺寸总和不超过 C 并且背包中物品的价值总和尽可能的大。

下面我们列出背包问题的一个近似算法，其最坏情况界为 2[26]。

算法 $Ext - Greedy$:

1. $j = 1$;
 $s = 0$; s 表示当前背包中物品的尺寸之和
 $v = 0$; v 表示当前背包中物品的价值之和
2. 若 $j \leq n$, $s + s_j \leq C$,
 则 $x_j = 1$; 把 J_j 放入背包中
 否则, $x_j = 0$;

3. 若 $j = n$, 输出 $\max\{\sum_{j=1}^n x_j \cdot v_j, \max_{j=1}^n v_j\}$. 停止。
 否则, $j = j + 1$, 进入第 2 步。

Lawler [28] 给出了背包问题的 *FPTAS*, 其时间复杂性为 $O(n \log(\frac{1}{\epsilon}) + \frac{1}{\epsilon})$ 。
 另外, Kellerer 和 Pferschy [25] 也提出了背包问题的 *FPTAS*, 其时间复杂性是 $O(n \min\{\log n, \log \frac{1}{\epsilon}\} + \frac{1}{\epsilon^2} \min\{n, \frac{1}{\epsilon} \log \frac{1}{\epsilon}\})$ 。下面我们给出一个较为简单的背包问题的 *FPTAS*[26], 它的时间计算复杂性为 $O(n^2 \cdot \frac{1}{\epsilon})$ 。

算法 *Basic - FPTAS*:

1. 运行算法 *Ext - Greedy*, 记得到的解值为 Z 。
2. 将 $\{J_1, J_2, \dots, J_n\}$ 分成 n/ϵ 个子集合 $N_i, i = 1, 2, \dots, \frac{n}{\epsilon} - 1$, 使得

$$N_i = \{j \mid \frac{iZ\epsilon}{n} \leq p_j \leq \frac{(i+1)Z\epsilon}{n}\}.$$

3. 对任意物品 $J_j, j = 1, 2, \dots, n$, 若 $j \in N_i$, 则

$$p_j = \frac{iZ\epsilon}{n}, sp_j = i.$$

4. $y(0) = 0; y(q) = C + 1, q = 1, 2, \dots, \frac{2n}{\epsilon}; j = 1$ 。

5. $q = \frac{2n}{\epsilon}$ 。

(5.1) 若 $y(q - sp_j) + s_j < y(q)$, 则 $y(q) = y(q - sp_j) + s_j$ 。

(5.2) 若 $q = sp_j$, 进入第 5.3 步; 否则, $q = q - 1$ 。进入第 5.1 步。

(5.3) 若 $j = n$, 输出 $Z^{Basic-FPTAS} = \max\{q \mid y(q) \leq C\} \cdot \frac{Z\epsilon}{n}$ 。停止; 否则, $j = j + 1$ 。进入第 5 步。

1.5 论文概述

本文研究了若干个供应链管理中的排序问题。第二章研究工件占用运输工具空间不同的综合考虑运输和排序的问题。工件在机器上完成加工后, 需要由唯一的一辆运输工具运送到相应的顾客处。运输工具的空间是有限的, 而且每个工件占用运输工具的空间各不相同。目标函数是极小化最后一个到达顾客的

工件的到达时间。当机器环境是单台机，所有工件的顾客相同时，我们设计了最坏情况界为 $3/2 + \epsilon$ (ϵ 是任意正常数) 的渐近最优算法。当机器环境是两台平行机，我们给出了最坏情况界为 $5/3$ 的近似算法。

第三章讨论了允许在两个加工工厂之间运送原材料或成品的排序问题。每个工厂可以加工所有属于本身的工件，也可以把一些工件运送到另一个工厂去加工。这样的运送需要一定的时间，所有目的地一样的工件被同一批运送。若某个工厂需要另一家工厂加工一些工件，根据工厂和工件的不同要求，有以下三种情形：（1）在工件加工之前，需要先运送原材料，工件完成加工后不需要被送回有需求的工厂；（2）在工件加工之前，原材料不需要运送，工件完成加工后，需要被运送回有需求的工厂；（3）在工件加工之前，需要先运送原材料，工件完成加工后，需要被运送回有需求的工厂。问题的目标函数都是极小化最后一个完工工件的完工时间。对这三个问题，我们分别设计了最坏情况界为 $4/3$, $4/3$ 和 $3/2$ 的线性时间近似算法，并且给出了动态规划算法。

第四章研究了带承诺到货时间的排序问题。企业根据顾客的订单来加工产品，并把完成的订单交由第三方物流公司运送给顾客。每个订单包含不同的产品数量，而且必须在顾客要求的时间之前送达。订单产生的运输费用与其中的产品数量以及运输需要的时间有关。我们的目标是安排一个加工订单的排序并为每个订单选择运输时间，使得所有订单都能在承诺的最迟到货时间之前到达其顾客处，并且产生的总运输费用尽量少。我们给出了此问题的最坏情况界为 2 的近似算法，并且证明了这个界是紧的。

第二章 工件占运输工具不同空间的带运输排序问题

2.1 引言

本章主要研究了每个工件占用运输工具不同空间的带运输排序问题，具体描述如下：制造工厂需要加工 n 个工件， $N = \{J_1, J_2, \dots, J_n\}$ ，然后把他们运送到各自的顾客处。工件 $J_j (j = 1, 2, \dots, n)$ 所需的加工时间为 p_j ，加工过程是不可中断的。在运输过程中， J_j 占用运输工具的空间为 s_j 。在整个系统运作之初，一辆运输工具会在制造工厂处等待运输完成加工的工件，其容量为 z ，这表示运输工具每次装的工件的总尺寸不能超过 z 。所有被运输工具在同一次运送的工件称为一个批。运输工具从工厂到某一个顾客以及从这一顾客处再回到工厂需要的时间是相同的。我们的目标是找到制造工厂如何加工工件以及运输工具如何安排运输完工工件的方法，使得最后一个到达其顾客的工件的到达时间尽可能的小。为了方便分析问题，我们把一个排序的最大完工时间定义为运输工具运送完最后一批工件并且回到工厂的时间，记为 C_{max} 。我们的目标就是极小化 C_{max} 。

文献 [2] 研究了上述问题的三种情形：

问题 1 $1 \rightarrow D, k = 1 | v = 1, c = z | C_{max}$ 。

问题 2 $P2 \rightarrow D, k = 1 | v = 1, c = z | C_{max}$ 。

问题 3 $1 \rightarrow D, k = 2 | v = 1, c = z | C_{max}$ 。

在问题 1 中，“ $1 \rightarrow D, k = 1$ ”表示工件首先被单台机加工，然后被送到同一顾客处。“ $v = 1, c = z$ ”表示该问题中有一辆运输工具，其容量大小为 z 。在问题 2 中，机器环境为两台平行机，其余的约束条件和问题 1 一样。在问题 3 中，有两个顾客在不同的地点，运输工具到不同顾客所需的时间不同，其余的条件和问题 1 一样。

文献 [2] 对以上三个问题分别给出了最坏情况界为 $5/3$ ， 2 和 2 的多项式时间近似算法。不难发现对前两个问题，除非 $P = NP$ ，否则不存在最坏情况界比 $3/2$ 好的多项式时间近似算法，这是因为当所有工件的加工时间为 0 时，问题 1 和问题 2 就等价于装箱问题，而除非 $P = NP$ ，否则装箱问题不存在最坏情况界比 $3/2$ 好的多项式时间近似算法 [13]。He 等 [21] 对问题 1 进行了研究，并且给出了最坏情况界为 $53/35$ 的多项式时间近似算法，这与问题的下界 $3/2$

已经非常接近。在本文中，我们给出了问题 1 的渐近最优算法，其最坏情况界为 $\frac{3}{2} + \epsilon$ ，其中 ϵ 是可以任意趋近于 0 的正数。进一步的，我们还提出了问题 2 的一个多项式时间近似算法，其最坏情况界是 $5/3$ 。

在我们目前知道的相关文献中，只有 [2]，[21] 和 [40] 研究了工件占用运输工具空间不同的情形，这比工件占用运输工具空间相同的情形更为复杂，因为在安排工件在机器上加工的同时还要考虑工件在运输工具上的安排，这包含装箱问题作为其子问题。

2.2 $1 \rightarrow D, k = 1 | v = 1, c = z | C_{\max}$ 的近似算法及其最坏情况界分析

我们假设运输工具从制造工厂到顾客和从顾客回到工厂的时间都是 t ，因此来回一趟需要的时间就为 $T = 2t$ 。我们首先列出关于问题 $1 \rightarrow D, k = 1 | v = 1, c = z | C_{\max}$ 的一些结论。

命题 2.2.1: ([2]) 存在问题 $1 \rightarrow D, k = 1 | v = 1, c = z | C_{\max}$ 的一个最优解满足以下条件:

1. 在完成加工所有工件之前，机器无空闲。
2. 同一批工件的加工不会被其他的批中的工件分隔开。
3. 机器可以按任意顺序加工同一批中的工件。
4. 各批工件按其中工件加工时间之和的非减序在机器上被加工，并且运输工具按此顺序运送各个批。

因此，我们接下来在满足上述性质的条件下寻找算法。

引理 2.2.2: ([2]) 对任何满足命题 2.2.1 的解，若 $C_{\max} > P + T$ ，则 $P_1 < T$ 并且 $C_{\max} = P_1 + KT$ ，其中 P 指所有工件加工时间之和， P_1 指第一批中工件总加工时间， K 代表工件被分成的批数。

基于以上结论，Chang 和 Lee [2] 给出了下面的多项式时间近似算法 $H1$ 。

算法 $H1$:

1. 把工件占用运输工具的空间作为物品的尺寸，运输工具的容量作为箱子的容量。（接下来，对工件集合运行算法 FF 和 FFD 时，与此类似。）对

工件集合运行算法 FFD ，装入同一个箱子中的工件形成一个批。记得到的批数为 b_1 。

2. 把第 k 批中的工件总加工时间记为 P_k , $k = 1, 2, \dots, b_1$ 。对这些批进行重新排序，使得 $P_1 \leq P_2 \leq \dots \leq P_{b_1}$ ，并记第 k 批为 B_k 。
3. 从 B_1 开始，依次加工 B_k 中的工件， $k = 1, 2, \dots, b_1$ 。同一批中的工件可以按任意顺序加工。
4. 一旦运输工具允许，就立即运送每一个已经在机器上完工但是未运送的批。如果存在多个这样的批，优先运送下标较小的批。

Chang 和 Lee [2] 证明了算法 $H1$ 的最坏情况界是 $5/3$ ，而该问题的下界是 $3/2$ 。我们接下来会给出一类算法 $MH1(\epsilon)$ ，其中 ϵ 是可以趋近于 0 的任意正常数，可以证明算法 $MH1(\epsilon)$ 的最坏情况界为 $3/2 + \epsilon$ 。因此，算法 $MH1(\epsilon)$ 是问题 $1 \rightarrow D, k = 1 | v = 1, c = z | C_{\max}$ 的渐近最优算法。另外，我们还可以看出，文献 [21] 中的算法 $MH1$ 正是算法 $MH1(\frac{1}{70})$ 。

下面我们给出算法 $MH1(\epsilon)$ 的具体描述

算法 $MH1(\epsilon)$:

1. 运行算法 $H1$ ，令得到的排序为 σ_1 ，其目标函数值为 C_1 。如果 $b_1 \neq 3$ ，输出 σ_1 ，停止；否则，转 2。
2. 构造背包问题的实例如下：对每个工件 J_j , $j = 1, 2, \dots, n$ ，其对应的物品的尺寸大小为 s_j ，价值为 p_j ，背包的容量为 z 。令最坏情况界为 $1 - 4\epsilon$ ，其中 ϵ 是逼近于 0 的任意正常数，运行背包问题的 $FPTAS$ 。把放入背包中的物品对应的工件集合记为 N_1 ，并且对所有的工件进行重新排序，使得 N_1 中的工件在最前面。
3. 对重新排序后的工件集合运行算法 FF ，装入同一个箱子中的工件形成一个批。记得到的批数为 b_2 。
4. 把第 k 批中的工件总加工时间记为 P'_k , $k = 1, 2, \dots, b_2$ 。对这些批进行重新排序，使得 $P'_1 \leq P'_2 \leq \dots \leq P'_{b_2}$ ，并记第 k 批为 B'_k 。
5. 从 B'_1 开始，依次加工 B'_k 中的工件， $k = 1, 2, \dots, b_2$ 。同一批中的工件可以按任意顺序加工。

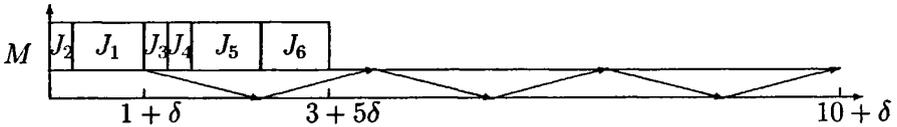
6. 一旦运输工具允许, 就立即运送每一个已经在机器上完工但是未运送的批。如果存在多个这样的批, 优先运送下标较小的批。记此时得到的解为 σ_2 , 其目标函数值是 C_2 。

7. 将 σ_1 与 σ_2 中目标函数值较小的作为输出。

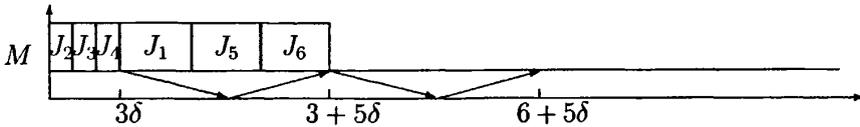
注记 2.2.1: 在算法的第 3 步, 算法 FF 把 N_1 中的工件分配给同一个批。

下面我们用文献 [2] 中的一个实例来说明算法 $MH1(\epsilon)$ 是如何运行的。

例 2.2.1: 运输工具来回一趟所需时间为 $T = 3$, 其容量为 $z = 7$, $s_1 = s_2 = 3$, $s_3 = s_4 = s_5 = s_6 = 2$, $p_1 = p_5 = 1$, $p_2 = p_3 = p_4 = \delta$, $p_6 = 1 + 2\delta$, 其中 δ 是一充分小的正数。取 $\epsilon = 1/20$, 算法 $MH1(1/20)$ 首先运行算法 $H1$, 我们得到 $b_1 = 3$, $B_1 = \{J_1, J_2\}$, $P_1 = 1 + \delta$, $B_2 = \{J_3, J_4, J_5\}$, $P_2 = 1 + 2\delta$, $B_3 = \{J_6\}$, $P_3 = 1 + 2\delta$ 。工件的加工和运输安排如下:



我们得到 $C_1 = 10 + \delta$ 。因为 $b_1 = 3$, $MH1(\epsilon)$ 继续运行第 2 步, 可以得到 $N_1 = \{J_1, J_5, J_6\}$, $b_2 = 2$, $B'_1 = \{J_2, J_3, J_4\}$, $P'_1 = 3\delta$, $B'_2 = \{J_1, J_5, J_6\}$, 以及 $P'_2 = 3 + 2\delta$ 。工件的加工和运输安排如下:



可以看到, $C_2 = 6 + 5\delta$, 并且算法 $MH1(1/20)$ 输出 $\min\{C_1, C_2\} = 6 + 5\delta$ 。

在分析算法 $MH1(\epsilon)$ 的最坏情况界时, 我们用到一下这些记号:

b_L^* = 装箱问题的最优算法将工件分成的批数。

b^* = 该问题的最优算法中将工件分成的批数。

P^* = 算法第 2 步中构造的背包问题实例的最优解值。

u = 该问题的最优算法中开始运送第一批工件的时间。

C^* = 该问题的最优解值。

$C^{MH1(\epsilon)}$ = 由算法 $MH1(\epsilon)$ 得到的解值。

引理 2.2.3: $C^* = \max\{u + b^*T, P + T\}$ 。

证明: 由引理 2.2.2 可以得到此结论。 ■

由引理 2.2.3 我们知道, 若 $C^{MH1(\epsilon)} = P + T$, 则 $C^{MH1(\epsilon)} = C^*$ 。因此, 接下来我们假设 $C^{MH1(\epsilon)} > P + T$ 。

引理 2.2.4: 如果在最优解中工件被分成两批, 则 $P - u \leq P^*$ 。

证明: 每一批中的工件对应的物品的集合都构成背包问题实例的可行解, 而 $P - u$ 是最优解中第二批工件的加工时间之和, 因此 $P - u \leq P^*$ 。 ■

引理 2.2.5: $P'_2 \geq (1 - 4\epsilon)P^*$, 其中 P'_2 是 σ_2 (如果存在) 中最后一批工件的总加工时间。

证明: 由注记 2.2.1 我们知道存在 k , $1 \leq k \leq b_2$, 使得 $P'_k \geq (1 - 4\epsilon)P^*$ 。因为 $P'_1 \leq P'_2 \leq \dots \leq P'_k$, 所以 $P'_2 \geq P'_k \geq (1 - 4\epsilon)P^*$ 。 ■

引理 2.2.6: 如果 $b_1 \neq 3$, 则 $\frac{C^{MH1(\epsilon)}}{C^*} \leq \frac{3}{2}$ 。

证明: 若 $b_1 \neq 3$, 则算法 $MH1(\epsilon)$ 只运行算法 $H1$, 因此 $C^{MH1(\epsilon)} = C_1$, 并且根据前面的假设我们只需要考虑 $C_1 = P_1 + b_1T$ 的情形。注意到此时有 $P_1 < T$ 。

若 $b^* = 1$, 即所有工件尺寸之和不超过 z , 则 $b_1 = 1$ 以及 $C_1 = C^* = P + T$ 。因此, 我们下面只考虑 $b^* \geq 2$ 的情形。

若 $b_1 \leq b^*$, 由引理 2.2.3 可知

$$\begin{aligned} \frac{C_1}{C^*} &\leq \frac{P_1 + b_1T}{u + b^*T} < \frac{P_1 + b^*T}{b^*T} = 1 + \frac{P_1}{T} \cdot \frac{1}{b^*} \\ &< 1 + \frac{1}{b^*} \leq \frac{3}{2}. \end{aligned}$$

因此, 我们接下来只研究 $b_1 > b^*$ 的情形。

情形 1: $C^* = u + b^*T$ 。

由引理 2.2.3 我们可知: $u + b^*T \geq P + T$, 即 $P \leq u + (b^* - 1)T$ 。注意到有 $P_1 \leq P_2 \leq \dots \leq P_{b_1}$, 因此 $P_1 \leq \frac{P}{b_1} \leq \frac{u + (b^* - 1)T}{b_1}$, 进一步的我们可以得到

$$\begin{aligned}
\frac{C_1}{C^*} &= \frac{P_1 + b_1 T}{u + b^* T} \leq \frac{\frac{u + (b^* - 1)T}{b_1} + b_1 T}{u + b^* T} \\
&= \frac{1}{b_1} \cdot \frac{u + b^* T + (b_1^2 - 1)T}{u + b^* T} \\
&= \frac{1}{b_1} + \frac{1}{b_1} \cdot \frac{(b_1^2 - 1)T}{u + b^* T} \\
&< \frac{1}{b_1} + \frac{1}{b_1} \cdot \frac{b_1^2 - 1}{b^*}.
\end{aligned}$$

情形 2: $C^* = P + T$.

由引理 2.2.3 可得 $C^* = P + T \geq u + b^* T > b^* T$, 所以

$$P > u + (b^* - 1)T > (b^* - 1)T.$$

由于 $P_1 \leq \frac{P}{b_1}$, 我们可以推出

$$\begin{aligned}
\frac{C_1}{C^*} &= \frac{P_1 + b_1 T}{P + T} \leq \frac{\frac{P}{b_1} + b_1 T}{P + T} \\
&= \frac{1}{b_1} \cdot \frac{P + T + (b_1^2 - 1)T}{P + T} \\
&= \frac{1}{b_1} + \frac{b_1^2 - 1}{b_1} \cdot \frac{T}{P + T} \\
&< \frac{1}{b_1} + \frac{b_1^2 - 1}{b_1} \cdot \frac{T}{b^* T} \\
&= \frac{1}{b_1} + \frac{b_1^2 - 1}{b_1} \cdot \frac{1}{b^*}.
\end{aligned}$$

综合以上两种情形, 我们可以得到

$$\frac{C_1}{C^*} = \frac{C_1}{\max\{u + b^* T, P + T\}} < \frac{1}{b_1} + \frac{1}{b_1} \cdot \frac{b_1^2 - 1}{b^*}. \quad (2.1)$$

注意到在算法 $H1$ 中, 算法 FFD 把工件分成了 b_1 批, 由引理 1.4.1 可知

$$b_1 \leq \frac{11}{9} b_L^* + 1 \leq \frac{11}{9} b^* + 1. \quad (2.2)$$

若 $b^* = 2$, 则 $b_1 \leq \frac{31}{9}$ 。由 $b_1 > b^*$ 可知 $b_1 = 3$, 这与假设 $b_1 \neq 3$ 矛盾。

若 $b^* = 3$, 由式 (2.2) 可知 $b_1 \leq \frac{42}{9} < 5$ 。注意到有 $b_1 > b^*$, 因此 $b_1 = 4$, 结合式 (2.1) 可以得到 $\frac{C_1}{C^*} < \frac{3}{2}$ 。

类似的, 若 $b^* = 4$, 则 $b_1 = 5$ 并且 $\frac{C_1}{C^*} < \frac{7}{5}$ 。若 $b^* = 5$, 由式 (2.2) 和引理 1.4.2 可知 $b_1 = 6$, 因此 $\frac{C_1}{C^*} < \frac{4}{3}$ 。若 $b^* = 6$, 则 $b_1 = 7$ 或 8 , 因此 $\frac{C_1}{C^*} < \frac{9}{7}$ (若 $b_1 = 7$) 或者 $\frac{C_1}{C^*} < \frac{23}{16}$ (若 $b_1 = 8$)。

若 $b^* \geq 7$, 由式 (2.2) 可以得到 $b^* \geq \frac{9(b_1-1)}{11}$ 。将其代入式 (2.1), 我们可以推出

$$\begin{aligned} \frac{C_1}{C^*} &< \frac{1}{b_1} + \frac{1}{b_1} \cdot \frac{(b_1^2 - 1)}{\frac{9(b_1-1)}{11}} = \frac{11}{9} + \frac{20}{9b_1} \\ &< \frac{11}{9} + \frac{20}{9} \cdot \frac{1}{8} = \frac{3}{2}. \end{aligned}$$

■

引理 2.2.7: 若 $b_1 = 3$, 则 $\frac{C^{MH1(\epsilon)}}{C^*} < \frac{3}{2} + \epsilon$ 。

证明: 与证明引理 2.2.6 类似, 我们只需要考虑 $b^* \geq 2$ 的情形。若 $C_1 = P + T$ 或者 $C_2 = P + T$, 则由引理 2.2.3 可知 $C^{MH1(\epsilon)} = \min\{C_1, C_2\} = C^*$ 。因此, 我们假设 $C_1 > P + T$ 并且 $C_2 > P + T$, 由引理 2.2.2 可知 $P_1 < T$ 且 $P'_1 < T$ 。

若 $b_1 \leq b^*$, 在引理 2.2.6 中我们已经证明了 $\frac{C_1}{C^*} < \frac{3}{2}$, 所以下面我们假设 $b_1 > b^*$ 。因此有 $b^* = 2$ 。由引理 1.4.1 可以得到 $b_2 \leq \frac{7}{4} * 2 = \frac{7}{2}$, 也就是 $b_2 \leq 3$ 。进一步的, 如果 $b_2 \leq b^*$, 与引理 2.2.6 中类似的证明, 我们可以得到 $\frac{C_2}{C^*} \leq \frac{3}{2}$ 。因此, 下面我们假设 $b_2 > b^*$ 。注意到有 $b_2 \leq 3$ 且 $b^* = 2$, 我们可以得出 $b_2 = 3$ 。

由引理 2.2.3 可以知道 $C^* = \max\{u + 2T, P + T\}$, 下面分两种情形讨论:

情形 1 $C^* = u + 2T$ 。

此时有 $P \leq u + T$ 。由引理 2.2.4 和 2.2.5, 我们知道 $P'_3 \geq (1 - 4\epsilon)(P - u)$ 。注意到有 $P'_1 \leq P'_2 \leq P'_3$ 以及 $P = P'_1 + P'_2 + P'_3$, 我们可以推出

$$\begin{aligned} P'_1 &\leq \frac{P - P'_3}{2} \leq \frac{P - (1 - 4\epsilon)(P - u)}{2} = \frac{4\epsilon P + (1 - 4\epsilon)u}{2} \\ &\leq \frac{4\epsilon(u + T) + (1 - 4\epsilon)u}{2} = \frac{4\epsilon T + u}{2}. \end{aligned}$$

因此,

$$\begin{aligned} \frac{C_2}{C^*} &\leq \frac{P'_1 + 3T}{u + 2T} \leq \frac{\frac{4\epsilon T + u}{2} + 3T}{u + 2T} = \frac{\frac{2T + u}{2} + (2 + 2\epsilon)T}{u + 2T} \\ &= \frac{1}{2} + \frac{2(1 + \epsilon)T}{u + 2T} \leq \frac{3}{2} + \epsilon. \end{aligned}$$

情形 2 $C^* = P + T$.

此时有 $P > u + T$ 。由引理 2.2.4 和 2.2.5, 我们知道 $P'_3 \geq (1 - 4\epsilon)(P - u) > (1 - 4\epsilon)T$ 。注意到有 $P'_1 \leq P'_2 \leq P'_3$ 以及 $P = P'_1 + P'_2 + P'_3$, 我们可以推出

$$P'_1 \leq \frac{1}{2}(P - P'_3) < \frac{1}{2}(P - (1 - 4\epsilon)T).$$

因此,

$$\begin{aligned} \frac{C_2}{C^*} &\leq \frac{P'_1 + 3T}{P + T} < \frac{\frac{1}{2}(P - (1 - 4\epsilon)T) + 3T}{P + T} \\ &= \frac{\frac{1}{2}(P + T) + (2 + 2\epsilon)T}{P + T} \\ &< \frac{1}{2} + \frac{(2 + 2\epsilon)T}{P + T} \leq \frac{3}{2} + \epsilon. \end{aligned}$$

■

综合引理 2.2.6 和 2.2.7, 可以得到下面的结论:

定理 2.2.8: $\frac{C^{MH1(\epsilon)}}{C^*} \leq \frac{3}{2} + \epsilon$ 。

2.3 $P2 \rightarrow D, k = 1 | v = 1, c = z | C_{\max}$ 的近似算法及其最坏情况界分析

本节我们研究的问题与上一节的区别在于有两台平行机加工工件, 该问题记作 $P2 \rightarrow D, k = 1 | v = 1, c = z | C_{\max}$ 。Chang 和 Lee [2] 给出了以下算法 $H2$:

算法 $H2$:

1. 对工件集合运行算法 FFD ，装入同一个箱子中的工件形成一个批。记得到的批数为 b^{H2} 。
2. 把第 k 批中的工件总加工时间记为 P_k ， $k = 1, 2, \dots, b^{H2}$ 。对这些批进行重新排序，使得 $P_1 \leq P_2 \leq \dots \leq P_{b^{H2}}$ ，并记第 k 批为 B_k 。
3. 从 B_1 开始，把 B_1 分配给第一台机器，依次把各批工件分配给当前负载较小的那台机器（同一批中的工件分配给同一台机器）。同一批中的工件可以按任意顺序加工。
4. 一旦运输工具允许，就立即运送每一个已经在机器上完工但是未运送的批。如果存在多个这样的批，优先运送下标较小的批。

记算法 $H2$ 得到的排序为 σ^{H2} ，其目标函数值为 C^{H2} ，以及机器的完工时间为 $C(M)$ 。文献 [2] 证明了算法 $H2$ 的最坏情况界是 2。仔细研究算法 $H2$ ，我们可以发现其有两个关键缺陷导致了 C^{H2} 与最优解目标函数值之间的差距。其一，算法 $H2$ 在对工件进行分批时调用算法 FFD ，没有把工件的另一个重要因素 - 加工时间考虑进去。这可能导致开始运送第一批工件的时间比在最优解中的迟很多，延长了运输工具在机器处的等待时间。因此，我们提出算法 A ，通过运行背包问题的 $FPTAS$ 使得运输工具尽早的开始运输在机器上完工的工件。其二，算法 $H2$ 把同一批中的工件分配给同一台机器，没有考虑到这样会使一台机器上有未加工的工件的同时，另一台机器反而可能空闲。所以，我们提出算法 B ，把排序 σ^{H2} 中最后一批工件分为两部分，使得它们相对较为平均的分配给两台机器。

接下来，我们首先给出算法 A 和 B 的具体描述，然后提出对问题 $P2 \rightarrow D, k = 1 | v = 1, c = z | C_{\max}$ 的改进算法 $MH2$ ，必要时算法 $MH2$ 会调用算法 A 或 B 作为子过程。

算法 A :

1. 构造背包问题的实例如下：对每个工件 J_j ， $j = 1, 2, \dots, n$ ，其对应的物品的尺寸大小为 s_j ，价值为 p_j ，背包的容量为 z 。令 $\epsilon = 1/5$ ，运行背包问题的最坏情况界为 $1 - \epsilon$ 的 $FPTAS$ 。把放入背包中的物品对应的工件集合记为 N_1 ，并且对所有的工件进行重新排序，使得 N_1 中的工件在最前面。

2. 对工件集合运行算法 FF , 装入同一个箱子中的工件形成一个批。记得到的批数为 \bar{b} 。
3. 把第 k 批中的工件总加工时间记为 P_k , $k = 1, 2, \dots, \bar{b}$ 。对这些批进行重新排序, 使得 $P_1 \leq P_2 \leq \dots \leq P_{\bar{b}}$, 并记第 k 批工件为 B_k 。
4. 从 B_1 开始, 把 B_1 分配给第一台机器, 依次把各批工件分批给当前负载较小的那台机器 (同一批中的工件分配给同一台机器)。同一批中的工件可以按任意顺序加工。
5. 一旦运输工具允许, 就立即运送每一个已经在机器上完工但是未运送的批。如果存在多个这样的批, 优先运送下标较小的批。

记算法 A 得到的排序为 σ , 其目标函数值为 \bar{C} , 以及机器的完工时间为 $\overline{C(M)}$ 。

注记 2.3.1: 在算法 A 的第 2 步, N_1 中的工件被装入同一批中。

算法 B :

令 σ^{H2} 中的最后一个批为 $B_{b^{H2}}$, $B_{b^{H2}}$ 中的工件为 $\{J_{i_1}, J_{i_2}, \dots, J_{i_m}\}$, 并且 $p_{i_1} \leq p_{i_2} \leq \dots \leq p_{i_m}$ 。如果 $C(M) \leq \frac{5}{6}P$ 或者 $p_{i_m} \geq \frac{3}{5}P$, B 输出的排序 $\tilde{\sigma}$ 就是 σ^{H2} 。否则, B 输出新的排序 $\tilde{\sigma}$ 如下:

1. 令 l 是最大的正整数, 满足 $1 \leq l \leq m$ 并且 $\sum_{j=l}^m p_{i_j} > \frac{P}{6}$ 。记 $B^s = \{J_{i_l}, J_{i_{l+1}}, \dots, J_{i_m}\}$ 为 $B_{b^{H2}}$ 的一个子批。
2. 采用与 σ^{H2} 中一样的方式加工和运输 $B_{b^{H2}}$ 之前的各个批中的工件。把 $B_{b^{H2}} \setminus B^s$ 中的工件分配给 σ^{H2} 中 $B_{b^{H2}}$ 所在的机器, 把 B^s 中的工件分配给另一台机器。仍然把 $B_{b^{H2}} \setminus B^s$ 与 B^s 中的工件一起运送。

记排序 $\tilde{\sigma}$ 的目标函数值为 \tilde{C} , 机器的完工时间为 $\widetilde{C(M)}$ 。

下面我们列出对问题 $P2 \rightarrow D, k = 1 | v = 1, c = z | C_{\max}$ 的改进算法 $MH2$:

算法 $MH2$:

1. 运行算法 $H2$ 。
2. 若 $b^{H2} \neq 3$ 且 $C^{H2} \neq C(M) + T$, 输出 C^{H2} , 停止。

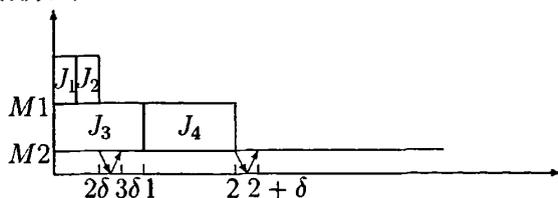
3. 若 $C^{H2} = C(M) + T$, 运行算法 B 。输出 \tilde{C} , 停止。
4. 若 $b^{H2} = 3$ 且 $C^{H2} \neq C(M) + T$, 运行算法 A 。输出 $\min\{C^{H2}, \bar{C}\}$, 停止。

在给出一些实例之前, 我们先列出本节以下部分用到的一些记号。

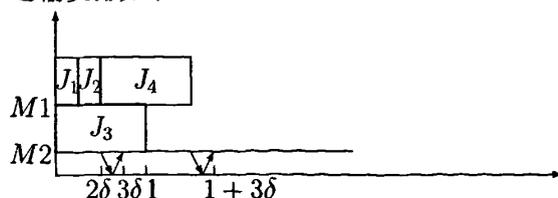
算法	$H2$	A	B	最优
得到的排序	σ^{H2}	$\bar{\sigma}$	$\tilde{\sigma}$	算法
工件被分成的批数	b^{H2}	\bar{b}	b^{H2}	b^*
目标函数值	C^{H2}	\bar{C}	\tilde{C}	C^*
机器完工时间	$C(M)$	$\widetilde{C(M)}$	$\widetilde{C(M)}$	$C(M)^*$
开始运送第一批工件的时间	x	\bar{x}		u
第二批工件的总加工时间	y	\bar{y}		v

另外, 记由算法 $MH2$ 得到的排序的目标函数值为 C^{MH2} 。 P, T, b_L^* 以及 P^* 的定义与上一节中的相同。把第一台机器记为 $M1$, 第二台机器记为 $M2$ 。

例 2.3.1: 我们考虑文献 [2] 中的一个例子。令 $T = \delta, z = 2, s_1 = s_2 = s_3 = s_4 = 1, p_1 = p_2 = \delta$ 以及 $p_3 = p_4 = 1$ 。算法 $MH2$ 首先运行算法 $H2$, 得到 $b^{H2} = 2, B_1 = \{J_1, J_2\}, B_2 = \{J_3, J_4\}, P_1 = 2\delta$ 以及 $P_2 = 2$ 。工件的加工和运输安排如下:



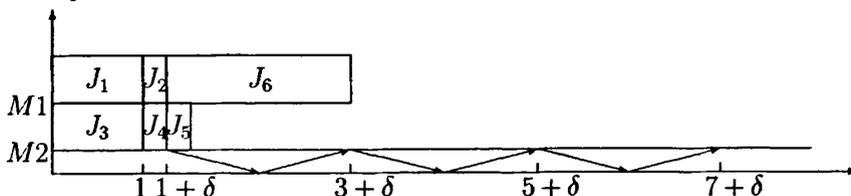
可以得到 $C(M) = 2, C^{H2} = C(M) + T = 2 + \delta$ 。因为 $C^{H2} = C(M) + T$, 算法 $MH2$ 进入第 3 步, 运行算法 B , 我们得到 $B^s = \{J_4\}$ 。在排序 $\tilde{\sigma}$ 中, 工件的加工和运输安排如下:



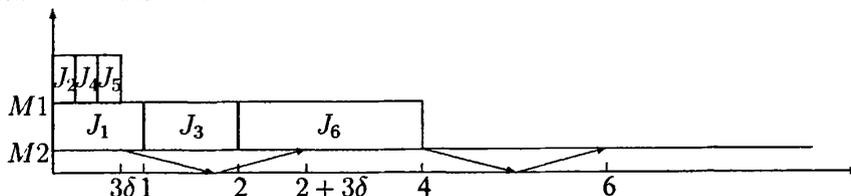
因此 $\widetilde{C(M)} = 1 + 2\delta, \tilde{C} = \widetilde{C(M)} + T = 1 + 3\delta$ 。

不难看出, $C^* = 1 + 2\delta$, $\frac{C^{MH2}}{C^*} = \frac{\bar{C}}{C^*} \leq \frac{5}{3}$ 而 $\frac{C^{H2}}{C^*} \rightarrow 2(\delta \rightarrow 0)$, 这是因为在 $H2$ 中, 算法将 B_2 中的工件都由 $M2$ 加工, 而 $M1$ 的完工时间相对较早。

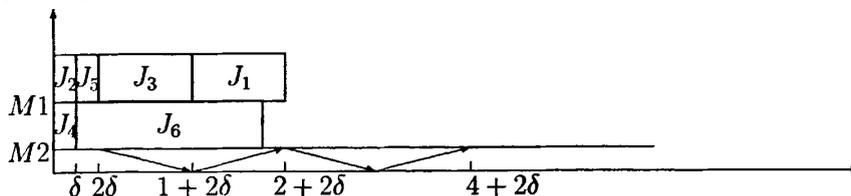
例 2.3.2: $T = 2$, $z = 7$, $s_1 = s_2 = 3$, $s_3 = s_4 = s_5 = s_6 = 2$, $p_1 = p_3 = 1$, $p_2 = p_4 = p_5 = \delta$ 以及 $p_6 = 2$ 。算法 $MH2$ 首先运行算法 $H2$, 可以得到 $b^{H2} = 3$, $B_1 = \{J_1, J_2\}$, $B_2 = \{J_3, J_4, J_5\}$, $B_3 = \{J_6\}$, $P_1 = 1 + \delta$, $P_2 = 1 + 2\delta$, $P_3 = 2$ 。在排序 δ^{H2} 中, 工件的加工和运输安排如下:



可以看到 $C(M) = 3 + \delta$, $C^{H2} = P_1 + 3T = 7 + \delta$ 。因为 $b^{H2} = 3$ 且 $C^{H2} \neq C(M) + T$, 算法 $MH2$ 进入第 4 步, 运行算法 A , 可以得到 $\bar{b} = 2$, $\bar{B}_1 = \{J_2, J_4, J_5\}$, $\bar{B}_2 = \{J_1, J_3, J_6\}$, $\bar{P}_1 = 3\delta$ 以及 $\bar{P}_2 = 4$ 。在排序 $\bar{\sigma}$ 中, 工件的加工和运输安排如下:



我们得到 $\bar{C}(M) = 4$, $\bar{C} = \bar{C}(M) + T = 6$ 。算法 $MH2$ 输出 $\min\{C^{H2}, \bar{C}\} = 6$ 。不难发现, 最优解中工件的加工和运输安排如下:



所以, $C^* = 4 + 2\delta$, $\frac{C^{MH2}}{C^*} \leq \frac{5}{3}$ 而 $\frac{C^{H2}}{C^*} \rightarrow \frac{7}{4}(\delta \rightarrow 0)$ 。这是因为在 $H2$ 中, 算法在工件进行分批时, 只考虑到了工件的尺寸, 忽略了另一个重要因素 - 加工时间。 $MH2$ 运行了背包问题的 FPTAS, 同时考虑了工件的尺寸与加工时间。

引理 2.3.1: ([2]) (1) $P \leq 2C(M)^*$ 。

(2) 若 $C(M)$ 等于 $M1$ 的完工时间, 则 b^{H2} 是奇数; 若 $C(M)$ 等于 $M2$ 的完工时间, 则 b^{H2} 是偶数。

(3) 在排序 σ^{H2} 中, 若 (i) 存在一个批 B_k , $k \geq 3$, 在 $M1$ 上完成加工, 并且一加工完就被运送走, 则 $P_k \geq 2T$; 或者 (ii) 存在一个批 B_q , $q \geq 4$, 在 $M2$ 上完成加工, 并且一加工完就被运送走, 则 $P_q \geq 2T$ 。

引理 2.3.2: ([2]) $C^* \geq \max\{C(M)^* + T, u + b^*T\}$ 。

引理 2.3.3: ([2]) $C^{H2} = \max\{C(M) + T, P - C(M) + 2T, x + b^{H2}T, y + (b^{H2} - 1)T\}$ 。

接下来, 我们分别分析算法 $MH2$ 执行不同的步骤时得到的解的性质。

引理 2.3.4: 当 $b^{H2} \neq 3$ 且 $C^{H2} \neq C(M) + T$ 时, 如果 $C^{H2} = x + b^{H2}T$ 或者 $C^{H2} = y + (b^{H2} - 1)T$, 则 $\frac{C^{H2}}{C^*} \leq \frac{5}{3}$ 。

证明: 因为所有的批都是按照各批中的工件总加工时间 P_k 的非减顺序排列的, 所以 $x \leq \frac{P}{b^{H2}} \leq \frac{2}{b^{H2}}C(M)^*$, $y \leq \frac{P}{b^{H2}-1} \leq \frac{2}{b^{H2}-1}C(M)^*$ 。

若 $u + b^*T \geq C(M)^* + T$, 则 $C(M)^* \leq u + (b^* - 1)T$, 因此 $x \leq \frac{2(u+(b^*-1)T)}{b^{H2}}$, $y \leq \frac{2(u+(b^*-1)T)}{b^{H2}-1}$ 。故我们有

$$\begin{aligned} \frac{x + b^{H2}T}{u + b^*T} &\leq \frac{\frac{2(u+(b^*-1)T)}{b^{H2}} + b^{H2}T}{u + b^*T} \\ &= \frac{2}{b^{H2}} + \frac{(b^{H2} - \frac{2}{b^{H2}})T}{u + b^*T} \\ &< \frac{2}{b^{H2}} + \frac{b^{H2} - \frac{2}{b^{H2}}}{b^*} \end{aligned}$$

和

$$\begin{aligned} \frac{y + (b^{H2} - 1)T}{u + b^*T} &\leq \frac{\frac{2(u+(b^*-1)T)}{b^{H2}-1} + (b^{H2} - 1)T}{u + b^*T} \\ &= \frac{2}{b^{H2} - 1} + \frac{(b^{H2} - 1 - \frac{2}{b^{H2}-1})T}{u + b^*T} \\ &< \frac{2}{b^{H2} - 1} + \frac{b^{H2} - 1 - \frac{2}{b^{H2}-1}}{b^*}. \end{aligned}$$

若 $u + b^*T < C(M)^* + T$, 则我们有

$$\frac{x + b^{H2}T}{C(M)^* + T} \leq \frac{\frac{2C(M)^*}{b^{H2}} + b^{H2}T}{C(M)^* + T}$$

$$\begin{aligned} &< \frac{2}{b^{H2}} + \frac{(b^{H2} - \frac{2}{b^{H2}})T}{u + b^*T} \\ &< \frac{2}{b^{H2}} + \frac{b^{H2} - \frac{2}{b^{H2}}}{b^*}, \end{aligned}$$

和

$$\begin{aligned} \frac{y + (b^{H2} - 1)T}{C(M)^* + T} &\leq \frac{\frac{2C(M)^*}{b^{H2}-1} + (b^{H2} - 1)T}{C(M)^* + T} \\ &< \frac{2}{b^{H2} - 1} + \frac{(b^{H2} - 1 - \frac{2}{b^{H2}-1})T}{u + b^*T} \\ &< \frac{2}{b^{H2} - 1} + \frac{b^{H2} - 1 - \frac{2}{b^{H2}-1}}{b^*}. \end{aligned}$$

综合上面的不等式，我们可以得到

$$\begin{aligned} \frac{x + b^{H2}T}{C^*} &\leq \frac{x + b^{H2}T}{\max\{C(M)^* + T, u + b^*T\}} \\ &< \frac{2}{b^{H2}} + \frac{b^{H2} - \frac{2}{b^{H2}}}{b^*} \end{aligned} \quad (2.3)$$

和

$$\begin{aligned} \frac{y + (b^{H2} - 1)T}{C^*} &\leq \frac{y + (b^{H2} - 1)T}{\max\{C(M)^* + T, u + b^*T\}} \\ &< \frac{2}{b^{H2} - 1} + \frac{b^{H2} - 1 - \frac{2}{b^{H2}-1}}{b^*}. \end{aligned} \quad (2.4)$$

注意到算法 $H2$ 调用 FFD 对工件进行分批，因此 $b^{H2} \leq \frac{11}{9}b_L^* + 1 \leq \frac{11}{9}b^* + 1$ 。

若 $b^{H2} = 1$ ，则 $C^{H2} = C(M) + T$ ，这与假设 $C^{H2} \neq C(M) + T$ 矛盾。

若 $b^{H2} = 2$ ，则 $b^* \geq 2$ 。由于 $y + (b^{H2} - 1)T = C(M) + T$ ，我们只需要考虑 $C^{H2} = x + b^{H2}T$ 的情形。根据式 (2.3) 可以推出 $\frac{x + b^{H2}T}{C^*} < \frac{3}{2}$ 。

若 $b^{H2} = 4$ ，则 $b^* \geq 3$ ，由式 (2.3) 可以得到

$$\frac{x + b^{H2}T}{C^*} < \frac{1}{2} + \frac{4 - \frac{1}{2}}{3} = \frac{5}{3},$$

由式 (2.4) 可以得到

$$\frac{y + (b^{H^2} - 1)T}{C^*} < \frac{2}{3} + \frac{3 - \frac{2}{3}}{3} = \frac{13}{9}.$$

若 $b^{H^2} = 5$, 则 $b^* \geq 4$, 由式 (2.3) 可以得到

$$\frac{x + b^{H^2}T}{C^*} < \frac{2}{5} + \frac{5 - \frac{2}{5}}{4} = \frac{31}{20},$$

由式 (2.4) 可以得到

$$\frac{y + (b^{H^2} - 1)T}{C^*} < \frac{2}{4} + \frac{4 - \frac{2}{4}}{4} = \frac{11}{8}.$$

若 $b^{H^2} = 6$, 则 $b^* \geq 5$, 由式 (2.3) 可以得到

$$\frac{x + b^{H^2}T}{C^*} < \frac{1}{3} + \frac{6 - \frac{1}{3}}{5} = \frac{22}{15},$$

由式 (2.4) 可以得到

$$\frac{y + (b^{H^2} - 1)T}{C^*} < \frac{2}{5} + \frac{5 - \frac{2}{5}}{5} = \frac{33}{25}.$$

若 $b^{H^2} \geq 7$, 则有

$$\frac{x + b^{H^2}T}{C^*} < \frac{2}{b^{H^2}} + \frac{11b^{H^2} - \frac{2}{b^{H^2}}}{9b^{H^2} - 1}$$

和

$$\frac{y + (b^{H^2} - 1)T}{C^*} < \frac{2}{b^{H^2} - 1} + \frac{11b^{H^2} - 1 - \frac{2}{b^{H^2} - 1}}{9b^{H^2} - 1}.$$

定义

$$f(x) = \frac{2}{x} + \frac{11x - \frac{2}{x}}{9x - 1}$$

和

$$g(x) = \frac{2}{x} + \frac{11x - \frac{2}{x}}{9x}.$$

可以得到

$$f'(x) = -\frac{2}{x^2} - \frac{11x^2 - 4x + 2}{9x^2(x-1)^2}$$

和

$$g'(x) = -\frac{2}{x^2} + \frac{44}{9x^3}.$$

不难证明 $f(7) = \frac{625}{42 \times 9} < \frac{5}{3}$, 且当 $x \geq 7$ 时, $f'(x) < 0$; $g(6) = \frac{241}{18 \times 9} < \frac{5}{3}$, 且当 $x \geq 6$ 时, $g'(x) < 0$.

因此, 可以推出当 $b^{H2} \geq 7$ 时, $\frac{x+b^{H2}T}{C^*} < \frac{5}{3}$ 以及 $\frac{y+(b^{H2}-1)T}{C^*} < \frac{5}{3}$. ■

注记 2.3.2: 注意到 $\bar{\sigma}$ 与 σ^{H2} 的区别仅仅在于各个批所装的工件不同, 因此引理 2.3.1, 2.3.3 和 2.3.4 中的结论对 $\bar{\sigma}$ 也成立.

引理 2.3.5: 若 $b^{H2} \neq 3$ 并且 $C^{H2} = P - C(M) + 2T$, 则 $\frac{C^{H2}}{C^*} < \frac{5}{3}$.

证明: 情形 1 $C(M)$ 等于 $M1$ 的完工时间.

由引理 2.3.1(2) 可以知道 b^{H2} 是奇数, 且 $b^{H2} \geq 3$, 因此 $b^{H2} \geq 5$. 由引理 2.3.1(3) 我们可以推出 $P > P_{b^{H2}-1} \geq 2T$, 因此 $T < \frac{P}{2}$. 所以

$$\begin{aligned} \frac{C^{H2}}{C^*} &\leq \frac{P - C(M) + 2T}{C(M)^* + T} \leq \frac{P - \frac{P}{2} + 2T}{\frac{P}{2} + T} \\ &= 1 + \frac{T}{\frac{P}{2} + T} < \frac{3}{2}. \end{aligned}$$

情形 2 $C(M)$ 等于 $M2$ 的完工时间.

由引理 2.3.1(2) 可以知道 b^{H2} 是偶数, 且 $b^{H2} \geq 2$.

如果 $b^{H2} = 2$, 则 $C^{H2} = x + 2T$, 由引理 2.3.4 可以得到 $\frac{C^{H2}}{C^*} \leq \frac{5}{3}$.

若 $b^{H2} \geq 4$, 由引理 2.3.1(3) 可得 $P > P_{b^{H2}-1} \geq 2T$, 因此 $T < \frac{P}{2}$. 所以

$$\begin{aligned} \frac{C^{H2}}{C^*} &\leq \frac{P - C(M) + 2T}{C(M)^* + T} \leq \frac{P - \frac{P}{2} + 2T}{\frac{P}{2} + T} \\ &= 1 + \frac{T}{\frac{P}{2} + T} < \frac{3}{2}. \end{aligned}$$

■

总结引理 2.3.3, 2.3.4 和 2.3.5, 我们可以知道, 如果 $b^{H2} \neq 3$ 并且 $C^{H2} \neq C(M) + T$, 则 $\frac{C^{MH2}}{C^*} = \frac{C^{H2}}{C^*} \leq \frac{5}{3}$.

引理 2.3.6: 如果 (i) $C^{H2} = C(M) + T$ 且 $C(M) \leq \frac{5}{6}P$, 或者 (ii) $C^{H2} = C(M) + T$ 且 $p_{im} \geq \frac{3}{5}P$, 则 $\frac{C^{H2}}{C^*} \leq \frac{5}{3}$.

证明: 注意到此时有 $\frac{C^{H2}}{C^*} \leq \frac{C(M)+T}{C(M)^*+T}$.

如果 $C(M) \leq \frac{5}{6}P$, 则

$$\frac{C^{H2}}{C^*} \leq \frac{\frac{5}{6}P + T}{\frac{P}{2} + T} = 1 + \frac{\frac{P}{3}}{\frac{P}{2} + T} < \frac{5}{3}.$$

如果 $p_{im} \geq \frac{3}{5}P$, 则 $C(M)^* \geq \frac{3}{5}P$, 因此

$$\frac{C^{H2}}{C^*} \leq \frac{P + T}{\frac{3}{5}P + T} = 1 + \frac{\frac{2}{5}P}{\frac{3}{5}P + T} < \frac{5}{3}.$$

■

引理 2.3.7: 若算法 B 得到新的排序 $\tilde{\sigma}$, 则有

- (1) $\widetilde{C(M)} \leq \frac{5}{6}P$,
- (2) $\tilde{C} = x + b^{H2}T$ 或者 $\tilde{C} = \widetilde{C(M)} + T$.

证明: (1) 注意到只有当 $C(M) > \frac{5}{6}P$ 且 $p_{im} < \frac{3}{5}P$ 时, 算法 B 才得到新的排序 $\tilde{\sigma}$. 若 $p_{im} \geq \frac{P}{6}$, 则 $B^s = \{J_{im}\}$ 并且 $\frac{P}{6} \leq p_{im} < \frac{3}{5}P$. 若 $p_{im} < \frac{P}{6}$, 则 $B^s = \{J_{i_1}, J_{i_{l+1}}, \dots, J_{i_m}\}$ 并且 $\frac{P}{6} \leq \sum_{j=l}^m p_{i_j} = \sum_{j=l+1}^m p_{i_j} + p_{i_l} < \frac{P}{6} + \frac{P}{6} < \frac{3}{5}P$. 所以, 在排序 σ^{H2} 中加工 $B_{b^{H2}}$ 的机器的新的负载小于 $C(M) - \frac{P}{6} \leq P - \frac{P}{6} = \frac{5}{6}P$, 而此时另一台机器的负载为 $P - C(M) + \frac{3}{5}P \leq \frac{P}{6} + \frac{3}{5}P = \frac{23}{30}P$. 所以我们可以得到 $\widetilde{C(M)} \leq \max\{\frac{23}{30}P, \frac{5}{6}P\} = \frac{5}{6}P$.

(2) 若 $b^{H2} = 1$, 显然有 $\tilde{C} = \widetilde{C(M)} + T$. 当 $b^{H2} = 2$ 时, 如果运输工具运送完 B_1 并回到机器的时间比 $\widetilde{C(M)}$ 大, 则 $\tilde{C} = x + 2T$, 不然有 $\tilde{C} = \widetilde{C(M)} + T$.

当 $b^{H2} \geq 3$ 时, 不失一般性, 我们假设 $C(M)$ 等于 M_1 的完工时间, 把运输工具运送完 $B_{b^{H2}-1}$ 并回到机器的时间记为 F . 不难证明 F 必定等于下面四个值之一: $x + (b^{H2} - 1)T$, $y + (b^{H2} - 2)T$, $P - C(M) + T$ 或者 $C(M) - P_{b^{H2}} + 2T$.

如果 $F = x + (b^{H2} - 1)T$, 显然有 $\tilde{C} = x + b^{H2}T$ 或者 $\tilde{C} = \widetilde{C(M)} + T$. 对其他情形, 我们将证明 $F < \frac{P}{2} \leq \widetilde{C(M)}$, 因此有 $\tilde{C} = \widetilde{C(M)} + T$.

如果 $F = y + (b^{H^2} - 2)T$, 则第二批工件一旦在机器上完成加工就被运送走, 因此 $y \geq x + T > T$. 注意到 $C(M) > \frac{5}{6}P$ 且 $P_1 \leq P_2 \leq \dots \leq P_{b^{H^2}}$, 我们可以得到

$$y \leq \frac{P - C(M)}{\frac{b^{H^2} - 1}{2}} < \frac{\frac{P}{6}}{\frac{b^{H^2} - 1}{2}} = \frac{P}{3(b^{H^2} - 1)},$$

因此

$$F = y + (b^{H^2} - 2)T < (b^{H^2} - 1)y < \frac{P}{3} < \frac{P}{2}.$$

若 $F = P - C(M) + T$, 则 $B_{b^{H^2-1}}$ 中的工件一旦在机器上完工就被运送走, 由引理 2.3.1(3) 和 $C(M) > \frac{5}{6}P$ 可知

$$2T \leq P_{b^{H^2-1}} < P - C(M) < \frac{P}{6},$$

因此

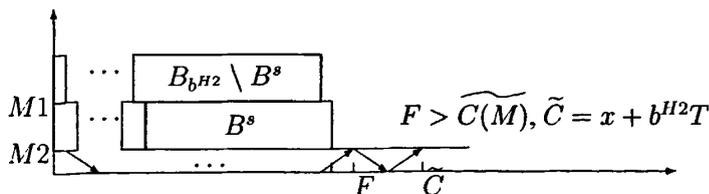
$$F = P - C(M) + T < \frac{P}{6} + \frac{P}{12} < \frac{P}{2}.$$

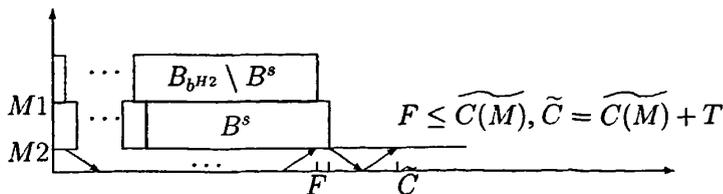
若 $F = C(M) - P_{b^{H^2}} + 2T$, 则 $B_{b^{H^2-2}}$ 中的工件一旦在机器上完工就被运送走, 由引理 2.3.1(3) 和 $C(M) > \frac{5}{6}P$ 可知

$$2T \leq P_{b^{H^2-2}} \leq P_{b^{H^2-1}} < P - C(M) < \frac{P}{6},$$

因此

$$\begin{aligned} F &= C(M) - P_{b^{H^2}} + 2T \leq P - C(M) + 2T \\ &< \frac{P}{6} + \frac{P}{6} < \frac{P}{2}. \end{aligned}$$





引理 2.3.8: 如果 $C^{H2} = C(M) + T$, 则 $\frac{\widetilde{C}}{C^*} \leq \frac{5}{3}$.

证明: 若排序 $\tilde{\sigma}$ 就是 σ^{H2} , 则 $C(M) \leq \frac{5}{6}P$ 或者 $p_{i_m} \geq \frac{3}{5}P$, 由引理 2.3.6 我们已经知道 $\frac{\widetilde{C}}{C^*} = \frac{C^{H2}}{C^*} \leq \frac{5}{3}$.

若 $\tilde{\sigma}$ 是一个新的排序, 则由引理 2.3.7(2) 我们知道 $\widetilde{C} = x + b^{H2}T$ 或者 $\widetilde{C} = \widetilde{C}(M) + T$. 若 $\widetilde{C} = \widetilde{C}(M) + T$, 由于 $\widetilde{C}(M) \leq \frac{5}{6}P$, 与引理 2.3.6 中类似的可以证明 $\frac{\widetilde{C}}{C^*} \leq \frac{5}{3}$. 若 $\widetilde{C} = x + b^{H2}T$, 则 $b^{H2} \geq 2$. 若 $b^{H2} \neq 3$, 引理 2.3.4 已经证明了 $\frac{\widetilde{C}}{C^*} = \frac{x + b^{H2}T}{C^*} \leq \frac{5}{3}$. 若 $b^{H2} = 3$, 则 $C(M)$ 等于 $M1$ 的完工时间, 且 $b^* \geq 2$. 因为 $C(M) > \frac{5}{6}P$, 那么 $x < \frac{P}{6}$, 所以

$$\begin{aligned} \frac{x + 3T}{C^*} &< \frac{\frac{P}{6} + 3T}{C^*} = \frac{\frac{P}{6} + \frac{T}{3} + \frac{8}{3}T}{C^*} \\ &\leq \frac{\frac{P}{6} + \frac{T}{3}}{\frac{P}{2} + T} + \frac{\frac{8}{3}T}{u + 2T} < \frac{1}{3} + \frac{4}{3} \\ &= \frac{5}{3}. \end{aligned}$$

引理 2.3.9: 如果 $b^* = 2$ 且 $\bar{b} = 3$ (如果存在), 则 $\bar{x} \leq \frac{1}{3}C(M)^* + \frac{2}{3}u$.

证明: 由注记 2.3.1 和背包问题的 FPTAS, 我们得到 $\bar{P}_3 \geq \frac{4}{5}P^* > \frac{2}{3}P^* \geq \frac{2}{3}v$. 显然可见, 存在一个最优解, 使得机器首先加工第一批中的工件, 因此 $P \leq 2u + v$, 所以 $\bar{P}_3 \geq \frac{2}{3}(P - 2u)$. 注意到 $\bar{x} \leq \bar{y} \leq \bar{P}_3$, 所以

$$\begin{aligned} \bar{x} &\leq \frac{P - \bar{P}_3}{2} \leq \frac{P - \frac{2}{3}(P - 2u)}{2} \\ &= \frac{1}{6}P + \frac{2}{3}u \leq \frac{1}{3}C(M)^* + \frac{2}{3}u. \end{aligned}$$

引理 2.3.10: 如果 $b^* = 2$ 且 $\bar{C} \neq \widetilde{C}(M) + T$ (如果存在), 则 $\frac{\bar{C}}{C^*} \leq \frac{5}{3}$.

证明: 由引理 1.4.1(1) 和注记 2.3.1, 我们知道 $\bar{b} \leq 3$. 若 $\bar{b} = 2$, 由于 $\bar{C} \neq \overline{C(M)} + T = \bar{y} + T$, 则 $\bar{C} = \bar{x} + 2T = P - \overline{C(M)} + 2T$. 由引理 2.3.4 和注记 2.3.2, 我们可以推出 $\frac{\bar{C}}{C^*} < \frac{5}{3}$.

若 $\bar{b} = 3$, 因为 $\bar{C} \neq \overline{C(M)} + T$, 则 $\bar{C} = \bar{x} + 3T$ 或者 $\bar{y} + 2T = P - \overline{C(M)} + 2T$. 当 $\bar{C} = \bar{y} + 2T$ 时, 由式 (2.4) 和注记 2.3.2 可以得到 $\frac{\bar{C}}{C^*} < \frac{5}{3}$. 当 $\bar{C} = \bar{x} + 3T$ 时, 由引理 2.3.9, 我们知道 $\bar{x} \leq \frac{1}{3}C(M)^* + \frac{2}{3}u$. 根据引理 2.3.2, 可以得到 $C^* \geq \max\{C(M)^* + T, u + 2T\}$.

若 $u + 2T \geq C(M)^* + T$, 则

$$\begin{aligned}\bar{x} &\leq \frac{1}{3}C(M)^* + \frac{2}{3}u \leq \frac{1}{3}(u + T) + \frac{2}{3}u \\ &= u + \frac{1}{3}T.\end{aligned}$$

因此,

$$\begin{aligned}\frac{\bar{C}}{C^*} &\leq \frac{\bar{x} + 3T}{u + 2T} < \frac{u + \frac{1}{3}T + 3T}{u + 2T} \\ &= 1 + \frac{\frac{4}{3}T}{u + 2T} \leq \frac{5}{3}.\end{aligned}$$

若 $u + 2T < C(M)^* + T$, 则

$$\begin{aligned}\bar{x} &\leq \frac{1}{3}C(M)^* + \frac{2}{3}u < \frac{1}{3}C(M)^* + \frac{2}{3}(C(M)^* - T) \\ &= C(M)^* - \frac{2}{3}T.\end{aligned}$$

因此

$$\begin{aligned}\frac{\bar{C}}{C^*} &\leq \frac{\bar{x} + 3T}{C(M)^* + T} < \frac{C(M)^* - \frac{2}{3}T + 3T}{C(M)^* + T} \\ &< 1 + \frac{\frac{4}{3}T}{u + 2T} < \frac{5}{3}.\end{aligned}$$

■

引理 2.3.11: 若 $b^{H2} = 3$ 且 $C^{H2} \neq C(M) + T$, 则 $\frac{C^{MH2}}{C^*} \leq \frac{5}{3}$.

证明: 注意到当 $b^{H2} = 3$ 时, $P - C(M) + 2T = y + 2T$ 。若 $C^{H2} \neq C(M) + T$, 则 $C^{H2} = x + 3T$ 或者 $C^{H2} = y + 2T$ 。

情形 1 $b^* \geq 3$ 。

由式 (2.3) 和 (2.4), 我们可以推出

$$\frac{x + 3T}{C^*} \leq \frac{2}{3} + \frac{3 - \frac{2}{3}}{3} < \frac{5}{3}$$

和

$$\frac{y + 2T}{C^*} \leq 1 + \frac{2 - 1}{3} < \frac{5}{3}.$$

因此, $\frac{C^{MH2}}{C^*} \leq \frac{C^{H2}}{C^*} \leq \frac{5}{3}$ 。

情形 2 $b^* = 2$ 。

若 $C^{H2} = y + 2T$, 则 $y + 2T \geq x + 3T$, 因此 $y \geq x + T > T$ 。注意到有 $y \leq \frac{P}{2}$, 则 $T < \frac{P}{2}$ 。因此,

$$\begin{aligned} \frac{C^{MH2}}{C^*} &\leq \frac{C^{H2}}{C^*} \leq \frac{y + 2T}{C^*(M) + T} \\ &\leq \frac{\frac{P}{2} + 2T}{\frac{P}{2} + T} = 1 + \frac{T}{\frac{P}{2} + T} \\ &< \frac{3}{2}. \end{aligned}$$

若 $C^{H2} = x + 3T$ 且 $T \leq \frac{3}{8}P$, 则由 $x \leq \frac{P}{3}$, 我们得到

$$\begin{aligned} \frac{C^{MH2}}{C^*} &\leq \frac{C^{H2}}{C^*} \leq \frac{x + 3T}{C^*(M) + T} \\ &\leq \frac{\frac{P}{3} + 3T}{\frac{P}{2} + T} = \frac{2}{3} + \frac{\frac{7}{3}T}{\frac{P}{2} + T} \\ &\leq \frac{5}{3}. \end{aligned}$$

若 $C^{H2} = x + 3T$ 且 $T > \frac{3}{8}P$, 我们考虑由算法 A 得到的排序 $\bar{\sigma}$ 。如果 $\bar{C} = \overline{C(M)} + T$, 则

$$\frac{C^{MH2}}{C^*} \leq \frac{\bar{C}}{C^*} = \frac{\overline{C(M)} + T}{C^*}$$

$$\begin{aligned} &\leq \frac{P+T}{\frac{P}{2}+T} = 1 + \frac{\frac{P}{2}}{\frac{P}{2}+T} \\ &< 1 + \frac{\frac{P}{2}}{\frac{P}{2} + \frac{3}{8}T} < \frac{5}{3}. \end{aligned}$$

若 $\bar{C} \neq \overline{C(M)} + T$, 由引理 2.3.10, 我们可以推出 $\frac{C^{MH2}}{C^*} \leq \frac{\bar{C}}{C^*} \leq \frac{5}{3}$. ■

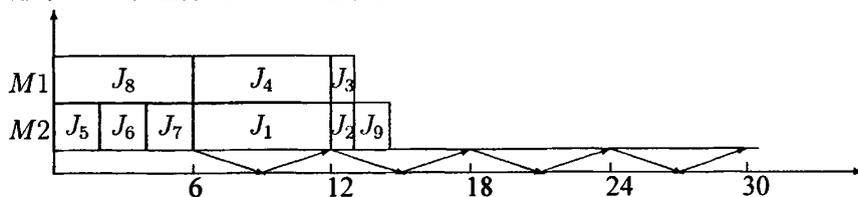
定理 2.3.12: $\frac{C^{MH2}}{C^*} \leq \frac{5}{3}$ 且这个界是紧的。

证明: 由引理 2.3.4, 2.3.5, 2.3.8 和 2.3.11, 我们得到 $\frac{C^{MH2}}{C^*} \leq \frac{5}{3}$ 。为了证明这个界是紧的, 我们考虑如下实例: $T = 6, z = 7, s_1 = s_2 = s_3 = s_4 = 3, s_5 = s_6 = s_7 = s_8 = 2, s_9 = 1, p_1 = p_4 = p_8 = 6, p_2 = p_3 = \delta, p_5 = p_6 = p_7 = 2$ 和 $p_9 = 2\delta$ 。运行算法 $MH2$, 得到 $b^{H2} = 4$, 因此算法 $MH2$ 输出 C^{H2} 并且

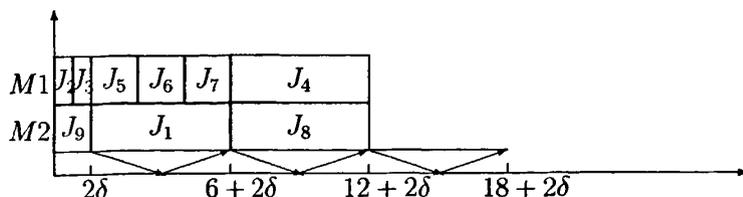
$$B_1 = \{J_8\}, B_2 = \{J_5, J_6, J_7\}, B_3 = \{J_3, J_4\}, B_4 = \{J_1, J_2, J_9\},$$

$$P_1 = 6, P_2 = 6, P_3 = 6 + \delta, P_4 = 6 + 3\delta,$$

排序 σ^{H2} 中工件的加工和运输安排如下:



可以看到 $C^{MH2} = 30$ 。然而, 在最优解中, 工件被分成如下三个批, $B_1^* = \{J_2, J_3, J_9\}, B_2^* = \{J_1, J_5, J_6\}$ 以及 $B_3^* = \{J_4, J_7, J_8\}$ 。工件的加工和运输安排如下:



我们可以看到 $C^* = 18 + 2\delta$ 。因此 $\frac{C^{MH2}}{C^*} = \frac{30}{18+2\delta} \rightarrow \frac{5}{3} (\delta \rightarrow 0)$. ■

第三章 在加工工厂运送原材料或成品的带运输排序问题

3.1 引言

在本章综合考虑的运输与排序的问题中，运输工具可以在不同的加工中心之间运送原材料或成品。这类问题在实际生活中有较多的应用，例如，顾客将汽车送到一家修理工厂进行维修或保养，这家修理工厂可能接到很多这样的工作，而另一家修理工厂的工作量却比较少。这种情况下，工作量多的工厂可以把一部分汽车送到比较空闲的修理工厂去进行维护。在维护完成之后，汽车可能必须被送回原来的修理工厂，也有可能被留在当前的修理工厂等待顾客来提取。另外，在制造行业，同一家公司的不同加工工厂收到的订单数量可能会相差很大，为了尽快完成工作，可以让收到订单数量较少的工厂多加工一些产品，然后送到收到订单数量较多的工厂去。当然，原材料或成品在加工工厂之间的运输需要的一定的时间。文献 [43] 对这类问题也做了一些研究，问题的具体描述如下：有两个加工工厂 M_1 和 M_2 ，每个加工工厂 M_i 需要加工 n_i 个工件 $\bar{J}_i = \{J_{i1}, J_{i2}, \dots, J_{in_i}\}$ ， $i = 1, 2$ 。所有的工件在零时刻已经到达加工工厂，其加工过程是不可中断的。工件 J_{ij} 可以在 M_i 被加工，也可以在 M_k ， $k \neq i$ ，被加工，这样就需要在 M_i 与 M_k 之间运输工件，假设所有目的地一样的工件被同一批运走，我们主要研究三类模型：在第一类模型中，若工件 J_{ij} 被 M_k ， $k \neq i$ ，加工，在加工之前不需要运送原材料，在加工完成之后，需要把 J_{ij} 送回 M_i ，其完工时间定义为回到 M_i 的时间；在第二类模型中，若工件 J_{ij} 被 M_k ， $k \neq i$ ，加工，在加工之前，需要把 J_{ij} 先运送到 M_k ，其完工时间定义为在 M_k 的完工时间；而在第三类模型中，若工件 J_{ij} 被 M_k ， $k \neq i$ ，加工，在加工之前，需要把 J_{ij} 先运送到 M_k ，在完成加工后，还要把 J_{ij} 送回 M_i ，其完工时间定义为回到 M_i 的时间。当然，对以上三个问题，如果工件 J_{ij} 被 M_i 加工，其完工时间就定义为在 M_i 的完工时间。我们的目标都是极小化最大完工时间 (makespan)。我们把第一类模型记作 $P2|trans - completed|C_{max}$ ，第二类模型记作 $P2|trans - unprocessed|C_{max}$ ，第三类模型记作 $P2|trans - two\ way|C_{max}$ 。

在定义记号时，我们用 j 作为工件的标号， i 和 k 作为机器的标号。当 i 和 k 同时出现时，我们总认为 $k \neq i$ 。为了具体定义问题，我们需要如下记号：

p_{ij} = 工件 J_{ij} 的加工时间。（ J_{ij} 在 M_1 和 M_2 的加工时间都是 p_{ij} 。）

$P_i = \bar{J}_i$ 中所有工件的加工时间之和， $P_i = \sum_{j=1}^{n_i} p_{ij}$ 。

$t_{12}(\tau_{12})$ = 把未完工（已完工）工件从 M_1 运送到到 M_2 需要的时间。

$t_{21}(\tau_{21})$ = 把未完工（已完工）工件从 M_2 运送到到 M_1 需要的时间。

$S_{21}^*(S_{12}^*)$ = 最优解中由 M_1 (M_2) 加工的 $\bar{J}_2(\bar{J}_1)$ 中的工件的集合。

$x_{21}^*(x_{12}^*)$ = $S_{21}^*(S_{12}^*)$ 中所有工件的加工时间之和。

C^* = 最优解中工件的最大完工时间。

σ_0 = 工件 J_{ij} 皆由 M_i 加工的排序， σ_0 中不存在工件的运送。

C_0 = σ_0 中工件的最大完工时间。

$\sigma(A)$ = 由算法 A 得到的排序。

C^A = 排序 $\sigma(A)$ 的目标函数值。

不失一般性，我们假设 $P_1 \leq P_2$ ， $p_{2n_2} = \max_{\{j=1,2,\dots,n_2\}} p_{2j}$ 。

由于两台平行机排序是我们研究的问题的特殊情形，因此，本章中的问题都是 NP -难的。本章的第二节中，我们对每个模型都给出了一个线性时间的近似算法并且分析了其最坏情况界。在第三节，我们对第三个模型分别给出了动态规划算法，在此算法基础之上，可以得到前两个问题的动态规划算法。在第四节，我们讨论了将两台机器的运作综合考虑带来的益处。

3.2 近似算法及其最坏情况界分析

在本节中，我们对以上三个问题分别提出线性时间的近似算法。在我们的算法中，需要调用下面这个算法作为子过程：

算法 C ：

构造装箱问题的实例如下：对每个工件 J_{2j} ， $j = 1, 2, \dots, n_2$ ，其对应的物品的尺寸为 p_{2j} ，箱子的容量为 $P_2/2$ 。对这个实例运行算法 FF ，令得到的解中所装物品的总尺寸最大的箱子为 B_1 ， B_1 中物品对应的工件的集合为 S_{21} ， S_{21} 中所有工件的加工时间之和为 x_{21} 。

注记 3.2.1: 算法 C 得到的解中至多用三个箱子。不然，如果用了四个以上的箱子，那么所有物品的尺寸之和超过 P_2 ，这与实例的构造矛盾。因此，我们得出 $\frac{P_2}{3} \leq x_{21} \leq \frac{P_2}{2}$ 。

3.2.1 $P2|trans - completed|C_{max}$

在这个问题中, 若工件 J_{ij} 被 M_k 加工, 则在加工完成之后, J_{ij} 需要被送回 M_i 。

引理 3.2.1: 若 $S_{21}^*(S_{12}^*) \neq \emptyset$, 则问题 $P2|trans - completed|C_{max}$ 有一个最优解, 使得 $M1(M2)$ 首先加工 $S_{21}^*(S_{12}^*)$ 中的工件, 然后加工 $\bar{J}_1 \setminus S_{12}^*(\bar{J}_2 \setminus S_{21}^*)$ 中的工件。

由此引理我们知道, 当 $x_{21}^* > 0$ ($S_{21}^* \neq \emptyset$) 且 $x_{12}^* > 0$ ($S_{12}^* \neq \emptyset$) 时,

$$C^* = \max\{x_{21}^* + \tau_{12}, x_{21}^* + P_1 - x_{12}^*, x_{12}^* + \tau_{21}, x_{12}^* + P_2 - x_{21}^*\}.$$

注记 3.2.2: 若 $x_{21}^*(x_{12}^*) = 0$, 则上式中第一(三)项为 0。

引理 3.2.2: 在最优解中, 如果没有工件在 $M1$ 与 $M2$ 之间被运送, 则 $C^* = P_2$; 如果有工件在 $M1$ 与 $M2$ 之间被运送, 则 $C^* \geq \max\{\frac{P_1+P_2}{2}, \frac{\tau_{12}+P_2}{2}, p_{2n_2}\}$ 。

证明: 第一个结论由 $P_1 \leq P_2$ 显然可见。

要是有工件在 $M1$ 与 $M2$ 之间被运送, 由于 $P_1 \leq P_2$, 则必定有 \bar{J}_2 中的工件在 $M1$ 加工, 然后被运送回 $M2$ 。因此, $x_{21}^* > 0$, 由引理 3.2.1 和注记 3.2.2 可以推出 $C^* \geq \max\{x_{21}^* + \tau_{12}, P_2 - x_{21}^*\}$ 。所以

$$C^* \geq \frac{(x_{21}^* + \tau_{12}) + (P_2 - x_{21}^*)}{2} = \frac{\tau_{12} + P_2}{2}.$$

$C^* \geq \frac{P_1+P_2}{2}$ 以及 $C^* \geq p_{2n_2}$ 由平行机排序的性质显然可见。 ■

算法 A1:

1. 若 $\max\{P_1, \tau_{12}\} \geq \frac{1}{2}P_2$, 则输出 C_0 , 停止。
2. 若 $\max\{P_1, \tau_{12}\} < \frac{1}{2}P_2$ 且 $p_{2n_2} \geq \frac{1}{2}P_2$, 则

(2.1) M_2 加工 J_{2n_2} 。

(2.2) M_1 首先加工 $\bar{J}_2 \setminus \{J_{2n_2}\}$ 中的工件, 然后加工 \bar{J}_1 中的工件。

(2.3) 在时刻 $\sum_{j=1}^{n_2-1} p_{2j}$ 把 $\bar{J}_2 \setminus \{J_{2n_2}\}$ 的工件运送到 M_2 。

记得到的排序记为 σ_{11} ，其目标函数值为 C_{11} 。将 σ_0 与 σ_{11} 中目标函数值较小的那个作为输出。停止。

3. 若 $\max\{P_1, \tau_{12}\} < \frac{1}{2}P_2$ 且 $p_{2n_2} < \frac{1}{2}P_2$ ，则

(3.1) 运行算法 \mathcal{C} 。

(3.2) M_2 加工 $\bar{J}_2 \setminus S_{21}$ 中的工件。

(3.3) M_1 首先加工 S_{21} 中的工件，然后加工 \bar{J}_1 中的工件。

(3.4) 在时刻 x_{21} 把 S_{21} 中的工件运送到 M_2 。

记得到的排序记为 σ_{12} ，其目标函数值为 C_{12} 。将 σ_0 与 σ_{12} 中目标函数值较小的那个作为输出。停止。

定理 3.2.3: $\frac{C^{A1}}{C^*} \leq \frac{4}{3}$ 且这个界是紧的。

证明: 如果最优解中没有运送任何工件，则 $C^* = P_2 = C_0 \geq C^{A1}$ 。下面我们假设最优解中存在工件运送，由引理 3.2.2，我们知道

$$\begin{aligned} C^* &\geq \max\left\{\frac{P_1 + P_2}{2}, p_{2n_2}, \frac{\tau_{12} + P_2}{2}\right\} \\ &= \max\left\{p_{2n_2}, \frac{P_2}{2} + \frac{1}{2}\max\{P_1, \tau_{12}\}\right\}. \end{aligned}$$

若 $\max\{P_1, \tau_{12}\} \geq \frac{1}{2}P_2$ ，则

$$\frac{C^{A1}}{C^*} = \frac{C_0}{C^*} \leq \frac{P_2}{\frac{P_2}{2} + \frac{1}{2}\max\{P_1, \tau_{12}\}} \leq \frac{4}{3}.$$

下面假设 $\max\{P_1, \tau_{12}\} < \frac{1}{2}P_2$ 。

若 $p_{2n_2} \geq \frac{1}{2}P_2$ ，则

$$\begin{aligned} C^{A1} &= C_{11} = \max\{P_2 - p_{2n_2} + \tau_{12}, P_2 - p_{2n_2} + P_1, p_{2n_2}\} \\ &\leq \max\left\{p_{2n_2}, \frac{P_2}{2} + \max\{P_1, \tau_{12}\}\right\}. \end{aligned}$$

由于

$$\frac{\frac{P_2}{2} + \max\{P_1, \tau_{12}\}}{\frac{P_2}{2} + \frac{1}{2} \max\{P_1, \tau_{12}\}} \leq \frac{4}{3} \quad (3.1)$$

以及 $p_{2n_2} \leq C^*$, 我们可以得到 $\frac{C^{A1}}{C^*} = \frac{C_{11}}{C^*} \leq \frac{4}{3}$ 。

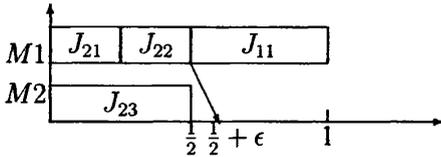
若 $p_{2n_2} < \frac{P_2}{2}$, 则 $C^{A1} = C_{12} = \max\{x_{21} + \tau_{12}, x_{21} + P_1, P_2 - x_{21}\}$ 。由注记 3.2.1, 我们知道 $\frac{P_2}{3} \leq x_{21} \leq \frac{P_2}{2}$ 。因此, $C_{12} \leq \max\{\frac{P_2}{2} + \max\{\tau_{12}, P_1\}, \frac{2}{3}P_2\}$ 。由式 (3.1) 以及

$$\frac{\frac{2}{3}P_2}{C^*} \leq \frac{\frac{2}{3}P_2}{\frac{P_1+P_2}{2}} < \frac{4}{3},$$

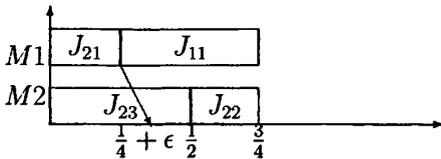
我们知道 $\frac{C^{A1}}{C^*} = \frac{C_{12}}{C^*} \leq \frac{4}{3}$ 。

综合以上, 我们可以推出 $\frac{C^{A1}}{C^*} \leq \frac{4}{3}$ 。

为了证明这个界是紧的, 考虑如下实例: $\bar{J}_1 = \{J_{11}\}$, $\bar{J}_2 = \{J_{21}, J_{22}, J_{23}\}$, $p_{11} = p_{23} = 1/2$, $p_{21} = p_{22} = 1/4$, $t_{12} = t_{21} = \epsilon \leq \frac{1}{2}$ 。运行算法 A1, 工件的加工和运输安排如下:



可以看到 $C^{A1} = 1$ 。而最优解中工件的加工和运输安排如下:



可以看到 $C^* = \frac{3}{4}$ 。所以, 当 $\epsilon \leq \frac{1}{2}$ 时, $\frac{C^{A1}}{C^*} = \frac{4}{3}$ 。 ■

3.2.2 $P2|trans - unprocessed|C_{max}$

在这个问题中, 若 \bar{J}_i 中有工件被 M_k 加工, 则它们首先要被一起运送到 M_k , 在 M_k 上完成加工后, 不需要被运回 M_i 。

注意到若 $P_1 \leq t_{21}$ 或者 $P_2 \leq t_{12}$, 这个问题就相当于带不同机器准备时间的平行机排序问题, Lee[29] 已经对该问题作了充分研究。因此接下来我们假设

$P_1 > t_{21}$ 且 $P_2 > t_{12}$ 。

引理 3.2.4: 若 $S_{21}^* \neq \emptyset (S_{12}^* \neq \emptyset)$, 则问题 $P2|trans - unprocessed|C_{max}$ 有一个最优解, 使得 $M1(M2)$ 首先加工 $\bar{J}_1 \setminus S_{12}^* (\bar{J}_2 \setminus S_{21}^*)$ 中的工件, 然后加工 $S_{21}^* (S_{12}^*)$ 中的工件。

由此引理, 我们知道, 当 $x_{21}^* > 0 (S_{21}^* \neq \emptyset)$ 且 $x_{12}^* > 0 (S_{12}^* \neq \emptyset)$ 时,

$$C^* = \max\{t_{21} + x_{21}^*, P_1 - x_{12}^* + x_{21}^*, t_{12} + x_{12}^*, P_2 - x_{21}^* + x_{12}^*\}.$$

注记 3.2.3: 若 $x_{21}^*(x_{12}^*) = 0$, 则上式中第一(三)项为 0。

引理 3.2.5: 在最优解中, 如果没有工件在 $M1$ 与 $M2$ 之间被运送, 则 $C^* = P_2$; 如果有工件在 $M1$ 与 $M2$ 之间被运送, 则 $C^* \geq \max\{\frac{P_1+P_2}{2}, \frac{t_{21}+P_2}{2}, p_{2n_2}\}$ 。

证明: 第一个结论由 $P_1 \leq P_2$ 显然可见。

要是有工件在 $M1$ 与 $M2$ 之间被运送, 由于 $P_1 \leq P_2$, 则必定有 \bar{J}_2 中的工件在 $M1$ 加工, 然后被运送到 $M2$ 。因此, $x_{21}^* > 0$, 由引理 3.2.4 和注记 3.2.3 可以推出 $C^* \geq \max\{x_{21}^* + t_{21}, P_2 - x_{21}^*\}$ 。所以

$$C^* \geq \frac{(x_{21}^* + t_{21}) + (P_2 - x_{21}^*)}{2} = \frac{t_{21} + P_2}{2}.$$

$C^* \geq \frac{P_1+P_2}{2}$ 以及 $C^* \geq p_{2n_2}$ 由平行机排序的性质显然可见。 ■

算法 A2:

1. 若 $\max\{P_1, t_{21}\} \geq \frac{1}{2}P_2$, 输出 σ_0 , 停止。
2. 若 $\max\{P_1, t_{21}\} < \frac{1}{2}P_2$ 且 $p_{2n_2} \geq \frac{1}{2}P_2$, 则
 - (2.1) M_1 从零时刻开始加工 \bar{J}_1 中的工件。
 - (2.2) M_2 加工 J_{2n_2} 。
 - (2.3) 在零时刻把 $\bar{J}_2 \setminus \{J_{n_2}\}$ 中的工件运送到 M_1 , M_1 从时刻 $\max\{P_1, t_{21}\}$ 开始加工这些工件。

记得到的排序为 σ_{21} , 其目标函数值为 C_{21} 。将 σ_0 与 σ_{21} 中目标函数值较小的那个作为输出。停止。

3. 若 $\max\{P_1, t_{21}\} < \frac{1}{2}P_2$ 且 $p_{2n_2} < \frac{1}{2}P_2$, 则

(3.1) 运行算法 C.

(3.2) M_1 从零时刻开始加工 \bar{J}_1 中的工件。

(3.3) M_2 加工 $\bar{J}_2 \setminus S_{21}$ 中工件。

(3.4) 在零时刻把 S_{21} 中的工件运送到 M_1 , M_1 从时刻 $\max\{P_1, t_{21}\}$ 开始加工这些工件。

记得到的排序为 σ_{22} , 其目标函数值为 C_{22} . 将 σ_0 与 σ_{22} 中目标函数值较小的那个作为输出. 停止.

定理 3.2.6: $\frac{C^{A2}}{C^*} \leq \frac{4}{3}$ 且这个界是紧的.

证明: 如果最优解中没有运送任何工件, 则 $C^* = P_2 = C_0 \geq C^{A2}$. 下面我们假设最优解中存在工件运送, 由引理 3.2.5, 我们知道

$$C^* \geq \max\left\{\frac{P_1 + P_2}{2}, p_{2n_2}, \frac{t_{21} + P_2}{2}\right\} = \max\left\{p_{2n_2}, \frac{P_2}{2} + \frac{1}{2}\max\{P_1, t_{21}\}\right\}.$$

若 $\max\{P_1, t_{21}\} \geq \frac{1}{2}P_2$, 则 $C^{A2} = C_0 = P_2$.

若 $\max\{P_1, t_{21}\} < \frac{1}{2}P_2$ 且 $p_{2n_2} \geq \frac{1}{2}P_2$, 则

$$\begin{aligned} C^{A2} = C_{21} &= \max\{P_2 - p_{2n_2} + \max\{P_1, t_{21}\}, p_{2n_2}\} \\ &\leq \max\left\{p_{2n_2}, \frac{P_2}{2} + \max\{P_1, t_{21}\}\right\}. \end{aligned}$$

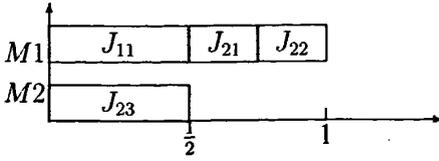
由与定理 3.2.3 类似地可以证明 $\frac{C^{A2}}{C^*} \leq \frac{4}{3}$.

若 $\max\{P_1, t_{21}\} < \frac{1}{2}P_2$ 且 $p_{2n_2} < \frac{1}{2}P_2$, 则

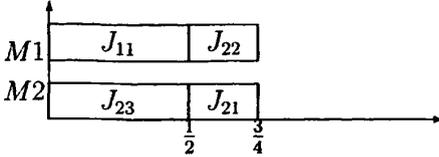
$$\begin{aligned} C^{A2} = C_{22} &= \max\{x_{21} + \max\{P_1, t_{21}\}, P_2 - x_{21}\} \\ &\leq \max\left\{\frac{P_2}{2} + \max\{t_{21}, P_1\}, \frac{2}{3}P_2\right\}. \end{aligned}$$

由与定理 3.2.3 类似地可以证明 $\frac{C^{A2}}{C^*} \leq \frac{4}{3}$.

综合以上各种情形, 我们可以证明 $\frac{C^{A2}}{C^*} \leq \frac{4}{3}$. 为了证明这个界是紧的, 考虑如下实例: $\bar{J}_1 = \{J_{11}\}$, $\bar{J}_2 = \{J_{21}, J_{22}, J_{23}\}$, $p_{11} = p_{23} = 1/2$, $p_{21} = p_{22} = 1/4$, $t_{12} = t_{21} = \epsilon \leq \frac{1}{2}$. 运行算法 A2, 工件的加工和运输安排如下:



可以看到 $C^{A2} = 1$ 。而在最优解中，工件的加工和运输安排如下：



我们可以看到 $C^* = \frac{3}{4}$ ，并且当 $\epsilon \leq \frac{1}{2}$ 时， $\frac{C^{A2}}{C^*} = \frac{4}{3}$ 。 ■

3.2.3 $P2|trans - two\ way|C_{max}$

在这个问题中，若 \bar{J}_i 中有工件被 M_k 加工，则它们首先要被一起运送到 M_k ，在 M_k 上完成加工后，还需要被一起运回 M_i 。

引理 3.2.7: 当 $x_{21}^* > 0$ 且 $x_{12}^* > 0$ 时， $C^* \geq \max\{\frac{P_1+P_2}{2}, p_{2n_2}, t_{21} + x_{21}^* + \tau_{12}, t_{12} + x_{12}^* + \tau_{21}, P_1 - x_{12}^* + x_{21}^*, P_2 - x_{21}^* + x_{12}^*\}$ 。

注记 3.2.4: 若 $x_{21}^*(x_{12}^*) = 0$ ，则上式中第三（四）项为 0。

引理 3.2.8: 在最优解中，如果没有工件在 $M1$ 与 $M2$ 之间被运送，则 $C^* = P_2$ ；如果有工件在 $M1$ 与 $M2$ 之间被运送，则 $C^* \geq \max\{\frac{P_1+P_2}{2}, p_{2n_2}, \frac{P_2}{2} + \frac{1}{2}(t_{21} + \tau_{12})\}$ 。

证明: 第一个结论由 $P_1 \leq P_2$ 可以推出。若有工件在 $M1$ 与 $M2$ 之间被运送，因为 $P_1 \leq P_2$ ，则必定有 \bar{J}_2 中的工件在 $M1$ 加工，然后被运送到 $M2$ 。因此， $S_{21}^* \neq \emptyset$ 以及 $x_{21}^* > 0$ ，所以

$$\begin{aligned} C^* &\geq \max\{t_{21} + \tau_{12} + x_{21}^*, P_2 - x_{21}^*\} \\ &\geq \frac{(t_{21} + \tau_{12} + x_{21}^*) + (P_2 - x_{21}^*)}{2} \\ &= \frac{t_{21} + \tau_{12} + P_2}{2}. \end{aligned}$$

$C^* \geq \frac{P_1+P_2}{2}$ 以及 $C^* \geq p_{2n_2}$ 由平行机排序的性质显然可见。 ■

算法 A3:

1. 若 $P_1 \geq \frac{1}{3}P_2$, 或者 $t_{21} + \tau_{12} \geq \frac{1}{2}P_2$, 输出 σ_0 , 停止。
2. 若 $P_1 < \frac{1}{3}P_2$, $t_{21} + \tau_{12} < \frac{1}{2}P_2$ 且 $p_{2n_2} \geq \frac{1}{2}P_2$, 则
 - (2.1) M_2 加工 J_{2n_2} 。
 - (2.2) 在零时刻把 $\bar{J}_2 \setminus \{J_{n_2}\}$ 中的工件运送到 M_1 。
 - (2.3) 若 $t_{21} \leq \tau_{12}$, 让 M_1 在时间段 $[0, t_{21}]$ 内空闲。 M_1 从时刻 t_{21} 开始加工 $\bar{J}_2 \setminus \{J_{n_2}\}$ 中的工件, 然后加工 \bar{J}_1 中的工件。若 $t_{21} > \tau_{12}$, 则 M_1 从零时刻开始加工 \bar{J}_1 中的工件, 然后从时刻 $\max\{P_1, t_{21}\}$ 开始加工 $\bar{J}_2 \setminus \{J_{n_2}\}$ 中的工件。
 - (2.4) 一旦 M_1 完成加工 $\bar{J}_2 \setminus \{J_{n_2}\}$ 中的工件, 就把这些工件运送回 M_2 。

记得到的排序为 σ_{31} , 其目标函数值为 C_{31} 。将 σ_0 与 σ_{31} 中目标函数值较小的那个作为输出。停止。
3. 若 $P_1 < \frac{1}{3}P_2$, $t_{21} + \tau_{12} < \frac{1}{2}P_2$ 且 $p_{2n_2} < \frac{1}{2}P_2$, 则
 - (3.1) 运行算法 C。
 - (3.2) M_2 加工 $\bar{J}_2 \setminus S_{21}$ 中的工件。
 - (3.3) 在零时刻把 S_{21} 中的工件运送到 M_1 。
 - (3.4) 若 $t_{21} \leq \tau_{12}$, 让 M_1 在时间段 $[0, t_{21}]$ 内空闲。 M_1 从时刻 t_{21} 开始加工 S_{21} 中的工件, 然后加工 \bar{J}_1 中的工件。若 $t_{21} > \tau_{12}$, 则 M_1 从零时刻开始加工 \bar{J}_1 中的工件, 然后从时刻 $\max\{P_1, t_{21}\}$ 开始加工 S_{21} 中的工件。
 - (3.5) 一旦 M_1 完成加工 S_{21} 中的工件, 就把这些工件运送回 M_2 。

记得到的排序为 σ_{32} , 其目标函数值为 C_{32} 。将 σ_0 与 σ_{32} 中目标函数值较小的那个作为输出。停止。

定理 3.2.9: $\frac{C^{A3}}{C^*} \leq \frac{3}{2}$ 且这个界是紧的。

证明: 若最优解中没有运送任何工件, 则 $C^* = P_2 \geq C^{A3}$ 。因此, 下面我们假设最优解中存在工件运送。由引理 3.2.8, 我们知道 $C^* \geq \max\{\frac{P_1+P_2}{2}, p_{2n_2}, \frac{P_2}{2} + \frac{1}{2}(t_{21} + \tau_{12})\}$ 。

若 $P_1 \geq \frac{1}{3}P_2$, 则

$$\frac{C^{A3}}{C^*} = \frac{C_0}{C^*} = \frac{P_2}{\frac{P_1+P_2}{2}} \leq \frac{3}{2}.$$

若 $t_{21} + \tau_{12} \geq \frac{1}{2}P_2$, 则

$$\frac{C^{A3}}{C^*} = \frac{C_0}{C^*} \leq \frac{P_2}{\frac{P_2}{2} + \frac{1}{2}(t_{21} + \tau_{12})} \leq \frac{4}{3} < \frac{3}{2}.$$

下面我们假设 $P_1 < \frac{1}{3}P_2$ 且 $t_{21} + \tau_{12} < \frac{1}{2}P_2$.

若 $p_{2n_2} \geq \frac{1}{2}P_2$, 则当 $t_{21} \leq \tau_{12}$ 时 $C_{31} = \max\{t_{21} + P_2 - p_{2n_2} + P_1, t_{21} + P_2 - p_{2n_2} + \tau_{12}, p_{2n_2}\}$; 当 $t_{21} > \tau_{12}$ 时 $C_{31} = \max\{P_2 - p_{2n_2} + \max\{P_1, t_{21}\} + \tau_{12}, p_{2n_2}\}$. 因此, $C_{31} \leq \max\{\frac{1}{2}P_2 + \max\{t_{21} + \tau_{12}, P_1 + \frac{1}{2}(t_{21} + \tau_{12})\}, p_{2n_2}\}$.

因为有

$$\begin{aligned} \frac{\frac{1}{2}P_2 + P_1 + \frac{1}{2}(t_{21} + \tau_{12})}{\max\{\frac{P_1+P_2}{2}, \frac{P_2}{2} + \frac{1}{2}(t_{21} + \tau_{12})\}} &\leq \frac{\frac{P_2}{2} + \frac{1}{2}(t_{21} + \tau_{12})}{\frac{1}{2}(t_{21} + \tau_{12}) + \frac{P_2}{2}} + \frac{P_1}{\frac{P_1+P_2}{2}} \\ &\leq \frac{3}{2}, \end{aligned} \quad (3.2)$$

$$\begin{aligned} \frac{t_{21} + \tau_{12} + \frac{P_2}{2}}{\frac{P_2}{2} + \frac{1}{2}(t_{21} + \tau_{12})} &\leq 1 + \frac{\frac{1}{2}(t_{21} + \tau_{12})}{\frac{P_2}{2} + \frac{1}{2}(t_{21} + \tau_{12})} \\ &\leq \frac{4}{3} < \frac{3}{2}, \end{aligned} \quad (3.3)$$

以及 $p_{2n_2} \leq C^*$, 我们可以推出 $\frac{C^{A3}}{C^*} = \frac{C_{31}}{C^*} \leq \frac{4}{3}$.

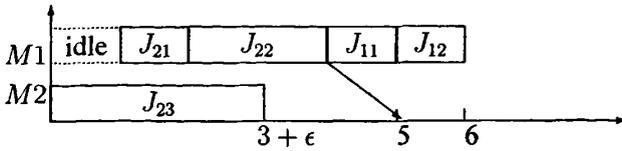
若 $p_{2n_2} < \frac{P_2}{2}$, 则当 $t_{21} \leq \tau_{12}$ 时, $C_{32} = \max\{t_{21} + x_{21} + P_1, t_{21} + x_{21} + \tau_{12}, P_2 - x_{21}\}$; 当 $t_{21} > \tau_{12}$ 时, $C_{32} = \max\{x_{21} + \tau_{12} + \max\{P_1, t_{21}\}, P_2 - x_{21}\}$. 结合注记 3.2.1, 我们可以得到 $C_{32} \leq \max\{\frac{P_2}{2} + \max\{P_1 + \frac{1}{2}(t_{21} + \tau_{12}), t_{21} + \tau_{12}\}, \frac{2P_2}{3}\}$. 由式 (3.2) 和 (3.3) 以及

$$\frac{\frac{2}{3}P_2}{C^*} \leq \frac{\frac{2}{3}P_2}{\frac{P_1+P_2}{2}} < \frac{4}{3} < \frac{3}{2},$$

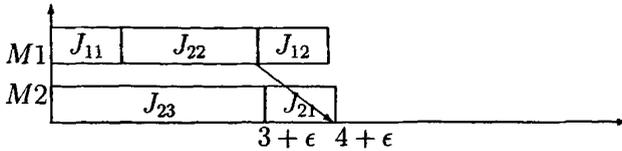
我们可以得到 $\frac{C^{A3}}{C^*} = \frac{C_{32}}{C^*} \leq \frac{3}{2}$.

综合以上各种情形, 我们可以推出 $\frac{C^{A3}}{C^*} \leq \frac{3}{2}$. 为了证明这个界是紧的, 考虑如下实例: $\bar{J}_1 = \{J_{11}, J_{12}\}$, $\bar{J}_2 = \{J_{21}, J_{22}, J_{23}\}$, $p_{11} = p_{12} = 1$, $p_{21} = 1$, $p_{22} =$

2, $p_{23} = 3 + \epsilon$, $t_{12} = t_{21} = \tau_{12} = \tau_{21} = 1$ 。运行算法 $A3$, 工件的加工和运输安排如下:



我们可以看到, $C^{A3} = 6$ 。而最优解中, 工件的加工和运输安排如下:



我们看到 $C^* = 4 + \epsilon$, 并且当 $\epsilon \rightarrow 0$ 时, $\frac{C^{A3}}{C^*} \rightarrow \frac{3}{2}$ 。 ■

3.3 $P2|trans - two\ way|C_{max}$ 的动态规划算法

文献 [46] 应用了添加构造 (adding structure) 的方法给出了一系列排序问题的 $FPTAS$ 。在这里, 我们也利用这个方法提出问题 $P2|trans - two\ way|C_{max}$ 的动态规划最优算法, 并且将此算法简化后可以得到 $P2|trans-completed|C_{max}$ 与 $P2|trans - unprocessed|C_{max}$ 的动态规划算法。

我们把 \bar{J}_1 和 \bar{J}_2 中的工件一起排序, 记为 $\{J_1, J_2, \dots, J_n\}$, 它们的加工时间为 $\{p_1, p_2, \dots, p_n\}$ 。假设 $J_1 \in \bar{J}_2$, 并且所有的参数都是整数。

不难发现, 存在 $P2|trans - two\ way|C_{max}$ 的一个最优解, 使得 $M_i (i = 1, 2)$ 加工的工件按加工顺序可以分成三组 $\{B_{i1}, B_{i2}, B_{i3}\}$, 每组中的工件被连续加工。 B_{i1} 和 B_{i3} 中的工件来自 \bar{J}_i , 而 B_{i2} 中的工件来自 $\bar{J}_k, k \neq i$ 。若 $B_{i2} \neq \emptyset$, 则 M_i 在时刻 t_{ki} 之前不能加工 B_{i2} 中的工件, 而且在 B_{i1} 中至多有一个工件的完工时间大于或等于 t_{ki} , 否则, 至少存在 B_{i1} 中的一个工件, 其开工时间在 t_{ki} 之后, 我们可以把这个工件放到 B_{i3} 中, 而这样并不会增加目标函数值。

我们下面提出一个动态规划算法 DPT , 它可以求出问题 $P2|trans - two\ way|C_{max}$ 的最优解。在算法 DPT 的第 k 步, 生成一个六维向量的集合 $V_k, 1 \leq k \leq n$, 其中的向量 $[L_1, L_2, L_3, L_4, L_5, L_6]$ 对应由前 k 个工件组成的部分可行解的目标函数值, 前三个分量代表 $\{B_{11}, B_{12}, B_{13}\}$ 中工件的完工时间, 而后三个分量分别代表 $\{B_{21}, B_{22}, B_{23}\}$ 中工件的完工时间。我们还构造一个向

量集 X ，在算法的第 k 步，会对 X 中的每个向量进行更新。 X 中每个向量 $[X_1, X_2, X_3, X_4, X_5, X_6]$ 代表一个前 k 个工件的部分排序：前三个变量分别代表 $\{B_{11}, B_{12}, B_{13}\}$ 中工件的集合，后三个变量分别代表 $\{B_{21}, B_{22}, B_{23}\}$ 中工件的集合。在算法的最后，会输出近似解的目标函数值以及相应的排序。

算法 *DPT*:

1. $k = 1, V_1 = \{[0, 0, 0, p_1, 0, 0], [0, 0, 0, 0, 0, p_1 + t_{21}], [0, p_1 + t_{12} + \tau_{21}, 0, 0, 0, 0]\}$, $X = \{[\emptyset, \emptyset, \emptyset, \{1\}, \emptyset, \emptyset], [\emptyset, \emptyset, \emptyset, \emptyset, \emptyset, \{1\}], [\emptyset, \{1\}, \emptyset, \emptyset, \emptyset, \emptyset]\}$.
2. $k = k + 1, V_k = \emptyset$, 对 V_{k-1} 中的每个向量 $[a, b, c, d, e, f]$, 假设其对应于 $[X_1, X_2, X_3, X_4, X_5, X_6] \in X$, 令

$$x_B = \begin{cases} 0, & \{a + p_k \leq t_{21}\} \text{ 或 } \{a + p_k > t_{21}, b = 0\} \\ a + p_k - t_{21}, & a + p_k > t_{21}, b \neq 0 \end{cases},$$

$$x_C^1 = \begin{cases} 0, & \{a + p_k \leq t_{21}\} \text{ 或 } \{a + p_k > t_{21}, c = 0\} \\ a + p_k - t_{21}, & a + p_k > t_{21}, c \neq 0 \end{cases},$$

$$x_C^2 = \begin{cases} p_k, & c \neq 0 \\ b - \tau_{12} + p_k, & b \neq 0, c = 0 \\ \max\{a, t_{21}\} + p_k, & b = 0, c = 0 \end{cases},$$

$$x_E = \begin{cases} \max\{d, t_{12}\} + p_k + \tau_{21}, & e = 0 \\ p_k, & e \neq 0 \end{cases}, x_F = \begin{cases} 0, & f = 0 \\ p_k, & f \neq 0 \end{cases},$$

$$y_E = \begin{cases} 0, & \{d + p_k \leq t_{12}\} \text{ 或 } \{d + p_k > t_{12}, e = 0\} \\ d + p_k - t_{12}, & d + p_k > t_{12}, e \neq 0 \end{cases},$$

$$y_F^1 = \begin{cases} 0, & \{d + p_k \leq t_{12}\} \text{ 或 } \{d + p_k > t_{12}, f = 0\} \\ d + p_k - t_{12}, & d + p_k > t_{12}, f \neq 0 \end{cases},$$

$$y_F^2 = \begin{cases} p_k, & f \neq 0 \\ e - \tau_{21} + p_k, & e \neq 0, f = 0 \\ \max\{d, t_{12}\} + p_k, & e = 0, f = 0 \end{cases},$$

$$y_B = \begin{cases} \max\{a, t_{21}\} + p_k + \tau_{12}, & b = 0 \\ p_k, & b \neq 0 \end{cases}, y_C = \begin{cases} 0, & c = 0 \\ p_k, & c \neq 0 \end{cases}.$$

若 $J_k \in \overline{J_1}$,

(i) 若 $a < t_{21}$, 把 $[a + p_k, b + x_B, c + x_C^1, d, e, f]$ 放入 V_k , 把 $[X_1 \cup \{k\}, X_2, X_3, X_4, X_5, X_6]$ 放入 X 。

(ii) 把 $[a, b, c + x_C^2, d, e, f]$ 放入 V_k , 把 $[X_1, X_2, X_3 \cup \{k\}, X_4, X_5, X_6]$ 放入 X 。

(iii) 把 $[a, b, c, d, e + x_E, f + x_F]$ 放入 V_k , 把 $[X_1, X_2, X_3, X_4, X_5 \cup \{k\}, X_6]$ 放入 X 。

若 $J_k \in \overline{J_2}$,

(i) 若 $d < t_{12}$, 把 $[a, b, c, d + p_k, e + y_E, f + y_F^1]$ 放入 V_k , 把 $[X_1, X_2, X_3, X_4 \cup \{k\}, X_5, X_6]$ 放入 X 。

(ii) 把 $[a, b, c, d, e, f + y_F^2]$ 放入 V_k , 把 $[X_1, X_2, X_3, X_4, X_5, X_6 \cup \{k\}]$ 放入 X 。

(iii) 把 $[a, b + y_B, c + y_C, d, e, f]$ 放入 V_k , 把 $[X_1, X_2 \cup \{k\}, X_3, X_4, X_5, X_6]$ 放入 X 。

将 $[X_1, X_2, X_3, X_4, X_5, X_6]$ 从 X 中除去。

3. 若 $b = 0$ 且 $a < t_{21}$, 则重置 $c = c - t_{21} + a$ 。

若 $e = 0$ 且 $d < t_{12}$, 则重置 $f = f - t_{12} + d$ 。

输出 V_n 中的向量 $[a, b, c, d, e, f]$ 以及对应的 X 中的向量, 使得

$$\max\{a, b, c, d, e, f\} = \min_{Z=\{z_1, z_2, z_3, z_4, z_5, z_6\} \in V_n} \max\{z_1, z_2, z_3, z_4, z_5, z_6\}.$$

因为 V_k 中所有向量的各个分量都是界于 0 与 $L \triangleq P_1 + P_2 + \max\{t_{12} + \tau_{21}, t_{21} + \tau_{12}\}$ 之间的整数, 所以 V_k 中所含的向量的个数不超过 $O(L^6)$ 。可以看出算法 DPT 的时间计算复杂性与 $\sum_{k=1}^n |V_k|$ 成正比, 因此, DPT 在伪多项式时间 $O(nL^6)$ 内可以得到最优解。

注记 3.3.1: 对问题 $P2|trans - completed|C_{max}$, 有 $t_{12} = t_{21} = 0$, 并且对 V_k 中任意向量 $[a, b, c, d, e, f]$, 有 $a = d = 0$ 。在算法 DPT 的第 2 步, 令 $t_{12} = t_{21} = 0$ 和 $a = d = 0$, 我们只需要定义如下四个变量: x_C^2, x_E, y_B, y_F^2 。另外, $J_k \in \overline{J_1}$

或 $J_k \in \bar{J}_2$ 时的步骤 (i) 不会运行。

对问题 $P2|trans - unprocessed|C_{max}$, 有 $\tau_{12} = \tau_{21} = 0$, 且对 V_k 中任意向量 $[a, b, c, d, e, f]$, 有 $c = f = 0$ 。但是, 仅仅令算法 DPT 中 $\tau_{12} = \tau_{21} = 0$ 和 $c = f = 0$ 不可行, 因为若有 \bar{J}_1 (\bar{J}_2) 中的工件被 M_1 (M_2) 加工, 无论 “ $a < t_{21}(d < t_{12})$ ” 与否, 这个工件被加入到 B_{11} (B_{21}) 中。在算法 DPT 的第 2 步, 我们需要定义如下四个变量: x_B, x_E, y_B, y_E 。

$$x_B = \begin{cases} 0, & b = 0 \text{ 或 } \{a + p_k \leq t_{21}, b \neq 0\} \\ p_k + a - t_{21}, & a \leq t_{21}, a + p_k > t_{21}, b \neq 0 \\ p_k, & a > t_{21}, b \neq 0 \end{cases},$$

$$x_E = \begin{cases} \max\{d, t_{12}\} + p_k, & e = 0 \\ p_k, & e \neq 0 \end{cases}, y_B = \begin{cases} \max\{a, t_{21}\} + p_k, & b = 0 \\ p_k, & b \neq 0 \end{cases},$$

$$y_E = \begin{cases} 0, & e = 0 \text{ 或 } \{d + p_k \leq t_{12}, e \neq 0\} \\ p_k + a - t_{12}, & d \leq t_{12}, d + p_k > t_{12}, e \neq 0 \\ p_k, & e > t_{12}, b \neq 0 \end{cases}.$$

$J_k \in \bar{J}_1$ 或 $J_k \in \bar{J}_2$ 时的步骤 (ii) 不运行, 而无论 “ $a < t_{21}(d < t_{12})$ ” 与否, $J_k \in \bar{J}_1$ ($J_k \in \bar{J}_2$) 的步骤 (i) 都要运行。另外, 步骤 (iii) 中不需要对 “ c ” 或 “ f ” 重置。

不难发现, $P2|trans - completed|C_{max}$ 和 $P2|trans - unprocessed|C_{max}$ 的动态规划的计算复杂性都是 $O(nL^4)$ 。

3.4 综合考虑不同加工中心运作的益处

在这一节, 我们来讨论问题 $P2|trans - two\ way|C_{max}$ 中在两台机器之间运送工件有什么好处。

记 $C(t_{12}, t_{21}, \tau_{12}, \tau_{21})$ 是运输时间为 $\{t_{12}, t_{21}, \tau_{12}, \tau_{21}\}$ 的实例的目标函数的最小值。显然可见, $C(t_{12}, t_{21}, \tau_{12}, \tau_{21}) \leq C_0$, 因为每个机器只加工原本属于自己的工件是可行的。我们用下面的式子来定义综合考虑两台机器运作带来的益

处:

$$S(t_{12}, t_{21}, \tau_{12}, \tau_{21}) = 1 - \frac{C(t_{12}, t_{21}, \tau_{12}, \tau_{21})}{C_0}.$$

在最坏情况下, 有工件运输的排序与原来的排序目标函数值相同, 接下来, 我们来研究综合考虑两台机器运作的益处的上界 $\bar{S} = \max_{t_{12}, t_{21}, \tau_{12}, \tau_{21}} S(t_{12}, t_{21}, \tau_{12}, \tau_{21})$ 。可以看到, 运输时间的减少不会增加目标函数值, 因此, 当 $t_{12} = t_{21} = \tau_{12} = \tau_{21} = 0$ 时, 综合考虑两台机器运作的益处达到最大。这个结论可以用下面这个引理表示。

引理 3.4.1: $\bar{S} = S(0, 0, 0, 0)$ 。

对一个给定的工件集合, 我们要把带工件运输的排序的目标函数值与无工件运输的排序的目标函数值作比较, 而后的最大值通常在所有工件都来自同一台机器时达到。

定理 3.4.2: 综合考虑两台机器运作的益处最多可达 50%, 并且这个界是紧的。

证明: 由引理 3.4.1, 我们知道对任意的工件集合, 当 $t_{12} = t_{21} = \tau_{12} = \tau_{21} = 0$ 时, 带运输的排序的益处最大。另一方面, 当 $\bar{J}_1 \cup \bar{J}_2$ 中所有工件都属于同一台机器, 即 $\bar{J}_1 = \emptyset$ 或 $\bar{J}_2 = \emptyset$ 时, C_0 达到最大值。因此, $C_0 = p_1 + p_2 + \cdots + p_n$, 而带运输的排序的目标函数值 $\geq \frac{p_1 + p_2 + \cdots + p_n}{2} = \frac{C_0}{2}$ 。所以,

$$\bar{S} = S(0, 0, 0, 0) = 1 - \frac{C(0, 0, 0, 0)}{C_0} \leq 1 - \frac{1}{2} = 50\%.$$

为了证明这个界是紧的, 我们考虑如下实例:

$$\bar{J}_1 = \{J_{11}, J_{12}\}, \bar{J}_2 = \emptyset, p_{11} = p_{12} = 1, t_{12} = t_{21} = \tau_{12} = \tau_{21} = 0.$$

综合考虑两台机器的运作时, 每个机器加工一个工件。

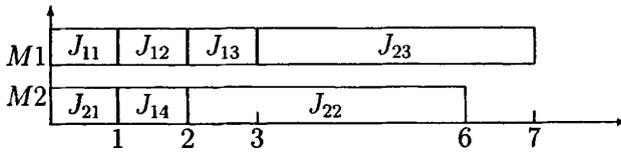
从以上分析我们可以看到, 综合考虑两台机器的运作对优化整个系统的运行大有益处。直观看来, 这样会使一台机器加工原本不属于本身的工件, 那么整个系统的目标函数值的降低是否一定要以某台机器的目标函数值的增加为代价呢? 下面这个例子说明了整个系统的优化未必要增加其中的机器上工件的完

工时间。

$$\bar{J}_1 = \{J_{11}, J_{12}, J_{13}, J_{14}\}, \bar{J}_2 = \{J_{21}, J_{22}, J_{23}\}, p_{11} = p_{12} = p_{13} = p_{14} = 1,$$

$$p_{21} = 1, p_{22} = p_{23} = 4, t_{12} = t_{21} = \tau_{12} = \tau_{21} = 1.$$

若两台机器之间没有工件运输， \bar{J}_1 中工件的完工时间为 4， \bar{J}_2 中工件的完工时间为 9。在下面这个排序中，我们综合考虑两台机器的运作，可以看到 \bar{J}_1 中工件的完工时间为 3， \bar{J}_2 中工件的完工时间为 8，这两个值比在之前的排序中都要好。



第四章 带承诺到货时间的极小化运输费用的排序问题

4.1 引言

在现代制造行业中，许多企业都根据顾客的订单来生产（make to order），这样既可以减少库存成本，也可以根据外部需求的变化及时调整企业的生产。根据顾客的订单来生产一般都需要在规定的时间内完成加工，然后运送给顾客，通常运输工作由第三方物流（third-party logistics，以下记作 3PL）公司来承担，例如 FedEx 或者 UPS。3PL 公司会根据运送的时间例如，一天、两天或三天，提供不同的运输模式（为方便叙述，本文中的运输模式等同于运输时间），而运输费用与运送时间成反比，与送货量成正比。生产企业为了达到顾客对订单的及时性要求，可以采用两种发货方式：承诺发货时间（commit to ship）和承诺到货时间（commit to delivery）。承诺发货时间指生产企业接到订单时承诺在某一时间之前把货物从企业发送出去，交由 3PL 公司来运送，而顾客在发出订单时，必须自己选择运送模式以及付给 3PL 公司运送的费用。承诺到货时间指企业接到订单时会承诺在某一时间之前把货物送到顾客处。这种发货方式下，生产企业选择各个订单的运送模式以及付给 3PL 公司运送费用。

采用承诺发货时间的企业只需要安排好企业内部的生产调度，不用去安排由顾客自己选择的运送过程。而采用承诺到货时间的企业必须同时考虑生产与运送问题，如果一个订单的任务提前完成了，则可以选择较长的运送时间来降低运送成本，若一个订单的完成时间与承诺的时间较为接近，则必须采用较快的运输模式以满足顾客的要求。

本章中，我们研究根据订单生产的企业在承诺到货时间的模式下如何安排工件的加工与运输问题。在开动一条加工线之初，企业收到了来自不同顾客的订单，每个订单要求一定数量的同一产品，并且都有各自的承诺到货时间。企业要首先完成加工这些订单然后把它们交给 3PL 公司运送给顾客。我们假设 3PL 公司对一个订单索取的费用与订单中产品的数量成线性递增关系，与运送需要的时间成线性递减关系。3PL 公司会在每天的固定时刻，例如下午三点，来提取在当天完成加工的订单。我们要做的工作是安排一个生产产品的排序并且给每个订单安排一种运输模式，使得所有订单都在各自的承诺时间之前运送到顾客处，而且付给 3PL 公司的运送费用尽可能的少。

参考以前的研究带运输问题的排序的文章，除了文献 [49] 和 [47]，大多数文章都假设对完成加工的产品只有一种运输模式，不论其完工时间是早或者晚，而这一因素在本章的问题中是很关键的。Wang 和 Lee [49] 研究了两个带两种运输模式的问题。第一个问题与本章的类似，除了 (1) 只有两种运输模式，而我们的问题没有这个限制条件。(2) 一个订单完成加工后立即被运送走，而在本章的问题中，3PL 公司只在每天的一个固定时刻来提取完成加工的工件。若有订单在这个时刻之前完成了加工，就必须等一段时间才能被运送走。Wang 和 Lee 证明了此问题至少是 NP - 难的，同时对其中一类特殊情形给出了伪多项式时间动态规划算法。文献 [49] 中的第二个问题是极小化加权总误工时间和总运输费用。因为经典的单台机极小化加权总误工时间问题是强 NP - 难的 [32]，所以这个问题是强 NP - 难的。他们对这个问题给出了一个分支定界算法。

文献 [47] 中研究了与本章类似的问题，只是每个订单的运送费用与订单中产品的数量成线性递增关系，而与运送时间成凸减关系。他们研究了两种情形，第一种情形允许订单的部分运送，这表示一个订单中的产品可以作为多个小订单的组合，在不同的时间分开运送。例如，某个顾客定了十台电脑，生产线在安排时，有三台电脑在第一天完工，而另外的七台在第二天完工，那么 3PL 公司在第一天可以先提走那三台，剩下的七台第二天再提走。这种情形可以很简单的在多项式时间内求到最优解：按照承诺到货时间的不减顺序加工订单中的产品，3PL 公司在每天固定的时刻把已经完成加工的产品（可以是一个订单的部分产品）运送出去。文献 [47] 中的第二类情形不允许订单的部分运送，这表示一个订单只有当其中所有产品都完成了加工才可以运送出去。在上面那个例子中，在第一天完工的三台电脑必须等到第二天其余的电脑组装完了一起运送出去。Stecke 和 Zhao [47] 证明了这类情形是强 NP - 难的，并给出了一个启发式算法。

在本章，我们考虑的问题不允许订单的部分运送，每个订单的运送费用与订单中产品的数量成正比，与运送时间成线性递减关系，这是文献 [47] 中第二类情形的一个特殊问题。从文献 [47] 证明强 NP - 难的过程可以看出，本章的问题也是强 NP - 难的。

在本章的第二节，我们给出问题的具体描述。在第三节，我们提出一个多项式时间近似算法，并且分析其最坏情况界。

4.2 问题的具体描述

在开启一条计划生产 m 天的生产线之初, 企业接受了 n 个订单 $N = \{O_1, O_2, \dots, O_n\}$ 。 N 中每个订单 O_i 包含 Q_i 个产品, O_i 的承诺到货时间为 d_i 。与文献 [47] 类似, 我们假设所有产品的重量和加工时间都一样, 这在电脑制造业是很普遍的, 因为不同的电脑的区别在于其中的配件的不同, 而把不同的配件组装成型号不同的电脑需要的时间是一样的, 其重量也相同。生产线每天能生产 c 个产品, 我们假设对 $i = 1, 2, \dots, n$, $Q_i \leq c$ 并且 $\sum_{i=1}^n Q_i > c$ 。

一个 3PL 公司在每天生产结束后来提取完工的订单, 把它们运送给顾客。加工企业给每个订单选择运送时间, 并付给 3PL 公司相应的运送费用。运送时间可以是 1 天, 2 天, …… , 直至 m 天。若订单的运送天数是 x , 则其运送费用为 $G(x, Q_i) = Q_i(\alpha - \beta x)$, 其中 $\alpha > 0$, $\beta > 0$, 并且满足

$$\alpha - \beta(d_i - 1) > 0, \quad i = 1, 2, \dots, n. \quad (4.1)$$

这个不等式代表即使运送时间最长 (第一天就完成了加工, 运送时间为 $(d_i - 1)$ 天), 运输费用最小, 其费用最起码是大于零的。

为了方便描述问题, 我们定义最迟完工时间 $d_i^0 = d_i - 1$, $i = 1, 2, \dots, n$, 这代表即使用最快的运输模式 (运输时间为一天), 订单 O_i 必须在这一天完成加工。为了保证可行解 (所有订单在最迟完工时间之前完成加工) 的存在, 我们假设

$$\sum_{i \in \{i | d_i^0 \leq j\}} Q_j \leq c \cdot j, \quad j = 1, 2, \dots, m. \quad (4.2)$$

注意到每个订单的运输费用与运输时间成线性递减关系, 企业自然会在一个订单完成加工的那一天就把它由 3PL 公司发送出去, 也就是说若订单 O_i 在第 t_i 天完成加工, 则企业给 O_i 选择的运输模式为 $r_i = d_i - t_i$ 天。

在本文中, 我们不允许订单的部分运送, 即订单 O_i 必须等其中所有产品都完成了加工才可以一起运送给顾客。目标是找到一个加工排序使得每个订单能在最迟加工时间之前完成加工并且产生的运输费用尽可能的少。

我们将定义问题时用到的记号总结如下：

$O_i =$ 订单 i , $i = 1, 2, \dots, n$ 。

$N =$ 所有订单的集合; $N = \{O_1, O_2, \dots, O_n\}$ 。

$c =$ 生产线每天能够加工的产品数。

$Q_i =$ 订单 O_i 中的产品数量。

$m =$ 生产线计划加工的天数。

$d_i =$ 订单 O_i 的承诺到货时间, $d_i \leq m + 1$ 。

$t_i =$ 订单 O_i 的发货时间, $t_i \leq d_i - 1$; 这也是订单 O_i 中最后一个产品在生产线上的完工时间。

$r_i =$ 订单 O_i 地运送模式 (运送时间); $r_i = d_i - t_i$ 。

$G(r_i, Q_i) =$ 订单 O_i 的运送费用; $G(r_i, Q_i) = Q_i(\alpha - \beta r_i) = Q_i[\alpha - \beta(d_i - t_i)]$ 。

4.3 近似算法及其最坏情况界分析

我们首先给出一些定义和记号。称一个排序为 EDD 序列, 如果订单按照承诺到货时间的非减序排列。称一个排序为 $EDD-LPT$ 序列, 如果订单按照承诺到货时间的非减序排列, 而且如果不同的订单有相同的承诺到货时间, 则把它们按照订单中包含产品数量的非增序排列。

我们的算法及其最坏情况界分析建立在以下两个排序的基础上。第一个排序记为 σ_0 , σ_0 允许订单的部分运送, 订单的加工和运输安排如下: (1) 将订单按照 $EDD-LPT$ 排序并按照此序加工; (2) 把完成加工的订单 (包括部分完成的订单) 在 3PL 公司来提货时运送出去。第二个排序记作 σ_1 , 除了不允许订单的部分运送, 每个订单中的产品必须一起运送, 其余的与 σ_0 一样。

其他的一些符号定义如下:

$N_1 =$ 最迟加工时间为第一天的订单的集合。

$N_2 =$ 最迟加工时间为第二天的订单的集合。

$C_0 =$ 排序 σ_0 产生的总运输费用。

$C_1 =$ 排序 σ_1 产生的总运输费用。

$C^* =$ 问题最优解中产生的总运输费用。

$k: (k-1)c < \sum_{i=1}^n Q_i \leq kc, 2 \leq k \leq m$ 。

在排序 σ_1 中, 假设对 $j = 1, 2, \dots, k-1$, 订单 O_j^1 被分成两部分 (O_{j1}^1, O_{j2}^1) , 其中 O_{j1}^1 (O_{j2}^1) 在第 j ($(j+1)$) 天完成加工。 (O_{j1}^1, O_{j2}^1) 中包含产品的数量为 (Q_{j1}^1, Q_{j2}^1) , O_j^1 的承诺到货时间为 d_j^1 。注意到 (O_{j1}^1, O_{j2}^1) 可能不存在, 这种情形下, 我们定义 $Q_{j1}^1 = Q_{j2}^1 = 0$ 。

为了分析问题以及我们的算法, 还需要下面这些结论。

引理 4.3.1: [47] 如果所有订单都满足式 (4.2), 则任何 *EDD* 序列 (不要求 *EDD-LPT*) 都是可行解。

引理 4.3.2: [47] 允许订单部分运送的 *EDD* 序列 (不要求 *EDD-LPT*) 的总运输费用是我们研究的问题的最优解值的下界。

引理 4.3.3: 在排序 σ_1 中, 若 $\alpha - \beta d_1^1 < 0$, 则所有排在 O_1^1 之后的订单的承诺到货时间都是 d_1^1 。

证明: 假设不然, 存在一个订单 O_j , 满足 $d_j \geq d_1^1 + 1$, 且在 O_1^1 之后加工。如果把订单 O_j 放在第一天加工, 其单位运输费用为 $\alpha - \beta(d_j - 1) \leq \alpha - \beta d_1^1 < 0$, 这与式 (4.1) 矛盾。注意到 σ_1 是 *EDD* 序列, 因此所有排在 O_1^1 之后的订单的承诺到货时间都是 d_1^1 。 ■

为了方便叙述, 我们记 $N_3 = \{O_i | d_i \leq d_1^1 - 1\}$, 并且如果 $2c < \sum_{i=1}^n Q_i < 3c$, 记 $N_4 = \{O_i | d_i = d_1^1\} \setminus \{O_1^1, O_2^1\}$ 。可以看到 $N_1 \subseteq N_3$, 并且在排序 σ_1 中, N_3 中的订单都在 O_1^1 之前在第一天完成加工。

下面我们给出算法 *MEDD*

算法 *MEDD*

1. 生成排序 σ_1 。若下面的条件之一满足:

(1) $\alpha - \beta d_1^1 \geq 0$,

(2) $Q_{11}^1 \leq \sum_{O_i \in N_3} Q_i$,

(3) $\sum_{i=1}^n Q_i \geq 3c$,

(4) $2c < \sum_{i=1}^n Q_i < 3c$ 且 $Q_{11}^1 \leq (c - Q_{21}^1) + (\sum_{i=1}^n Q_i - 2c)$,

(5) $c < \sum_{i=1}^n Q_i \leq 2c$ 且 $Q_{11}^1 \leq \sum_{i=1}^n Q_i - c + \sum_{O_i \in N_3} Q_i$,

则输出 σ_1 和其总运输费用 C_1 , 停止。

2. 若 $2c < \sum_{i=1}^n Q_i < 3c$ 且 $Q_{11}^1 > (c - Q_{21}^1) + (\sum_{i=1}^n Q_i - 2c)$, 则按如下方式调整 σ_1 : 把安排在 O_1^1 和 O_2^1 之间的订单和安排在 O_2^1 之后的订单放在 O_1^1 之前加工。记此排序为 σ_2 , 其总运输费用为 C_2 。在 σ_2 中, 订单 O_1^1 与 O_2^1 仍被分割, 设 O_1^1 (O_2^1) 被分割成 (O_{11}^2, O_{12}^2) ((O_{21}^2, O_{22}^2)), 其中包含的产品的数量为 (Q_{11}^2, Q_{12}^2) ((Q_{21}^2, Q_{22}^2))。分以下三种情形考虑:

(1) 若 $Q_{11}^2 \leq \max\{\sum_{O_i \in N_3} Q_i, Q_{12}^2 + \sum_{i=1}^n Q_i - 2c\}$, 输出 $\{\sigma_2, C_2\}$, 停止。

(2) 若 $Q_{11}^2 > \max\{\sum_{O_i \in N_3} Q_i, Q_{12}^2 + \sum_{i=1}^n Q_i - 2c\}$ 且 $\sum_{O_i \in N_1} Q_i + Q_1^1 + Q_2^1 + \sum_{O_i \in N_2} Q_i \leq 2c$, 则选以下两个排序中总运输费用较小的那个为输出:

$$\pi_1 = (N_1, O_1^1, O_2^1, N_2, N \setminus \{N_1 \cup N_2 \cup O_1^1 \cup O_2^1\}),$$

与

$$\pi_2 = (N_1, O_2^1, O_1^1, N_2, N \setminus \{N_1 \cup N_2 \cup O_1^1 \cup O_2^1\}).$$

(3) 若 $Q_{11}^2 > \max\{\sum_{O_i \in N_3} Q_i, Q_{12}^2 + \sum_{i=1}^n Q_i - 2c\}$ 且 $\sum_{O_i \in N_1} Q_i + Q_1^1 + Q_2^1 + \sum_{O_i \in N_2} Q_i > 2c$, 则选以下两个排序中总运输费用较小的那个为输出:

$$\pi_3 = (N_1, O_1^1, N_2, N \setminus \{N_1 \cup N_2 \cup O_1^1 \cup O_2^1\}, O_2^1),$$

与

$$\pi_4 = (N_1, O_2^1, N_2, N \setminus \{N_1 \cup N_2 \cup O_1^1 \cup O_2^1\}, O_1^1).$$

对情形 (2) 和 (3), 记得到的排序为 σ_3 , 总运输费用为 C_3 。

3. 若 $c < \sum_{i=1}^n Q_i \leq 2c$ 且 $Q_{11}^1 > \sum_{i=1}^n Q_i - c + \sum_{O_i \in N_3} Q_i$, 记 σ_1 中安排在 O_1^1 之后的订单的集合为 N_5 。按如下方式调整 σ_1 : 把 N_5 中的订单移到 O_1^1 之前。记得到的排序为 σ_4 , 总运输费用为 C_4 。在 σ_4 中, 订单 O_1^1 仍被分割, 假设其被分割成 (O_{11}^4, O_{12}^4) , 其中所包含的产品数量为 (Q_{11}^4, Q_{12}^4) 。考虑如下两种情形:

(1) 若以下条件之一满足 (a) $Q_{11}^4 \leq \sum_{i=1}^n Q_i - c + \sum_{O_i \in N_3} Q_i$, 或者 (b)

$Q_{11}^4 > \sum_{i=1}^n Q_i - c + \sum_{O_i \in N_3} Q_i$ 且 $\sum_{O_i \in N_3 \setminus N_1} Q_i + \sum_{O_i \in N_5} Q_i < Q_{12}^4$, 输出 $\{\sigma_4, C_4\}$, 停止。

(2) 若 $Q_{11}^4 > \sum_{i=1}^n Q_i - c + \sum_{O_i \in N_3} Q_i$ 且 $\sum_{O_i \in N_3 \setminus N_1} Q_i + \sum_{O_i \in N_5} Q_i \geq Q_{12}^4$, 按如下方式调整 σ_4 : 把 $(N_3 \setminus N_1) \cup N_5$ 中的订单都移到最后加工。记得到的排序为 σ_5 , 总运输费用为 C_5 。输出 $\{\sigma_5, C_5\}$, 停止。

我们把从算法 *MEDD* 得到的排序记为 σ_{MEDD} , 其总运输费用为 C_{MEDD} 。

引理 4.3.4: 若算法 *MEDD* 的输出是 $\{\sigma_1, C_1\}$, 则 $\frac{C_{MEDD}}{C^*} \leq 2$ 。

证明: 在排序 σ_1 中, 我们用 $Cost(no-split)$ 表示所有未被分割的订单 (订单中的产品在同一天完成加工) 产生的运输费用。显然可见,

$$C_0 = Cost(no-split) + Q_{11}^1[\alpha - \beta(d_1^1 - 1)] + Q_{12}^1[\alpha - \beta(d_1^1 - 2)] + \cdots \\ + Q_{k-1,1}^1[\alpha - \beta(d_{k-1}^1 - (k-1))] + Q_{k-1,2}^1[\alpha - \beta(d_{k-1}^1 - k)], \quad (4.3)$$

和

$$C_1 = Cost(no-split) + (Q_{11}^1 + Q_{12}^1)[\alpha - \beta(d_1^1 - 2)] + \cdots \\ + (Q_{k-1,1}^1 + Q_{k-1,2}^1)[\alpha - \beta(d_{k-1}^1 - k)]. \quad (4.4)$$

所以, $C_1 - C_0 = \beta(Q_{11}^1 + Q_{21}^1 + \cdots + Q_{k-1,1}^1)$ 。

下面我们根据算法第一步的各种情形讨论:

(1) 若 $\alpha - \beta d_1^1 \geq 0$, 则由式 (4.1) 我们知道 $\alpha - \beta(d_h^1 - h) > \beta$, $h = 1, 2, \dots, k-1$, 再结合式 (4.3), 可以得出

$$C_0 > \beta(Q_{11}^1 + Q_{21}^1 + \cdots + Q_{k-1,1}^1) \\ = C_1 - C_0.$$

因此, $C_1 \leq 2C_0 \leq 2C^*$ 。

(2) 若 $Q_{11}^1 \leq \sum_{O_i \in N_3} Q_i$, 则由 N_3 的定义和式 (4.1), 可知

$$\begin{aligned} Cost(no-split) &\geq \sum_{O_i \in N_3} Q_i [\alpha - \beta(d_i - 1)] \\ &\geq \left(\sum_{O_i \in N_3} Q_i \right) [\alpha - \beta(d_1^1 - 2)] \\ &> \beta \left(\sum_{O_i \in N_3} Q_i \right) \\ &\geq \beta Q_{11}^1. \end{aligned}$$

由式 (4.1), 我们知道 $\alpha - \beta(d_h^1 - h) > \beta$, $h = 2, \dots, k-1$. 这表明

$$\begin{aligned} C_0 &> Cost(no-split) + \beta(Q_{21}^1 + \dots + Q_{k-1,1}^1) \\ &> \beta(Q_{11}^1 + Q_{21}^1 + \dots + Q_{k-1,1}^1) \\ &= C_1 - C_0. \end{aligned}$$

因此, $C_1 \leq 2C_0 \leq 2C^*$.

(3) 若 $\sum_{i=1}^n Q_i \geq 3c$, 则在排序 σ_0 中, 第三天完成加工的产品总量为 c . 由式 (4.1) 我们知道第三天运送的单位产品的费用至少为 2β . 因此在 σ_0 中, 第三天完成加工的产品总运送费用至少为 $2c\beta$. 由式 (4.3), 我们可以推出

$$\begin{aligned} C_0 &> Q_{21}^1 \beta + 2c\beta + \beta(Q_{41}^1 + \dots + Q_{k-1,1}^1) \\ &> \beta(Q_{11}^1 + Q_{21}^1 + \dots + Q_{k-1,1}^1) \\ &= C_1 - C_0. \end{aligned}$$

因此, $C_1 \leq 2C_0 \leq 2C^*$.

(4) 若 $2c < \sum_{i=1}^n Q_i < 3c$ 且 $Q_{11}^1 \leq (c - Q_{21}^1) + (\sum_{i=1}^n Q_i - 2c)$. 由式 (4.1) 我们知道, 在第二天之后运送的单位产品的费用至少为 β , 再根据式 (4.3) 可以得到

$$\begin{aligned} C_0 &> \beta \left(\sum_{i=1}^n Q_i - c \right) \\ &= \beta(c - Q_{21}^1 + \sum_{i=1}^n Q_i - 2c) + \beta Q_{21}^1 \end{aligned}$$

$$\begin{aligned} &\geq \beta(Q_{11}^1 + Q_{21}^1) \\ &= C_1 - C_0. \end{aligned}$$

因此, $C_1 \leq 2C_0 \leq 2C^*$.

(5) 若 $c < \sum_{i=1}^n Q_i \leq 2c$ 且 $Q_{11}^1 \leq \sum_{i=1}^n Q_i - c + \sum_{O_i \in N_3} Q_i$. 由 N_3 的定义我们知道, 在 σ_0 中, N_3 中单位产品的运输费用至少为 β . 同样, 在 σ_0 中, 第二天运送的单位产品的费用至少为 β . 因此,

$$\begin{aligned} C_0 &\geq \beta \left(\sum_{O_i \in N_3} Q_i \right) + \beta \left(\sum_{i=1}^n Q_i - c \right) \\ &= \beta \left(\sum_{i=1}^n Q_i - c + \sum_{O_i \in N_3} Q_i \right) \\ &\geq \beta Q_{11}^1 = C_1 - C_0 \end{aligned}$$

因此, $C_1 \leq 2C_0 \leq 2C^*$.

综合以上的讨论, 我们知道若算法 *MEDD* 的输出是 $\{\sigma_1, C_1\}$, 则 $\frac{C_{MEDD}}{C^*} \leq 2$. ■

引理 4.3.5: 若输出是 $\{\sigma_2, C_2\}$, 则 $\frac{C_{MEDD}}{C^*} \leq 2$.

证明: 当输出是 $\{\sigma_2, C_2\}$ 时, 以下四个条件成立: (C1) $\alpha - \beta d_1^1 < 0$, (C2) $2c < \sum_{i=1}^n Q_i < 3c$, (C3) $Q_{11}^1 > (c - Q_{21}^1) + (\sum_{i=1}^n Q_i - 2c)$, (C4) $Q_{11}^2 \leq \max\{\sum_{O_i \in N_3} Q_i, Q_{12}^2 + \sum_{i=1}^n Q_i - 2c\}$.

在排序 σ_1 中, 我们用 Δ_1 , Δ_2 和 Δ_3 分别表示排在 O_1^1 之前的, 在 O_1^1 和 O_2^1 之间的, 在 O_2^1 后面的订单的集合. 排序 σ_2 可以表示为 $(\Delta_1, \Delta_2, \Delta_3, O_1^1, O_2^1)$. 由条件 (C3) 可知, 在排序 σ_2 中, O_1^1 和 O_2^1 仍被分割. 由条件 (C1) 和引理 4.3.3 我们知道, σ_2 是可行的且是一个 EDD 序列. 记 $\bar{\sigma}_0$ 为与 σ_2 一样的加工序列, 但是允许订单的部分运送, \bar{C}_0 为 $\bar{\sigma}_0$ 的总运输费用. 由引理 4.3.2 我们得到 $\bar{C}_0 \leq C^*$.

根据条件 (C4), 我们分下面两种情形考虑.

情形 (i) 若 $Q_{11}^2 \leq \sum_{O_i \in N_3} Q_i$, 由与引理 4.3.4 的第二种情形类似的证明 (*LPT* 在证明中没有用到), 我们可以推出

$$C_2 - \bar{C}_0 \leq \bar{C}_0,$$

所以 $C_2 \leq 2\overline{C_0} \leq 2C^*$ 。

情形 (ii) 若 $Q_{11}^2 \leq Q_{12}^2 + \sum_{i=1}^n Q_i - 2c$ ，由与引理 4.3.4 的第四种情形类似的证明 (*LPT* 在证明中没有用到)，我们可以推出

$$C_2 - \overline{C_0} \leq \overline{C_0},$$

所以 $C_2 \leq 2\overline{C_0} \leq 2C^*$ 。

综合以上讨论，我们知道若算法的输出是 $\{\sigma_2, C_2\}$ ，则 $\frac{C_{MEDD}}{C^*} \leq 2$ 。 ■

引理 4.3.6: 若算法的输出是 $\{\sigma_3, C_3\}$ ，则 $\frac{C_{MEDD}}{C^*} = \frac{C_3}{C^*} \leq 2$ 。

证明: 当算法的输出是 $\{\sigma_3, C_3\}$ 时，下面四个条件成立: (C1) $\alpha - \beta d_1^1 < 0$, (C2) $2c < \sum_{i=1}^n Q_i < 3c$, (C3) $Q_{11}^1 > (c - Q_{21}^1) + (\sum_{i=1}^n Q_i - 2c)$, (C4) $Q_{11}^2 > \max\{\sum_{O_i \in N_3} Q_i, Q_{12}^2 + \sum_{i=1}^n Q_i - 2c\}$ 。

由这四个条件，我们可以得到以下结论:

1. 在排序 σ_1 中，排在 O_1^1 之后的订单的承诺到货时间为 d_1^1 。这由条件 (C1) 与引理 4.3.3 可得。
2. $Q_1^1 > \frac{2}{3}c$ 且 O_1^1 是承诺到货时间为 d_1^1 的订单中包含产品数量最多的。我们用反证法来证明。由条件 (C2) 和 (C4)，我们知道

$$Q_{12}^2 < Q_{11}^2, \tag{4.5}$$

因此，

$$Q_{12}^2 < \frac{1}{2}Q_1^1. \tag{4.6}$$

现在假设 $Q_1^1 \leq \frac{2}{3}c$ ，则 $Q_{12}^2 < \frac{1}{3}c$ 且 $Q_{21}^2 = c - Q_{12}^2 > \frac{2}{3}c > Q_1^1$ ，这与 $Q_1^1 \geq Q_2^1$ 矛盾 (因为 σ_1 是 EDD-LPT 序列)。因此 $Q_1^1 > \frac{2}{3}c$ 。由式 (4.6) 以及 $Q_{12}^1 < Q_{12}^2$ 可知， $Q_{12}^1 < \frac{1}{2}Q_1^1$ ，所以 $Q_{11}^1 > \frac{1}{2}Q_1^1 > \frac{1}{3}c$ 。这表明在第一天不可能有产品数量大于 $\frac{2}{3}c$ 的订单完成加工。因此， O_1^1 是承诺到货时间为 d_1^1 的订单中包含产品数量最多的。

3. $Q_{21}^2 > \frac{1}{2}c$ 。否则， $Q_{21}^2 \leq \frac{1}{2}c$ ，由式 (4.5) 可知， $Q_{11}^2 > Q_{12}^2 = c - Q_{21}^2 \geq \frac{1}{2}c$ 且 $Q_1^1 = Q_{11}^2 + Q_{12}^2 > c$ ，这与 $Q_1^1 \leq c$ 矛盾。

4. 在 σ_1 中, N_4 中所有的订单都在第三天完成加工。否则, 若存在 $O_j \in N_4$ 在第二天完成加工, 则 O_j 被排在 O_1^1 与 O_2^1 之间, 因此 $d_j = d_1^1$ 且由于 σ_1 是 *EDD-LPT* 序列和上面已经得到的第三个结论, 我们知道 $Q_j \geq Q_2^1 > Q_{21}^2 > \frac{1}{2}c$ 。因为 $Q_{21}^1 \geq Q_{21}^2$, 由上面得到的第三个结论可知 $Q_{21}^1 > \frac{1}{2}c$ 。因此 $Q_j + Q_{21}^1 > c$, 这与第二天完成加工的产品数为 c 矛盾。若存在 $O_j \in N_4$ 在第一天完成加工, 由于 σ_1 是 *EDD-LPT* 序列, 则 $Q_j \geq Q_1^1$ 。根据已经得到的第二个结论, 可知 $Q_1^1 > \frac{2}{3}c$ 且 $Q_{11}^1 > \frac{1}{3}c$ 。因此 $Q_j + Q_{11}^1 > c$, 这与第一天完成加工的产品数为 c 矛盾。
5. $\sum_{O_i \in N_3} Q_i < \frac{c}{2}$ 。否则, $\sum_{O_i \in N_3} Q_i \geq \frac{c}{2}$, 则 $Q_{11}^1 = c - \sum_{O_i \in N_3} Q_i \leq \frac{c}{2} \leq \sum_{O_i \in N_3} Q_i$, 则输出是 $\{\sigma_1, C_1\}$ 而不是 $\{\sigma_3, C_3\}$ 。

从以上分析我们知道 σ_1 (σ_0) 的顺序为 (N_3, O_1^1, O_2^1, N_4) , 并且 N_3 包含了 N_1 与 N_2 中的所有订单。因为第二(三)天运送的单位产品的费用至少为 β (2β), 我们可以得到

$$C_0 > \beta c + 2\beta \sum_{O_i \in N_4} Q_i.$$

不难发现算法第二步中, $\pi_1, \pi_2, \pi_3, \pi_4$ 都是可行的, 由 N_1 与 N_2 的定义可知, N_1 中的订单必须在第一天完成加工, 而 N_2 中的订单必须在第二天完成加工。我们可以发现在排序 σ_3 中由订单 O_1^1 与 O_2^1 产生的费用是在满足可行性的基础上产生的最小费用。另外, 若我们把 π_3 (π_4) 中的 O_2^1 (O_1^1) 移到 N_2 之前, O_2^1 (O_1^1) 仍然在第三天才能完成加工。因此,

$$\begin{aligned} C_3 - C^* &\leq 2\beta \left(\sum_{O_i \in N_3 \setminus N_1} Q_i + \sum_{O_i \in N_4} Q_i \right) \\ &< 2\beta \cdot \frac{c}{2} + 2\beta \cdot \sum_{O_i \in N_4} Q_i \\ &< C_0 \leq C^*. \end{aligned}$$

所以 $C_3 < 2C^*$. ■

引理 4.3.7: 若输出是 $\{\sigma_4, C_4\}$, 则 $\frac{C_{MEDD}}{C^*} \leq 2$ 。

证明: 当输出是 $\{\sigma_4, C_4\}$ 时, 下面两个条件成立: (C1) $c < \sum_{i=1}^n Q_i \leq 2c$, (C2) $Q_{11}^1 > \sum_{i=1}^n Q_i - c + \sum_{O_i \in N_3} Q_i$. 我们用 Δ 表示在 σ_1 中安排在 N_3 和 O_1^1 之间的订单的集合, 那么 $\sigma_1 = (N_3, \Delta, O_1^1, N_5)$. 若 $\Delta \neq \emptyset$, 由于 σ_1 是 EDD-LPT 序列, 我们知道 $d_j = d_1^1$ 且对 $O_j \in \Delta$, 有 $Q_j \geq Q_1^1$. σ_4 可以表示为 $(N_3, \Delta, N_5, O_1^1)$. 显然可见, σ_4 是可行的.

当 $Q_{11}^4 \leq \sum_{i=1}^n Q_i - c + \sum_{O_i \in N_3} Q_i$ 时, 我们知道

$$C_4 - C_0 = \beta [Q_{11}^4 - \sum_{O_i \in N_5} Q_i] \quad (4.7)$$

由式 (4.7), $C_0 > \beta [\sum_{O_i \in N_3} Q_i + (\sum_{i=1}^n Q_i - c)]$ 以及 $Q_{11}^4 = Q_{11}^1 - \sum_{O_i \in N_5} Q_i$, 我们可以得到 $C_4 - C_0 \leq C_0$, 因此 $C_4 \leq 2C_0 \leq 2C^*$.

当 $Q_{11}^4 > \sum_{i=1}^n Q_i - c + \sum_{O_i \in N_3} Q_i$ 且 $\sum_{O_i \in N_3 \setminus N_1} Q_i + \sum_{O_i \in N_5} Q_i < Q_{12}^4$ 时, 我们可以证明 σ_4 是最优解. 由条件 (C2) 可知, 在 σ_4 中, 订单 O_1^1 是唯一被分割的订单, 且其在第二天完成加工. 由于对 $O_j \in \Delta$ 有 $Q_j \geq Q_1^1$, 因此任何有 Δ 中的订单在第二天完成加工的可行解的总运输费用都不会少于 C_4 . 下面我们考虑 Δ 中订单都在第一天完成加工的任意排序 π . N_1 中的订单必须在第一天完成加工. 因为 $\sum_{O_i \in N_3 \setminus N_1} Q_i + \sum_{O_i \in N_5} Q_i < Q_{12}^4$, 即使 $(N_3 \setminus N_1) \cup N_5$ 中所有订单都在第二天完成加工, O_1^1 仍然在第二天完成加工, 因此 π 的总运输费用至少为 C_4 , 所以 σ_4 是最优解. ■

引理 4.3.8: 若输出是 $\{\sigma_5, C_5\}$, 则 $\frac{C_{MEDD}}{C_0} \leq 2$.

证明: 当输出是 $\{\sigma_5, C_5\}$ 时, 以下两个条件成立: (C1) $Q_{11}^4 > \sum_{i=1}^n Q_i - c + \sum_{O_i \in N_3} Q_i$, (C2) $\sum_{O_i \in N_3 \setminus N_1} Q_i + \sum_{O_i \in N_5} Q_i \geq Q_{12}^4$. 沿用引理 4.3.7 中 Δ 的定义, 我们知道 $\sigma_5 = (N_1, \Delta, O_1^1, (N_3 \setminus N_1) \cup N_5)$. 不难发现, σ_5 是可行的. 由条件 (C2) 我们知道在排序 σ_5 中, 订单 O_1^1 未被分割, 在第一天就完成了加工. 因此,

$$C_0 > \beta \cdot \sum_{O_i \in N_3} Q_i + \beta \cdot \sum_{O_i \in N_5} Q_i$$

且

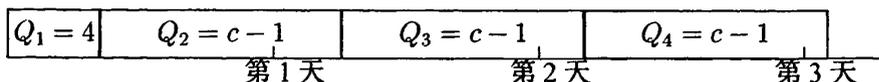
$$C_5 - C_0 < \beta \cdot \sum_{O_i \in N_3 \setminus N_1} Q_i.$$

所以 $C_5 < 2C_0 \leq 2C^*$. ■

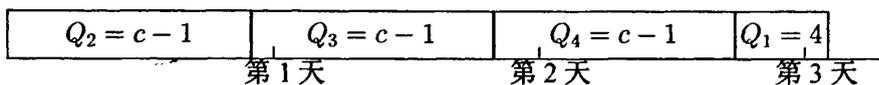
定理 4.3.9: $\frac{C_{MEDD}}{C^*} \leq 2$, 且这个界是紧的。

证明: $\frac{C_{MEDD}}{C^*} \leq 2$ 由引理 4.3.4 至 4.3.8 可以很容易推出。

为了证明这个界是紧的, 考虑如下实例: 企业每天可以生产 c 个产品, $n = 4$, $Q_1 = 4$, $Q_2 = Q_3 = Q_4 = c - 1$, $d_1 = 5$, $d_2 = d_3 = d_4 = 6$, 单位产品运送时间为 x 天的运输费用为 $(5\beta + \eta) - \beta x$, 其中 $\eta \rightarrow 0+$ 。运行算法 $MEDD$, 订单被安排如下:



可以看到, $C_{MEDD} = 4[(5\beta + \eta) - 4\beta] + (c - 1)[(5\beta + \eta) - 4\beta + (5\beta + \eta) - 3\beta + (5\beta + \eta) - 2\beta] = (6c - 2)\beta + (3c + 1)\eta$ 。不难发现, 在最优解中, 订单被安排如下:



因此, $C^* = (c - 1)[(5\beta + \eta) - 5\beta + (5\beta + \eta) - 4\beta + (5\beta + \eta) - 3\beta] + 4[(5\beta + \eta) - \beta] = (3c + 13)\beta + (3c + 1)\eta$ 。所以, $\frac{C_{MEDD}}{C^*} = \frac{(6c - 2)\beta + (3c + 1)\eta}{(3c + 13)\beta + (3c + 1)\eta} \rightarrow 2, c \rightarrow +\infty$ 且 $\eta \rightarrow 0+$ 。 ■

参考文献

- [1] B. Bilgen, I. Ozkarahan, Strategic tactical and operational production-distribution models: A review, *International Journal of Technology Management*, (2004), 28, 151–171.
- [2] Y. C. Chang, C. Y. Lee, Machine scheduling with job delivery coordination, *European Journal of Operational Research*, 158, (2004), 470–487.
- [3] B. Chen, C.N. Potts, G.J. Woeginger, A review of machine scheduling: Complexity, algorithms and approximability. In *Handbook of combinatorial optimization*. eds. D.Z. Du, P.M. Pardahs, (1998), *Kluwer Academic Publishers*.
- [4] Z.-L. Chen, Integrated production and outbound distribution scheduling: Review and extensions, (2007), Revised for *Operations Research*.
- [5] Z.-L. Chen, Integrated production and distribution operations: Taxonomy, models, and review. In: *Handbook of Quantitative Supply Chain Analysis: Modeling in the E-Business Era*, Edited by D.Simchi-Levi, S.D. Wu, and Z.-J. Shen, *Kluwer Academic Publishers*.
- [6] Z.-L. Chen, G.Pundoor, Integrating ourder scheduling with packing and delivery. Working paper, Robert H.Smith School of Business, University of Maryland, College Park, MD 20742.
- [7] Z.-L. Chen, G.Pundoor, Order assignment and scheduling in a supply chain, *Operations Research*, 54, (2006), 555–572.
- [8] Z.-L. Chen, G. L. Vairaktarakis, Integrated scheduling of production and distribution operations, *Management Science*, 51, (2005), 614–628.
- [9] T.C.E. Cheng, V.S. Gordon, M.Y. Kovalyov, Single machine scheduling with batch deliveries, *European Journal of Operational Research*, 94, (1996), 277–283.
- [10] S.S. Erenguc, N.C. Simpson, A.J. Vakharia, Integrated production/distribution planning in supply chains: An invited review, *European Journal of Operational Research*, 115, (1999), 219–236.
- [11] J.M. Garcia, S. Lozano, Production and vehicle scheduling for ready-mixed operations, *Computers & Industrial Engineering*, 46, (2005), 803–816.
- [12] J.M. Garcia, K. Smith, S. Lozano, F. Guerrero, A comparison of GRASP and exact method for solving a production and delivery scheduling problem, In: *Proceedings of hybrid information systems: Advances in soft computing*, Heidelberg, Physica-Verlag, 431–447.
- [13] M.R. Garey, D.S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*, (1979), Freeman, San Francisco.

- [14] H.N. Geismar, G. Laporte, L. Lei, C. Sriskandarajah, The integrated production and transportation scheduling problem for a product with a short life span and non-instantaneous transportation time, To appear in *INFORMS Journal on computing*.
- [15] M. Goetschalckx, C.J. Vidal, K. Dogan, Modelong and design of global logistics systems: A review of integrated strategic and tactical models and design algorithms, *European Journal of Operational Research*, **143**, (2002), 1–18.
- [16] R.L. Graham, Bounds for certain multiprocessing anomalies, *The Bell System Technical Journal*, **45**, (1966), 1563–1581.
- [17] R.L. Graham, Bounds on multiprocessing finishing anomalies, *SIAM Journal on Applied Mathematics*, **17**, (1969), 416–429.
- [18] R.L. Graham, E.L. Lawler, J.K. Lenstra, A.H.G. Rinnooy Kan, Optimization and approximation in deterministic suquencing and scheduling: A survey, *Annals of Operations Research*, **5**, (1979), 287–326.
- [19] N.G. Hall, C.N. Potts, Supply chain chain scheduling: Batching and delivery, *Operations Research*, **51**, (2003), 566–584.
- [20] N.G. Hall, C.N. Potts, The coordination of scheduling and batch deliveries, *Annals of Operations Research*, **135**, (2005), 41–64.
- [21] Y. He, W. Zhong, H. Gu, Improved algorithms for two single machine scheduling problems, *Theoretical Computer Science*, **363**, (2006), 257–265.
- [22] J.W. Herrmann, C. Y. Lee, On scheduling to minimize earliness-tardiness and batch delivery costs with common due date, *European Journal of Operational Research*, **70**, (1993), 272–288.
- [23] J.S. Hurink, S.K. Knust, Makespan minimization for flow-shop problems with transportation times and a single robot, *Discrete Applied Mathematics*, **112**, (2001), 199–216.
- [24] M. Ji, Y. He, T. C. E. Cheng, Batch delivery scheduling with batch delivery cost on a single machine, *European Journal of Operational Research*, **176**, (2007), 745–755.
- [25] H. Kellerer, U. Pferschy, A new fully polynomial approximation scheme for the knapsack problem, *Lecture Notes in Computer Science*, **1444**, (1998), 123–134.
- [26] H. Kellerer, U. Pferschy, David Pisinger, Knapsack problems, (2004), *Springer*.
- [27] M.A. Langston, Interstage transportation planning in deterministic flow-shop environment, *Operations Research*, **35**, (1987), 556–564.

- [28] E. Lawler, Fast approximation algorithms for knapsack problems, *Mathematics of Operations Research*, **4**, (1979), 339–356.
- [29] C.Y. Lee, Parallel machine scheduling with non-simultaneous machine available time, *Discrete Applied Mathematics*, **30**, (1991), 53–61.
- [30] C. Y. Lee, Z.-L. Chen, Machine scheduling with transportation considerations, *Journal of Scheduling*, **4**, (2001), 3–24.
- [31] C. Y. Lee, V.A. Strusevich, Two-machine shop scheduling with an uncapacitated interstage transporter, *IIE Transactions*, **37**, (2005), 725–736.
- [32] J.K. Lenstra, A.H.G. Rinnooy Kan, P. Brucker, Complexity of machine scheduling problems, *Annals of Discrete Mathematics*, **1**, (1977), 343–362.
- [33] D. Simchi-Levi, New worst-case results for the bin packing problem, *Naval Research Logistics*, **41**, (1994), 579–585.
- [34] C.-L. Li, J. Ou, Machine scheduling with pickup and delivery, *Naval Research Logistics*, **52**, (2005), 617–630.
- [35] C.-L. Li, J. Ou, Coordinated scheduling of customer orders with decentralized machine locations, *IIE Transactions*, **39**, (2007), 899–909.
- [36] C.-L. Li, G. L. Vairaktarakis, Coordinating production and distribution of jobs with bundling operations, *IIE Transactions*, **39**, (2007), 203–215.
- [37] C.-L. Li, G. L. Vairaktarakis, C.Y. Lee, Machine scheduling with deliveries to multiple customer locations, *European Journal of Operational Research*, **164**, (2005), 39–51.
- [38] 林诒勋, 动态规划与序贯最优化, (1997), 河南大学出版社.
- [39] L. Lu, J. Yuan, L. Zhang, Single machine scheduling with release dates and job delivery to minimize the makespan, *Theoretical Computer Science*, **393**, (2008), 102–108.
- [40] J. C.-H. Pan, C.-L. Wu, H.-C. Huang, C.-S. Su, Coordinating scheduling with batch deliveries in a two-machine flow shop, To appear in *International Journal of Advanced Manufacturing Technology*.
- [41] M. Pinedo, *Scheduling: Theory, algorithms and systems*, (2002), 2nd ed. Prentice Hall.
- [42] G.Pundoor, Z.-L. Chen, Scheduling a production-distribution system to optimize the tradeoff between delivery tardiness and distribution cost, *Naval Research Logistics*, **52**, (2005), 571–589.

-
- [43] X. Qi, A logistics scheduling model: scheduling and transshipment for two processing centers, *IIE Transactions*, **38**, (2006), 609–618.
- [44] X. Qi, Coordinated logistics scheduling for in-house production and outsourcing, *IIE Transactions on automatic science and engineering*, **5**, (2008), 188–192.
- [45] A.M. Sarmiento, R. Nagi, A review of integrated analysis of production-distribution systems, *IIE Transactions*, **31**, (1999), 1061–1074.
- [46] P. Schuurman, G. J. Woeginger, Approximation Schemes- A tutorial, In *Lectures on Scheduling*. eds. R. H. Moehring, C. N. Potts, A. S. Schulz, G. J. Woeginger, L. A. Wolsey, (2007).
- [47] K.E. Stecke, X. Zhao, Production and transportation integration for a make-to-order manufacturing company with a commit-to-delivery business mode, *Manufacturing & Service Operations Management*, **9**, (2007), 206–224.
- [48] D.J. Thomas, P.M. Griffin, Coordinated supply chain management, *European Journal of Operational Research*, **94**, (1996), 1–15.
- [49] H. Wang, C.-Y. Lee, Production and transport logistics scheduling with two transport mode choices, *Naval Research Logistics*, **52**, (2005), 796–809.
- [50] X. Wang, T. C. E. Cheng, Machine scheduling with availability constraint and job delivery coordination, *Naval Research Logistics*, **54**, (2007), 11–20.
- [51] M. Yue, A simple proof of the inequality $FFD(L) \leq \frac{11}{9}OPT(L) + 1 \forall L$, for the *FFD* bin-packing algorithm, *Acta Mathematica Applicata Sinica*, **7**, (1991), 321–331.
- [52] W. Zhong, G. Dósa, Z. Tan, On the machine scheduling problem with job delivery coordination, *European Journal of Operational Research*, **182**, (2007), 1057–1072.

致 谢

即将完成论文的时候，心中百感交集，还记得五年前刚进入浙大时的那份懵懂，而今我将要完成我的博士阶段的学习。首先要感谢我的导师：何勇教授。尽管何老师指导我的时间不长，但是何老师是我进入组合优化领域的启蒙老师，他治学严谨，学识渊博，平易近人，对学生极其负责和关爱。虽然恩师已经离开了我们，但是我将永远铭记这位好老师！

感谢谈之奕教授对我的学习和生活上的指导和关心，他学术积累深厚，治学严谨踏实。我的论文中渗透着他辛勤的汗水。

感谢课题组的姚恩瑜教授、杨启帆教授、黄庆学教授、张国川教授，他们在学业上的授业解惑，让我具备更加扎实的知识基础，他们的敬业精神和关爱学子的高尚情操，让我终生收益。感谢课题组的季敏、陈汀、鲁海燕、周昊、蒋义伟、韩曙光、盖玲、窦文卿、林凌、魏麒、唐春梅、周萍、吴用、曹顺娟、刘龙城、张安、鄢楚楠、宣竞、俞少华、王钟、陈永、陈思霞等，从欧阳纯美楼到第十一教学楼，都留下了我们讨论与欢笑的身影。

感谢美国马里兰大学的 Zhi-Long Chen（陈志龙）副教授。由于他的帮助，我在美国的一切都很顺利。也由于他为我提供了良好的科研环境，我能够充分利用学习机会，学习到很多新知识，开拓了学术视野。

感谢美国马里兰大学的 Shijun Chen 副教授、Min Chen、Jing Yang、Weiqiang Wu、Qinnan Liu、Qiang Liu 等等，你们给我在美国的生活提供了那么多的帮助！感谢从浙大一起到美国学习的曹娟、龙珑、何洁，你们让我在异国的生活充满了温情！

感谢徐惠霞、赵宏艳、韦学辉、陈孝春、王晓娟、柳玉田、阎丹丹、王静等诸多室友，你们让我的浙大生活充满了温暖和欢乐。希望我们的友谊永存！

我要感谢我的父母这么多年来对我学业的支持和生活的照顾，没有你们就没有我的今天！也要感谢我的先生和他的父母对我的支持和关爱！

攻博期间完成的论文

- [1] Y. He, W. Zhong, H. Gu, Improved algorithms for two single machine scheduling problems, *Theoretical Computer Science*, **363**, 2006, 257–265.
- [2] W. Zhong, G. Dósa, Z. Tan, On the machine scheduling problem with job delivery coordination, *European Journal of Operational Research*, **182**, 2007, 1057–1072.
- [3] W. Zhong, Z. Tan, Coordinated scheduling with transportation for a manufacturing system with multiple processing centers, *submitted*.
- [4] W. Zhong, Z. Chen, Integrated production and distribution scheduling with committed delivery dates, *submitted*.