文章编号: 1001-0920(2009)01-0055-06

两台流水机器协调分解调度问题

关 静[a,b], 唐立新[a], 宋国骄[a]
(东北大学 a. 信息科学与工程学院, b. 教育部流程工业综合自动化重点实验室, 沈阳 110004)

摘 要: 研究钢管加工流程中一类新型两台机器流水车间调度问题,工件在第一台机器上加工后被分解成多个子工件. 对于最小化最大完成时间的情况,给出一个多项式时间的最优算法;对于最小化最大完成时间与惩罚费用之和的情况,给出一个拟多项式时间的动态规划算法;对于考虑生产前运输的最小化最大完成时间的情况,分析了问题的复杂性. 证明了第一种情况的最优算法可作为后两种情况的 2-近似算法. 数值实验表明了算法的有效性.

关键词: 流水机; 调度; 分解; 算法复杂性; 动态规划; 界分析
中图分类号: TP29 **文献标识码**: A

Two machines flowshop scheduling problem with disintegration consideration

GUAN Jing[a,b], *TANG Li-xin*[a], *SONG Guo-jiao*[a]
(a. College of Information Science and Engineering, b. Key Laboratory of Integrated for Automation for the Process Industry of Ministry of Education, Northeastern University, Shenyang 110004, China. Correspondent: TANG Li-xin, E-mail: qhjytlx@mail.neu.edu.cn)

Abstract: A new type of two-machine flowshop scheduling problem is studied. Jobs are disintegrated into subjobs on the first machine. For the scenario when minimizing the maximum completion time as the objective, we provide a polynomial time optimal algorithm. For the scenario when minimizing the sum of maximum completion time and penality cost as the objective, we propose a pseudo-polynomial dynamic algorithm. For the scenario with inbound transportation consideration when minimizing the maximum completion time as the objective, we analysis the complexity of the problem. The optimal algorithm of the first scenario can provide hueristic algorithms with worst case ratio no more than 2 for other secnarios. Computational experiments show the effectiveness of the proposed algorithm.

Key words: Flowshop; Scheduling; Disintegration; Computational complexity; Dynamic algorithm; Worst case analysis

1 引 言

本文以某大型钢管加工企业为背景,研究套管加工流程中切管机和下游机器的协调调度问题. 较长的毛管从中间库出来,经过切管机被切割成多个适当长度的子钢管,切好的子管在下游机器上进行车丝、通径等子管加工,最后到达成品库. 整个机器环境为多台机器流水车间. 当流水车间的机器数超过 3 时,为强 NP 难问题,不存在多项式时间的最优算法. 因此,这里进行一个合理的简化,将下游子管加工工序视为一台机器,同时选取整个工艺流程中具有代表性的切管机进行研究. 这样,机器环境变成一个新型的两台流水机器,进而可以对问题的可解性进行研究.

具有分解特点的两台流水机器调度问题类似于传统的两台流水机器调度问题,但又与传统问题有着明显的不同. 在传统的两台流水机器调度问题中,每个工件只有在第 1 台机器上加工完成后才能开始其在第 2 台机器上加工. 而本文研究的问题中,工件在第 1 台机器上每分解出一个子工件,都可以开始在第 2 台机器上加工,不用等待工件的所有子工件

收稿日期: 2007-09-25; **修回日期**: 2007-12-21.
基金项目: 国家杰出青年科学基金项目(70425003); 国家 863 计划项目(2006AA04Z174); 国家自然科学基金项目(60674084); 高等学校学科创新引智计划项目(B08015).
作者简介: 关静(1977 —),女,辽宁锦州人,博士生,从事生产调度、物流管理的研究; 唐立新(1966 —),男,黑龙江绥化人,教授,博士生导师,从事生产调度、智能优化和物流管理等研究.

全部在第 1 台机器上分解完成. 换言之,传统的两台机器流水调度问题中,一个工件在加工时对两台机器的占用在时间上不存在交集;而具有分解特点的调度问题中,对两台机器的占用在时间上存在交集,并且随着工件在调度中的位置而变化. 因此,传统的求解两台机器流水调度最大完成时间问题的 Johnson 规则不适用于该问题.

在研究传统流水车间问题的文献中,有很多是针对不同目标函数和约束条件进行研究的. Allahverdi[1]考虑的目标函数为最大完成时间与平均流水时间加权求和;Allahverdi 和 Aldowaisan[2]研究两台机器流水环境下总完成时间问题;Gupta 等[3]研究满足最大完成时间最小条件下总流水时间最小问题;Cepek 等[4]研究机器没有闲置的两机流水问题;Allahverdi 和 Aldowaisan[5]研究的问题与文献[4]类似,工件间需考虑调整时间;Sung 和 Kim[6]研究的问题允许工件有动态的到达时间;Lin 和 Cheng[7]研究流水环境下成批调度;Vickson 和 Alfredsson[8]研究两台机器流水族调度问题,证明了族加工可中断问题比不可中断问题具有更优的解;Cheng 等[9]给出了流水车间调度综述.

传统调度问题中,工件数目在调度前后不发生变化. 到目前为止,研究调度前后工件数目发生变化的只有装配型和分裂型两种调度.

装配类型调度中,每个成品工件都由若干个部件组成. 部件首先在不同类型的机器上进行加工,然后进行部件组装. Potts 等[10]研究的部件加工在第 1 阶段的多台机器上进行,装配则在第 2 阶段的机器上进行;Lin 和 Cheng[11]以及 Cheng 和 Wang[12]都研究了两台机器流水环境下组件装配问题;Yang[13]研究部件加工在并行机上完成;Koulamas 和 Kyparisis[14]考虑部件加工在流水机上完成,之后他们又进一步研究了部件加工和组装之间存在的运输问题[15].

至于分裂类型调度,据掌握资料所知,只有 Namman 和 Rom[16]进行了研究. 他们研究了数据传输过程中信息包的分裂问题. 在第 1 阶段,多组大小不等的数据被压缩在容量固定的信息包内进行传输;在第 2 阶段,信息包进行释放,把初始的多组数据释放出来,并各自添加一些新的信息. 这个问题在第 1 阶段类似背包问题,需要把不同长度的数据压缩到多个容量固定的信息包中. 他们将此问题归结到传统的背包问题,提出近似算法,并分析了算法的最坏情况. 本文研究问题与此分裂类型调度有明显不同.

2 两台流水机器分解调度问题

2.1 两台机器无限缓冲问题

这里考虑有无限缓冲的两台流水机器分解调度问题. 问题描述如下:一组 n 个工件 $N = \{J_1, J_2, \cdots, J_n\}$,首先在一个具有切割功能的机器上进行加工,每个工件 J_i 连续分解成多个子工件 $J_{i1}, J_{i2}, \cdots, J_{in_i}$;然后所有子工件在第 2 台机器上一个接一个地加工. 子工件是项目可利用的,即工件在第 1 台机器上加工时,每分解出一个子工件,该子工件便可在第 2 台机器上加工. 如果工件在第 1 台机器上加工完成时,第 2 台机器不可利用,则工件进入缓冲等待. 其中,工件在第 1 台机器上的分解过程不能中断,即一个工件的所有子工件必须全部分解完成才能开始分解下一个工件. 目标函数为所有工件的最大完成时间最小.

根据三域表示法[17],把这种工件在第 1 台机器上具有分解特征的两台流水机器求最大完成时间最小问题表示成 $F_2 \mid m_1, \mathrm{disi} \mid C_{\max}$. 在研究该问题的算法之前,首先给出两个基本性质.

性质 1 在一个 $F_2 \mid m_1, \mathrm{disi} \mid C_{\max}$ 问题的最优调度中,工件在两台机器上的加工顺序必然相同.

证明 如果顺序不同,则总可以通过工件之间的交换,使之连续加工且最大完成时间不会增大. 这是由标准的工件互换得到的. □

性质 2 在一个 $F_2 \mid m_1, \mathrm{disi} \mid C_{\max}$ 问题的最优调度中,第 1 台机器上必然没有闲置时间.

证明 如果在一个最优调度中第 1 台机器上有闲置时间,则总可以向前移动工件以消除闲置时间,同时不增大问题的目标函数值. □

在给出问题最优算法之前,首先假设每个钢管分解出子工件的个数相同(每个工件分解出子工件个数任意的情况同理可得),并给出算法中要用到的变量:定义 n 为初始阶段被调度工件的个数(钢管切割前的个数);l 为在第 1 台机器上每个工件分解出子工件的个数;$P^k_{i,j}$ 为第 i 个工件分解出的第 j 个子工件在第 k 台机器上的处理时间,其中 $i = 1, 2, \cdots, n$, $j = 1, 2, \cdots, l$, $k = 1, 2$.

算法 1

1) 每个工件的子工件按照在两台机器上处理时间的 Johnson 规则排列连续加工,重新定义子工件的序号为 $1, 2, \cdots, l$.

2) 把每个准备切割的工件当成一个整体,重新定义 a_i, b_i(如图 1 所示) 为

$$a_i = \max\Big\{\sum_{j=1}^{l} P^1_{i,j} - \sum_{j=1}^{l-1} P^2_{i,j},\ P^1_{i,1}\Big\},$$

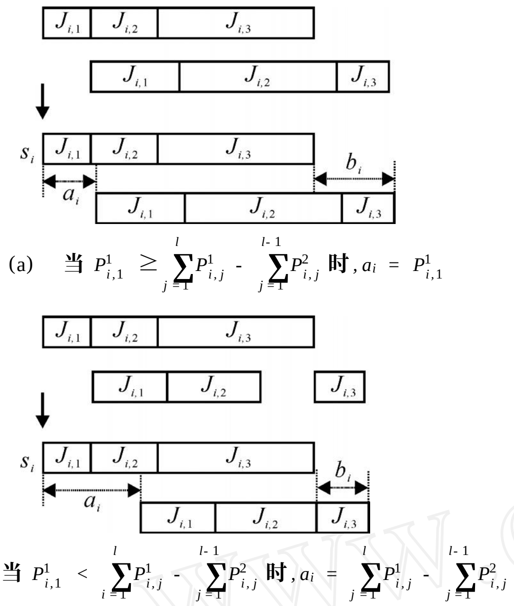

图 1　算法 1 中工件 a_i, b_i 图示

$$b_i = \sum_{j=1}^{l} P^2_{i,j} - \left(\sum_{j=1}^{l} P^1_{i,j} - a_i \right).$$

3) 根据 a_i, b_i 之间大小的关系,把工件分成两类:第 1 类满足 $a_i = b_i$,第 2 类满足 $a_i > b_i$. 对于第 1 类工件按照 a_i 的非减次序排列,对于第 2 类工件按照 b_i 的非增次序排列. 第 1 类工件安排在第 2 类之前加工.

算法 1 为问题的最优算法,这是由标准邻域交换得到的,详细证明略. 同时易见,该算法为多项式时间算法,其时间复杂性为 $O(n\log n)^2$.

例 1　对于问题 $F_2 | m_1, disi | C_{max}$,给出下面的例子,其中 $n = 3$, $l = 2$,所有子工件的处理时间见表 1.

表 1　例 1 的数据

	$P^1_{i,1}$	$P^1_{i,2}$	$P^2_{i,1}$	$P^2_{i,2}$
J_1	0.5	1	1	2
J_2	1	2	1	1
J_3	1	2	2	0.5

用 Johnson 规则确定的工件顺序为 $J_{1,1}$, $J_{1,2}$, $J_{3,1}$, $J_{3,2}$, $J_{2,1}$, $J_{2,2}$,求得最大完成时间为 8.5. 算法 1 确定的工件顺序为 $J_{1,1}$, $J_{1,2}$, $J_{2,1}$, $J_{2,2}$, $J_{3,1}$, $J_{3,2}$,最大完成时间为 $8 < 8.5$.

下面进一步分析当考虑缓冲限制和生产前运输时,是否仍可找到问题的最优解.

2.2　带有缓冲惩罚的分解调度问题

管加工企业中,在制库存通常很有限,体积太大的工件有可能放不进中间库,此时企业就面临一个选择:是将这些工件进行无等待加工以避免库存,还是把这样的大工件存储到其他仓库,同时支付额外存储费用. 下面研究机器 2 前缓冲有限制的问题,并采用实际体积对机器前的缓冲进行度量. 这里认为那些体积超过缓冲限制的工件在下游机器不可利用时,可以选择放入缓冲,但要支付一定的惩罚费用;而那些小于缓冲限制的工件在进入缓冲时,则不需要支付费用.

根据三域表示法[17],把这种工件在第 1 台机器上具有分解特征的两台流水机器求最大完成时间最小问题表示成 $F_2 | m_1, disi, b | C_{max} + P_c$,其中 $P_c = \sum_{i=1}^{n} \alpha_i$. 惩罚系数 α_i 是工件相关的. 对于所有工件定义体积为 $b'_{i,j} = KP^1_{i,j}$. 其中:$i = 1, 2, \cdots, n$; $j = 1, 2, \cdots, l$. 不失一般性,设 $K = 1$.

下面给出两个性质. 其中:第 1 个结论给出了各子工件在机器 1 上的开始时间,第 2 个结论限制了工件在两机器上的加工顺序.

性质 3　子工件在机器 1 上的开始时间或者刚好是其紧前工件在机器 1 上的结束时间,或者满足在机器 1 加工完成时,机器 2 刚好可以利用.

性质 4　在最优调度中,工件在两机器上加工顺序相同.

基于性质 3 和性质 4,下面证明问题 $F_2 | m_1, disi, b | C_{max} + P_c$ 是 NP 难的,并给出拟多项式时间的动态规划算法,从而说明该问题是一个一般意义 NP 难问题. 问题的证明可通过把问题规约到一个已知的 NP 难问题 ——等划分问题来实现. 等划分问题描述如下:给定 $2t$ 项, $T = \{1, 2, \cdots, 2t\}$,每一项都对应一个非负的整数 a_i,且满足对于某个 A 有 $\sum_{i \in T} a_i = 2A$. 问是否存在一个划分满足 $\sum_{i=1}^{t} a_i = \sum_{i=t+1}^{2t} a_i = A$.

定理 1　$F_2 | m_1, disi, b | C_{max} + P_c$ 是一个 NP 难的问题.

证明　构造一个如图 2 的调度.

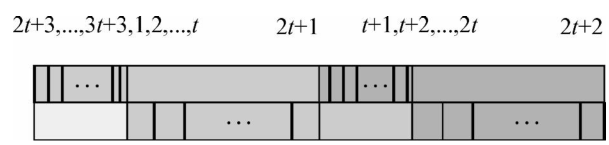

图 2　问题 $F_2 | m_1, disi, b | C_{max} + P_c$ 调度示意

机器 1 上要切割的工件数 $n = 3$,每个工件分解出 $t + 1$ 个子工件. 不失一般性,设第 1 个工件分解出子工件为 $2t + 3, \cdots, 3t + 3$;第 2 个工件分解出的子工件为 $1, 2, \cdots, t, 2t + 1$;第 3 个工件分解出的子工件为 $t + 1, t + 2, \cdots, 2t, 2t + 2$. 所有子工件在两机器上加工顺序为 $2t + 3, \cdots, 3t + 3; 1, 2, \cdots, t, 2t + 1; t + 1, t + 2, \cdots, 2t, 2t + 2$.

所有子工件的处理时间为

$$P_i^1=\begin{cases}a_i,\ i=1,\cdots,2t;\\2A,\ i=2t+1,\ 2t+2;\\0,\ i=2t+3,\cdots,3t+3;\end{cases}$$

$$P_i^2=\begin{cases}2A/t,\ i=1,\cdots,2t;\\0,\ i=2t+2,\cdots,3t+2;\end{cases}$$

$$P_{2t+1}^2=A;\ P_{3t+3}^2=A.$$

缓冲空间限制 $b=A$.

门槛值为 $6A$.

如果等划分问题有解,显然调度可行,且门槛值不超过 $6A$.

下面往证如果目标函数值不超过 $6A$,则等划分问题有解. 问题的证明可由反证法得到. 注意到所有子工件在两机器上的处理时间之和均为 $6A$,这就要求两机器从开始加工时刻起就没有闲置时间,三工件必然如图 2 所示顺序切割,且不能出现存储惩罚工件. 如果 $\sum_{i=1}^{t}a_i>A$,则工件 $2t+1$ 在机器 2 上加工前机器上有闲置时间;如果 $\sum_{i=1}^{t}a_i<A$,则工件 $2t+1$ 会出现存储惩罚,矛盾. 由此得出必有 $\sum_{i=1}^{t}a_i=A$,即等划分问题有解. □

算法 2

定义 $f(i,j,t)$ 为一个部分调度的最大完成时间,其中调度的最后一个工件为 J_i 的第 j 个子工件,$j=1,2,\cdots,l$. 定义 t 为该部分调度在机器 1 上的完成时间,$P_{i,j}^k$ 为工件 J_i 的第 j 个子工件在机器 k 上的完成时间. 设 N_i 为工件 J_i 还没有切割出的子工件集合,定义 I 为还没有全部加工完的工件集合,I_1 为没有加工的工件集合,I_2 为子件全部加工完成的工件集合.

初始解为

$f(0,0,0)=0.$

递归函数为

$$f(i,j,t)=\min_{i\in II}\min_{j\in N_i}\begin{cases}t+P_{i,j}^2,\\\quad t\geq f(i',j',t-P_{i,j}^1);\\f(i',j',t-P_{i,j}^1)+P_{i,j}^2,\\\quad t\leq\min\{f(i',j',t-P_{i,j}^1),\\\quad t-P_{i,j}^1+b\};\\f(i',j',t-P_{i,j}^1)+P_{i,j}^2+\alpha_{i,j},\\\quad t-P_{1,j}^1+b'<t<\\\quad f(i',j',t-P'_{i,j});\\\min\limits_{t'\leq t-P_{i,j}^1}f(i',j',t')+P_{i,j}^2,\\\quad t-P_{i,j}^1+b'<t=f(i',j',t').\end{cases}$$

最优解为

$$\min_{|I|=0,|N_i|=0,1\leq i\leq n,t}f(i,j,t),$$

其中集合 $II=\{\{i\}$ 如果 $i'\in I;I_1$ 如果 $i'\in I_2\}$. 循环的第 1 项表示如果机器 1 上连续加工当前工件完成时,机器 2 可利用,则此时选择当前工件在机器 1 上连续加工. 循环的第 2 项表示当前调度工件体积不超过容量限制,且工件连续加工结束时机器 2 不可利用,此时选择当前工件在机器 1 上连续加工. 第 3 项表示当前调度工件在机器 1 上连续加工完成后机器 2 可利用,且当前子工件超过容量限制,此时选择该工件在机器 1 上连续加工,将会出现存储费用. 第 4 项表示当前调度工件超过容量限制,选择工件在机器 1 上加工前有闲置时间,使得当前子工件在机器 1 上刚加工完成,机器 2 即可利用,因此不会出现存储费用.

定理 2 算法 2 的时间复杂性为 $O(L^2n)$,其中

$$L=\sum_{i=1}^{n}\sum_{j=1}^{l}(P_{i,j}^1+P_{i,j}^2).$$

证明 当 l 给定时 (i,j,t) 共有 $O(Ln)$ 个状态,循环的时间为 $O(L)$,因此算法总的时间复杂性为 $O(L^2n)$. □

定理 3 问题 $F_2|m_1,\mathrm{disi},b|C_{\max}+P_c$ 是一般意义 NP 难的.

定理 4 算法 1 为问题 $F_2|m_1,\mathrm{disi},b|C_{\max}+P_c$ 提供不超过 2 的界.

证明 把算法 1 作为该问题 $F_2|m_1,\mathrm{disi},b|C_{\max}+P_c$ 的启发式算法,所有工件按照算法 1 中的排序加工,此时有 $P_c=\sum_{i=1}^{N}P_i^1=C^{\mathrm{opt}}$. 因此,$C^H/C^*=(C^{\mathrm{opt}}+P_c)/C^{\mathrm{opt}}\leq2$,其中 C^{opt} 为问题 $F_2|m_1,\mathrm{disi}|C_{\max}$ 的最优解. □

2.3 带有前运输的工件分解调度问题

这里重点考虑有一个运输车辆负责切管机前工件的运输,其中两加工机器前的缓冲无限,考虑车辆的运输时间. 如果工件运输到达切管机时机器不可利用,工件等待时不占用运输车辆. 根据三域表示法,把这种机器前带有运输车辆的问题表示成 $T\to F_2|m_1,\mathrm{disi}|C_{\max}$.

定理 5 $T\to F_2|m_1,\mathrm{disi}|C_{\max}$ 是强 NP 难的.

证明 问题的证明可由一个已知的强 NP 难问题 ——三划分问题归约得到. 三划分问题描述如下:有 $3t$ 个非负数,$T=\{1,2,\cdots,3t\}$ 每一项都对应一个介于 $A/4$ 和 $A/2$ 之间的正整数 a_i,有 $\sum_{i=1}^{3t}a_i=tA$,三划分有解,即 T 的 t 个不相交子集 $T_1,T_2,\cdots,T_t$,满足每个子集中恰好有 3 项,且总和为 A.

构造调度如图 3 所示:每个钢管 i 在切管机上分解出 4 个子钢管 $3t+i, 3i-1, 3i-2, 3_i$,每个工件的子工件加工顺序均为 $3t+i, 3i-1, 3i-2, 3i$.

所有子钢管的处理时间分别为

$$P_i^1 = \begin{cases} a_i, & i = 1,2,\cdots,3t; \\ 0, & i = 3t+1, 3t+2, \cdots, 4t; \end{cases}$$

$$P_i^2 = \begin{cases} 0, & i = 1,2,\cdots,3t; \\ A, & i = 3t+1, 3t+2, \cdots, 4t. \end{cases}$$

钢管的前运输往返时间为 $t = A/2$.

门槛值为 tA.

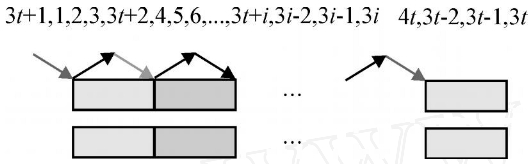

图 3　问题 $T \to F_2 \mid m_1, \text{disi} \mid C_{\max}$ 实例调度示意

如果三划分问题有解,显然调度可行,且门槛值不超过 $tA + A/2$.

下面往证如果目标函数值不超过 $tA + A/2$,则三划分问题有解. 问题的证明可由反证法得到. 注意到生产前至少有一个 $A/2$ 的运输时间,所有子工件在两机器上的处理时间之和均为 tA,这就要求两机器从第 1 个钢管运输到达时刻起就没有闲置时间. 不妨设工件 j 为不满足 $a_{3j-2} + a_{3j-1} + a_{3j} = A$ 的第 1 个工件. 如果,$a_{3j-2} + a_{3j-1} + a_{3j} > A$,则工件 j 的紧后工件在机器 2 上加工前有闲置时间;如果 $a_{3j-2} + a_{3j-1} + a_{3j} < A$,则工件 j 的紧后工件在机器 1 上加工前有闲置时间. 可见对所有的 j 都有 $a_{3j-2} + a_{3j-1} + a_{3j} = A$,即三划分问题有解. □

定理 6　算法 1 为问题 $T \to F_2 \mid m_1, \text{disi} \mid C_{\max}$ 提供不超过 2 的界.

证明　考虑运输的两台机器流水问题有最优解,不妨设其为 C^*,所有 n 个工件的运输时间最少为 $2nt - t$. C^*和 $2nt - t$ 均为考虑运输的两台机器流水切割问题的下界. 问题的上界不超过 $C^* + 2nt - t$,此时有

$$C^H / C^{\text{LB1}} = (C^* + 2nt - t) / C^* = 1 + (2nt - t) / C^*.$$

另一方面,

$$C^H / C^{\text{LB2}} = (C^* + 2nt - t) / (2nt - t) = 1 + C^* / (2nt - t).$$

可见,启发式算法的界不超过 2. □

2.4　实验与数值计算结果

对于近似算法的评价,除了界分析等理论分析之外,还可通过仿真实验做进一步的验证,如文献[18,19]. 这里针对算法 1 作为 2.2 节和 2.3 节中两个 NP 难问题的启发式算法给出数值实验. 算法是由 C 语言编程,在 Pentium- Ⅳ的 PC 机上运行,操作系统是 XP,CPU 是 2.40 GHz. 分别对于工件个数 $n \in \{5, 15, 20, 25\}$,子工件个数 $l \in \{3, 4, 5\}$ 的每种组合,针对 $b \in \{7, 8, 9\}$ 和 $t \in [1, 5]$, $t \in [1, 10]$, $t \in [1, 15]$ 的每个问题各产生 10 个算例,处理时间 $P_{i,j}^1$, $P_{i,j}^2$ 分别在 $[1, 10]$ 内服从均匀分布.

2.4.1　问题 $F_2 \mid m_1, \text{disi}, b \mid C_{\max} + P_c$

定义该问题采用算法 1 对工件排序求得的目标函数值为 C_1,其下界为采用算法 1 确定工件和子工件顺序求得所有工件的最大完成时间(不考虑惩罚费用),记作 LB_1,并用 C_1/LB_1 表示启发式与下界的间隙,详见表 2.

表 2　C_1 相对于 LB_1 的改进量

n	l	$b = 7$	$b = 8$	$b = 9$
5	3	1.29167	**1.00000**	**1.00000**
	4	**1.53333**	1.08000	**1.00000**
	5	1.42177	1.21481	1.07194
10	3	1.15135	1.35294	1.00000
	4	1.29493	1.30738	1.17021
	5	1.49147	1.29814	1.10381
15	3	1.45357	1.32432	1.21739
	4	1.45915	1.33939	1.17241
	5	1.43066	1.34812	1.14354
20	3	1.33739	1.22515	**1.00000**
	4	1.45714	1.29647	1.06397
	5	1.41361	1.25418	1.11385
25	3	1.33248	1.23918	1.19465
	4	1.40975	1.27188	1.08741
	5	1.26294	1.28468	1.07174
平均值		1.38275	1.25578	1.09406

由表 2 的数据分析可得如下结论:

1) 对于随机产生的数据,启发式与下界间隙最大不超过 54 %,最好情况下启发式可以达到最优解;

2) 随着缓冲惩罚系数的增大,启发式与下界的间隙明显变小;

3) 算法解的质量随着工件个数以及子工件的变化改进不明显.

2.4.2　问题 $T \to F_2 \mid m_1, \text{disi} \mid C_{\max}$

定义该问题采用算法 1 对工件进行排序求得的目标函数值为 C_2,其下界定义为分别忽略运输、第 1 阶段生产以及第 2 阶段生产求得 3 个 $C_{\max}$ 中最大的一个,记作 LB_2,并用 C_2/LB_2 表示启发式与下界的间隙,详见表 3.

表 3 C_2 相对于 LB_2 的改进量

n	l	$t \in [1,5]$	$t \in [1,10]$	$t \in [1,15]$
5	3	1.01020	1.04040	1.07407
	4	1.00840	1.01010	1.03240
	5	1.00633	1.06135	1.01183
10	3	1.01786	1.00575	1.06148
	4	1.00461	1.05285	1.00405
	5	1.00380	1.00360	1.00385
15	3	1.00394	1.01460	1.00366
	4	1.00279	**1.10000**	1.00271
	5	1.00235	1.00240	1.01422
20	3	1.00265	1.00321	1.00279
	4	1.00911	1.01190	1.02227
	5	1.00196	1.00723	1.00176
25	3	1.00218	1.04829	1.00243
	4	1.00349	1.03265	1.03597
	5	1.00150	1.00618	**1.00138**
平均值		1.00541	1.02670	1.01832

由表 3 的数据分析可得如下结论:

1) 对于实验产生的数据,启发式与问题下界的间隙很小,最大值不超过 10 %,最小值可以达到 0.138 %. 当运输时间取 $t \in [1,5]$ 时平均间隙最小,可以达到0.541 %;当运输时间取 $t \in [1,10]$ 时平均间隙较大,为 2.67 %.

2) 随着工件个数、子工件个数以及运输时间的变化,解的质量改进并不明显.

3 结 论

本文研究了两台机器流水车间分解类型调度问题,并给出了 $O(n\log n)^2$ 时间的最优算法. 对于更复杂的缓冲有限制的情况,证明该问题是一般意义 NP 难的,并给出了 $O(L^2 n)$ 时间的拟多项式时间算法,同时进行了界分析. 对于考虑生产前运输时间的复杂情况,证明其为强 NP 难问题,并进行了界分析和数值实验. 实验结果表明了算法的有效性.

参考文献(References)

[1] Allahverdi A. The two- and m-machine flowshop scheduling problems with bicriteria of makespan and mean flowtime[J]. European J of Operational Research, 2003, 147(2): 373-396.

[2] Allahverdi A, Aldowaisan T. Two-machine flowshop scheduling to minimize total completion time with separate setup and removal times[J]. Int J of Industrial Engineering-theory Application and Practice, 2002, 9(2): 275-286.

[3] Gupta J N D, Neppalli V R, Werner F. Minimizing total flow time in a two-machine flowshop problem with minimum makespan[J]. Int J of Production Economics, 2001, 69(3): 323-338.

[4] Cepek O, Okada M, Vlach M. Note: On the two-machine no-idle flowshop problem[J]. Naval Research Logistics, 2000, 47(4): 353-358.

[5] Allahverdi A, Aldowaisan T. Minimizing total completion time in a no-wait flowshop with sequence-dependent additive changeover times [J]. J of Operational Research Society, 2001, 52(4): 449-462.

[6] Sung C S, Kim Y H. Minimizing makespan in a two-machine flowshop with dynamic arrivals allowed[J]. Computers and Operations Research, 2002, 29(3): 275-294.

[7] Lin B M T, Cheng T C E. Batch scheduling in the no-wait two-machine flowshop to minimize the makespan [J]. Computers and Operations Research, 2001, 28(7): 613-624.

[8] Vickson R G, Alfredsson B E. Two- and three-machine flow shop scheduling problems with equal sized transfer families[J]. Int J of Production Research, 1992, 30(7): 1551-1574.

[9] Cheng T C E, Gupta J N D, Wang G Q. A review of flowshop scheduling research with setup times[J]. Production and Operations Management, 2000, 9(3): 262-282.

[10] Potts C N, Sevast'janov S V, Strusevich V A, et al. The two-stage assembly scheduling problem: Complexity and approximation [J]. Operations Research, 1995, 43(2): 346-355.

[11] Lin B M T, Cheng T C E. Fabrication and assembly scheduling in a two-machine flowshop[J]. IIE Trans, 2002, 34(11): 1015-1020.

[12] Cheng T C E, Wang G Q. Scheduling the fabrication and assembly of components in a two machine flowshop [J]. IIE Trans, 1999, 31(2): 135-143.

[13] Yang W H. Scheduling two-component products on parallel machines[J]. Omega, 2004: 32(5): 353-359.

[14] Koulamas C, Kyparisis G J. Concurrent flowshop scheduling to minimize makespan[J]. European J of Operational Research, 2004, 156(2): 524-529.

[15] Koulamas C, Kyparisis G J. The three-stage assembly flowshop scheduling problem [J]. Computers and Operations Research, 2001, 28(7): 689-704.

[16] Namman N, Rom R. Analysis of transmission scheduling with packet fragmentation[J]. Discrete Mathematics and Theoretical Computer Science, 2001, 4(1): 139-156.

(下转第 65 页)

[2] Akyildiz L F, Su W, Sankarasubramaniam Y, et al. A survey on sensor networks[J]. IEEE Communications Magazine, 2002, 40(8): 102-114.

[3] 于海斌, 曾鹏, 梁韡. 智能无线传感器网络系统[M]. 北京: 科学出版社, 2006.
(Yu H B, Zeng P, Liang W. Intelligent wireless sensor networks systems[M]. Beijing: Science Press, 2006.)

[4] Jennings N R, Sycara K, Wooldridge M J. A roadmap of agent research and development[J]. Autonomous Agent and Multi-agent Systems, 1998, 1(1): 7-38.

[5] Zambonelli F, Omicini A. Challenges and research directions in agent-oriented software engineering[J]. Autonomous Agents and Multi-agent Systems, 2004, 9(3): 253-283.

[6] Lesser V, Ortiz C, Tambe M. Distributed sensor networks: A multiagent perspective[M]. Boston: Kluwer Publishers, 2003.

[7] Horling B, Mailler R, Sims M, et al. Using and maintaining organization in a large-scale distributed sensor network[C]. Proc of the Workshop on Autonomy, Delegation, and Control. Melbourne, 2003.

[8] Krishnamachari B, Wicker S B, Béjar R, et al. On the complexity of distributed self-configuration in wireless networks[J]. Telecommunication Systems, 2003, 22(1-4): 33-59.

[9] Soh L K, Tsatsoulis C. Reflective negotiating agents for real-time multisensor target tracking[C]. Proc of the Int Joint Conf on Artificial Intelligence. Seattle, 2001: 1121-1127.

[10] Smith R G. The contract net protocol: High level communication and control in distributed problem solver[J]. IEEE Trans on Computers, 1980, 29(12): 1104-1113.

[11] 于振华, 蔡远利. 基于面向对象 Petri 网的软件体系结构描述语言[J]. 西安交通大学学报, 2004, 38(12): 1236-1240.
(Yu Z H, Cai Y L. Software architecture description language based on object-oriented Petri nets[J]. J of Xi'an Jiaotong University, 2004, 38(12): 1236-1240.)

[12] Huynh T D, Jennings N R, Shadbolt N R. An integrated trust and reputation model for open multi-agent systems[J]. J of Autonomous Agents and Multi-agent Systems, 2006, 13(2): 119-154.

[13] Xiong C, Murata T, Tsai J. Modeling and simulation of routing protocol for mobile Ad Hoc networks using colored Petri nets[C]. Proc of the Conf on Application and Theory of Petri Nets: Formal Methods in Software Engineering and Defence Systems. Adelaide, 2002: 145-153.

[14] Haines R J, Clemo G R, Munro A T D. Petri-nets for formal verification of MAC protocols[J]. IET Software, 2007, 1(2): 39-47.

[15] Shatz S M, Tu S, Murata T, et al. An application of Petri net reduction for Ada tasking deadlock analysis[J]. IEEE Trans on Parallel and Distributed Systems, 1996, 7(12): 1307-1322.

[16] Roch S, Starke P H. INA: Integrated net analyzer, Version 2.2[EB/OL]. www2.informatik.huberlin.de/～starke/ina.html, 2007.

[17] Heinzelman W B, Chandrakasan A P, Balakrishnan H. An application specific protocol architecture for wireless microsensor networks[J]. IEEE Trans on Wireless Communications, 2002, 1(4): 660-670.

(上接第 60 页)

[17] Graham R L, Lawler E L, Lenstra J K, et al. Optimization and approximation in deterministic sequencing and scheduling theory: A survey[J]. Annals of Discrete Mathematics, 1979, 5(1): 287-326.

[18] 冯大光, 唐立新. 单台批处理机总加权完成时间最小化的启发式算法[J]. 控制与决策, 2006, 21(11): 1293-1297.
(Feng D G, Tang L X. Heuristic algorithms for single batching machine with total weighted completion time[J]. Control and Decision, 2006, 21(11): 1293-1297.)

[19] 金锋, 宋士杰, 吴澄. 一类基于 TSP 问题 Block 性质的快速 TS 算法[J]. 控制与决策, 2007, 22(3): 247-251.
(Jin F, Song S J. Wu C. Fast TS algorithm based on Block properties of TSP[J]. Control and Decision, 2007, 22(3): 247-251.)