

# 等待时间受限的置换流水车间调度启发式算法<sup>①</sup>

王柏林<sup>1,2</sup>, 李铁克<sup>1,2</sup>

(1. 北京科技大学东凌经济管理学院, 北京 100083;

2. 钢铁生产制造执行系统技术教育部工程研究中心, 北京 100083)

**摘要:** 针对等待时间受限的置换流水车间调度问题, 分析了其可行解与流水车间调度最优解的关系, 给出了计算最大完工时间的有向图, 证明了等待时间受限的置换流水车间调度问题的可逆性, 并以此为基础提出了一种启发式算法. 算法首先根据等待时间受限约束与无等待(no-wait)约束的相似特征, 生成初始工件序列集; 然后利用问题可逆性给出了复杂度为  $O(n^2m)$  的插入优化机制, 进一步优化初始解. 数据实验的结果验证了启发式算法的可行性和有效性.

**关键词:** 生产调度; 置换流水车间; 等待时间受限; 启发式; 可逆性

**中图分类号:** TP278    **文献标识码:** A    **文章编号:** 1007-9807(2012)06-0022-11

## 0 引言

等待时间受限的流水车间调度问题要求工件在相邻两阶段间的等待时间不超过给定的时间上限, 它广泛存在于中间产品性质不稳定或高温连续作业的流程性工业中, 例如食品加工、玻璃制造、钢铁生产等. 目前关于这类流水车间调度问题的研究成果不多, 由于问题本身具有很强的工业背景和理论研究意义, 现已引起研究者的关注. 文献[1]初步研究了等待时间受限的两机流水车间调度问题, 证明了 NP 难特性, 并采用分支定界法求解; 文献[2]针对第一台机器为批处理机的两机流水车间环境下的调度问题, 建立了混合整数规划模型, 并提出启发式方法求解; 文献[3]考虑了等待时间同时存在上下限的置换流水车间调度问题, 指出该问题是强 NP 难的, 并给出了分支定界算法; 文献[4]对不同车间背景下等待时间受限的调度问题的建模机制, 从工件和机器两个角度进行了研究; 文献[5]分析了等待时间上限与可行调度的解析关系以及目标函数的特殊性, 提出了一种启发式算法; 文献[6]针对等待时

间受限的流水车间问题, 采用嵌入约束满足和邻域搜索技术的混合遗传算法求解; 文献[7]将问题进一步扩展到混合流水车间调度, 考虑了交货期要求, 提出回溯、启发式修复与邻域搜索相结合的混合算法.

置换流水车间调度(permutation flowshop scheduling)是一类重要的流水车间调度问题, 要求所有机器上的工件加工顺序均相同. 现有的流水车间调度方法研究多以置换流水车间为背景<sup>[8,9]</sup>. 本文针对等待时间受限的置换流水车间调度问题, 探讨问题的性质特征, 在此基础上设计启发式算法求解, 并通过数据实验考察算法的可行性和有效性.

## 1 问题描述及分析

### 1.1 问题描述

等待时间受限的置换流水车间调度问题可以描述为:  $n$  个工件  $\{J_1, J_2, \dots, J_n\}$  以相同次序在  $m$  台机器  $\{M_1, M_2, \dots, M_m\}$  上加工, 即任意工件均先经过  $M_1$ , 再经过  $M_2$ , 顺次加工, 最后经过  $M_m$ , 且

<sup>①</sup> 收稿日期: 2011-01-11; 修订日期: 2011-04-14.

基金项目: 国家自然科学基金资助项目(70771008); 中央高校基本科研业务费专项资金资助项目(FRF-AS-09-007B; FRF-TR-12-A16A).  
作者简介: 王柏林(1983—), 女, 河北石家庄人, 博士, 讲师. Email: wangbailin\_83@sina.com

所有机器上的工件加工顺序都是相同的; 工件  $J_i$  在机器  $M_j$  上的加工时间已知, 记为  $p_{ij}$ . 问题具有以下约束: 一个工件在某一时刻只能在一台机器上加工(时序约束); 一台机器在某一时刻只能加工一个工件(析取约束); 且由于工艺限制, 工件  $J_i (i = 1, \dots, n)$  在相邻机器  $M_j$  和  $M_{j+1}$  间的等待时间  $w_{ij}$  不能超过给定上限值  $\alpha$  (等待时间受限约束). 调度的目标是确定各台机器上的工件加工顺序和完工时间  $C_{ij}$ , 使得最大完工时间  $C_{\max}$  (makespan) 最小化. 应用 Graham<sup>[10]</sup> 提出的三元组表示法, 此问题可表示为  $Fm | w_{ij} \leq \alpha, prmu | C_{\max}$ .

### 1.2 问题分析

与流水车间调度问题相比, 置换调度具有较小的解空间 ( $n!$ ), 且在不考虑等待时间受限的情况下, 置换流水车间调度解可以作为近优解指导流水车间调度问题的求解. 因此, 置换流水车间调度方法的研究在一般流水车间调度(即不考虑等待时间受限约束的调度问题) 领域占有重要地位. 而在等待时间受到上限约束时, 置换流水车间调度问题是否依然与流水车间调度存在密切关联, 这将通过以下定理做进一步探讨.

**定理 1** 对于  $Fm | w_{ij} \leq \alpha | C_{\max}$  问题, 令  $\pi$  表示满足置换流水车间要求的任一可行解,  $C_{\max}(opt)$  表示流水车间调度的最优值, 则有

$$C_{\max}(\pi) / C_{\max}(opt) \leq m \quad (1)$$

**证明** 若工件在相邻两阶段间不允许等待, 即要求等待时间严格为 0, 这一条件称之为无等待 (no-wait) 约束. 可以看出, 无等待 (no-wait) 约束是等待时间受限的一种极端情况, 即上限  $\alpha = 0$ , 而无等待流水车间调度问题  $Fm | nwt | C_{\max}$  作为等待时间受限流水车间调度的一类特例问题, 其任意可行解  $\pi$  显然也是与其具有相同问题结构(即机器数、工件数及加工时间均相同), 仅等待时间上限不同的相应问题  $Fm | w_{ij} \leq \alpha | C_{\max}$  的可行解. 因而对于  $Fm | w_{ij} \leq \alpha | C_{\max}$  问题有  $C_{\max}(\pi) \leq C_{\max}^0(\pi)$ , 其中  $C_{\max}^0(\pi)$  为无等待 (no-wait) 约束下调度解  $\pi$  的 makespan 值.

无等待流水车间调度  $Fm | nwt | C_{\max}$  是一类特殊的 TSP 问题, 由  $n + 1$  个节点  $J_1, \dots, J_n, J_{n+1}$  构

成, 其中  $J_{n+1}$  为虚拟工件, 在各机器上的加工时间均为 0. 节点  $J_i$  和  $J_j$  之间的距离  $d_{ij}$  定义为  $J_i$  作为  $J_j$  的紧前工件时两工件在第一台机器上的开工时间之差, 计算公式如下

$$d_{ij} = \max_{1 \leq k \leq m} \left\{ \sum_{h=1}^k p_{ih} - \sum_{h=1}^{k-1} p_{jh} \right\}$$

对于工件序列  $\pi$  可将虚拟工件  $J_{n+1}$  作为最后加工的工件, 有

$$C_{\max}^0(\pi) = \sum_{i=1}^n \max_{1 \leq k \leq m} \left\{ \sum_{h=1}^k p_{\pi(i)h} - \sum_{h=1}^{k-1} p_{\pi(i+1)h} \right\}$$

其中  $\pi(i)$  为序列  $\pi$  中第  $i$  个加工工件. 由此得到的  $C_{\max}^0(\pi)$  即为工件序列  $\pi$  能够获得的最小 makespan 值<sup>[11]</sup>.

因有  $\max_{1 \leq k \leq m} \left\{ \sum_{h=1}^k p_{\pi(i)h} - \sum_{h=1}^{k-1} p_{\pi(i+1)h} \right\} \leq \sum_{j=1}^m p_{\pi(i)j}$ , 所以  $C_{\max}^0(\pi) \leq \sum_{i=1}^n \sum_{j=1}^m p_{\pi(i)j}$ , 有

$$\frac{C_{\max}(\pi)}{C_{\max}(opt)} \leq \frac{C_{\max}^0(\pi)}{C_{\max}(opt)} \leq \frac{\sum_{i=1}^n \sum_{j=1}^m p_{\pi(i)j}}{C_{\max}(opt)} \leq$$

$$\frac{m \max_{1 \leq j \leq m} \left\{ \sum_{i=1}^n p_{\pi(i)j} \right\}}{C_{\max}(opt)}$$

显然对于  $C_{\max}(opt)$  有  $C_{\max}(opt) \geq \max_{1 \leq j \leq m} \left\{ \sum_{i=1}^n p_{\pi(i)j} \right\}$ , 因而  $C_{\max}(\pi) / C_{\max}(opt) \leq m$ .

证毕.

定理 1 分析了等待时间受限时, 任意置换调度可行解的 makespan 值与流水车间调度最优值的关系. 而对于置换调度最优解, 文献 [5] 给出了定理 2.

**定理 2** 对于  $Fm | w_{ij} \leq \alpha | C_{\max}$  问题, 如果等待时间上限  $\alpha$  满足  $0 \leq \alpha < 2 \min\{p_{ij}\}$ , 其可行解只存在于置换调度中.

定理 2 说明, 在一定条件下, 等待时间受限的置换流水车间调度问题最优解也是流水车间最优调度解. 结合定理 1 可知, 在等待时间受限的约束作用下, 置换流水车间调度问题的可行解与流水车间调度问题的最优解之间存在着相近关系, 对其求解方法的探索对于流水车间调度问题的研究具有重要意义. 其中, 启发式作为调度领域的一类传统求解方法, 能够利用问题领域的知识快速获

得比较好的调度解,具有方法简单、易于实现、计算复杂度低等特点<sup>[12]</sup>. 为此,本文将结合问题特征设计启发式方法,实现问题求解.

## 2 完工时间表示方法

对于  $Fm | w_{ij} \leq \alpha, prmu | C_{max}$ , 主要解决两个问题: 1) 确定工件的加工顺序, 即工件序列; 2) 对于给定的工件序列, 确定工件的完工时间. 其中,

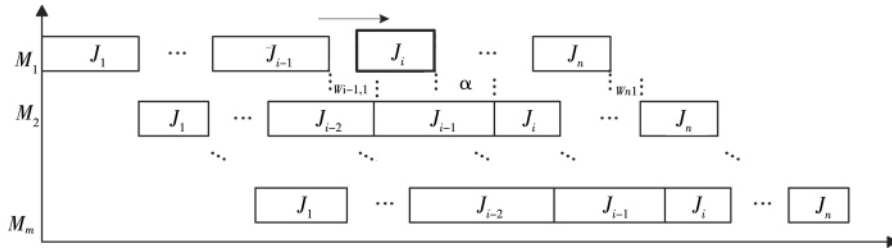


图1  $Fm | w_{ij} \leq \alpha, prmu | C_{max}$  问题调度甘特图

Fig. 1 The Gantt chart of schedules in  $Fm | w_{ij} \leq \alpha, prmu | C_{max}$

基于这一思想,文献[3]给出了完工时间的计算方法 Timetabling, 令  $\pi$  表示给定的工件序列, 则各工件的完工时间计算如下.

步骤1  $C_{\pi(1)1} \leftarrow p_{\pi(1)1}$ , 令  $j$  从 2 到  $m$ , 分别计算  $C_{\pi(1)j} \leftarrow C_{\pi(1)j-1} + p_{\pi(1)j}$ ;

步骤2 令  $i$  从 2 到  $n$ , 分别执行以下步骤

1)  $C_{\pi(i)1} \leftarrow C_{\pi(i-1)1} + p_{\pi(i)1}$ ;

2) 令  $j$  从 2 到  $m$ , 分别计算  $C_{\pi(i)j} \leftarrow \max\{C_{\pi(i-1)j}, C_{\pi(i)j-1}\} + p_{\pi(i)j}$ ;

3) 令  $j$  从  $m-1$  到 1, 若  $C_{\pi(i)j+1} - C_{\pi(i)j} > p_{\pi(i)j+1} + \alpha$ , 则令  $C_{\pi(i)j} \leftarrow C_{\pi(i)j+1} - p_{\pi(i)j+1} - \alpha$ ;

步骤3  $C_{max}(\pi) \leftarrow C_{\pi(n)m}$  输出  $C_{max}(\pi)$  算法结束.

上述算法的复杂度为  $O(nm)$ .

### 2.2 makespan 的有向图

对于给定的工件序列  $\pi$ , 若增加虚拟机器  $M_0$  作为首先加工阶段, 且所有工件在  $M_0$  上的加工时间为零, 即  $p_{i0} = 0 (i = 1, \dots, n)$ , 工件在  $M_0$  和  $M_1$  间的等待时间也要求满足上限  $\alpha$  的约束, 可以得到如图 2 所示的有向图  $G$ . 工件  $J_{\pi(i)}$  在机器  $M_j$  上的操作与节点  $x_{ij}$  对应, 节点  $y_{ij}$  为辅助节点; 工件间的时序约束、析取约束和等待时间受限约束通过节点  $y_{i,j-1}$  和  $x_{i-1,j}$  通往节点  $x_{ij}$  的两条加权入弧

前者是决定算法有效性的关键, 后者则是保证算法可行性的基础. 因此, 首先对问题 2) 进行分析.

### 2.1 完工时间的计算方法

由于工件在相邻机器间的等待时间存在上界限制, 计算  $Fm | w_{ij} \leq \alpha, prmu | C_{max}$  的最大完工时间不同于一般置换流水车间调度问题, 在工件的等待时间超出上限时, 需要逆向调整相应工件在前几台机器上的完工时间以使调度可行, 如图 1 所示.

以及节点  $y_{ij}$  和  $x_{i,j+1}$  通往节点  $x_{ij}$  的两条加权入弧来表示. 因而在有向图  $G$  中, 工件  $J_{\pi(i)}$  离开机器  $M_j$  的时间  $C_{\pi(i)j}$  可以通过计算从节点  $x_{10}$  到  $x_{ij}$  的最大加权路径长度来得到; 从节点  $x_{10}$  到  $y_{ij}$  最长路径的距离则表示不考虑等待时间受限约束时  $J_{\pi(i)}$  在  $M_j$  上的最早完工时间;  $\pi$  的最小 makespan 值  $C_{max}(\pi)$  与从节点  $x_{10}$  到节点  $x_{nm}$  的最大加权路径长度相对应.

考虑如下  $Fm | w_{ij} \leq \alpha, prmu | C_{max}$  实例, 其中  $n = 3, m = 3, \alpha = 1$ ,

$$P = \begin{pmatrix} p_{11} & p_{21} & p_{31} \\ p_{12} & p_{22} & p_{32} \\ p_{13} & p_{23} & p_{33} \end{pmatrix} = \begin{pmatrix} 2 & 2 & 8 \\ 4 & 7 & 2 \\ 6 & 4 & 3 \end{pmatrix}$$

与工件序列  $\pi = \{J_1, J_2, J_3\}$  对应的有向图和甘特图如图 3 所示, 有  $C_{max}(\pi) = 20$ .

易见有向图  $G$  中节点  $x_{10}$  到达节点  $y_{nm}$  与到达节点  $x_{nm}$  的最大加权路径长度是相同的, 均为调度解的 makespan 值. 因此, 记图 2 中方框部分的有向图为  $G'$ , 则  $G'$  亦可获得序列  $\pi$  的最大完工时间. 根据有向图  $G'$  可以得到等待时间受限的置换流水车间调度问题的可逆性定理. 这里首先给出逆问题的定义.

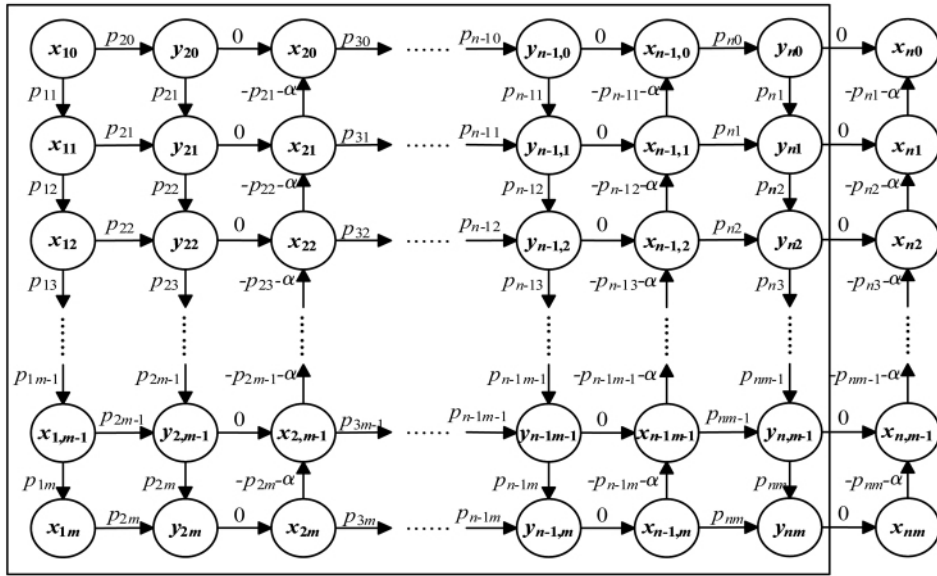
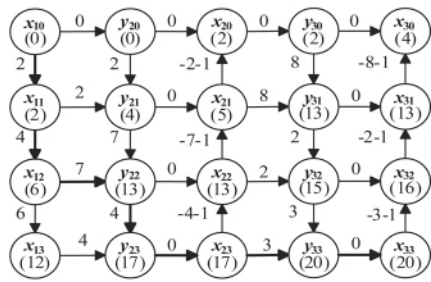
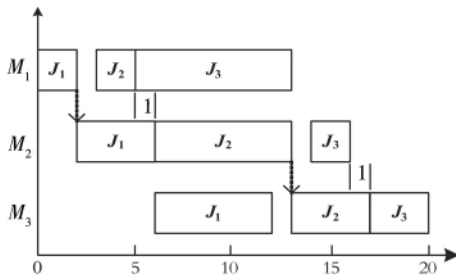


图 2 计算调度方案 makespan 的有向图

Fig. 2 The directed graph for the makespan computation of a permutation schedule



(a) 实例有向图



(b) 实例甘特图

图 3 调度实例的有向图和甘特图

Fig. 3 The directed graph and Gantt chart for the given problem

**定义 1** 对于两个具体的  $Fm | w_{ij} \leq \alpha, prmu | C_{max}$  问题, 令  $p_{ij}^{(1)}$  和  $p_{ij}^{(2)}$  分别表示这两个问题中工件  $J_i$  在机器  $M_j$  上的加工时间, 若满足  $p_{ij}^{(1)} = p_{i, m-j+1}^{(2)}$ , 则这两个问题互为对方的逆问题.

**定理 3**  $Fm | w_{ij} \leq \alpha, prmu | C_{max}$  问题的可逆性 (reversibility): 对于可行调度  $\pi$ , 若工件按相反的顺序  $\bar{\pi}$  逆向经过各阶段的机器 (由  $M_m$  到  $M_1$ ), 则 makespan 值不变.

**证明** 显然对于一个给定的  $Fm | w_{ij} \leq \alpha, prmu | C_{max}$  及其逆问题, 若以下结论成立则定理得证: 若原问题中工件按序列  $\pi = \{i_1, i_2, \dots, i_n\}$  加工, 逆问题中工件按  $\pi$  的逆序  $\bar{\pi} = \{i_n, i_{n-1}, \dots, i_1\}$  加工, 则这两个调度解具有相同的 makespan 值.

现证明这一结论: 记原问题中阶段  $j$  的机器为  $M_j$ , 逆问题中为  $\bar{M}_j$ , 则有  $M_j = \bar{M}_{m-j+1}$ . 在原问题中增设虚拟机器  $M_0$ , 逆问题相应增设  $\bar{M}_{m+1}$ , 各工件在这两台机器上的加工时间均为零. 若原问题按工件序列  $\pi$  加工, 对应有向图  $G'$  可以得到调度解的 makespan, 那么将图  $G'$  的有向弧反转方向即可得到逆问题按序列  $\bar{\pi}$  加工时计算 makespan 的有向图  $\bar{G}'$ , 其中, 从节点  $y_{nm}$  到  $y_{ij}$  的最大路径长度表示序列  $\bar{\pi}$  的第  $n-i+1$  个工件 (工件  $i_i$ ) 在机器  $\bar{M}_{m-j+1}$  上的开工时间, 从节点  $y_{nm}$  到  $x_{10}$  的最大路径长度表示序列  $\bar{\pi}$  的最后加工工件在  $\bar{M}_{m+1}$  的开工时间, 等于在机器  $\bar{M}_m$  上的完工时间, 即逆问题按序列  $\bar{\pi}$  加工的 makespan 值. 显然有向图  $G'$  和  $\bar{G}'$  的节点和弧权重一一对应, 仅有向弧的方向不同, 因而这两个问题对应的调度解  $\pi$  和  $\bar{\pi}$  的 makespan 相同. 证毕.

### 2.3 完工时间的数学表达式

根据 Timetabling 方法和有向图  $G'$ , 可以得到给定工件序列  $\pi$  中, 工件  $J_{\pi(i)}$  在机器  $M_j$  上完工时间的数学表达式.

引理1  $C_{\pi(i)j} = \max_{1 \leq k \leq j, j+1 \leq l \leq m} \left\{ C_{\pi(i-1)k} + \sum_{t=k}^j P_{\pi(i)t} \right.$   
 $C_{\pi(i-1)l} - \sum_{t=j+1}^{l-1} P_{\pi(i)t} - \alpha(l-j) \left. \right\}$  (2)

证明 数学归纳法.

令  $C'_{\pi(i)j}$  表示不考虑等待时间受限的情况下, 工件  $J_{\pi(i)}$  在  $M_j$  上的最早完工时间 则有

$$C'_{\pi(i)j} = \max\{C'_{\pi(i)j-1}, \mathcal{C}_{\pi(i-1)j}\} + P_{\pi(i)j}$$

$$= \max\{\max\{C'_{\pi(i)j-2}, \mathcal{C}_{\pi(i-1)j-1}\} + P_{\pi(i)j-1}, \mathcal{C}_{\pi(i-1)j}\} + P_{\pi(i)j}$$

$$= \max\{C'_{\pi(i)j-2} + P_{\pi(i)j-1} + P_{\pi(i)j}, C_{\pi(i-1)j-1} + P_{\pi(i)j-1} + P_{\pi(i)j}, C_{\pi(i-1)j} + P_{\pi(i)j}\}$$

$$= \dots$$

$$= \max\left\{C_{\pi(i-1)1} + \sum_{t=1}^j P_{\pi(i)t}, \mathcal{C}_{\pi(i-1)2} + \sum_{t=2}^j P_{\pi(i)t}, \dots, \mathcal{C}_{\pi(i-1)j} + P_{\pi(i)j}\right\}$$

$$= \max_{1 \leq k \leq j} \left\{ C_{\pi(i-1)k} + \sum_{t=k}^j P_{\pi(i)t} \right\}$$

1) 当  $j = m$  时, 显然工件  $J_{\pi(i)}$  在最后一台机器上的完工时间满足  $C_{\pi(i)m} = C'_{\pi(i)m}$ , 即

$$C_{\pi(i)m} = C'_{\pi(i)m} = \max_{1 \leq k \leq m} \left\{ C_{\pi(i-1)k} + \sum_{t=k}^m P_{\pi(i)t} \right\}$$

因此, 当  $j = m$  时, 引理成立.

2) 假设  $j = r + 1$  时, 引理成立 即

$$C_{\pi(i)r+1} = \max_{1 \leq k \leq r+1, r+2 \leq l \leq m} \left\{ C_{\pi(i-1)k} + \sum_{t=k}^{r+1} P_{\pi(i)t} \right.$$

$$C_{\pi(i-1)l} - \sum_{t=r+2}^{l-1} P_{\pi(i)t} - \alpha(l-r-1) \left. \right\}$$

下面将证明当  $j = r$  时, 引理同样成立.

由  $C_{\pi(i)r+1}$  可以逆向推得工件  $J_{\pi(i)}$  在  $M_r$  上的完工时间, 为满足等待时间受限约束, 有

$$C_{\pi(i)r} = \max\{C'_{\pi(i)r}, \mathcal{C}_{\pi(i)r+1} - P_{\pi(i)r+1} - \alpha\}$$

$$= \max\left\{\max_{1 \leq k \leq r} \left\{ C_{\pi(i-1)k} + \sum_{t=k}^r P_{\pi(i)t} \right\}, \max_{1 \leq k \leq r+1} \left\{ C_{\pi(i-1)k} + \sum_{t=k}^{r+1} P_{\pi(i)t} \right\} - P_{\pi(i)r+1} - \alpha, \max_{r+2 \leq l \leq m} \left\{ C_{\pi(i-1)l} - \sum_{t=r+2}^{l-1} P_{\pi(i)t} - \alpha(l-r-1) \right\} -$$

$$P_{\pi(i)r+1} - \alpha \left. \right\}$$

$$= \max\left\{\max_{1 \leq k \leq r} \left\{ C_{\pi(i-1)k} + \sum_{t=k}^r P_{\pi(i)t} \right\}, C_{\pi(i-1)r+1} - \alpha, \max_{r+2 \leq l \leq m} \left\{ C_{\pi(i-1)l} - \sum_{t=r+1}^{l-1} P_{\pi(i)t} - \alpha(l-r) \right\}\right\}$$

$$= \max_{1 \leq k \leq r, r+1 \leq l \leq m} \left\{ C_{\pi(i-1)k} + \sum_{t=k}^r P_{\pi(i)t}, C_{\pi(i-1)l} - \sum_{t=r+1}^{l-1} P_{\pi(i)t} - \alpha(l-r) \right\}$$

因此, 若  $j = r + 1$  时引理成立, 则  $j = r$  时, 引理同样成立.

综合 1)、2) 可得, 对于  $1 \leq j \leq m$ , 工件  $J_{\pi(i)}$  在机器  $M_j$  上的完工时间满足

$$C_{\pi(i)j} = \max_{1 \leq k \leq j, j+1 \leq l \leq m} \left\{ C_{\pi(i-1)k} + \sum_{t=k}^j P_{\pi(i)t}, C_{\pi(i-1)l} - \sum_{t=j+1}^{l-1} P_{\pi(i)t} - \alpha(l-j) \right\}$$
 证毕.

### 3 启发式算法

通过对工件完工时间的分析, 本文进一步提出启发式算法 HLW(heuristics for limited waiting-times), 将调度问题的求解过程分为两个阶段: 第1阶段根据约束特征, 生成若干个较好的工件序列作为初始序列集合; 第2阶段采用插入机制进一步优化初始序列, 从中选取最好的调度解.

#### 3.1 确定初始序列集

无等待流水线车间调度问题是等待时间上限为零的一类  $Fm | w_{ij} \leq \alpha | C_{\max}$  问题, 且由定理2可得, 其可行解只存在于置换流水线车间调度问题中. 对于具体的  $Fm | nwt | C_{\max}$  问题及其相应的等待时间受限的置换流水线车间问题, 二者满足以下关系.

定理4 对于  $Fm | w_{ij} \leq \alpha, pmu | C_{\max}$  问题, 令  $C_{\max}^*$  和  $C_{\max}^0$  分别表示原问题和相应无等待问题的最优值, 则有

$$0 \leq C_{\max}^0 - C_{\max}^* \leq \alpha(n-1)(m-1) \quad (3)$$

证明 首先证明对于  $\forall i \in [1, n], \forall j \in [1, m]$  均满足

$$0 \leq C_{\pi(i)j}^0 - C_{\pi(i)j} \leq \alpha(m-1)(i-1) \quad (4)$$

1) 当  $i = 1$  时, 有  $\alpha(m-1)(i-1) = 0$ , 此时上下限均为 0, 因此需证明  $C_{\pi(i)j}^0 - C_{\pi(i)j} = 0$ .

对于工件序列  $\pi$ , 当  $1 \leq j \leq m$  时, 有  $C_{\pi(1)j} =$

$$C_{\pi(1)j}^0 = \sum_{k=1}^j P_{\pi(1)k} \text{ 满足式 (4).}$$

2) 假设  $i = r$  时, 式 (4) 成立, 即对于  $1 \leq j \leq m$  均满足

$$0 \leq C_{\pi(r)j}^0 - C_{\pi(r)j} \leq \alpha(m-1)(r-1)$$

下面将证明当  $i = r+1$  时, 有  $0 \leq C_{\pi(r+1)j}^0 - C_{\pi(r+1)j} \leq \alpha(m-1)$ .

无等待 (no-wait) 问题实质为  $\alpha = 0$  的等待时间受限调度问题, 由引理 1 可得

$$C_{\pi(r+1)j} = \max_{1 \leq k \leq j} \left\{ C_{\pi(r)k} + \sum_{t=k}^j P_{\pi(r+1)t} \right\}$$

$$C_{\pi(r)l} - \sum_{t=j+1}^{l-1} P_{\pi(r+1)t} - \alpha(l-j)$$

$$C_{\pi(r+1)j}^0 = \max_{1 \leq k \leq j} \left\{ C_{\pi(r)k}^0 + \sum_{t=k}^j P_{\pi(r+1)t} \right\}$$

$$C_{\pi(r)l}^0 - \sum_{t=j+1}^{l-1} P_{\pi(r+1)t}$$

① 因为  $C_{\pi(r)j}^0 \geq C_{\pi(r)j}$ , 因而有

$$\max_{1 \leq k \leq j} \left\{ C_{\pi(r)k}^0 + \sum_{t=k}^j P_{\pi(r+1)t} \right\} \geq$$

$$\max_{1 \leq k \leq j} \left\{ C_{\pi(r)k} + \sum_{t=k}^j P_{\pi(r+1)t} \right\};$$

$$\max_{j+1 \leq l \leq m} \left\{ C_{\pi(r)l}^0 - \sum_{t=j+1}^{l-1} P_{\pi(r+1)t} \right\} \geq$$

$$\max_{j+1 \leq l \leq m} \left\{ C_{\pi(r)l} - \sum_{t=j+1}^{l-1} P_{\pi(r+1)t} \right\} \geq$$

$$\max_{j+1 \leq l \leq m} \left\{ C_{\pi(r)l} - \sum_{t=j+1}^{l-1} P_{\pi(r+1)t} - \alpha(l-j) \right\}$$

即  $C_{\pi(r+1)j} \geq C_{\pi(r+1)j}$ ,  $C_{\pi(r+1)j}^0 - C_{\pi(r+1)j} \geq 0$ .

② 因为  $C_{\pi(r)j} + \alpha(m-1)(r-1) \geq C_{\pi(r)j}^0$ , 因而有

$$\max_{1 \leq k \leq j} \left\{ C_{\pi(r)k} + \alpha(m-1) + \sum_{t=k}^j P_{\pi(r+1)t} \right\} \geq$$

$$\max_{1 \leq k \leq j} \left\{ C_{\pi(r)k}^0 + \sum_{t=k}^j P_{\pi(r+1)t} \right\}$$

又因为  $m \geq l$  且  $j \geq 1$ , 因而有

$$\max_{j+1 \leq l \leq m} \left\{ C_{\pi(r)l} + \alpha(m-1) - \sum_{t=j+1}^{l-1} P_{\pi(r+1)t} - \alpha(l-j) \right\} =$$

$$\max_{j+1 \leq l \leq m} \left\{ C_{\pi(r)l} + \alpha(m-1)(r-1) -$$

$$\sum_{t=j+1}^{l-1} P_{\pi(r+1)t} + \alpha(m-1) - \alpha(l-j) \right\} \geq$$

$$\max_{j+1 \leq l \leq m} \left\{ C_{\pi(r)l}^0 - \sum_{t=j+1}^{l-1} P_{\pi(r+1)t} \right\}$$

因此  $C_{\pi(r+1)j} + \alpha(m-1) \geq C_{\pi(r+1)j}^0$ , 即  $C_{\pi(r+1)j}^0 - C_{\pi(r+1)j} \leq \alpha(m-1)$ .

综上可得  $0 \leq C_{\pi(r+1)j}^0 - C_{\pi(r+1)j} \leq \alpha(m-1)$ . 因此若  $i = r$  时式 (4) 成立, 则  $i = r+1$  时式 (4) 同样成立.

综合 1) 2) 可得, 对于  $1 \leq i \leq n$  且  $1 \leq j \leq m$  均满足  $0 \leq C_{\pi(i)j}^0 - C_{\pi(i)j} \leq \alpha(m-1)(i-1)$ .

因为  $C_{\max}(\pi) = C_{\pi(n)m}$  且  $C_{\max}^0(\pi) = C_{\pi(n)m}^0$ , 有  $0 \leq C_{\max}^0(\pi) - C_{\max}(\pi) \leq \alpha(n-1)(m-1)$ .

定义  $\pi_1^*$  为  $Fm | w_{ij} \leq \alpha, prmu | C_{\max}$  的最优工件序列, 则有  $C_{\max}^0 \leq C_{\max}^0(\pi_1^*)$ ,  $C_{\max} = C_{\max}(\pi_1^*)$ , 因而  $C_{\max}^0 - C_{\max} \leq C_{\max}^0(\pi_1^*) - C_{\max}(\pi_1^*) \leq \alpha(n-1)(m-1)$ .

定义  $\pi_2^*$  为  $Fm | nwt | C_{\max}$  的最优工件序列, 则有  $C_{\max}^0 = C_{\max}^0(\pi_2^*)$ ,  $C_{\max} \leq C_{\max}(\pi_2^*)$ , 因而  $C_{\max}^0 - C_{\max} \geq C_{\max}^0(\pi_2^*) - C_{\max}(\pi_2^*) \geq 0$ .

由此可得  $0 \leq C_{\max}^0 - C_{\max} \leq \alpha(n-1)(m-1)$ .

证毕.

定理 4 表明, 等待时间受限约束 ( $\alpha \geq 0$ ) 与无等待 (no-wait) 约束 ( $\alpha = 0$ ) 不仅在性质上相似, 且对置换流水车间调度问题的最优 makespan 值也相近. 这点在等待时间上限约束比较紧时尤为明显, 因此可以借鉴无等待流水车间调度问题的求解方法生成初始调度.

无等待两机流水车间调度问题的最优解可由 Gilmore 和 Gomory<sup>[13]</sup> 提出的 TSP 算法 (下称 Gilmore-Gomory 算法) 在  $O(n \lg n)$  时间内获得<sup>[11]</sup>.

因此, 可以将  $Fm | w_{ij} \leq \alpha, prmu | C_{\max}$  问题模拟为两台机器的生产环境进而生成工件序列, 具体方法如下

1) 增加虚拟机器  $M_0$  和  $M_{m+1}$  作为开始和结

束阶段 其中各工件在这两台虚拟机器上的加工时间  $p_{i0} = p_{i,m+1} = 0 (i = 1, \dots, n)$ ;

2) 对于  $\forall j \in [0, m]$  构建由  $n$  个工件组成的无等待两机流水车间调度问题  $F2 | nwt | C_{max}$ ,

$$P_{i1} = \sum_{k=0}^j p_{ik} P_{i2} = \sum_{k=j+1}^{m+1} p_{ik} (i = 1, \dots, n);$$

3) 对于转化后的  $m + 1$  个  $F2 | nwt | C_{max}$  问题, 分别采用 Gilmore-Gomory 算法求解, 获得  $m + 1$  个工件加工序列, 构成初始序列集合.

### 3.2 插入优化机制

初始序列集合中的调度解主要借鉴了无等待 (no-wait) 约束的特征, 在等待时间受限约束较紧时 makespan 接近最优值, 但当等待时间上限放宽时, 初始解的目标值与最优值的偏差就会增大, 这就需要引入优化机制进一步获得更好解.

在已有的置换流水车间调度启发式中, NEH 算法<sup>[14]</sup> 提供了一种优化效果显著的插入机制<sup>[9]</sup>, 现将其扩展到等待时间受限的置换流水车间调度问题并记为 Inserting 方法, 步骤如下

步骤 1 输入初始工件序列  $\pi_0$ , 将  $\pi_0$  的第一个工件放入调度工件集  $\pi$ ;

步骤 2 令  $L$  从 2 到  $n$ , 依次执行以下步骤

(1) 令  $k^* \leftarrow 0, C_{max}^* \leftarrow +\infty$ ;

(2) 令  $k$  从 1 到  $L$ , 将  $\pi_0(L)$  插入工件  $\pi(k-1)$  和  $\pi(k)$  之间, 调用 Timetabling 方法计算由这  $L$  个工件组成的部分工件序列的 makespan 值  $C_k^L$ ; 若  $C_k^L < C_{max}^*$ , 则  $k^* \leftarrow k, C_{max}^* \leftarrow C_k^L$ ;

(3) 更新调度工件集  $\pi$ , 将  $\pi_0(L)$  插入到  $\pi(k^* - 1)$  和  $\pi(k^*)$  之间;

步骤 3 输出  $\pi$  和  $C_{max}^*$ .

已知 Timetabling 的复杂度为  $O(nm)$ , 易见 Inserting 方法的复杂度为  $O(n^3m)$ . 由于等待时间受限的置换流水车间调度问题具有可逆性, 可以利用这一性质简化上述方法, 降低时间复杂度. 将简化后的新插入优化方法记为 Inserting-N, 步骤如下

步骤 1 初始化

(1) 输入初始工件序列  $\pi_0$ ;

(2) 构建调度问题的逆问题, 即令  $\tilde{p}_{ij} \leftarrow$

$$p_{i,m+1-j} (i = 1, \dots, n; j = 1, \dots, m);$$

步骤 2 取  $\pi_0$  的前两个工件, 分别计算这两个工件先后两种排序下的 makespan, 取其中具有较小 makespan 的排序放入调度工件集  $\pi$ ;

步骤 3 令  $L$  从 3 到  $n$ , 依次执行以下步骤

(1) 调用 Timetabling 方法, 计算  $\pi$  中所有工件在各台机器上的完工时间  $C_{\pi(i)j}$ ;

(2) 生成工件集  $\pi$  的逆序  $\tilde{\pi}$ , 调用 Timetabling 方法计算逆问题中, 序列  $\tilde{\pi}$  的各工件完工时间  $\tilde{C}_{\tilde{\pi}(i)j}$ ;

(3) 令  $k \leftarrow 1, k^* \leftarrow 0, C_{max}^* \leftarrow +\infty$ ;

(4) 计算将工件  $\pi_0(L)$  插入到位置  $k$  时, 该工件在各阶段的完工时间最小值  $ect_j$ ;

若  $k = 1$ , 则  $ect_1 \leftarrow p_{\pi_0(L)1}$ , 令  $j$  从 2 到  $m$  计算  $ect_j \leftarrow ect_{j-1} + p_{\pi_0(L)j}$ ;

否则  $ect_1 \leftarrow C_{\pi(k-1)1} + p_{\pi_0(L)1}$ , 令  $j$  从 2 到  $m$  计算  $ect_j \leftarrow \max\{C_{\pi(k-1)j}, ect_{j-1}\} + p_{\pi_0(L)j}$ , 令  $j$  从  $m-1$  到 1, 若有  $ect_{j+1} - ect_j > p_{\pi_0(L)j+1} + \alpha$ , 则令  $ect_j \leftarrow ect_{j+1} - p_{\pi_0(L)j+1} - \alpha$ ;

(5) 计算将工件  $\pi_0(L)$  插入到位置  $k$  时部分工件序列的 makespan 值  $C_k^L$ ;

若  $k < L$ , 则  $C_k^L \leftarrow \max_{1 \leq j \leq m} \{ect_j + \tilde{C}_{\tilde{\pi}(L+1-k), m+1-j}\}$ , 否则  $C_k^L \leftarrow ect_m$ ;

若  $C_k^L < C_{max}^*$ , 则  $k^* \leftarrow k, C_{max}^* \leftarrow C_k^L$ ;

(6)  $k \leftarrow k + 1$ , 若  $k \leq L$ , 转步骤 (4), 否则转步骤 3(7);

(7) 更新调度工件集  $\pi$ , 将  $\pi_0(L)$  插入到  $\pi(k^* - 1)$  和  $\pi(k^*)$  之间;

步骤 4 输出  $\pi$  和  $C_{max}^*$ .

在 Inserting-N 方法中,  $Fm | w_{ij} \leq \alpha, \mu | C_{max}$  问题的可逆性保证了在插入工件后计算部分序列 makespan 值的方法是正确和可行的. 就时间复杂度而言, Inserting-N 方法的步骤 3 由内外两层循环组成, 其中步骤 3(1) — (2) 分别计算由  $L$  个工件构成的调度问题及其逆问题的工件完工时间, 计算时间均为  $O(Lm)$ ; 步骤 3(3) — (6) 为内循环, 每循环一次执行步骤 3(4) 计算最早完工时间用时  $O(m)$ , 执行步骤 3(5) 计算部分工件序列的 makespan 值的复杂度  $O(m)$ , 内循环共计迭代  $L$  次, 因而计算时间为  $O(Lm)$ ; 步骤 3(7) 采用插入算法更新调度工件集, 复杂度为  $O(L)$ , 因此步

骤3(1)~(7)执行一次的计算时间为  $O(Lm)$ ;  $L$ 取值为从3到  $n$ , 因此步骤3的复杂度为  $O(n^2m)$ . 由于步骤2计算两工件排序的 makespan 的复杂度为  $O(m)$ , 所以 Inserting-N 方法的计算时间受制于步骤3, 为  $O(n^2m)$ , 与 Inserting 方法 ( $O(n^3m)$ ) 相比降低了时间复杂度, 提高了计算效率.

### 3.3 算法步骤

HLW 算法的具体步骤如下

#### 步骤1 初始化

(1) 令  $i$  从1到  $n$ , 赋值  $p_{i0} \leftarrow 0$ ,  $p_{i, m+1} \leftarrow 0$ ;

(2) 令  $j \leftarrow 0$ ,  $\pi^* \leftarrow \emptyset$ ,  $C_{\max}^* \leftarrow \infty$ ;

#### 步骤2 确定初始序列

(1) 令  $i$  从1到  $n$ , 计算  $P_{i1} \leftarrow \sum_{k=0}^j p_{ik}$ ,  $P_{i2} \leftarrow \sum_{k=j+1}^{m+1} p_{ik}$ ;

(2) 将  $P_{i1}$ ,  $P_{i2}$  视为工件  $J_i$  在两机流水车间环境下的加工时间, 应用 Gilmore-Gomory 算法生成初始工件序列  $\pi_j$ ;

#### 步骤3 插入优化机制

(1) 采用 Inserting-N 方法在序列  $\pi_j$  的基础上进一步优化, 得到新的工件序列  $\pi_j^*$  及对应的最大完工时间  $C_{\max}(\pi_j^*)$ ;

(2) 若  $C_{\max}(\pi_j^*) < C_{\max}^*$ , 则  $\pi^* \leftarrow \pi_j^*$ ,  $C_{\max}^* \leftarrow C_{\max}(\pi_j^*)$ ;

#### 步骤4 循环终止判断

令  $j \leftarrow j + 1$ ; 若  $j \leq m$ , 转步骤2, 否则转步骤5;

步骤5 算法结束, 输出  $\pi^*$  和  $C_{\max}^*$ .

HLW 算法在确定初始序列时, 首先模拟两机环境下的加工时间, 耗时  $O(nm)$ , 然后采用 Gilmore-Gomory 算法生成初始序列, 复杂度为  $O(n \lg n)$ ; 优化机制的 Inserting-N 方法的复杂度已知为  $O(n^2m)$ , 因此生成一个初始序列并实现插入优化的计算时间为  $O(n^2m)$ . HLW 算法共计循环  $m + 1$  次, 可知时间复杂度为  $O(n^2m^2)$ .

## 4 数据实验

### 4.1 实验环境与数据

本文采用 C# 语言在 PC Pentium4 / CPU3.0GHz / RAM1.0G 上编程实现上述启发式. 为进一步考察 HLW 算法的求解性能, 实验中将其与以下4种算法比较

1) NEH-G 算法. 在不考虑等待时间受限的情况下, 将问题转化为一般流水车间调度并采用 NEH 算法求解, 对得到的工件序列采用 Timetabling 方法计算最大完工时间.

2) NEH-W 算法. NEH-W 算法在优化过程中考虑等待时间受限约束, 应用 Inserting-N 方法生成最终调度解.

3) FN 算法. Framinan 和 Nagano<sup>[15]</sup> 针对无等待流水车间调度问题设计了一种启发式, 利用工件间的完工时间关系定义一类 TSP 问题并求解, 这里将该启发式记为 FN 算法.

4) FN-W 算法. 即对由 FN 算法得到的工件序列通过 Inserting-N 方法进一步优化, 从中选择最优的序列生成调度解.

考虑到影响算法的主要因素为问题规模、等待时间上限的松紧程度以及加工时间的分布, 将实验参数设置为: 工件数  $n = 20, 40, 60, 80$ ; 机器数  $m = 5, 10$ ; 等待时间上限  $\alpha = 1, 5, 10, 15, 20$ ; 加工时间  $p_{ij} \in \text{DU}[1, 99]$ , 其中  $\text{DU}[a, b]$  表示区间为  $[a, b]$  的离散均匀分布. 根据工件数、机器数和等待时间上限的不同取值, 将实验分成  $4 \times 2 \times 5 = 40$  组, 每组随机生成 50 个算例, 并采用以下指标衡量算法的求解质量

1) 与已知最好解的相对偏离率  $Dev(Hrt)$ , 计算公式如下

$$Dev(Hrt) = \frac{C_{\max}(Hrt) - C_{\max}^{\text{best}}}{C_{\max}^{\text{best}}} \times 100\% \quad (5)$$

其中  $C_{\max}(Hrt)$  表示由当前算法获得的工件序列的最大完工时间,  $C_{\max}^{\text{best}}$  表示上述5种启发式能够获得的最佳调度解的 makespan 值.

2) 获得最好解的成功概率  $pbt(Hrt)$ , 计算公式如下

$$pbt(Hrt) = \frac{\sum_{i=1}^N (1 - \min\{C_{\max}(Hrt) - C_{\max}^{\text{best}}, 1\})}{N} \quad (6)$$

其中  $N$  为每组生成的算例总数,  $pbt(Hrt)$  表示当前算法获得最好 makespan 值的次数占算例总数的比例.

### 4.2 实验结果与分析

实验结果见表1, 其中  $\overline{Dev}$  表示相对偏离率的平均值, CPU 表示 HLW 算法的计算时间.



表 1 实验结果  
Table 1 Experimental results

n	m	$\alpha$	CPU/s	NEH-G		NEH-W		FN		FN-W		HLW	
				$\overline{Dev}$	Pbt	$\overline{Dev}$	Pbt	$\overline{Dev}$	Pbt	$\overline{Dev}$	Pbt	$\overline{Dev}$	Pbt
20	5	1	0.005	9.95	0	2.51	0.12	15.36	0	1.99	0.08	0.07	0.92
		5	0.003	10.16	0	2.44	0.10	16.06	0	2.29	0.06	0.06	0.92
		10	0.003	9.71	0	2.70	0.08	16.02	0	2.51	0.12	0.12	0.86
		15	0.002	9.28	0	1.99	0.20	17.23	0	2.50	0.10	0.08	0.84
		20	0.002	8.31	0	2.12	0.10	17.96	0	1.89	0.26	0.12	0.80
	10	1	0.008	7.93	0	2.27	0.10	19.26	0	2.52	0.08	0.01	0.92
		5	0.009	7.30	0	2.50	0.06	19.59	0	2.69	0.02	0.02	0.98
		10	0.009	6.41	0	1.85	0.10	19.32	0	2.22	0.12	0.10	0.88
		15	0.009	6.36	0	2.01	0.14	21.07	0	2.27	0.06	0.06	0.94
		20	0.008	6.15	0	2.12	0.06	21.67	0	2.36	0.04	0.04	0.92
40	5	1	0.011	11.10	0	2.28	0.04	10.33	0	1.32	0.24	0.13	0.78
		5	0.011	11.75	0	1.96	0.08	10.71	0	1.43	0.14	0.13	0.82
		10	0.011	11.41	0	2.04	0.08	11.01	0	1.64	0.14	0.07	0.88
		15	0.010	11.25	0	2.14	0.06	11.69	0	1.39	0.14	0.14	0.86
		20	0.010	11.16	0	1.93	0.08	13.36	0	1.31	0.32	0.21	0.66
	10	1	0.034	8.91	0	2.38	0	14.70	0	1.57	0.06	0.03	0.96
		5	0.034	9.63	0	1.97	0.08	15.63	0	1.76	0.06	0.04	0.92
		10	0.034	8.69	0	1.92	0.06	16.56	0	1.74	0.02	0.02	0.96
		15	0.035	8.38	0	1.58	0.08	17.50	0	1.73	0.02	0.02	0.96
		20	0.033	8.32	0	1.57	0.06	17.61	0	1.74	0.10	0.06	0.88
60	5	1	0.023	12.09	0	1.65	0.06	7.93	0	1.21	0.10	0.06	0.86
		5	0.023	12.79	0	1.94	0.06	8.93	0	1.49	0.12	0.07	0.84
		10	0.023	12.77	0	1.98	0.06	9.17	0	1.03	0.24	0.16	0.70
		15	0.023	12.82	0	1.84	0.08	9.92	0	1.55	0.10	0.12	0.86
		20	0.022	12.02	0	1.65	0.04	10.92	0	1.18	0.16	0.08	0.80
	10	1	0.078	10.21	0	2.12	0	12.47	0	1.35	0.04	0.02	0.96
		5	0.076	10.16	0	1.82	0.02	12.56	0	1.45	0.08	0.05	0.90
		10	0.074	10.56	0	1.89	0	13.85	0	1.64	0.06	0.02	0.94
		15	0.075	9.74	0	1.79	0.04	15.06	0	1.39	0.12	0.09	0.84
		20	0.075	8.79	0	1.79	0	15.72	0	1.53	0.02	0.01	0.98
80	5	1	0.041	12.98	0	1.68	0.04	6.70	0	0.93	0.20	0.14	0.76
		5	0.039	12.71	0	1.94	0.02	7.27	0	0.89	0.22	0.11	0.76
		10	0.040	13.28	0	2.00	0	7.64	0	1.08	0.06	0.01	0.94
		15	0.040	12.83	0	1.67	0.04	8.31	0	0.98	0.20	0.08	0.76
		20	0.039	12.92	0	1.48	0.06	9.66	0	1.12	0.18	0.11	0.78
	10	1	0.138	10.76	0	1.85	0.02	11.18	0	1.43	0.08	0.02	0.94
		5	0.137	11.25	0	2.01	0.06	11.72	0	1.24	0.04	0.03	0.90
		10	0.137	10.46	0	2.10	0	13.08	0	1.54	0.02	0.02	0.98
		15	0.134	10.44	0	1.86	0	13.54	0	1.31	0.02	0.01	0.98
		20	0.133	10.52	0	1.87	0.02	13.73	0	1.47	0.02	0.03	0.96

由表1可以得到以下结论

1) HLW算法的相对偏离率明显优于其他算法,且获得相对最好解的成功概率均值达到了0.88.另外4个算法的优化效果从好到坏依次为:FN-W、NEH-W、NEH-G、FN.其中,FN-W和NEH-W的求解质量相近,远优于另两种算法.

2)在大多数情况下,相对最好解由HLW算法或FN-W算法获得,这说明借鉴无等待流水车间调度问题的求解方法生成初始序列,并利用考虑等待时间受限约束特征的插入机制进一步优化的解决思路是正确且有效的.

3)算法NEH-G和NEH-W的初始解构造方法相同,仅优化策略不同,而求解质量却相差甚远.此外,FN-W算法在FN启发式的基础上利用Inserting-N方法进行优化,使得相对偏离率平均降低了11.93%,最好时对偏离率的改进达到了30.6%.这均说明本文提出的插入优化机制具有很好的优化效果.

4)当等待时间上限和工件数增大时,HLW算法的两个指标均未有明显的变化趋势.这是因为实验指标与相对最好解作比对,而最好解的质量受制于这些因素的影响程度,指标值变化不明显说明因素波动对HLW算法的影响与其对最好解的影响程度相近.当机器数增加时,HLW的相对偏离率下降且成功概率增加,说明机器的增减

对HLW的影响相对较小.因而,HLW算法的求解质量比较稳定,问题规模和等待时间上限的波动对其影响较小.

5)HLW算法具有很好的计算效率,即使对于80个工件、10台机器的调度问题,也能在0.14s内给出调度解,完全能够满足调度的需要.

## 5 结束语

等待时间具有最大值限制是中间产品性质不稳定或具有高温连续作业特征的流程工业生产中常见的一类约束.本文针对等待时间受限的置换流水车间调度问题,分析了其可行解与流水车间调度最优解的关系,讨论了工件完工时间的有向图表示和数学表达式,证明了置换流水车间调度问题的可逆性,并分析了等待时间受限约束与无等待(no-wait)约束的关联特征.在此基础上,根据问题特征提出了启发式算法,首先将问题模拟为两台机器的情况,应用Gilmore-Gomory算法求解,获得初始序列集;然后对于初始序列集中的元素采用插入机制进一步优化,并利用问题的可逆性,将插入优化机制的复杂度由 $O(n^3m)$ 降至 $O(n^2m)$ .数据实验表明,本文提出的算法能够有效求解等待时间受限的置换流水车间调度问题.

## 参考文献:

- [1]Yang D L, Chern M S. A two-machine flowshop sequencing problem with limited waiting time constraints [J]. Computers and Industrial Engineering, 1995, 28(1): 63-70.
- [2]Su L H. A hybrid two-stage flowshop with limited waiting time constraints [J]. Computers and Industrial Engineering, 2003, 44: 409-424.
- [3]Fondrevelle J, Oulamar A, Portmann M C. Permutation flowshop scheduling problems with maximal and minimal time lags [J]. Computers & Operations Research, 2006, 33: 1540-1556.
- [4]Chen J S, Yang J S. Model formulations for the machine scheduling problem with limited waiting time constraints [J]. Journal of Information and Optimization Sciences, 2006, 27: 225-240.
- [5]李铁克,尹兆涛. 等待时间受限的流水车间调度问题的启发式算法 [J]. 管理学报, 2009, 6(10): 1335-1339.  
Li Tiek, Yin Zhaotao. A heuristic algorithm for flowshop scheduling with limited waiting times [J]. Chinese Journal of Management, 2009, 6(10): 1335-1339. (in Chinese)
- [6]尹兆涛,李铁克,肖拥军. 等待时间受限 Flowshop 调度的 HGA 算法 [J]. 计算机工程, 2009, 35(21): 4-6.  
Yin Zhaotao, Li Tiek, Xiao Yongjun. HGA algorithm for flowshop scheduling with limited waiting time [J]. Computer Engi-

- neering, 2009, 35(21): 4–6. (in Chinese)
- [7] 尹兆涛, 李铁克. 考虑交货期和等待时间受限的 HFS 调度问题的混合算法 [J]. 工业工程, 2009, 12(1): 79–83.  
Yin Zhaotao, Li Tiek. Hybrid algorithms for hybrid flowshop scheduling considering due dates and limited waiting times [J].  
Industrial Engineering Journal, 2009, 12(1): 79–83. (in Chinese)
- [8] Reza H S, Saghafianb S. Flow shop-scheduling problems with makespan criterion: A review [J]. International Journal of  
Production Research, 2005, 43(14): 2895–2929.
- [9] Framinan J M, Gupta J N D, Leisten R. A review and classification of heuristics for permutation flow-shop scheduling with  
makespan objective [J]. Journal of the Operational Research Society, 2004, 55: 1243–1255.
- [10] Graham R L, Lawler E L, Lenstra J K, et al. Optimization and approximation in deterministic sequencing and scheduling:  
A survey [J]. Annals of Discrete Mathematics, 1979, 5: 287–326.
- [11] Bagchi T P, Jatinder N D G, Sriskandarajah C. A review of TSP based approaches for flowshop scheduling [J]. European  
Journal of Operational Research, 2006, 169: 816–854.
- [12] 冯大光, 唐立新. 具有最大总加权满意度的单机调度问题的 dynasearch 算法 [J]. 管理科学学报, 2006, 9(4): 40  
– 50.  
Feng Daguang, Tang Lixin. Dynasearch algorithms for single machine scheduling problem with total weighted satisfaction  
[J]. Journal of Management Sciences in China, 2006, 9(4): 40–50. (in Chinese)
- [13] Gilmore P C, Gomory R E. Sequencing a one-state variable machine: A solvable case of the traveling salesman problem  
[J]. Operations Research, 1964, 12(5): 655–679.
- [14] Nawaz M, Enscore E E, Ham I. A heuristic algorithm for the m-machine, n-job flow-shop sequencing problem [J]. Ome-  
ga, 1983, 11: 91–95.
- [15] Framinan J M, Nagano M S. Evaluating the performance for makespan minimisation in no-wait flowshop sequencing [J].  
Journal of Materials Processing Technology, 2008, 197(1–3): 1–9.

## Heuristic algorithm for permutation flowshop scheduling problem with limited waiting time constraints

WANG Bai-lin<sup>1, 2</sup>, LI Tie-ke<sup>1, 2</sup>

1. Dongling School of Economics and Management, University of Science and Technology Beijing, Beijing 100083, China;
2. Engineering Research Center of MES Technology for Iron & Steel Production, Ministry of Education, Beijing 100083, China

**Abstract:** The permutation flowshop scheduling problem with limited waiting time constraints is studied. The relationship between permutation schedules and optimal solutions of the corresponding flowshop scheduling are analyzed. With the discussion of the directed graph for the makespan computation of a permutation schedule, the reversibility of permutation flowshop scheduling with limited waiting time constraints is proved. Based on these properties, a heuristic algorithm is proposed. In the algorithm, an initial set of job permutations are obtained based on the similar characteristics to the no-wait constraints. Then, an inserting optimization mechanism, which can be done in  $O(n^2m)$  time by the reversibility of the problem, is introduced to improve the initial schedules. Numerical results demonstrate the feasibility and effectiveness of the algorithm.

**Key words:** scheduling; permutation flowshop; limited waiting times; heuristic; reversibility