文章编号: 1001-0920 (2012) 05-0768-05

基于 TSP 方法求解等待时间受限的置换流水车间调度

王柏琳, 李铁克, 孙 彬

(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: 等待时间受限的置换流水车间调度问题要求工件在连续两个机器间的等待时间满足上限值约束. 对此, 分析了工件序列中相邻工件的加工持续时间及其上下界关系, 并且提出一种启发式方法. 首先, 建立旅行商问题 (TSP) 以生成初始调度; 然后, 采用扩展插入方法优化调度解. 为了衡量算法性能, 给出问题下界的计算方法和相关评价指标, 并通过数据实验验证了该启发式和下界计算方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

中图分类号: TP301 **文献标识码:** A

TSP-based heuristic algorithm for permutation flowshop scheduling with limited waiting time constraints

WANG Bai-lin, *LI Tie-ke*, *SUN Bin*

(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. Correspondent：WANG Bai-lin，E-mail：wangbailin_83@sina.com)

Abstract: In the permutation flowshop scheduling problem with limited waiting time constraints, the waiting time of each job between two consecutive machines is restricted by some upper bound. Based on the deep discussion of the duration time between two adjacent jobs in one job permutation and the relations of its upper and lower bound, a heuristic algorithm is proposed. In the algorithm, a travelling salesman problem(TSP) concerning the duration times is built to obtain an initial schedule, and then an extended inserting method is presented for the further optimization. To measure algorithm performance, an approach to calculate a lower bound of the problem is proposed. Numerical results show effectiveness and feasibility of this heuristic algorithm.

Key words: scheduling；permutation flowshop；limited waiting time；heuristic

1 引 言

流水车间调度问题在工业中具有广泛的应用背景, 一直是生产调度领域中的研究热点之一[1]. 然而该问题在钢铁生产、玻璃制造等高温连续生产或中间产品性质不稳定的车间生产过程中, 往往存在等待时间受限的特殊约束, 要求工件在相邻阶段间的等待时间不超过一定的上限值. 目前, 关于这类流水车间调度问题的研究成果不多. 对于问题复杂性, 文献 [2] 证明了两机情况下问题具有 NP 难特性, [3] 证明了等待时间同时具有上下限约束的置换流水车间调度是强 NP 难的; 在算法求解方面, [2-3] 采用分支定界法求解; [4] 针对第 1 阶段存在批处理过程的两机流水车间环境, 建立了混合整数规划模型, 并提出启发式方法求解; [5] 对在不同车间环境下, 等待时间受限的调度问题的建模机制进行了研究; [6] 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质, 并提出一种启发式算法; [7] 提出最小化加权设备完工时间的优化目标, 并对该目标下具有等待时间上下限约束的置换流水车间调度问题设计了分支定界算法; [8] 采用分支定界法求解一类等待时间严格等于定值的置换流水车间调度问题; [9] 针对等待时间受限的流水车间问题，提出嵌入约束满足

收稿日期: 2010-11-07；**修回日期:** 2011-01-23.

基金项目: 教育部博士学科点专项科研基金项目 (20100006110006)；国家自然科学基金项目 (70771008)；中央高校基本科研业务费专项资金项目 (FRF-AS-09-007B).

作者简介: 王柏琳 (1983−), 女, 讲师, 博士, 从事生产调度、智能算法等研究；李铁克 (1958−), 男, 教授, 博士生导师, 从事先进制造管理、生产计划与调度等研究.

和变邻域搜索的混合遗传算法; [10] 提出一种结合交换邻域、变邻域和禁忌搜索的动态变邻域搜索方法; [11] 将问题扩展到混合流水车间调度, 考虑了交货期要求, 并提出回溯、启发式修复与邻域搜索相结合的混合算法.

置换流水车间调度是流水车间调度领域中的一类重要问题, 要求所有机器上的工件加工顺序均相同[12-13]. 本文针对等待时间受限的置换流水车间调度, 探讨相邻工件的完工时间关系, 在此基础上设计启发式算法, 并通过数据实验考察算法的求解性能.

2　问题模型

等待时间受限的置换流水车间调度问题可描述为: n 个工件 $\{J_1, J_2, \cdots, J_n\}$ 以相同次序在 m 台机器 $\{M_1, M_2, \cdots, M_m\}$ 上加工, 且所有机器上各工件的加工顺序均相同, 工件 $J_i(i \in I = \{1, 2, \cdots, n\})$ 在机器 $M_j(j \in J = \{1, 2, \cdots, m\})$ 上的加工时间已知, 记为 p_{ij}. 问题要求确定工件加工顺序 π 和完工时间 C_{ij}, 使其满足以下约束并优化某个目标函数:

1) 工件依序在机器上加工, 且在离开前一台机器后才能在进入下一台机器, 即

$$C_{i,j+1} - C_{ij} \geqslant p_{i,j+1},\ i \in I,\ j \in J\backslash\{m\}; \tag{1}$$

2) 一台机器在某一时刻只能加工一个工件, 令 $\pi(i)$ 表示工件序列 π 中第 i 个加工工件, 即

$$C_{\pi(k),j} - C_{\pi(i),j} - p_{\pi(k),j} \geqslant 0,\ i < k,\ i, k \in I,\ j \in J; \tag{2}$$

3) 工件 J_i 在相邻机器 M_j 和 M_{j+1} 间的等待时间 w_{ij} 受上限值 α 约束, 即

$$w_{ij} = C_{i,j+1} - C_{ij} - p_{i,j+1} \leqslant \alpha,\ i \in I,\ j \in J\backslash\{m\}. \tag{3}$$

调度目标为最小化最大完工时间 (Makespan)

$$C_{\max} = \max\{C_{ij}\}. \tag{4}$$

采用 Graham[14]提出的三元组表示法, 此问题可表示为: $Fm\,|w_{ij} \leqslant \alpha, \mathrm{prmu}\,|C_{\max}$.

3　启发式算法

对于 $Fm\,|\mathrm{prmu}, w_{ij} \leqslant \alpha\,|C_{\max}$, 主要解决以下两个问题: 1) 确定工件在机器上的加工顺序, 即工件序列; 2) 对于给定的工件序列, 确定工件在各台机器上的完工时间. 对于问题 2), 令 π 表示给定的工件序列, 文献 [3] 给出了复杂度为 $O(nm)$ 的完工时间计算方法, 记为 Timetabling 方法, 具体步骤如下:

Step 1: $C_{\pi(1)1} \leftarrow p_{\pi(1)1}$, 令 $j = 2, 3, \cdots, m$, 分别计算 $C_{\pi(1),j} \leftarrow C_{\pi(1),j-1} + p_{\pi(1)j}$;

Step 2: 令 $i = 2, 3, \cdots, n$, 分别执行以下步骤:

Step 2.1: $C_{\pi(i),1} \leftarrow C_{\pi(i-1),1} + p_{\pi(i)1}$;

Step 2.2: 令 $j = 2, 3, \cdots, m$, 分别计算

$$C_{\pi(i),j} \leftarrow \max\left\{C_{\pi(i-1),j}, C_{\pi(i),j-1}\right\} + p_{\pi(i),j};$$

Step 2.3: 令 $j = m-1, m-2, \cdots, 1$, 若 $C_{\pi(i),j+1} - C_{\pi(i),j} > p_{\pi(i),j+1} + \alpha$, 则令 $C_{\pi(i),j} \leftarrow C_{\pi(i),j+1} - p_{\pi(i),j+1} - \alpha$;

Step 3: $C_{\max}(\pi) \leftarrow C_{\pi(n),m}$, 输出 $C_{\max}(\pi)$.

由以上步骤可以看出, 算法设计的关键与难点在于问题 1), 即如何生成好的工件序列. 针对该问题特征, 本文提出一种基于 TSP 的启发式算法 (THA), 并采用扩展插入方法进一步优化调度解.

3.1　基于 TSP 的初始调度

由 Timetabling 方法可知, 当仅有两个工件时, 调度解的 Makespan 值与相同工件序列下不考虑等待时间受限约束的相应一般置换流水车间调度问题的 Makespan 值相同. 而当工件数多于两个时, 任意工件序列 π 的相邻工件 J_i 和 J_k 之间的关系如图 1 所示.

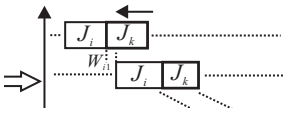

图 1　相邻工件 J_i 和 J_k 的甘特图

图 1(a) 为序列 π 中, 相邻工件 J_i 和 J_k 在甘特图中的位置关系. 仅就这两个工件在车间生产中的加工持续时间 (即从 J_i 进入 M_1 到 J_k 离开 M_m 的时间长度 $\mathrm{DT}_{ik} = C_{km} - S_{i1}$, 其中 S_{i1} 表示 J_i 在 M_1 上的开工时间) 而言, 图 1(a) 和图 1(b) 是等价的. 换言之, 由图 1(a) 所得的 $\mathrm{DT}_{ik}^{(1)}$ 等于图 1(b) 所得的 $\mathrm{DT}_{ik}^{(2)}$. 因此, 可由图 1(b) 得到 DT_{ik} 的计算公式如下:

对于工件 J_i 有

$$C_{ij} - S_{i1} = \sum_{t=1}^{j-1}(p_{it} + w_{it}) + p_{ij},\ j \in J;$$

对于工件 J_k 有

$$C_{k1} = C_{i1} + p_{k1},$$

$$C_{kj} = \max\{C_{ij}, C_{k,j-1}\} + p_{kj},\ j \in J\backslash\{1\}.$$

因此, 对于 $\mathrm{DT}_{ik} = C_{km} - S_{i1}$, 有

$$\mathrm{DT}_{ik} = C_{km} - S_{i1} =$$

$$\max\{C_{im} - S_{i1}, C_{k,m-1} - S_{i1}\} + p_{km} = \cdots =$$

$$\max\left\{\sum_{t=1}^{m} p_{it} + \sum_{t=1}^{m-1} w_{it} + p_{km},\right.$$

$$\left.\sum_{t=1}^{m-1} p_{it} + \sum_{t=1}^{m-2} w_{it} + \sum_{t=m-1}^{m} p_{kt}, \cdots\right\} =$$

$$\max_{1\leqslant j\leqslant m}\Big\{\sum_{t=1}^{j}p_{it}+\sum_{t=1}^{j-1}w_{it}+\sum_{t=j}^{m}p_{kt}\Big\}.$$

已知对于$1\leqslant j<m$, 有$0\leqslant w_{ij}\leqslant\alpha$, 因此$\mathrm{DT}_{ik}$的下界$\mathrm{DT}_{ik}^{L}$和上界$\mathrm{DT}_{ik}^{U}$为

$$\mathrm{DT}_{ik}^{L}=\max_{1\leqslant j\leqslant m}\Big\{\sum_{t=1}^{j}p_{it}+\sum_{t=j}^{m}p_{kt}\Big\},\tag{5}$$

$$\mathrm{DT}_{ik}^{U}=\max_{1\leqslant j\leqslant m}\Big\{\sum_{t=1}^{j}p_{it}+\sum_{t=j}^{m}p_{kt}+\alpha(j-1)\Big\}.\tag{6}$$

由此可知, DT_{ik}^{U}和DT_{ik}^{L}满足

$$0\leqslant\mathrm{DT}_{ik}^{U}-\mathrm{DT}_{ik}^{L}\leqslant\alpha\,(m-1)\,.\tag{7}$$

基于上述分析, 对于一个具体的$Fm|\mathrm{prmu},w_{ij}\leqslant\alpha\,|C_{\max}$, 均可构建如下TSP问题: 问题由$n+1$个节点$\{J_0,J_1,\cdots,J_n\}$组成, 其中: J_0为虚拟工件, 在各台机器上的加工时间为零, 等待时间上限为零, 且对应TSP问题的起始节点. 节点J_i到J_k的距离

$$d_{ik}=r_{ik}\times\mathrm{DT}_{ik}^{L}+(1-r_{ik})\times\mathrm{DT}_{ik}^{U}.\tag{8}$$

其中: r_{ik}为权重系数, 且$r_{ik}\in[0,1]$, 其计算公式如下:

$$r_{ik}=\begin{cases}\dfrac{\mathrm{DT}_{ik}^{U}-\mathrm{DT}_{ik}^{L}}{\alpha\,(m-1)},\ (i,k\in I)\bigwedge(i\neq0)\,;\\ 0.5,\ \text{otherwise}.\end{cases}\tag{9}$$

通过求解上述TSP问题可获得调度问题$Fm\,|\mathrm{prmu},w_{ij}\leqslant\alpha\,|C_{\max}$的工件加工序列. 这里采用TSP问题的一类经典启发式方法—— 最近插入法(NI)[15]来求得近优解, 进而转化为调度问题的初始解.

3.2 扩展插入方法

在已有的置换流水车间调度启发式中, NEH算法是一类最好的排序规则, 它提供了一种优化效果显著的插入方法[13]. 为了进一步减小初始解的目标值与最优值的偏差, 本文将其扩展到等待时间受限的置换流水车间调度问题, 在初始调度的基础上进行优化, 获得更好解, 具体步骤如下:

Step 1: 输入初始调度π_0, 令$\pi\leftarrow\{\pi_0\,(1)\}$.

Step 2: 按以下步骤优化工件序列:

Step 2.1: $L\leftarrow2$.

Step 2.2: $k\leftarrow1$, $k^*\leftarrow0$, $C_{\max}^{*}\leftarrow+\infty$.

Step 2.3: $\pi_k^L\leftarrow\pi\bigcup\{\pi_0\,(L)\}$, 且$\pi_k^L\,(k)\leftarrow\pi_0\,(L)$. 对于$0\leqslant i<k$, 有$\pi_k^L\,(i)\leftarrow\pi\,(i)$; 对于$k\leqslant i'<L$, 有$\pi_k^L\,(i'+1)\leftarrow\pi\,(i')$. 调用Timetabling方法计算部分工件序列π_k^L的Makespan值C_k^L.

Step 2.4: 若$C_k^L<C_{\max}^{*}$, 则$k^*\leftarrow k$, $C_{\max}^{*}\leftarrow C_{k^*}^{L}$.

Step 2.5: $k\leftarrow k+1$, 若$k\leqslant L$, 转Step 2.3; 否则转Step 2.6.

Step 2.6: 更新调度工件集$\pi\leftarrow\pi\bigcup\{\pi_0\,(L)\}$, 其中工件$\pi_0\,(L)$排在工件$\pi\,(k^*-1)$之后.

Step 2.7: $L\leftarrow L+1$, 若$L\leqslant n$, 转Step 2.2; 否则, 转Step 3.

Step 3: 输出π和$C_{\max}^{*}$.

已知Timetabling方法的复杂度为$O\,(nm)$, 因此对于每一个L, 计算部分工件序列的Makespan用时$O\,(Lm)$. 工件$\pi_0\,(L)$要插入到当前序列的L个位置, 计算时间为$O\left(L^2m\right)$, 其中$L=2,3,\cdots,n$, 因此插入优化方法的复杂度为$O\left(n^3m\right)$.

3.3 算法步骤

THA算法的具体步骤如下:

Step 1: 构建TSP问题.

Step 1.1: 引入虚拟工件J_0, 加工时间$p_{0j}=0(j\in J)$.

Step 1.2: 令$i=0,1,\cdots,n$, $k=0,1,\cdots,n$, 若$i=k$, 则$d_{ik}\leftarrow\infty$; 否则, 执行Step 1.4∼Step 1.6.

Step 1.3: 建立以$n+1$个工件为节点, 节点J_i到J_k的距离为d_{ik}的TSP问题, 转Step 2.

Step 1.4: 计算DT_{ik}的下界DT_{ik}^{L}, 具体过程如下:

1) $C_{i1}^{L}\leftarrow p_{i1}$;

2) 令$j=2,3,\cdots,m$, 计算$C_{ij}^{L}\leftarrow C_{i,j-1}^{L}+p_{ij}$;

3) $C_{k1}^{L}\leftarrow C_{i1}^{L}+p_{k1}$;

4) 令$j=2,3,\cdots,m$, 计算$C_{kj}^{L}\leftarrow\max\{C_{k,j-1}^{L},C_{ij}^{L}\}+p_{kj}$;

5) $\mathrm{DT}_{ik}^{L}\leftarrow C_{km}^{L}$.

Step 1.5: 计算DT_{ik}的上界DT_{ik}^{U}, 具体过程如下:

1) $C_{i1}^{U}\leftarrow p_{i1}$;

2) 令$j=2,3,\cdots,m$, 计算$C_{ij}^{U}\leftarrow C_{i,j-1}^{U}+p_{ij}+\alpha$;

3) $C_{k1}^{U}\leftarrow C_{i1}^{U}+p_{k1}$;

4) 令$j=2,3,\cdots,m$, 计算$C_{kj}^{U}\leftarrow\max\{C_{k,j-1}^{U},C_{ij}^{U}\}+p_{kj}$;

5) $\mathrm{DT}_{ik}^{U}\leftarrow C_{km}^{U}$.

Step 1.6: 根据式(8)和(9), 计算节点J_i到J_k的距离d_{ik}.

Step 2: TSP问题求解.

Step 2.1: 采用NI启发式求解, 得到节点排序π_0.

Step 2.2: 确定J_0在π_0中的位置i, 令$k=i+1,\cdots,n$, $\pi\leftarrow\pi_0\,(k)$; 令$k=1,2,\cdots,i-1$, $\pi\leftarrow\pi_0\,(k)$; 集合π即为初始工件序列.

Step 3: 优化初始解.

Step 3.1: 采用扩展插入方法在序列π的基础上进一步优化, 得到新的工件序列π^*以及对应的最大完工时间$C_{\max}^{*}$.

Step 3.2: 算法结束, 输出π^*和$C_{\max}^{*}$.

3.4 算法复杂度

THA 算法在构建 TSP 问题过程中, 计算 DT_{ik} 上下界的时间均为 $O(m)$, 故计算任意两点间距离用时 $O(m)$, 共计算 n^2 次, 因此 Step 1 的复杂度为 $O(n^2m)$; Step 2 中, NI 算法的复杂度为 $O\left(n^2\right)$, 确定工件序列需 $O(n)$, 因此 Step 2 用时 $O\left(n^2\right)$; 通过第 3.2 节的分析可知, Step 3 的复杂度为 $O\left(n^3m\right)$. 因此, 算法的计算时间受制于 Step 3, 整体复杂度为 $O\left(n^3m\right)$.

4 数据实验

4.1 实验设计

为了考察 THA 算法的求解性能, 以及初始调度生成方法和插入优化方法的有效性, 本文将 THA 算法与以下 4 种启发式算法进行比较:

1) NEH-0 算法: 仅执行 NEH 算法中生成初始序列的过程, 也称为总加工时间最大 (LTPT) 优先规则;

2) THA-0 算法: 根据 THA 算法的思想构建 TSP 问题并采用 NI 求解, 不采用扩展插入方法优化调度解;

3) NEH-G 算法: 针对一般置换流水车间调度问题采用 NEH 算法生成工件序列, 调用 Timetabling 方法计算最终序列的 Makespan 值;

4) NEH-W 算法: 采用扩展插入方法对由 LTPT 规则获得的初始序列进一步优化, 得到最终调度解.

由此可以看出, NEH-0 算法和 THA-0 算法均是依据一定的排序规则获得调度解, 而 NEH-G 算法, NEH-W 算法和 THA 算法则进一步采用不同的优化方法更新调度解. 本文采用 C# 语言在 PC Pentium 4/CPU 3.0 GHz/RAM 1.0 G 上编程实现上述 5 种启发式算法, 实验参数设置为: $\alpha = 1, 5, 10, 15, 20$; $n = 20, 50, 100$; $m = 3, 5$; $p_{ij} \in \mathrm{DU}\,[1, 50]$. 根据等待时间上限和问题规模的不同取值, 分为 30 组实验, 每组随机生成 50 个算例, 共计 1 500 个算例.

算法性能重点考察其计算效率和求解质量. 其中: 计算效率通过算法运行的 CPU 时间来衡量; 求解质量则是通过启发式 H 的下界偏差率 $\mathrm{Dev}\,(H)$ 来衡量, 计算公式为

$$\mathrm{Dev}\,(H) = \frac{C_{\max}\,(H) - \mathrm{Lower}}{\mathrm{Lower}} \times 100\%. \qquad (10)$$

现对 $Fm\,|\mathrm{prmu}, w_{ij} \leqslant \alpha\,|C_{\max}$ 的下界 (Lower) 分析如下: 由 Timetabling 方法可以看出, $Fm|\mathrm{prmu}, w_{ij} \leqslant \alpha\,|C_{\max}$ 的 Makespan 值必然不小于 $Fm\,|\mathrm{prmu}\,|C_{\max}$ 的最优值, 而对于 $F2\,|\mathrm{prmu}\,|C_{\max}$, 采用 Johnson 规则可获得最优解[13]. 因此, 可采用以下方法求得 $Fm|\mathrm{prmu}, w_{ij} \leqslant \alpha\,|C_{\max}$ 的下界:

Step 1: 令 $\mathrm{Lower} \leftarrow 0$.

Step 2: 令 $k = 1, 2, \cdots, m-1$, 执行以下步骤:

Step 2.1: 若 $k > 1$, 则计算

$$C_1 \leftarrow \min_{1 \leqslant i \leqslant n} \left\{ \sum_{h=1}^{k-1} p_{ih} \right\};$$

否则 $C_1 \leftarrow 0$.

Step 2.2: 对于在机器 M_k 和 M_{k+1} 上加工的工件, 将其视为两机流水车间, 并按 Johnson 规则排序. 不考虑等待时间受限约束, 计算该两机流水车间调度问题的 Makespan 值, 记为 C_2.

Step 2.3: 若 $k < m-1$, 则计算

$$C_3 \leftarrow \min_{1 \leqslant i \leqslant n} \left\{ \sum_{h=k+2}^{m} p_{ih} \right\};$$

否则 $C_3 = 0$.

Step 2.4: $L_k \leftarrow C_1 + C_2 + C_3$, 若 $L_k > \mathrm{Lower}$, 则 $\mathrm{Lower} \leftarrow L_k$.

Step 3: 输出 Lower.

4.2 实验结果

实验结果如表 1 所示. 其中: $\overline{\mathrm{Dev}}$ 表示下界偏离率的平均值, CPU 表示 THA 算法的计算时间.

表 1　实验结果

$n \times m$	α	CPU/s	$\overline{\mathrm{Dev}}$				
			NEH-0	THA-0	NEH-G	NEH-W	THA
20 × 3	1	0.001	29.18	16.51	20.35	11.33	**10.53**
20 × 3	5	0.001	26.36	13.47	16.26	7.52	**6.88**
20 × 3	10	0.001	23.01	12.77	13.10	5.44	**4.73**
20 × 3	15	0.001	21.05	13.54	10.80	3.54	**3.45**
20 × 3	20	0.002	20.93	13.19	7.96	**2.40**	2.44
20 × 5	1	0.002	40.03	28.26	28.21	18.80	**18.07**
20 × 5	5	0.003	36.75	26.76	23.53	15.40	**14.75**
20 × 5	10	0.001	32.52	23.90	17.20	10.44	**10.41**
20 × 5	15	0.002	30.18	23.36	14.79	8.59	**7.73**
20 × 5	20	0.002	29.27	20.94	11.81	**6.20**	6.52
50 × 3	1	0.018	28.28	12.95	22.26	9.87	**8.85**
50 × 3	5	0.018	25.75	10.28	18.65	6.80	**5.64**
50 × 3	10	0.016	22.55	9.82	15.50	4.53	**3.77**
50 × 3	15	0.017	20.49	8.12	13.02	2.53	**2.29**
50 × 5	1	0.031	38.94	25.35	32.34	19.88	**18.71**
50 × 5	5	0.031	34.09	21.25	24.99	14.57	**13.60**
50 × 5	10	0.034	30.99	20.94	21.73	10.65	**10.41**
50 × 5	15	0.034	28.27	18.54	18.11	8.24	**7.69**
50 × 5	20	0.032	26.20	16.91	14.36	5.17	**5.05**
100 × 3	1	0.150	29.76	11.82	23.64	10.07	**8.85**
100 × 3	5	0.151	25.97	8.62	20.24	6.47	**5.37**
100 × 3	10	0.146	22.51	6.61	17.15	3.31	**2.48**
100 × 3	15	0.128	20.89	6.45	14.20	2.25	**1.68**
100 × 3	20	0.121	19.14	5.58	12.14	0.89	**0.72**
100 × 5	1	0.260	39.69	23.59	33.78	19.92	**18.81**
100 × 5	5	0.249	35.76	20.38	29.06	16.02	**14.77**
100 × 5	10	0.240	30.84	17.64	23.58	10.31	**9.72**
100 × 5	15	0.240	28.90	17.33	20.62	8.10	**7.76**
100 × 5	20	0.250	25.48	14.45	17.19	5.09	**4.84**

由表 1 可得以下结论:

1) 5 种算法中下界偏离率 $\overline{\mathrm{Dev}}$ 的最好值在表 1 中加粗表示, 可以看出, THA 算法在多数情况下优于另外 4 个算法.

2) 对于未对调度解进行优化的 NEH-0 算法和 THA-0 算法, THA-0 算法的求解质量明显优于 NEH-0 算法; 尤其当工件数 $n \geqslant 50$ 时, THA-0 算法的下界偏差率显著低于经调度优化后的 NEH-G 算法. 这说明, 本文提出的初始调度生成方法是有效的.

3) NEH-G 算法在优化过程中未考虑等待时间受限约束, 对 NEH-0 算法的平均改进率为 9.11%; NEH-W 算法采用扩展插入方法, 对 NEH-0 的平均改进率达到 19.55%, 这说明本文给出的优化方法具有很好的效果.

4) 问题规模 $n \times m$ 相同的情况下, 5 种算法的下界偏差率均随着等待时间上限 α 的增大而下降, 这表明 α 的增加会对越来越多的调度解失去约束力, 因此算法的优化效果显著.

5) 实验中共有 25 个算例的相对最优解与下界的偏离率为 0, 这表明由本文给出下界计算方法获得的问题下界是紧的.

6) THA 算法具有很好的计算效率, 即使对于 100 个工件, 5 台机器的调度问题, 也能在 0.3 s 内给出调度解.

5 结 论

等待时间受限约束多存在于高温连续生产或中间产品性质不稳定的流程工业中. 本文针对其中的置换流水车间调度问题, 分析了工件序列中相邻工件的加工持续时间. 在此基础上构建 TSP 问题, 并采用最近插入法生成初始工件序列. 对于初始调度, 进一步利用针对等待时间受限的置换流水车间调度问题的扩展插入方法进行优化. 仿真实验表明: 本文提出的启发式算法能有效地求解等待时间受限的流水车间调度问题; 基于 TSP 的调度规则能获得质量较高的初始解, 且扩展插入方法具有良好的优化效果; 本文提出的下界计算方法能够获得较紧的下界值.

参考文献(References)

[1] 金锋, 宋士吉, 吴澄. 一类基于 FSP 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 FSP[J]. Control and Decision, 2007, 22(3): 247-251.)

[2] Yang D L, Chern M S. A two-machine flowshop sequencing problem with limited waiting time constraints[J]. Computers and Industrail Engineering, 1995, 28(1): 63-70.

[3] Fondrevelle J, Oulamara A, Portmann M C. Permutation flowshop scheduling problems with maximal and minimal time lags[J]. Computers & Operations Research, 2006, 33(6): 1540-1556.

[4] Su L H. A hybrid two-stage flowshop with limited waiting time constraints[J]. Computers and Industrial Engineering, 2003, 44(3): 409-424.

[5] Chen J S, Yang J S. Model formulations for the machine scheduling problem with limited waiting time constraints[J]. J of Information and Optimization Sciences, 2006, 27(1): 225-240.

[6] 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 T K, Yin Z T. A heuristic algorithm for flowshop scheduling with limited waiting times[J]. Chinese J of Management, 2009, 6(10): 1335-1339.)

[7] Fondrevelle J, Oulamara A, Portmann M C. Permutation flowshop scheduling problems with time lags to minimize the weighted sum of machine completion times[J]. Int J of Production Economics, 2008, 112(1): 168-176.

[8] Fondrevelle J, Oulamara A, Portmann M C, et al. Permutation flowshops with exact time lags to minimize maximum lateness[J]. Int J of Production Research, 2009, 47(23): 6759-6775.

[9] 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 Z T, Li T K, Xiao Y J. HGA algorithm for flowshop scheduling with limited waiting time[J]. Computer Engineering, 2009, 35(21): 4-6.)

[10] 王晶, 姚辉, 王艳亮. 有限等待流水车间调度邻域搜索算法[J]. 工业工程与管理, 2010, 15(3): 55-59.
(Wang J, Yao H, Wang Y L. Neighborhood search algotithm for flow shop scheduling problem with limited waiting time constraints[J]. Industrial Engineering and Management, 2010, 15(3): 55-59.)

[11] 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 Z T, Li T K. Hybrid algorithms for hybrid flowshop scheduling considering due dates and limited waiting times[J]. Industrial Engineering J, 2009, 12(1): 79-83.)

[12] Liu H, Gao L, Pan Q. A hybrid particle swarm optimization with estimation of distribution algorithm for solving permutation flowshop scheduling problem[J]. Expert Systems with Applications, 2011, 38(4): 4348-4360.

[13] Framinan J M, Gupta J N D, Leisten R. A review and classification of heuristics for permutation flowshop scheduling with makespan objective[J]. J of the Operational Research Society, 2004, 55(12): 1243-1255.

(下转第776页)