运筹学与控制论

DOI:CNKI:50-1165/N.20120115.1809.001

具有学习效应的重新排序问题*

张新功
(重庆师范大学 数学学院,重庆 400047)

摘要:重新排序问题是一种新型的排序模型,它有着重要的实际应用背景。生产部门根据自己的生产计划或是由客户提出的要求,在生产前一定时期内事先有一个作业方案,将已有的任务或订单按照某一规则安排好,使某一目标值最优。但是在即将开始生产之前或在生产过程中又有新的客户订单或任务到达。这时就要把新的任务和原有的还未加工的任务一起加工。为了不失信于对原客户的承诺或不耽误原任务的完成,这就要求在原有的工件或任务的次序不至于打乱得过多的前提下,使得总的目标函数值达到最优。本文考虑学习效应作用下的最小化总完工时间的重新排序问题,其中工件的加工时间是其所在序列加工位置有关的函数。对于最大序列错位、总序列错位和最大时间错位下的最小化总完工时间问题均给出了多项式时间算法,对于总时间错位下的最小化总完工时间问题提出了动态规划算法,并证明这个算法是拟多项式时间的。

关键词:重新排序;学习效应;多项式时间算法
中图分类号:O221.1 文献标志码:A 文章编号:1672-6693(2012)01-0001-06

重新排序问题研究可追溯到 20 世纪 80,90 年代。Bean 等[1]给出一种有效方式使得新的排列顺序与先前的排列顺序相匹配。Wu 等[2]提出单机下组合目标函数的重新排序问题,包括提前费用、推迟费用和序列改变的费用。Wu 等[3]研究重新车间排序问题,使得工件的最大完工时间与原排序的错位和最小,度量错位的方法是用原排序和新的排序中工件开工时间的差。Unal 等[4]考虑新工件的准备时间依赖于工件类型的单机排序问题,考虑在原排序中插入新的工件,以使得新工件的加权完工时间和为最小,而没有增加额外的准备时间或者使工件误工,要么给出了有效的算法,要么给出了计算复杂性分析,要么给出了启发式算法及其界。Hall 和 Potts[5]在前人研究的基础上,系统地研究了重新排序问题,引入了序列错位和时间错位的概念,研究在原最优排序和任意排序的基础上进行重新排序;在原排序的序列错位和时间错位不至于太大的情况下,使目标函数最优。给出了问题的多项式时间或拟多项式时间算法,或者给出计算复杂性分析。Yang[6]研究工件重新排序问题,为了减少扰动对原序列的负面影响,缩短新到达工件的加工时间减低其费用,称之为时间压缩费用,目标函数为:$\sum C_j + \sum c_j x_j + h\sum \Delta_j$,$x$ 为压缩时间向量,Δ_j 是工件 j 的时间错位。Yuan 和 Mu[7]考虑问题 $1\mid r_j, D_{\max}(\pi^*) \leqslant k \mid C_{\max}$,其中 $D_{\max}(\pi^*)$ 是最优序列中最大序列扰动,基于弱 ERD 性质(原始工件和新嵌入工件按照 ERD 规则排序),证明该模型能够在时间复杂性为 $O(n_N^2(n_0 + n_N))$ 内得到最优序,其中 n_N,n_0 分别为新到工件和以前系列中工件的个数。同时提出总序列扰动问题仍是开放的。幕运动[8]在他的博士论文中研究了重新排序在多目标情形、在线情形时的性质。Zhao 和 Tang[9]第一次尝试把重新排序问题引入工件加工时间可变的情形。对于简单的线性退化问题作了分析。

1 模型描述

设 $J_0 = \{J_1, \cdots, J_{n_0}\}$ 表示不可中断的原始工件集合,同时这些工件已经按照某种目标函数的要求排好

* 收稿日期:2011-08-29 修回日期:2011-11-23 网络出版时间:2012-01-15 18:09:00
资助项目:天元基金(No. 11126348);重庆师范大学博士研究基金(No. 11XLB027)
作者简介:张新功,男,讲师,博士,研究方向为组合优化、排序理论。
网络出版地址:http://www.cnki.net/kcms/detail/50.1165.N.20120115.1809.201201.1_001.html

最优序列,不妨设为π^*。设$J_N = \{J_{n_0+1},\cdots,J_{n_0+n_N}\}$表示新到达的工件集合。假设这些工件在0时刻到达,此时J_0中的工件已经事先确定好了。不失一般性,如果J_N中的工件在0时刻到达,接着只需把J_0中移除新工件到达前已经加工的工件,更新J_0和n_0即可。本文研究重新排序问题是将$J = J_0 \cup J_N$中的工件重新排列,使得考虑的目标函数在原始工件的错位约束条件下达到最优的问题,本文沿用Hall和Potts[5]的概念和记号,设$n = n_0 + n_N$,每个工件$J_j \in J$有整数基本加工时间p_j,如果工件$J_j \in J$在序列中第r个位置加工,则其实际加工时间为$p_{jr} = p_j r^a$,其中$a(<0)$是学习因子。对于J中工件的任意序列σ,定义以下的变量:$S_j(\sigma)$:工件$J_j \in J$开工时间;$C_j(\sigma)$:工件$J_j \in J$的完工时间;$D_j(\pi^*,\sigma)$:工件$J_j \in J_0$的序列错位,如果工件J_j在序列π^*和σ的位置分别为x和y,则

$$D_j(\pi^*,\sigma) = |y - x|;\Delta_j(\pi^*,\sigma) = |C_j(\sigma) - C_j(\pi^*)|:\text{工件 } J_j \in J_0 \text{ 的时间错位}$$

上述记号在不至于引起混淆的情况下,分别简记为$S_j, C_j, T_j, D_j(\pi^*)$和$\Delta_j(\pi^*)$。

本文仅仅考虑单机的情形,定义下面几种错位约束(其中$k \geq 0$):

$D_{\max}(\pi^*) \leqslant k:\max_{J_j\in J_0}\{D_j(\pi^*)\} \leqslant k$,工件的最大序列错位不能超过$k$;

$\sum D_j(\pi^*) \leqslant k:\sum_{J_j\in J_0} D_j(\pi^*) \leqslant k$,工件的总序列错位不能超过$k$;

$\Delta_{\max}(\pi^*) \leqslant k:\max_{J_j\in J_0}\{\Delta_j(\pi^*)\} \leqslant k$,工件的最大加工时间错位不可能超过$k$;

$\sum \Delta_j(\pi^*) \leqslant k:\sum_{J_j\in J_0}\Delta_j(\pi^*) \leqslant k$,工件的总的加工时间错位不可能超过$k$。

2 最小化总完工时间问题

引理1 对于问题$1|p_j r^a, \Gamma \leqslant k|\sum C_j$,其中

$$\Gamma \in \{D_{\max}(\pi^*), \sum D_j(\pi^*), \Delta_{\max}(\pi^*), \sum \Delta_j(\pi^*)\}$$

存在一个最优序列使得J_0中的工件和在序列π^*中一样按照SPT排列,J_N中的工件按照SPT序排列,并且工件之间没有空闲。

证明 首先分析J_0中的工件。考虑一个最优序σ^*,其中J_0中的工件不是和序列π^*中工件一样按照SPT序排列的。假设J_i是J_0中的而在序列σ^*中不是按照SPT序排列的最小下标的工件,工件$J_j(j>i)$是在σ^*中排在J_i之前的J_0中的最后一个工件。假设工件J_i和工件J_j之间的工件为$J_1^0,\cdots,J_h^0$。由于$i<j$,则$p_i < p_j$。假设工件J_j排在序列σ^*的第r个位置加工,并且序列σ^*中工件J_j的开工时间为t_0,则有

$$C_j(\sigma^*) = t_0 + p_j r^a, C_k(\sigma') = t_0 + p_j r^a + p_1^0(r+1)^a + \cdots + p_k^0(r+k)^a, k = 1,\cdots,h$$

$$C_i(\sigma^*) = t_0 + p_j r^a + p_1^0(r+1)^a + \cdots + p_h^0(r+h)^a + p_i(r+h+1)^a$$

对于工件J_j和工件J_i执行一个位置交换,其他位置的工件不变,构成一个新的序列σ'。在序列σ'中,工件J_i的开工时间为t_0,则有

$$C_i(\sigma') = t_0 + p_i r^a, C_k(\sigma') = t_0 + p_i r^a + p_1^0(r+1)^a + \cdots + p_k^0(r+k)^a, k = 1,\cdots,h$$

$$C_j(\sigma') = t_0 + p_i r^a + p_1^0(r+1)^a + \cdots + p_h^0(r+h)^a + p_j(r+h+1)^a$$

由于$p_i < p_j$,则

$$C_j(\sigma^*) - C_i(\sigma') = (p_j - p_i)r^a \geqslant 0, C_k(\sigma^*) - C_k(\sigma') = (p_j - p_i)r^a \geqslant 0, k = 1,\cdots,h$$

$$C_i(\sigma^*) - C_j(\sigma') = (p_j - p_i)(r^a - (r+h+1)^a) \geqslant 0$$

于是经过交换后,序列σ^*中工件的总完工时间并没有比序列σ'中工件的总完工时间增加。

假设工件J_j在序列π^*的位置为k_1,工件J_i在序列π^*的位置为k_2。若$r+h+1 \geqslant k_2$,则有

$$D_j(\pi^*,\sigma') = r + h + 1 - k_2;D_i(\pi^*,\sigma^*) = r + h + 1 - k_1$$

由于$i<j$意味着$k_1 < k_2$,有$D_j(\pi^*,\sigma') \leqslant D_i(\pi^*,\sigma^*)$。

若$r+h+1 < k_2$,则有$D_j(\pi^*,\sigma') = k_2 - (r+h+1), D_j(\pi^*,\sigma^*) = k_2 - r - 1$。

因此$D_j(\pi^*,\sigma') \leqslant D_i(\pi^*,\sigma^*)$,则有$D_{\max}(\pi^*,\sigma') \leqslant D_{\max}(\pi^*,\sigma^*)$。

另一种情形,由于

$$D_i(\pi^*,\sigma') = D_i(\pi^*,\sigma^*) - h \text{ 和 } D_j(\pi^*,\sigma') \leqslant D_j(\pi^*,\sigma^*) + h$$

接着可以演绎为 $\sum D_j(\pi^*,\sigma') \leqslant \sum D_j(\pi^*,\sigma^*)$。

此外,如果 $C_j(\sigma') \geqslant C_j(\pi^*)$,则有 $\Delta_j(\pi^*,\sigma') = C_j(\sigma') - C_j(\pi^*)$。由于

$$\Delta_i(\pi^*,\sigma^*) = C_i(\sigma^*) - C_i(\pi^*) \geqslant C_j(\sigma') - C_i(\pi^*) \geqslant C_j(\sigma') - C_j(\pi^*) = \Delta_j(\pi^*,\sigma')$$

则 $\Delta_j(\pi^*,\sigma') < \Delta_i(\pi^*,\sigma^*)$。如果 $C_j(\sigma') < C_j(\pi^*)$, 则有

$$\Delta_j(\pi^*,\sigma') = C_j(\pi^*) - C_j(\sigma') \leqslant C_j(\pi^*) - C_j(\sigma^*) = \Delta_j(\pi^*,\sigma^*)$$

接着可以得到 $\Delta_{\max}(\pi^*,\sigma') < \Delta_{\max}(\pi^*,\sigma^*)$。

类似地,由于 $\Delta_i(\pi^*,\sigma') = \Delta_i(\pi^*,\sigma^*) - h'$ 和 $\Delta_j(\pi^*,\sigma') \leqslant \Delta_j(\pi^*,\sigma^*) + h'$, 其中 $h' = C_i(\sigma^*) - C_i(\sigma')$。可以推出 $\sum \Delta_j(\pi^*,\sigma') \leqslant \sum \Delta_j(\pi^*,\sigma^*)$。

可以得出 σ' 是可行序列和最优序列。这样经过有限次的交换可以证明,存在这样的最优序列使得 J_0 中的工件和 π^* 中工件一样按照 SPT 序排列。类似地可以证明 J_N 中的工件也是按照 SPT 序排列。同时也正是 J_0 中的工件和 π^* 中工件具有相同的 SPT 序和序列的最优性,可以证明这样的最优序中工件之间不存在空闲时间。否则,仅仅需要去掉这些空闲时间即可,序列仍然是可行的并且序列中工件的总误工并不会增加。

证毕

定义　这种原始工件集合 J_0 和新工件集合 J_N 中工件都按照 SPT 序排列的性质,称之为(SPT,SPT) 性质。

接下来首先考虑问题 $1\,|\,p_{jr} = p_j r^a, D_{\max}(\pi^*) \leqslant k\,|\,\sum C_j$,根据引理 1, 集合 J_N 中的工件最多有 k 个排在集合 J_0 中的最后一个工件的前面,并且这些工件具有较小的基本加工时间。从而提出最大序列错位约束下的算法。

算法 1　输入 :输入 p_j, d_j,其中 $j = 1,\cdots,n$;k 和 π^*, 其中 $k \leqslant n_N$;

标号 :将 J_N 中的工件按照 SPT 序进行排列 ;

构造序列 :将工件 $1,\cdots,n_0,n_0+1,\cdots,n_0+k$ 按照 SPT 序排列在前 n_0+k 个位置。再将工件 $n_0+k+1,\cdots,n_0+n_N$ 按照 SPT 序排在后面的 n_N-k 个位置。

定理 1　对于问题 $1\,|\,p_{jr} = p_j r^a, D_{\max}(\pi^*) \leqslant k\,|\,\sum C_j$,算法 1 可以在多项式时间 $O(n + n_N \log n_N)$ 内给出最优序。

证明　根据引理 1,约束条件 $D_{\max}(\pi^*) \leqslant k$ 允许集合 J_N 中的具有最小工期的前 k 个工件排到集合 J_0 中的最后一个工件之前,且这 k 个集合 J_N 中的工件具有最小的工期。经典的排序理论证明前一组的工件按照 SPT 序排列,而引理 1 确立集合 J_N 中剩下的 n_N-k 个工件也按照 SPT 序排列。注意到对集合 J_N 中的工件进行标号需要 $O(n_N \log n_N)$ 时间 ,构造序列的前 n_0+k 个位置和后面的 n_N-k 各位置仅仅需要 $O(n)$ 时间。因此算法的总时间界为 $O(n + n_N \log n_N)$。　证毕

考虑问题 $1\,\left|\,p_{jr} = p_j r^a, \sum D_j(\pi^*) \leqslant k\,\right|\,\sum C_j$。由引理 1, 集合 J_N 中的工件排在集合 J_0 中的最后一个工件的前面造成 J_0 中的工件总的序列错位不超过 k,并且这些工件具有较小的工期。从而提出总序列错位约束下的算法。

算法 2　输入 :输入 p_j,d_j,其中 $j = 1,\cdots,n$;k 和 π^*,其中 $k \leqslant n_0 n_N$;

标号 :将 J_N 中的工件按照 EDD 序进行排列 ;

值函数 :$f(i,j,\delta,r)$ = 表示工件 $J_1,\cdots,J_i$ 和 $J_{n_0+1},\cdots,J_{n_0+j}$ 的部分序列在工件 $J_1,\cdots,J_i$ 保持其相对位置不变且总序列错位为 δ 并且最后一个工件的位置为 r 时总误工的最小值 ;

边界条件 :$f(0,0,0,0) = 0$;

最优值 :$\min_{0\leqslant\delta\leqslant k}\{f(n_0,n_N,\delta,n)\}$;

递归条件：

$$f(i,j,\delta,r) = \min\begin{cases} f(i-1,j,\delta-j,r-1) + C_{i-1}^{j} + p_j r^a \\ f(i,j-1,\delta,r-1) + C_i^{j-1} + p_j r^a \end{cases}$$

其中 C_i^j 表示对于工件 $J_1,\cdots,J_i$ 和 $J_{n_0+1},\cdots,J_{n_0+j}$ 的部分序列最大完工时间。

在递归关系中，第一项对应部分序列以集合 J_0 中的工件 J_i 结束的情况，此时 J_N 中的各 j 工件在工件 J_i 之前已经排好，此时总的序列错位增加量为 j；第二项对应部分序列是以 J_N 中的工件 J_{n_0+j} 结尾的情况。

定理2 对于问题 $1\left|p_{jr}=p_j r^a, \sum D_j(\pi^*) \leqslant k\right| \sum C_j$，算法2能够在 $O(n_0^2 n_N^2)$ 时间内得到最优序。

证明 根据引理1，仅仅需要计算集合 J_0 的工件和集合 J_N 的工件的SPT序进行所有可能的排列方式。算法2就是通过比较所有状态的费用函数，从而找到最优的序列。因为 $i \leqslant n_0$，$j \leqslant n_N$ 和 $\delta \leqslant k \leqslant n_0 n_N$，因此总共有 $O(n_0^2 n_N^2)$ 个状态变量的值，标号阶段需要 $O(n_N \log n_N)$ 时间。递归关系要求每一组状态变量计算函数值为常数，因此算法2的计算复杂性为 $O(n_0^2 n_N^2)$。 证毕

进一步考虑问题 $1\,|\,p_{jr}=p_j r^a, \Delta_{\max}(\pi^*) \leqslant k\,|\,\sum C_j$，由引理1，$C_j \geqslant C_j(\pi^*)$，$J_j \in J_0$。接着构造限制约束 $C_j \leqslant C_j(\pi^*)+k$，相当于每个工件 $J_j \in J_0$ 有一个截止工期 $\bar{d}_j = C_j(\pi^*)+k$。限制约束的结果把该问题转化为带有截止工期的总完工时间问题，即是 $1\,|\,p_{jr}=p_j r^a, \bar{d}_j\,|\,\sum C_j$，对于 $J_j \in J_0$，$\bar{d}_j = C_j(\pi^*)+k$；对于 $J_j \in J_N$，$\bar{d}_j = \infty$。置 J 表示需要排的工件集合，J^c 表示已经排好的工件集合，$C_{\max}(J)$ 表示 J 中工件的最大完工时间。在随后的向后追溯算法，集合 J 排在集合 J^c 前面。

算法3 步骤1：$J=\{J_1,\cdots,J_n\}$，$J^c=\varnothing$；

步骤2：把集合 J 中的工件按照SPT序排列，计算最大完工时间 $C_{\max}(J)$。找出集合 $\hat{J} \in \{J_j \in J\,|\,\bar{d}_j > C_{\max}(J)\}$，取 J_j^*，使得 $p_j^* = \max\{p_j\,|\,J_j \in \hat{J}\}$。把工件 J_j^* 排在序列的最后一个位置；

步骤3：置 $J = J - \{J_j^*\}$，$J^c = J^c + \{J_j^*\}$，如果 $J=\varnothing$ 停止，否则回到步骤2。

证明 考虑一个序列 $\pi=(\pi(1),\cdots,\pi(n))$，其中 $J_j=\pi(i)$ 意味着工件 J_j 安排在序列 π 的第 i 个位置。$S=\{\pi(1),\cdots,\pi(r)\}$，假设存在工件 $J_l=\pi(k)$，$1 \leqslant k \leqslant r-1$。使得 $p_l = \max\limits_{J_j \in \hat{J}}\{p_j\} > p_{\pi(r)}$，其中 $\hat{J}=\{J_j \in S\,|\,\bar{d}_j \geqslant C_{\max}(J)\}$。

很明显序列 π 并不是有算法3生成的序列，否则的话 $p_{\pi(r)} = \max\limits_{J_j \in \hat{J}}\{p_j\}$ 和 $J_j=\pi(k)$。对于序列 π，通过交换工件 J_l 和 $J_{\pi(r)}$ 得到一个新的序列 π'。根据引理1和通过计算可得

$$C_i(\pi') = C_i(\pi), i = 1,\cdots,k-1$$

$$C_{\pi(k)}(\pi') = p_{\pi(1)} + \cdots + p_{\pi(k-1)}(k-1)^a + p_{\pi(r)}k^a \leqslant C_{\pi(r)}(\pi) = p_{\pi(1)} + \cdots + p_{\pi(k-1)}(k-1)^a + p_{\pi(k)}k^a$$

$$C_i(\pi') = p_{\pi(1)} + \cdots + p_{\pi(k-1)}(k-1)^a + p_{\pi(r)}k^a + \cdots + p_{\pi(i)}i^a \leqslant$$

$$C_i(\pi) = p_{\pi(1)} + \cdots + p_{\pi(k-1)}(k-1)^a + p_{\pi(k)}k^a + \cdots + p_{\pi(i)}i^a, i = k+1,\cdots,r-1$$

$$C_{\pi(r)}(\pi') = p_{\pi(1)} + \cdots + p_{\pi(k-1)}(k-1)^a + p_{\pi(r)}k^a + \cdots + p_{\pi(k)}r^a \leqslant$$

$$C_{\pi(k)}(\pi) = p_{\pi(1)} + \cdots + p_{\pi(k-1)}(k-1)^a + p_{\pi(k)}k^a + \cdots + p_{\pi(r)}r^a$$

$$C_i(\pi') = C_{\pi(k)}(\pi') + p_{\pi(r+1)}(r+1)^a + \cdots + p_{\pi(i)}i^a \leqslant C_i(\pi) =$$

$$C_{\pi(k)}(\pi) + p_{\pi(r)}k^a + p_{\pi(r+1)}(r+1)^a + \cdots + p_{\pi(i)}i^a, i = r+1,\cdots,n$$

上述证明意味着，经过这样的交换能够改进序列的总完工时间，类似地经过重复的交换使得所有的工件满足算法3的条件，并且也可以得到算法2能够使这个问题得到最优序列。下面确定算法3的时间复杂性，步骤2需要的时间复杂性为 $O(n \log n)$，步骤3的运行时间为 $O(n)$，因此算法3的总时间复杂性为 $O(n \log n)$。 证毕

基于上述分析，得到定理3。

定理3 对于问题 $1\,|\,p_{jr}=p_j r^a, \Delta_{\max}(\pi^*) \leqslant k\,|\,\sum C_j$ 能够在 $O(n+n_N \log n_N)$ 时间内得到最优序列。

证明　为了构造问题 $1\mid p_{jr}=p_j r^a,\vec{d}_j\mid\sum C_j$ 的实例,集合 J_0 中的工件的加工时间与截止工期是一致关系的,即 $p_1\leqslant\cdots\leqslant p_{n_0}$ 和 $\vec{d}_1\leqslant\cdots\leqslant\vec{d}_{n_0}$。因此根据(SPT,SPT)性质,利用算法 3 安排工件。为了有效地执行算法 3,首先将 J_N 中的工件按照 SPT 序进行排列,可以在 $O(n_N\log n_N)$ 时间内完成。将要排在最后位置的工件,要么是部分序列中 π^* 中的最后一个未安排的工件;要么是部分 J_N 中按照 SPT 序排列的最后一个工件,依赖于这两个工件的截止时间和加工时间。由于构造序列需要 $O(n)$ 时间,则总的时间复杂性为 $O(n+n_N\log n_N)$。　　证毕

最后考虑问题 $1\left|p_{jr}=p_j r^a,\sum\Delta_j(\pi^*)\leqslant k\right|\sum C_j$,由引理 1,工件 J_0 中工件总的时间错位最多为 k,并且集合 J_0 中最后一个工件之前的集合 J_N 中工件具有最小的加工时间。相似于算法 2,可以通过将集合 J_0 和 J_N 的工件进行融合,下面的动态规划方法得到在总的时间错位约束下的最优序列,首先假设:$P_N=\sum_{J_j\in J_N}p_j$ 和 $P_N=\sum_{J_j\in J_0}p_j$。

算法 4　输入:输入 p_j,其中 $j=1,\cdots,n$;k 和 π^*,其中 $k\leqslant n_0P_N$;

标号:将 J_N 中的工件按照 SPT 序进行排列;

值函数:$f(i,j,\delta,r)$ 表示工件 $J_1,\cdots,J_i$ 和 $J_{n_0+1},\cdots,J_{n_0+j}$ 的部分序列在工件 $J_1,\cdots,J_i$ 保持其相对位置不变且总的时间错位为 δ 和最后一个工件的位置为 r 时的总误工的最小值;

边界条件:$f(0,0,0,0)=0$;

最优值:$\min_{0\leqslant\delta\leqslant k}\{f(n_0,n_N,\delta,n)\}$;

递归关系:

$$f(i,j,\delta,r)=\min\begin{cases}f\left(i-1,j,\delta-\sum_{h=n_0+1}^{n_0+j}p^0_{[h]},r-1\right)+C^j_{i-1}+p_j r^a\\ f(i,j-1,\delta,r-1)+C^{j-1}_i+p_j r^a\end{cases}$$

其中 $p^0_{[h]}$ 为工件 J^0_h 的实际加工时间,C^j_i 表示对于工件 $J_1,\cdots,J_i$ 和 $J_{n_0+1},\cdots,J_{n_0+j}$ 的部分序列最大完工时间。

在递归关系中,第一项对应部分序列以集合 J_0 中的工件 J_i 结束的情况,此时 J_N 中的各 j 工件在工件 J_i 之前已经排好,此时总的时间错位增加量为 $\sum_{h=n_0+1}^{n_0+j}p^0_{[h]}$;第二项对应部分序列是以 J_N 中的工件 J_{n_0+j} 结尾的情况。

定理 4　对于问题 $1\left|p_{jr}=p_j r^a,\sum\Delta_j(\pi^*)\leqslant k\right|\sum C_j$,在 $O(n_0^2n_N\min\{n_0P_N,n_NP_0\})$ 时间内可以得到算法 4 的最优解。

证明　考虑时间复杂性,因为 $i\leqslant n_0$,$j\leqslant n_N$ 和 $\delta\leqslant k\leqslant n_0P_N$,因此有 $O(n_0^2n_NP_N)$ 个状态变量的值。标号阶段需要 $O(n_N\log n_N)$ 时间。相似于定理 2,可以得到算法 4 的时间复杂性为 $O(n_0^2n_NP_n)$。通过逆向 J_0 和 J_N 规则,可以认为 J_N 中的工件具有时间错位,同时也可以证明(SPT,SPT)规则对于与非空闲序列也是成立的,总完工时间等于 J_0 中的工件的总时间错位加上 J_N 中的工件的总时间错位。因此相似于算法 4 可以提出一个替代互斥算法,其中 i,j 和 J_N 中的工件的总时间错位作为状态变量,J_0 中的工件的总时间偏差的约束在最优值阶段执行。这个替代互斥算法的总时间复杂性为 $O(n_0^2n_N\min\{n_0P_N,n_NP_0\})$。　　证毕

3 结论

本文讨论了重新排序问题。主要贡献是把重新排序问题引入学习效应概念。学习效应就是工件的实际加工时间和其在序列中加工所处的位置有关,考虑了在序列错位和时间错位扰动下的总完工时间问题,提出了动态规划算法或多项式时间算法,拟多项式时间算法。

参考文献:

[1] Bean J C,Birge I R,Mitenthal J,et al. Match up scheduling with multiple resource,release dates and disruption[J]. Operations Research,1991,39:470-483.

[2] Wu S D,Storer R H,Chang P C. One-machine rescheduling heuristics with efficiency and stability as criteria[J]. Computer and

Operations Research,1993,20:1-14.
[3] Wu S D,Storer R H,Chang P C. A rescheduling procedure for manufacturing systems under random disruptions[C]//Fandel G T, Gulledge A J. New directions for operations research in manufacturing. Berlin,Germany:Springer,1992:292-308.
[4] Unal A T,Uzsoy R,Kiran A S. Rescheduling on a single machine with part-type depend setup times and deadlines[J]. Annals of Operations Research,1997,70:93-113.
[5] Hall N G,Potts C N. Rescheduling for new orders[J]. Operations Research,2004,52:440-453.
[6] Yang B B. Single machine rescheduling with new jobs arrivals and processing time compression[J]. The International Journal of Advance Manufacture Technology,2009,34:378-384.
[7] Yuan J J,Mu Y D. Rescheduling with release dates to minimize makespan under a limit on the maximum sequence disruption[J]. European Journal of Operation Research,2007,182:936-944.
[8] 幕运动. 关于重新排序问题的研究[D]. 郑州:郑州大学,2007.
[9] Zhao C L,Tang H Y. Rescheduling problems with deteriorating jobs under disruptions[J]. Applied Mathematical Modelling,2010, 34:238-243.

Operations Research and Cybernetics

Rescheduling Problems with Learning Effect

ZHANG Xin-gong

(College of Mathematics Science, Chongqing Normal University, Chongqing 400047, China)

Abstract: Rescheduling is one of significant modern scheduling models and it is motivated by important applications. The problem of rescheduling in the face of dynamically arriving orders is faced constantly by manufacturing companies. The manufacturing companies have been made their production schedules earlier according to their planes or some customers' call, and scheduled the jobs by some rules to optimize some objective. However it frequently happens that additional orders arrived after a schedule has been developed, these orders must then be integrated into the schedule so as to optimize some measure of system performance without causing undue disruption in the shop. In this paper, I consider make span problems with rescheduling under the learning effect. The job processing times is defined by a function of its position in the sequence. For the maximum sequence disruption, the total sequence disruption and the maximum time disruption, some polynomial time algorithms for minimizing the total completion time problems are presented. For minimizing the total completion time problems, a dynamic programming algorithm is given under the total time disruption. Furthermore, it is showed that this algorithm is pseudpolynomial time algorithm.

Key words: rescheduling; learning effect; polynomial-time algorithms

(责任编辑 黄 颖)