

问题 $1|d_j = d|\sum w_j T_j$ 的一个全多项式近似方案

张 喆¹, 李文华²

(1. 中原工学院理学院, 河南 郑州 450007)

(2. 郑州大学数学系, 河南 郑州 450001)

摘要: 本文对具有相同工期的单机最小化加权总误工问题进行了讨论. 利用强 NP - 困难问题 $1||\sum w_j T_j$ 的一个 $O(n^2)$ 时间的近似算法, 把该算法得到的目标值作为问题 $1|d_j = d|\sum w_j T_j$ 的一个上界, 对问题 $1|d_j = d|\sum w_j T_j$ 给出全多项式近似方案 (FPTAS). 已知问题 $1|d_j = d|\sum w_j T_j$ 是一般意义下的 NP - 困难问题, 并且已经有人对该问题给出了拟多项式时间算法, 本文对已有结果进行了扩充.

关键词: 相同工期; 加权总误工; 全多项式近似方案

MR(2010) 主题分类号: 90C27

中图分类号: 0221.7

文献标识码: A

文章编号: 0255-7797(2015)04-1005-07

1 引言

我们研究的问题可简记为 $1|d_j = d|\sum w_j T_j$, 这是一个经典的排序模型. 最小化加权总误工问题在实际中经常遇到, 然而有关的结果却并不是很多, 由于对工期为任意值的问题的研究相当困难, 因此近来对特殊工期的模型的研究比较多. 已知问题 $1||\sum w_j T_j$ 是强 NP - 困难的^[1,2], Potts 和 Van Wassenhove^[3] 给出了该问题的分支定界算法, Schrage 和 Baker^[4] 利用动态规划的方法得到了 $O(2^n)$ 时间的算法, Cheng 等人^[5] 给出了此问题的 $O(n^2)$ 时间的近似算法. 对于特殊工期的模型研究集中在具有相同工期 $d = d_j$, 工期等量松弛 (SLK) $d_j = p_j + q$, 其中 q 为给定的常数, 以及更一般情形工时加等待工期 (PPW) $d_j = \alpha p_j + q$, 其中 α, q 都为给定的常数. 除了研究特殊工期模型, 还有一些学者则关注了在其他参数上进行特殊限制的情形. 如问题 $1|w_j = 1|\sum w_j T_j$, $1|w_j = kp_j|\sum w_j T_j$, $1|p_j = p|\sum w_j T_j$ 在文献 [1, 2, 6, 7] 中都有所研究. 对于我们研究的问题 $1|d_j = d|\sum w_j T_j$, 已知有下面的引理:

引理 1.1^[8] 问题 $1|d_j = d|\sum w_j T_j$ 是一般意义下的 NP - 困难问题.

Lawler 和 Moore^[9] 已经对该问题给出了一个拟多项式时间算法, 我们将给出该问题的 FPTAS 算法, 方法类似于 Cheng 等人在文 [5] 中对问题 $1|d_j = p_j + q|\sum w_j T_j$ 给出的算法, 即将问题 $1||\sum w_j T_j$ 的近似算法得到的目标值作为问题 $1|d_j = d|\sum w_j T_j$ 的一个上界, 从而得到我们的结果.

下面首先介绍一下 Cheng 等对问题 $1||\sum w_j T_j$ 设计的近似算法 “Modified Lawler’s Algorithm”. 它可以描述如下: 有 n 个工件, 设每个工件 J_j 的加工时间为 p_j , 工期为 d_j , 权重为 w_j , $1 \leq j \leq n$. 对每个工件, 将工期修改为 $d'_j = \max\{p_j, d_j\}$, 这样得到一个排序 π^* 使

*收稿日期: 2014-03-30

接收日期: 2014-07-31

基金项目: 国家自然科学基金资助 (11401604; 11401605); 国家天元基金资助 (11326191); 河南省自然科学基金资助 (132300410392).

作者简介: 张喆 (1979-), 女, 河南郑州, 讲师, 主要研究方向: 组合最优化.

得目标函数 $g(\pi) = \max_{1 \leq j \leq n} w_j T'_j(\pi)$ 最小, 其中

$$T'_j(\pi) = \max\{0, C_j(\pi) - d'_j\},$$

则 π^* 为原问题的一个近似解. 下面是该算法的具体步骤:

Modified Lawler's 算法:

第一步 修改工期为 $d'_j = \max\{p_j, d_j\}$, $1 \leq j \leq n$;

第二步 令 $k := n$, $\mathcal{J} := \{J_1, J_2, \dots, J_n\}$, $P := \sum_{j=1}^n p_j$;

第三步 对所有的 $J_j \in \mathcal{J}$, 选择使 $w_j \times \max\{0, P - d'_j\}$ 最小的工件 J_i , 然后令 $\pi^*(k) = i$;

第四步 令 $k := k - 1$, $\mathcal{J} := \mathcal{J} \setminus \{J_i\}$ 且 $P := P - p_i$;

第五步 如果 $k = 0$, 则停止; 否则, 转回第三步.

引理 1.2 [9] 对于问题 $1|\Sigma w_j T_j$, Modified Lawler's 算法是一个 $(n-1)$ -近似算法, 时间复杂性为 $O(n^2)$.

2 动态规划算法

下面我们将要研究问题 $1|d_j = d|\Sigma w_j T_j$ 的有效算法:

引理 2.1 对问题 $1|d_j = d|\Sigma w_j T_j$ 的任意给定的排序 π , 如果 $J_k(\pi)$ 是误工工件, 则 $J_i(\pi)$ ($k \leq i \leq n$) 也为误工工件, 且有

$$T_i(\pi) = T_k(\pi) + \sum_{j=k+1}^i p_j(\pi).$$

证 注意到 $C_k(\pi) = T_k(\pi) + d_k(\pi) = T_k(\pi) + d$, 因此, 对 $k \leq i \leq n$ 有

$$C_i(\pi) = C_k(\pi) + \sum_{j=k+1}^i p_j(\pi) = T_k(\pi) + \sum_{j=k+1}^i p_j(\pi) + d,$$

即

$$T_i(\pi) = T_k(\pi) + \sum_{j=k+1}^i p_j(\pi).$$

引理 2.2 对问题 $1|d_j = d|\Sigma w_j T_j$, 设 $J_k(\pi)$ 是最优排序 π 下第一个误工工件, 则以下结论成立:

(1) $\frac{w_k(\pi)}{p_k(\pi)} \geq \frac{w_{k+1}(\pi)}{p_{k+1}(\pi)} \geq \dots \geq \frac{w_n(\pi)}{p_n(\pi)}$;

(2) 如果存在两个工件 $J_i(\pi)$ 和 $J_t(\pi)$ 满足 $i, t \geq k$ 且 $\frac{w_i(\pi)}{p_i(\pi)} = \frac{w_t(\pi)}{p_t(\pi)}$, 则交换 i 和 t 的位置得到的仍为最优排序;

(3) 前 $k-1$ 个不误工工件按任意顺序排列都不影响最优排序.

证 (1) 令 π 是一个最优排序, 但不满足 $\frac{w_k(\pi)}{p_k(\pi)} \geq \frac{w_{k+1}(\pi)}{p_{k+1}(\pi)} \geq \dots \geq \frac{w_n(\pi)}{p_n(\pi)}$. 则在 π 中一定至少存在两个相邻的工件假设 J_l 和 J_m , 满足 $k \leq l < m$ 且 $\frac{w_l(\pi)}{p_l(\pi)} < \frac{w_m(\pi)}{p_m(\pi)}$. 设工件 J_l 的开始时间为 $S_l(\pi)$, 我们可以交换工件 J_l 和 J_m 的位置得到一个新的排序 π' . 则有

$$\begin{aligned} C_l(\pi) &= S_l(\pi) + p_l(\pi), C_m(\pi) = S_l(\pi) + p_l(\pi) + p_m(\pi), \\ C_m(\pi') &= S_l(\pi) + p_m(\pi), C_l(\pi') = S_l(\pi) + p_m(\pi) + p_l(\pi). \end{aligned}$$

因为设 $J_k(\pi)$ 是 π 下第一个误工工件, 则由引理 2.1, 目标值可以写为

$$f(\pi) = \sum_{j=k}^n w_j(\pi) C_j(\pi) - \sum_{j=k}^n w_j(\pi) d.$$

那么

$$\begin{aligned} f(\pi) - f(\pi') &= (w_l(\pi) C_l(\pi) + w_m(\pi) C_m(\pi)) - (w_l(\pi') C_l(\pi') + w_m(\pi') C_m(\pi')) \\ &= w_m(\pi) p_l(\pi) - w_l(\pi) p_m(\pi) = w_l(\pi) w_m(\pi) \left(\frac{p_l(\pi)}{w_l(\pi)} - \frac{p_m(\pi)}{w_m(\pi)} \right) > 0, \end{aligned}$$

与 π 的最优性矛盾. 故结论 (1) 成立.

(2) 类似于 (1) 的证明由二交换法可证得.

(3) 由于 $J_1(\pi), \dots, J_{k-1}(\pi)$ 是不误工工件, 则 $C_j(\pi) \leq d$ ($j = 1, \dots, k-1$), 因此有 $\sum_{j=1}^{k-1} p_j(\pi) \leq d$, 故前 $k-1$ 个不误工工件按任意顺序排列都能保证它们不误工.

我们将满足以上三条的排序 π 称为一个正规排序, 以下我们只考虑正规排序. 下面是算法的思想: 假设先将工件按 WLPT 序排好, 即 $\frac{w_1}{p_1} \leq \frac{w_2}{p_2} \leq \dots \leq \frac{w_n}{p_n}$. 令

$$\begin{aligned} P &= p_1 + p_2 + \dots + p_n, \\ \mathcal{J}_j &= \{J_1, J_2, \dots, J_j\}, \\ P(j) &= p_1 + p_2 + \dots + p_j \quad (1 \leq j \leq n). \end{aligned}$$

对每一个 j , 我们构造一个包含三元组 (X, t, c) 的所有可能性的集合 M_j , 使得下面三条成立:

1. $X \subseteq \mathcal{J}_j$ 为正规排序 π 下最后 $|X|$ 个误工工件的集合, 则 $\mathcal{J}_j \setminus X$ 为 π 下不误工工件的集合.

2. $t = P - \sum_{J_i \in X} p_i$ 是 π 下 X 中第一个误工工件的开始时间.

3. c 为 π 下 X 中所有工件的加权误工时间总和.

实际中, 一些无用的三元组将会从 M_j 中删除掉. 我们用 $SL = (X, c)$ 表示存在一个正规排序, 其中集合 X 包含所有的误工工件, c 为加权误工时间和.

一开始, 令 $M_0 = \{(\emptyset, P, 0)\}$, 且 $SL = (\emptyset, +\infty)$. 由于 $\frac{w_1}{p_1}$ 是 $\frac{w_i}{p_i}$ ($1 \leq i \leq n$) 中最小的, 由引理 4 知或者存在一最优排序 π 使得 J_1 是 π 下最后一个误工工件, 或者存在一个最优排序 π' 使得 J_1 是 π' 下的不误工工件. 对第一种情形, 则 M_1 应包含三元组 $(\{J_1\}, P - p_1, w_1(P - d) | P > d)$; 对第二种情形, 则 M_1 应包含三元组 $(\emptyset, P, 0)$, 故

$$M_1 = \{(\emptyset, P, 0)\} \cup \{(\{J_1\}, P - p_1, w_1(P - d) | P > d)\}.$$

一般地, 假设 M_{j-1} 已确定, 对每个 $(X, t, c) \in M_{j-1}$, 存在一族正规排序 $\{\sigma\}$, 使得在 σ 下 X 中最后 $|X|$ 个误工工件按 WSPT 序排列, 且 σ 下 $\mathcal{J}_j \setminus X$ 中的为不误工工件, 由 σ 的正规性, J_j 或者是 σ 下的不误工工件, 或者是 σ 下工件集 $\{J_j, J_{j+1}, \dots, J_n\}$ 中的最后一个误工工件.

如果 J_j 是不误工的, 则为确保 $\mathcal{J}_j \setminus X$ 中工件按时完工, 我们必须有 $P(j) - (P - t) \leq d$. 也即是这种情况下存在一族正规排序 $\{\sigma'\} \subseteq \{\sigma\}$ 使得 σ' 下 X 中最后 $|X|$ 个误工工件以 WSPT 序排列, $\mathcal{J}_j \setminus X$ 中的工件为 σ' 下的不误工工件, σ' 下 X 中工件的开始时间为 t , 且 σ' 下 X 中工件的加权误工时间和为 c . 结构见图 1.

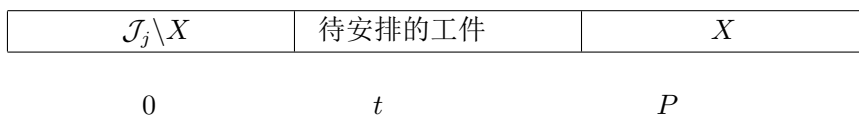


图 1

如果工件 J_j 为误工工件, 我们须有 $t > d$. 也就是说这种情况下, 存在一族正规排序 $\{\sigma''\} \subseteq \{\sigma\}$ 使得 σ'' 下最后 $|X| + 1$ 个误工工件是 $X \cup \{J_j\}$ 中工件按 WSPT 序排列, 且工件 J_j 是工件集 $X \cup \{J_j\}$ 中的第一个误工工件, $\mathcal{J}_j \setminus (X \cup \{J_j\}) = \mathcal{J}_{j-1} \setminus X$ 是 σ'' 下的不误工工件, σ'' 下 $X \cup \{J_j\}$ 中工件的开始时间为 $t - p_j$, σ'' 下 $X \cup \{J_j\}$ 中工件的加权误工时间总和为 $c + w_j(t - d)$. 结构见图 2.

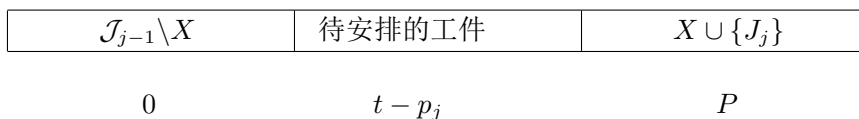


图 2

因此, 我们令

$$M_j = \{(X, t, c) | (X, t, c) \in M_{j-1}, P(j) - (P - t) \leq d\} \\ \cup \{(X \cup \{J_j\}, t - p_j, c + w_j(t - d)) | (X, t, c) \in M_{j-1}, t > d\}.$$

为了减少计算量, 假如在 M_j 中有两个不同的元素 (X_1, t_1, c_1) 和 (X_2, t_2, c_2) 且 $t_1 \leq t_2, c_1 \leq c_2$, 则元素 (X_2, t_2, c_2) 可以删去不再考虑. 证明见本节附录 1. 我们在下面的算法 G 中令 t 相同, 比较 c , 进而删减元素.

另外, 如果 (X, t, c) 是 M_j 中的元素满足 $t \leq d$ 且 c 是最小的, 则存在一个正规排序 π 满足误工工件集为 X 且 c 为加权误工时间和. 证明见本节附录 2. 因此, 如果 $c < b$ (其中 b 是 SL 的第二部分), 我们将改写 SL 为 $SL := (X, c)$, 然后从 M_j 中删除所有满足 $c' \geq c$ 的元素 (X', t', c') . 根据以上讨论可以得到下面的算法:

算法 A

输入: n 个工件 $J_1, J_2, \dots, J_n$, 加工时间和权重分别为 p_j 和 $w_j (j = 1, 2, \dots, n)$, 工期为 d . 不失一般性, 我们可以假设工件已经按 WLPT 序排好, 即 $\frac{w_1}{p_1} \leq \frac{w_2}{p_2} \leq \dots \leq \frac{w_n}{p_n}$, 否则, 我们可以在 $O(n \log n)$ 时间内得到按 WLPT 序排好的工件.

第一步 计算 $P = p_1 + p_2 + \dots + p_n$ 和 $P(j) = p_1 + p_2 + \dots + p_j (1 \leq j \leq n)$;

第二步 令 $M_0 = \{(\emptyset, P, 0)\}$, $SL = (\emptyset, +\infty)$;

第三步 对 $j = 1, 2, \dots, n$, 作下面的循环:

(a) 令 $M_j = \emptyset$;

(b) 对 M_{j-1} 的每个元素 (X, t, c) , 如果 $P(j) - (P - t) \leq d$, 则 $M_j = M_{j-1}$; 如果 $t > d$ 且 $c + w_j(t - d) < b$ (其中 b 是 SL 的第二部分), 则添加元素 $(X \cup \{J_j\}, t - p_j, c + w_j(t - d))$ 于 M_j 中;

(c) 如果 M_j 中存在第二部分相同的元素 (X_1, t, c_1) 和 (X_2, t, c_2) 满足 $c_1 \leq c_2$, 则删除 (X_2, t, c_2) , 否则删除 (X_1, t, c_1) ;

(d) 令 (X, t, c) (可以是多个) 为 M_j 中满足 $t \leq d$ 且 c 最小的元素, 如果 $c < b$ (其中 b 是 SL 的第二部分), 则令 $SL := (X, c)$, 然后从 M_j 中删除所有满足 $c' \geq c$ 的元素 (X', t', c') ;

第四步 得到最终的排序 $SL = (X, c)$.

输出: 一个将误工工件按 WSPT 序排列, 按时完工的工件按 WLPT 序排列的最优排序 π .

定理 2.3 算法 A 在 $O(nP)$ 时间内解决了问题 $1|d_j = d|\Sigma w_j T_j$, 其中 $P = \Sigma p_j$.

证 正确性可由前面的分析看出. 下面分析时间界: 注意到由于每个 M_j 中第二部分 t 不会有相同的情形, 故每个 M_j 所含元素个数不会超过 P . 而对 M_j 中的每个元素执行循环步骤 3(b) 需要时间 $O(n)$, 步骤 3(c)、3(d) 的执行和步骤 3(b) 的执行是同时进行的, 因此需要的时间也为 $O(n)$, 故总的时间界为 $O(nP)$.

为了方便下面的讨论, 我们考虑接下来的一个改进的算法.

改进算法 A'

令 U 是目标值的上界, 我们将上面算法的初始值设为 $SL = (\emptyset, U)$, 然后将第三步 (c) 改写为下面的形式 (可以由附录 1 的证明中得到):

第三步 (c) 找到 M_j 中的第三部分相同的元素 (X_1, t_1, c) 和 (X_2, t_2, c) , 如果 $t_1 \geq t_2$, 则删除 (X_1, t_1, c) , 否则删除 (X_2, t_2, c) ;

类似于定理 2.3 的证明, 我们容易得到下面的定理.

定理 2.4 改进的算法 A' 在 $O(nU)$ 时间内解决了问题 $1|d_j = d|\Sigma w_j T_j$, 其中 $U(\leq \sum_{1 \leq i \leq j \leq n} w_j p_i)$ 是目标值的上界.

注 U 值由式子 $U = f(\pi^*)$ 确定, 其中 π^* 是指由 Modified Lawler's 算法得到的排序, 因此是最优值的 $n-1$ 倍. 假设 $U > 0$, 我们得到原问题的一个 FPTAS 算法:

算法 B

输入: 一个任意的 $\varepsilon > 0$, n 个工件 $J_1, J_2, \dots, J_n$, 加工时间和权重分别为 p_j 和 w_j ($j = 1, 2, \dots, n$), 工期为 d , 且使得 $\frac{w_1}{p_1} \leq \frac{w_2}{p_2} \leq \dots \leq \frac{w_n}{p_n}$. 令 U 是由 Modified Lawler's 算法得到排序 π^* 的目标值.

第一步 计算 $K = \lceil \frac{\varepsilon U}{n^2} \rceil$, $P = p_1 + p_2 + \dots + p_n$ 和 $P(j) = p_1 + p_2 + \dots + p_j$ ($1 \leq j \leq n$);

第二步 令 $M_0 = \{(\emptyset, P, 0)\}$, $SL = (\emptyset, U)$;

第三步 对 $j = 1, 2, \dots, n$, 作下面的循环:

(a) 令 $M_j = \emptyset$;

(b) 对 M_{j-1} 的每个元素 (X, t, c) , 如果 $P(j) - (P - t) \leq d$, 则 $M_j = M_{j-1}$; 如果 $t > d$ 且 $c + K \lfloor w_j(t-d)/K \rfloor < b$ (其中 b 是 SL 的第二部分), 则添加元素 $(X \cup \{J_j\}, t - p_j, c + K \lfloor w_j(t-d)/K \rfloor)$ 于 M_j 中; 找到 M_j 中的第三部分相同的元素 (X_1, t_1, c) 和 (X_2, t_2, c) , 如果 $t_1 \geq t_2$, 则删除 (X_1, t_1, c) , 否则删除 (X_2, t_2, c) ; (X, t, c) (可以是多个) 为 M_j 中满足 $t \leq d$ 且 c 是最小的元素, 如果 $c < b$ (其中 b 是 SL 的第二部分), 则令 $SL := (X, c)$, 然后从 M_j 中删除所有满足 $c' \geq c$ 的元素 (X', t', c') ;

第四步 得到最终的排序 $SL = (\bar{X}, \bar{c})$.

该算法与前面算法的重要区别是每个 M_j 最多有 $O(U/K)$ 个元素. 并且每个工件 J_j 的加权误工时间被统一成 $K \lfloor w_j T_j / K \rfloor$. 下面的定理说明算法 B 是一个 FPTAS.

定理 2.5 对任给的 $\varepsilon > 0$, 算法 H 对问题 $1|d_j = d|\Sigma w_j T_j$ 是一个 $(1 + \varepsilon)$ -近似算法, 时间复杂性为 $O(n^3/\varepsilon)$.

证 令 c 是原问题 $1|d_j = d|\Sigma w_j T_j$ 的最优的目标值, $(\bar{X}, \bar{c})$ 为算法 H 得到的最终的 SL 的值且 σ 为对应的排序. 由于 $\bar{c}$ 实际上是问题 $1|d_j = d, \pi$ 是正规排序 $|\Sigma K \lfloor w_j T_j / K \rfloor$ 的最优的目标值, 因此我们有 $c \geq \bar{c}$. 进一步, 令 c^* 是算法 H 得到的原问题的近似值, 即排序 σ 下的加权误工时间总和. 由 c 的最优性, 我们有 $c^* \geq c$ 且 $U \leq (n-1)c < nc$. 我们记 $c^* = f(\sigma) = \Sigma w_j T_j$, 则 $\bar{c} = \Sigma K \lfloor w_j T_j / K \rfloor$. 由于

$$w_j T_j - K \lfloor w_j T_j / K \rfloor \leq K - 1 \leq \varepsilon U / n^2 \leq \varepsilon c / n,$$

我们有

$$c^* - c \leq c^* - \bar{c} = \Sigma (w_j T_j - K \lfloor w_j T_j / K \rfloor) \leq \varepsilon c.$$

这就意味着 $c^* \leq (1 + \varepsilon)c$, 故得到的是原问题的一个 FPTAS 算法. 又由于每个 M_j 最多有 $O(U/K)$ 个元素, 因此最多需要 $(nU/K) = O(n^3/\varepsilon)$ 时间.

3 结论

本文研究的是具有相同工期的单机最小化加权总误工这一经典排序问题, 已知该问题 $1|d_j = d|\Sigma w_j T_j$ 是一般意义下的 NP - 困难问题, 本文利用问题 $1|\Sigma w_j T_j$ 的一个 $O(n^2)$ 时间算法得到的目标值作为我们研究问题的上界, 给出了该问题的全多项式近似方案.

参 考 文 献

- [1] Lawler E L. A pseudo polynomial algorithm for sequencing jobs to minimize total tardiness[J]. Annals of Discrete Math., 1997(1): 331-342.
- [2] Lenstra J K, Rinnooy Kan A H G, Brucker P. Complexity of machine scheduling problems[J]. Annals of Discrete Math., 1977(1): 343-362.
- [3] Potts C, Van Wassenhove L. A branch and bound algorithm for the total weighted tardiness problem[J]. Operations Research, 1985(33): 363-377.
- [4] Schrage L, Baker K L. Dynamic programming solution of sequencing problems with precedence constraints[J]. Operations Research, 1978(26): 444-449.
- [5] Cheng T C E, Ng C T, Yuan J J, Liu Z H. Single machine scheduling to minimize total weighted tardiness[J]. European J. Operational Research, 2005(165): 423-443.
- [6] Du J Z, Leung J Y T. Minimizing total tardiness on one machine is NP-hard[J]. Math. Operations Research, 1990(15): 483-495.
- [7] Arkin E M, Roundy R O. Weighted-tardiness scheduling on parallel machines with proportional weights[J]. Operations Research, 1991(39): 64-81.
- [8] Yuan J J. The NP-hardness of the single machine common due date weighted tardiness problem[J]. Systems Science and Math. Sciences, 1992(5): 328-333.
- [9] Lawler E L, Moore J M. A functional equation and its application to resource allocation and sequencing problems[J]. Management Science, 1969(16): 77-84.

MINIMIZING THE SUM OF THE TOTAL COMPLETION TIME AND BATCHING COSTS ON BATCH PROCESSING MACHINE

ZHANG Zhe¹, LI Wen-hua²

(1. College of Science, Zhongyuan University of Technology, Zhengzhou 450007, China)

(2. Department of Mathematics, Zhengzhou University, Zhengzhou 450001, China)

Abstract: The article is about the single machine scheduling problem with common due date to minimize total weighted tardiness. It's also known that the problem $1|\sum w_j T_j$ is strongly NP-hard and there is a $O(n^2)$ time approximation algorithm for this problem. The article took the objective value of this algorithm as the upper bound of our problem $1|d_j = d|\sum w_j T_j$ and gave the problem $1|d_j = d|\sum w_j T_j$ a full polynomial time approximation scheme (FPTAS). For the problem of $1|d_j = d|\sum w_j T_j$, it's known that it is NP-hard in the ordinary sense. And it has been provided a pseudo-polynomial algorithm. The results have been expanded by this article.

Keywords: common due date; total weighted tardiness; FPTAS

2010 MR Subject Classification: 90C27

附 录

1. 如果在 M_j 中存在两个不同的元素 (X_1, t_1, c_1) 和 (X_2, t_2, c_2) 且 $t_1 \leq t_2, c_1 \leq c_2$, 则元素 (X_2, t_2, c_2) 可以删去不再考虑.

证 令 $\{\sigma_k\}$ 是对应 $(X_k, t_k, c_k) \in M_j, (k = 1, 2)$ 的正规排序, 令 π_2 是 $\{\sigma_2\}$ 中的任一排序. 对每一个工件 $J \in \{J_{j+1}, J_{j+2}, \dots, J_n\}$, 设 $s(J, \pi_2)$ 是 π_2 下工件 J 的开始时间. 由于 σ_k 下 $\{J_{j+1}, J_{j+2}, \dots, J_n\}$ 中的工件加工时间应该在时间 $P(j) - P + t_k$ 和 t_k 之间 ($k = 1, 2$), 因此, 一定存在一个排序 $\pi_1 \in \{\sigma_1\}$ 使得对每一个工件 $J \in \{J_{j+1}, J_{j+2}, \dots, J_n\}$ 的开始时间为 $s(J, \pi_2) - (t_2 - t_1) \leq s(J, \pi_1)$, 故工件 J 在 π_1 下的误工时间小于或等于在 π_2 下的误工时间. 又因为已知 $c_1 \leq c_2$, 故 π_1 下的总加权误工时间和小于或等于 π_2 下的总加权误工时间和, 所以 (X_2, t_2, c_2) 可以删除掉.

2. 如果 (X, t, c) 是 M_j 中的元素满足 $t \leq d$ 且 c 是最小的, 则存在一个正规排序满足误工工件集为 X 且 c 为加权误工时间和.

证 令 π 是对应于 (X, t, c) 的一个正规排序, 在 π 中保持 X 中工件不变, 因为 $t \leq d$, 所以 $\mathcal{J}_n \setminus X$ 中工件无论如何排列都不会误工, 故存在一个正规排序 π 满足误工工件集为 X 且 c 为加权误工时间和.