运筹学与控制论

DOI:10.11721/cqnuj20160207

带有分段线性递减加工时间和拒绝工件的单机排序问题*

隋 敏，赵传立
(沈阳师范大学 数学与系统科学学院，沈阳 110034)

摘要:讨论了带有分段线性递减加工时间和拒绝工件的单机排序问题。在这一模型中,工件的实际加工时间是关于开始时间的分段线性递减函数,目标函数是极小化被接受工件的最大完工时间和被拒绝工件的总惩罚之和。这一问题是 NP-难的。基于对问题的分析,给出了一个全多项式近似策略。全多项式近似策略的计算复杂性为 $O(n^4L^4/\varepsilon^3)$。

关键词:单机排序;分段线性递减;拒绝;全多项式近似策略

中图分类号:O223 **文献标志码**:A **文章编号**:1672-6693(2016)02-0015-05

在经典的排序理论中,加工时间为固定的常数[1]。Brown,Yechiali[2] 和 Gupta,Gupta[3] 最早提出了工件实际加工时间依赖开始时间的排序模型。随后,加工时间依赖开始时间的排序问题受到了越来越多的关注[4-7]。Kang 和 Ng[8] 研究了工件加工时间关于开始时间的线性函数的平行机问题,目标函数是极小化最大完工时间,给出了全多项式近似策略。Ji 和 Cheng[9] 对工件加工时间依赖开始时间的分段线性递减函数的单机排序问题做了研究,目标是极小化最大完工时间,给出了全多项式近似策略。Luo 和 Ji[10] 讨论了带有变量维修和线性退化工件的单机排序模型,目标是极小化最大完工时间,证明了问题是 NP 难的。近年来带有加工时间依赖开始时间的线性函数和工件可拒绝的排序模型受到了广泛关注[11-12]。对于带有退化工件、释放时间和拒绝工件的单机排序问题,Liu,Zheng,Chu 和 Xu[13] 给出了动态规划和全多项式近似策略。Li 和 Yuan[14] 提出了加工时间是关于开始时间的线性递增函数和工件可拒绝的平行机排序问题,目标是极小化被接受工件的总费用加上被拒绝工件的总惩罚费用之和,给出了全多项式近似策略。

本文讨论了带有分段线性递减加工时间和拒绝工件的单机排序问题,在文献[9]基础上增加了拒绝。因为文献[9]中的问题是 NP 难的,所以本文讨论问题也是 NP 难的。基于对问题的分析,给出了一个全多项式近似策略。

1 问题描述

设有 n 个工件的集合 $J=\{J_1,J_2,\cdots,J_n\}$ 在一台机器上加工。全部工件零时刻到达,且机器在同一时间只能加工一个工件。工件 J_j 的实际加工时间为 $p_j=a_j-b_j\min\{s_j,D\}$,a_j 表示工件 J_j 的正常加工时间,$0<b_j<1$ 表示工件 J_j 的递减率,s_j 表示工件 J_j 的开始时间,D 表示公共递减工期,$w_j>0$ 表示工件 J_j 被拒绝的费用,其中 $a_j>b_j\min\left\{D,\sum\limits_{i=1}^{n}a_i-a_j\right\}$,$A$ 表示被接受工件集,R 表示被拒绝工件集。在 $b_j\leqslant a_j/(2D)$ 前提下,文献[9]给出了一个全多项近似策略,所以本文也在 $b_j\leqslant a_j/(2D)$ 条件下进行讨论。运用三参数表示法,本文所考虑的问题可表示:$1|p_j=a_j-b_j\min\{s_j,D\},rej|C_{\max\limits_{J_j\in A}}+\sum\limits_{J_j\in R}w_j$。

2 全多项式近似策略

由文献[9]知,本文给出如下性质。

---

* **收稿日期**:2015-06-15 **修回日期**:2015-11-02 **网络出版时间**:2016-1-20 21:26

资助项目:辽宁省教育厅科学研究基金(No. L2014433)

作者简介:隋敏,女,研究方向为排序理论,E-mail:1035427834@qq.com;**通信作者**:赵传立,教授,E-mail:zhaochuanli@synu.edu.com

网络出版地址:http://www.cnki.net/kcms/detail/50.1165.n.20160120.2126.050.html

性质1 对于问题 $1|p_j=a_j-b_j\min\{s_j,D\},rej|C_{\max\limits_{J_j\in A}}+\sum\limits_{J_j\in R}w_j$，存在一个最优排序使得开始时间小于 D 的被接受工件按 $a_1/b_1\geqslant a_2/b_2\geqslant\cdots\geqslant a_n/b_n$ 排序。

性质2 对于问题 $1|p_j=a_j-b_j\min\{s_j,D\},rej|C_{\max\limits_{J_j\in A}}+\sum\limits_{J_j\in R}w_j$，存在一个最优排序使得开始时间不小于 D 的被接受工件按任意顺序排序。

性质3 对于问题 $1|p_j=a_j-b_j\min\{s_j,D\},rej|C_{\max\limits_{J_j\in A}}+\sum\limits_{J_j\in R}w_j$，在最优排序中机器无空闲时间。

根据性质1所有被接受工件按 $a_1/b_1\geqslant a_2/b_2\geqslant\cdots\geqslant a_n/b_n$ 排序，定义变量 x_j，$j=1,2,\cdots,n$，其中 $x_j=\begin{cases}0,\text{被接受工件 } J_j \text{ 的开始时间不小于 } D\\1,\text{被接受工件 } J_j \text{ 的开始时间小于 } D\\2,\text{工件 } J_j \text{ 被拒绝}\end{cases}$。记 X 为所有向量 $\boldsymbol{x}=(x_1,x_2,\cdots,x_n)$ 所组成的向量集，其中 $x_j=k$，$k=0,1,2$。记 $f_j(\boldsymbol{x})$ 表示目标函数值，$h_j(\boldsymbol{x})$ 表示 min{开始时间小于 D 的工件加工时间总和，D}，$W_j(\boldsymbol{x})$ 表示工件 $J_1,J_2,\cdots,J_j$ 拒绝总费用。定义在集合 X 上的函数如下：

$$W_0(\boldsymbol{x})=0;h_0(\boldsymbol{x})=0;f_0(\boldsymbol{x})=0;W_j(\boldsymbol{x})=W_{j-1}(\boldsymbol{x})+\frac{1}{2}x_j(x_j-1)w_j;$$

$$h_j(\boldsymbol{x})=\min\{h_{j-1}(\boldsymbol{x})+x_j(2-x_j)(a_j-b_jh_{j-1}(\boldsymbol{x})),D\};$$

$$f_j(\boldsymbol{x})=f_{j-1}(\boldsymbol{x})+x_j(2-x_j)(a_j-b_jh_{j-1}(\boldsymbol{x}))+\frac{1}{2}(1-x_j)(2-x_j)(a_j-b_jD)+\frac{1}{2}x_j(x_j-1)w_j。$$

显然问题 $1|p_j=a_j-b_j\min\{s_j,D\},rej|C_{\max\limits_{J_j\in A}}+\sum\limits_{J_j\in R}w_j$ 可归结为：$\min f_n(\boldsymbol{x}),\boldsymbol{x}\in X$。

首先利用文献[15-16]给出划分程序 (A,e,δ)，其中 $A\subseteq X$，e 是定义在集合 X 上的非负整数函数，$0<\delta\leqslant1$。将 A 分割为两两不相交的子集 $A_1^e,A_2^e,\cdots,A_{k_e}^e$，使得对于 $\forall\,\boldsymbol{x},\boldsymbol{x}'\in A_j^e$，都有 $|e(\boldsymbol{x})-e(\boldsymbol{x}')|\leqslant\delta\min\{e(\boldsymbol{x}),e(\boldsymbol{x}')\}$ $(j=1,2,\cdots,k_e)$。

下面对划分程序进行描述：第1步，将 A 中向量 $\boldsymbol{x}$ 进行标号 $\boldsymbol{x}^{(1)},\boldsymbol{x}^{(2)},\cdots,\boldsymbol{x}^{(|A|)}$ 使得 $0\leqslant e(\boldsymbol{x}^{(1)})\leqslant e(\boldsymbol{x}^{(2)})\leqslant\cdots\leqslant e(\boldsymbol{x}^{(|A|)})$。第2步，将向量 $\boldsymbol{x}^{(1)},\boldsymbol{x}^{(2)},\cdots,\boldsymbol{x}^{(i_1)}$ 按顺序逐个加入集合 A_1^e 中，直到找到 i_1 使得 $e(\boldsymbol{x}^{(i_1)})\leqslant(1+\delta)e(\boldsymbol{x}^{(1)})$ 和 $e(\boldsymbol{x}^{(i_1+1)})>(1+\delta)e(\boldsymbol{x}^{(1)})$ 成立。如果不存在这样的 i_1，那么令 $A_{k_e}^e=A_1^e=A$ 且划分停止。

将向量 $\boldsymbol{x}^{(i_1+1)},\boldsymbol{x}^{(i_1+2)},\cdots,\boldsymbol{x}^{(i_2)}$ 按顺序逐个加入集合 A_2^e 中，直到找到 i_2 使得 $e(\boldsymbol{x}^{(i_2)})\leqslant(1+\delta)e(\boldsymbol{x}^{(i_1+1)})$ 和 $e(\boldsymbol{x}^{(i_2+1)})>(1+\delta)e(\boldsymbol{x}^{(i_1+1)})$ 成立。如果不存在这样的 i_2，那么令 $A_{k_e}^e=A_2^e=A-A_1^e$ 且划分停止。

重复以上步骤直到找到 k_e 使得 $\boldsymbol{x}^{(|A|)}\in A_{k_e}^e$ 为止。

划分程序 (A,e,δ) 在运行时间 $o(|A|\log|A|)$ 内将 A 中向量按 $e(\boldsymbol{x})$ 非减顺序排序，在运行时间 $o(|A|)$ 内提供划分。文献[12-13]提出划分的主要性质将其应用到算法 A_ε 中。

性质4 对于 $\forall\,\boldsymbol{x},\boldsymbol{x}'\in A_j^e$，$j=1,2,\cdots,k_e$，则有 $|e(\boldsymbol{x})-e(\boldsymbol{x}')|\leqslant\delta\min\{e(\boldsymbol{x}),e(\boldsymbol{x}')\}$ $(j=1,2,\cdots,k_e)$。

性质5 对于 $0<\delta\leqslant1$ 且 $1\leqslant e(\boldsymbol{x}^{(|A|)})$，则有 $k_e\leqslant\log e(\boldsymbol{x}^{(|A|)})/\delta+2$。

对于问题 $1|p_j=a_j-b_j\min\{s_j,D\},rej|C_{\max\limits_{J_j\in A}}+\sum\limits_{J_j\in R}w_j$ 给出一个FPTAS。

算法 A_ε 第1步，初始化。将工件按 $a_1/b_1\geqslant a_2/b_2\geqslant\cdots\geqslant a_n/b_n$ 顺序排列。在没考虑任何工件的初始状态，令 $Y_0=\{(0,0,\cdots,0)\}$，$j=1$。

第2步，构造集合 $Y_1,Y_2,\cdots,Y_n$。通过对集合 Y_{j-1} 中每个向量的第 j 位置加 k $(k=0,1,2)$ 产生新的3个向量来组成 Y_j'。对于 $\forall\,\boldsymbol{x}'\in Y_j'$，假设 $x_j=k$，计算：

$$W_j(\boldsymbol{x})=W_{j-1}(\boldsymbol{x})+\frac{1}{2}x_j(x_j-1)w_j,h_j(\boldsymbol{x})=\min\{h_{j-1}(\boldsymbol{x})+x_j(2-x_j)(a_j-b_jh_{j-1}(\boldsymbol{x})),D\},$$

$$f_j(\boldsymbol{x})=f_{j-1}(\boldsymbol{x})+x_j(2-x_j)(a_j-b_jh_{j-1}(\boldsymbol{x}))+\frac{1}{2}(1-x_j)(2-x_j)(a_j-b_jD)+\frac{1}{2}x_j(x_j-1)w_j。$$

如果 $j=n$，那么令 $Y_n=Y_n'$ 且进行第3步。

如果 $j<n$，那么令 $\delta=\varepsilon/(2(n+1))$。调用划分程序 (Y_j',W_j,δ) 将集合 Y_j' 分割成两两不相交的子集 $Y_1^W,Y_2^W,\cdots,Y_{k_W}^W$。调用划分程序 (Y_j',h_j,δ) 将集合 Y_j' 分割成两两不相交的子集 $Y_1^h,Y_2^h,\cdots,Y_{k_h}^h$。调用划分程序 (Y_j',f_j,δ)

将集合 Y_j' 分割成两两不相交的子集 $Y_1^f, Y_2^f, \cdots, Y_{k_f}^f$。

将集合 Y_j' 划分成两两不相交的非空子集，即 $Y_{abc}=Y_a^W \cap Y_b^h \cap Y_c^f, a=1,2,\cdots,k_W, b=1,2,\cdots,k_h, c=1,2,\cdots,k_f$。在每个非空子集 Y_{abc} 选出向量 $\boldsymbol{x}^{(abc)}$，使得 $f_j(\boldsymbol{x}^{(abc)})=\min\{f_j(\boldsymbol{x}) \mid \boldsymbol{x}\in Y_{abc}\}$。令 $Y_j:=\{\boldsymbol{x}^{(abc)} \mid Y_a^W \cap Y_b^h \cap Y_c^f \neq \varnothing, a=1,2,\cdots,k_W; b=1,2,\cdots,k_h; c=1,2,\cdots,k_f\}$ 且令 $j=j+1$，转第 2 步。

第 3 步，选择向量 $\boldsymbol{x}^0 \in Y_n$ 使得 $f_n(\boldsymbol{x}^0)=\min\{f_n(\boldsymbol{x}) \mid \boldsymbol{x}\in Y_n\}$。令 $\boldsymbol{x}^*=(x_1^*, x_2^*, \cdots, x_n^*)$ 为问题 $1 \mid p_j=a_j-b_j\min\{s_j,D\}, rej \mid C_{\max\limits_{J_j\in A}}+\sum\limits_{J_j\in R} w_j$ 的最优解。

定义 $L=\log(\max\{n,1/\varepsilon,a_{\max},W\})$，其中 $a_{\max}=\max\limits_{j=1,2,\cdots,n}\{a_j\}$，$W=\sum\limits_{j=1}^{n} w_j$。有如下定理。

定理 1　对于问题 $1 \mid p_j=a_j-b_j\min\{s_j,D\}, rej \mid C_{\max\limits_{J_j\in A}}+\sum\limits_{J_j\in R} w_j$，其中 $j=1,\cdots,n$，当 $b_j\leqslant a_j/(2D)$ 时，利用算法 A_ε 可以在运行时间 $O(n^4L^4/\varepsilon^3)$ 内找到一个向量 $\boldsymbol{x}^0\in X$，使得 $f_n(\boldsymbol{x}^0)\leqslant(1+\varepsilon)f_n(\boldsymbol{x}^*)$ 成立。

证明　假设 $(x_1^*,\cdots,x_j^*,0,\cdots,0)\in Y_{abc}\subseteq Y'_j$，对于 j 和 a,b,c，根据算法 A_ε 的总是存在这样的 j，例如 $j=1$。在算法 A_ε 进行进一步消减 $(x_1^*,\cdots,x_j^*,0,\cdots,0)$ 可能不被选出，然而一定存在一个向量 $\boldsymbol{x}^{(abc)}$ 被选出代替 $(x_1^*,\cdots,x_j^*,0,\cdots,0)$。

根据性质 4，有：

$$|W_j(\boldsymbol{x}^*)-W_j(\boldsymbol{x}^{(abc)})|\leqslant\delta\min\{W_j(\boldsymbol{x}^*),W_j(\boldsymbol{x}^{(abc)})\}\leqslant\delta W_j(\boldsymbol{x}^*),\ |h_j(\boldsymbol{x}^*)-h_j(\boldsymbol{x}^{(abc)})|\leqslant \delta\min\{h_j(\boldsymbol{x}^*),h_j(\boldsymbol{x}^{(abc)})\}\leqslant\delta h_j(\boldsymbol{x}^*),\ |f_j(\boldsymbol{x}^*)-f_j(\boldsymbol{x}^{(abc)})|\leqslant\delta\min\{f_j(\boldsymbol{x}^*),f_j(\boldsymbol{x}^{(abc)})\}\leqslant\delta f_j(\boldsymbol{x}^*)。$$

令 $\delta_1=\delta$，考虑向量 $(x_1^*,\cdots,x_j^*,x_{j+1}^*,0,\cdots,0)$ 和 $\tilde{\boldsymbol{x}}^{(abc)}=(x_1^{(abc)},\cdots,x_j^{(abc)},x_{j+1}^*,0,\cdots,0)$。

不失一般性，假设 $x_{j+1}^*=k$，则有：

$$|W_{j+1}(\boldsymbol{x}^*)-W_{j+1}(\tilde{\boldsymbol{x}}^{(abc)})|=\left|\left[W_j(\boldsymbol{x}^*)+\frac{1}{2}x_{j+1}^*(x_{j+1}^*-1)w_{j+1}\right]-\left[W_j(\boldsymbol{x}^{(abc)})+\frac{1}{2}x_{j+1}^*(x_{j+1}^*-1)w_{j+1}\right]\right|\leqslant\delta W_j(\boldsymbol{x}^*)\leqslant\delta_1 W_{j+1}(\boldsymbol{x}^*) \tag{1}$$

若 $D\leqslant\min\{h_j(\boldsymbol{x}^*)+x_{j+1}^*(2-x_{j+1}^*)(a_{j+1}-b_{j+1}h_j(x^*)), h_j(\boldsymbol{x}^{(abc)})+x_{j+1}^*(2-x_{j+1}^*)(a_{j+1}-b_{j+1}h_j(\boldsymbol{x}^{(abc)}))\}$，则

$$|h_{j+1}(\boldsymbol{x}^*)-h_{j+1}(\tilde{\boldsymbol{x}}^{(abc)})|=|D-D|\leqslant\delta_1 h_{j+1}(\boldsymbol{x}^*) \tag{2}$$

若 $D\geqslant\max\{h_j(\boldsymbol{x}^*)+x_{j+1}^*(2-x_{j+1}^*)(a_{j+1}-b_{j+1}h_j(\boldsymbol{x}^*)), h_j(\boldsymbol{x}^{(abc)})+x_{j+1}^*(2-x_{j+1}^*)(a_{j+1}-b_{j+1}h_j(\boldsymbol{x}^{(abc)}))\}$，则

$$|h_{j+1}(\boldsymbol{x}^*)-h_{j+1}(\tilde{\boldsymbol{x}}^{(abc)})|=|[h_j(\boldsymbol{x}^*)+x_{j+1}^*(2-x_{j+1}^*)(a_{j+1}-b_{j+1}h_j(\boldsymbol{x}^*))]-[h_j(\boldsymbol{x}^{(abc)})+x_{j+1}^*(2-x_{j+1}^*)(a_{j+1}-b_{j+1}h_j(\boldsymbol{x}^{(abc)}))]|\leqslant(1-x_{j+1}^*(2-x_{j+1}^*)b_{j+1})|h_j(\boldsymbol{x}^*)-h_j(\boldsymbol{x}^{(abc)})|\leqslant \delta(1-x_{j+1}^*(2-x_{j+1}^*)b_{j+1})h_j(\boldsymbol{x}^*)\leqslant\delta_1 h_{j+1}(\boldsymbol{x}^*) \tag{3}$$

若

$[h_j(\boldsymbol{x}^*)+x_{j+1}^*(2-x_{j+1}^*)(a_{j+1}-b_{j+1}h_j(\boldsymbol{x}^*))-D]\,[h_j(\boldsymbol{x}^{(abc)})+x_{j+1}^*(2-x_{j+1}^*)(a_{j+1}-b_{j+1}h_j(\boldsymbol{x}^{(abc)}))-D]\leqslant 0$，

则

$$|h_{j+1}(\boldsymbol{x}^*)-h_{j+1}(\tilde{\boldsymbol{x}}^{(abc)})|\leqslant|[h_j(\boldsymbol{x}^*)+x_{j+1}^*(2-x_{j+1}^*)(a_{j+1}-b_{j+1}h_j(\boldsymbol{x}^*))]-[h_j(\boldsymbol{x}^{(abc)})+x_{j+1}^*(2-x_{j+1}^*)(a_{j+1}-b_{j+1}h_j(\boldsymbol{x}^{(abc)}))]|\leqslant(1-x_{j+1}^*(2-x_{j+1}^*)b_{j+1})|h_j(\boldsymbol{x}^*)-h_j(\boldsymbol{x}^{(abc)})|\leqslant \delta(1-x_{j+1}^*(2-x_{j+1}^*)b_{j+1})h_j(\boldsymbol{x}^*)\leqslant\delta_1 h_{j+1}(\boldsymbol{x}^*) \tag{4}$$

因为对于所有 k，$h_j(\boldsymbol{x}^*)\leqslant D$ 和 $b_j\leqslant a_j/(2D)$ 成立，所以有 $b_{j+1}h_j(\boldsymbol{x}^*)\leqslant a_{j+1}-b_{j+1}h_j(\boldsymbol{x}^*)$。

$$|f_{j+1}(\boldsymbol{x}^*)-f_{j+1}(\tilde{\boldsymbol{x}}^{(abc)})|=\left|\left[f_j(\boldsymbol{x}^*)+x_{j+1}^*(2-x_{j+1}^*)(a_{j+1}-b_{j+1}h_j(\boldsymbol{x}^*))+\frac{1}{2}(1-x_{j+1}^*)(2-x_{j+1}^*)(a_{j+1}-b_{j+1}D)+\frac{1}{2}x_{j+1}^*(x_{j+1}^*-1)w_{j+1}\right]-\left[f_j(\boldsymbol{x}^{(abc)})+x_{j+1}^*(2-x_{j+1}^*)(a_{j+1}-b_{j+1}h_j(\boldsymbol{x}^{(abc)}))+\frac{1}{2}(1-x_{j+1}^*)(2-x_{j+1}^*)(a_{j+1}-b_{j+1}D)+\frac{1}{2}x_{j+1}^*(x_{j+1}^*-1)w_{j+1}\right]\right|\leqslant |f_j(\boldsymbol{x}^*)-f_j(\boldsymbol{x}^{(abc)})|+x_{j+1}^*(2-x_{j+1}^*)b_{j+1}|h_j(\boldsymbol{x}^*)-h_j(\boldsymbol{x}^{(abc)})|\leqslant \delta f_j(\boldsymbol{x}^*)+\delta x_{j+1}^*(2-x_{j+1}^*)b_{j+1}h_j(\boldsymbol{x}^*)\leqslant\delta f_j(\boldsymbol{x}^*)+\delta x_{j+1}^*(2-x_{j+1}^*)(a_{j+1}-b_{j+1}h_j(\boldsymbol{x}^*))\leqslant\delta_1 f_{j+1}(\boldsymbol{x}^*) \tag{5}$$

由(1)～(5)式可得：

$$\begin{cases}|W_{j+1}(\boldsymbol{x}^*)-W_{j+1}(\tilde{\boldsymbol{x}}^{(abc)})|\leqslant\delta_1W_{j+1}(\boldsymbol{x}^*)\\|h_{j+1}(\boldsymbol{x}^*)-h_{j+1}(\tilde{\boldsymbol{x}}^{(abc)})|\leqslant\delta_1h_{j+1}(\boldsymbol{x}^*)\\|f_{j+1}(\boldsymbol{x}^*)-f_{j+1}(\tilde{\boldsymbol{x}}^{(abc)})|\leqslant\delta_1f_{j+1}(\boldsymbol{x}^*)\end{cases}\tag{6}$$

这样导出：

$$\begin{cases}W_{j+1}(\tilde{\boldsymbol{x}}^{(abc)})\leqslant(1+\delta_1)W_{j+1}(\boldsymbol{x}^*)\\h_{j+1}(\tilde{\boldsymbol{x}}^{(abc)})\leqslant(1+\delta_1)h_{j+1}(\boldsymbol{x}^*)\\f_{j+1}(\tilde{\boldsymbol{x}}^{(abc)})\leqslant(1+\delta_1)f_{j+1}(\boldsymbol{x}^*)\end{cases}\tag{7}$$

假设 $\tilde{\boldsymbol{x}}^{(abc)}\in Y_{dem}\subseteq Y'_{j+1}$ 和利用算法 A_ε 在第 $j+1$ 次迭代时选出向量 $\boldsymbol{x}^{(dem)}\in Y_{dem}$ 代替向量 $\tilde{\boldsymbol{x}}^{(abc)}$，可以导出：

$$\begin{cases}|W_{j+1}(\tilde{\boldsymbol{x}}^{(abc)})-W_{j+1}(\boldsymbol{x}^{(dem)})|\leqslant\delta W_{j+1}(\tilde{\boldsymbol{x}}^{(abc)})\leqslant\delta(1+\delta_1)W_{j+1}(\boldsymbol{x}^*)\\|h_{j+1}(\tilde{\boldsymbol{x}}^{(abc)})-h_{j+1}(\boldsymbol{x}^{(dem)})|\leqslant\delta h_{j+1}(\tilde{\boldsymbol{x}}^{(abc)})\leqslant\delta(1+\delta_1)h_{j+1}(\boldsymbol{x}^*)\\|f_{j+1}(\tilde{\boldsymbol{x}}^{(abc)})-f_{j+1}(\boldsymbol{x}^{(dem)})|\leqslant\delta f_{j+1}(\tilde{\boldsymbol{x}}^{(abc)})\leqslant\delta(1+\delta_1)f_{j+1}(\boldsymbol{x}^*)\end{cases}\tag{8}$$

由(1)式和(8)式有：

$$|W_{j+1}(\boldsymbol{x}^*)-W_{j+1}(\boldsymbol{x}^{(dem)})|\leqslant|W_{j+1}(\boldsymbol{x}^*)-W_{j+1}(\tilde{\boldsymbol{x}}^{(abc)})|+|W_{j+1}(\tilde{\boldsymbol{x}}^{(abc)})-W_{j+1}(\boldsymbol{x}^{(dem)})|\leqslant$$
$$[\delta_1+(1+\delta_1)\delta]W_{j+1}(\boldsymbol{x}^*)=[\delta+(1+\delta)\delta_1]W_{j+1}(\boldsymbol{x}^*)。\tag{9}$$

类似地，$|h_{j+1}(\boldsymbol{x}^*)-h_{j+1}(\boldsymbol{x}^{(dem)})|\leqslant[\delta+(1+\delta)\delta_1]h_{j+1}(\boldsymbol{x}^*)$ $|f_{j+1}(\boldsymbol{x}^*)-f_{j+1}(\boldsymbol{x}^{(dem)})|\leqslant[\delta+(1+\delta)\delta_1]f_{j+1}(\boldsymbol{x}^*)$。令 $\delta_l=\delta+(1+\delta)\delta_{l-1}$，$l=2,3,\cdots,n-j+1$。由(9)式可得 $|W_{j+1}(\boldsymbol{x}^*)-W_{j+1}(\boldsymbol{x}^{(dem)})|\leqslant\delta_2W_{j+1}(\boldsymbol{x}^*)$。

类似地，$|h_{j+1}(\boldsymbol{x}^*)-h_{j+1}(\boldsymbol{x}^{(dem)})|\leqslant\delta_2h_{j+1}(\boldsymbol{x}^*)$；$|f_{j+1}(\boldsymbol{x}^*)-f_{j+1}(\boldsymbol{x}^{(dem)})|\leqslant\delta_2f_{j+1}(\boldsymbol{x}^*)$。

对于 $j+2,\cdots,n$，重复论证，存在 $\boldsymbol{x}'\in Y_n$ 使得 $|W_n(\boldsymbol{x}^*)-W_n(\boldsymbol{x}')|\leqslant\delta_{n-j+1}W_n(\boldsymbol{x}^*)$，$|h_n(\boldsymbol{x}^*)-h_n(\boldsymbol{x}')|\leqslant\delta_{n-j+1}h_n(\boldsymbol{x}^*)$，$|f_n(\boldsymbol{x}^*)-f_n(\boldsymbol{x}')|\leqslant\delta_{n-j+1}f_n(\boldsymbol{x}^*)$成立。

因为

$$\delta_{n-j+1}\leqslant\delta\sum_{j=0}^{n}(1+\delta)^j=(1+\delta)^{n+1}-1=\sum_{j=1}^{n+1}\frac{(n+1)n\cdots(n-j+2)}{j!}\delta^j=$$
$$\sum_{j=1}^{n+1}\frac{(n+1)n\cdots(n-k+2)}{j!\ (n+1)^j}\left(\frac{\varepsilon}{2}\right)^j\leqslant\sum_{j=1}^{n+1}\frac{1}{j!}\left(\frac{\varepsilon}{2}\right)^j\leqslant\sum_{j=1}^{n+1}\left(\frac{\varepsilon}{2}\right)^j\leqslant\varepsilon,$$

所以有 $|W_n(\boldsymbol{x}^*)-W_n(\boldsymbol{x}')|\leqslant\varepsilon W_n(\boldsymbol{x}^*)$，$|h^n(\boldsymbol{x}^*)-h_n(\boldsymbol{x}')|\leqslant\varepsilon h_n(\boldsymbol{x}^*)$，$|f_n(\boldsymbol{x}^*)-f_n(\boldsymbol{x}')|\leqslant\varepsilon f_n(\boldsymbol{x}^*)$。因此通过算法 A_ε 的第 3 步，得到向量 x^0 使得 $f_n(\boldsymbol{x}^0)\leqslant f_n(\boldsymbol{x}')\leqslant(1+\varepsilon)f_n(\boldsymbol{x}^*)$。

下面讨论算法 A_ε 的时间复杂性。由 $|Y'_j|\leqslant3|Y_j|\leqslant3k_W\cdot k_h\cdot k_f$ 和性质 5 知：

$$k_W\leqslant\frac{2(n+1)\log W}{\varepsilon}+2\leqslant\frac{2(n+1)L}{\varepsilon}+2,$$
$$k_h\leqslant\frac{2(n+1)\log(na_{\max})}{\varepsilon}+2\leqslant\frac{2(n+1)L}{\varepsilon}+2,$$
$$k_f\leqslant\frac{2(n+1)\log(na_{\max}+W)}{\varepsilon}+2\leqslant\frac{2(n+1)L}{\varepsilon}+2。$$

因此由上述不等式可推出 $|Y'_j|=O(n^3L^3/\varepsilon^3)$，$O(|Y_j{}'|\log|Y_j{}'|)=O(n^3L^4/\varepsilon^3)$。算法 A_ε 的时间复杂性为 $O(n^4L^4/\varepsilon^3)$。由定理 1 可知，算法 A_ε 是一个全多项式近似策略。 证毕

3 结论

本文考虑了工件实际加工时间是关于开始时间的分段线性递减函数和拒绝工件的排序问题，目标函数是极小化被接受工件的最大完工时间和被拒绝工件的总惩罚之和。针对单机情形给出了一个全多项式近似策略，可以将本文所研究问题的结果推广到平行机。在以后的研究中也可以考虑不同的目标函数和其他机器环境的问题。

参考文献：

[1] Pinedo M. Scheduling: theory, algorithms, and systems[M]. NJ: Prentice-Hall, Upper Saddle River, 2002.

[2] Brown S, Yechiali U. Scheduling deteriorating jobs on a single process[J]. Operations Research, 1990, 38: 495-498.

[3] Gupta J N D, Gupta S K. Single facility scheduling with nonlinear processing times[J]. Computers and Industrial Engineering, 1988, 14: 387-393.

[4] Cheng T C E, Ding Q, Kovalyov M Y, et al. Scheduling jobs with piecewise linear decreasing processing times[J]. Naval Research Logistics, 2003, 50: 531-554.

[5] Cheng T C E, Ding Q, Lin B M T. A concise survey of scheduling with time-dependent processin times[J]. European Journal of Operational Research, 2004, 152: 1-13.

[6] Ji M, Cheng T C E. Parallel machine scheduling of simple linear deteriorating jobs[J]. Theoretical Computer Science, 2009, 410: 3761-3768.

[7] Li S S, Ng C T, Cheng T C E, et al. Parallel-batch scheduling of deteriorating jobs with release dates to minimize the makespan[J]. European Journal of Operational Research, 2011, 210: 482-488.

[8] Kang L Y, Ng C T. A note on a fully polynomial-time approximation scheme for parallel-machine scheduling with deteriorating jobs[J]. International Journal of Production Economics, 2007, 109: 180-184.

[9] Ji M, Cheng T C E. An FPTAS for scheduling jobs with piecewise linear decreasing processing times to minimize makespan[J]. Information Processing Letters, 2007, 102: 41-47.

[10] Luo W C, Ji M. Scheduling a variable maintenance and linear deteriorating jobs on a single machine[J]. Information Processing Letters, 2015, 115: 33-39.

[11] 刘澈，罗成新. 带到达时间、不可用区间、拒绝工件的单机排序问题[J]. 重庆师范大学学报：自然科学版，2013，30(1)：17-20.

Liu C, Luo C X. Single machine scheduling problem with release dates, rejection and an unavailable interval[J]. Journal of Chongqing Normal University: Natural Science, 2013, 30(1): 17-20.

[12] Li W X, Zhao C L. Deteriorating jobs scheduling on a single machine with release dates, rejection and a fixed non-availability interval[J]. Journal of Applied Mathematics and Computing, 2015, 48: 585-605.

[13] Liu M, Zheng F F, Chu C B, et al. Scheduling deteriorating jobs on a single machine with release times and rejection [J]. Discrete Mathematics, Algorithms and Applications, 2012, 4(2): 1250032.

[14] Li S S, Yuan J J. Parallel-machine scheduling with deteriorating jobs and rejection[J]. Theoretical Computer Science, 2010, 411: 3642-3650.

[15] Kovalyov M Y, Kubiak W. A fully polynomial approximation scheme for minimizing makespan of deteriorating jobs [J]. Journal of Heuristics, 1998, 3: 287-297.

[16] Kovalyov M Y, Kubiak W. A fully polynomial approximation scheme for the weighted earliness-tardiness problem [J]. Operations Research, 1999, 47: 757-761.

Operations Research and Cybernetics

Single Machine Scheduling with Piecewise Linear Decreasing Processing Times and Rejection Jobs

SUI Min, ZHAO Chuanli

(School of Mathematics and Systems Science, Shenyang Normal University, Shenyang 110034, China)

Abstract: This paper considers single-machine scheduling with piecewise linear decreasing processing times and rejection. In this model, the actual processing time of a job is a piecewise linear decreasing function of its starting time. The objective is to minimize the sum of the makespan of the accepted jobs and the total penalty of the rejected jobs. The problem is NP-hard. Based on the analysis of the problem, we give a fully polynominal time approximation scheme. The complexity of the fully polynominal time approximation scheme is $O(n^4 L^4/\varepsilon^3)$.

Key words: single machine scheduling; piecewise linear decreasing; rejection; fully polynominal time approximation scheme

（责任编辑　黄　颖）