2015 年 7 月 重庆师范大学学报(自然科学版) Jul. 2015
第 32 卷 第 4 期 Journal of Chongqing Normal University (Natural Science) Vol. 32 No. 4

运筹学与控制论

DOI:10.11721/cqnuj20150401

带有不可用区间及拒绝的最大完工时间单机排序问题*

闫力君, 赵玉芳
(沈阳师范大学 数学与系统科学学院, 沈阳 110034)

摘要:研究带有退化效应、拒绝工件及不可用区间的单机排序问题。该问题中,工件可以被排在机器上进行加工,也可以被拒绝,但是需要支付一定的拒绝惩罚。加工工件的开始加工时间越晚,则工件的实际加工时间越大。机器带有不可用区间,在此区间内任何工件都不能被加工。目标函数为所有拒绝工件的拒绝惩罚与接受工件的最大完工时间之和。首先给出了拟多项式时间的动态规划算法,最后得到了一个全多项式近似方案。

关键词:拒绝惩罚;退化效应;全多项式近似方案;不可用区间;排序
中图分类号:O223 **文献标志码:**A **文章编号:**1672-6693(2015)04-0017-06

在实际生产过程中,退化现象是由于生产设备长时间使用而导致的。而且,厂家通常会拒绝一些加工时间较长且获利相对较小的工件,虽然需要付出一定的惩罚费用,但是可以获取更大的利润,该惩罚费用即为拒绝惩罚。Shabtay 等[1]给出了与拒绝费用有关的排序问题的综述。Li 等[2]研究了最大完工时间的平行机排序问题,且拒绝费用有上限,给出了该问题的 2-近似算法,并给出全多项式近似方案。对于工件的实际加工时间与其开始加工时间有关的模型,Mosheiov[3]研究了单机排序问题,说明了最大完工时间、加权总完工时间等问题都是多项式时间可解的。Wu 等[4]研究了加权总完工时间问题,给出了该问题的启发式算法和分支定界算法。Ji 等[5]研究了总完工时间以及加权总完工时间问题,给出了该问题的动态规划算法以及全多项式近似方案。刘澈等及赵升华等[6-7]研究了机器带有不可用区间的单机排序问题,目标函数是所有拒绝工件的拒绝惩罚与所有接受工件的最大完工时间之和。Cheng 等[8]研究了工件允许被拒绝的单机排序问题,说明了总完工时间问题以及加权总完工时间问题是 NP-难的。Li 等[9]研究了带有退化效应和拒绝工件的平行机排序问题。Seiden[10]研究了带有退化效应和优先约束的排序问题。

此外,机器通常需要进行保养,或发生故障时进行维修等原因,导致机器在某一时间段内无法工作,该时间段即为机器的不可用区间。对于机器带有不可用区间的模型,Kacem[11]研究了总完工时间问题。Kacem 等[12]研究了加权总完工时间问题,此外给出了该问题的一个全多项式时间近似方案。Lee[13]研究了加权总完工时间问题,证明了单机及平行机排序问题都是 NP-难的。此外给出问题的启发式算法,并且对其进行了误差界分析。Chen[14]研究了机器是中断重复的单机极小化总流水时间的排序问题,且机器带有有限个不可用区间,对于这个问题,给出了分枝定界算法。Vahedi-Nouri 等[15]研究了带有学习效应以及多个不可用区间的排序问题。Zhao 等[16]研究了平行机排序问题。对于只有一台机器带有不可用区间特殊情况,给出了的全多项式近似方案。

机器带有不可用区间以及拒绝惩罚同时存在的现象更普遍,本文将上述问题结合起来,使问题变得更加复杂,讨论了带有退化效应、工件允许被拒绝及机器带有不可用区间的单机排序问题,目标函数是所有拒绝工件的拒绝惩罚与接受工件的最大完工时间之和。首先给出了拟多项式时间的动态规划算法,然后通过构造输入将退化因子进行取整运算,再通过动态规划算法得到退化因子取整后的一个最优排序,按照这个工件排序得到原问题的一个可行排序,最后得到了一个全多项式近似方案。

1 问题描述

具体问题描述如下:n 个互不相关的工件 $J_1,J_2,\cdots,J_n$ 在一台机器上加工,所有工件具有相同的到达时间

* **收稿日期:**2014-04-24 **修回日期:**2015-03-13 **网络出版时间:**2015-3-24 13:54
资助项目:辽宁省教育厅科学技术研究项目(No. L2014433)
作者简介:闫力君,女,研究方向为组合最优化,E-mail:1015399330@qq.com;**通信作者:**赵玉芳,教授,E-mail: yfzhao2004@163.com
网络出版地址:http://www.cnki.net/kcms/detail/50.1165.n.20150324.1354.040.html

t_0,且 $t_0>0$;工件 J_j 的实际加工时间为 $p_j=b_jt$,其中 $b_j>0$ 为工件 J_j 的退化因子;t 为其开始加工时间,显然 $t\geqslant t_0>0$;$e_j>0$ 是工件 J_j 被拒绝时的惩罚。记被加工的工件的集合为 S,所有被拒绝的工件的集合为 R。$[T_1,T_2)$ 为机器的不可用区间,在此区间内机器不能加工任何工件,$nr-a$ 表示机器为不可恢复的,即某个工件在不可用区间之前没加工完,则在不可用区间之后需要重新加工这个工件。目标函数是所有拒绝工件的惩罚与接受工件的最大完工时间之和。同时带有退化效应、工件允许被拒绝及机器带有不可用区间的单机排序问题,利用三参数表示法 $\alpha|\beta|\gamma$ 可记为

$$1|nr-a,p_j=b_jt,x_{\text{reject}}|C_{\max}(S)+\sum\nolimits_{j\in \mathbf{R}}e_j\text{ 。}$$

2 动态规划算法

由于问题 $1|p_j=b_jt,x_{\text{reject}}|C_{\max}(S)+\sum\nolimits_{j\in R}e_j$ 是 NP-难的,显然问题 $1|nr-a,p_j=b_jt,x_{\text{reject}}|C_{\max}(S)+\sum\nolimits_{j\in R}e_j$ 至少是 NP-难的。为了解决这个问题,首先给出以下引理。

引理 1[3] 工件的开始加工时间是 t_0,工件的实际加工时间是退化函数 $p_j=b_jt$,则集合 S 中工件的最大完工时间是 $C_{\max}(S)=t_0\prod\nolimits_S(1+b_j)$ 。

由于加工工件既可以排在不可用区间之前,也可以排在不可用区间之后,假设加工工件排在 T_2 之后的工件集合为 S_1,可得如下引理。

引理 2 若 $S_1=\varnothing$,则集合 S 中工件的最大完工时间是 $C_{\max}(S)=t_0\prod\nolimits_S(1+b_j)$,$\forall S\subseteq N$;若 $S_1\neq\varnothing$,则集合 S 中工件的最大完工时间是 $C_{\max}(S)=T_2\prod\nolimits_{S_1}(1+b_j)$,$\forall S\subseteq N$ 。

证明 若 $S_1=\varnothing$,即所有加工工件都被排在不可用区间之前加工,工件的开始加工时间是 t_0。由引理 1,集合 S 中工件的最大完工时间是 $C_{\max}(S)=t_0\prod\nolimits_S(1+b_j)$,$\forall S\subseteq N$;若 $S_1\neq\varnothing$,由于排在不可用区间 $[T_1,T_2)$ 之后的加工工件开始加工时间为 T_2,由引理 1,集合 S 中工件的最大完工时间是 $C_{\max}(S)=T_2\prod\nolimits_{S_1}(1+b_j)$ $\forall S\subseteq N$ 。 证毕

由引理 1 可知,工件的加工顺序不影响目标函数值,不失一般性,假设工件按照 SPT 规则排序。工件可以被拒绝也可以被加工。由于机器带有不可用区间,则工件既可以排在不可用区间之前也可以被排在不可用区间之后,从而有下述动态规划算法。

首先工件按照 SPT 规则排序。假设工件 $J_1,J_2,\cdots,J_{i-1}$ 已排完,用 $f_i(u,E)$ 表示工件 $J_1,J_2,\cdots,J_i$ 当前状态下在 T_2 之后加工工件最小的总加工时间与拒绝工件的总惩罚之和,其中 u 表示工件 $J_1,J_2,\cdots,J_i$ 中在 T_1 之前加工工件的总加工时间,E 表示工件 $J_1,J_2,\cdots,J_i$ 中拒绝加工工件的总惩罚。在最优排序中,工件 J_i 或者被加工或者被拒绝;加工工件 J_i 或者排在 T_1 之前或者排在 T_2 之后。

假设工件 $J_1,J_2,\cdots,J_{i-1}$ 已排完,工件 J_i 分为以下两种情况。

1)工件 J_i 被拒绝,则工件 $J_1,J_2,\cdots,J_{i-1}$ 中拒绝加工工件的总惩罚为 $E-e_i$,在 T_1 之前加工工件的总加工时间不变,有 $f_i(u,E)=f_{i-1}(u,E-e_i)+e_i$;

2)工件 J_i 被加工,工件 J_i 分为以下两种情况。

如果 J_i 排在 T_1 之前加工。设工件 $J_1,J_2,\cdots,J_{i-1}$ 中在 T_1 之前加工工件的总加工时间为 u_{i-1},则在 T_1 之前加工工件的总加工时间增加 $b_i(u_{i-1}+t_0)$,排在 T_2 之后加工工件的总加工时间与拒绝加工工件的总惩罚之和不变。

如果 J_i 排在 T_2 之后加工。则在 T_1 之前加工工件的总加工时间不变,排在 T_2 之后加工工件的总加工时间与拒绝加工工件的总惩罚之和增加 $b_i(f_{i-1}(u,E)-E+T_2)$。

如果 $\dfrac{u-b_it_0}{1+b_i}$ 是整数,则工件 J_i 或者排在 T_1 之前或者排在 T_2 之后,有

$$f_i(u,E)=\min\left\{f_{i-1}\left(\frac{u-b_it_0}{1+b_i},E\right),f_{i-1}(u,E)+b_i(T_2+f_{i-1}(u,E)-E)\right\};$$

如果 $\dfrac{u-b_it_0}{1+b_i}$ 不是整数,则工件 J_i 排在 T_2 之后,有 $f_i(u,E)=f_{i-1}(u,E)+b_i(T_2+f_{i-1}(u,E)-E)$。

由以上分析可得拟多项式时间动态规划算法如下。

动态规划算法 1(DP1)　1）初始条件：如果 $j=0, u=0, E=0$，则 $f_j(u,E)=0$；否则 $f_j(u,E)=\infty$。

2）对每个工件 J_i，当 $1\leqslant i\leqslant n$，$0\leqslant u\leqslant T_1-t_0$，$0\leqslant E\leqslant \sum_{j=1}^{n} e_j$ 计算：

$$f_i(u,E)=\min\begin{cases}\begin{cases}\min\left\{f_{i-1}\left(\dfrac{u-b_it_0}{1+b_i},E\right),f_{i-1}(u,E)+b_i(T_2+f_{i-1}(u,E)-E)\right\},\text{若}\dfrac{u-b_it_0}{1+b_i}\text{是整数，}\\ f_{i-1}(u,E)+b_i(T_2+f_{i-1}(u,E)-E)\text{，否则，}\end{cases}\\ f_{i-1}(u,E-e_i)+e_i\text{。}\end{cases}$$

如果 $\min f_n(u,E)=E$，此时所有加工工件都在不可用区间之前加工，不可用区间之后没有加工工件，令 $Z=\{(u,E)\mid f_n(u,E)=E\}$，则 $\min\limits_{(u,E)\in Z}\{u+t_0+E\}$ 为最优目标函数值。否则 $T_2+\min f_n(u,E)$ 为最优目标函数值。

例　有 3 个工件 J_1, J_2, J_3，其中 $t_0=2, b_1=1, b_2=1, b_3=2, T_1=12, T_2=16, e_1=10, e_2=10, e_3=15$。应用上述动态规划算法得出主要过程如下。

$i=1$ 时，$u=0, E=0$，此时 $\dfrac{u-b_it_0}{1+b_i}$ 不是整数，$f_1(0,0)=f_{i-1}(u,E)+b_i(T_2+f_{i-1}(u,E)-E)=16$；$u=0$，$E=10$，$f_1(0,10)=f_{i-1}(u,E-e_i)+e_i=10$；$u=2, E=0$，此时 $\dfrac{u-b_it_0}{1+b_i}$ 是整数，

$$f_1(2,0)=\min\left\{f_{i-1}\left(\frac{u-b_it_0}{1+b_i},E\right),f_{i-1}(u,E)+b_i(T_2+f_{i-1}(u,E)-E)\right\}=0\text{。}$$

以此类推，有：

$$f_2(0,0)=f_{i-1}(u,E)+b_i(T_2+f_{i-1}(u,E)-E)=48,$$

$$f_2(0,10)=f_{i-1}(u,E-e_i)+e_i=26,\ f_2(0,20)=f_{i-1}(u,E-e_i)+e_i=20,$$

$$f_2(2,0)=\min\left\{f_{i-1}\left(\frac{u-b_it_0}{1+b_i},E\right),f_{i-1}(u,E)+b_i(T_2+f_{i-1}(u,E)-E)\right\}=16,$$

$$f_2(2,10)=f_{i-1}(u,E-e_i)+e_i=10,$$

$$f_2(6,0)=\min\left\{f_{i-1}\left(\frac{u-b_it_0}{1+b_i},E\right),f_{i-1}(u,E)+b_i(T_2+f_{i-1}(u,E)-E)\right\}=0,$$

$$f_3(0,0)=f_{i-1}(u,E)+b_i(T_2+f_{i-1}(u,E)-E)=176,$$

$$f_3(0,10)=f_{i-1}(u,E)+b_i(T_2+f_{i-1}(u,E)-E)=90,$$

$$f_3(0,15)=f_{i-1}(u,E-e_i)+e_i=63,$$

$$f_3(0,20)=f_{i-1}(u,E)+b_i(T_2+f_{i-1}(u,E)-E)=52,$$

$$f_3(0,25)=f_{i-1}(u,E-e_i)+e_i=41,\ f_3(0,35)=f_{i-1}(u,E-e_i)+e_i=35,$$

$$f_3(2,0)=\min\left\{f_{i-1}\left(\frac{u-b_it_0}{1+b_i},E\right),f_{i-1}(u,E)+b_i(T_2+f_{i-1}(u,E)-E)\right\}=80,$$

$$f_3(2,10)=\min\left\{f_{i-1}\left(\frac{u-b_it_0}{1+b_i},E\right),f_{i-1}(u,E)+b_i(T_2+f_{i-1}(u,E)-E)\right\}=42,$$

$$f_3(2,15)=f_{i-1}(u,E-e_i)+e_i=31,\ f_3(2,25)=f_{i-1}(u,E-e_i)+e_i=25,$$

$$f_3(4,0)=\min\left\{f_{i-1}\left(\frac{u-b_it_0}{1+b_i},E\right),f_{i-1}(u,E)+b_i(T_2+f_{i-1}(u,E)-E)\right\}=48,$$

$$f_3(4,20)=\min\left\{f_{i-1}\left(\frac{u-b_it_0}{1+b_i},E\right),f_{i-1}(u,E)+b_i(T_2+f_{i-1}(u,E)-E)\right\}=20,$$

$$f_3(6,0)=\min\left\{f_{i-1}\left(\frac{u-b_it_0}{1+b_i},E\right),f_{i-1}(u,E)+b_i(T_2+f_{i-1}(u,E)-E)\right\}=32,$$

$$f_3(6,15)=f_{i-1}(u,E-e_i)+e_i=15,$$

$$f_3(10,0)=\min\left\{f_{i-1}\left(\frac{u-b_it_0}{1+b_i},E\right),f_{i-1}(u,E)+b_i(T_2+f_{i-1}(u,E)-E)\right\}=16,$$

$$f_3(10,10)=\min\left\{f_{i-1}\left(\frac{u-b_it_0}{1+b_i},E\right),f_{i-1}(u,E)+b_i(T_2+f_{i-1}(u,E)-E)\right\}=10。$$

则 $\min f_3(10,10)=10$,即 $\min f_n(u,E)=E$,则最优目标函数值是 $\min\limits_{(u,E)\in Z}\{u+t_0+E\}=22$。

下面分析 DP1 的算法复杂性。

在状态 $f_i(u,E)$ 中 i 可以取值 $1,2,\cdots,n$,u 可以取值 $0,\cdots,T_1-t_0$,E 可以取值为 $0,1,\cdots,\sum_{j=1}^{n}e_j$。因此算法共有 $O\left(nT_1\sum_{j=1}^{n}e_j\right)$ 种状态,则算法的计算复杂性为 $O\left(n(T_1-t_0)\sum_{j=1}^{n}e_j\right)$ 。

定理 1 动态规划算法 1 计算复杂性为 $O\left(n(T_1-t_0)\sum_{j=1}^{n}e_j\right)$ 。

3 全多项式近似方案

由定理 1 可知动态规划算法 1 为拟多项式时间算法,在此基础上给出问题 1

$$|nr-a,p_j=b_jt,x_{\mathrm{reject}}|C_{\max}(S)+\sum_{j\in\mathbf{R}}e_j$$

的一个 FPTAS 算法。

对于任意给定的 $\varepsilon'>0$,$x\geqslant1$,如果 $(1+\varepsilon')^{k-1}<x<(1+\varepsilon')^k$ 成立,定义 $\lceil x\rceil_{\varepsilon'}=(1+\varepsilon')^k$,$\lfloor x\rfloor_{\varepsilon'}=(1+\varepsilon')^{k-1}$。如果 $x=(1+\varepsilon')^k$,则 $\lceil x\rceil_{\varepsilon'}=\lfloor x\rfloor_{\varepsilon'}$,并且 $\lceil x\rceil_{\varepsilon'}\leqslant(1+\varepsilon')x$。对于任意 $\varepsilon>0$,定义 $b_j'=\lceil 1+b_j\rceil_{\varepsilon'}-1$,这里 $\varepsilon'=\frac{\varepsilon}{2n}$。定义 L_j 使得 $1+b_j'=(1+\varepsilon')^{L_j}$,所以 $L_j=\frac{\log\lceil 1+b_j\rceil_{\varepsilon'}}{\log(1+\varepsilon')}$,由 $\lceil 1+b_j\rceil_{\varepsilon'}\leqslant(1+\varepsilon')(1+b_j)$ 有 $\log\lceil 1+b_j\rceil_{\varepsilon'}\leqslant\log(1+\varepsilon')(1+b_j)$,所以 $L_j=\frac{\log\lceil 1+b_j\rceil_{\varepsilon'}}{\log(1+\varepsilon')}=O\left(\frac{n\log(1+b_j)}{\varepsilon}\right)$。由于 $b_j'>b_j$,按照 b_j' 得到的排序可行,按照加工时间替换为 b_j 的排序一定可行。从而得到下列算法。

算法 第 1 步,对于任意的 $\varepsilon'>0$,令 $\varepsilon'=\frac{\varepsilon}{2n}$,对于一个例子 I,通过令 $b_j'=\lceil 1+b_j\rceil_{\varepsilon'}-1$ 的方式构造出新的例子 I′;第 2 步,对于例子 I′,通过下面的动态规划算法 2(DP2)得到例子 I′的最优排序 π^*(I′);第 3 步,用 b_j 替换 π^*(I′)中的 b_j',工件排序不变,从而得到例子 I 的可行排序。

定义 $\tau_k=(1+\varepsilon')^k$,用 $f_i(k,E)$ 表示考虑工件 $J_1,J_2,\cdots,J_i$ 的加工工件在 T_2 之后加工的工件总加工时间与拒绝工件的最小的总惩罚之和,其中 k 表示前 i 个工件中加工工件在 T_1 之前加工的工件总加工时间为 $t_0\tau_k-t_0$,E 表示前 i 个工件中拒绝加工的总惩罚。在任意排序中,工件 J_i 或者被加工或者被拒绝。

假设工件 $J_1,J_2,\cdots,J_{i-1}$ 已排完,工件 J_i 分为以下两种情况:

1)工件 J_i 拒绝,有 $f_i(k,E)=f_{i-1}(k,E-e_i)+e_i$;

2)工件 J_i 加工,工件 J_i 分为以下两种情况:如果 J_i 排在 T_1 之前加工,则 k 增加 L_i,排在 T_2 之后加工工件的总完工时间与拒绝加工工件的总惩罚之和不变;如果 J_i 排在 T_2 之后加工,则 k 不变,排在 T_2 之后加工工件的总完工时间与拒绝加工工件的总惩罚之和增加 $b_i'(f_{i-1}(k,E)-E+T_2)$。

如果 $\frac{t_0\tau_k-t_0-b_i't_0}{1+b_i'}$ 是整数,则工件 J_i 或者排在 T_1 之前或者排在 T_2 之后,有

$$f_i(k,E)=\min\{f_{i-1}(k-L_i,E),f_{i-1}(k,E)+b_i'(T_2+f_{i-1}(k,E)-E)\};$$

如果 $\frac{t_0\tau_k-t_0-b_i't_0}{1+b_i'}$ 不是整数,则工件 J_i 排在 T_2 之后,有

$$f_i(k,E)=f_{i-1}(k,E)+b_i'(T_2+f_{i-1}(k,E)-E)。$$

由以上分析可得拟多项式时间动态规划算法如下。

动态规划算法 2(DP2) 1)初始条件:如果 $j=0$,$k=0$,$e=0$,则 $f_j(k,E)=0$;否则 $f_j(k,E)=\infty$。2)对每个工件 J_i,当 $1\leqslant i\leqslant n$,$L_1\leqslant k\leqslant\sum_{i=1}^{n}L_i$,$0\leqslant E\leqslant\sum_{j=1}^{n}e_j$ 计算:

$$f_i(k,E)=\min\begin{cases}\begin{cases}\min\{f_{i-1}(k-L_i,E),f_{i-1}(k,E)+b_i'(T_2+f_{i-1}(k,E)-E)\},若\dfrac{t_0\tau_k-t_0-b_i't_0}{1+b_i'}是整数,\\ f_{i-1}(k,E)+b_i'(T_2+f_{i-1}(k,E)-E),否则,\end{cases}\\ f_{i-1}(k,E-e_i)+e_i。\end{cases}$$

如果 $\min f_n(k,E)=E$，令 $Z=\{(t_0\tau_k,E)\mid f_n(k,E)=E\}$，则 $\min\limits_{(k,E)\in Z}\{t_0\tau_k+E\}$ 为最优目标函数值。否则 $T_2+\min f_n(k,E)$ 为最优值。

下面分析算法复杂性。在 $f_i(k,E)$ 中 i 可以取值 $1,2,\cdots,n$，k 可以取值 $L_1,L_2,\cdots,\sum_{i=1}^{n}L_i$，$E$ 可以取值 $0,1,\cdots,\sum_{j=1}^{n}e_j$。故算法共有 $O\left(n\sum_{j=1}^{n}e_j\sum_{j=1}^{n}L_j\right)$ 种状态，由前面假设可知 $L_j=\frac{\log\lceil 1+b_j\rceil_{\varepsilon'}}{\log(1+\varepsilon')}=O\left(\frac{\log(1+b_j)}{\varepsilon}\right)$，所以上述的全多项式近似算法的计算复杂性为 $O\left(\frac{n^2}{\varepsilon}\sum_{j=1}^{n}e_j\sum_{j=1}^{n}\log(1+b_j)\right)$。

于是得到下面定理。

定理 2　算法得到的目标值 $C'_{\max}(S)+\sum_{j\in R}e_j\leqslant(1+\varepsilon)C_{\max}(S)+\sum_{j\in R}e_j$，且算法的运行时间为 $O\left(\frac{n^2}{\varepsilon}\sum_{j=1}^{n}\log(1+b_j)\right)$。

证明　设加工工件排在 T_2 之后的工件集合为 S_1，则：

若 $S_1=\varnothing$，则

$$C'_{\max}(S)+\sum_{\bar S}e_j=t_0\prod_S(1+b'_j)+\sum_{\bar S}e_j\leqslant t_0\prod_S(1+\varepsilon')(1+b_j)+\sum_{\bar S}e_j=$$
$$t_0(1+\varepsilon')^{|S|}\prod_S(1+b_j)+\sum_{\bar S}e_j\leqslant t_0(1+\varepsilon')^n\prod_S(1+b_j)+\sum_{\bar S}e_j\ ;$$

若 $S_1\neq\varnothing$，则

$$C'_{\max}(S)+\sum_{\bar S}e_j=T_2\prod_{S_1}(1+b'_j)+\sum_{\bar S}e'_j\leqslant T_2\prod_{S_1}(1+\varepsilon')(1+b_j)+\sum_{\bar S}e_j=$$
$$T_2(1+\varepsilon')^{|S_1|}\prod_{S_1}(1+b_j)+\sum_{\bar S}e_j\leqslant T_2(1+\varepsilon')^n\prod_{S_1}(1+b_j)+\sum_{\bar S}e_j\ 。$$

由于 $(1+\varepsilon')^n=\left(1+\frac{\varepsilon}{2n}\right)^n\leqslant 1+\varepsilon$，对于任意的 $0<\varepsilon\leqslant 2$，有

$$C'_{\max}(S)+\sum_{\bar S}e_j\leqslant(1+\varepsilon)\left(C_{\max}(S)+\sum_{\bar S}e_j\right)\ 。$$

由前面算法复杂性分析可知，算法的计算复杂性为 $O\left(\frac{n^2}{\varepsilon}\sum_{j=1}^{n}e_j\sum_{j=1}^{n}\log(1+b_j)\right)$。　　证毕

4 结论

本文考虑的是带有退化效应、拒绝工件及不可用区间的单机排序问题，目标函数为所有拒绝工件的拒绝惩罚与接受工件的最大完工时间之和。首先给出该问题的一个动态规划算法，在此基础上给出了一个全多项式时间近似方案，该全多项式时间近似方案的计算复杂性为 $O\left(\frac{n^2}{\varepsilon}\left(\sum_{j=1}^{n}e_j\sum_{j=1}^{n}\log(1+b_j)\right)\right)$。在实际生产过程中，机器通常需要进行保养，或发生故障时采取维修措施等原因，导致机器在某一时间段内无法工作。因此，进一步研究退化效应模型以及机器带有不可用区间的排序问题具有实际意义和应用价值。对于具有其他以及机器带有不可用区间的排序问题，如总完工时间与所有拒绝惩罚之和等问题有待进一步研究。

参考文献：

[1] Shabtay D, Gaspar N, Kaspi M. A survey on offline scheduling with rejection[J]. Journal of Scheduling, 2013, 16(1): 3-28.

[2] Li W, Li J, Zhang X, et al. Parallel-machine scheduling problem under the job rejection constraint[J]. Lecture Notes in Computer Science, 2014, 8497: 158-169.

[3] Mosheiov G. Scheduling jobs under simple linear deterioration[J]. Computers & Operations Research, 1994, 21(6): 653-659.

[4] Wu C C, Hsu P H, Chen J C, et al. Genetic algorithm for minimizing the total weighted completion time scheduling problem with learning and release times[J]. Computers & Operations Research, 2011, 38(7): 1025-1034.

[5] Ji M, He Y, Cheng T C E. Scheduling linear deteriorating jobs with an availability constraint on a single machine[J]. Theoretical Computer Science, 2006, 362(1): 115-126.

[6] 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. Scheduling problem with release time, non-availability interval and rejection on a single machine[J]. Journal of Chongqing Normal University: Natural Science, 2013, 30(1): 17-20.

[7] 赵升华,罗成新. 带有拒绝工件和机器具有不可用区间的单机排序问题[J]. 重庆师范大学学报:自然科学版,2014,31(2):5-9.

Zhao SH, Luo C X. Single machine scheduling problem with rejection jobs and a machine non—availability interval [J]. Journal of Chongqing Normal University: Natural Science, 2014, 31(2): 5-9.

[8] Cheng Y, Sun S. Scheduling linear deteriorating jobs with rejection on a single machine[J]. European Journal of Operational Research, 2009, 194(1): 18-27.

[9] Li S S, Yuan J J. Parallel-machine scheduling with deteriorating jobs and rejection[J]. Theoretical Computer Science, 2010, 411(40): 3642-3650.

[10] Seiden S S. Preemptive multiprocessor scheduling with rejection[J]. Theoretical Computer Science, 2001, 262(1): 437-458.

[11] Kacem I. Approximation algorithms for the makespan minimization with positive tails on a single machinewith a fixed non-availability interval[J]. Journal of Combinatorial Optimization, 2009, 17(2): 117-133.

[12] Kacem I, Mahjoub A R. Fully polynomial time approximation scheme for the weighted flow-time minimization on a single machine with a fixed non-availability interval[J]. Computers & Industrial Engineering, 2009, 56(4): 1708-1712.

[13] Lee C Y. Machine scheduling with an availability constraint[J]. Journal of global optimization, 1996, 9(3/4): 395-416.

[14] Chen W J. Minimizing total flow time in the single-machine scheduling problem with periodic maintenance[J]. Journal of the Operational Research Society, 2006, 57(4): 410-415.

[15] Vahedi-Nouri B, Fattahi P, Rohaninejad M, et al. Minimizing the total completion time on a single machine with the learning effect and multiple availability constraints [J]. Applied Mathematical Modelling, 2013, 37(5): 3126-3137.

[16] Zhao C L, Tang H Y. Parallel machines scheduling with deteriorating jobs and availability constraints[J]. Japan Journal of Industrial and Applied Mathematics, 2014, 31(3): 501-512.

Operations Research and Cybernetics

Scheduling Problem with Non-availability Interval and Rejection to Minimize the Makespan on a Single Machine

YAN Lijun, ZHAO Yufang

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

Abstract: In this paper, we consider a single machine scheduling problems with a deteriorating effect, rejection penalty and an availability interval. In this model, the processing time of a job is related with its starting time and jobs can be rejected. A job is either processed on the machine, or rejected, in which case a rejection penalty has to be paid. The machine can't process any job in the non-availability interval. The objective is to minimize the total rejection penalty of all the rejected jobs and the sum of the make span of the accepted jobs. First, we propose a pseudo-polynomial time dynamic programming algorithm. And then, we design a fully polynomial-time approximation scheme to solve the problem.

Key words: rejection penalty; deteriorating; fully polynomial time approximation scheme; non-availability interval; scheduling

(责任编辑 黄 颖)