2023 年 5 月 重庆师范大学学报(自然科学版) May 2023
第 40 卷 第 3 期 Journal of Chongqing Normal University (Natural Science) Vol. 40 No. 3

运筹学与控制论 DOI:10.11721/cqnuj20230308

带有退化、拒绝和不可用区间的单机排序问题*

何欣怡，赵玉芳，陈状状
(沈阳师范大学 数学与系统科学学院，沈阳 110034)

摘要:【目的】考虑带有退化工件、拒绝和不可用区间的单机排序问题。【方法】假设工件有不同的基本加工时间和相同的退化率，工件可以被拒绝，被拒绝的工件需要支付拒绝惩罚，机器在给定的时间区间内是不可用的且工件不可恢复。目标是极小化接受工件的总完工时间与被拒绝工件的总拒绝惩罚之和。【结果】对于这个 NP-难问题，在不可用区间前、后，工件按照基本加工时间 a_j 的非减顺序排列可以得到最优解，给出一个拟多项式时间动态规划算法和一个完全多项式时间近似策略。【结论】推广了已有文献的模型。

关键词:单机排序;退化;拒绝;不可用区间
中图分类号:O223 **文献标志码:**A **文章编号:**1672-6693(2023)03-0008-08

当前有关排序问题的理论研究已经涉及到很多实际生产环境中，例如在轧钢调度问题中，铸锭在等待加工时温度会下降，因此它需要被重新加热，让温度达到一定要求再进行轧制，由此导致轧制时间增加;而轧制过程中机器也可能因故障或检修等原因而不可用，此时工厂为了节约成本或按时交货，可能会选择外包一部分货品，进而支付一定费用。事实上，加工过程中由于中途维修机器或者更换工具等原因，导致一台机器在此时不可用的情形在工业中很常见，因此带有不可用区间约束的问题受到了广泛关注。Lee[1]定义了两种情况:可恢复和不可恢复。当工件在不可用区间之前没有完工，机器再次可用时继续加工，则称为可恢复的，反之为不可恢复的。Ma 等人[2]对带有机器不可用区间的问题进行了综述。Zhao 等人[3]研究了两台平行机排序问题，其中一台机器有一段固定的不可用区间，给出了这个问题的完全多项式时间近似策略(fully polynomial time approximation scheme，FPTAS)。Zhao 等人[4]研究了平行机的情况，每台机器都有一个固定的不可用区间，提出了一个拟多项式时间动态规划算法。Kacem 等人[5]研究了两个目标函数为最大延误的单机排序问题，且工件带有释放时间，其中一个问题是机器在一个给定的区间中不可用，另一个问题是与操作员有关的不可用区间，他们分别给出了这两个问题的多项式时间近似策略(polynomial-time approximation scheme，PTAS)算法。金苗苗等人[6]研究了机器具有不可用区间且工件可拒绝下的单机重新排序问题，目标为极小化接受工件的总加权完工时间、总拒绝费用及加权最大延误之和，给出了拟多项式时间动态规划算法和 FPTAS。

此外，在传统的研究中工件的加工时间总是固定的。但在实际生产过程中，工件的加工时间可能会随着时间发生变化，Gawiejnowicz[7]对近 40 年有关工件加工时间可变的排序问题进行了综述。机器由于磨损、升温等原因，导致工件的加工时间与开始加工时间有关，开始加工时间越晚，加工时间就越长，这种现象称为退化效应。有很多文献研究了线性退化工件的模型。此外还有一种有相同退化率的模型[8-10]。当机器持续可用时，Cheng 等人[11]研究了单机排序问题，目标是极小化工期与提前惩罚、误工惩罚之和，他们给出了求最优解的多项式时间算法。对于上述的目标函数，Cheng 等人[12]还研究了平行机排序问题，证明了这个问题是 NP-难的，并给出求近似最优解的遗传算法。Lee 等人[13]研究了带有释放时间的单机极小化最大完工时间问题，构造了一个分支定界算法。Zhao 等人[14]研究了带有机器不可用区间，工件有相同退化率的单机排序问题，目标是极小化总完工时间，构造了一个拟多项式时间动态规划算法和一个 FPTAS，还研究了平行机的情况，给出了拟多项式时间动态规划算法。王吉波等人[15]研究了带有学习效应、恶化效应和公共工期指派的单机排序问题，目标函数为工件的

* 收稿日期:2022-03-20 修回日期:2023-01-08 网络出版时间:2023-06-15T17:13
资助项目:国家自然科学基金青年项目(No. 12101417);辽宁省教育厅科学研究项目(No. LFW202001)
第一作者简介:何欣怡，女，研究方向为组合最优化、随机运筹学，E-mail:1226237549@qq.com;通信作者:赵玉芳，女，副教授，博士，E-mail:yfzhao2004@163.com
网络出版地址:https://kns.cnki.net/kcms2/detail/50.1165.N.20230615.1320.006.html

提前、延误和公共工期的加权费用和，证明了该问题是多项式时间可解的。

Bartal 等人[16]第 1 次提出带有拒绝的排序问题，他们研究了多机排序问题，目标是极小化接受工件的最大完工时间与拒绝惩罚之和。此后得到了学者们的广泛关注。Engels 等人[17]研究了带有拒绝的单机排序问题，目标函数为接受工件的总加权完工时间与拒绝工件的总拒绝惩罚之和，证明了该问题是 NP-难的，并给出了动态规划算法和 FPTAS。Cheng 等人[18]考虑了带有退化和拒绝的单机排序问题，目标函数为接受工件的最大完工时间与拒绝工件的总拒绝惩罚之和，证明了该问题是一般意义 NP-难的，给出了拟多项式时间的动态规划算法。Zhao 等人[19]研究了带有机器不可用区间和拒绝的单机排序问题，目标是极小化接受工件的总加权完工时间与拒绝工件的总拒绝惩罚之和，给出了问题的动态规划算法和 FPTAS。Li 等人[20]研究了带有释放时间、退化、不可用区间和拒绝的单机排序问题，目标是极小化接受工件的最大完工时间和拒绝工件的总拒绝惩罚，给出一个 FPTAS 算法。国峰等人[21]研究了带有拒绝和学习效应的资源约束排序问题，对线性和凸资源分配函数的两种模型给出了最优求解算法，时间复杂度分别为 $O(n^4)$ 和 $O(n^3)$。徐晨等人[22]研究了工件可拒绝的单机多任务排序问题，目标是极小化最大完工时间、总完工时间和总加权完工时间，对上述 3 个问题都给出了拟多项式时间动态规划算法。林凌等人[23]以及张玉忠[24]都对工件可拒绝问题的研究进行了综述。

1　问题描述

假设有 n 个互相独立的工件 $J=\{J_1,J_2,\cdots,J_n\}$ 在一台机器上加工。所有工件 0 时刻可用，每个工件 J_j 有一个基本加工时间 a_j 和退化率 $b(a_j,b\geqslant 0)$，工件 J_j 的实际加工时间为 $p_j=a_j+bt$，其中 t 是工件 J_j 的开始加工时间。工件 J_j 的拒绝惩罚为 e_j，同一时间机器只能加工 1 个工件。$[T_1,T_2)$ 为机器的不可用区间 $(T_1<T_2)$，工件不可恢复。工件 J_j 要么被加工，要么被拒绝，加工工件集为 S，拒绝工件集为 $\overline{S}$，则 $J=S\cup\overline{S}$。假设 a_j，b，T_1，T_2 和 e_j 都是整数。目标是极小化总完工时间与拒绝惩罚之和。本文研究问题可以表示为：

$$1,h\,|\,nr-a,p_j=a_j+bt,\mathrm{rej}\,|\,\sum\nolimits_S C_j+\sum\nolimits_{\overline{S}} e_j,$$

式中：C_j 是工件 J_j 的完工时间，h 表示机器上一个固定的不可用区间，$nr-a$ 表示任何接受工件都是不可恢复的。

2　动态规划算法

首先给出一些有用的引理。

当不考虑拒绝时，问题 $1,h\,|\,nr-a,p_j=a_j+bt\,|\,\sum C_j$ 被证明是 NP-难的[14]，显然考虑拒绝之后，问题 $1,h\,|\,nr-a,p_j=a_j+bt,\mathrm{rej}\,|\,\sum\nolimits_S C_j+\sum\nolimits_{\overline{S}} e_j$ 至少是 NP-难的，则有下列结论。

引理 1　问题 $1,h\,|\,nr-a,p_j=a_j+bt,\mathrm{rej}\,|\,\sum\nolimits_S C_j+\sum\nolimits_{\overline{S}} e_j$ 是 NP-难的。

为了叙述方便，引入下列术语。

SPT 序　按照 a_j 的非减顺序排列工件。

引理 2[25]　对于问题 $1\,|\,p_j=a_j+bt\,|\,C_{\max}$，将工件按 SPT 序排列可以得到一个最优排序。对于某个排序 $\pi=[J_1,J_2,\cdots,J_n]$，第一个工件 J_1 的开始加工时间为 t_0，则最大完工时间 $C_{\max}=t_0(1+b)^n+\sum_{j=1}^{n}a_j(1+b)^{n-j}$。

引理 3[14]　问题 $1,h\,|\,nr-a,p_j=a_j+bt\,|\,\sum C_j$ 存在一个最优排序，使得 T_1 之前的工件按照 SPT 序排列，T_2 之后的工件按照 SPT 序排列。

由上述引理，对于问题 $1,h\,|\,nr-a,p_j=a_j+bt,\mathrm{rej}\,|\,\sum\nolimits_S C_j+\sum\nolimits_{\overline{S}} e_j$ 有如下结论。

引理 4　问题 $1,h\,|\,nr-a,p_j=a_j+bt,\mathrm{rej}\,|\,\sum\nolimits_S C_j+\sum\nolimits_{\overline{S}} e_j$ 存在一个最优排序，使得 T_1 之前接受工件按照 SPT 序排列，T_2 之后接受工件按照 SPT 序排列。

假设工件按照 SPT 序排列，使得 $a_1\leqslant a_2\leqslant\cdots\leqslant a_n$，基于引理 4 构造求解问题 $1,h\,|\,nr-a,p_j=a_j+bt,\mathrm{rej}\,|\,\sum\nolimits_S C_j+\sum\nolimits_{\overline{S}} e_j$ 的动态规划算法。

根据引理 4，工件已经按照 SPT 序排列，因此工件 J_1 要么是 T_1 之前的第 1 个加工工件，要么是 T_2 之后的第 1 个加工工件，要么被拒绝。如果工件 J_1 是 T_1 之前的第 1 个加工工件，则工件 J_2 要么是 T_1 之前的第 2 个

加工工件,要么是 T_2 之后的第1个加工工件,要么被拒绝。不失一般性,对于任意给定集合 $S_j=\{J_1,J_2,\cdots,J_j\}(j=1,2,\cdots,n)$,工件 J_j 要么是 T_1 之前的最后一个加工工件,要么是 T_2 之后的最后一个加工工件,要么被拒绝。

设 u 表示集合 S_j 中在 T_1 之前加工工件的总加工时间,v 表示集合 S_j 中在 T_2 之后加工工件的总加工时间,$F_j(u,v)$ 表示对于当前状态 (u,v),S_j 中工件的目标函数值。给定一个状态 (u,v),如果工件 J_j 是 T_1 之前的最后一个加工工件,那么它的完工时间为 u,它的开始加工时间为 $(u-a_j)/(1+b)$,则 $F_j(u,v)=F_{j-1}((u-a_j)/(1+b),v)+u$;如果工件 J_j 是 T_2 之后的最后一个加工工件,那么它的完工时间为 T_2+v,开始加工时间为 $(T_2+v-a_j)/(1+b)$,则 $F_j(u,v)=F_{j-1}(u,(v-a_j-bT_2)/(1+b))+(T_2+v)$;如果工件 J_j 被拒绝,则 $F_j(u,v)=F_{j-1}(u,v)+e_j$。

设 $F_j^{(1)}(u,v)=F_{j-1}((u-a_j)/(1+b),v)+u$,$F_j^{(2)}(u,v)=F_{j-1}(u,(v-a_j-bT_2)/(1+b))+(T_2+v)$,$F_j^{(3)}(u,v)=F_{j-1}(u,v)+e_j$。由上述分析可得动态规划算法如下。

动态规划算法(DP) 1) 初始条件:$F_1(a_1,0)=a_1$,$F_1(0,a_1+bT_2)=a_1+(1+b)T_2$,$F_1(0,0)=e_1$,否则 $F_1(u,v)=\infty$;$F_j^{(i)}(u,v)=\infty(i=1,2)$,其中 u,v 不是非负整数。

2) 对于 $j=2,3,\cdots,n$,$0\leqslant u\leqslant T_1$,$0\leqslant v\leqslant T_2(1+b)^j+D_j$,其中 $D_j=\sum\limits_{k=1}^{j}a_k(1+b)^{j-k}$,有:

$$F_j(u,v)=\min\{F_j^{(1)}(u,v),F_j^{(2)}(u,v),F_j^{(3)}(u,v)\}。$$

3) 目标函数的最优值 $F=\min\{F_n(u,v)\}$,其中 $0\leqslant u\leqslant T_1$,$0\leqslant v\leqslant T_2(1+b)^n+D_n$。

例 考虑问题 $1,h|nr-a,p_j=a_j+bt,\mathrm{rej}|\sum_S C_j+\sum_{\bar{S}}e_j$,$J=\{J_1,\cdots,J_4\}$,$a_j=\{2,3,4,5\}$,$e_j=\{6,4,30,35\}$,$[T_1,T_2]=[6,9)$。

初始条件:$F_1(2,0)=2$,$F_1(0,20)=29$,$F_1(0,0)=6$,否则 $F_1(u,v)=\infty$;$F_j^{(i)}(u,v)=\infty(i=1,2)$,其中 u,v 不是非负整数。

对于 $j=2$,依次计算可得:

$$F_2(2,21)=\min\{F_2^{(1)}(2,21),F_2^{(2)}(2,21),F_2^{(3)}(2,21)\}=32,$$
$$F_2(2,0)=\min\{F_2^{(1)}(2,0),F_2^{(2)}(2,0),F_2^{(3)}(2,0)\}=6,$$
$$F_2(3,20)=\min\{F_2^{(1)}(3,20),F_2^{(2)}(3,20),F_2^{(3)}(3,20)\}=32,$$
$$F_2(0,81)=\min\{F_2^{(1)}(0,81),F_2^{(2)}(0,81),F_2^{(3)}(0,81)\}=119,$$
$$F_2(0,20)=\min\{F_2^{(1)}(0,20),F_2^{(2)}(0,20),F_2^{(3)}(0,20)\}=33,$$
$$F_2(3,0)=\min\{F_2^{(1)}(3,0),F_2^{(2)}(3,0),F_2^{(3)}(3,0)\}=9,$$
$$F_2(0,21)=\min\{F_2^{(1)}(0,21),F_2^{(2)}(0,21),F_2^{(3)}(0,21)\}=36,$$
$$F_2(0,0)=\min\{F_2^{(1)}(0,0),F_2^{(2)}(0,0),F_2^{(3)}(0,0)\}=10。$$

J_3,J_4 类似。最后经过计算可得 $F_4(5,0)=45$ 为问题的最优值;加工工件为 J_4,拒绝工件集为 $\{J_1,J_2,J_3\}$。

下面分析动态规划算法(DP)的算法复杂度。因为 $j=1,2,\cdots,n$,$0\leqslant u\leqslant T_1$,$0\leqslant v\leqslant T_2(1+b)^n+D_n$,因此算法共有 $O(nT_1(T_2(1+b)^n+D_n))$ 种状态,对于固定的 j,u,v,计算 $F_j(u,v)$ 的复杂度为 $O(1)$,所以算法DP的复杂度为 $O(nT_1(T_2(1+b)^n+D_n))$。

定理1 动态规划算法(DP)的算法复杂度为 $O(nT_1(T_2(1+b)^n+D_n))$。

3 FPTAS

接下来构造问题 $1,h|nr-a,p_j=a_j+bt,\mathrm{rej}|\sum_S C_j+\sum_{\bar{S}}e_j$ 的一个FPTAS。

假设工件按照SPT序编号,使得 $a_1\leqslant a_2\leqslant\cdots\leqslant a_n$,引入变量 x_j,$j=1,2,\cdots,n$,令:

$$x_j=\begin{cases}1,\text{如果工件 } J_j \text{ 在 } T_1 \text{ 之前加工},\\2,\text{如果工件 } J_j \text{ 在 } T_2 \text{ 之后加工},\\3,\text{如果工件被拒绝。}\end{cases}$$

假设 X 为所有向量 $\boldsymbol{x}=(x_1,x_2,\cdots,x_n)$ 组成的向量集,其中 $x_j=k$,$j=1,2,\cdots,n$,$k=1,2,3$。定义关于 X 的初始函数和递推函数如下:

$$f_0^1(\boldsymbol{x})=0, f_0^2(\boldsymbol{x})=T_2, g_0(\boldsymbol{x})=0;$$
$$f_j^k(\boldsymbol{x})=f_{j-1}^k(\boldsymbol{x})+a_j+bf_{j-1}^k(\boldsymbol{x}), x_j=k, k=1,2;$$
$$f_j^i(\boldsymbol{x})=f_{j-1}^i(\boldsymbol{x}), x_j=i, i\neq k;$$
$$g_j(\boldsymbol{x})=g_{j-1}(\boldsymbol{x}), x_j=k, k=1,2;$$
$$g_j(\boldsymbol{x})=g_{j-1}(\boldsymbol{x})+e_j, x_j=3;$$
$$F_0(\boldsymbol{x})=0;$$
$$F_j(\boldsymbol{x})=F_{j-1}(\boldsymbol{x})+f_j^k(\boldsymbol{x}), x_j=k, k=1,2;$$
$$F_j(\boldsymbol{x})=F_{j-1}(\boldsymbol{x})+e_j, x_j=3;$$
$$P_0(\boldsymbol{x})=0;$$
$$P_j(\boldsymbol{x})=P_{j-1}(\boldsymbol{x})+a_j+bP_{j-1}(x), x_j=1;$$
$$P_j(\boldsymbol{x})=P_{j-1}(\boldsymbol{x}), x_j\neq 1, \boldsymbol{x}\in X。$$

式中：$f_j^k(\boldsymbol{x})(k=1,2)$分别表示工件 J_j 在 T_1 之前加工和在 T_2 之后加工的完工时间，$g_j(\boldsymbol{x})$表示工件 $J_1,J_2,\cdots,J_j$ 的总拒绝惩罚，$F_j(\boldsymbol{x})$表示工件 $J_1,J_2,\cdots,J_j$ 的目标函数值，$P_j(\boldsymbol{x})$表示 $J_1,J_2,\cdots,J_j$ 中在 T_1 之前加工工件的总完工时间，而且只需要关注使得 $P_j(\boldsymbol{x})\leqslant T_1$ 的向量 $\boldsymbol{x}$，则该向量 $\boldsymbol{x}$ 对应 $J_1,J_2,\cdots,J_j$ 的一个可行排序。值得注意的是，$f_j^k(\boldsymbol{x})$，$g_j(\boldsymbol{x})$，$F_j(\boldsymbol{x})$ 和 $P_j(\boldsymbol{x})$ 都与 $x_{j+1},x_{j+2},\cdots,x_n$ 无关。因此，问题 $1,h\,|\,nr-a, p_j=a_j+bt,\mathrm{rej}\,|\,\sum_S C_j+\sum_{\bar{S}}e_j$ 可以简化成下面的问题：

$$\min F_n(\boldsymbol{x}),$$
$$\text{s.t. } P_n(\boldsymbol{x})\leqslant T_1, \boldsymbol{x}\in X。$$

接下来介绍 Kovalyov 等人[26]提出的过程划分(A,h,δ)，其中 $A\subseteq X$，h 是关于 X 的一个非负整数函数，$0<\delta\leqslant 1$。将 A 划分成互不相交的子集 $A_1^h,A_2^h,\cdots,A_{k_h}^h$，使得对于同一子集 $A_j^h(j=1,2,\cdots,k_h)$中的任意 $\boldsymbol{x},\boldsymbol{x}'$，有 $|h(\boldsymbol{x})-h(\boldsymbol{x}')|\leqslant\delta\min\{h(\boldsymbol{x}),h(\boldsymbol{x}')\}$。过程划分$(A,h,\delta)$的具体步骤如下：

第 1 步，将向量 $\boldsymbol{x}\in X$ 按 $\boldsymbol{x}^{(1)},\boldsymbol{x}^{(2)},\cdots,\boldsymbol{x}^{(|A|)}$ 顺序排列，使得 $0\leqslant h(\boldsymbol{x}^{(1)})\leqslant h(\boldsymbol{x}^{(2)})\leqslant\cdots\leqslant h(\boldsymbol{x}^{(|A|)})$。

第 2 步，将向量 $\boldsymbol{x}^{(1)},\boldsymbol{x}^{(2)},\cdots,\boldsymbol{x}^{(i_1)}$ 分配到集合 A_1^h 中，直到找出满足 $h(\boldsymbol{x}^{(i_1)})\leqslant(1+\delta)h(\boldsymbol{x}^{(1)})$ 和 $h(\boldsymbol{x}^{(i_1+1)})>(1+\delta)h(\boldsymbol{x}^{(1)})$的 i_1。如果不存在满足条件的 i_1，那么令 $A_{k_h}^h=A_1^h=A$ 并停止。将向量 $\boldsymbol{x}^{(i_1+1)},\boldsymbol{x}^{(i_1+2)},\cdots,\boldsymbol{x}^{(i_2)}$ 分配到集合 A_2^h 中，直到找出满足 $h(\boldsymbol{x}^{(i_2)})\leqslant(1+\delta)h(\boldsymbol{x}^{(i_1+1)})$和 $h(\boldsymbol{x}^{(i_2+1)})>(1+\delta)h(\boldsymbol{x}^{(i_1+1)})$的 i_2。如果不存在满足条件的 i_2，那么令 $A_{k_h}^h=A_2^h=A-A_1^h$ 并停止。继续上述操作，直到存在某个 k_h，使得 $x^{(|A|)}$包含于 $A_{k_h}^h$。

过程划分将 A 中向量按照 $h(\boldsymbol{x})$非减顺序排列需要 $O(|A|\log|A|)$时间，产生一个划分需要 $O(|A|)$时间。下面的引理将在 FPTAS 中使用。

引理 5[26]　对于 $\forall\,\boldsymbol{x},\boldsymbol{x}'\in A_j^h, j=1,2,\cdots,k_h$，有 $|h(\boldsymbol{x})-h(\boldsymbol{x}')|\leqslant\delta\min\{h(\boldsymbol{x}),h(\boldsymbol{x}')\}$。

引理 6[26]　对于 $0<\delta\leqslant 1, h(\boldsymbol{x}^{(|A|)})\geqslant 1$，有 $k_h\leqslant\dfrac{\log h(\boldsymbol{x}^{|A|})}{\delta}+2$。

引理 7　对于 $\forall\,0<\boldsymbol{x}\leqslant 1, n\in\mathbf{Z}^+$，有 $\left(1+\dfrac{\boldsymbol{x}}{n}\right)^n\leqslant 1+2\boldsymbol{x}$。

证明　设 $f(\boldsymbol{x})=\left(1+\dfrac{\boldsymbol{x}}{n}\right)^n-1-2\boldsymbol{x}, \boldsymbol{x}\in(0,1]$，则 $f''(\boldsymbol{x})=\dfrac{n-1}{n}\left(1+\dfrac{\boldsymbol{x}}{n}\right)^{n-2}$。由于 $\boldsymbol{x}\in(0,1]$，所以 $f''(\boldsymbol{x})>0$，因此 $f(\boldsymbol{x})$是凸函数。又因为 $f(0)=0, f(1)=\left(1+\dfrac{1}{n}\right)^n-3<0$，故当 $\boldsymbol{x}\in(0,1]$时，$f(\boldsymbol{x})\leqslant 0$。即 $\left(1+\dfrac{\boldsymbol{x}}{n}\right)^n\leqslant 1+2\boldsymbol{x}$。

证毕

根据上述引理，给出问题 $1,h\,|\,nr-a, p_j=a_j+bt,\mathrm{rej}\,\Big|\,\sum_S C_j+\sum_{\bar{S}}e_j$ 的 FPTAS 算法如下。

算法 $\boldsymbol{A_\varepsilon}$　第 1 步，(初始化)将工件按照 SPT 序编号，使得 $a_1\leqslant a_2\leqslant\cdots\leqslant a_n$。

第 2 步，(构造集合 $Y_1,Y_2,\cdots,Y_n$)在集合 Y_{j-1} 中每个向量的第 j 个分量上添加 $k(k=1,2,3)$，从而构造出集合 Y_j'。对于 $\forall\,\boldsymbol{x}\in Y_j'$进行如下计算：当 $x_j=1$ 时，$P_j(\boldsymbol{x})=P_{j-1}(\boldsymbol{x})+a_j+bP_{j-1}(\boldsymbol{x})$；当 $x_j=2,3$ 时，$P_j(\boldsymbol{x})=P_{j-1}(\boldsymbol{x})$。若 $P_j(\boldsymbol{x})>T_1$，则从 Y_j'中删除 $\boldsymbol{x}$。则有：

$$f_j^k(\boldsymbol{x})=f_{j-1}^k(\boldsymbol{x})+a_j+bf_{j-1}^k(\boldsymbol{x}),x_j=k,k=1,2;$$

$$f_j^i(\boldsymbol{x})=f_{j-1}^i(\boldsymbol{x}),x_j=i,i\neq k;$$

$$g_j(\boldsymbol{x})=g_{j-1}(\boldsymbol{x}),x_j=k,k=1,2;$$

$$g_j(\boldsymbol{x})=g_{j-1}(\boldsymbol{x})+e_j,x_j=3;$$

$$F_j(\boldsymbol{x})=F_{j-1}(\boldsymbol{x})+f_j^k(\boldsymbol{x}),x_j=k,k=1,2;$$

$$F_j(\boldsymbol{x})=F_{j-1}(\boldsymbol{x})+e_j,x_j=3。$$

若 $j=n$,则令 $Y_n=Y_n'$,转到第 3 步。

若 $j<n$,则令 $\delta=\dfrac{\varepsilon}{2(n+1)}$,继续下面的计算。

用函数 f_j^i 将 Y_j'分成互不相交的集合 $Y_1^{f^i},Y_2^{f^i},\cdots,Y_{k_i}^{f^i}$,记为划分$(Y_j',f_j^i,\delta),i=1,2$;

用函数 g_j 将 Y_j'分成互不相交的集合 $Y_1^g,Y_2^g,\cdots,Y_{k_g}^g$,记为划分$(Y_j',g_j,\delta)$;

用函数 F_j 将 Y_j'分成互不相交的集合 $Y_1^F,Y_2^F,\cdots,Y_{k_F}^F$,记为划分$(Y_j',F_j,\delta)$。

将集合 Y_j'划分成互不相交的子集 $Y_{\alpha_1\alpha_2\alpha_3\beta}=Y_{\alpha_1}^{f^1}\cap Y_{\alpha_2}^{f^2}\cap Y_{\alpha_3}^g\cap Y_\beta^F,\alpha_1=1,2,\cdots,k_1,\alpha_2=1,2,\cdots,k_2,\alpha_3=1,2,\cdots,k_g,\beta=1,2,\cdots,k_F$。在每一个非空子集 $Y_{\alpha_1\alpha_2\alpha_3\beta}$ 中,选出向量 $\boldsymbol{x}^{(\alpha_1\alpha_2\alpha_3\beta)}$,使得 $P_j(\boldsymbol{x}^{(\alpha_1\alpha_2\alpha_3\beta)})=\min\{P_j(\boldsymbol{x})\mid\boldsymbol{x}\in Y_{\alpha_1\alpha_2\alpha_3\beta}\}$。

令 $Y_j:=\{\boldsymbol{x}^{(\alpha_1\alpha_2\alpha_3\beta)}\mid\alpha_1=1,\cdots,k_1;\alpha_2=1,\cdots,k_2;\alpha_3=1,\cdots,k_g;\beta=1,\cdots,k_F$ 且 $Y_{\alpha_1}^{f^1}\cap Y_{\alpha_2}^{f^2}\cap Y_{\alpha_3}^g\cap Y_\beta^F\neq 0\},j=j+1$。重复第 2 步。

第 3 步,(求解)选择向量 $\boldsymbol{x}^0\in Y_n$,使得 $F_n(\boldsymbol{x}^0)=\min\{F_n(\boldsymbol{x})\mid\boldsymbol{x}\in Y_n\}$。

4 算法复杂度

设 $\boldsymbol{x}^*=(x_1^*,x_2^*,\cdots,x_n^*)$是问题 $1,h\mid nr-a,p_j=a_j+bt,\mathrm{rej}\mid\sum_S C_j+\sum_{\bar{S}}e_j$ 的最优解,$L=\log(\max\{n,T_2,1/\varepsilon,\alpha_{\max},b,e_{\max}\})$,其中 $\alpha_{\max}=\max\limits_{1\leqslant j\leqslant n}\{\alpha_j\},e_{\max}=\max\limits_{1\leqslant j\leqslant n}\{e_j\}$。

定理 2 对于问题 $1,h\mid nr-a,p_j=a_j+bt,\mathrm{rej}\mid\sum_S C_j+\sum_{\bar{S}}e_j$,算法 A_ε 在$O(n^7L^5/\varepsilon^4)$时间内找到向量 $\boldsymbol{x}^0\in X$,使得 $P_n(\boldsymbol{x}^0)\leqslant T_1$ 且 $F_n(\boldsymbol{x}^0)\leqslant(1+\varepsilon)F_n(\boldsymbol{x}^*)$。

证明 设 $\boldsymbol{x}^*[j]=(x_1^*,x_2^*,\cdots,x_j^*,0,\cdots,0)\in Y_{\alpha_1\alpha_2\alpha_3\beta}\subseteq Y_j'$,对于 $j,\alpha_1,\alpha_2,\alpha_3,\beta$,根据 A_ε 的定义,这样的 j 总是存在,例如 $j=1$。但算法 A_ε 可能不会选择 $\boldsymbol{x}^*[j]$继续构造,而是选择使得 $P_j(\boldsymbol{x}^{(\alpha_1\alpha_2\alpha_3\beta)})=\min\{P_j(\boldsymbol{x})\mid\boldsymbol{x}\in Y_{\alpha_1\alpha_2\alpha_3\beta}\}$的向量 $\boldsymbol{x}^{(\alpha_1\alpha_2\alpha_3\beta)}$ 代替 $\boldsymbol{x}^*[j]$,则有 $P_j(\boldsymbol{x}^{(\alpha_1\alpha_2\alpha_3\beta)})\leqslant P_j(\boldsymbol{x}^*[j])\leqslant T_1$,又因为 $Y_{\alpha_1\alpha_2\alpha_3\beta}=Y_{\alpha_1}^{f^1}\cap Y_{\alpha_2}^{f^2}\cap Y_{\alpha_3}^g\cap Y_\beta^F$ 且 $\boldsymbol{x}^{(\alpha_1\alpha_2\alpha_3\beta)},\boldsymbol{x}^*[j]\in Y_{\alpha_1\alpha_2\alpha_3\beta}$,所以 $\boldsymbol{x}^{(\alpha_1\alpha_2\alpha_3\beta)},\boldsymbol{x}^*[j]\in Y_{\alpha_1}^{f^1},\boldsymbol{x}^{(\alpha_1\alpha_2\alpha_3\beta)},\boldsymbol{x}^*[j]\in Y_{\alpha_2}^{f^2},\boldsymbol{x}^{(\alpha_1\alpha_2\alpha_3\beta)},\boldsymbol{x}^*[j]\in g,\boldsymbol{x}^{(\alpha_1\alpha_2\alpha_3\beta)},\boldsymbol{x}^*[j]\in F$。

根据引理 5 可得:

$$|f_j^i(\boldsymbol{x}^*[j])-f_j^i(\boldsymbol{x}^{(\alpha_1\alpha_2\alpha_3\beta)})|\leqslant\delta f_j^i(\boldsymbol{x}^*[j]),i=1,2,$$

$$|g_j(\boldsymbol{x}^*[j])-g_j(\boldsymbol{x}^{(\alpha_1\alpha_2\alpha_3\beta)})|\leqslant\delta g_j(\boldsymbol{x}^*[j]),|F_j(\boldsymbol{x}^*[j])-F_j(\boldsymbol{x}^{(\alpha_1\alpha_2\alpha_3\beta)})|\leqslant\delta F_j(\boldsymbol{x}^*[j])。$$

令 $\delta_1=\delta$,考虑向量 $\boldsymbol{x}^*[j+1]=(x_1^*,x_2^*,\cdots,x_j^*,x_{j+1}^*,0,\cdots,0)$和 $\tilde{\boldsymbol{x}}^{(\alpha_1\alpha_2\alpha_3\beta)}=(x_1^{(\alpha_1\alpha_2\alpha_3\beta)},x_2^{(\alpha_1\alpha_2\alpha_3\beta)},\cdots,x_j^{(\alpha_1\alpha_2\alpha_3\beta)},x_{j+1}^*,0,\cdots,0)$。不失一般性,假设 $x_{j+1}^*=k$,则 $P_{j+1}(\boldsymbol{x}^{(\alpha_1\alpha_2\alpha_3\beta)})\leqslant P_j(\boldsymbol{x}^*[j+1])\leqslant T_1$。

若 $k=1$,则:

$$|f_{j+1}^1(\boldsymbol{x}^*[j+1])-f_{j+1}^1(\tilde{\boldsymbol{x}}^{(\alpha_1\alpha_2\alpha_3\beta)})|=|f_j^1(\boldsymbol{x}^*[j])+a_j+bf_j^1(\boldsymbol{x}^*[j])-(f_j^1(\boldsymbol{x}^{(\alpha_1\alpha_2\alpha_3\beta)})+a_j+bf_j^1(\boldsymbol{x}^{(\alpha_1\alpha_2\alpha_3\beta)})|=$$

$$|(1+b)(f_j^1(\boldsymbol{x}^*[j])-f_j^1(\boldsymbol{x}^{(\alpha_1\alpha_2\alpha_3\beta)}))|\leqslant\delta(1+b)f_j^1(\boldsymbol{x}^*[j])\leqslant\delta_1 f_{j+1}^1(\boldsymbol{x}^*[j+1]);$$

$$|f_{j+1}^2(\boldsymbol{x}^*[j+1])-f_{j+1}^2(\tilde{\boldsymbol{x}}^{(\alpha_1\alpha_2\alpha_3\beta)})|=|f_j^2(\boldsymbol{x}^*[j])-f_j^2(\boldsymbol{x}^{(\alpha_1\alpha_2\alpha_3\beta)})|\leqslant\delta f_j^2(\boldsymbol{x}^*[j])=\delta_1 f_{j+1}^2(\boldsymbol{x}^*[j+1]);$$

$$|g_{j+1}(\boldsymbol{x}^*[j+1])-g_{j+1}(\tilde{\boldsymbol{x}}^{(\alpha_1\alpha_2\alpha_3\beta)})|\leqslant|g_j(\boldsymbol{x}^*[j])-g_j(\boldsymbol{x}^{(\alpha_1\alpha_2\alpha_3\beta)})|\leqslant\delta g_j(\boldsymbol{x}^*[j])=\delta_1 g_{j+1}(\boldsymbol{x}^*[j+1]);$$

$$|F_{j+1}(\boldsymbol{x}^*[j+1])-F_{j+1}(\tilde{\boldsymbol{x}}^{(\alpha_1\alpha_2\alpha_3\beta)})|=|F_j(\boldsymbol{x}^*[j])+f_{j+1}^1(\boldsymbol{x}^*[j+1])-(F_j(\boldsymbol{x}^{(\alpha_1\alpha_2\alpha_3\beta)})+f_{j+1}^1(\tilde{\boldsymbol{x}}^{(\alpha_1\alpha_2\alpha_3\beta)}))|\leqslant$$

$$|F_j(\boldsymbol{x}^*[j])-F_j(\boldsymbol{x}^{(\alpha_1\alpha_2\alpha_3\beta)})|+|f_{j+1}^1(\boldsymbol{x}^*[j+1])-f_{j+1}^1(\tilde{\boldsymbol{x}}^{(\alpha_1\alpha_2\alpha_3\beta)})|\leqslant$$
$$\delta_1F_j(\boldsymbol{x}^*[j])+\delta_1f_{j+1}^1(\boldsymbol{x}^*[j+1])=\delta_1F_{j+1}(\boldsymbol{x}^*[j+1])。$$

若 $k=2$，则：

$$|f_{j+1}^1(\boldsymbol{x}^*[j+1])-f_{j+1}^1(\tilde{\boldsymbol{x}}^{(\alpha_1\alpha_2\alpha_3\beta)})|=|f_j^1(\boldsymbol{x}^*[j])-f_j^1(\boldsymbol{x}^{(\alpha_1\alpha_2\alpha_3\beta)})|\leqslant\delta f_j^1(\boldsymbol{x}^*[j])=\delta_1f_{j+1}^1(\boldsymbol{x}^*[j+1])；$$
$$|f_{j+1}^2(\boldsymbol{x}^*[j+1])-f_{j+1}^2(\tilde{\boldsymbol{x}}^{(\alpha_1\alpha_2\alpha_3\beta)})|=|f_j^2(\boldsymbol{x}^*[j])+a_j+bf_j^2(\boldsymbol{x}^*[j])-(f_j^2(\boldsymbol{x}^{(\alpha_1\alpha_2\alpha_3\beta)})+a_j+bf_j^2(\boldsymbol{x}^{(\alpha_1\alpha_2\alpha_3\beta)}))|=$$
$$|(1+b)(f_j^2(\boldsymbol{x}^*[j])-f_j^2(\boldsymbol{x}^{(\alpha_1\alpha_2\alpha_3\beta)}))|\leqslant\delta(1+b)f_j^2(\boldsymbol{x}^*[j])\leqslant\delta_1f_{j+1}^2(\boldsymbol{x}^*[j+1])；$$
$$|g_{j+1}(\boldsymbol{x}^*[j+1])-g_{j+1}(\tilde{\boldsymbol{x}}^{(\alpha_1\alpha_2\alpha_3\beta)})|\leqslant|g_j(\boldsymbol{x}^*[j])-g_j(\boldsymbol{x}^{(\alpha_1\alpha_2\alpha_3\beta)})|\leqslant\delta g_j(\boldsymbol{x}^*[j])=\delta_1g_{j+1}(\boldsymbol{x}^*[j+1])；$$
$$|F_{j+1}(\boldsymbol{x}^*[j+1])-F_{j+1}(\tilde{\boldsymbol{x}}^{(\alpha_1\alpha_2\alpha_3\beta)})|=|F_j(\boldsymbol{x}^*[j])+f_{j+1}^2(\boldsymbol{x}^*[j+1])-(F_j(\boldsymbol{x}^{(\alpha_1\alpha_2\alpha_3\beta)})+f_{j+1}^2(\tilde{\boldsymbol{x}}^{(\alpha_1\alpha_2\alpha_3\beta)}))|\leqslant$$
$$|F_j(\boldsymbol{x}^*[j])-F_j(\boldsymbol{x}^{(\alpha_1\alpha_2\alpha_3\beta)})|+|f_{j+1}^2(\boldsymbol{x}^*[j+1])-f_{j+1}^2(\tilde{\boldsymbol{x}}^{(\alpha_1\alpha_2\alpha_3\beta)})|\leqslant$$
$$\delta_1F_j(\boldsymbol{x}^*[j])+\delta_1f_{j+1}^2(\boldsymbol{x}^*[j+1])=\delta_1F_{j+1}(\boldsymbol{x}^*[j+1])。$$

若 $k=3$，则：

$$|f_{j+1}^1(\boldsymbol{x}^*[j+1])-f_{j+1}^1(\tilde{\boldsymbol{x}}^{(\alpha_1\alpha_2\alpha_3\beta)})|=|f_j^1(\boldsymbol{x}^*[j])-f_j^1(\boldsymbol{x}^{(\alpha_1\alpha_2\alpha_3\beta)})|\leqslant\delta f_j^1(\boldsymbol{x}^*[j])=\delta_1f_{j+1}^1(\boldsymbol{x}^*[j+1])；$$
$$|f_{j+1}^2(\boldsymbol{x}^*[j+1])-f_{j+1}^2(\tilde{\boldsymbol{x}}^{(\alpha_1\alpha_2\alpha_3\beta)})|=|f_j^2(\boldsymbol{x}^*[j])-f_j^2(\boldsymbol{x}^{(\alpha_1\alpha_2\alpha_3\beta)})|\leqslant\delta f_j^2(\boldsymbol{x}^*[j])=\delta_1f_{j+1}^2(\boldsymbol{x}^*[j+1])；$$
$$|g_{j+1}(\boldsymbol{x}^*[j+1])-g_{j+1}(\tilde{\boldsymbol{x}}^{(\alpha_1\alpha_2\alpha_3\beta)})|\leqslant|g_j(\boldsymbol{x}^*[j])+e_{j+1}-(g_{j+1}(\boldsymbol{x}^{(\alpha_1\alpha_2\alpha_3\beta)})+e_{j+1})|\leqslant$$
$$\delta g_j(\boldsymbol{x}^*[j])\leqslant\delta_1g_{j+1}(\boldsymbol{x}^*[j+1])；$$
$$|F_{j+1}(\boldsymbol{x}^*[j+1])-F_{j+1}(\tilde{\boldsymbol{x}}^{(\alpha_1\alpha_2\alpha_3\beta)})|=|F_j(\boldsymbol{x}^*[j])+e_{j+1}-(F_j(\boldsymbol{x}^{(\alpha_1\alpha_2\alpha_3\beta)})+e_{j+1})|\leqslant$$
$$|F_j(\boldsymbol{x}^*[j])-F_j(\boldsymbol{x}^{(\alpha_1\alpha_2\alpha_3\beta)})|\leqslant\delta F_j(\boldsymbol{x}^*[j])\leqslant\delta_1F_{j+1}(\boldsymbol{x}^*[j+1])。$$

对于 $x_{j+1}^*=k,k=1,2,3$，有：

$$|f_{j+1}^i(\boldsymbol{x}^*[j+1])-f_{j+1}^i(\tilde{\boldsymbol{x}}^{(\alpha_1\alpha_2\alpha_3\beta)})|\leqslant\delta_1f_{j+1}^i(\boldsymbol{x}^*[j+1]),i=1,2；$$
$$|g_{j+1}(\boldsymbol{x}^*[j+1])-g_{j+1}(\tilde{\boldsymbol{x}}^{(\alpha_1\alpha_2\alpha_3\beta)})|\leqslant\delta_1g_{j+1}(\boldsymbol{x}^*[j+1])；$$
$$|F_{j+1}(\boldsymbol{x}^*[j+1])-F_{j+1}(\tilde{\boldsymbol{x}}^{(\alpha_1\alpha_2\alpha_3\beta)})|\leqslant\delta_1F_{j+1}(\boldsymbol{x}^*[j+1])。$$

因此有：

$$f_{j+1}^i(\tilde{\boldsymbol{x}}^{(\alpha_1\alpha_2\alpha_3\beta)})\leqslant(1+\delta_1)f_{j+1}^i(\boldsymbol{x}^*[j+1]),i=1,2;g_{j+1}(\tilde{\boldsymbol{x}}^{(\alpha_1\alpha_2\alpha_3\beta)})\leqslant(1+\delta_1)g_{j+1}(\boldsymbol{x}^*[j+1])；$$
$$F_{j+1}(\tilde{\boldsymbol{x}}^{(\alpha_1\alpha_2\alpha_3\beta)})\leqslant(1+\delta_1)F_{j+1}(\boldsymbol{x}^*[j+1])。\tag{1}$$

假设 $\tilde{\boldsymbol{x}}^{(\alpha_1\alpha_2\alpha_3\beta)}\in Y_{c_1c_2c_3d}\subseteq Y'_{j+1}$，算法 A_ε 在第 $j+1$ 次迭代中选择 $\boldsymbol{x}^{c_1c_2c_3d}\in Y_{c_1c_2c_3d}$ 替代 $\tilde{\boldsymbol{x}}^{(\alpha_1\alpha_2\alpha_3\beta)}$，通过上述分析，同理可以推导出：

$$P_{j+1}(\boldsymbol{x}^{(c_1c_2c_3d)})\leqslant P_{j+1}(\tilde{\boldsymbol{x}}^{(\alpha_1\alpha_2\alpha_3\beta)})\leqslant P_{j+1}(\boldsymbol{x}^*[j+1])\leqslant T_1；$$
$$|f_{j+1}^i(\tilde{\boldsymbol{x}}^{(\alpha_1\alpha_2\alpha_3\beta)})-f_{j+1}^i(\boldsymbol{x}^{(c_1c_2c_3d)})|\leqslant\delta f_{j+1}^i(\tilde{\boldsymbol{x}}^{(\alpha_1\alpha_2\alpha_3\beta)})\leqslant\delta(1+\delta_1)f_{j+1}^i(\boldsymbol{x}^*[j+1]),i=1,2；$$
$$|g_{j+1}(\tilde{\boldsymbol{x}}^{(\alpha_1\alpha_2\alpha_3\beta)})-g_{j+1}(\boldsymbol{x}^{(c_1c_2c_3d)})|\leqslant\delta g_{j+1}(\tilde{\boldsymbol{x}}^{(\alpha_1\alpha_2\alpha_3\beta)})\leqslant\delta(1+\delta_1)g_{j+1}(\boldsymbol{x}^*[j+1])；$$
$$|F_{j+1}(\tilde{\boldsymbol{x}}^{(\alpha_1\alpha_2\alpha_3\beta)})-F_{j+1}(\boldsymbol{x}^{(c_1c_2c_3d)})|\leqslant\delta F_{j+1}(\tilde{\boldsymbol{x}}^{(\alpha_1\alpha_2\alpha_3\beta)})\leqslant\delta(1+\delta_1)F_{j+1}(\boldsymbol{x}^*[j+1])。\tag{2}$$

由式(1)、(2)可知：

$$|F_{j+1}(\boldsymbol{x}^*[j+1])-F_{j+1}(\boldsymbol{x}^{(c_1c_2c_3d)})|\leqslant|F_{j+1}(\boldsymbol{x}^*[j+1])-F_{j+1}(\tilde{\boldsymbol{x}}^{(\alpha_1\alpha_2\alpha_3\beta)})|+|F_{j+1}(\tilde{\boldsymbol{x}}^{(\alpha_1\alpha_2\alpha_3\beta)})-F_{j+1}(\boldsymbol{x}^{(c_1c_2c_3d)})|\leqslant$$
$$[\delta_1+\delta(1+\delta_1)]F_{j+1}(\boldsymbol{x}^*[j+1])\leqslant[\delta+\delta_1(1+\delta)]F_{j+1}(\boldsymbol{x}^*[j+1])。\tag{3}$$

设 $\delta_l=\delta+\delta_{l-1}(1+\delta),l=2,3,\cdots,n-j+1$。由式(3)可知：

$$|F_{j+1}(\boldsymbol{x}^*[j+1])-F_{j+1}(\boldsymbol{x}^{(c_1c_2c_3d)})|\leqslant\delta_2F_{j+1}(\boldsymbol{x}^*[j+1])。$$

对 $j+2,\cdots,n$ 重复上述论证，可以证明存在 $\boldsymbol{x}'\in Y_n$，使得 $P_n(\boldsymbol{x}')\leqslant P_n(\boldsymbol{x}^*)\leqslant T_1$ 且 $|F_n(\boldsymbol{x}^*)-F_n(\boldsymbol{x}')|\leqslant\delta_{n-j+1}F_n(\boldsymbol{x}^*)$。

由引理 7 可知：

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

因此 $|F_n(\boldsymbol{x}^*)-F_n(\boldsymbol{x}')|\leqslant\varepsilon F_n(\boldsymbol{x}^*)$。

那么在算法 A_ε 的第 3 步中,将选出满足 $P_n(\boldsymbol{x}^0)\leqslant P_n(\boldsymbol{x}')\leqslant T_1$ 且 $F_n(\boldsymbol{x}^0)\leqslant F_n(\boldsymbol{x}')\leqslant(1+\varepsilon)F_n(\boldsymbol{x}^*)$的向量 $\boldsymbol{x}^0$。

算法 A_ε 的时间复杂度可以通过最耗时的第 2 步过程划分的第 j 次迭代来确定,需要 $O(|Y_j'|\log|Y_j'|)$时间完成,通过 $|Y_{j+1}'|\leqslant 3|Y_j'|\leqslant 3k_1k_2k_gk_F$ 估计 $|Y_j'|$。根据引理 6 可以得到:

$$K_1\leqslant\frac{2(n+1)\log T_1}{\varepsilon}+2\leqslant\frac{2(n+1)L}{\varepsilon}+2,$$

$$K_2\leqslant\frac{2(n+1)\log(T_2+n\alpha_{\max})(1+b)^n}{\varepsilon}+2\leqslant\frac{2(n+1)[\log(T_2+n\alpha_{\max})+n\log(1+b)]}{\varepsilon}+2\leqslant$$
$$\frac{2(n+1)[\log(2T_2+2n\alpha_{\max})+n\log(1+b)]}{\varepsilon}+2\leqslant\frac{2(n+1)^2L}{\varepsilon}+2,$$

$$K_g\leqslant\frac{2(n+1)\log(ne_{\max})}{\varepsilon}+2\leqslant\frac{4(n+1)L}{\varepsilon}+2,$$

$$K_F\leqslant\frac{2(n+1)\log(\max\{n\alpha_{\max}(1+b)^n,ne_{\max}\})}{\varepsilon}+2\leqslant\frac{2(n+1)^2L}{\varepsilon}+2。$$

因此 $|Y_j'|=O(n^6L^4/\varepsilon^4)$且 $O(|Y_j'|\log|Y_j'|)=O(n^6L^5/\varepsilon^4)$,则算法 A_ε 的时间复杂度为 $O(n^7L^5/\varepsilon^4)$。 证毕

5 结论

本文研究了带有退化、拒绝和不可用区间的单机排序问题,目标是极小化接受工件的总完工时间与被拒绝工件的拒绝惩罚之和。每个工件有不同的基础加工时间和相同的退化率,且考虑工件是不可恢复的。给出了一个拟多项式时间动态规划算法,并构造出一个时间复杂度为 $O(n^7L^5/\varepsilon^4)$的 FPTAS 算法。在今后的研究中,可以选择在相同的条件下研究不同的目标函数,例如最大延误、误工任务数等等。也可以选择将这个问题扩展到不同的机器环境下,例如流水作业等。

参考文献:

[1] LEE C Y. Machine scheduling with an availability constraints[J]. Journal of Global Optimization,1996,9:363-382.

[2] MA Y,CHU C B,ZUO C R. A survey of scheduling with deterministic machine availability constraints[J]. Computers and Industrial Engineering,2010,58:199-211.

[3] ZHAO C L, JI M, TANG H Y. Parallel-machine scheduling with an availability constraint[J]. Computer and Industrial Engineering,2011,61:778-781.

[4] 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.

[5] KACEM I,KELLERER H. Approximation schemes for minimizing the maximum lateness on a single machine with release times under non-availability or deadline constraints[J]. Algorithmica,2018,80(12):3825-3843.

[6] 金苗苗,吴蒙洁,罗文昌. 具有不可用区间且工件可拒绝下的单机重新排序问题的近似方案[J]. 运筹与管理,2021,30(8):87-92.
JIN M M,WU M J,LUO W C. Approximation scheme for single machine rescheduling problem with an unavailable interval and job rejection[J]. Operations Research and Management Science,2021,30(8):87-92.

[7] GAWIEJNOWICZ S. A review of four decades of time-dependent scheduling: main results, new topics, and open problems[J]. Journal of Scheduling,2020,23(1):3-47.

[8] KUO W H,YANG D L. Parallel-machine scheduling with time dependent processing times[J]. Theoretical Computer Science, 2008,393:204-210.

[9] LI S S, NG C T, YUAN J J. Scheduling deteriorating jobs with CON/SLK due date assignment on a single machine[J]. International Journal of Production,2011,131:747-751.

[10] ZHAO C L,TANG H Y. Scheduling deteriorating jobs under disruption[J]. International Journal of Production,2010,125:294-299.

[11] CHENG T C E,KANG L Y,NG C T. Due-date assignment and single machine scheduling with deteriorating jobs[J]. Journal of the Operational Research Society,2004,55:198-203.

[12] CHENG T C E,KANG L Y,NG C T. Due-date assignment and parallel-machine scheduling with deteriorating jobs[J]. Journal

of the Operational Research Society,2007,58:1103-1108.
[13] LEE W C,WU C C,CHUNG Y H. Scheduling deteriorating jobs on a single machine with release times[J]. Computer and Industrial Engineering,2008,54:441-452.
[14] ZHAO C L,HSU C J. Scheduling deteriorating jobs with machine availability constraints to minimize the total completion time [J]. Journal of Industrial and Production Engineering,2017,34(5):323-329.
[15] 王吉波,梁茜茜,张博. 带有学习与恶化效应的共同工期指派问题[J]. 重庆师范大学学报(自然科学版),2019,36(3):1-6.
WANG J B,LIANG X X,ZHANG B. Common due date assignment problem with learning and deterioration effects[J]. Journal of Chongqing Normal University (Natural Science),2019,36(3):1-6.
[16] BARTAL Y, LEONARDI S, MARCHETTI-SPACCAMELA A, et al. Multiprocessor scheduling with rejection[J]. SIAM Journal of Discrete Mathematics,2000,13(1):64-78.
[17] ENGELS D W, KARGER D R, KOLLIOPOULOS S G, et al. Technique for scheduling with rejection[J]. Journal of Algorithms,2003,49(1):175-191.
[18] CHENG Y S, SUN S J. Scheduling linear deteriorating jobs with rejection on a single machine[J]. European Journal of Operations Research,2009,194:18-27.
[19] ZHAO C L,TANG H Y. Single machine scheduling with an availability constraint and rejection[J]. Asia-Pacific Journal of Operational Research,2014,31(5):1450037.
[20] 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(1/2):585-605.
[21] 国峰,王吉波. 带有拒绝工件和学习效应的资源约束排序问题研究[J]. 重庆师范大学学报(自然科学版),2021,38(1):114-120.
GUO F,WANG J B. Research on resource constraint scheduling with job rejection and learning effect[J]. Journal of Chongqing Normal University (Natural Science),2021,38(1):114-120.
[22] 徐晨,徐寅峰,郑斐峰,等. 工件可拒绝的单机多任务排序问题研究[J]. 系统科学与数学,2022,42(8):2198-2206.
XU C,XU Y F,ZHENG F F,et al. Multitasking scheduling problems with job rejection on single machine[J]. Journal of Systems Science and Mathematical Sciences,2022,42(8):2198-2206.
[23] 林凌,谈之奕. 平行机在线排序综述[J]. 中国科学:数学,2020,50(9):1183-1200.
LIN L,TAN Z Y. Online scheduling on parallel machines:a survey[J]. Scientia Sinica (Mathematica),2020,50(9):1183-1200.
[24] 张玉忠. 工件可拒绝排序问题综述[J]. 运筹学学报,2020,24(2):111-130.
ZHANG Y Z. A survey on job scheduling with rejection[J]. Operations Research Transactions,2020,24(2):111-130.
[25] GAWIEJNOWICZ S,PANKOWSKA L. Scheduling jobs with varying processing times[J]. Information Processing Letters, 1995,54:175-178.
[26] KOVALYOV M Y,KUBIAK W. A fully polynomial approximation scheme for minimizing makespan of deteriorating jobs[J]. Journal of Heuristics,1998,3:287-297.

Operations Research and Cybernetics

Deteriorating Jobs Scheduling on a Single Machine with Rejection and a Fixed Non-Availability Interval

HE Xinyi, ZHAO Yufang, CHEN Zhuangzhuang

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

Abstract: [**Purposes**]Single machine scheduling with deteriorating jobs, rejection and availability constraints are considered. [**Methods**]It is assumed that jobs have a different basic processing time and same deterioration rate, jobs can be rejected by paying penalties, the machines may be unavailable in a specified time intervals and jobs are non-resumable. The objective is to minimize the sum of the total completion time of the accepted jobs and the total rejection penalty of the rejected job. [**Results**]For the NP-hard problem, the optimal solution can be obtained by sequencing in non-decreasing order of a_j of the jobs before and after the unavailable interval. A pseudo-polynomial-time dynamic programming and a fully polynomial-time approximation scheme are designed. [**Conclusions**]The existing models are generalized.

Keywords: single-machine scheduling; deteriorating jobs; rejection; non-availability interval

(责任编辑 黄 颖)