2013 年 1 月 重庆师范大学学报(自然科学版) Jan. 2013
第 30 卷 第 1 期 Journal of Chongqing Normal University (Natural Science) Vol. 30 No. 1

运筹学与控制论

DOI:10.11721/cqnuj20130104

带到达时间、不可用区间、拒绝工件的单机排序问题*

刘 澈,罗成新
(沈阳师范大学 数学与系统科学学院,沈阳 110034)

摘要:考虑的是带有到达时间、拒绝工件、不可用区间的单机排序问题。若工件被拒绝加工,厂家必须支付一定的拒绝惩罚;若工件被接受,则把工件放在机器上进行加工。机器带有不可用区间,在不可用区间内不能加工工件,并且在同一时刻至多加工一个工件。本文的目标函数是极小化所有接受工件的时间表长与所有拒绝工件的拒绝惩罚之和。首先给出了一个近似算法,并通过引理 1 证明出此算法是 3-因子算法;其次提出了一个动态规划算法,然后通过修改这个动态规划算法的执行过程来减少运行时间,进而得到了一个全多项式时间近似方案,证明出该方案的时间复杂性为 $O\left(\frac{n^2}{\varepsilon}\right)$。

关键词:到达时间;拒绝工件;不可用区间;时间表长
中图分类号:O223 文献标志码:A 文章编号:1672-6693(2013)01-0017-04

在经典排序问题中,所有的工件必须被接受,并且在某些机器上进行加工,即拒绝工件是不允许的。然而在实际生产中,厂家为了获得更大的利润,通常会拒绝那些加工时间长且获利相对小的工件,也因此得到一定的惩罚。在加工工件时,通常也会因为机器故障、工人问题等其他原因造成机器在某一时间段内不能加工工件,即产生了不可用区间。由于上述问题的实用性,带有拒绝工件及不可用区间的排序问题得到了越来越多人的关注[1-9]。

带有到达时间、拒绝工件、不可用区间的单机排序问题 P 描述如下:n 个互不相关的工件在一台机器上加工,p_j 是第 j 个工件的加工时间,r_j 为第 j 个工件的到达时间,即释放时间,$w_j>0$ 是工件被拒绝时的惩罚,即若工件 J_j 被拒绝,那么厂家必须进行拒绝赔偿 w_j。$[T_1,T_2]$ 为不可用区间,在此区间内机器不能加工工件。本文假设 $\sum_{j=1}^{n}p_j>T_1$,即并不是所有工件都可以在 T_1 前完工,否则区间 $[T_1,T_2]$ 不起作用。不失一般性,假设 p_j,r_j,w_j 都是整数。目标函数是极小化接受工件的时间表长与所有拒绝工件惩罚的和。记所有被拒绝的工件的集合为 R,那么此排序问题可记为 $1,h|r_j,reject|C_{\max}+\sum_{j\in R}w_j$。

文献[2]证明出带有到达时间、拒绝工件的单机排序问题是 NP 难的,本文在此基础上加了一个不可用区间。本文第二部分给出了一个 3-因子算法 H,此算法得到的目标值给出了问题 P 的最优目标值的一个上界;第三部分给出了一个动态规划算法 A;第四部分给出了一个全多项式时间近似方案,其时间复杂性为 $O\left(\frac{n^2}{\varepsilon}\right)$。

1 3-因子算法 H

假设 S 是一个工件的集合,定义 $P(S)=\sum_{J_j\in S}p_j$,$W(S)=\sum_{J_j\in S}w_j$ 分别为工件的总加工时间和总拒绝惩罚,$[T_1,T_2]$ 是机器的不可用区间。以下首先给出一个 3-因子算法 H,其运行时间是 $O(n^2)$。

算法 H 第 1 步,对于每一个 $t\in\{r_j,j=1,2,\cdots,n\}$,将工件分为 3 个集合,使得 $S_1(t)=\{J_j:r_j\leq t,p_j\leq w_j\}$,$S_2(t)=\{J_j:r_j\leq t,p_j>w_j\}$,$S_3(t)=\{J_j:r_j>t\}$;

第 2 步,对 $S_1(t)$ 中所有的工件进行加工,$S_2(t)\cup S_3(t)$ 中所有的工件被拒绝,将 $S_1(t)$ 中的工件按 r_j 不减排序,其中 $J_j\in S_1(t)$。若 $\max\{t,r_j\}+p_j\leq T_1$,$J_j$ 排在 T_1 之前;若 $\max\{t,r_j\}+p_j>T_1$,J_j 排在 T_2 之后,这样得到的排

* 收稿日期:2012-04-15 修回日期:2012-06-27 网络出版时间:2013-01-18 15:05
资助项目:国家自然科学基金(No. 11171050)
作者简介:刘澈,女,硕士研究生,研究方向为应用数学,E-mail:cheche_7828845@126.com;通讯作者:罗成新,E-mail:luochengxin@163.com
网络出版地址:http://www.cnki.net/kcms/detail/50.1165.N.20130118.1505.201301.17_003.html

序定义为 $\pi(t)$;

第3步,令 $Z(t)$ 为排序 $\pi(t)$ 所得的目标函数值,即 $Z(t)$ 为加工工件的时间表长与拒绝工件的惩罚之和。由以上所得的所有 $\pi(t)$ 中,找到使 $Z(t)$ 最小的排序 $\pi(t)$,记为 π。设 π^* 是问题P的最优排序。

引理1 设 Z,Z^* 分别是 π,π^* 所得的目标值,则 $Z\leqslant 3Z^*$。

证明 令 A^*,R^* 分别是排序 π^* 所得的加工工件和拒绝工件的集合,$r^*=\max\{r_j:J_j\in A^*\}$。由 r^* 的定义有 $S_3(r^*)=\{J_j:r_j>r^*\}\subset R^*$,那么

$$Z^*\geqslant r^*+W(S_3(r^*)),Z^*\geqslant P(A^*)+W(R^*)\geqslant\sum_{J_j\in A^*}\min\{p_j,w_j\}+\sum_{J_j\in R^*}\min\{p_j,w_j\}=\sum_{j=1}^{n}\min\{p_j,w_j\}$$

显然 $T_2-T_1\leqslant Z^*$,因此有

$$Z\leqslant Z(R^*)=r^*+P(S_1(r^*))+T_2-T_1+W(S_2(r^*))+W(S_3(r^*))\leqslant r^*+\sum_{J_j\in S_1(r^*)}\min\{p_j,w_j\}+T_2-T_1+\sum_{J_j\in S_2(r^*)}\min\{p_j,w_j\}+W(S_3(r^*))\leqslant r^*+W(S_3(r^*))+\sum_{j=1}^{n}\min\{p_j,w_j\}+T_2-T_1\leqslant 3Z^*$$

证毕

2 动态规划算法A

用 VS_k 表示一个状态集,$[t,f]$ 表示 VS_k 中的一个状态向量,其中 t 表示 T_1 之前接受工件中最后一个工件的完工时间,f 表示是目标函数,即时间表长与被拒绝工件的拒绝惩罚之和,w 表示前 $k-1$ 个工件中被拒绝的工件的惩罚之和,w_k 表示第 k 个工件的拒绝惩罚。

对于下面的算法A,$\max\{r_k,T_2\}+p_k+w$ 表示的是第 k 个工件在 T_2 后加工,并且前 $k-1$ 个工件中接受的工件均在 T_1 之前加工的情况下的目标值;$\max\{f-w,r_k\}+p_k+w$ 表示的是第 k 个工件在 T_2 后加工,但是前 $k-1$ 个工件中接受的工件已有排在 T_2 之后的情况下的目标值;$\max\{r_k,t\}+p_k+w$ 表示的是第 k 个工件在 T_1 前加工,并且前 $k-1$ 个工件中接受的工件均排在 T_1 之前加工的情况下的目标值。

算法A 第1步,置 $VS_1=\{[0,w_1],[r_1+p_1,r_1+p_1],[0,\max\{r_1,T_2\}+p_1]\}$;

第2步,当 $k\in\{2,3,\cdots,n\}$,对于每一个状态向量 $[t,f]\in VS_{k-1}$,有:1)如果工件 J_k 被拒绝,置 $[t,f+w_k]$ 于 VS_k 中;2)如果工件 J_k 被接受,置 $[t,\max\{\max\{r_k,T_2\}+p_k+w,\max\{f-w,r_k\}+p_k+w\}]$ 于 VS_k 中,置 $[\max\{r_k,t\}+p_k,\max\{\max\{r_k,t\}+p_k+w,f\}]$ 于 VS_k 中,如果 $\max\{r_k,t\}+p_k\leqslant T_1$;

第3步,$\varphi^*(P)=\min_{[t,f]\in VS_n}\{f\}$。令 UB 是问题P的上界。假设对任意的 $[t,f]\in VS_k$,有 $f\leqslant UB$ 成立,那么动态规划算法A的运行时间一定以 nT_1UB 为界。实际上,当 t,f 为整数时,最多用 T_1UB 个向量去构造 VS_k,而A的时间复杂性与 $\sum_{k=1}^{n}|VS_k|$ 成正比。

3 全多项式时间近似方案

如果算法H产生的目标值 $\varphi_H(P)$ 至多是问题P的最优目标值 $\varphi^*(P)$ 的 ρ 倍,即 $\varphi_H(P)\leqslant\rho\varphi^*(P)$,并且算法的运行时间是输入规模的多项式函数,则称此算法H是一个 ρ 因子算法。如果对于任意的 $\varepsilon>0$,有 $\varphi_H(P)\leqslant(1+\varepsilon)\varphi^*(P)$,并且算法的运行时间是输入规模及 $\frac{1}{\varepsilon}$ 的多项式,那么此算法称为一个全多项式时间近似方案,简记为FPTAS。

对于任意 $\varepsilon>0$,定义 $LB=\frac{\varphi_H(P)}{3}$,$q=\left\lceil\frac{3n}{\varepsilon}\right\rceil$,$\delta=\frac{\varphi_H(P)}{q}$。将区间 $[0,\varphi_H(P)]$ 等分成 q 个子区间,$I_r=[(r-1)\delta,r\delta]_{1\leqslant r\leqslant q}$,长度为 δ。以下面的算法 A_ε 产生的状态 $VS_k^{\#}$ 来替代 VS_k,算法 A_ε 描述如下。

算法 A_ε 第1步,置 $VS_1^{\#}=\{[0,w_1],[r_1+p_1,r_1+p_1],[0,\max\{r_1,T_2\}+p_1]\}$;

第2步,当 $k\in\{2,3,\cdots,n\}$,对于每一个状态向量 $[t,f]\in VS_k^{\#}$,有:1)如果工件 J_k 被拒绝,置 $[t,f+w_k]$ 于 $VS_k^{\#}$ 中;2)如果工件 J_k 被接受,置 $[t,\max\{\max\{r_k,T_2\}+p_k+w,\max\{f-w,r_k\}+p_k+w\}]$ 于 $VS_k^{\#}$ 中,置 $[\max\{r_k,t\}+p_k,\max\{\max\{r_k,t\}+p_k+w,f\}]$ 于 $VS_k^{\#}$ 中,如果 $\max\{r_k,t\}+p_k\leqslant T_1$,令 $[t,f]_r\in VS_k^{\#}$,选择最小的 t 使 $f\in I_r$(当一个 t 对应了两个 f 值时,选择最小的 f),$VS_k^{\#}=\{[t,f]_r|1\leqslant r\leqslant q\}$;

第 3 步，$\varphi_{A_\varepsilon}(P)=\min\limits_{[t,f]\in VS_n^\#}\{f\}$。

引理 2　对于任意的 $[t,f]\in VS_k$，存在由算法 A_ε 产生的 $[t^\#,f^\#]\in VS_k^\#$ 使得 $t^\#\leqslant t,f^\#\leqslant f+k\delta$。

证明　数学归纳法。当 $k=1$ 时，$VS_1^\#=VS_1$，结论成立。假设当 $k-1$ 时引理成立，下证当 k 时也成立。分 3 种情形：

情形 1，$[t,f]=[t',f'+w_k]$。

$[t',f']\in VS_{k-1}$，由归纳假设，存在 $[t'^\#,f'^\#]\in VS_{k-1}^\#$，使得 $t'^\#\leqslant t',f'^\#\leqslant f'+(k-1)\delta$，因此，算法 A_ε 第 k 次迭代产生 $VS_k^\#$ 中的状态向量 $[t'^\#,f'^\#+w_k]$，通过削减状态空间得新的 $VS_k^\#$，令 $[\lambda,\mu]\in VS_k^\#$，从而，$\lambda\leqslant t'^\#\leqslant t'=t,\mu\leqslant f'^\#+w_k+\delta\leqslant f'+(k-1)\delta+w_k+\delta=f+k\delta$。

情形 2，$[t,f]=[\max\{r_k,t'\}+p_k,\max\{\max\{r_k,t'\}+p_k+w,f'\}]$。

$[t',f']\in VS_{k-1}$，由归纳假设，存在 $[t'^\#,f'^\#]\in VS_{k-1}^\#$，使得 $t'^\#\leqslant t',f'^\#\leqslant f'+(k-1)\delta$，因此，算法 A_ε 第 k 次迭代产生 $VS_k^\#$ 中的状态向量 $[\max\{r_k,t'^\#\}+p_k,\max\{\max\{r_k,t'^\#\}+p_k+w,f'^\#\}]$，通过削减状态空间得新的 $VS_k^\#$，令 $[\lambda,\mu]\in VS_k^\#$，从而

$$\lambda\leqslant\max\{r_k,t'^\#\}+p_k\leqslant\max\{r_k,t'\}+p_k=t,\mu\leqslant\max\{\max\{r_k,t'^\#\}+p_k+w,f'^\#\}+\delta\leqslant$$
$$\max\{\max\{r_k,t'\}+p_k+w,f'^\#\}+\delta\leqslant\max\{\max\{r_k,t'\}+p_k+w,f'+(k-1)\delta\}+\delta\leqslant$$
$$\max\{\max\{r_k,t'\}+p_k+w,f'\}+(k-1)\delta+\delta=\max\{\max\{r_k,t'\}+p_k+w,f'\}+k\delta=f+k\delta$$

情形 3，$[t,f]=[t',\max\{\max\{r_k,T_2\}+p_k+w,\max\{f'-w,r_k\}+p_k+w\}]$。

$[t',f']\in VS_{k-1}$，由归纳假设，存在 $[t'^\#,f'^\#]\in VS_{k-1}^\#$，使得 $t'^\#\leqslant t',f'^\#\leqslant f'+(k-1)\delta$，因此，算法 A_ε 第 k 次迭代产生 $VS_k^\#$ 中的状态向量

$$[t'^\#,\max\{\max\{r_k,T_2\}+p_k+w,\max\{f'^\#-w,r_k\}+p_k+w\}]$$

通过削减状态空间得新的 $VS_k^\#$，令 $[\lambda,\mu]\in VS_k^\#$，从而有

$$\lambda\leqslant t'^\#\leqslant t'=t,\mu\leqslant\max\{\max\{r_k,T_2\}+p_k+w,\max\{f'^\#-w,r_k\}+p_k+w\}+\delta\leqslant$$
$$\max\{\max\{r_k,T_2\}+p_k+w,\max\{f'+(k-1)\delta-w,r_k\}+p_k+w\}+\delta\leqslant$$
$$\max\{\max\{r_k,T_2\}+p_k+w,\max\{f'-w,r_k\}+p_k+w\}+(k-1)\delta+\delta=$$
$$\max\{\max\{r_k,T_2\}+p_k+w,\max\{f'-w,r_k\}+p_k+w\}+k\delta=f+k\delta$$

证毕

引理 3　对于任意的 $\varepsilon>0$，由算法 A_ε 产生的 $\varphi^*_{A\varepsilon}(P)$ 满足 $\varphi_{A_\varepsilon}(P)\leqslant(1+\varepsilon)\varphi^*(P)$。

证明　由定义有，$[t^*,f^*]\in VS_n$，由引理 2 知存在 $[t^\#,f^\#]\in VS_n^\#$，使得

$$t^\#\leqslant t^*,f^\#\leqslant f^*+n\delta=f^*+n\cdot\frac{\varphi_H(P)}{q}=f^*+n\cdot\frac{\varphi_H(P)}{\left\lceil\frac{3n}{\varepsilon}\right\rceil}\leqslant$$

$$f^*+n\cdot\frac{\varphi_H(P)}{\frac{3n}{\varepsilon}}=f^*+\frac{\varphi_H(P)}{3}\cdot\varepsilon=\varphi^*(P)+\varepsilon LB$$

因为 $\varphi_H(P)\leqslant3\varphi^*(P)$，所以 $\varphi^*(P)\geqslant\dfrac{\varphi_H(P)}{3}=LB$，则 $f^\#\leqslant\varphi^*(P)+\varepsilon LB<\varphi^*(P)+\varepsilon\varphi^*(P)=(1+\varepsilon)\varphi^*(P)$。

算法 A_ε 的第 3 步 $\varphi_{A_\varepsilon}(P)=\min\limits_{[t,f]\in VS_n^\#}\{f\}$，即 $\varphi_{A_\varepsilon}(P)\leqslant f^\#$，所以 $\varphi_{A_\varepsilon}(P)\leqslant(1+\varepsilon)\varphi^*(P)$。　证毕

引理 4　对于任意的 $\varepsilon>0$，算法 A_ε 的时间复杂性为 $O\left(\dfrac{n^2}{\varepsilon}\right)$。

证明　第 1 步算法 H 的时间复杂性为 $O(n^2)$，第 2 步 A_ε 产生的状态空间 $VS_k^\#(k=1,2,\cdots,n)$，$|VS_k^\#|\leqslant q$，有 $\sum\limits_{k=1}^{n}|VS_k^\#|\leqslant nq=n\left\lceil\dfrac{3n}{\varepsilon}\right\rceil<n\left(\dfrac{3n}{\varepsilon}+1\right)$。即算法 A_ε 的时间复杂性为 $O\left(\dfrac{n^2}{\varepsilon}\right)$。　证毕

定理 1　对于问题 1，$h|r_j,reject|C_{\max}+\sum\limits_{j\in R}w_j$，算法 A_ε 是一个全多项式时间近似方案。

4 结论

本文考虑的是带有到达时间、拒绝工件、不可用区间的单机排序问题，目标函数是极小化时间表长与所有拒

绝工件的惩罚的和,即 $1,h|r_j,reject|C_{\max}+\sum_{j\in R}w_j$,给出了 1 个 3-因子算法,1 个动态规划算法及 1 个全多项式时间近似方案,并且此全多项式时间近似方案的时间复杂性为 $O\left(\frac{n^2}{\varepsilon}\right)$。

参考文献:

[1] Bartal Y, Leonardi S, Spaccamela A M, et al. Multi-processor scheduling with rejection[J]. SIAM Journal on Discrete Mathematics, 2000, 13: 64-78.

[2] Zhang L Q, Lu L F, Yuan J J. Single machine scheduling with release dates and rejection[J]. European Journal of Operation Research, 2009, 198: 975-978.

[3] Kacem I. Approximation algorithms for the makespan minimization with positive tails on a single machine with a fixed non-availability interval[J]. J Comb Optim, 2009, 17: 117-133.

[4] 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: 1708-1712.

[5] Kacem I. Fully polynomial time approximation scheme for the total weighted tardiness minimization with a common due date [J]. Discrete Applied Mathematics, 2010, 158: 1035-1040.

[6] Woeginger G J. When does a dynamic programming formulation guarantee the existence of a fully polynomial time approximation scheme (FPTAS)? [J]. Informs Journal on Computing, 2000, 12: 57-75.

[7] Kellerer H, Strusevich V A. A fully polynomial approximation scheme for the single machine weighted total tardiness problem with a common due date[J]. Theoretical Computer Science, 2006, 369: 230-238.

[8] Mosheiov G, Oron D. Due-date assignment and maintenance activity scheduling problem[J]. Mathematical and Computer Modeling, 2006, 44: 1053-1057.

[9] Kellerer H, Kubzin M A, Strusevich V A. Two simple constant ratio approximation algorithms for minimizing the total weighted completion time on a single machine with a fixed non-availability interval[J]. European Journal of Operational Research, 2009, 199: 111-116.

Operations Research and Cybernetics

Single Machine Scheduling Problem with Release Dates, Rejection and an Unavailable Interval

LIU Che, LUO Cheng-xin

(College of Mathematics and System Science, Shenyang Normal University, Shenyang 110034, China)

Abstract: In this paper, we consider the single machine scheduling problem with release dates, rejection and an unavailable interval. A job is either rejected, in which a rejection penalty has to be paid, or accepted and processed on the machine. The machine is unavailable between T_1 and T_2 and it can process most one job at a time. The objective is to minimize the sum of the makespan of the accepted jobs and total rejection penalty of the rejected jobs. Firstly we propose a 3-approximation algorithm by lemma 1. Secondly we give a dynamic programming algorithm and reduce the running time by modifying the execution of the dynamic programming algorithm. Finally we obtain a fully polynomial-time approximation scheme for this problem, its running time is $O\left(\frac{n^2}{\varepsilon}\right)$.

Key words: release dates; rejection; unavailable interval; makespan

(责任编辑 黄 颖)