Single Machine Scheduling with Linear Processing Times[1)]

ZHAO Chuan-Li[1,2]　ZHANG Qing-Ling[1]　TANG Heng-Yong[2]
[1](College of Science, Northeastern University, Shenyang 110004)
[2](Department of Mathematics, Shenyang Normal University, Shenyang 110034)
(E-mail: zhaochuanli@etang.com)

Abstract This paper considers the single machine scheduling problem. It is assumed that jobs have the same basic processing time, but the actual processing time of each job grows linearly with its starting time. Based on the analysis of the problem, optimal algorithms are presented for the problems to minimize the sum of earliness penalties subject to no tardy jobs, to minimize the total resource consumption with makespan constraints, and to minimize makespan with the total resource consumption constraints.

Key words Scheduling, single machine, earliness penalties, resource constrained, optimal algorithm

1 Introduction

Machine scheduling problems with job processing time given as a starting time dependent function have received increasing attention in recent years[1]. In this model, the processing time of a job can be described by a basic processing time and an increasing rate. Gupta and Gupta were the first to consider the problem of this type[2]. Mosheiov studied some special cases of this model[3~5]. Bachman and Janiak proved that the problem to minimize the maximum lateness with an arbitrary linear processing time is NP-complete[6]. Cheng and Ding considered single machine problems with deadlines[7]. In this paper we focus on single machine scheduling that jobs have the same basic processing time but different increasing rates. Optimal algorithms are presented for the problems to minimize the sum of earliness penalties subject to no tardy jobs, to minimize the total resource consumption with makespan constraints and to minimize makespan with the total resource consumption constraints, respectively.

The model can be described as follows.

Suppose that n jobs $J_1, J_2, \cdots, J_n$ are to be processed on a single machine. Associated with job J_j are a basic processing time p and an increasing rate α_j (without loss of generality we assume $p=1$ and $\alpha_1 \geqslant \alpha_2 \geqslant \cdots \geqslant \alpha_n$). If t is the starting time of job J_j then the actual processing time of J_j is $1+\alpha_j t, j=1,2,\cdots,n$.

2 Minimization of the sum of earliness penalties

In this section we consider the problem to minimize the sum of earliness subject to no tardy jobs. For the classical scheduling problem, there are some results in [8,9]. In this section, it is assumed that all jobs have a common due date d and the release time is 0.

For a schedule $\pi=[J_{[1]}, J_{[2]}, \cdots, J_{[n]}]$, let $C_{[j]}$ and $E_{[j]}=d-C_{[j]}$ be the completion time and earliness of job $J_{[j]}$ in π, respectively. The sum of earliness penalties is $E(\pi)=\sum_{j=1}^{n} g(E_{[j]})$, where g is a strictly increasing function. We denote all schedules by Π. The problem is to find an optimal schedule $\pi^* \in \Pi$ that minimizes the sum of earliness penalties under the condition $C_{[j]} \leqslant d$. The problem can be denoted as

1) Supported by Doctor Thesis Foundation of Northeastern University (200208)
Received April 23, 2002; in revised form October 8, 2002
收稿日期 2002-04-23；收修改稿日期 2002-10-08

$$1 \mid 1+\alpha_j t \mid \sum g(E_j) \tag{1}$$

Since the problem is to minimize earliness, the jobs should be processed as late as possible. It is obviously that in the optimal schedule the completion time of the last job is d and there is no idle time between jobs.

First, we give some results of the problem to minimize the makespan. The problem can be denoted as $1|1+\alpha_j t|C_{\max}$.

Lemma 1[1]. For the problem $1|1+\alpha_j t|C_{\max}$, the makespan is minimized by arranging the jobs in a non-increasing order of α_j.

Lemma 2. For the problem $1|1+\alpha_j t|C_{\max}$, if $\pi=[J_1, J_2, \cdots, J_n]$ and the starting time of the first job is t_0, then the makespan is

$$C_{\max}(t_0 \mid J_1, J_2, \cdots, J_n) = 1 + \sum_{j=1}^{n-1} \prod_{k=j+1}^{n} (1+\alpha_k) + t_0 \prod_{k=1}^{n} (1+\alpha_k)$$

Lemma 3. For the problem $1|1+\alpha_j t|C_{\max}$, if $\pi=[J_1, J_2, \cdots, J_n]$ and the makespan is C, then the starting time of the first job is $t_0 = \left(C-1-\sum_{j=1}^{n-1}\prod_{k=j+1}^{n}(1+\alpha_k)\right)\Big/\prod_{k=1}^{n}(1+\alpha_k)$.

Based on the above lemmas, we have the following theorem.

Theorem 1. For the problem $1|1+\alpha_j t|\sum g(E_j)$, an optimal schedule can be obtained by arranging the jobs in a non-increasing order of α_j, where the starting time of the first job is $t_0 = \left(C-1-\sum_{j=1}^{n-1}\prod_{k=j+1}^{n}(1+\alpha_k)\right)\Big/\prod_{k=1}^{n}(1+\alpha_k)$.

Proof. Consider an optimal schedule π. Suppose that under π, there are two adjacent jobs, job J_i followed by job J_j, such that $\alpha_i<\alpha_j$. Denote the completion time of J_j as $C_j = C_0$. Then $C_i=(C_0-1)/(1+\alpha_j)$, $E_j=d-C_0$, $E_i=d-(C_0-1)/(1+\alpha_j)$. Perform a pairwise interchange on jobs J_i and J_j, and call the new schedule $\bar{\pi}$. Under $\bar{\pi}$, $\bar{C}_i=C_0$, $\bar{C}_j=(C_0-1)/(1+\alpha_i)$. $\bar{E}_i=d-C_0$, $\bar{E}_j=d-(C_0-1)/(1+\alpha_i)$.

Since $\alpha_i<\alpha_j$, $\bar{C}_j>C_i$, and $\bar{E}_j<E_i$. Hence, $g(\bar{E}_i)+g(\bar{E}_j)<g(E_i)+g(E_j)$.

The completion times of the jobs processed after jobs J_i and J_j are not affected by the interchange, the completion times of the jobs processed before jobs J_i and J_j become larger. Hence the sum of earliness penalties under $\bar{\pi}$ is strictly less than that under π. This contradicts the optimality of π. □

3 Resource constrained problems

In this section, the release time of job J_j is $r_j=f(u_j)$, $\underline{u}\leqslant u_j\leqslant\bar{u}$, where u_j is the amount of resource allocated to J_j. $\bar{u}$, $\underline{u}$ are known constraints and f is a strictly decreasing function($f(\underline{u})>f(\bar{u})>0$). $0\leqslant\underline{u}<\bar{u}$, $\sum_{j=1}^{n}u_j\leqslant U$, U is the global amount of continuously divisible non-renewable resource.

For the problem with job release time being a function of resource allocated, Cheng and Janiak have considered the case where jobs have constant processing times[10,11]. In this section we consider the problem i. e. , jobs have the same basic processing time and the different increasing rates. The first is to minimize the total resource consumption with the makespan constraint while the second is to minimize the makespan with the total resource consumption constraint.

$$1 \mid 1+\alpha_j t,\ r_j = f(u_j),\ C_{\max} \leqslant C \mid \sum u_j \tag{2}$$

$$1 \mid 1+\alpha_j t,\ r_j = f(u_j),\ \sum u_j \leqslant U \mid C_{\max} \tag{3}$$

3.1 Minimization of resource consumption

We denote the set of all resource allocations $u=[u_1, u_2, \cdots, u_n]$ by $\tilde{U}(\underline{u}\leqslant u_j\leqslant\bar{u},\ \sum_{j=1}^{n}u_j\leqslant U)$. Given a schedule $\pi=[J_{[1]}, J_{[2]}, \cdots, J_{[n]}]$ and a resource allocation $u_\pi=[u_{[1]}, u_{[2]}, \cdots, u_{[n]}]$, we have

$$r_{[j]} = f(u_{[j]}),\quad j=1,2,\cdots,n$$

$$C_{[1]}(\pi,u) = r_{[1]}+1+\alpha_{[1]}r_{[1]} = 1+(1+\alpha_{[1]})r_{[1]}$$

$$C_{[j]}(\pi,u) = \max\{r_{[j]}, C_{[j-1]}(\pi,u)\}+1+\alpha_{[j]}\max\{r_{[j]}, C_{[j-1]}(\pi,u)\}$$

$$= \max_{1\leqslant i\leqslant j}\Big\{1+\sum_{l=i}^{j-1}\prod_{k=l+1}^{j}(1+\alpha_{[k]})+r_{[i]}\prod_{k=i}^{j}(1+\alpha_{[k]})\Big\},\quad j=2,\cdots,n.$$

Hence, $C_{\max}(\pi,u)=\max_{1\leqslant j\leqslant n}\{C_{[j]}(\pi,u)\}$, $U(\pi,u)=\sum_{j=1}^{n}u_{[j]}$.

Let C be a given makespan, the problem (2) is to find $\pi^*\in\Pi$ and $u^*\in\tilde{U}$ which minimize the total resource consumption, i. e., $U(\pi^*, u^*)=\min_{\pi\in\Pi}\min_{u\in\tilde{U}}\{U(\pi,u)\}$ subject to $C_{\max}(\pi^*, u^*)\leqslant C$.

From Lemma 1, the minimum makespan of the schedule is

$$C^* = C_{\max}(f(\bar{u})\mid J_1, J_2, \cdots, J_n) = 1+\sum_{j=1}^{n-1}\prod_{k=j+1}^{n}(1+\alpha_k)+f(\bar{u})\prod_{k=1}^{n}(1+\alpha_k) \qquad (4)$$

So, a schedule $\pi=[J_{[1]}, J_{[2]}, \cdots, J_{[n]}]$ is feasible only if

$$C_{\max}(f(\bar{u})\mid J_{[1]}, J_{[2]}, \cdots, J_{[n]}) = 1+\sum_{j=1}^{n-1}\prod_{k=j+1}^{n}(1+\alpha_{[k]})+f(\bar{u})\prod_{k=1}^{n}(1+\alpha_{[k]})\leqslant C \quad (5)$$

We denote a resource allocation by u_π^* with which the resource consumption is minimized, subject to a given C, i. e.,

$$U(\pi, u_\pi^*) = \min_{u\in\tilde{U}}\{U(\pi,u)\}$$

Since releasing jobs sooner consumes more resources, jobs should be released as late as possible and the completion time of the last job is C. If $C_{\max}(\pi,\cdot)=C$, then the starting time of the first job is $t_0=\Big(C-1-\sum_{j=1}^{n-1}\prod_{k=j+1}^{n}(1+\alpha_{[k]})\Big)\Big/\prod_{k=1}^{n}(1+\alpha_{[k]})$. Hence, the resource allocation u_π^* is determined as follows.

Algorithm 1.

1) Let $t_0 = \Big(C-1-\sum_{j=1}^{n-1}\prod_{k=j+1}^{n}(1+\alpha_{[k]})\Big)\Big/\prod_{k=1}^{n}(1+\alpha_{[k]})$

2) If $1+t_0(1+\alpha_{[1]})\geqslant f(\underline{u})$, then $u_{[1]}^*=f^{-1}(t_0)$, $u_{[j]}^*=\underline{u}$, $j=2,3,\cdots,n$; stop. Otherwise, go to 3)

3) Let k be the maximum natural number satisfying

$1+\sum_{l=1}^{k-2}\prod_{i=l+1}^{k-1}(1+\alpha_{[i]})+t_0\prod_{i=1}^{k-1}(1+\alpha_{[i]})\leqslant f(\underline{u})$; then

$u_{[1]}^*=f^{-1}(t_0)$,

$u_{[j]}^*=f^{-1}\Big(1+\sum_{l=1}^{j-2}\prod_{k=l+1}^{j-1}(1+\alpha_{[k]})+t_0\prod_{k=1}^{j-1}(1+\alpha_{[k]})\Big)$, $j=2,3,\cdots,k$,

$u_{[j]}^*=\underline{u}$, $j=k+1,\cdots,n$.

Theorem 2. For the problem $1\,|\,1+\alpha_j t, r_j=f(u_j), C_{\max}\leqslant C\,|\sum u_j$, $\pi^*\in\Pi$ and $u^*\in\tilde{U}$ are obtained by arranging jobs in a non-increasing order of the increasing rates and allocating the resource according to Algorithm 1.

Proof. It is obviously that for a given π, allocating the resource according to Algorithm 1 minimizes the resource consumption.

Suppose under π, there are two adjacent jobs, job J_i followed by job J_j, such that

$\alpha_i < \alpha_j$. Denote the completion time of J_j is $C_j = C_0$; then $r_j = (C_0 - 1)/(1+\alpha_j)$, $r_i = (C_0 - 2 - \alpha_j)/(1+\alpha_i)(1+\alpha_j)$. Perform a pair-wise interchange on jobs J_i and J_j, and call the new schedule $\bar{\pi}$. Under $\bar{\pi}$, $\bar{r}_i = (C_0 - 1)/(1+\alpha_i)$, $\bar{r}_j = (C_0 - 2 - \alpha_i)/(1+\alpha_i)(1+\alpha_j)$.

Since $\alpha_i < \alpha_j$, $\bar{r}_i > r_j$, and $\bar{r}_j > r_i$. Hence, $f^{-1}(\bar{r}_i) + f^{-1}(\bar{r}_j) < f^{-1}(r_i) + f^{-1}(r_j)$.

The release times of the jobs processed after jobs J_i and J_j are not affected by the interchange, the release times of the jobs processed before jobs J_i and J_j become larger. Hence the resource consumption under $\bar{\pi}$ is strictly less than that under π. This contradicts the optimality of π. □

If we are able to calculate f and f^{-1} in $O(g(n))$ time, then to find $\pi^* \in \Pi$ and $u^* \in \widetilde{U}$ needs time $O(\max\{g(n), n\log n\})$.

3.2 Minimization of makespan

From Lemma 1 and results of Section 3.1, for the problem (3), we need only consider the schedule that the jobs are arranged in a non-increasing order of the increasing rates, i.e., $\pi = [J_1, J_2, \cdots, J_n]$.

If $u_j = \underline{u}$, $j = 1, 2, \cdots, n$, then $C_{\max}(\pi, \underline{u}) = 1 + \sum_{j=1}^{n-1} \prod_{k=j+1}^{n} (1+\alpha_k) + f(\underline{u}) \prod_{k=1}^{n} (1+\alpha_k)$.

At this condition, the release times of all jobs equal to $f(\underline{u})$, the makespan is constrained by $r_1 = f(\underline{u})$. If we increase the resource allocated to J_1 then r_1 will be smaller and the makespan will be smaller too. Let the maximal amount of resource allocated to job J_1 be $u_1^{\max}$; then $u_1^{\max} = \min\{U - (n-1)\underline{u}, \bar{u}\}$.

Let $\bar{r}_1 = f(u_1^{\max})$. The completion time of J_1 is $1 + \bar{r}_1(1+\alpha_1)$.

If $1 + \bar{r}_1(1+\alpha_1) \geq f(\underline{u})$, the optimal resource allocation is

$$u_1^* = u_1^{\max}$$
$$u_j^* = \underline{u},\ j = 2, \cdots, n$$
$$C_{\max} = 1 + \sum_{j=1}^{n-1} \prod_{k=j+1}^{n} (1+\alpha_k) + f(u_1^{\max}) \prod_{k=1}^{n} (1+\alpha_k)$$

If $1 + \bar{r}_1(1+\alpha_1) < f(\underline{u})$, there must be a natural number k, such that

$$C_j(\pi, u_\pi^*) \leq f(\underline{u}),\ j = 1, 2, \cdots, k-1$$
$$C_j(\pi, u_\pi^*) > f(\underline{u}),\ j = k, k+1, \cdots, n$$
$$u_j^* = \underline{u},\ j = k+1, k+2, \cdots, n$$

Let $d = f(\underline{u}) - C_{k-1}(\pi, u_\pi^*)$. From Lemma 3

$$r_1^* = \Big(f(\underline{u}) - d - 1 - \sum_{j=1}^{k-2} \prod_{i=j+1}^{k-1} (1+\alpha_i)\Big) \Big/ \prod_{i=1}^{k-1} (1+\alpha_i)$$
$$r_2^* = 1 + r_1^*(1+\alpha_1)$$
$$\cdots\cdots$$
$$r_{k-1}^* = 1 + \sum_{j=1}^{k-3} \prod_{i=j+1}^{k-2} (1+\alpha_i) + r_1^* \prod_{i=1}^{k-2} (1+\alpha_i)$$
$$r_k^* = 1 + \sum_{j=1}^{k-2} \prod_{i=j+1}^{k-1} (1+\alpha_i) + r_1^* \prod_{i=1}^{k-1} (1+\alpha_i) = f(\underline{u}) - d \tag{6}$$
$$u_j^* = f^{-1}(r_j^*),\ j = 1, 2, \cdots, k$$
$$u_j^* = \underline{u},\ j = k+1, \cdots, n \tag{7}$$
$$\sum_{j=1}^{k} u_j^* + (n-k)\underline{u} = U \tag{8}$$

k and d can be determined by (8).

First, we consider k.

When $r_k = f(\underline{u})$, suppose the release time of J_j is r_j, $j = 1, 2, \cdots, k$. Then

$$
\begin{aligned}
r_1 &= \left(f(\underline{u}) - 1 - \sum_{j=1}^{k-2}\prod_{i=j+1}^{k-1}(1+\alpha_i)\right)\Big/\prod_{i=1}^{k-1}(1+\alpha_i) \\
r_2 &= 1 + r_1(1+\alpha_1) \\
&\cdots\cdots \\
r_{k-1} &= 1 + \sum_{j=1}^{k-3}\prod_{i=j+1}^{k-2}(1+\alpha_i) + r_1\prod_{i=1}^{k-2}(1+\alpha_i) \\
r_k &= 1 + \sum_{j=1}^{k-2}\prod_{i=j+1}^{k-1}(1+\alpha_i) + r_1\prod_{i=1}^{k-1}(1+\alpha_i) = f(\underline{u}) \qquad (9) \\
u_j &= f^{-1}(r_j),\ j = 1,2,\cdots,k-1, \\
u_j &= \underline{u},\ j = k, k+1, \cdots, n \\
&\sum_{j=1}^{k-1} u_j + (n-(k-1))\,\underline{u} \leqslant U \qquad (10)
\end{aligned}
$$

From (10), we can get k and then d can be calculated by (8).

Based on the above analysis, we have following algorithm.

Algorithm 2.

1) Arrange jobs in a non-increasing order of the increasing rates, and let

$$u_1^{\max} = \min\{U-(n-1)\,\underline{u}, \bar{u}\},\ \bar{r}_1 = f(u_1^{\max})$$

2) If $1+\bar{r}_1(1+\alpha_1) \geqslant f(\underline{u})$, then $u_1^* = u_1^{\max}, u_j^* = \underline{u}, j = 2,\cdots,n$, and stop. Otherwise, go to 3)

3) Let k be the maximum natural number such that $\sum_{j=1}^{k-1} u_j + (n-(k-1))\underline{u} \leqslant U$, $u_j = f^{-1}(r_j), r_j$ satisfying (9)

4) The resource allocation is

$$
\begin{aligned}
u_j^* &= f^{-1}(r_j^*),\ j = 1,2,\cdots,k \\
u_j^* &= \underline{u},\ j = k+1,\cdots,n
\end{aligned}
$$

where r_j^* can be determined by (6)and d satisfies $\sum_{j=1}^{k} u_j^* + (n-k)\underline{u} = U$.

Theorem 3. For the problem $1|1+\alpha_j t, r_j = f(u_j), \sum u_j \leqslant U|C_{\max}$, an optimal schedule can be obtained by Algorithm 2.

If we are able to calculate f, f^{-1} and d in $O(g(n))$ time, then the complexity of Algorithm 2 is $O(\max\{g(n), n\log n\})$.

4 Conclusion

We studied single machine scheduling problems under the assumption that job processing time is a function of their starting time. There are many practical scheduling problems that can be modeled in this way. In the most general case, processing time is a strictly increasing function of the starting time and the job is characterized by a basic processing time and an increasing rate. We considered a special case where jobs have the same basic processing time and different increasing rates. For the problems to minimize the sum of earliness penalties subject to no tardy jobs, to minimize the total resource consumption with makespan constraints and to minimize makespan with the total resource consumption constraints, the optimal algorithms are presented, respectively.

References

1 Alidaee B, Womer NK. Scheduling with time dependent processing times: Review and extensions. *Journal of Operational Research Society*, 1999, **50**(5): 711~720

2 Gupta J N D, Gupta S K. Single facility scheduling with nonlinear processing times. *Computers and Industrial Engineering*, 1988, **14**(4):387~393

3 Mosheiov G. V-shaped policies for scheduling deteriorating jobs. *Operations Research*, 1991,**39** (6):979～991
4 Mosheiov G. A-shaped policies for scheduling deteriorating jobs. *Journal of Operational Research Society*, 1996,**47** (6):1184～1191
5 Mosheiov G. Scheduling jobs under simple linear deterioration. *Computers and Operations Research*, 1994,**21**(6): 653～659
6 Bachman A, Janiak A. Minimizing maximum lateness under linear deterioration. *European Journal of Operational Research*, 2000,**126**(1):557～566
7 Cheng T C E, Ding Q. Single machine scheduling with deadlines and increasing rates of processing times. *Acta Informatica*, 2000,**36**(5):673～692
8 Chang S, Schneeberger H. Single machine scheduling to minimize weighted earliness subject to no tardy jobs. *European Journal of Operational Research*, 1988,**34**(2):221～230
9 Qi Xiang-Tong, Tu Feng-Sheng. Scheduling a single machine to minimize earliness penalties subject to the SLK due-date determination method. *European Journal of Operational Research*, 1998,**105**(3):502～508
10 Cheng T C E, Janiak A. Resource optimal control in some single-machine scheduling problem. *IEEE Transactions on Automatic Control*, 1994,**39**(6):1243～1246
11 Janiak A. Time-optimal control in a single machine problem resource constraints. *Automatica*, 1986,**22**(6): 745～747

ZHAO Chuan-Li Professor in Department of Mathematics, Shenyang Normal University and a Ph. D. candidate in College of Science, Northeastern University. His research interests include scheduling and optimal control.

ZHANG Qing-Ling Professor in College of Science, Northeastern University. His research interests include singular systems and decentralized control.

TANG Heng-Yong Professor in Department of Mathematics, Shenyang Normal University. His research interests include scheduling and stochastic programming.

一类线性加工时间单机调度问题

赵传立[1,2] 张庆灵[1] 唐恒永[2]

[1](东北大学理学院 沈阳 110004)

[2](沈阳师范大学数学系 沈阳 110034)

(E-mail: zhaochuanli@etang. com)

摘 要 讨论一类线性加工时间单机调度问题.在这类问题中,工件具有相同的基本加工时间,但每个工件的实际加工时间以其开工时间线性增长.对满足无延迟工件条件下极小化提前惩罚和问题,满足最大完工时间限制条件下极小化资源消耗总量的问题和满足资源消耗总量限制条件下极小化最大完工时间的问题,分别给出了最优算法.

关键词 调度,单机,提前惩罚,资源约束,最优算法

中图分类号 TP13