

# 带时间延迟的极小化总完工时间的单机排序问题

胡觉亮, 王焕男, 蒋义伟  
(浙江理工大学理学院, 杭州 310018)

**摘要:** 研究工件带有两道工序的单机排序问题。在该问题中, 工件的第一道工序先于第二道工序加工, 并且第二道工序的开工时间与第一道工序的完工时间至少间隔一定的延迟时间, 目标是极小化所有工件的总完工时间。文章考虑所有工件相同且两道工序的加工时间均为单位时间的情形。通过引入  $k$ -连续加工的概念和分析最优解的性质, 根据延迟时间的大小, 分别设计了两个算法并证明了算法所得的排序为最优排序。

**关键词:** 单台机; 时间延迟; 总完工时间; 算法设计与分析; 最优排序

**中图分类号:** O233      **文献标志码:** A

## 0 引言

本文主要研究带延迟时间的单机排序问题。每个工件  $J_j$  有两道工序  $a_j$  和  $b_j$ , 第一道工序先于第二道工序加工, 第一道工序的完工时间  $c_{a_j}$  与第二道工序的开工时间  $s_{b_j}$  之间至少存在  $l_j$  个单位时间延迟, 也就是说第二道工序至少等待  $l_j$  个单位时间才能开工。此类问题在一些产品制造工艺流程、布匹印染和服装订单的生产中有重要的应用背景。对于带有延迟的排序问题, 主要分为两类, 一类是工件  $J_j$  前后两道工序的延迟时间恰好为  $l_j$ , 本文称之为精确延迟排序; 另一类是两道工序间的延迟时间至少为  $l_j$ , 称之为至少延迟排序。

关于精确延迟的排序问题, Orman 等<sup>[1]</sup> 证明了单台机的一些特殊情况是多项式可解的, 并证明即便所有工序的加工时间相同, 该问题还是强 NP-难的。Leung 等<sup>[2]</sup> 利用贪婪算法研究延迟非增的情形, 给出了问题  $F2 | \bar{l}_j, a_j = a, b_j = b, a \geq b | \sum C_j$  的最优排序, 并对问题  $F2 | \bar{l}_j, a_j = a, b_j = b, a < b | \sum C_j$  给出了一个 2-近似算法。Ageev 等<sup>[3]</sup> 分别研究了单台机和两台流水作业机器问题的一些近似算

法。Ageev 等<sup>[4]</sup> 给出了问题  $1 | \text{exact } l_j, a_j = b_j = 1 | C_{\max}$  的一个  $7/4$ -近似算法, 并证明界不可改进, 同时给出了问题  $F2 | \text{exact } l_j, a_j = b_j = 1 | C_{\max}$  的一个  $3/2$ -近似算法, Yu 等<sup>[5]</sup> 证明了上述两个问题是强 NP-难的。Huo 等<sup>[6]</sup> 和 Glascock 等<sup>[7]</sup> 研究了问题  $F2 | \text{exact } l_j | \sum C_j$  的计算复杂性并给出相应的启发式算法。

关于至少延迟的排序问题, Kern 等<sup>[8]</sup> 证明了以极小化最大完工时间为目标的单台机排序在解不限于排列排序的情况下是 NP-难的。而对于两台流水作业排序问题  $F2 | l_j | C_{(\max)}$ , Dell'Amico<sup>[9]</sup> 给出了该问题的一个 2-近似算法, 并提出了一个禁忌搜索算法。若两道工序的加工时间相同时, Dell'Amico 等<sup>[10]</sup> 证明了问题  $F2 | l_j, a_j = b_j | C_{\max}$  是强 NP-难的, Johnson<sup>[11]</sup> 和 Mitten<sup>[12]</sup> 证明该问题存在最优排列排序 (permutation schedule)。若两道工序加工时间相同且延迟时间  $l_j \in \{0, l\}$  时, Yu<sup>[13]</sup> 证明了问题  $F2 | l_j \in \{0, l\}, a_j = b_j | C_{\max}$  是强 NP-难的。进一步, 即使两道工序的加工时间均为单位时间, Yu<sup>[5]</sup> 证明了问题  $F2 | l_j, a_j = b_j = 1 | C_{\max}$  也是强 NP-难的。

对上述研究综述进行汇总后, 如表 1 所示。

表1 带有至少(精确)延迟时间的单机和两台机流水作业排序

	单台机	两台机流水作业
至少延迟排序	$1 l_j C_{\max}$	[8] $F2 l_j C_{\max}$ [9]
		$F2 l_j, p_{i,j}=1 C_{\max}$ [5]
		$F2 l_j, a_j=b_j C_{\max}$ [10]
		$F2 l_j=l C_{\max}$ [11][12]
		$F2 l_j \in \{0, l\}, a_j=b_j C_{\max}$ [13]
		$F2 \bar{l}_j, a_j=a, b_j=b, a \geq b \sum C_j$ [2]
精确延迟排序		$F2 \bar{l}_j, a_j=a, b_j=b, a < b \sum C_j$ [2]
	$1 \text{exact } l_j, a_j=b_j=l_j C_{\max}$	[11] $F2 \text{exact } l_j \sum C_j$ [6][7]
	$1 \text{exact } l_j, a_j=b_j C_{\max}$	[13][16] $F2 \text{exact } l_j C_{\max}$ [3]
	$1 \text{exact } l_j, a_j \leq b_j C_{\max}$	[13] $F2 \text{exact } l_j, a_j \leq b_j C_{\max}$ [3]
	$1 \text{exact } l_j, a_j \geq b_j C_{\max}$	[13] $F2 \text{exact } l_j, a_j \geq b_j C_{\max}$ [3]
	$1 \text{exact } l_j, a_j=b_j=1 C_{\max}$	[8][14] $F2 \text{exact } l_j, a_j=b_j=1 C_{\max}$ [4][5]

本文研究的问题属于至少延迟排序。由上述结果可以看出,此类问题主要研究单台机和两台机的流水作业问题,且均以极小化最大完工时间为目标。然而,在很多实际问题中,最大完工时间只能体现局部最优性,为了考虑全局的优劣,通常需要考虑总完工时间这一目标函数。例如,在服装订单的问题上,总的完工时间越小说明每个订单的平均等待时间越小,从而体现了全局的最优性。

因此,本文考虑目标为极小化总完工时间的至少延迟问题,即  $1|l, a_j=b_j=1|\sum C_j$ , 其中  $l_j=l$  表示所有工件的延迟时间相同并且假定所有工序的加工时间均为单位时间,即  $a_j=b_j=1$ 。 $C_j$  表示工件  $J_j$  的完工时间。

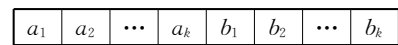
若延迟时间大于工序的加工时间,即  $l>1$ 。不妨假设  $l \downarrow =q$ , 这里  $q$  为不超过  $l$  的最大整数,显然第一道工序的后面可直接安排  $q$  个其他工件的工序而不影响该工件的完工,故只需考虑在此之后的延迟时间段  $l \downarrow -q$  是否需要考虑安排其他工件的工序即可。注意到  $l \downarrow -q < 1$ , 因此本文主要考虑延迟时间小于工序加工时间的情形,即  $0 \leq l < 1$ 。根据  $l$  的大小分别对  $l \in [0, \frac{1}{3})$  和  $l \in [\frac{1}{3}, 1)$  给出了最优排序。

## 1 预备结果

本节主要给出相关的定义以及一些初步的结论。

**定义 1.1** 将  $k$  个工件的第一道工序连续加工,紧接着按照第一道工序的加工顺序连续加工这

$k$  个工件的第二道工序,我们称之为  $k$ -连续加工(如图 1 所示)。

图1  $k$ -连续加工

**定义 1.2** 若同一个工件的两道工序之间不存在其他工件的工序,称这种工件的加工方式为单独加工。

**引理 1.1** 最优排序中,只存在单独加工和 2-连续加工这两种加工方式,即不存在  $k(k \geq 3)$ -连续加工。

**证明:** 由图 1 可知,  $l$ -连续加工的各个工件的完工时间是一个等差数列,不难得到第一个工件的完工时间  $C_1=k+1$ ,第  $k$  个工件的完工时间  $\frac{3k^2}{2} + \frac{k}{2}$ ,故所有的完工时间为

$$\sum C_j = \frac{C_1 + C_k}{2} \cdot k = \frac{k+1 + \frac{3k^2}{2} + \frac{k}{2}}{2} \cdot k = \frac{3k^2}{2} + \frac{k}{2} \quad (1)$$

若存在  $k(k \geq 3)$ -连续加工,可以将这些工件按照 2-连续加工或者单独加工得到更好的排序,从而得出结论。下面分别根据  $k$  的奇偶性对这  $k$  个工件进行重新排序。

若  $k$  为偶数,即必有  $k \geq 4$ ,则将这些工件进行 2-连续加工,即有  $\frac{k}{2}$  个 2-连续。不难得出这  $k$  个工件的完工时间如下:第一个工件与第二个工件的总完工时间为  $3+4=7$ ;第三个工件与第四个工件的总完工时间为  $7+8=15$ ;第  $k-1$  个工件与第  $k$  个工件的总完工时间为  $4k-1$ 。由等差数列的求和可知,这  $k$  个工件通过 2-连续加工的总完工时间为

$$\sum C_j = \frac{c_1 + c_k}{2} \cdot k = \frac{7+4k-1}{2} \cdot \frac{k}{2} = k^2 + \frac{3k}{2} \quad (2)$$

由  $k \geq 4$  可得

$$(1)-(2) = \frac{3k^2}{2} + \frac{k}{2} - \left(k^2 + \frac{3k}{2}\right) = \frac{k^2}{2} - k > 0$$

当  $k$  为奇数时,则将  $k-1$  个工件分批进行 2-连续加工,由(2)式知,这  $k-1$  个工件 2-连续加工的总完工时间为  $(k-1)^2 + \frac{3(k-1)}{2}$ 。而第  $k$  个工件单独加工,完工时间为  $2(k-1) + 2 + l$ 。所以,这  $k$  个工件的总完工时间为

$$\begin{aligned} \sum C_j &= (k-1)^2 + \frac{3(k-1)}{2} + 2(k-1) + 2 + l \\ &= k^2 + \frac{3k}{2} - \frac{1}{2} + l, (0 \leq l < 1) \end{aligned} \quad (3)$$

由  $k \geq 3$  和  $0 \leq l < 1$  可得

$$(1)-(3) = \frac{3k^2}{2} + \frac{k}{2} - \left(k^2 + \frac{3k}{2} - \frac{1}{2} + l\right) = \frac{k^2}{2} - k + \frac{1}{2} - l > 0$$

因此,最优排序中只存在 2-连续加工和单独加工这两种方式。

**引理 1.2** 如果最优排序中存在单独加工的工件,则该工件必在连续加工的工件之后进行加工。

**证明:**假设在  $A$  时刻有一个工件单独加工,后面存在连续加工的工件,不妨设有  $m$  ( $m$  为偶数)个工件,由引理 1.1 可知,这些工件都是 2-连续加工。由(2)式可知,这  $m+1$  个工件的总完工时间为

$$A + 2 + l + m^2 + \frac{3m}{2} + m(A + 2 + l) =$$

$$(m+1)A + 2m + 2 + l + ml + m^2 + \frac{3m}{2} \quad (4)$$

而将这个工件放在  $m$  个工件之后加工的总完工时间为

$$m^2 + \frac{3m}{2} + mA + (A + 2m + 2 + l) =$$

$$(m+l)A + 2m + 2 + l + m^2 + \frac{3m}{2} \quad (5)$$

比较这两个总完工时间可知,  $(4)-(5) = ml > 0$ 。

因此,在最优排序中,若有 2-连续加工的工件,必在零时刻开始加工所有的 2-连续工件,然后再加工那些单独加工的工件。

## 2 最优算法

本节将给出问题  $1 \mid l, a_j = b_j = 1 \mid \sum C_j$  的最优算法,对  $l \in [0, \frac{1}{3})$  和  $l \in [\frac{1}{3}, 1)$  两种情形分别

进行考虑。下面先给出  $l \in [0, \frac{1}{3})$  情形的最优排序。

**算法 H1** (1)若  $0 < k \leq \frac{l+1}{2l}$  时,所有工件单独加工;

(2)若存在正整数  $i$  使得  $\frac{l+1}{2l} + 2(i-1) < k \leq \frac{l+1}{2l} + 2i$  时,则将前  $2i$  个工件分别进行 2-连续加工,其余工件单独加工。

**定理 2.1** 算法 H1 是问题  $1 \mid l \in [0, \frac{1}{3}), a_j = b_j = 1 \mid \sum C_j$  的最优算法。

**证明:**先考虑第一种情况,即  $0 < k \leq \frac{l+1}{2l}$ 。反证法,若此时存在工件是 2-连续加工的,由引理 1.2 可知连续加工的都在前面,不妨假设最后进行 2-连续加工的两个工件  $J_i$  和  $J_j$  在  $A$  时刻开工,设在此之后单独加工的工件数为  $s$ 。现将  $J_i$  和  $J_j$  与后面  $s$  个工件一样进行单独加工,考虑前后两种排序的变化。

$J_i$  和  $J_j$  两个工件的完工时间减少了  $(3+4) - (2+l+4+2l) = 1-3l$ ,而后  $s$  个工件的总完工时间的增加了  $2ls$ 。由于  $s \leq k-2 \leq \frac{l+1}{2l} - 2$ ,不难得到  $1-3l \geq 2ls$ ,即  $J_i$  和  $J_j$  完工时间的减少量不小于后面  $s$  个工件的总完工时间的增加量。

因此,当工件的个数  $k \leq \frac{l+1}{2l}$  时,所有工件单独加工为最优排序。

下面考虑第二种情况,若存在正整数  $i$  使得

$$\frac{l+1}{2l} + 2(i-1) < k \leq \frac{l+1}{2l} + 2i$$

即

$$\frac{2kl-l-1}{4l} \leq i < \frac{2kl+3l-1}{4l} \quad (6)$$

由引理 1.2 可知,最优排序将 2-连续工件先加工,再加工单独工件。设前  $2j$  个工件两个一组进行 2-连续加工,其余单独加工。此时总完工时间为

$$\begin{aligned} \sum C_j &= (2j)^2 + \frac{3}{2} \cdot 2j + 4j(k-2j) + \\ &\quad \frac{(k-2j)(k-2j+1)(2+l)}{2} = \end{aligned}$$

$$2lj^2 + (1-l-2kl)j + \frac{kl^2 + kl + 2k^2 + 2k}{2} \quad (7)$$

类似可以计算得到算法的目标函数值为

$$2li^2 + (1-l-2kl)i + \frac{kl^2 + kl + 2k^2 + 2k}{2} \quad (8)$$

下面证明(7)-(8) =  $2l(j-i)(j+i+\frac{1-l-2kl}{2l}) \geq 0$ 。

若  $j > i$ , 即  $j \geq i+1$ , 由(6)式可知  $i \geq \frac{2kl-l-1}{4l}$ , 则

$$2l(j-i)(j+i+\frac{1-l-2kl}{2l}) \geq$$

$$2l(2i+1+\frac{1-l-2kl}{2l}) =$$

$$4l(i-\frac{2kl-l-1}{4l}) \geq 0$$

若  $j < i$ , 即有  $j \leq i-1$ , 由(6)式可知  $i < \frac{2kl+3l-1}{4l}$ , 则

$$2l(j-i)(j+i+\frac{1-l-2kl}{2l}) >$$

$$-2l(2i-1+\frac{1-l-2kl}{2l}) =$$

$$-4l(i-\frac{2kl+3l-1}{4l}) > 0$$

因此, 当  $j=i$  时, 目标函数值最小, 即算法所得是最优排序。

下面给出  $l \in [\frac{1}{3}, 1)$  时的最优排序算法。

**算法 H2** (1) 若工件个数为偶数则将所有工件进行 2-连续加工;

(2) 若工件个数为奇数, 则将前  $n-1$  个工件进行 2-连续加工, 最后一个工件单独加工。

**定理 2.2** 算法 H2 是问题  $1 \mid l \in [\frac{1}{3}, 1), a_j = b_j = 1 \mid \sum C_j$  的最优算法。

**证明:** 由引理 1.1 和 1.2 知, 最优排序中只存在 2-连续加工和单独加工, 且单独加工的工件都在连续加工之后加工。结合笔者的算法, 只需证明最优排序不可能存在两个或者两个以上单独加工的工件。

不妨假设在 A 时刻有两个相邻的工件单独加工, 现将这两个工件放在一起加工, 比较这两个工件在前后两种排序中的总完工时间。若单独加工, 不难得到这两个工件的总完工时间为

$$A+2+l+A+4+2l=2A+3l+6。$$

而将这两个工件放在一起进行 2-连续加工, 总完工时间为

$$A+3+A+4=2A+7。$$

比较这两个总完工时间知  $(2A+3l+6)-(2A$

$+7)=3l-1 \geq 0$ 。

因此, 算法 H2 得到的排序为最优排序。

### 3 结束语

本文研究了带有至少  $l(0 \leq l < 1)$  个单位时间延迟的单位工件单机排序问题, 目标是极小化总完工时间。分别针对  $l \in [0, \frac{1}{3})$ ,  $l \in [\frac{1}{3}, 1)$  两种情形给出了相应的最优算法。对于该类问题的进一步研究, 可以考虑工件的加工时间更为一般的情况, 包括两道工序加工时间不同的情形, 以及工件的延迟时间不同的情形下的近似算法设计及分析。

### 参考文献:

- [1] Orman A J, Potts C N. On the complexity of coupled-task scheduling [J]. Discrete Applied Mathematics, 1997, 72 (1): 141-154.
- [2] Leung J, Li H, Zhao H. Scheduling two-machine flow shops with exact delay [J]. International Journal of Foundations of Computer Science, 2007, 18 (2): 341-360.
- [3] Ageev A A, Kononov A V. Approximation algorithms for scheduling problems with exact delays [C] // Erlebach T, Kaklamanis C. 4th International Workshop, WAOA 2006, Lecture Notes in Computer Science. Berlin: Springer, 2007, 4368: 1-14.
- [4] Ageev A A, Kononov A V. Approximation algorithms for the single and two-machine scheduling problems with exact delays [J]. Operations Research Letters, 2007, 35 (4): 533-540.
- [5] Yu W, Hoogeveen H, Lenstra J K. Minimizing makespan in a two-machine flow shop with delays and unit-time operations is NP-hard [J]. Journal of Scheduling, 2004, 7 (5): 333-348.
- [6] Huo Y, Li H, Zhao H. Minimizing total completion time in two-machine flow shops with exact delays [J]. Computers and Operations Research, 2009, 36 (6): 2018-2030.
- [7] Glascock J, Hunter B. Minimizing total completion time in two-machine flow shops with exact delay using genetic algorithm & ant colony algorithm [C] // Rothlauf F. Proceedings of the 11th Annual Conference Companion on Genetic and Evolutionary Computation Conference, New York: ACM, 2009: 2733-2736.
- [8] Kern W, Nawijn W M. Scheduling multi-operation jobs with time lags on a single machine [C] // Faigle U, Hoedde C. Proceedings 2nd Twente Workshop on Graphs and

- Combinatorial Optimization. Enschede: University of Twente, 1991.
- [9] Dell'Amico M. Shop problems with two machines and time lags[J]. Operations Research, 1996, 44(5): 777-787.
- [10] Dell'Amico M, Vaessens R J M. Flow and open shop scheduling on two machines with transportation times and machine-independent processing times is NP-hard [M]. Modena: Universita Degli Studi di Modena, 1996: 103-121.
- [11] Johnson S M. Discussion: sequencing  $n$  jobs on two machines with arbitrary time-lags [J]. Management Science, 1959, 5(3): 299-303.
- [12] Mitten L G. Sequencing  $n$  jobs on two machines with arbitrary time-lags[J]. Management Science, 1959, 5(3): 293-298.
- [13] Yu W. The two-machine flow shop problem with delays and the one-machine total tardiness problem[D]. Germany: Eindhoven University of Technology, 1996.

## Single Machine Sequencing Problem of Minimization of Total Completion Time with Time Delay

HU Jue-liang, WANG Huan-nan, JIANG Yi-wei

(School of Sciences, Zhejiang Sci-Tech University, Hangzhou 310018, China)

**Abstract:** This paper studies the sequencing problem of single machine with its workpieces having two processes. In this problem, the first process of workpieces is implemented earlier than the second one and the commencement time of the second one and the completion time of the first one should at least have a certain time interval, called as delay time. The objective is to minimize the total completion time of all workpieces. This paper considers situations when all workpieces are the same and the processing time of two processes is unit time, respectively designs two algorithms according to the delay time by introducing the concept of  $k$ -continuous process and analyzing the property of optimal solution and proves that the sequencing obtained by the algorithm is the optimal sequencing.

**Key words:** single machine; time delay; total completion time; design and analysis of algorithm; optimal sequencing

(责任编辑: 许惠儿)