Scheduling Problems under Linear Deterioration\(^1\)

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 scheduling problem under linear deterioration. It is assumed that the deterioration function is a linear function. Optimal algorithms are presented respectively for single machine scheduling problems of minimizing the makespan, weighted sum of completion times, maximum lateness and maximum cost. For two-machine flow shop scheduling problem to minimize the makespan, it is proved that the optimal schedule can be obtained by Johnson's rule. If the processing times of operations are equal for each job, flow shop scheduling problems can be transformed into single machine scheduling problems.

Key words Scheduling, single machine, flow shop, linear deterioration

1 Introduction

In classical scheduling problems it is assumed that the processing times of jobs are constant and independent of the starting time. However, there are many situations where the processing time of the job depends on its starting time\(^2\). In this model, the processing time of a job can be described by a deterioration function. Browne and Yechiali\(^2\) studied the single machine stochastic scheduling problem. Mosheiov\(^3\) considered the single machine scheduling under simple linear deterioration. There are some solutions for the scheduling problems with arbitrary linear deterioration in [4,5]. Bachman and Jania\(^k\) proved that the single machine problem to minimize the maximum lateness with an arbitrary linear processing time is NP-complete. Yang and Chern\(^7\) considered the two-machine flow shop problem.

This paper considers the scheduling problems under linear deterioration in which the actual processing time of a job is the product of its basic processing time and the deterioration function. It is the generalization case of the model in [3].

2 Single machine problems

The single machine problem can be described as follows.

There are \(n\) jobs \(J_1,J_2,\ldots,J_n\). Associated with job \(J_j\) is a weight \(w_j\) and a due date \(d_j\). The processing time of job \(J_j\) is \(p_j(a+bt)\) if its starting time is \(t\), where \(a\) and \(b\) are positive constants. The basic processing time of job \(J_j\) is \(p_j\) and deterioration function is \(d(x)=-a+bx, j=1,2,\ldots,n\). The problem is denoted as

\[
1 \mid p_j(a+bt) \mid f(C)
\]

where \(f(C) = f(C_1,C_2,\ldots,C_n)\) is a non-decreasing function of completion time.

For a given schedule \(\pi\), if the completion time of job \(J_j\) is \(C_j\), the lateness of job \(J_i\) is \(L_i=C_i-d_i, i=1,2,\ldots,n\). The makespan of schedule \(\pi\) is \(C_{\text{max}}=\max_{1\leq j \leq n} C_j\), the weighted sum of completion times is \(\sum_{j=1}^{n} w_j C_j = \sum_{j=1}^{n} w_j C_j\), the maximum lateness is \(L_{\text{max}} = \max_{1\leq j \leq n} \{L_j\}\), the maximum cost is \(h_{\text{max}} = \max_{1\leq j \leq n} \{h_j(C_j)\}\), \(h_j(C_j)\) is a non-decreasing function of \(C_j\).

\(^1\) Supported by the Ph. D. thesis Foundation of Northeastern University(200208)
Received January 21, 2002; in revised form October 30, 2002

2002-01-21; 2002-10-30
2.1 The problem \(1 \mid p_j(a + bt) \mid C_{\text{max}}\)

**Theorem 1.** For the problem \(1 \mid p_j(a + bt) \mid C_{\text{max}}\), the makespan is sequence independent. If the starting time of the first job is \(t\), the makespan is

\[
C_{\text{max}}(t \mid J_1, J_2, \ldots, J_n) = t + (a + bt) \sum_{k=1}^{n} b^{-1} C^k_n(p_1, p_2, \ldots, p_n)
\]

where \(C^k_n(p_1, p_2, \ldots, p_n)\) is the sum of \(C^k_n\) items, each item is the product of \(k\) numbers of \(p_1, p_2, \ldots, p_n\).

**Proof** (by induction). Without loss of generality we consider schedule \(\pi = [J_1, J_2, \ldots, J_n]\),

\[
C_1 = t + p_1(a + bt)
\]

\[
C_2 = C_1 + p_2(a + bC_1) = t + (a + bt)(\sum_{j=1}^{i} p_j + bp_1, p_2) = t + (a + bt) \sum_{k=1}^{n} b^{-1} C^k_n(p_1, p_2)
\]

Suppose Theorem 1 holds for job \(J_j\), i.e.,

\[
C_j = t + (a + bt) \sum_{k=1}^{j} b^{-1} C^k_j(p_1, p_2, \ldots, p_j)
\]

Consider job \(J_{j+1}\)

\[
C_{j+1} = C_j \mid p_{j+1}(a + bC_j) = t \mid (a + bt) \sum_{k=1}^{j} b^{-1} C^k_j(p_1, p_2, \ldots, p_j) + p_{j+1}(a + bt) + (a + bt) \sum_{k=1}^{j} p_{j+1} b^{-1} C^k_j(p_1, p_2, \ldots, p_j)
\]

\[
= t + (a + bt) [C_j(p_1, p_2, \ldots, p_j) + p_{j+1}(a + bt) + \sum_{k=1}^{j} p_{j+1} b^{-1} C^k_j(p_1, p_2, \ldots, p_j)]
\]

\[
= t + (a + bt) [C_j(p_1, p_2, \ldots, p_j) + p_{j+1}(a + bt) + \sum_{k=1}^{j} b^{-1} C^k_j(p_1, p_2, \ldots, p_j)]
\]

Hence, Theorem 1 holds for \(J_{j+1}\). This completes the proof of Theorem 1.

**Corollary 1.** \(C_{\text{max}}(t \mid J_1, J_2, \ldots, J_n) = C_{\text{max}}(C_{\text{max}}(t \mid J_1, \ldots, J_k) \mid J_{k+1}, \ldots, J_n)\) \((1 \leq k \leq n)\).

**Corollary 2.** If \(p' \leq p''\), then \(C_{\text{max}}(t \mid J_1, J_2, \ldots, J_k, J') \leq C_{\text{max}}(t \mid J_1, J_2, \ldots, J_k, J'')\).

2.2 The problem \(1 \mid p_j(a + bt) \mid \sum w_j C_j\)

**Theorem 2.** For the problem \(1 \mid p_j(a + bt) \mid \sum w_j C_j\), an optimal schedule can be obtained by arranging jobs in an order of non-decreasing \(p_j/w_j(1+bp_j)\).

**Proof** (by contradiction). Suppose under an optimal schedule \(\pi\), there are two adjacent jobs \(J_j\) and \(J_k\) followed by \(J_l\), such that \(p_j/w_j(1+bp_j), p_k/w_k(1+bp_k)\).

Let the starting time of job \(J_j\) is \(t\), and the weighted sum of completion times of \(J_j\) and \(J_k\) is

\[
w_j C_j(\pi) + w_k C_k(\pi) = w_j[t + p_j(a + bt)] + w_k[t + (p_k + b p_k p_j)(a + bt)]
\]

Perform an adjacent pair wise interchange on jobs \(J_j\) and \(J_k\), and call the new schedule \(\tilde{\pi}\). Under \(\tilde{\pi}\), the weighted sum of completion times of \(J_j\) and \(J_k\) is

\[
w_j C_j(\tilde{\pi}) + w_k C_k(\tilde{\pi}) = w_j[t + p_j(a + bt)] + w_k[t + (p_k + b p_k p_j)(a + bt)]
\]

If \(p_j/w_j(1+bp_j) > p_k/w_k(1+bp_k)\), it is easily verified that

\[
w_j C_j(\pi) + w_k C_k(\pi) < w_j C_j(\tilde{\pi}) + w_k C_k(\tilde{\pi})
\]

Since the weighted sum of completion times of other jobs is not affected by the interchange, then the weighted sum of completion times under \(\tilde{\pi}\) is strictly less than that under \(\pi\). This contradicts the optimality of \(\pi\).
2.3 The problem \(1 \mid p_j(a+bt) \mid h_{\text{max}}\).

For the due date related model, we consider the general problem \(1 \mid p_j(a+bt) \mid h_{\text{max}}\). From Theorem 1, for a set of jobs, the makespan is sequence independent. Let \(A\) denote the set of jobs to be scheduled, \(B\) denote the set of jobs already scheduled, \(C_{\text{max}}(A)\) is the makespan of jobs in \(A\).

Algorithm 1.
Step 1. Let \(A=(J_1, J_2, \ldots, J_s), B=\emptyset\).
Step 2. Calculate \(t=C_{\text{max}}(A)\). Find \(j^*\) such that \(h_{j^*}(t) = \min\{h_j(t) \mid J_i \in A\}\).
Arrange job \(J_{j^*}\) in the last position.
Step 3. Set \(A=A-\{J_{j^*}\}, B=B+\{J_{j^*}\}\). If \(A=\emptyset\), stop; otherwise go to Step 2.

Theorem 3. Algorithm 1 generates an optimal schedule for the problem \(1 \mid p_j(a+bt) \mid h_{\text{max}}\).

Proof. Let \(\pi\) be an optimal schedule, \(\pi\) be the schedule of Algorithm 1. We will show that schedule \(\pi\) can be transformed into schedule \(\bar{\pi}\) and the maximum cost does not increase. Without loss of generality we assume that the last job of \(\pi\) is \(J_{j^*}\) and that of \(\pi\) is \(J_{j^*}\). Let \(\pi=[S_1, J_{j^*}, S_2, J_{j^*}]\), where \(h_{j^*}(t) \geq h_{j^*}(t)\), \(S_1\) and \(S_2\) are the sets of the other \(n-2\) jobs.

From \(\pi\), by placing job \(J_{j^*}\) in the last position we can get a new schedule \(\bar{\pi}=[S_1, S_2, J_{j^*}, J_{j^*}]\). Since only the completion time of job \(J_{j^*}\) becomes larger and \(h_{j^*}(t) \geq h_{j^*}(t)\), the maximum cost of \(\bar{\pi}\) cannot be larger than that of \(\pi\). Now \(\bar{\pi}\) and \(\bar{\pi}\) have the same job in the last position. By a similar method we can transform \(\pi\) into \(\bar{\pi}\) and the maximum cost is not increased.

The problem \(1 \mid p_j(a+bt) \mid L_{\text{max}}\) is a special case of the problem \(1 \mid p_j(a+bt) \mid h_{\text{max}}, h_j(C_j) = C_j - d_j\). At this condition, the schedule of Algorithm 1 satisfies the EDD (Earliest Due Date First) rule.

Theorem 4. For the problem \(1 \mid p_j(a+bt) \mid L_{\text{max}}\), an optimal schedule can be obtained by the EDD rule.

3 Flow shop problems

The Flowshop problems can be described as follows.

There are \(n\) jobs \(J_1, J_2, \ldots, J_n\) to be processed successively on \(m\) machines \(M_1, M_2, \ldots, M_m\) in that order. Moreover, we assume that the same job order is chosen on each machine. Associated with job \(J_i\) is a weight \(w_i\) and a due date \(d_i\). The operation of job \(J_i\) on \(M_i\) is denoted by \(T_i\). The processing time of operation \(T_i\) is \(p_{ij}(a+bt)(i=1,2,\ldots,m, j=1,2,\ldots,n)\) if its starting time is \(t\). The problem is denoted as \(Fm \mid p_j(a+bt) \mid f(C)\).

3.1 The problem \(F2 \mid p_j(a+bt) \mid C_{\text{max}}\)

Johnson rule (\(SPT(M_1) - LPT(M_2)\) rule)[8,9]

Partition the jobs into two sets with set \(A\) containing all jobs with \(p_{ij} < p_{ij}\) and set \(B\) all the jobs with \(p_{ij} > p_{ij}\). The jobs with \(p_{ij} - p_{ij}\) may be in either set. The jobs in \(A\) go first in the order of non-decreasing \(p_j\), the jobs in \(B\) follow in the order of non-increasing \(p_j\).

Theorem 5. For the problem \(F2 \mid p_j(a+bt) \mid C_{\text{max}}\), an optimal schedule can be obtained by Johnson’s rule.

Proof. Suppose an optimal \(\pi\) does not satisfy Johnson’s rule, there must be two adjacent jobs \(J_i\) and \(J_k\), \(J_j\) followed by \(J_k\), which satisfy one of the following three conditions:
1) job \(J_i\) belongs to set \(B\) and \(J_k\) to set \(A\);
2) jobs \(J_i\) and \(J_k\) belong to set \(A\) and \(p_{ij} > p_{ik}\);
3) jobs \(J_i\) and \(J_k\) belong to set \(B\) and \(p_{ij} < p_{ik}\).
In what follows we will show that under any of these three conditions the makespan is
reduced after a pair-wise interchange of jobs \( J_i \) and \( J_k \).

Let job \( J_i \) be followed by job \( J_j \) and let \( J_j \) follow job \( J_i \). Perform an adjacent pair-wise interchange on jobs \( J_i \) and \( J_k \), and call the new schedule \( \bar{\pi} \). We denote the completion time of job \( J_i \) on machine \( M_1 \) under \( \pi(\bar{\pi}) \) by \( C_i \). It is obvious that \( C_i = \bar{C}_i \) (if \( i = 1, 2 \)). Interchanging jobs \( J_i \) and \( J_k \) clearly does not affect the starting time of job \( J_i \) on machine \( M_1 \), and it is \( C_{\max} = (C_i \mid T_{1j} \cap T_{1k}) \). Now consider the starting time of job \( J_i \) on machine \( M_2 \). Under \( \pi \), the starting time of job \( J_i \) on machine \( M_2 \) is \( \text{max} \{ C_{\max}(C_i \mid T_{1j} \cap T_{1k} \cap T_{1j} \cap T_{1k}) \} \), under \( \bar{\pi} \), it is \( \text{max} \{ C_{\max}(C_i \mid T_{1j} \cap T_{1j} \cap T_{1j}) \} \).

Under \( \pi \)

\[
C_{2k} = \text{max} \{ C_{\max}(C_{2j} \mid T_{2k}), C_{\max}(C_{1k} \mid T_{2k}) \} = \text{max} \{ C_{\max}(C_{2j} \mid T_{j}), C_{\max}(C_{1k} \mid T_{2k}) \} = \text{max} \{ C_{\max}(C_{2j} \mid T_{j}), C_{\max}(C_{1k} \mid T_{2k}) \}.
\]

Under \( \bar{\pi} \)

\[
\bar{C}_{2k} = \text{max} \{ C_{\max}(C_{2j} \mid T_{1k}), C_{\max}(C_{1k} \mid T_{2k}) \} = \text{max} \{ C_{\max}(C_{2j} \mid T_{j}), C_{\max}(C_{1k} \mid T_{2k}) \} = \text{max} \{ C_{\max}(C_{2j} \mid T_{j}), C_{\max}(C_{1k} \mid T_{2k}) \}.
\]

From Theorem 1, \( C_{\max}(C_{2j} \mid T_{2j} \cap T_{2k}) < C_{\max}(C_{2j} \mid T_{j} \cap T_{2k}) \). Under condition (1), \( p_{ij} \leq p_{jk} \), \( p_{1k} < p_{2k} \). According to Theorem 1 and Corollary 2, \( C_{\max}(C_{2j} \mid T_{1j} \cap T_{2k}) < C_{\max}(C_{1k} \mid T_{1j} \cap T_{2k}) \).

So, \( \bar{C}_{2k} \leq C_{2k} \). Conditions (2) and (3) can be shown in a similar way as condition (1).

3.2 The problem \( Fm \mid p_j(a + bt) \), \( p_q = p_j \mid f(C) \)

For the classical problem \( Fm \mid p_q = p_j \mid f(C) \), the makespan is sequence independent. If the starting time of the first job is \( t \), \( p_j = \text{max}(p_j) \), the makespan is

\[
C_{\max}(\pi) = t + \sum_{i=1}^{n} p_j + (m-1)p_i.
\]

The above solution can be generalized to the problem \( Fm \mid p_j(a + bt), p_q = p_j \mid f(C) \). We consider each operation as a job, and the problem \( Fm \mid p_j(a + bt), p_q = p_j \mid f(C) \) is equivalent to the single problem \( \text{max}(p_j(a + bt) \mid f(C)) \) with \( n \) jobs. Job \( J_i \) is considered as \( m \) jobs (\( p_j = \text{max}(p_j) \)). Hence, for the problem \( Fm \mid p_j(a + bt), p_q = p_j \mid f(C) \), the makespan is

\[
C_{\max}(J_1, \cdots, J_i, \cdots, J_n) = t + (a + bt) \sum_{k=1}^{n-1} b^{t-1} C_i + p_1, \cdots, p_i, \cdots, p_n.
\]

By the results of single machine problem we have the following solutions.

Theorem 6. For the problem \( Fm \mid p_j(a + bt), p_q = p_j \mid C_{\max} \), the makespan is sequence independent.

Theorem 7. For the problem \( Fm \mid p_j(a + bt), p_q = p_j \mid L_{\max} \), an optimal schedule can be obtained by the EDD rule.

Theorem 8. Algorithm 1 generates an optimal schedule for the problem

\[
Fm \mid p_j(a + bt), p_q = p_j \mid h_{\max}(\text{In step 1}, \text{ set } A = \{ J_1, J_2, \cdots, J_i, \cdots, J_n \}).
\]

Not all results of single machine problems can be generalized to the problem \( Fm \mid p_j(a + bt), p_q = p_j \mid f(C) \). It is easily verified that for the problem \( Fm \mid p_j(a + bt), p_q = p_j \mid \sum w_i C_i \), placing jobs in the order of non-decreasing \( p_i/w_i (1 + bp_i) \) does not necessarily minimize the weighted sum of completion times. Moreover, if all jobs have equal weights, we have the following theorem.

Theorem 9. For the problem \( Fm \mid p_j(a + bt), p_q = p_j \mid \sum C_i \), an optimal schedule can be obtained by placing jobs in the order of non-decreasing \( p_i \).
4 Conclusions

In this paper we consider scheduling problems with deteriorating jobs, the deterioration function being \( d(x) = a + bx \). In this model, the actual processing time of a job is the product of its basic processing time and the deterioration function. For single machine problems and flow shop problems, various objective functions, including the makespan, the weighted sum of completion times, the maximum lateness and maximum cost functions are discussed, respectively.

References

8. Johnson S M. Optimal two and three stage production schedules with setup times included. *Naval Research Logistics Quarterly*, 1954, **11**(1): 61~68

ZHAO Chuan-Li  Professor in Department of Mathematics, Shenyang Normal University and a doctor 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.

具有线性恶化加工时间的调度问题

赵传立\(^1\), 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: zhaochuansi@stang.com)

摘要 讨论了工件具有线性恶化加工时间的调度问题. 在这类问题中，工件的恶化函数为线性函数. 对单机调度问题中目标函数为最小化最大完工时间加权完工时间和和，最大延误以及最大费用等问题分别给出了最优算法. 对两台机器极小化最大完工时间的 Flowshop 问题，证明了利用 Johnson 规则可以得到最优调度. 对于一般情况，如果同一工件的工序的加工时间均相等，则 Flowshop 问题可以转化成单机问题.

关键词 调度, 单机, Flowshop, 线性恶化

中国分类号 O223