基于重新排序的退化工件最小化总延误时间问题*

许小艳[1] 慕运动[1,†] 郝 赟[1]

摘要 考虑了错位限制下的含有退化工件的重新排序问题，即工件的实际加工时间看作是工件开工时间的线性函数. 重新排序就是在原始工件已经按照某种规则使目标函数达到最优时有一新工件集到达，新工件的安排使得原始工件重新排序进而产生错位. 研究了最大序列错位和总序列错位限制下的退化工件最小化总延误时间问题，其最优排序的结构性质是使得原始工件集和新工件集中的工件是按加工率α_j非减的序列排列，基于此通过分阶段排序和动态规划方法给出了两个问题的多项式时间的最优算法.

关键词 序列错位，截止日期，总延误时间，实际加工时间

中图分类号 O223

2010 数学分类号 68B20, 90B35

Minimize total lateness with deteriorating jobs based on rescheduling problem*

XU Xiaoyan[1] MU Yundong[1,†] HAO Yun[1]

Abstract In this paper, we consider two single machine rescheduling problems with deteriorating jobs under sequence disruptions, the actual processing time of a job is a linear function of the starting time for deteriorating job. Rescheduling means that a set of original jobs has already been scheduled to minimize some classical objective, then a new set of jobs arrives and creates a disruption. We consider the rescheduling problem to minimize the total lateness under a limit of the sequence disruption for deteriorating job. We research the properties of feasible schedules and optimal schedules for two problems, the jobs in the set of original jobs or new jobs are ordered by non-decreasing order of the processing rate α_j. For each problem, the polynomial algorithm is proposed by sorting by stages or dynamic programming method, respectively.

Keywords sequence disruption, deadline, total lateness, actual processing time

Chinese Library Classification O223

2010 Mathematics Subject Classification 68B20, 90B35

0 引 言

在经典排序中，工件的加工时间一般被认为是一个常数，然而在实际问题中我们常

收稿日期：2013年6月14日

*基金项目：河南省自然科学基金 (No. 112300410078), 河南省教育厅自然科学基金 (No. 2011B110008)

1. 河南工业大学理学院, 郑州 450001; College of Science, Henan University of Technology, Zhengzhou 450001, China

† 通讯作者 Corresponding author, Email: muyundong@163.com

常遇到工件的加工时间和工件的开工时间有关，多位研究者已经将这种情况系统地阐述成不同的模型. 所谓退化工件就是工件的加工时间是工件开工时间的一个函数，含有退化工件的排序问题已经得到了很广泛的研究. Wang等[1]研究了线性退化的单机排序问题，其中工期是根据SLK规则(即每个工件的工期都等于工件的加工时间加上一个正的松弛量)确定的，即$d_j = p_j + q, j = 1, \cdots, n$，目标是使得在不误工的条件下总的加权提前完工时间最小. Wu等[2]考虑了在线性退化模型下目标是使得极小化时间表长的单机问题，他们用分支定界法给出了问题的最优解，两个启发式算法得到了近似最优解. Cheng等[3]考虑了含有截止日期和加工时间增加率的排序问题, 目标是使得最大完工时间和最大延误最小，证明了当工件的加工时间增加率相等时，最大完工时间问题等价于总完工时间和的问题，并且给出了这两个问题的动态规划算法. Wang等[4]考虑了在线性递减退化工件限制下的排序问题，他们分别给出了极小化最大完工时间和最大延误的最优算法.

Hall和Potts[5]系统地研究了重新排序问题，引入了序列错位(sequence disruption)和时间错位(time disruption)的概念，研究了在原来最优排序和任意排序的基础上进行重新排序，即在原来排序的序列错位和时间错位不至于太大的情况下，使目标函数最优的问题，给出了很多基础问题的多项式时间或拟多项式时间算法，或者给出了计算复杂性分析. Yuan和Mu等[6−8]考虑了在时间表长限制下含有准备时间的单机重新排序问题，目标是使得总序列错位最小的问题和最小化最大延迟以及总误工时间问题，给出了相关问题的算法和复杂性. Zhao和Tang[9]研究了具有退化工件的使得总完工时间最小的重新排序问题，对两类问题进行了讨论，给出了多项式时间的最优算法.

本文考虑工件的实际加工时间与工件的开工时间有关，并且工件的工期等于工件的实际加工时间加上一个松弛变量，在序列错位限制下使得总延误时间最小的重新排序问题.

1 问题描述

根据Hall和Potts[5]的描述，单机重新排序问题可以表述为：$J_0 = \{1, \cdots, n_0\}$为单机上的原始工件集，在零时刻J_0中的工件已经按某一规则最优排列好，但还没开始加工，且π^*是J_0的一个最优序列. $J_N = \{n_0+1, \cdots, n_0+n_N\}$为新到达的工件集. 我们要把这些后到达的工件安排到原工件集的最优排序中，但不能过分地扰动原工件集的排列顺序. 记$J = J_0 \cup J_N$. π^*和σ^*分别表示J_0和J中工件的最优序列. 对于J中工件的任意排序σ，我们定义$D_j(\pi^*, \sigma)$表示J_0中工件j的序列错位，即：如果工件j在序列π^*和σ中的位置分别是第x和第y个位置，那么$D_j(\pi^*, \sigma) = |x - y|$；$\sum D_j(\pi^*, \sigma) \leqslant k$表示$\sum_{j \in J_0} D_j(\pi^*, \sigma) \leqslant k$，即工件的总序列错位不超过$k$；$L_j(\sigma)$表示工件$j$的延误时间, 即$L_j = C_j - d_j$；$\sum L_j(\sigma)$表示在序列$\sigma$中, 工件总的延误时间. 在不引起混淆的情况下，$D_j(\pi^*, \sigma)$，$\sum D_j(\pi^*, \sigma)$，$L_j(\sigma)$，$\sum L_j(\sigma)$可以简写为$D_j(\pi^*), \sum D_j(\pi^*), L_j, \sum L_j$.

本文考虑目标函数是总延误时间最小的排序问题，其中工件的实际加工时间与工件的开工时间是呈线性关系的，并且工期是加工时间加上一个正的松弛变量，限制条件是新到达的工件集对原始工件产生的最大序列错位或总序列错位不大于一个正整数k. 我们研究的问题可以用下面的形式表示：

$1|D_{\max}(\pi^*) \leqslant k, p_j = \alpha_j(a + bt), \quad d_j = p_j + q|\sum L_j$；

$1|\sum D_j(\pi^*) \leqslant k, p_j = \alpha_j(a + bt), \quad d_j = p_j + q|\sum L_j$.

2 结构性质

假设所有的工件都在某一时刻t_0到达准备加工，每个工件$j \in J$有一个正的加工时间p_j和一个工期d_j，并假设工件j的实际加工时间p_j是关于工件j开工时间的一个线性函数，即$p_j = \alpha_j(a+bt)$，其中$\alpha_j > 0$ $(j=1,\cdots,n)$, 称为加工率；$a \geqslant 0, b > 0, t \geqslant t_0$是工件的开工时间. 每个工件的工期都是该工件的实际加工时间p_j加上一个正的松弛变量q，即$d_j = p_j + q, j = 1,\cdots,n$. 我们假设$0 < q < \frac{1}{b}(b^3t_0\alpha_1 + b^2t_0 + ab^2t_0\alpha_1 + abt_0 - a)$.

根据文献[9]，我们有下面的引理2.1.

引理 2.1 对于问题$1|p_j = \alpha_j(a+bt)|C_{\max}$，如果$\pi = [1,2,\cdots,n]$，第一个工件的开工时间是$t_0$，那么

$$C_{\max} = t_0\prod_{i=1}^{n}(1+b\alpha_i) + \frac{a}{b}\Big(\prod_{i=1}^{n}(1+b\alpha_i) - 1\Big).$$

引理 2.2 对于一个给定的排序$\pi = [1,2,\cdots,n]$，如果第一个工件的开工时间为t_0，那么第j个工件的实际加工时间为

$$p_j = (a+bt_0)\alpha_j\prod_{i=1}^{j-1}(1+b\alpha_i).$$

引理 2.3 对于一个给定的排序$\pi = [1,2,\cdots,n]$，如果第一个工件的开工时间为$t_0 \geqslant 0$，工件j的工期为$d_j = p_j + q, q \geqslant 0$，那么工件$j$的延误时间为

$$L_j = \Big(t_0 + \frac{a}{b}\Big)\prod_{i=1}^{j-1}(1+b\alpha_i) - \frac{a}{b} - q.$$

证明 对于一个给定的排序$\pi = [1,2,\cdots,n]$，工件j的延误时间为$L_j = C_j - d_j$. 根据引理2.1和引理2.2，我们知道

$$\begin{aligned}
L_j &= t_0\prod_{i=1}^{j}(1+b\alpha_i) + \frac{a}{b}\Big(\prod_{i=1}^{j}(1+b\alpha_i) - 1\Big) - \Big((a+bt_0)\alpha_j\prod_{i=1}^{j-1}(1+b\alpha_i) + q\Big)\\
&= t_0\prod_{i=1}^{j}(1+b\alpha_i) + \frac{a}{b}\prod_{i=1}^{j}(1+b\alpha_i) - \frac{a}{b} - (a+bt_0)\alpha_j\prod_{i=1}^{j-1}(1+b\alpha_i) - q\\
&= \prod_{i=1}^{j}(1+b\alpha_i)[t_0(1+b\alpha_j) + \frac{a}{b}(1+b\alpha_j) - \alpha_j(a+bt_0)] - \frac{a}{b} - q\\
&= \Big(t_0 + \frac{a}{b}\Big)\prod_{i=1}^{j-1}(1+b\alpha_i) - \frac{a}{b} - q.
\end{aligned}$$

由引理2.3我们知道$\sum L_j$的大小仅与α_i $(i=1,\cdots,n)$的排列有关，当加工率α_i是按照由小到大排的时候$\sum L_j$能得到最小值，由此我们得出下面的引理.

引理 2.4 问题$1|p_j = \alpha_j(a+bt), d_j = p_j + q|\sum L_j$通过按加工率$\alpha_j$非减的序列排列，可以得到一个最优解. 如果$\pi = [1, 2, \cdots, n]$，第一个工件的开工时间为$t_0$，那么总延误时间为

$$\sum L_j = \Big(t_0 + \frac{a}{b}\Big)\sum_{j=1}^{n}\prod_{i=1}^{j-1}(1+b\alpha_i) - n\Big(\frac{a}{b}+q\Big).$$

证明 假定某最优排序σ不满足α_j按非减的序列排列规则，则在此排序中，至少有两个相邻工件j与k，使得$\alpha_j > \alpha_k (j > k)$，设工件$j$在时间$t$时开始加工. 现将$\sigma$做如下变换：对调工件$j$与工件$k$的位置，其余工件位置保持不变，从而得到一个新的排序σ'. 在σ'中，工件k的开始加工时间是t. 由$C_k(\sigma) = C_j(\sigma')$可知，其余工件的实际加工时间不变，从而延误时间也不变. 因此σ与σ'总延误时间的差别在于工件j和工件k的延误时间不同. 在σ中，工件j和工件k的延误时间为$C_j - d_j + C_k - d_k = 2t + \alpha_j(a+bt) - 2q$；在$\sigma'$中，工件$k$和工件$j$的延误时间为$C_k - d_k + C_j - d_j = 2t + \alpha_k(a+bt) - 2q$. 从而在两种顺序下，总延误时间之差为

$(\sum L_j)_\sigma - (\sum L_j)_{\sigma'} = 2t + \alpha_j(a+bt) - 2q - [2t + \alpha_k(a+bt) - 2q] = (\alpha_j - \alpha_k)(a+bt) > 0.$
则σ也是最优排序，重复有限次上述变换，可得到一个最优排序，使得该排序是按照α_j的非减序列排列的.

根据引理2.3，可得

$$\sum L_j = \Big(t_0 + \frac{a}{b}\Big)\sum_{j=1}^{n}\prod_{i=1}^{j-1}(1+b\alpha_i) - n\Big(\frac{a}{b}+q\Big).$$

引理 2.5 对于问题$1|D_{\max}(\pi^*) \leqslant k, p_j = \alpha_j(a+bt), d_j = p_j + q|\sum L_j$和$1|\sum D_j(\pi^*) \leqslant k, p_j = \alpha_j(a+bt), d_j = p_j + q|\sum L_j$，存在工件之间没有空闲时间的最优排序，并且

(1) 问题$1|D_{\max}(\pi^*) \leqslant k$, $p_j = \alpha_j(a+bt)$, $d_j = p_j + q|\sum L_j$存在一个可行排序，当且仅当J_N中至多有k个工件被安排在J_0的最后一个工件前；

(2) 问题$1|\sum D_j(\pi^*) \leqslant k$, $p_j = \alpha_j(a+bt)$, $d_j = p_j + q|\sum L_j$存在一个可行排序，当且仅当安排在J_0的每个工件前的J_N中工件的个数和不大于k.

引理 2.6 对于问题$1|D_{\max}(\pi^*) \leqslant k$, $p_j = \alpha_j(a+bt)$, $d_j = p_j+q|\sum L_j$和$1|\sum D_j(\pi^*) \leqslant k$, $p_j = \alpha_j(a+bt)$, $d_j = p_j+q|\sum L_j$，存在一个最优排序，使得J_0中的工件是按加工率α_j非减的序列排列，J_N中的工件也是按加工率α_j非减的序列排列，并且工件之间没有空闲时间.

证明 首先考虑J_0中的工件. 设σ^*是最优序列，但J_0中的工件不是按α_j非减的序列排列的. 记工件i是J_0中在σ^*中出现的位置比在π^*中出现的位置相对较晚，并且具有最小下标，工件$j(j > i)$是在序列σ^*中排在工件i前的J_0中的最后一个工件，排在工件i和工件j之间的工件是$\widetilde{J}_1, \widetilde{J}_2, \cdots, \widetilde{J}_h$，其加工时间分别设为$\widetilde{p}_1, \widetilde{p}_2, \cdots, \widetilde{p}_h$. 因为$i < j$，所以$\alpha_i < \alpha_j$，根据引理2.4及条件$0 < q < \frac{1}{b}(b^3t_0\alpha_1 + b^2t_0 + ab^2t_0\alpha_1 + abt_0 - a)$，也说明了$\sum\limits_{k=1}^{i} L_k(\pi^*) \leqslant \sum\limits_{k=1}^{j} L_k(\pi^*)$. 假设工件$j$在序列$\sigma^*$中开始加工的时间为$t_0$，则

$$L_j(\sigma^*) = t_0 - q, \quad L_i(\sigma^*) = \Big(t_0 + \frac{a}{b}\Big)(1+b\alpha_j)\prod_{k=1}^{h}(1+b\alpha_k) - \frac{a}{b} - q,$$

交换工件i和工件j的位置，得到一个新的序列σ'. 在序列σ'中，工件i的开始加工时间为t_0，则

$$L_i(\sigma') = t_0 - q,\quad L_j(\sigma') = \Big(t_0 + \frac{a}{b}\Big)(1 + b\alpha_i)\prod_{k=1}^{h}(1 + b\alpha_k) - \frac{a}{b} - q.$$

因为$\alpha_i < \alpha_j$,则

$$L_j(\sigma') \leqslant L_i(\sigma^*), L_i(\sigma') \leqslant L_i(\sigma^*).$$

根据引理2.3，工件$\widetilde{J}_1, \widetilde{J}_2, \cdots, \widetilde{J}_h$在序列$\sigma'$中的延误时间比在序列$\sigma^*$中的延误时间小，其余工件的延误时间不变，因此交换后总的延误时间不会增加.

记工件i在π^*中的第k_1个位置上，j 在π^*中的第k_2个位置上，j在σ'中的第k_3个位置上. 如果$k_3 > k_2$，那么$D_j(\pi^*, \sigma') = k_3 - k_2, D_i(\pi^*, \sigma^*) = k_3 - k_1$. 由$k_1 < k_2 (i < j)$可知$D_j(\pi^*, \sigma') < D_i(\pi^*, \sigma^*)$. 如果$k_2 > k_3$，那么$D_j(\pi^*, \sigma') = k_2 - k_3, D_j(\pi^*, \sigma^*) = k_2 - k_3 + h$，其中$h$为非负整数，也就有$D_j(\pi^*, \sigma') < D_j(\pi^*, \sigma^*)$. 因此有$D_{\max}(\pi^*, \sigma') \leqslant D_{\max}(\pi^*, \sigma^*)$.

根据上面的分析可知，σ'是可行最优排序. 重复有限次的上述变换，可以证明存在一个最优排序，使得J_0中的工件是按α_j非减的序列排列的. 对J_N中的工件进行类似的讨论，能够得出在最优排序中J_N中的工件也是按α_j非减的序列排列. 如果工件之间有空闲时间，根据引理2.5，移去空闲时间得到的排序也是可行的，并且总延误时间不会增加.

3 算法设计

我们首先考虑问题

$$1|D_{\max}(\pi^*) \leqslant k\ \ , p_j = \alpha_j(a + bt), d_j = p_j + q|\sum L_j,$$

根据引理2.6中α_j 按非减的序列排列的性质以及引理2.5，我们知道安排在J_0的最后一个工件前至多有k个J_N中的工件，并且J_N中的这些工件的加工率是最小的. 为方便起见，不妨假设J_0中的工件按α_j非减的序列排列的顺序为$[1, \cdots, n_0]$，J_N中的工件按α_j非减的序列排列的顺序为$[n_0 + 1, \cdots, n]$. 因此，我们给出下面的算法.

算法1

输入 输入$\alpha_j, j = 1, \cdots, n, k$和$\pi^*$，其中$k \leqslant n_N$；

标号 对J_N中的工件按α_j非减的序列规则标号；

序列构造 将工件$1, \cdots, n_0 + k$按α_j非减的序列规则排列在前$n_0 + k$个位置上，

将工件$n_0 + k + 1, \cdots, n$按α_j非减的序列规则排列在最后$n_N - k$个位置上.

定理 3.1 对于问题$1|D_{\max}(\pi^*) \leqslant k, p_j = \alpha_j(a + bt), d_j = p_j + q|\sum L_j$，算法1给出一个最优排序，并且其时间复杂性为$O(n + n_N \log n_N)$.

证明 根据引理2.5和引理2.6，限制条件$D_{\max}(\pi^*) \leqslant k$允许排在$J_0$的最后一个工件前$J_N$的工件个数最多为$k$个，并且$J_N$中的这些工件具有最小的$\alpha_j$. 根据引理2.4，第一组的工件是按$\alpha_j$非减的序列规则排列的. 引理2.6则说明了$J_N$中的后$n_N - k$个工件也是按$\alpha_j$非减的序列排列.

下面说明算法1的时间复杂性，在标号阶段，对J_N的标号需要$O(n_N \log n_N)$时间，在序列构造阶段需要$O(n)$时间；将按α_j非减的序列规则排列的J_N中的前k个工件与按π^*中的

顺序排列的J_0中的工件相结合，然后再将按α_j非减的序列规则排列的J_N中的后n_N-k个工件排在最后.

下面考虑问题$1|\sum D_j(\pi^*)\leqslant k, p_j=\alpha_j(a+bt), d_j=p_j+q|\sum L_j$，由引理2.6中$\alpha_j$按非减的序列排列性质知，最优排序可以通过合并按$\alpha_j$非减的序列规则排列的$J_0$和$J_N$中的工件，给出该问题的一个动态规划算法.

算法2

输入 输入t_0,α_j，$j=1,\cdots,n$，k和π^*，其中$k\leqslant n_0n_N$；

标号 对J_N中的工件按α_j非减的序列规则进行排列标号；

预处理 计算

$$\left(t_0+\frac{a}{b}\right)\prod_{h=1}^{i}(1+b\alpha_h)\prod_{k=n_0+1}^{n_0+j}(1+b\alpha_k)-\left(\frac{a}{b}+q\right)\ \ (i=1,2,\cdots,n_0,\ \ j=1,2,\cdots,n_N);$$

目标函数 $f(i,j,\delta)$ =部分序列中的工件$1,\cdots,i$和$n_0+1,\cdots,n_0+j$的最小总延误时间，其中总序列错位为δ；

边界条件 $f(0,0,0)=0$；

最优解值 $\min_{0\leqslant\delta\leqslant k}\{f(n_0,n_N,\delta)\}$；

递推关系

$$\begin{aligned}f(i,j,\delta)=\min\Bigg\{&\left(t_0+\frac{a}{b}\right)\prod_{h=1}^{i-1}(1+b\alpha_h)\prod_{k=n_0+1}^{n_0+j}(1+b\alpha_k)-\left(\frac{a}{b}+q\right)+f(i-1,j,\delta-j),\\&\left(t_0+\frac{a}{b}\right)\prod_{h=1}^{i}(1+b\alpha_h)\prod_{k=n_0+1}^{n_0+j-1}(1+b\alpha_k)-\left(\frac{a}{b}+q\right)+f(i,j-1,\delta)\Bigg\}.\end{aligned}$$

在递推关系中，最小化中的第一部分相当于工件$i\in J_0$排在最后的部分序列，此时工件i的延误时间为

$$\left(t_0+\frac{a}{b}\right)\prod_{h=1}^{i-1}(1+b\alpha_h)\prod_{k=n_0+1}^{n_0+j}(1+b\alpha_k)-\left(\frac{a}{b}+q\right).$$

因为J_N中有j个工件排在这个部分序列中工件i的前面，总序列错位的增加量是j. 第二部分相当于工件$n_0+j\in J_N$排在最后的部分序列, 此时工件n_0+j的延误时间为

$$\left(t_0+\frac{a}{b}\right)\prod_{h=1}^{i}(1+b\alpha_h)\prod_{k=n_0+1}^{n_0+j-1}(1+b\alpha_k)-\left(\frac{a}{b}+q\right).$$

定理 3.2 对于问题$1|\sum D_j(\pi^*)\leqslant k,\ p_j=\alpha_j(a+bt),\ d_j=p_j+q|\sum L_j$，算法2可以得到一个最优序列，且算法复杂性为$O(n_0^2n_N^2)$.

证明 根据引理2.6，仅仅需要列举按α_j非减的序列规则排列的J_0和J_N中的工件的所有可能合并的情况. 算法2是通过比较所有可能状态的费用，从而得到一个最优排序.

下面考虑算法2的时间复杂性，因为$i\leqslant n_0, j\leqslant n_N, \delta\leqslant k\leqslant n_0n_N$，因此有$O(n_0^2n_N^2)$个状态变量；在预处理阶段，计算

$$\left(t_0+\frac{a}{b}\right)\prod_{h=1}^{i}(1+b\alpha_h)\prod_{k=n_0+1}^{n_0+j}(1+b\alpha_k)-\left(\frac{a}{b}+q\right)\ \ (i=1,2,\cdots,n_0,\ \ j=1,2,\cdots,n_N)$$

需要$O(n_0 n_N)$时间. 对J_N中的工件按α_j非减的序列规则进行排列，需要$O(n_N \log n_N)$时间；对每种状态变量的值递推公式的计算需要常数时间，故算法2的时间复杂性为$O(n_0^2 n_N^2)$.

4 结 论

本文研究了工件的实际加工时间与工件的开工时间有关，并且工件的工期等于工件的实际加工时间加上一个松弛变量，在序列错位限制下使得总延误时间最小的重新排序问题. 对于最大序列错位和总序列错位限制下的退化工件最小化总延误时间问题，最优排序的结构性质是使得原始工件集J_0和新工件集J_N中的工件是按加工率α_j非减的序列排列，通过分阶段排序和动态规划的方法给出了两个问题的多项式时间的最优算法.

参 考 文 献

[1] Wang X R, Huang X, Wang J B. Single-machine scheduling with decreasing deterioration to minimize earliness penalties [J]. *Applied Mathematical Modelling*, 2011, **35**: 3509-3515.
[2] Wu C C, Shiau Y R, Lee L H, et al. Scheduling deteriorating jobs to minimize the makespan on a single machine [J]. *The International Journal of Advanced Manufacturing Technology*, 2009, **44**: 1230-1236.
[3] Cheng T C E, Ding Q. Single machine scheduling with deadlines and increasing rates of processing times [J]. *Acta Informatica*, 2000, **36**: 673-692.
[4] Wang J B, Xia Z Q. Scheduling jobs under decreasing linear deterioration [J]. *Information Processing Letters*, 2005, **94**: 63-69.
[5] Hall N G, Potts C N. Rescheduling for new orders [J]. *Operations Research*, 2004, **52**: 440-453.
[6] Yuan J J, Mu Y D. Rescheduling with release dates to minimize makespan under a limit on the maximum sequence disruption [J]. *European Journal of Operational Research*, 2007, **182**: 936-944.
[7] Yuan J J, Mu Y D，Lu L F, et al. Rescheduling with release dates to minimize total sequence disruption under a limit on the makespan [J]. *Asia-Pacific Journal of Operational Research*, 2007, **24**: 789-796.
[8] Mu Y D, Yuan J J. Rescheduling to minimize maximum lateness and total tardiness under compatible job systems [J]. *OR Transactions*, 2007, **11**: 39-48.
[9] Zhao C L, Tang H Y. Rescheduling problems with deteriorating jobs under disruptions [J]. *Applied Mathematical Modelling*, 2010, **34**: 238-243.