运筹学与控制论

DOI:10.11721/cqnuj20130106

时间错位和序列错位呈线性关系的重新排序*

卢宁丹,许小艳,郝 赟,慕运动
(河南工业大学 理学院,郑州 450001)

摘要:讨论了时间错位和序列错位呈线性关系,即最大时间错位与最大序列错位之和、最大时间错位与总序列错位之和、总时间错位与最大序列错位之和、总时间错位与总序列错位之和限制下,以使总完工时间最小为目标的重新排序问题。重新排序就是原始工件已经按照某种规则使目标函数值达到了最优,但还没有开始加工,这时又有一批新工件到达,要求将新工件与原始工件一起重排使目标函数为最优的排序问题。根据经典排序理论,证明了原始工件与新工件按最短加工时间优先规则可以使目标函数达到最优。由动态规划原理,对每个问题设计并证明了不同的算法及其时间复杂性,最后结合实例作了进一步论证。

关键词:重新排序;错位;线性关系;总完工时间和
中图分类号:O223 文献标志码:A 文章编号:1672-6693(2013)01-0025-05

本文研究时间错位和序列错位成线性关系,目标为使总完工时间和最小的重新排序问题。重新排序就是原始工件已经按照某种规则使目标函数值达到了最优,但还没有开始加工,这时又有一批新工件到达,要求将新工件与原始工件一起重排使目标函数为最优的排序问题。Hall 和 Potts[1] 系统地研究了重新排序的经典问题,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的基础上,在原来排序的序列错位和时间错位不至于太大的情况下,使目标函数最优的重新排序问题,并给出了很多基本问题的算法设计,分析了算法的复杂性。Hall,Liu 和 Potts[2] 研究了任意排序下序列错位和时间错位不至于太大的情况下,使目标函数最优的重新排序问题,利用分枝定界的方法确定了问题的上界,给出了启发式算法并利用实例进行了检验。原晋江和慕运动[3-4] 研究了含有到达时间在最大序列错位限制下使总完工时间最小的重新排序问题。Wang 等[5] 研究了线性退化的单机排序问题,其中工期是根据 SLK 规则(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 $d_j=p_j+d$,$j=1,\cdots,n$,目标是使得在不误工的条件下总的加权提前费用最小。Wu C. C. 等[6] 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,用分枝定界法给出了问题的最优解,2 个启发式算法得到了近似最优解。

本文在 Hall 和 Potts 研究的基础上,研究了时间错位和序列错位成线性关系,目标为使总完工时间和最小的 4 个重新排序问题,讨论了它们满足的基本性质,根据动态规划原理分别设计出了它们的算法,并分析了它们的算法复杂性。

1 问题的描述与基本性质

定义 对于排序问题,把任务按加工时间 p_j 不减的顺序排列称为最短加工时间优先(Shortest processing time first)规则,简记为 SPT 规则。

根据 Hall 和 Potts[1] 的描述,单机随机重新排序问题可以表述为:$J_O=\{1,\cdots,n_O\}$ 记作在一台机器上不可中断加工的一组原始工件,假设这些工件已经按 SPT 规则最优的排好,π^* 是这些工件之间没有空闲时间的最优排序。$J_N=\{n_O+1,\cdots,n_O+n_N\}$ 记作新到达的一批新工件,假设这些工件在零时刻到达,这时 J_O 中的工件已经安排好,但还没开始加工。这个假设不失一般性,如果 J_N 中的工件在 t 时刻到达,那么把 J_O 中已经加工完的工件从中移去,正在加工的工件加工完毕后移去,J_O 和 n_O 的值就会相应的发生改变。记 $J=J_O\cup J_N$,$n=n_O+n_N$,p_j 记作工件 $j(j=1,\cdots,n)$ 的加工时间,并且假设 p_j 均为正整数。假设所有的 p_j 都是

* 收稿日期:2012-05-16 修回日期:2012-08-07 网络出版时间:2013-01-18 15:05
资助项目: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);河南工业大学博士科研基金;河南工业大学研究生科技创新基金项目(No. 10XJS056)
作者简介:卢宁丹,女,硕士研究生,研究方向为组合最优化,E-mail:luningdanl@126.com;通讯作者:慕运动,E-mail:muyd@haut.edu.cn
网络出版地址:http://www.cnki.net/kcms/detail/50.1165.N.20130118.1505.201301.25_005.html

已知的，并且记 $P_O=\sum_{j\in J_O}p_j$，$P_N=\sum_{j\in J_N}p_j$。π^* 是 J_O 的一个最优排序，对于 J 的任一个排序 σ，定义如下变量：$C_j(\sigma)$ 表示工件 j 的完工时间，其中 $j\in J$；$D_j(\pi^*,\sigma)=|x-y|$，表示 J_O 中工件 j 的序列错位。x 是 j 在 σ 中的位置，y 是 j 在 π^* 中的位置。$\Delta_j(\pi^*,\sigma)=|C_j(\sigma)-C_j(\pi^*)|$，表示 J_O 中工件 j 的时间错位。$D_{\max}(\pi^*,\sigma)=\max j\in J_O\{D_j(\pi^*,\sigma)\}$，表示 J_O 中工件 j 的最大序列错位。$\Delta_{\max}(\pi^*,\sigma)=\max j\in J_O\{\Delta_j(\pi^*,\sigma)\}$，表示 J_O 中工件 j 的最大时间错位。

在不引起混淆的情况下，可将 $C_j(\sigma)$，$D_j(\pi^*,\sigma)$，$\Delta_j(\pi^*,\sigma)$ 分别简化为 C_j，$D_j(\pi^*)$，$\Delta_j(\pi^*)$。

对于排序问题，Smith[7] 证明了 SPT 规则对于经典问题 $1\|\sum C_j$ 产生一个最优排序。因此，当排序的目标函数是使总完工时间最小时，假设 J_O 的工件在排序 π^* 中是按 SPT 规则排列的。

为了计算方便，假设 π^* 中的工件的排列顺序是 $(1,2,\cdots,n_O)$，并且工件之间没有空闲时间。因此，对于目标函数是总完工时间和的模型，有 $p_1\leqslant\cdots\leqslant p_{n_O}$。

引理1 对于问题

$$1\mid\Delta_{\max}(\pi^*)+D_{\max}(\pi^*)\leqslant k\mid\sum C_j$$

$$1\mid\Delta_{\max}(\pi^*)+\sum D_j(\pi^*)\leqslant k\mid\sum C_j$$

$$1\mid\sum\Delta_j(\pi^*)+D_{\max}(\pi^*)\leqslant k\mid\sum C_j$$

$$1\mid\sum\Delta_j(\pi^*)+\sum D_j(\pi^*)\leqslant k\mid\sum C_j$$

存在一个最优排序，使得 J_O 中的工件的排列顺序和 π^* 一致，按 SPT 规则排列的，J_N 中的工件也是按 SPT 规则排列，并且工件之间没有空闲时间。

证明 首先分析 J_O 中的工件。假设存在一个最优排序 σ^*，在 σ^* 中 J_O 的工件不是按 SPT 规则排列的。记 i 是 J_O 中的工件在 σ^* 中出现的位置比在 π^* 中出现的位置更靠后且具有最小下标，$j(j>i)$ 是在序列 σ^* 中排在工件 i 前的 J_O 中的最后一个工件，排在工件 i 和工件 j 之间的 J_N 的工件是 $\tilde{J}_1,\tilde{J}_2,\cdots,\tilde{J}_h$。因为 $i<j$，所以 $p_i\leqslant p_j$。考虑新序列 σ'，这个新序列 σ' 是在 σ^* 中交换 i 和 j 的位置得到的，使得 $C_i(\sigma')=C_j(\sigma^*)-p_j+p_i\leqslant C_j(\sigma^*)$，$C_j(\sigma')=C_i(\sigma^*)$，$i$ 和 j 之间的所有工件的完工时间在序列 σ^* 中比在序列 σ' 中的完工时间都大 $p_j-p_i\geqslant0$ 个单位，其余工件的完工时间不变，则交换后工件的总完工时间不会增加。如果工件 j 在 σ' 中的位置不比在 π^* 中的位置靠前(对于工件 i 是同样的)，$D_j(\pi^*,\sigma')<D_i(\pi^*,\sigma^*)$，又因为 $D_i(\pi^*,\sigma')=D_i(\pi^*,\sigma^*)-h$，$D_j(\pi^*,\sigma')\leqslant D_j(\pi^*,\sigma^*)+h$，其中 h 表示在 σ^* 中工件 i 和工件 j 的位置差。所以有

$$D_{\max}(\pi^*,\sigma')\leqslant D_{\max}(\pi^*,\sigma^*)$$

$$\sum D_j(\pi^*,\sigma')\leqslant\sum D_j(\pi^*,\sigma^*)$$

若 $C_j(\sigma')\geqslant C_j(\pi^*)$，那么 $\Delta_j(\pi^*,\sigma')<\Delta_i(\pi^*,\sigma^*)$；否则 $\Delta_j(\pi^*,\sigma')<\Delta_j(\pi^*,\sigma^*)$。

另一方面，因为 $\Delta_i(\pi^*,\sigma')=\Delta_i(\pi^*,\sigma^*)-h'$ 和 $\Delta_j(\pi^*,\sigma')\leqslant\Delta_j(\pi^*,\sigma^*)+h'$，其中 $h'=C_i(\sigma^*)-C_i(\sigma')$。根据以上讨论可以得出 $\Delta_{\max}(\pi^*,\sigma')\leqslant\Delta_{\max}(\pi^*,\sigma^*)$ 和 $\sum\Delta_j(\pi^*,\sigma')\leqslant\sum\Delta_j(\pi^*,\sigma^*)$。则有

$$D_{\max}(\pi^*,\sigma')+\Delta_{\max}(\pi^*,\sigma')\leqslant D_{\max}(\pi^*,\sigma^*)+\Delta_{\max}(\pi^*,\sigma^*)$$

$$\sum D_j(\pi^*,\sigma')+\Delta_{\max}(\pi^*,\sigma')\leqslant\sum D_j(\pi^*,\sigma^*)+\Delta_{\max}(\pi^*,\sigma^*)$$

$$D_{\max}(\pi^*,\sigma')+\sum\Delta_j(\pi^*,\sigma')\leqslant D_{\max}(\pi^*,\sigma^*)+\sum\Delta_j(\pi^*,\sigma^*)$$

$$\sum D_j(\pi^*,\sigma')+\sum\Delta_j(\pi^*,\sigma')\leqslant\sum D_j(\pi^*,\sigma^*)+\sum\Delta_j(\pi^*,\sigma^*)$$

因此 σ' 是可行的且是最优的。经过有限次的重复讨论可得到一个最优排序，使得 J_O 中的工件是按 SPT 规则排列。对 J_N 中的工件进行类似的讨论，可得出 J_N 中的工件也是按 SPT 规则排列的。J_O 中的工件的排列顺序与在 π^* 中的排列顺序一致，并且最优排序的工件之间没有空闲时间；否则，移去空闲时间会降低总完工时间，这仍是最优的。 证毕

2 算法设计

首先考虑问题

$$1\mid\Delta_{\max}(\pi^*)+D_{\max}(\pi^*)\leqslant k\mid\sum C_j$$

由引理1给出的最优排序的 SPT 性质，使对于 $j\in J_O$ 由 $C_j\geqslant C_j(\pi^*)$，由动态规划原理可设计下面的算法。

算法1 输入 输入 p_j，$j=1,\cdots,n$，k 和 π^*，其中 $k\leqslant n_N+P_N$；

标号 将 J_N 中的工件按 SPT 规则进行排列标号；

预处理 计算 $\sum_{h=1}^{i}p_h$，$i=1,\cdots,n_O$ 和 $\sum_{h=n_O+1}^{n_O+j}p_h$，$j=1,\cdots,n_N$，当已知序列中最后一个工件是 i 时，$i\in J_O$，记 $\tilde{w}_i$ 是工件 $i-1$ 和工件 $i(i-1,i\in J_O)$ 之间的 J_N 中的工件个数，$\tilde{p}_i$ 是工件 $i-1$ 和工件 $i(i-1,i\in J_O)$ 之间的 J_N 中的工件加工时间和；

目标函数 $f(i,j,\delta)$ 表示当最大序列错位与最大时间错位和等于 δ 时，部分工件 $1,\cdots,i$，和 $n_O+1,\cdots,n_O+j$ 的最小总完工时间；

边界条件 $f(0,0,0)=0$；

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

递推关系：

$$f(i,j,\delta)=\sum_{h=1}^{i}p_h+\sum_{h=1}^{j}p_{n_O+h}+\min\{f(i-1,j,\delta-\tilde{p}_i-\tilde{w}_i),f(i,j-1,\delta)\}$$

在递推关系中,极小化的第一项是该部分序列的最后一个工件是 $i(i\in J_O)$ 的情况,此时的最大序列错位为 j。因此在这个部分排序中,J_N 中的工件 j 在工件 i 的前面,序列错位的增加量就是工件 $i-1$ 和工件 i 之间的 J_N 中的所有工件个数 $\tilde{w}_i$,时间错位的增加量就是工件 $i-1$ 和工件 i 之间的 J_N 中的工件的加工时间和 $\tilde{p}_i$。第二项对应的情况是在这个部分序列中,最后一个工件是 $n_O+j\in J_N$,此时序列错位和时间错位均没有发生变化。

定理 1　对于问题

$$1\mid\Delta_{\max}(\pi^*)+D_{\max}(\pi^*)\leqslant k\mid\sum C_j$$

算法 1 给出了一个最优排序,且其时间复杂性为 $O(n_N\log n_N+n_On_N^2+n_On_NP_N+n)$。

证明　由引理 1 知,它列举并按 SPT 规则排列的 J_O 和 J_N 中的工件,满足错位约束条件的所有可能的方式。算法 1 就是这样通过比较所有可能情形的目标函数,因此可以得到一个最优序列。

现在考虑算法 1 的时间复杂性。因为 $i\leqslant n_O,j\leqslant n_N,\delta\leqslant k\leqslant n_N+P_N$,所以有 $O(n_On_N(n_N+P_N))$ 状态变量值。在标号阶段和预处理阶段,对 J_N 中的工件按 SPT 规则进行标号,计算部分和

$$\sum_{h=1}^{i}p_h,i=1,\cdots,n_O,\sum_{h=n_O+1}^{n_O+j}p_h,j=1,\cdots,n_N$$

需要 $O(n+n_N\log n_N)$ 时间。递推关系在每种状态都需要常数时间。因此算法 1 总的时间复杂性为

$$O(n_N\log n_N+n_On_N^2+n_On_NP_N+n)$$　证毕

假设 $J_O=\{J_1,J_2,J_3,J_4\}$,$J_N=\{J_5,J_6,J_7,J_8\}$,其中工件 J_i 的加工时间为 p_i,$p_1=3$,$p_2=5$,$p_3=7$,$p_4=8$;$p_5=2$,$p_6=4$,$p_7=6$,$p_8=9$。

第 1 步,J_O 和 J_N 中的工件都是按 SPT 规则进行排列的,$k=7$;

第 2 步,$p_1+p_2=8$,$p_1+p_2+p_3=15$,$p_1+p_2+p_3+p_4=23$;$p_5+p_6=6$,$p_5+p_6+p_7=12$,$p_5+p_6+p_7+p_8=21$;

第 3 步,递推求解 $f(1,0,0)=3$,$f(0,1,0)=2$,$\delta=0<7$,根据递推公式知道第 1 个位置应安排工件 J_5;

$$f(1,1,3)=3+2+f(0,1,0)=7$$

$$f(0,2,0)=6+f(0,1,0)=8,\delta=3<7$$

第 2 个位置应安排工件 J_1;

$$f(2,1,3)=8+2+f(1,1,3)=17$$

$$f(1,2,3)=3+6+f(1,1,3)=16,\delta=3<7$$

第 3 个位置应安排工件 J_6;

若第 4 个位置安排 J_O 中的剩余工件的任一个,则 $\delta=8>7$,故第 2 个位置后应安排 J_O 中的工件,且 J_O 中的工件是按 SPT 规则排列的。J_O 中的工件安排结束后再安排 J_N 中按 SPT 规则排列的工件。

第 4 步,故根据算法 1 知,最优的排列顺序为 J_5,J_1,J_2,J_3,J_4,J_6,J_7,J_8。

$$f(3,1,3)=15+2+f(2,1,3)=34$$

$$f(4,1,3)=23+2+f(3,1,3)=79$$

$$f(4,2,3)=23+6+f(4,1,3)=88$$

$$f(4,3,3)=23+12+f(4,2,3)=123$$

$$f(4,4,3)=23+21+f(4,3,3)=167$$

最优解为 167。

下面考虑问题

$$1\mid\Delta_{\max}(\pi^*)+\sum D_j(\pi^*)\leqslant k\mid\sum C_j$$

利用引理 1 类似得到如下动态规划算法。

算法 2　输入　输入 $p_j,j=1,\cdots,n,k$ 和 π^*,其中 $k\leqslant P_N+n_On_N$;

标号　将 J_N 中的工件按 SPT 规则进行排列标号;

预处理　计算 $\sum_{h=1}^{i}p_h,i=1,\cdots,n_O$ 和 $\sum_{h=n_O+1}^{n_O+j}p_h,j=1,\cdots,n_N$,当已知序列中最后一个工件是 i 时,$i\in J_O$,记 $\tilde{p}_i$ 是工件 $i-1$ 和工件 $i(i-1,i\in J_O)$ 之间的 J_N 中的工件加工时间和;

目标函数　$f(i,j,\delta)$ 表示当总序列错位与最大时间错位和等于 δ 时,部分工件 $1,\cdots,i$ 和 $n_O+1,\cdots,n_O+j$ 的最小总完工时间;

边界条件　$f(0,0,0)=0$;

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

递推关系:

$$f(i,j,\delta)=\sum_{h=1}^{i}p_h+\sum_{h=1}^{j}p_{n_O+h}+\min\{f(i-1,j,\delta-\tilde{p}_i-j),f(i,j-1,\delta)\}$$

在递推关系中,极小化的第一项是该部分序列的最后一个工件是 $i(i\in J_O)$ 的情况,此时的最大序列错位为 j。因此在这个部分排序中,J_N 中有 j 个工件排在工件 i 的前面,此时总序列错位的增加量就是 j,最大时间错位的增加量就是工件 $i-1$ 和工件 i 之间的 J_N 中的工件的加工时间和 $\tilde{p}_i$。第二项对应的情况是在这个部分序列中,最后一个工件是 $n_O+j\in J_N$,此时序列错位和时间错位均没有发生变化。

定理 2　对于问题

$$1\mid\Delta_{\max}(\pi^*)+\sum D_j(\pi^*)\leqslant k\mid\sum C_j$$

算法 2 给出了一个最优排序,且其时间复杂性为 $O(n_N\log n_N+n_O{}^2n_N^2+n_On_NP_N+n)$。

定理 2 的证明和定理 1 的类似。

下面考虑问题

$$1 \mid \sum \Delta_j(\pi^*) + D_{\max}(\pi^*) \leqslant k \mid \sum C_j$$

利用引理1类似得到如下算法。

算法3　输入 输入 $p_j, j = 1,\cdots,n, k$ 和 π^*,其中 $k \leqslant n_O P_N + n_N$;

标号 将 J_N 中的工件按SPT规则进行排列标号;

预处理 计算 $\sum_{h=1}^{i} p_h, i = 1,\cdots,n_O$ 和 $\sum_{h=n_O+1}^{n_O+j} p_h, j = 1,\cdots,n_N$,当已知序列中最后一个工件是 i 时,$i \in J_O$,记 $\tilde{w}_i$ 是工件 $i-1$ 和工件 $i(i-1, i \in J_O)$ 之间的 J_N 中的工件个数;

目标函数 $f(i,j,\delta)$ 表示当总序列错位与最大时间错位和等于 δ 时,部分工件 $1,\cdots,i$ 和 $n_O+1,\cdots,n_O+j$ 的最小总完工时间;

边界条件 $f(0,0,0) = 0$;

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

递推关系:

$$f(i,j,\delta) = \sum_{h=1}^{i} p_h + \sum_{h=n_O+1}^{n_O+j} p_h + \min \{f(i-1, j, \delta - \tilde{w}_i - \sum_{h=n_O+1}^{n_O+j} p_h), f(i, j-1, \delta)\}$$

在递推关系中,极小化的第一项是部分序列的最后一个工件是 $i(i \in J_O)$ 的情况,此时在这个部分排序中,J_N 中的工件 j 在工件 i 的前面,序列错位的增加量就是工件 $i-1$ 和工件 i 之间的 J_N 中的所有工件个数 $\tilde{w}_i$,J_N 中的 j 个工件已经排好,这 j 个工件总的加工时间为 $\sum_{h=n_O+1}^{n_O+j} p_h$,这个值就是在这个阶段总时间错位的增加量。第二项对应的情况是在这个部分序列中,最后一个工件是 $n_O + j \in J_N$,此时序列错位和总时间错位均没有变化。

定理3　对于问题

$$1 \mid \sum \Delta_j(\pi^*) + D_{\max}(\pi^*) \leqslant k \mid \sum C_j$$

算法3给出了一个最优排序,且其时间复杂性为

$$O(n_O^2 n_N P_N + n_O n_N^2 + n_N \log n_N + n)$$

定理3的证明和定理1的类似。

下面考虑问题

$$1 \mid \sum \Delta_j(\pi^*) + \sum D_j(\pi^*) \leqslant k \mid \sum C_j$$

利用引理1类似得到如下算法。

算法4　输入 输入 $p_j, j = 1,\cdots,n, k$ 和 π^*,其中 $k \leqslant n_O(P_N + n_N)$;

标号 将 J_N 中的工件按SPT规则进行排列标号;

预处理 计算 $\sum_{h=1}^{i} p_h, i = 1,\cdots,n_O$ 和 $\sum_{h=n_O+1}^{n_O+j} p_h, j = 1,\cdots,n_N$;

目标函数 $f(i,j,\delta)$ 表示当总序列错位与最大时间错位和等于 δ 时,部分工件 $1,\cdots,i$ 和 $n_O+1,\cdots,n_O+j$ 的最小总完工时间;

边界条件 $f(0,0,0) = 0$;

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

递推关系:

$$f(i,j,\delta) = \sum_{h=1}^{i} p_h + \sum_{h=n_O+1}^{n_O+j} p_h + \min \{f(i-1, j, \delta - j - \sum_{h=n_O+1}^{n_O+j} p_h), f(i, j-1, \delta)\}$$

在递推关系中,极小化的第一项是在该部分序列中最后一个工件是 $i(i \in J_O)$ 的情况,此时在这个部分排序中,J_N 中的工件 j 在工件 i 的前面,总序列错位的增加量就是工件 i 前的所有 J_N 中的工件,即总序列错位的增加量为 j,这 j 个工件总的加工时间为 $\sum_{h=n_O+1}^{n_O+j} p_h$,这个值就是在这个阶段总时间错位的增加量。第二项对应的情况是在这个部分序列中,最后一个工件是 $n_O + j \in J_N$,此时序列错位和总时间错位均没有变化。

定理4　对于问题

$$1 \mid \sum \Delta_j(\pi^*) + \sum D_j(\pi^*) \leqslant k \mid \sum C_j$$

算法4给出了一个最优排序,且其时间复杂性为

$$O(n_O^2 n_N P_N + n_O^2 n_N^2 + n_N \log n_N + n)$$

定理4的证明和定理1的类似。

本文利用经典的SPT规则,研究了时间错位和序列错位成线性关系,目标是使得总完工时间和最小的4个重新排序问题,讨论了它们满足的基本性质,根据动态规划原理设计出了它们的拟多项式时间的算法。

参考文献:

[1] Hall N G, Potts C N. Rescheduling for new orders[J]. Operation Research, 2004, 52(3): 440-453.

[2] Hall N G, Liu Z X, Potts C N. Rescheduling for multiple new orders[J]. INFORMS Journal on Computing, 2007, 19(4): 633-645.

[3] 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 Operation Research, 2007, 182(2): 936-944.

[4] 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(6): 789-796.

[5] Wang X R, Huang X, Wang J B. Single-machine scheduling with decreasing deterioration to minimize earliness penalties[J]. Applied Mathematical Modelling, 2011, 35(7): 3509-3515.

[6] Wu C C, Shiau Y R, Lee L H, et al. Scheduling deteriorating jobs to minimize the makespan on a single machine[J]. International Journal of Manufactory Technology, 2009, 44(11/12): 1230-1236.

[7] Smith W E. Various optimizers for single-stage production[J].

Naval Research Logistics, 1956(3): 59-66.

Operations Research and Cybernetics

The Rescheduling Problems with Linear Relationship for Sequence Disruption and Time Disruption

LU Ning-dan, XU Xiao-yan, HAO Yun, MU Yun-dong

(College of Science, Henan University of Technology, Zhengzhou 450001, China)

Abstract: In this paper, we consider the rescheduling problems which sequence disruption and time disruption is a linear relationship, that is maximum sequence disruption and maximum time disruption, maximum sequence disruption and total time disruption, total sequence disruption and maximum time disruption, total sequence disruption and total time disruption, the objective is to minimize the total completion time. Rescheduling problem is a set of original jobs that have been scheduled on a single machine according to some orders. There is another set of new jobs that arrive over time. Preemption is not permitted, so that no interruption in the processing of a job is allowed. The goal is to find a schedule of original jobs set and new jobs set that minimize some objective subject to some disruption constraints. According to classical scheduling theory, we show that shortest processing time first rule is to minimize objective functions, and design some algorithms by dynamic programming, given examples and time complexity of algorithms.

Key words: rescheduling; disruption; malposition; linear relationship; the total completion time

(责任编辑　黄　颖)

外刊摘要转载

Necessary Optimality Conditions and New Optimization Methods for Cubic Polynomial Optimization Problems with Mixed Variables

WU Zhi-you[1], QUN Jing[2], LI Guo-quan[2], TIN Jing[1]

(1. School of Information Technology and Mathematical Sciences, University of Ballarat, Ballarat 3353, Australia;
2. Dept. of Mathematics, Shanghai University, Shanghai 200444, China)

Abstract: Multivariate cubic polynomial optimization problems, as a special case of the general polynomial optimization, have a lot of practical applications in real world. In this paper, some necessary local optimality conditions and some necessary global optimality conditions for cubic polynomial optimization problems with mixed variables are established. Then some local optimization methods, including weakly local optimization methods for general problems with mixed variables and strongly local optimization methods for cubic polynomial optimization problems with mixed variables, are proposed by exploiting these necessary local optimality conditions and necessary global optimality conditions. A global optimization method is proposed for cubic polynomial optimization problems by combining these local optimization methods together with some auxiliary functions. Some numerical examples are also given to illustrate that these approaches are very efficient.

Key words: necessary local optimality conditions; necessary global optimality conditions; cubic polynomial programming problem; local optimizationmethods; global optimization methods

(转载于 Journal of Optimization Theory and Applications, 2012 年 153 卷)