2010 年 9 月 重庆师范大学学报(自然科学版) Sep. 2010
第 27 卷 第 5 期 Journal of Chongqing Normal University(Natural Science) Vol. 27 No. 5

运筹学与控制论

DOI:10.3969/J. ISSN. 1672-6693.2010.05.003

链约束下资源有限的单机排序问题*

金 霁
(苏州市职业大学 基础部,江苏 苏州 215104)

摘要:讨论一类链约束下的资源有限排序问题 $1 \mid \text{chains}, p_{i,j}=b_{i,j}-a_{i,j}u_{i,j}, \sum_{i=1}^{m}\sum_{j=1}^{k_i}u_{i,j} \leqslant \hat{U} \mid \sum_{i=1}^{m}\sum_{j=1}^{k_i}w_{i,j}C_{i,j}$,无论链是否可中断,都给出了启发式算法。对于给定的排列和相应的资源分配量,计算平行链的优先因子,任务按优先因子递增序加工得到一个新的排列,当新排列与原排列不同时,对新排列重新计算对应的资源分配量和优先因子,任务再次按优先因子递增序加工得到一个最新的排列,直到新排列与原排列相同时,停止计算。对于两个启发式算法,分别给出了两个算例对算法加以说明。

关键词:排序;单机;资源有限;链约束;算法
中图分类号:O221.7 文献标识码:A 文章编号:1672-6693(2010)05-0009-05

在经典排序问题中,通常假设工件的加工时间为常数,且不考虑资源消耗问题。但在许多实际问题中,以上假设并不能成立。例如,有 n 批货物 $T_1, T_2, \cdots, T_n$ 要从码头运至工厂,且两批货物不宜同时运输,货物 T_j 的运输时间 p_j 与工厂的投入 u_j 有关,$p_j=f_j(u_j)$。货物若寄存码头,码头要按货物的品种和寄存时间收取费用。第 j 批货物 T_j 的寄存费用与寄存时间 C_j 的关系为寄存费$(T_j)=w_jC_j$。工厂在运回货物时应当怎样分配资源,合理调度 n 批货物的运输,从而使得总费用最少。这是一个典型的资源约束排序问题,所以对此类问题的研究有着积极的实际意义。

Cheng T C E,Janiak A 研究了满足最大完工时间限制条件下极小化资源消耗总量的资源最优控制问题[1]。Janiak A 讨论了满足资源消耗限制条件下极小化最大完工时间问题[2]。王吉波、唐恒永对加权总完工时间受限制条件下,控制工件的资源消耗总量问题进行了讨论,对于一个给定的工件排列,得到了一个关于最优资源分配的算法[3]。唐恒永、赵琨讨论了 3 类任务的加工时间是资源线性函数的特殊情形,给出了最优资源分配,使得加权总完工时间最小[4]。赵传立、唐恒永研究了工件释放时间是所消耗资源的非负严格递减连续函数,工件的加工时间是开工时间的严格递增线性函数,对于最大完工时间限制条件下极小化资源消耗总量问题,以及满足资源消耗限制条件下极小化最大完工时间问题分别给出了求解最优资源分配的方法[5]。闫杨、赵传立考虑了安装时间受资源约束的单机成组加工问题,分别考虑了 3 类问题并分别给出了最优算法[6]。唐恒永等证明了资源约束的排序问题 $1 \mid p_j=b_j-a_ju_j, \sum u_j \leqslant \hat{U} \mid \sum w_jC_j$ 是 NP-难的[7]。当任务之间有优先约束时,问题将变得更加复杂。对于一般的优先约束,问题是 NP-难的[8]。所以本文考虑最简单的优先约束,即平行链约束的情形,待加工任务的实际加工时间受资源约束,目标函数为极小化加权总完工时间。

1 问题的描述

假设有 m 条链

$$L_1: T_{1,1} \to T_{1,2} \to \cdots \to T_{1,k_1}$$
$$L_2: T_{2,1} \to T_{2,2} \to \cdots \to T_{2,k_2}$$
$$\cdots\cdots$$
$$L_m: T_{m,1} \to T_{m,2} \to \cdots \to T_{m,k_m}$$

共 n 个任务$(k_1+k_2+\cdots+k_m=n)$,它们需在同一台机器上加工。任务 $T_{i,j}(i=1,2,\cdots,m, j=1,2,\cdots,k_i)$ 的实际加工时间 $p_{i,j}=b_{i,j}-a_{i,j}u_{i,j}, 0 \leqslant u_{i,j} \leqslant \bar{u}_{i,j}$,其中 $b_{i,j}(b_{i,j}>0)$ 为任务 $T_{i,j}$ 的正常加工时间,$a_{i,j}(a_{i,j}>0)$ 是任务 $T_{i,j}$ 的单位资源分配量系数,$u_{i,j}$

* 收稿日期:2009-12-14 修回日期:2010-07-06
作者简介:金霁,女,讲师,硕士,研究方向为排序。

是待定的对任务 $T_{i,j}$ 的资源分配量,$\bar{u}_{i,j}(0 \leqslant \bar{u}_{i,j} \leqslant \frac{b_{i,j}}{a_{i,j}})$ 是任务 $T_{i,j}$ 的资源分配量的上限,$\hat{U}$ 表示可使用的资源总量。用 T 表示任务集合,z 表示 n 个任务的一个可行下标排列,u 是给定的资源分配向量,Z 是所有可行下标排列集合。$C_{i,j}$ 是任务 $T_{i,j}$ 的完工时间,目标函数为极小化加权总完工时间 $\sum_{i=1}^{m}\sum_{j=1}^{k_i} w_{i,j}C_{i,j}$。该问题可记为(P)

$$1 \mid \text{chains}, p_{i,j} = b_{i,j} - a_{i,j}u_{i,j}$$
$$\sum_{i=1}^{m}\sum_{j=1}^{k_i} u_{i,j} \leqslant \hat{U} \mid \sum_{i=1}^{m}\sum_{j=1}^{k_i} w_{i,j}C_{i,j}$$

2 性质

算法 1[1]

1) 对于给定的下标排列 $z = [z_1, z_2, \cdots, z_n]$

置 $T = \{T_{z_1}, T_{z_2}, \cdots, T_{z_n}\}$,$u_{z_j^*} := 0, j = 1,2,\cdots,n$

2) 找一个 $T_{z_k} \in T$,使 $a_{z_k}\sum_{j=k}^{n} w_{z_j} = \max_{z_i}\{a_{z_i}\sum_{j=i}^{n} w_{z_j}\}$

3) 置 $u_{z_k^*} := \min\{\bar{u}_{z_k}, \max\{0, \hat{U}\}\}$

$\hat{U} := \hat{U} - u_{z_k^*}, T := T - \{T_{z_k}\}$

4) 如果 $T \neq \varnothing$,转 2),否则

置 $\boldsymbol{u}^* = (u_{z_1^*}, u_{z_2^*}, \cdots, u_{z_n^*})$,算法终止。

引理 1[1] 上述算法得到的 $\boldsymbol{u}^*$ 是对应于给定下标排列 z 的最优资源分配。

3 问题的求解

3.1 链不可中断情形

所谓链不可中断情形,指一旦加工某条链就必须加工完该链上的所有任务,然后再接着加工另一条链上的任务。因而关键是要确定 m 条平行链的加工顺序。

定义 1 假设有一条链 L,优先约束为 $T_1 \to T_2 \to \cdots \to T_n$,$u_j$ 是给定的对任务 T_j 资源分配量,称

$$\rho = \frac{\sum_{j=1}^{n}(b_j - a_ju_j)}{\sum_{j=1}^{n} w_j}$$

为链 L 的优先因子。

引理 2[6] 问题 $1 \mid \text{chains}, p_j \mid \sum w_jC_j$ 在链不可中断的情况下,按 $\frac{\sum p_j}{\sum w_j}$ 的单调递增序加工目标函数最优。

定理 1 假设有 m 条链 $L_1, L_2, \cdots, L_m$,在给定的资源分配下,记每条链的优先因子分别为 $\rho_1, \rho_2, \cdots, \rho_m$。则按 ρ_i 的单调递增序加工可得最优排列。

证明 在给定的资源分配下,每个任务的加工时间 $p_j = b_j - a_ju_j (j = 1,2,\cdots,n)$ 就是一个常数。所以,定理 1 等价于问题 $1 \mid \text{chains}, p_j \mid \sum w_jC_j$。 证毕

推论 设 $(z^*, \boldsymbol{u}^*)$ 是问题(P)的最优解,则一定有 $\rho_{z_1^*} \leqslant \rho_{z_2^*} \leqslant \cdots \leqslant \rho_{z_m^*}$,其中 $\rho_{z_i^*}$ 是链 $L_{z_i^*}$ 对应于最优排列 z^* 的优先因子。

由定理 1 及其推论可确定各条链的加工顺序,从而得到下面的启发式算法。

算法 2

1) 任意取一排列 $z^0 \in Z$ 最为初始排列,置 $t := 0$。

2) 调用算法 1 求出对应于 z^t 的最优资源分配 $\boldsymbol{u}^t$。

3) 计算链 $L_i (i = 1,2,\cdots,m)$ 对应于 $\boldsymbol{u}^t$ 的优先因子 $\rho_i^t = \frac{\sum_{j=1}^{k_i}(b_{i,j} - a_{i,j}u_{i,j}^t)}{\sum_{j=1}^{k_i} w_{i,j}}, i = 1,2,\cdots,m$。

4) 按 $\rho_i^t (i = 1,2,\cdots,m)$ 递增的顺序重新排列得到 z^{t+1},如果 $z^{t+1} = z^t$,算法终止。否则,置 $t := t+1$,转 2)。

算例 1 在链不可中断情况下,考虑问题

$$1 \mid \text{chains}, p_j = b_j - a_ju_j, \sum_{j=1}^{n} u_j \leqslant \hat{U} \mid \sum_{j=1}^{n} w_jC_j$$

其中 $n = 7, L_1: T_1 \to T_2 \to T_3 \to T_4, L_2: T_5 \to T_6 \to T_7$,记 $A_{z_i} = a_{z_i}\sum_{j=i}^{n} w_{z_j}$,$\rho_i^t$ 为链 $L_i (i = 1,2)$ 在第 t 次迭代时的优先因子。

$\boldsymbol{b} = (6,4,5,8,5,7,3), \boldsymbol{a} = (2,2,3,1,1,3,1)$

$\boldsymbol{w} = (2,1,1,2,2,3,2), \bar{u} = (2,1,1,4,3,2,2), \hat{U} = 12$

第一次迭代 取初始排列

$$z^0 = [1,2,3,4,5,6,7]$$
$$A_1 = 26, A_2 = 22, A_3 = 30, A_4 = 9$$
$$A_5 = 7, A_6 = 15, A_7 = 2$$
$$u_3^0 = 1, u_1^0 = 2, u_2^0 = 1, u_6^0 = 2, u_4^0 = 4$$
$$u_5^0 = 2, u_7^0 = 0$$
$$p_1 = 2, p_2 = 2, p_3 = 2, p_4 = 4, p_5 = 3$$
$$p_6 = 1, p_7 = 3$$
$$\rho_1^0 = \frac{5}{3}, \rho_2^0 = 1 < \frac{5}{3}$$

所以 $z^1 = [5,6,7,1,2,3,4]$,此时 $\sum w_{z_j^0}C_{z_j^0} = 136$。

第二次迭代:$A_5=13, A_6=33, A_7=8$

$$A_1=12, A_2=8, A_3=9, A_4=2$$

$$u_6^1=2, u_5^1=3, u_1^1=2, u_3^1=1, u_7^1=2, u_2^1=1, u_4^1=1$$

$$p_5=2, p_6=1, p_7=1, p_1=2, p_2=2, p_3=2, p_4=7$$

$$\rho_1^1=\frac{13}{6}, \rho_2^1=\frac{4}{7}<\frac{13}{6}$$

所以 $z^2=[5,6,7,1,2,3,4]$,此时 $\sum w_{z_j^1}C_{z_j^1}=81$。

因为 $z^2=z^1$,算法终止。

故任务加工顺序为 $[L_2,L_1]$,对应的资源分配 $\boldsymbol{u}=(3,2,2,2,1,1,1)$,目标函数值 $\sum_{j=1}^{n} w_{z_j^1}C_{z_j^1}=81$。

3.2 链可中断的情形

所谓链可中断,就是加工某条链时,可以不必加工完该条链上的所有任务,就可以先加工其他链上的任务,然后再回来加工原来那条链上剩余的任务。

定义 2　设有链 $L:T_1\rightarrow T_2\rightarrow\cdots\rightarrow T_k$,$u_j$ 是任务 T_j 所分配到的资源量,定义链 L 的优先因子为

$$\rho=\frac{\sum_{j=1}^{l^*}(b_j-a_ju_j)}{\sum_{j=1}^{l^*}w_j}=\min_{1\leqslant l\leqslant k}\left\{\frac{\sum_{j=1}^{l}(b_j-a_ju_j)}{\sum_{j=1}^{l}w_j}\right\} \tag{1}$$

称 T_{l^*} 为链 L 的关键任务。

定理 2　若 T_{l^*} 是链 $L:T_1\rightarrow T_2\rightarrow\cdots\rightarrow T_k$ 的唯一的关键任务,则在最优解中,任务 $T_1,T_2,\cdots,T_{l^*}$ 连续加工而不被其他链的任务打断。

证明　(反证法)设 u 是对应最优排列的最优资源分配。假设最优排列中含有子排列 $\pi=[T_1,T_2,\cdots,T_u,T_v,T_{v+1},\cdots,T_{v+k_1},T_{u+1},\cdots,T_{u+k_2},T_w,T_{w+1},\cdots,T_{w+k_3},T_{u+k_2+1},\cdots,T_{l^*}]$,其中任务 $T_v,T_{v+1},\cdots,T_{v+k_1},T_w,T_{w+1},\cdots,T_{w+k_3}(k_1\geqslant0,k_2\geqslant0,k_3\geqslant0)$ 都来自于其他链。

根据定理 1 有

$$\frac{\sum_{j=1}^{u}(b_j-a_ju_j)}{\sum_{j=1}^{u}w_j}\leqslant\frac{\sum_{j=v}^{v+k_1}(b_j-a_ju_j)}{\sum_{j=v}^{v+k_1}w_j} \tag{2}$$

$$\frac{\sum_{j=v}^{v+k_1}(b_j-a_ju_j)}{\sum_{j=v}^{v+k_1}w_j}\leqslant\frac{\sum_{j=u+1}^{u+k_2}(b_j-a_ju_j)}{\sum_{j=u+1}^{u+k_2}w_j} \tag{3}$$

$$\frac{\sum_{j=u+1}^{u+k_2}(b_j-a_ju_j)}{\sum_{j=u+1}^{u+k_2}w_j}\leqslant\frac{\sum_{j=w}^{w+k_3}(b_j-a_ju_j)}{\sum_{j=w}^{w+k_3}w_j} \tag{4}$$

$$\frac{\sum_{j=w}^{w+k_3}(b_j-a_ju_j)}{\sum_{j=w}^{w+k_3}w_j}\leqslant\frac{\sum_{j=u+k_2+1}^{l^*}(b_j-a_ju_j)}{\sum_{j=u+k_2+1}^{l^*}w_j} \tag{5}$$

由(2)、(3)式知

$$\frac{\sum_{j=1}^{u}(b_j-a_ju_j)}{\sum_{j=1}^{u}w_j}\leqslant\frac{\sum_{j=u+1}^{u+k_2}(b_j-a_ju_j)}{\sum_{j=u+1}^{u+k_2}w_j} \tag{6}$$

由(4)、(5)式知

$$\frac{\sum_{j=u+1}^{u+k_2}(b_j-a_ju_j)}{\sum_{j=u+1}^{u+k_2}w_j}\leqslant\frac{\sum_{j=u+k_2+1}^{l^*}(b_j-a_ju_j)}{\sum_{j=u+k_2+1}^{l^*}w_j} \tag{7}$$

另一方面,由 T_{l^*} 是链 L 唯一的关键任务

$$\frac{\sum_{j=1}^{u}(b_j-a_ju_j)+\sum_{j=u+1}^{u+k_2}(b_j-a_ju_j)+\sum_{j=u+k_2+1}^{l^*}(b_j-a_ju_j)}{\sum_{j=1}^{u}w_j+\sum_{j=u+1}^{u+k_2}w_j+\sum_{j=u+k_2+1}^{l^*}w_j}<\frac{\sum_{j=1}^{u}(b_j-a_ju_j)}{\sum_{j=1}^{u}w_j} \tag{8}$$

从(8)式可以推得

$$\frac{\sum_{j=u+1}^{u+k_2}(b_j-a_ju_j)+\sum_{j=u+k_2+1}^{l^*}(b_j-a_ju_j)}{\sum_{j=u+1}^{u+k_2}w_j+\sum_{j=u+k_2+1}^{l^*}w_j}<\frac{\sum_{j=1}^{u}(b_j-a_ju_j)}{\sum_{j=1}^{u}w_j} \tag{9}$$

由(6)式可知

$$\frac{\sum_{j=u+1}^{u+k_2}(b_j-a_ju_j)+\sum_{j=u+k_2+1}^{l^*}(b_j-a_ju_j)}{\sum_{j=u+1}^{u+k_2}w_j+\sum_{j=u+k_2+1}^{l^*}w_j}\leqslant\frac{\sum_{j=u+1}^{u+k_2}(b_j-a_ju_j)}{\sum_{j=u+1}^{u+k_2}w_j} \tag{10}$$

从(10)式可以推得

$$\frac{\sum_{j=u+k_2+1}^{l^*}(b_j-a_ju_j)}{\sum_{j=u+k_2+1}^{l^*}w_j}\leqslant\frac{\sum_{j=u+1}^{u+k_2}(b_j-a_ju_j)}{\sum_{j=u+1}^{u+k_2}w_j} \tag{11}$$

这与(7)式矛盾,这说明包含 π 的排列不是最优的。

同理可证链 L 被多次打断的情形。　　证毕

由此可以得到链可中断情形下,问题(P)的启

发式算法。

算法3

1) 置 $L_i := \{T_{i,1}, T_{i,2}, \cdots, T_{i,k_i}\}$, $(i = 1,2,\cdots,m)$,任取一排列 $z^0 \in Z$ 作为初始排列,置 $t := 0$。

2) 调用算法1求出对应于 z^t 的最优资源分配 u^t。

3) 计算链 $L_i(i = 1,2,\cdots,m)$ 对应于 $\boldsymbol{u}^t$ 的优先因子

$$\rho_i^t = \frac{\sum_{j=1}^{l_i^*}(b_{i,j} - a_{i,j}u_{i,j}^t)}{\sum_{j=1}^{l_i^*} w_{i,j}} = \min_{1\leqslant l_i\leqslant k_i}\left\{\frac{\sum_{j=1}^{l_i}(b_{i,j} - a_{i,j}u_{i,j}^t)}{\sum_{j=1}^{l_i} w_{i,j}}\right\}, i = 1,2,\cdots,m$$

4) 在当前未加工的链中,选择优先因子最小的那条链,连续加工该链中的任务,直到加工完该链的关键任务 T_{i,l_i^*}, $L_i := L_i - \{T_{i,1},\cdots,T_{i,l_i^*}\}$, $(i = 1,2,\cdots,m)$。

如果 $L_i \neq \varnothing$,则计算链 L_i 余下任务的优先因子。

5) 如果已加工完全部任务,则置 $t := t + 1$;否则,转4)。

6) 如果 $z^{t+1} = z^t$,算法终止。否则,转2)。

算例2 在链可中断情况下,考虑问题

$$1 \mid \text{chains}, p_j = b_j - a_j u_j, \sum_{j=1}^{n} u_j \leqslant \hat{U} \mid \sum_{j=1}^{n} w_j C_j$$

其中 $n = 7$, $L_1: T_1 \to T_2 \to T_3 \to T_4$, $L_2: T_5 \to T_6 \to T_7$,记

$$A_{z_i} = a_{z_i}\sum_{j=i}^{n} w_{z_j}, \boldsymbol{b} = (6,4,5,6,5,8,4)$$

$$\boldsymbol{a} = (2,2,3,1,1,3,1), w = (1.5,1,1,2,2,4,2)$$

$$\bar{u} = (2,1,1,4,3,2,2), \hat{U} = 12$$

第一次迭代 取初始排列

$$z^0 = [1,2,3,4,5,6,7]$$

$$A_1 = 27, A_2 = 24, A_3 = 33, A_4 = 10, A_5 = 8$$

$$A_6 = 18, A_7 = 2, u_3^0 = 1, u_1^0 = 2$$

$$u_2^0 = 1, u_6^0 = 2, u_4^0 = 4, u_5^0 = 2, u_7^0 = 0$$

$$p_1 = 2, p_2 = 2, p_3 = 2, p_4 = 2, p_5 = 3$$

$$p_6 = 2, p_7 = 4, \sum w_{z_j^0} C_{z_j^0} = 137$$

第一条链的优先因子为

$$\rho_{z_1^0}(1,2,3,4) = \min\left\{\frac{2}{1.5},\frac{4}{2.5},\frac{6}{3.5},\frac{8}{5.5}\right\} = \frac{4}{3}$$

T_1 是关键任务。

第二条链的优先因子为

$$\rho_{z_2^0}(5,6,7) = \min\left\{\frac{3}{2},\frac{5}{6},\frac{9}{8}\right\} = \frac{5}{6}$$

T_6 是关键任务。

因为 $\rho_{z_2^0}(5,6,7) < \rho_{z_1^0}(1,2,3,4)$,所以排列 z^1 中前两个任务为 T_5, T_6。

第一条链中余下的任务为 T_2, T_3, T_4,此时

$$\rho_{z_1^0}(2,3,4) = \min\left\{\frac{2}{1},\frac{4}{2},\frac{6}{4}\right\} = 1.5$$

T_4 是关键任务。

由于 $\rho_{z_1^0}(1,2,3,4) = \dfrac{4}{3} < \rho_{z_1^0}(2,3,4)$,所以任务 T_2, T_3, T_4 应该在 T_1 后加工。

第二条链中仅余下任务 T_7, $\rho_{z_2^0}(7) = 2$。由于

$$\rho_{z_1^0}(2,3,4) = 1.5 < \rho_{z_2^0}(7)$$

故 T_7 在 T_2, T_3, T_4 之后加工,即排列 $z^1 = [5,6,1,2,3,4,7]$。

第二次迭代:$A_5 = 13.5, A_6 = 34.5$

$$A_1 = 15, A_2 = 12, A_3 = 15, A_4 = 4, A_7 = 2$$

$$u_6^1 = 2, u_1^1 = 2, u_3^1 = 1, u_5^1 = 3, u_2^1 = 1, u_4^1 = 3, u_7^1 = 0$$

$$p_5 = 2, p_6 = 2, p_1 = 2, p_2 = 2, p_3 = 2, p_4 = 3, p_7 = 4$$

$$\sum w_{z_j^1} C_{z_j^1} = 107$$

优先因子 $\rho_{z_1^1}(5,6) = \min\left\{\dfrac{2}{2},\dfrac{4}{6}\right\} = \dfrac{2}{3}$, T_6 是关键任务,优先因子 $\rho_{z_2^1}(1,2,3,4) = \min\left\{\dfrac{2}{1.5},\dfrac{4}{2.5},\dfrac{6}{3.5},\dfrac{9}{5.5}\right\} = \dfrac{4}{3}$, T_1 是关键任务,优先因子 $\rho_{z_3^1}(7) = 2$, T_7 是关键任务。

由于 $\rho_{z_1^1}(5,6) < \rho_{z_2^1}(1,2,3,4) < \rho_{z_3^1}(7)$,从而排列 z^2 中前两个任务为 T_5, T_6,且 T_1 在 T_7 前加工。

第一条链余下 T_2, T_3, T_4, $\rho_{z_2^1}(2,3,4) = \min\left\{\dfrac{2}{1},\dfrac{4}{2},\dfrac{7}{4}\right\} = \dfrac{7}{4}$, T_4 是关键任务。

因为 $\rho_{z_2^1}(1,2,3,4) = \dfrac{4}{3} < \rho_{z_2^1}(2,3,4) < \rho_{z_3^1}(7)$,所以 T_1 在 T_2, T_3, T_4 前加工,T_7 在 T_2, T_3, T_4 之后加工。故排列 $z^2 = [5,6,1,2,3,4,7]$。

由于 $z^2 = z^1$,故算法终止。

而事实上,排列 $z^* = [5,6,7,1,2,3,4]$ 对应的目标函数值 $\sum w_{z_j^*} C_{z_j^*} = 100 < 107$。

定理3 对于问题(P),若 $\sum_{i=1}^{m}\sum_{j=1}^{k_i}\bar{u}_{i,j} \leqslant \hat{U}$,则由算法2及算法3得到的解一定是最优解。

证明　若$\sum_{i=1}^{m}\sum_{j=1}^{k_i}\bar{u}_{i,j} \leqslant \hat{U}$,则最优资源分配方案 u^* 中每个任务的资源分配量都等于上限,即 $u_{i,j}^* = \bar{u}_{i,j}$,$(i = 1,2,\cdots,m,j = 1,2,\cdots,k_i)$。

任务的加工时间 $p_{i,j} = b_{i,j} - a_{i,j}\bar{u}_{i,j}$ 是个常数,因而问题(P)等价于问题(q)

$$1 \mid \text{chains}, p_{i,j} \mid \sum_{i=1}^{m}\sum_{j=1}^{k_i} w_{i,j}C_{i,j}$$

而问题(q)有最优解[6]。　　证毕

4 结束语

本文对资源有限的单机排序问题进行了讨论,分别揭示了链不可中断和链可中断两种情形下最优解的性质,并分别给出了两个启发式算法。最后,通过两个数值例子对算法加以说明。

参考文献:

[1] Cheng T C E, Janiak A. Resource optimal control in some single-machine scheduling problem[J]. IEEE Trans Autom Control, 1994, 39: 1234-1246.

[2] Janiak A. Time-optimal control in single machine problem resource constraints [J]. Automatica, 1986, 22: 745-747.

[3] 王吉波,唐恒永. 一类资源约束排序问题 $1 \mid p_j = b_j - a_ju_j, \sum w_jC_j \leqslant A \mid \sum u_j$ [J]. 辽宁大学学报(自然科学版), 2001, 28(4): 310-312.

[4] 唐恒永,赵珉. 资源有限的加权总完工时间单机排序问题[J]. 运筹与管理, 2004, 13(3): 48-51.

[5] 赵传立,唐恒永. 一类资源约束单机排序问题[J]. 系统工程学报, 2004. 19(5): 451-456.

[6] 闫杨,赵传立. 安装时间受资源约束的单机成组调度问题[J]. 电机与控制学报, 2007. 11(1): 70-73.

[7] 唐恒永,赵传立,赵珉. 一类加工时间依赖资源的排序问题[J]. 系统工程, 2004. 22(7): 9-12.

[8] 唐恒永,赵传立. 排序引论[M]. 北京:科学出版社, 2005. 1.

Operations Research and Cybernetics

Single Machine Scheduling Problem of Resource Constrained Under Chains Constraints

JIN ji

(Dept. of Fundamental, Suzhou Vocational University, Suzhou Jiangsu 215104, China)

Abstract: In this paper, we discuss the following resource constrained scheduling problem $1 \mid \text{chains}, p_{i,j} = b_{i,j}\text{-}a_{i,j}u_{i,j}, \sum_{i=1}^{m}\sum_{j=1}^{k_i} u_{i,j} \leqslant \hat{U} \mid \sum_{i=1}^{m}\sum_{j=1}^{k_i} w_{i,j}C_{i,j}$. Whether the chains nonpreemptive or not, two heuristic algorithms are given. For an given permutation and resource allocation of this permutation, by calculating priority factor of every chain, we can find another new permutation according to the increasingly priority factor. If the new permutation is different from the old one, for the new permutation, calculating resource allocation and priority factor respectively, a new permutation according to the increasingly priority factor is obtained. If the new permutation is the same as the old one, the algorithm stops. It also illustrates the algorithms by two examples.

Key words: scheduling; single machine; resource constrained; chains constraints; algorithm

(责任编辑　黄　颖)