运筹学与控制论

DOI:10.3969/J. ISSN. 1672-6693. 2010. 02. 002

两个多重目标排序问题的多项式时间算法*

彭洪洁[1]，唐国春[1,2]
(1. 重庆师范大学 数学学院，重庆 400047；2. 上海第二工业大学 管理工程研究所，上海 201209)

摘要：多目标排序是排序论的一个重要分支，在解决经济、管理、工程、军事、社会等领域出现的复杂问题中起着越来越重要的作用。本文研究以误工个数$\sum U_j$为第 1 目标，$\sum w_jC_j$或者$\sum w_jT_j$为第 2 目标的多重目标排序问题，分别给出了这两个问题在不误工工件集不改变下工件加工时间和权重满足反一致性条件($p_i \leqslant p_j \Rightarrow w_i \geqslant w_j$)时复杂性为$O(n\log n)$的多项式时间算法：对于排序问题$1|(p_i \leqslant p_j) \Rightarrow (w_i \geqslant w_j)|(\sum w_jC_j/E)$，选取排序最后一个工件$k$满足条件：$p_k/w_k = \max\{p_i/w_i | i \in M \cup L\}$；对于排序问题$1|(p_i \leqslant p_j) \Rightarrow (w_i \geqslant w_j)|(\sum w_jT_j/E)$，选取排序最后一个工件$k$满足：1)若$M$为空集，$p_k/w_k = \max\{p_i/w_i | i \in L\}$；2)若$M$非空，任意选取$k \in M$。其中$L$是误工工件集，$M$是放在最后不误工的工件的集合。最后，证明了这两个算法可以得到相应问题的最优解。

关键词：排序；误工；算法；多目标；计算复杂性；最优性
中图分类号：O223　　文献标识码：A　　文章编号：1672-6693(2010)02-0004-05

多目标排序是排序论的重要分支。作为多目标决策问题，多目标排序在经济、管理、工程、军事、社会等领域中起着越来越重要的作用。例如，在生产过程中，决策者不但要使误工个数$\sum U_j$最小，同时还要满足顾客的需求使得总的完工时间$\sum C_j$最小或者最大延误$T_{\max}$最小等。在多目标排序中有两个或两个以上目标函数。如果是在第 1 个目标函数γ_1为最优的条件下，使第 2 个目标函数γ_2为最优，这样的问题称为第 1 类多目标排序问题，得到的最优解称为多重解。1956 年 Smith 研究在没有工件误工的、所谓完美的(Perfect)排序中寻找使平均完工时间(也就是等价使总的完工时间)为最小的排序问题，提出寻找这种最优解的 Smith 法则或者 Smith 算法，是研究第 1 类多目标排序多重解的第一篇论文[1]。经典排序论中，使误工工件的个数为最少的排序问题通常称为误工问题[2]，是排序论中最基本的问题之一，具有重要的理论意义和实用价值。1968 年 Moore 提出解决单台机器误工排序问题的多项式时间算法[3]，这个算法称为 Moore 算法，可以在时间$O(n\log n)$内得到误工问题的最优解。之后很多学者研究各种推广的误工问题[4-17]。1973 年，Sidney 研究在工件的一个子集T中的工件必须不误工的条件下，使误工工件的个数为最少的误工排序问题$1|T|\sum U_j$，并且给出该问题复杂性为$O(n\log n)$的多项式时间算法——Sidney 算法[4]。还有学者考虑以误工个数为第 1 目标的多重目标排序问题[18-23]。1975 年，Emmons 在文献[20]中推广了 Smith 法则给出在误工工件集给定下使$\sum C_j$最优的多项式时间算法。1983 年，Shanthikumar 在文献[21]中给出了在误工工件集给定下使$T_{\max}$最优的多项式时间算法。1995 年 Vairaktarakis 和 Lee 也在文献[22]中类似的给出了在误工工件集给定下使$\sum T_j$最优的多项式时间算法。最新的成果是 2007 年 Huo，Leung，Zhao 等证明，误工工件个数为第 1 目标、使总完工时间最小的多重目标排序问题$1\|(\sum C_j/\sum U_j)$或者使总延误最小的多重目标排序问题$1\|(\sum T_j/\sum U_j)$都是 NP 困难的[23]，解决国内外学术界长期来都十分关注的两个 open problems(悬而未决的问题)。本文研究以误工个数$\sum U_j$为第 1 目标，$\sum w_jC_j$或者$\sum w_jT_j$为第 2 目标的 strongly-NP-hard 多重目标排序问题[2]，给出了在误工工件集已知下的的多项式时间算法，并且证明在加工时间与权具有‘反向一致性’的前提下可得到最优解。

* 收稿日期：2009-10-12　　修回日期：2010-01-31
资助项目：国家自然科学基金项目(No. 70731160015，No. 0710015)；运筹学与系统工程重庆市市级重点实验室资助(No. YC200802)
作者简介：彭洪洁，男，硕士研究生，研究方向为组合最优化；通讯作者：唐国春，E-mail：gtang@sh163.net

1 初步结果

设有 n 个工件 $J_1,J_2,\cdots,J_n$ 要在一台机器上加工。记这 n 个工件的集合为 $J=\{J_1,J_2,\cdots,J_n\}$。在不至于混淆的情况下,也用工件 J_j 的下标 j 来表示这个工件。因此,这 n 个工件的集合也可以写成 $J=\{1,2,\cdots,n\}$。已知工件 j 的加工时间是 p_j,权值是 w_j,交货期是 d_j。这些工件已经全部准备就绪,都可以进行加工。这台机器每次只能加工一个工件,并且加工不允许中断,只有当一个工件加工完成后才能加工其他工件,如果一个工件在交货期之后完工,这个工件称为是误工的,否则称为是不误工的。安排一个加工次序,使误工工件的个数为最少,这就是误工(排序)问题,用三参数表示为 $1\parallel\sum U_j$[2]。如果要求某些指定的工件必须不误工,以 T 表示指定必须不误工的工件的集合,又使误工工件的个数为最少,这就是使某些工件必须不误工条件下的误工排序问题,用三参数表示为 $1\mid T\mid\sum U_j$[2]。记工件 j 在排序 $S=(s(1),s(2),\cdots,s(n))$ 中的完工时间为 C_j,延误为 T_j,带权总完工时间为 $\sum w_jC_j$,带权总延误为 $\sum w_jT_j$。在误工工件个数 $\sum U_j$ 最少的条件下,使 $\sum w_jC_j$ 或者 $\sum w_jT_j$ 分别为最小的多重目标排序问题,用三参数可以表示为 $1\parallel(\sum w_jC_j/\sum U_j)$、$1\parallel(\sum w_jT_j/\sum U_j)$[2]。不失一般性,假设工件已经按照交货期的从小到大的次序,即 EDD(Earliest due date)序进行编号,所以有 $d_1\leqslant d_2\leqslant\cdots\leqslant d_n$。这时工件 j 的下标是工件在 EDD 序中的序号。对于工件集合 J 的子集 σ,在不至于混淆的情况下,σ 也表示这些工件按照下标从小到大排成的排序,即工件集合 σ 的 EDD 序。

下面引理 1 和引理 2 的证明与文献[15]中类似,在此略去。

定义 1[15] 如果排序 σ 中每一个工件都是不误工的,则把子集 σ 称为是不误工子集。

引理 1 如果排序 $\sigma=\{t_1,t_2,\cdots,t_m\}$ 中的工件都是不误工的,那么排序 σ 的 EDD 序中的工件也都是不误工的。反之亦然。

定义 1 是根据 EDD 序来定义不误工子集,有了引理 1 可以把定义 1 推广到一般。

定义 2[15] 对于工件集 J 的子集 σ,如果在子集 σ 的一个排序中这些工件都是不误工的,那么这个子集称为是不误工子集。

有了不误工子集的定义,下面给出误工工件集的定义。

定义 3 如果工件集 J 的最大不误工子集为 E,那么子集 $L=J\backslash E$ 称为误工问题的误工工件集。

引理 2 误工问题存在这样的最优解,可以分为前后两部分:1)不误工的工件的全体(不误工工件集)E 按 EDD 序排在前面;2)误工工件集 L,以任意次序排在后面。

文献[16]把这种形式的最优解称为是误工排序问题的 E-L 最优解。那么对于推广的误工排序问题 $1\mid T\mid\sum U_j$ 的这种形式的最优解称为是推广误工排序问题 $1\mid T\mid\sum U_j$ 的 E-L 最优解。

1973 年,Sidney 研究在工件的一个子集 T 中的工件必须不误工的条件下,使误工工件的个数为最少的误工排序问题 $1\mid T\mid\sum U_j$,并且给出该问题复杂性为 $O(n\log n)$ 的多项式时间算法—Sidney 算法[4]。此算法可以写成如下形式[17]。

步骤 1 设 $E_0=T$,$J-E_0=\{j_1,j_2,\cdots,j_m\}$,$j_1<j_2<\cdots<j_m$,$m=n-|T|$,令 $k=1$;

步骤 2 若 $k=m+1$,算法终止,$(E_m,J-E_m)$ 就是最优排序;若 $k<m+1$,转入步骤 3;

步骤 3 设 $F_k=E_{k-1}\cup\{j_k\}$,计算 E_k 如下:如果 F_k 是不误工子集,令 $E_k=E_{k-1}\cup\{j_k\}$;否则,如果 F_k 不是不误工子集,令 $E_k=F_k\backslash\{j_r\}$。其中工件 j_r 的加工时间为 $p_r=\max\{p_i\mid j_i\in F_k\backslash T\}$。$E_k$ 中的工件是按 EDD 序排列。令 $k=k+1$,转入步骤 2。

此排序问题 $1\mid T\mid\sum U_j$ 也可以写成多目标排序的形式:$1\parallel(\sum U_j/T)$。为了方便起见,现假设文章的后面部分的误工工件集 L 是此算法得到的。

2 误工工件集给定的多项式时间算法

排序问题 $1\parallel(\sum w_jC_j/\sum U_j)$ 和 $1\parallel(\sum w_jT_j/\sum U_j)$ 的一般情形是 strongly-NP-hard[2],下面分别给出这两

个问题在误工工件集 L 已知下的多项式时间算法。令 $E=J\backslash L$，那么在不误工工件集不变的前提下，使带权完工时间或者带权延误最小的排序问题用三参数表示为：$1\mid E\mid\sum w_jC_j$ 和 $1\mid E\mid\sum w_jT_j$，写成多目标排序形式为 $1\parallel(\sum w_jC_j/E)$ 和 $1\parallel(\sum w_jT_j/E)$。记 $M=\{i\mid d_i\geqslant\sum_{j=1}^{n}p_j\}$。

定理1　对于排序问题 $1\mid(p_i\leqslant p_j)\Rightarrow(w_i\geqslant w_j)\mid(\sum w_jC_j/E)$，选取排序最后一个工件 k 满足条件

$$p_k/w_k=\max\{p_i/w_i\mid i\in M\cup L\}$$

其中 $M\cup L$ 中的工件是误工工件或者是放在最后不误工的工件。

证明　定理1等价于下面的算法1，证明算法1可得到最优解即可。

算法1

步骤0　初始化，$Q=N=(1,2,\cdots,n)$，$E'=N-E$；$r=n$，转到第一步。

步骤1　求得集合 $M=\{j:d_j\geqslant\sum_{i\in Q}p_i;j\in Q\}$ 找到集合 $M\cup E'$，即误工工件的集合 E'，或者排在最后不误工的工件集合 M，转下一步。

步骤2　找出 $M\cup E'$ 中的工件 k 满足 $p_k/w_k=\max\{p_i/w_i\mid i\in M\cup E'\}$，转下一步。

步骤3　置 $a_r=k$，$Q=Q-\{k\}$，$E'=Q\cap\{N-E\}$，$r=r-1$，如果 $r=0$，$S=(a_1,a_2,\cdots,a_n)$ 是最后的排序。算法终止；否则，转步骤1。　　**证毕**

定理2　算法1产生的排序是问题 $1\mid(p_i\leqslant p_j)\Rightarrow(w_i\geqslant w_j)\mid(\sum w_jC_j/E)$ 的最优解。

证明　不失一般性，考虑排序 $R=(b_1,b_2,\cdots,b_n)$，并且令 $u=(b_1,b_2,\cdots,b_r)$ 为排序 R 的前 r 个工件。记 $Q=\{u\}$ 是 u 中所有工件的集合。假设 Q 中有一个工件 b_k 满足：$p_k/w_k=\max\{p_i/w_i\mid i\in M\cup E'\}$，$b_k\in M\cup E'$，其中 $M=\{j:d_j\geqslant\sum_{i\in Q}p_i;j\in Q\}$，$E'=Q\cap\{N-E\}$，那么 $R\neq S$，因此 R 不是算法1产生的排序。假设算法1得出的解中后 $n-r$ 个工件的次序与 R 相同，根据算法1，在选择第 r 个工件，应该把 b_k 作为第 r 个工件，若把 b_k 与 b_r 交换，其他位置不变得到的新排序为 R'

$$R=(b_1,b_2,\cdots,b_{k-1},b_k,b_{k+1},\cdots,b_r,b_{r+1},\cdots,b_n)$$
$$R'=(b_1,b_2,\cdots,b_{k-1},b_r,b_{k+1},\cdots,b_k,b_{r+1},\cdots,b_n)。$$

比较这两个排序，则有

1）$C_i(R)=C_i(R')$，$i=1,2,\cdots,k-1$，那么 $w_iC_i(R)=w_iC_i(R')$；

2）$C_r(R)=C_k(R')$；

3）$C_i(R)=C_i(R')$，$i=r+1,\cdots,n$，那么 $w_iC_i(R)=w_iC_i(R')$。

其中 $C_i(R)$ 和 $C_i(R')$ 分别表示工件 b_i 在排序 R 和 R' 中的完工时间。

因此，两个排序的带权总完工时间之差

$$\sum_{i=1}^{n}(w_iC_i(R')-w_iC_i(R))=(p_r-p_k)(w_{k+1}+\cdots+w_{r-1})+(w_k-w_r)(p_{k+1}+\cdots+p_{r-1})+w_kp_r-w_rp_k$$

首先证明 $r\in M\cup E'$，若 $r\in E'$，已证；否则 $r\notin E'$，即 $r\in E$，从而 b_r 在任何可行排序中不延误，由 $C_r(R)=\sum_{i\in Q}p_i\leqslant d_r$ 知 $r\in M$。因为 $p_k/w_k=\max\{p_i/w_i\mid i\in M\cup E'\}$，所以 $p_k/w_k\geqslant p_r/w_r$，有 $w_kp_r-w_rp_k\leqslant 0$，并且加工时间与权具有“反一致性”关系，所以 $p_k\geqslant p_r$，$w_k\leqslant w_r$。因此 $\sum_{i=1}^{n}(w_iC_i(R')-w_iC_i(R))\leqslant 0$。

又因 $b_k\in M\cup E'$ 和 $p_k\geqslant p_r$，所以交换工件 b_k 与 b_r 得到的新排序 R' 中每个工件完工时间不增加，给定的不误工工件集不变。这样说明，选取算法1选择的工件可以得到一个更好的排序。由于 r 的任意性，算法1产生的排序是问题的最优解。　　**证毕**

定理3　对于排序问题 $1\mid(p_i\leqslant p_j)\Rightarrow(w_i\geqslant w_j)\mid(\sum w_jT_j/E)$，选取排序最后一个工件 k 满足

1）若 M 为空集，$p_k/w_k=\max\{p_i/w_i\mid i\in L\}$；

2）若 M 非空，任意选取 $k\in M$。

证明 上面的定理3等价于下面算法2,证明算法2可得到最优解即可。

算法2

步骤0 初始化,$Q=N=(1,2,\cdots,n)$,$E'=N-E$;$r=n$,转到第一步。

步骤1 求得集合 $M=\{j: d_j\geqslant\sum_{i\in Q}p_i; j\in Q\}$ 找到集合 $M\cup E'$,即误工工件的集合 E',或者排在最后不误工的工件集合 M,转下一步。

步骤2 如果 M 为空集,找出工件 k 满足 $p_k/w_k=\max\{p_i/w_i \mid i\in E'\}$ 如果 M 非空,在 M 中任意选取一个工件作为 k,转下一步。

步骤3 置 $a_r=k$,$Q=Q-\{k\}$,$E'=Q\cap\{N-E\}$,$r=r-1$,如果 $r=0$,$S=(a_1,a_2,\cdots,a_n)$ 是最后的排序。算法终止;否则,转步骤1。 证毕

定理4 算法2产生的排序是问题 $1\mid(p_i\leqslant p_j)\Rightarrow(w_i\geqslant w_j)\mid(\sum w_jT_j/E)$ 的最优解。

证明 不失一般性,考虑排序 $R=(b_1,b_2,\cdots,b_n)$,并且令 $u=(b_1,b_2,\cdots,b_r)$ 为排序 R 的前 r 个工件。记 $Q=\{u\}$ 是 u 中所有工件的集合。分两种情况证明。

1)若 M 非空,则有 $b_k\in M$,因此 $R\neq S$,那么 R 不是算法2产生的排序,排序 R 中的工件 b_r 是误工工件。假设算法2得出的解中后 $n-r$ 个工件的次序与 R 相同,根据算法2,在选择第 r 个工件,应该选择 M 中不误工工件 b_k 作为第 r 个工件。若把 b_k 排在 b_r 的后面,其他位置不变得到的新排序为 R'

$$R=(b_1,b_2,\cdots,b_{k-1},b_k,b_{k+1},\cdots,b_r,b_{r+1},\cdots,b_n)$$
$$R'=(b_1,b_2,\cdots,b_{k-1},b_{k+1},\cdots,b_r,b_k,b_{r+1},\cdots,b_n)$$

比较这两个排序则有:排序 R' 中误工工件的总完工时间不增,那么带权总完工时间不增,因为 $b_k\in M$,所以排序 R' 中的不误工工件集不改变。那么,作以上形式的变换可得到一个更好的排序。

2)若 M 为空集。假设 Q 中有一个工件 b_k 满足:$p_k/w_k=\max\{p_i/w_i \mid i\in E'\}$,$b_k\in E'$。与定理2证明一样。

综上所述,再由 r 的任意性可知算法2产生的排序是问题的最优解。 证毕

算法1与算法2都是从最后往前逐个安排工件,每次安排工件只进行赋值运算,所以这两个算法都可以在 $O(n\log n)$ 时间内运行。

参考文献:

[1] Smith W E. Various optimizers for single-stage production[J]. Naval Research Logistics,1956,3:59-66.

[2] 唐国春,张峰,罗守成,等. 现代排序论[M]. 上海:上海科学普及出版社,2003.

[3] Moore J M. An n-job,one machine sequencing algorithm for minimizing the number of late jobs[J]. Management Science,1968,15:102-109.

[4] Sidney J B. An extension of Moore's due date algorithm[M]. Elmaghraby S E. Symposium on the Theory of Scheduling and its Applications. Berlin:Springer,1973:393-398.

[5] Lawler E L. Sequencing to Minimize the Weighted Number of Tardy Jobs[J]. RAIRO,1976,S10(5):27-33.

[6] Kise H,Ibarakt T,Mine H. A solvable case of the one-machine scheduling problem with ready and due times[J]. Operations Research,1978,26:121-126.

[7] 黄婉珍,唐国春. 分支定界法求解带权误工工件数排序[J]. 应用数学学报,1992,15(2):194-199.

[8] 邓俊强,林诒勋. $1\parallel\sum U_j$ 最优解的唯一性及其全部最优解的生成[J]. 郑州大学学报(自然科学版),1997,29(4):18-22.

[9] 越民义. 组合优化导论[M]. 杭州:浙江科学技术出版社,2001.

[10] Pindo M. Scheduling:theory, algorithms, and systems [M]. 2nd edition. New Jersey:Prentice Hall,2002.

[11] Brucker P. Scheduling algorithms[M]. 4th edition. Heidelberg:Springer,2004.

[12] 孙叶平,唐万梅,唐国春. Moore-Hodgson 算法最优性的新证明[J]. 重庆师范大学学报(自然科学版),2007,24(3):4-7.

[13] 孙叶平,唐国春. KIM 算法的最优性[J]. 运筹学学报,2007, 11(4):116-120.

[14] 陈小林,苏文玉,唐国春. Moore-Hodgson 算法的最优性[J]. 上海第二工业大学学报,2008,25(1):25-28.

[15] 苏永英，蒋宗彩，孙叶平. 推广的误工排序问题的最优算法[J]. 上海第二工业大学学报，2008，25(3)：201-206.

[16] 唐国春. 误工排序问题的研究[J]. 重庆师范大学学报（自然科学版），2009，26(2)：1-6.

[17] 彭洪洁，苏永英，唐国春. 部分工件必须不误工的误工排序问题[J]. 重庆师范大学学报（自然科学版），2009，26(2)：18-21.

[18] Emmons H. One-machine sequencing to minimize certain functions of job tardiness[J]. Opns Res，1969，17：701-715.

[19] Heck H，Roberts S. A note on extension of a result on scheduling with a secondary criteria[J]. Nav Res Log，1972，19：403-405.

[20] Emmons H. One machine sequencing to minimize mean flow time with minimum number tardy[J]. NRL，1975，22：585-592.

[21] Shanthikuma J G. Scheduling n jobs on one machine to minimize the maximum tardiness with minimum number tardy[J]. Comput Oper Res，1983，10：255-266.

[22] Vairaktarakis G L，Lee C Y. The single-machine scheduling problem to minimize total tardiness subject to minimum number of tardy jobs[J]. IIE Trans，1995，27：250-256.

[23] Huo Y M，Leung J Y T，Zhao H R. Complexity of two dual criteria scheduling problems[J]. Operations Resrarch Letters，2007，35：211-220.

Operations Research and Cybernetics

Two Polynomial-Time Algorithms for Dual Scheduling Problems

PENG Hong-jie[1]，*TANG Guo-chun*[1,2]

(1. College of Mathematics Science，Chongqing Normal University，Chongqing 400047；

2. Institute of Management Engineering，Shanghai Second Polytechnic University，Shanghai 200041，China)

Abstract：Scheduling problems with multiple objectives play increasing important roles in solving complicated problems appearing in the fields of economy，management，engineering，military affairs and society etc. In this paper，we give two polynomial-time algorithms when all tardy jobs are given for the two binary NP-hard problems $1 \| Lex(\sum U_j, \sum w_j C_j)$ and $1 | | Lex(\sum U_j, \sum w_j T_j)$. For the problem $1 \mid p_i \leqslant p_j \Rightarrow w_i \geqslant w_j \mid Lex(E, \sum w_j C_j)$，schedule job k last，where $p_k / w_k = \max \{p_i / w_i \mid i \in M \cup L\}$，and $M = \{i \mid d_i \geqslant \sum_{i \in J} p_i\}$ is the set of jobs which are not tardy even when processed last，L is set of tardy jobs；For the problem $1 \mid p_i \leqslant p_j \Rightarrow w_i \geqslant w_j \mid Lex(E, \sum w_j T_j)$，schedule job k last，where $p_k / w_k = \max \{p_i / w_i \mid i \in L\}$ if M is empt；else choose any job in M. In the end，we give prooves of the schedule which got from the polynomial-time algorithm is an optimal solution for the scheduling problem with weighted agreeable condition resprctively.

Key words：scheduling；tardy；algorithm；multiple objectives；complexity；optimality

（责任编辑 黄 颖）