Moore-Hodgson 算法最优性的新证明*

孙叶平[1],唐万梅[1],唐国春[2]
(1. 重庆师范大学 数学与计算机科学学院,重庆 400047;2. 上海第二工业大学 管理工程研究所,上海 200041)

摘 要:经典排序论中使误工工件的个数为最少的单台机器排序问题,简称为误工问题,是排序论中最基本的问题之一。著名的 Moore-Hodgson 算法可以在时间 $O(n\log n)$ 内得到误工问题的最优解。虽然经过改进,然而 Moore-Hodgson 算法最优性的证明仍然非常复杂。本文给出 Moore-Hodgson 算法最优性的一个非常简洁的新的证明。由于误工问题在排序论里的重要性,本文给出的新的证明在理论上是有重要意义的,是可以为排序论的专著和教材所采纳的。此外,对于推广的误工问题,例如,某些工件必须不误工的排序问题,或者工件的就绪时间不相同、但是与交货期有"一致性"关系的排序问题,或者工件的加工时间与工件的权有反向"一致性"关系的排序问题等,也可能有简洁的证明。

关键词:排序;最优性;算法
中图分类号:O223 **文献标识码**:A **文章编号**:1672-6693(2007)03-0004-04

A New Proof of the Optimality of Moore-Hodgson Algorithm

SUN Ye-ping[1], *TANG Wan-mei*[1], *TANG Guo-chun*[2]
(1. College of Mathematics and Computer Science, Chongqing Normal University, Chongqing 400047;
2. Institute of Management Engineering, Shanghai Second Polytechnic University, Shanghai 200041, China)

Abstract: The single machine scheduling problem to minimize the number of tardy jobs is one of the most basic scheduling problems in scheduling theory. The famous Moore-Hodgson algorithm can get the problem's optimal solution in $O(n\log n)$. Though there is improvment, the proof of the optimality of Moore-Hodgson algorithm is still very complicated. Owing to the importance of the problem, the simplified proof in our paper is meaningful in terms of theory and should be adopted in scheduling books. In addition, for the generalized scheduling problems to minimize the number of tardy jobs, such as: the case in which some jobs must be on-time, or the case of agreeability of ready times with due dates, or the case of reverse agreeability of processing times with weights, etc. their optimalities could be simplified to prove, too.

Key words: scheduling; optimality; algorithm

经典排序论中使误工工件的个数为最少的单台机器排序问题,简称为误工问题[1],有很多学者对此进行了研究[1-10],它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[2] 提出解决这个误工排序问题的算法。这个算法经过 Hodgson 的修改,称为 Moore-Hodgson 算法,可以在时间 $O(n\log n)$ 内得到误工问题的最优解。Moore-Hodgson 算法最优性的证明非常复杂。以后,又有许多别的证明[3-7],可是仍然比较繁琐。本文给出 Moore-Hodgson 算法最优性的比较简洁的证明。此外,对于推广的误工问题,例如,某些工件必须不误工的排序问题[8],或者工件的就绪时间不相同、但是与交货期有"一致性"关系的排序问题[9],或者工件的加工时间与工件的权有反向"一致性"关系的排序问题[10]等,也可能有简洁的证明。

用排序论的语言,误工问题可以这样表述。设有 n 个工件 $1,2,\cdots,n$ 要在一台机器上加工。记这 n 个工件的集合为 $J=\{1,2,\cdots,n\}$。已知工件 j 的

* 收稿日期:2007-01-16
资助项目:国家自然科学基金项目(No. 10371071;No. 70618001)
作者简介:孙叶平(1980-),女,河南商城人,硕士研究生,研究方向为组合最优化。

加工时间是 p_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,而且加工不允许中断,只有当一个工件加工完成后才能加工其他工件。如果工件在交货期之后才完工,称为这个工件的加工是误工的,否则称为是不误工的。安排一个加工次序,使得工件加工不误工,这是实际生产最基本的要求之一。只有在所有加工的工件都不误工的前提下,才可能考虑其他目标,例如使总的加工时间为最短等。如果不可能使所有的工件都不误工,就要考虑缩短加工时间,或者推迟交货期,这是可控排序研究的问题[1]。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[1],用三参数表示为 $1 \parallel \sum U_j$。不失一般性,假设工件已经按照交货期的大小重新编号,有 $d_1 \le d_2 \le \cdots \le d_n$,并把这个按照交货期从小到大排列的序(EDD 序)记为 E_0。

1 初步分析

安排一个加工次序,使得工件的加工不误工,这是实际生产最基本的要求之一。那么,如何来判别一个问题是否存在使得工件都不误工的加工次序呢?下面的引理1给出了回答。

引理1 在误工问题中存在一个排序没有误工工件的充分必要条件是这个问题的 EDD 序 E_0 是没有误工工件的。

证明 必要性。用反证法来证明。假如在误工问题中存在一个排序 S 没有误工工件,而 EDD 序中有误工工件,那么 EDD 序的最大延误 $T_{\max}(EDD) > 0$。已知 EDD 序是使 $T_{\max}$ 为最小的排序[1],则对于任何排序的 $T_{\max} > 0$,当然对于排序 S 也有 $T_{\max}(S) > 0$。这与排序 S 没有误工工件相矛盾。故必要性成立。

充分性。是不证自明的。因为如果误工问题的 EDD 序没有误工工件,则该误工问题中当然存在排序(即 EDD 序)没有误工工件。 证毕

使得误工的工件的个数为最少的加工次序,称为是误工问题的最优排序,或者称为最优解。误工排序问题的最优解,有非常明确的结构。

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

证明 对于误工问题的一个最优解,如果有工件是误工的,那么可以把该工件往后排,当然仍然是误工的;而其余工件向前面移,这样做不会产生新的误工工件。因而,这样移动后仍然是最优解。排在后面的误工工件无论怎么排,仍然都是误工的,因为假如有可能不误工,原来的排法不可能是最优解。因此,误工问题存在一个最优解可以分为前后两部分:前面的都是不误工的工件;后面的都是误工的工件,可以以任意次序排列。由上面的引理1可知,前面没有误工工件的那部分的 EDD 序一定也都不误工。所以存在最优解,其前面不误工的工件是按照 EDD 序排列的。 证毕

引理3 如果工件集合 $\{j_1, j_2, \cdots, j_t\}$ $(1 \le t \le n)$ 中的工件是按 EDD 序排列,而且此时最后一个工件 j_t 是误工的,那么对于工件 $1, 2, \cdots, n$ 的任何排序,在工件 $j_1, j_2, \cdots, j_t$ 中至少有一个工件是误工的。

证明 用反证法来证。假设在工件 $1, 2, \cdots, n$ 的一个排序中 $j_1, j_2, \cdots, j_t$ 都不误工,那么把这些工件向前移动,排在最前面,仍然是都不误工的。由引理1知,这些工件的 EDD 序也都不误工。这与引理3的已知条件"工件 j_t 是误工的"相矛盾。 证毕

引理4 如果工件 k 是误工问题 EDD 序 E_0 的第一个误工工件。记工件 k 之前的所有工件(包括工件 k)中加工时间最长的工件为 r。如果把工件 r 作为误工工件排在最后,得到的排序记为 S,那么在工件集合 $\{1, 2, \cdots, k\}$ 中只有工件 r 是在排序 S 中误工的工件。

证明 如果工件 k 就是前 k 个工件中加工时间最长的工件,即 $r = k$,那么排序 S 是把工件 k 排在最后。由于工件 k 在排序 S 中是向后移动,当然还是误工的,而 k 之前的其他工件均不误工(因为位置没有变化)。因此,引理4成立。

如果工件 k 不是前 k 个工件中加工时间最长的工件 r,即 $r < k$。在排序 S 中,工件 k 的完工时间提前了 p_r。由于 k 是第一个误工工件,且 $p_r \ge p_k$,$d_k \ge d_{k-1}$,所以有

$$C_k(S) - d_k = C_k(EDD) - p_r - d_k =$$
$$C_{k-1}(EDD) + p_k - p_r - d_k \le d_{k-1} + p_k - p_r - d_k \le 0$$

即工件 k 在排序 S 中是不误工的工件。而 k 之前的工件(与 EDD 序相比)或者没有变化,或者向前面移,因此也不会误工。由引理3知,在工件 $1, 2, \cdots, k$ 中至少有一个工件在 S 中是误工的,所以工件 r 在排序 S 中一定是误工工件。从而此时在工件 $1, 2, \cdots, k$ 中只有工件 r 是在排序 S 中误工。 证毕

由此,很"自然"会提出如下 Moore-Hodgson 算法。

步骤 1　把所有的工件按 EDD 序放进集合 E,令 L 为空集;

步骤 2　如果在 E 中没有误工工件,那么算法终止,把 E 中的工件放在 L 中的工件前面得到的 (E,L),就是误工问题的最优解;否则,记 E 中第 1 个误工工件为 k;

步骤 3　在 E 中确定排在工件 k 之前的所有工件(包括工件 k)中加工时间最长的工件,记为 r,并把 r 从 E 中取出,按任意次序放进 L,转步骤 2。

Moore-Hodgson 算法的思想是基于引理 3,因为在 E 中工件 k 之前(包括工件 k)的所有工件中无论怎么样的排法,至少有一个工件是误工的。因此,选择这些工件中加工时间最长的工件作为误工的工件,排到最后,可以使得后面的工件向前移最多。在下一节将给出严格的证明。为了叙述方便,把 Moore-Hodgson 算法第 i 次迭代在执行步骤 2 中的排序记为 (E_{i-1},L_{i-1}),其中 E_{i-1} 中的工件是按照 EDD 序排列的,$L_{i-1}=\{r_1,r_2,\cdots,r_{i-1}\}$。$E_0$ 就是问题的 EDD 序,$L_0=\varnothing$(空集)。记 k_i 是步骤 3 中集合 E_{i-1} 中的第一个误工工件。步骤 3 是把工件集合 $\{1,2,\cdots,k_i\}\setminus L_{i-1}$(包括工件 k_i)中加工时间最长的工件 r_i 放进集合 L_{i-1},即 $E_i=E_{i-1}\setminus\{r_i\}$,$L_i=L_{i-1}\cup\{r_i\}$。

2 Moore-Hodgson 算法最优性的证明

证明　现证 Moore-Hodgson 算法(以下简称算法)产生的排序是误工排序问题 $1\parallel\sum U_j$ 的最优解。

假设误工问题最优解有 u 个误工工件(不妨设 $u\geq1$。因为当 $u=0$ 时,误工问题的最优排序中没有误工工件,由引理 1 知 EDD 序中也没有误工工件。此时由算法得到的排序为 EDD 序,因此算法得到的排序是最优解)。这时施行算法,至少要有 u 次迭代。如果算法不需要 u 次迭代就终止,那表明算法得到的排序中误工的工件个数少于 u 个,这与假设"误工问题最优解有 u 个误工工件"相矛盾。沿用前面的记号,算法第 u 次迭代执行步骤 3 所得到排序是 (E_u,L_u),其中 E_u 中的工件是 EDD 排序,$L_u=\{r_1,r_2,\cdots,r_u\}$。根据引理 2,不妨假设误工问题的最优解是 $S^*=(E^*,L^*)$,其中 E^* 中的工件按照 EDD 序排列,L^* 为 u 个误工工件的集合,以任意次序排在 E^* 后面。

对于算法的第一次迭代,工件 k_1 是 E_0 中第一个误工工件,算法是把工件集合 $\{1,2,\cdots,k_1\}$ 中加工时间最长的工件 r_1 放进集合 L_0,得到排序 (E_1,L_1)。由引理 4 知,集合 $\{1,2,\cdots,k_1\}\setminus\{r_1\}$ 中的工件在排序 (E_1,L_1) 中不会再有误工工件。如果 $r_1\in L^*$,记 $q_1=r_1$,那么 $q_1\in\{1,2,\cdots,k_1\}$。如果 $r_1\notin L^*$,那么由引理 3 知,在 $\{1,2,\cdots,k_1\}$ 中至少有一个工件在 S^* 中是误工的,也就是至少有一个工件是在 L^* 中,记其中编号最小的工件为 q_1。从而,对这两种情况,在 L^* 中都有工件 $q_1\in\{1,2,\cdots,k_1\}$,自然,$q_1\leq k_1$。因为工件 r_1 是工件集 $\{1,2,\cdots,k_1\}$ 中加工时间最长的工件,所以 $r_1\leq k_1$,并且 $p_{r_1}\geq p_{q_1}$,而且,此时有 $r_1\notin L^*\setminus\{q_1\}$。对于算法的第二次迭代,工件 k_2 是 E_1 中第一个误工工件,即工件 k_2 是 $\{1,2,\cdots,k_2\}\setminus\{r_1\}$ 中工件按照 EDD 排序的最后一个工件,而且是误工的,算法是把工件集合 $\{1,2,\cdots,k_2\}\setminus\{r_1\}$ 中加工时间最长的工件 r_2 $(r_2\neq r_1)$ 放进集合 L_1,得到排序 (E_2,L_2)。由引理 4 知,集合 $\{1,2,\cdots,k_2\}\setminus\{r_1,r_2\}$ 中的工件在排序 (E_1,L_1) 中不会再有误工工件。如果 $r_2\in L^*$,记 $q_2=r_2\neq r_1$,那么 $q_2\in\{1,2\cdots,k_2\}\setminus\{r_1\}$。如果 $r_2\notin L^*$,那么由引理 3 知,在 $\{1,2,\cdots,k_2\}\setminus\{r_1\}$ 中一定至少存在一个工件是在 L^* 中,记其中编号最小的一个工件为 $q_2\in\{1,2,\cdots,k_2\}\setminus\{r_1\}$。从而,对这两种情况,在 L^* 中都有工件 $q_2\in\{1,2,\cdots,k_2\}\setminus\{r_1\}$,自然,$q_2\leq k_2$。因为工件 r_2 是工件集 $\{1,2,\cdots,k_2\}\setminus\{r_1\}$ 中加工时间最长的工件,所以 $r_2\leq k_2$,并且 $p_{r_2}\geq p_{q_2}$,而且,此时 $r_1,r_2\notin L^*\setminus\{q_1,q_2\}$。如此这样继续下去,直到算法第 u 次迭代,工件 k_u 是 E_{u-1} 中第一个误工工件,即工件 k_u 是 $\{1,2,\cdots,k_u\}\setminus\{r_1,r_2,\cdots,r_{u-1}\}$ 中工件按照 EDD 排序的最后一个工件,而且是误工的。按照前面 $q_1,q_2,\cdots$ 的记号,此时 $r_1,r_2,\cdots,r_{u-1}\notin L^*\setminus\{q_1,q_2,\cdots,q_{u-1}\}$。算法是把工件集合 $\{1,2,\cdots,k_u\}\setminus\{r_1,r_2,\cdots,r_{u-1}\}$ 中加工时间最长的工件 r_u 放进集合 L_{u-1},得到排序 (E_u,L_u)。由引理 4 知,集合 $\{1,2,\cdots,k_u\}\setminus\{r_1,r_2,\cdots,r_u\}$ 中的工件在排序 (E_u,L_u) 中不会再有误工工件。那么由引理 3 知,在 $\{1,2,\cdots,k_u\}\setminus\{r_1,r_2,\cdots,r_{u-1}\}$ 中一定存在一个工件是在 L^* 中,记 L^* 中剩下一个工件为 q_u,显然 $q_u\in\{1,2,\cdots,k_u\}\setminus\{r_1,r_2,\cdots,r_{u-1}\}$,自然 $q_u\leq k_u$。因为工件 r_u 是工件集 $\{1,2,\cdots,k_u\}\setminus\{r_1,r_2,\cdots,r_{u-1}\}$ 中加工时间最长的工件,所以 $r_u\leq k_u$,并且 $p_{r_u}\geq p_{q_u}$。因此,对算法的每一次迭代都有 $p_{r_i}\geq p_{q_i}$ $(1\leq i\leq u)$ 成立。

由于 $q_i \le k_i$,$r_i \le k_i$,$k_i \le k_u$($1 \le i \le u$),所以 $q_i \le k_u$,$r_i \le k_u$($1 \le i \le u$),因而工件集合 $J \setminus \{r_1, r_2, \cdots, r_u\} = (\{1, 2, \cdots, k_u\} \setminus \{r_1, r_2, \cdots, r_u\}) \cup \{k_u + 1, \cdots, n\}$,工件集合 $J \setminus \{q_1, q_2, \cdots, q_u\} = (\{1, 2, \cdots, k_u\} \setminus \{q_1, q_2, \cdots, q_u\}) \cup \{k_u + 1, \cdots, n\}$。由此可知,这两个工件集合的 EDD 序 E_u 和 E^* 中都有工件 $k_u + 1$,并且在工件 $k_u + 1$ 之后的工件也都相同。此外,由于集合 $\{1, 2, \cdots, k_u\} \setminus \{r_1, r_2, \cdots, r_u\}$ 中的工件在排序(E_u,L_u)中不会再有误工工件,所以,在 E_u 中位于工件 $k_u + 1$ 之前的工件都不会误工。

对于 E_u 中的工件 $k_u + 1$ 及其之后的工件在排序(E_u,L_u)中的完工时间比他们在最优排序 $S^* = (E^*, L^*)$ 中的完工时间提前 $\sum_{i \in L_u} p_i - \sum_{i \in L^*} p_i \ge 0$。由于 E^* 中的工件都不误工,所以 E_u 中工件 $k_u + 1$ 及其之后的工件均不误工。从而 E_u 中没有误工工件。

由于 E_u 中没有误工工件,所以对 L_u 中的 u 个工件无论怎么排,这 u 个工件一定都是误工的。否则,与最优解有 u 个误工工件的假定相矛盾。

由此可知,算法经过 u 次迭代得到的排序(E_u,L_u)中 E_u 没有误工工件,只有 L_u 中的 u 个工件是误工的,所以(E_u,L_u)是误工问题的最优排序。这也就证明了 Moore-Hodgson 算法的最优性。 证毕

误工问题 $1 \parallel \sum U_j$ 的最优解一般来说不是唯一的,因为存在不是由 Moore-Hodgson 算法得到的最优解,举例如下。

例 对于下列 6 个工件,求使得 $\sum Uj$ 为最小的排法。

j	1	2	3	4	5	6
p_j	2	3	10	11	7	5
d_j	4	5	14	19	22	29

根据 Moore-Hodgson 算法得到的误工工件数为 $\sum U_j = 2$,且 $E = \{1, 2, 5, 6\}$,$L = \{3, 4\}$。如果记 $E' = \{1, 2, 4, 6\}$,$L' = \{3, 5\}$,那么 E' 中的工件都不误工,L' 中的工件都误工,所得的排序(E',L')也是最优排序。

3 结论

本文重新给出 Moore-Hodgson 算法最优性的比较简洁的证明。由于误工问题在经典排序论的重要性,本文给出的新的证明在理论上是有重要意义的,是可以为排序论的专著和教材所采纳的。此外,对于推广的误工问题,例如,某些工件必须不误工的排序问题,也可能有简洁的证明[8]。这是作者接下来要做的工作。

参考文献:

[1] 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.

[2] MOORE J M. An n-job, One Machine Sequencing Algorithm for Minimizing the Number of Late Jobs[J]. Management Science, 1968, 15:102-109.

[3] 唐国春. 带权误工工件数排序问题[J]. 上海第二工业大学学报,1990,1(1):10-15.

[4] 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.

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

[6] BRUCKER P. Scheduling Algorithms[M]. 4th edition, Heidelberg:Springer, 2004.

[7] PINEDO M. Scheduling:Theory, Algorithms, and Systems[M]. 2nd edition, New Jersey:Prentice Hall, 2002.

[8] SIDNEY J B. An extension of Moore's Due Date Algorithm[A]. Symposium on the Theory of Scheduling and its Applications[C]. Berlin:Springer, 1973. 393-398.

[9] KISE H, IBARAKI T, MINE H. A Solvable Case of the One-machine Scheduling Problem with Ready and Due Times[J]. Operations Research, 1978, 26:121-126.

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

(责任编辑 黄 颖)