第49卷 第6期 中山大学学报（自然科学版） Vol. 49 No. 6
2010年 11月 ACTA SCIENTIARUM NATURALIUM UNIVERSITATIS SUNYATSENI Nov. 2010

速度相同的具有 $m-2$ 台通用机的两组工件的 LS 算法分析*

丁 伟
（中山大学数学与计算科学学院，广东 广州 510275）

摘 要：对于实践中存在的具有两组任务的优化排序问题进行了讨论，在经典的LS算法的基础上提出了一种改进的LS算法，利用“首先空闲”准则选择机器，按照工件的到达顺序安排工件，讨论了将两组工件安排在两台速度相同的专用机，$m-2$ 台同速度的通用机上的 C_{max} 问题，其中工件具有准备或到达时间，且工件的准备或到达时间均不超过其加工时间的 α 倍。目标是在最短的时间内完成所有给定的任务。得到了利用该近似算法所得的解 T^{LS} 与最优解 T^* 的一个估计 $(1+\alpha)(2-1/m)$，并且证明了对任意的 α 此界是紧的。

关键词：启发式算法；性能指标；LS算法；LPT算法；通用机与专用机
中图分类号：O223 **文献标志码：**A **文章编号：**0529-6579（2010）06-0001-05

An Analysis of LS Algorithm for the Scheduling Problem on $m-2$ General-Purpose Machinery and Two Group Tasks with Uniform Processors

DING Wei
(School of Mathematics and Computational Science,
Sun Yat-sen University, Guangzhou 510275, China)

Abstract: The C_{max} problem on two-group jobs with $m-2$ general-purpose machineries and two special-purpose machineries that they are the same speed, and the jobs have the setup times and it is no more α time than its processing times is studied. An improved LS algorithm and the upper tight bound performance are given. The ratio of the approximate solution T^{LS} and the best way T^* is $(1+\alpha)(2-1/m)$, and it is tight for all α.

Key words: heuristic approach; performance indexes; LS algorithm; LPT algorithm; general-purpose and special-purpose machinery

对于将 n 个工件 $J=\{J_1,J_2,\cdots,J_n\}$ 安排在 m 台平行机 $M=\{M_1,M_2,\cdots,M_m\}$ 上加工，并且工件 J_j 紧跟在 J_i 后加工，需要 $w(i,j)$ 的准备时间或者调整时间，目标是使得所有工件尽早完工的调度问题，许多学者进行了研究，该问题产生于生产实际，且被证明即使是 $m=1$ 的情形也已经是一个NP-Hard问题了，因而研究其启发式算法及其有效性是十分有意义的。这类问题最初由Graham进行了研究，他在文［1］中对 $w(i,j)=0$ 的情形运用LS（List Scheduling）算法展开了讨论，得到该算法在最差情形下的性能指标为 $2-1/m$；而运用LPT（Longest Processing Time）算法，他得到一个紧界：$4/3-1/(3m)$。之后Ovacik等[2]研究了 $w(i,j)\leqslant t_j$（t_j 表示工件 J_j 的加工时间）的情形，他们证明了在LS算法下，该问题的最差性能指标为 $4-2/m$。

针对实际应用中存在的一类关于多组工件在多组加工速度相同或不同的机器上的 C_{max} 问题，许多

* 收稿日期：2009-12-07
基金项目：国家自然科学基金资助项目（10971234，10671213）
作者简介：丁伟（1969年生），女，讲师；E-mail：dingwei@mail.sysu.edu.cn

学者也进行了广泛的研究，秦成林等[3]针对两组工件，两台专用机和 m 台通用机（专用机与通用机的速度相同）的情形进行了讨论，并得到了严格的界 $T/T^* \leqslant 3/2$；秦成林等[4]针对两组工件，两组专用机和一组通用机（专用机与通用机的速度相同）的情形进行了讨论，并得到了界 $T/T^* \leqslant 2-1/m|M|$；秦成林等[5]针对两组专用机和一组通用机（专用机与通用机的速度相同）的情形进行了讨论，提出了一种随机的改进算法；秦成林等[6]针对一组工件，m 台速度不同的机器，且机器加工新工件有准备时间的情形进行了讨论，利用“首先空闲”准则，给出了一种重新分配的改进型算法；秦成林等[7]针对两组工件，两台速度不同专用机和两台速度相同通用机（专用机的速度不小于通用机的速度）的情形进行了讨论，并得到了严格的界 $T/T^* \leqslant 3/2$；丁伟[8]针对两组工件，两台速度不同的专用机，m 台速度相同的通用机，且专用机的速度不小于通用机的速度的情形展开了讨论，并得到了在一种改进 LPT 算法下有 $T/T^* \leqslant 2$ 的严格界；此作者还在文［9］中对三组工件，三台专用机和一台通用机（专用机与通用机的速度相同）的情形进行了讨论，并得到了一种改进 LPT 算法的界 $T/T^* \leqslant 4/3$。在文［10］中该作者还就 n 组工件，n 台专用机和一台通用机（专用机与通用机的速度相同）的情形进行了讨论，并得到了一种改进 LPT 算法的界 $T/T^* \leqslant (n+1)/n$。在文［11］中该作者还就 n 组工件，n 台专用机和 m 台通用机（专用机与通用机的速度相同）的情形进行了讨论，并得到了一种改进 LPT 算法的界

$$\frac{T^{\mathrm{LPT}}}{T^*} \leqslant \begin{cases} \dfrac{2m+1}{m+1}, & \text{当 } m \geqslant n-1 \text{ 时} \\ \dfrac{m+n}{m+1}, & \text{当 } m < n-1 \text{ 时} \end{cases}$$

Gairing 等[12]也考虑了 n 个工件，m 台不同类处理机的情形，提出了一种基于最小费用流算法的组合算法，证明了该算法更简单，复杂度更低，从而更有效；丁伟[13]还就 n 组工件，n 台专用机和一台通用机（专用机与通用机的速度不同）的情形进行了讨论，并得到了一种改进 LPT 算法的界 $T/T^* \leqslant 1+1/\sum\limits_{i \in I} s_i$。

本文讨论如下的调度问题：将两组工件 L_1，L_2 安排在 m 台机器 $M = \{M_1, M_2, \cdots, M_m\}$ 上加工，其中工件的加工时间分别为 $t(r,i), r=1,2, i=1,2,\cdots,n_r$，机器 M_1 为工件组 L_1 的专用机器，机器 M_2 为工件组 L_2 的专用机器，M_3，M_4，…，M_m 为通用机器，可以加工任何一组工件，所有机器的加工速度相同，我们不失一般性地假设其均为 1，并且工件 $t(h, j)$ 紧跟在 $t(l, i)$ 后加工，需要 $w(l, i; h, j)$ 的准备时间或者调整时间，若在某个机器上第一个加工，$w(h, 0; h, j)$ 则表示调整时间，且对所有 l, h, i, j，均有 $w(l, i; h, j) \leqslant \alpha t(h, j)$。目标是使得所有工件尽早完工。

本文在经典的 LS 算法的基础上提出了一种针对两组工件的改进 LS 算法，得到了问题在改进的 LS 算法下的界

$$\frac{T^{\mathrm{LS}}}{T^*} \leqslant (1+\alpha)\left(2-\frac{1}{m}\right)$$

并且对任意的 α 此界是紧的。

1 两组工件的改进 LS 算法的步骤

两组工件的改进 LS 算法在机器的选择上仍然利用“首先空闲”准则，即当某个机器一出现空闲就安排工件加工，如果有几台机器同时完工，则按照原先机器的排列顺序选择第一台完工的机器安排工件，此时如果这个机器是某组工件的专用机器，则安排该组工件中当前的第一个工件在其上加工，如果该机器是通用机器，则安排两组工件中第二下标较小的工件加工，如果两组工件的安排进度一样，则按照原先工件组的自然顺序，安排排在前面的那个工件组的当前工件进行加工。这就好像银行的自动叫号系统，总是优先服务拿到最小号码的顾客。而对于同一组工件中的工件，按照任意次序排列，不做任何特殊要求。

下面给出两组工件一些记号：

1）若工件 $t(l,i)$ 分配给机器 M_k，则记作 $t(l, i) \in M_k, k=1,2,\cdots,m$；

2）$ML_k(k=1,2,\cdots,m)$ 表示各机器上的工件集合；

3）$MT_k(k=1,2,\cdots,m)$ 表示各机器上的最后完工时间；

$$MT_k = \sum_{t(h,j) \in M_k} w(*,*;h,j) + t(h,j), \quad k=1,2,\cdots,m$$

4）T^{LS} 表示在改进 LS 算法下 m 台机器的最后完工时间；

5）T^* 表示最优排序中 m 台机器的最后完工时间。

下面给出两组工件的改进 LS 算法的步骤。

step 1：Let $i=1; j=1; Q=\{1,2,3,\cdots,m\}; ML_k=\Phi; MT_k=0, k \in Q$；

step 2：If $i>n_1$ and $j>n_2$ then goto 5；

else

if $i>n_1$ then $Q=Q-\{1\}$；

if $i>n_2$ then $Q=Q-\{2\}$；

let $p=\min_{k'}\{k' \mid MT_{k'}=\min_{k\in Q}MT_k\}$；

step 3：If $p=1$ then let $r=1,q=i,i=i+1$；

If $p=2$ then let $r=2,q=j,j=j+1$；

If $p\geqslant 3$ then

if $i\leqslant n_1$ then

if $j\leqslant n_2$ then

if $i\leqslant j$ then let $r=1,q=i,i=i+1$；

else let $r=2,q=j,j=j+1$；

else let $r=1,q=i,i=i+1$；

else

if $j\leqslant n_2$ then let $r=2,q=j,j=j+1$；

else goto step 5；

step 4：Let

$ML_p=ML_p\cup\{t(r,q)\},MT_p=MT_p+w(*,*;r,q)+t(r,q)$；goto step 2；

step 5：Let　$T^{LS}=\max_{1\leqslant k\leqslant m}\{MT_k\}$；

输出各个机器上的工件安排 $ML_k,k=1,2,\cdots,m$ 及最后完工时间 T^{LS}。

2　两组工件的改进 LS 算法分析

定理 1　本文所讨论的问题在上述改进的 LS 算法下的界为

$$\frac{T^{LS}}{T^*}\leqslant(1+\alpha)\left(2-\frac{1}{m}\right)\tag{1}$$

并且对任意的 α 此界是紧的。

证明　根据改进的 LS 算法，假设机器 $M_k(1\leqslant k\leqslant m)$ 最后完工，最后完工的工件为 $t(r,j)(r\in\{1,2\},1\leqslant j\leqslant n_r)$，则在机器 M_k 上有

$$T^{LS}=MT_k\tag{2}$$

而在其它机器上有

$$MT_i\geqslant MT_k-(w(*,*;r,j)+t(r,j))=T^{LS}-(w(*,*;r,j)+t(r,j)),\quad i=1,2,\cdots m,i\neq k\tag{3}$$

因此

$$\sum_{i=1}^{m}MT_i=MT_k+\sum_{\substack{i=1\\i\neq k}}^{m}MT_i\geqslant mT^{LS}-(m-1)(w(*,*;r,j)+t(r,j))\tag{4}$$

另一方面对最优排序 T^* 有

$$T^*\geqslant t(r,j)\tag{5}$$

$$T^*\geqslant\frac{\sum_{l=1}^{2}\sum_{i=1}^{n_l}t(l,i)}{m}\tag{6}$$

由假设 $w(*,*;r,j)\leqslant\alpha t(r,j)$ 及（5）式可得

$$w(*,*;r,j)+t(r,j)\leqslant(1+\alpha)t(r,j)\leqslant(1+\alpha)T^*\tag{7}$$

由（4）式和（7）式可得

$$\sum_{i=1}^{m}MT_i\geqslant mT^{LS}-(1+\alpha)(m-1)T^*\tag{8}$$

另一方面由假设和（6）式可得

$$\sum_{i=1}^{m}MT_i=\sum_{i=1}^{m}\sum_{\{t(h,p)\}\in ML_i}(w(*,*;h,p)+t(h,p))=\sum_{h=1}^{2}\sum_{p=1}^{n_h}(w(*,*;h,p)+t(h,p))\leqslant(1+\alpha)\sum_{h=1}^{2}\sum_{p=1}^{n_h}t(h,p)\leqslant(1+\alpha)mT^*\tag{9}$$

由（8）式和（9）式可得

$$(1+\alpha)mT^*\geqslant\sum_{i=1}^{m}MT_i\geqslant mT^{LS}-(1+\alpha)(m-1)T^*$$

即

$$(1+\alpha)(2m-1)T^*\geqslant mT^{LS}$$

所以

$$\frac{T^{LS}}{T^*}\leqslant\frac{(1+\alpha)(2m-1)}{m}=(1+\alpha)\left(2-\frac{1}{m}\right)$$

下面的例子可以说明此界对任意 α 的均可达到。

$$L_1=\left\{t(1,1),t(1,2),\cdots,t\left(1,\frac{m(m-1)}{2}\right),t\left(1,\frac{m(m-1)}{2}+1\right)\right\}$$

$$L_2=\left\{t(2,1),t(2,2),\cdots,t\left(2,\frac{m(m-1)}{2}\right)\right\}$$

其中，$t(r,i)=1$，$r=1,2,i=1,2,\cdots,\frac{m(m-1)}{2}$，$t\left(1,\frac{m(m-1)}{2}+1\right)=m$，并令工件的准备时间如下：

1）当 m 为偶数时的准备时间。

当 $r=1,2,i=0;j=1,2,\cdots,m/2$ 时，$w(r,i;r,j)=\alpha$；

当 $r=1,2,i=1,2,\cdots,m(m-1)/2-m/2$，$j=i+m/2$ 时，$w(r,i;r,j)=\alpha$；

当 $r=1$，$i=m(m-1)/2-m/2+1$，$j=m(m-1)/2+1$ 时，$w(r,i;r,j)=\alpha m$；

其他情况：$w(*,*;*,*)=0$；

2）当 m 为奇数时的准备时间。

当 $r=1$，$i=0$；$j=1,2,\cdots,(m-1)/2+1$ 时，$w(r,i;r,j)=\alpha$；

当 $r=2$，$i=0$；$j=1,2,\cdots,(m-1)/2$ 时，$w(r,i;r,j)=\alpha$；

当 $r\neq k$，$r,k=1,2$，$i,j=2,\cdots,m(m-1)/2$ 时，$w(r,i;r,j)=\alpha$；

当 $r=1$，$i=m(m-1)/2-(m-1)/2+1$，$j=m(m-1)/2+1$ 时，$w(r,i;r,j)=\alpha m$；

其他情况：$w(*,*;*,*)=0$。

当 m 为偶数时，上述实例的改进的 LS 算法排序和最优排序分别如表1和表2。

表1 LS 排序

Table 1 LS Order

机器	工件安排1	工件安排2	…	工件安排 $m-1$	工件安排 m
M_1	$t(1,1)$	$t(1,m/2+1)$	…	$t(1,m(m-1)/2-m/2+1)$	$t(1,m(m-1)/2+1)$
M_2	$t(2,1)$	$t(2,m/2+1)$	…	$t(2,m(m-1)/2-m/2+1)$	
M_3	$t(1,2)$	$t(1,m/2+2)$	…	$t(1,m(m-1)/2-m/2+2)$	
M_4	$t(2,2)$	$t(2,m/2+2)$	…	$t(2,m(m-1)/2-m/2+2)$	
M_5	$t(1,3)$	$t(1,m/2+3)$	…	$t(1,m(m-1)/2-m/2+3)$	
M_6	$t(2,3)$	$t(2,m/2+3)$	…	$t(2,m(m-1)/2-m/2+3)$	
…	…	…	…	…	…
M_{m-1}	$t(1,m/2)$	$t(1,m/2+m/2)$	…	$t(1,m(m-1)/2)$	
M_m	$t(2,m/2)$	$t(2,m/2+m/2)$	…	$t(2,m(m-1)/2)$	

表2 最优排序

Table 2 The optimal order

机器	安排1	安排2	…	$m-1$	安排 m
M_1	$t(1,m)$	$t(1,1)$	$t(1,2)$	…	$t(1,m-1)$
M_3	$t(1,m+1)$	$t(1,m+2)$	$t(1,m+3)$	…	$t(1,2m)$
M_4	$t(1,2m+1)$	$t(1,2m+2)$	$t(1,2m+3)$	…	$t(1,3m)$
…	…	…	…	…	…
$M_{m/2}$	$t(1,m(m-4)/2+1)$	…	…	…	$t(1,m(m-2)/2)$
M_2	$t(2,m)$	$t(2,1)$	$t(2,2)$	…	$t(2,m-1)$
$M_{m/2+1}$	$t(2,m+1)$	$t(2,m+2)$	$t(2,m+3)$	…	$t(2,2m)$
$M_{m/2+2}$	$t(2,2m+1)$	$t(2,2m+2)$	$t(2,2m+3)$	…	$t(2,3m)$
…	…	…	…	…	…
M_{m-2}	$t(2,m(m-4)/2+1)$	…	…	…	$t(2,m(m-2)/2)$
M_{m-1}	$t(1,m(m-2)/2+1)$	$t(1,m(m-1)/2)$	$t(2,m(m-2)/2+1)$	…	$t(2,m(m-1)/2)$
M_m		$t(1,m(m-1)/2+1)$			

当 m 为奇数时，上述实例的改进的 LS 算法排序和最优排序分别如表3和表4。

表3 LS 排序

Table 3 LS order

机器	工件安排1	工件安排2	工件安排3	工件安排4	…
M_1	$t(1,1)$	$t(1,(m-1)/2+2)$	$t(1,m+1)$	$t(1,(m-1)/2+m+2)$	…
M_2	$t(2,1)$	$t(2,(m-1)/2+1)$	$t(2,m+1)$	$t(2,(m-1)/2+m+1)$	…
M_3	$t(1,2)$	$t(2,(m-1)/2+2)$	$t(1,m+2)$	$t(2,(m-1)/2+m+2)$	…
M_4	$t(2,2)$	$t(1,(m-1)/2+3)$	$t(2,m+2)$	$t(2,(m-1)/2+m+3)$	…
…	…	…	…	…	…
M_{m-2}	$t(1,(m-1)/2)$	$t(2,m-1)$	$t(1,m+(m-1)/2)$	$t(2,2m-1)$	…
M_{m-1}	$t(2,(m-1)/2)$	$t(1,m)$	$t(2,m+(m-1)/2)$	$t(1,2m)$	…
M_m	$t(1,(m-1)/2+1)$	$t(2,m)$	$t(1,m+(m-1)/2+1)$	$t(2,2m)$	…

（上接表 3）

机器	…	工件安排 $m-2$	工件安排 $m-1$	工件安排 m
M_1	…	$t(1,\ m(m-1)/2-m+1)$	$t(1,\ m(m-1)/2-(m-1)/2+1)$	$t(1,\ m(m-1)/2+1)$
M_2	…	$t(2,\ m(m-1)/2-m+1)$	$t(2,\ m(m-1)/2-(m-1)/2)$	
M_3	…	$t(1,\ m(m-1)/2-m+2)$	$t(2,\ m(m-1)/2-(m-1)/2+1)$	
M_4	…	$t(2,\ m(m-1)/2-m+2)$	$t(1,\ m(m-1)/2-(m-1)/2+2)$	
…	…	…	…	…
M_{m-2}	…	$t(1,\ m(m-1)/2-(m-1)/2-1)$	$t(2,\ m(m-1/2)-1)$	
M_{m-1}	…	$t(2,\ m(m-1)/2-(m-1)/2-1)$	$t(1,\ m(m-1)/2)$	
M_m	…	$t(1,\ m(m-1)/2-(m-1)/2)$	$t(2,\ m(m-1)/2)$	

表 4　最优排序

Table 4　The optimal order

机器	安排 1	安排 2	…	安排 $m-1$	安排 m
M_1	$t(1,m)$	$t(1,\ 1)$	$t(1,\ 2)$	…	$t(1,\ m-1)$
M_3	$t(1,m+1)$	$t(1,m+2)$	$t(1,\ m+3)$	…	$t(1,2m)$
M_4	$t(1,2m+1)$	$t(1,\ 2m+2)$	$t(1,2m+3)$	…	$t(1,3m)$
…	…	…	…	…	…
$M_{(m+1)/2}$	$t(1,\ m(m-3)/2+1)$	…	…	…	$t(1,\ m(m-1)/2)$
M_2	$t(2,m)$	$t(2,\ 1)$	$t(2,\ 2)$	…	$t(2,\ m-1)$
$M_{(m+1)/2+1}$	$t(2,m+1)$	$t(2,\ m+2)$	$t(2,\ m+3)$	…	$t(2,2m)$
…	…	…	…	…	…
M_{m-1}	$t(2,\ m(m-3)/2+1)$	…	…	…	$t(2,\ m(m-1)/2)$
M_m			$t(1,\ m(m-1)/2+1)$		

在上述实例中

$$T^{\mathrm{LS}}=(1+\alpha)(m-1)+\alpha m+m=(1+\alpha)(2m-1);\qquad T^*=m$$

$$\frac{T^{\mathrm{LS}}}{T^*}=(1+\alpha)\left(2-\frac{1}{m}\right)$$

所以定理 1 中的界可达。证毕。

推论 1　若定理 1 中的工件均无准备时间，即所有 $w(*,*;*,*)=0$，则问题在上述改进的 LS 算法下的界为

$$\frac{T^{\mathrm{LS}}}{T^*}\leqslant\left(2-\frac{1}{m}\right)$$

且对任意的 α 此界是紧的。

参考文献：

[1]　GRAHAM R L. Bounds on multiprocessing timing anomalies[J]. SIAM J Appl Math, 1969, 17(2): 416－429.

[2]　OVALICK I M, UZSOY R. Worst-case error bounds for parallel machine scheduling problems with bounded sequence dependent setup times[J]. Operations Research Letters, 1993, 14:251－256.

[3]　秦成林,武俊奇. 具有 m 台通用机的 $P//C_{\max}$ 问题的两种算法[J]. 应用数学与计算数学学报, 1995, 9(1): 39－45.

[4]　秦成林,程建纲. 多专用机的两组工件的 $P//C_{\max}$ 问题[J]. 数学研究, 1995, 28(4): 100－104.

[5]　秦成林，程建纲. 两组工件的 $P//C_{\max}$ 问题的近似解的随机改进算法[J]. 上海大学学报: 自然科学版, 1996, 2(5): 479－486.

[6]　秦成林，唐黎平，程建纲. 有资源约束的 $Q/res/C_{\max}$ 问题的改进型算法[J]. 运筹学学报, 1997, 1(2): 69－75.

[7]　秦成林，潘家定. 具有两台专用机、两台通用机的 $Q_4//C_{\max}$ 问题的近似算法[J]. 运筹学学报, 1998, 2(1): 64－70.

[8]　丁伟. 具有通用机的两组工件的排序问题[J]. 中山大学学报: 自然科学版, 2004, 43(2): 33－36.

[9]　丁伟. 具有通用机的三组工件的排序问题[J]. 上海大学学报: 自然科学版, 2005, 11(1): 48－51.

[10]　丁伟. 具有通用机的 n 组工件的排序问题[J]. 运筹学学报, 2006, 10(4):122－126.

[11]　丁伟. 同速度的具有 m 台通用机的 n 组工件的排序问题[J]. 中山大学学报: 自然科学版, 2008, 47(3): 19－22.

[12]　GAIRING M, MONIEN B, WOCLAW A. A faster combinatorial approximation algorithm for scheduling unrelated parallel machines[J]. Theoretical Computer Science, 2007, 380: 87－99.

[13]　丁伟. 具有通用机的多组工件的 $Q//C_{\max}$ 问题的近似算法[J]. 中山大学学报: 自然科学版, 2010, 49(1): 5－8.