



带有装卸服务器的三台平行机排序问题的 LS 算法

马春磊, 胡觉亮, 蒋义伟
(浙江理工大学理学院, 杭州 310018)

摘要: 针对一个装载服务器和一个卸载服务器的情形, 研究三台平行机上的排序问题。每个工件在加工前需要由装载服务器安装到机器上, 加工结束后由卸载服务器进行卸载。装载和卸载时间均为单位时间, 目标是极小化最大完工时间。该问题是 NP-难问题, 因此采用经典的 List scheduling (LS) 算法进行求解。通过引入块的概念对 LS 排序的结构进行分析, 进而证明了 LS 算法的最坏情况界至多为 $17/9$ 。

关键词: 平行机排序; 服务器; 最坏情况界; makespan; LS 算法

中图分类号: O233

文献标志码: A

文章编号: 1673-3851 (2019) 01-0122-05

LS algorithm for scheduling three parallel machines with loading and unloading server

MA Chunlei, HU Jueliang, JIANG Yiwei

(School of Sciences, Zhejiang Sci-Tech University, Hangzhou 310018, China)

Abstract: In this paper, the scheduling problem for three parallel machines was studied for the case of one loading server and one unloading server. Before processing, each job needs to be installed on the machine by the loading server. After the processing ends, the unloading server unloads. The loading and unloading time is the unit time. Our goal is to minimize the makespan. Since the problem is NP-hard problem, the classical List scheduling (LS) algorithm is applied to solve it. The structure of LS scheduling is analyzed by introducing the concept of block. Finally, it is shown that the worst-case ratio of LS algorithm is at most $17/9$.

Key words: parallel machine scheduling; server; worst case ratio; makespan; LS algorithm

0 引言

带服务器的平行机排序问题是一类新型的排序模型, 该类问题在很多领域, 特别是在制造业中有着广泛的应用。在柔性制造系统 (Flexible manufacturing system) 中, 往往通过一个机器人^[1]或者一台自动导引车辆^[2]服务于各台生产机器, 负责安装和卸载产品。在单元式制造中 (Cellular manufacturing)^[3]和半导体行业^[4], 也需要服务装置来回于各个工作台进行物料的装卸操作。

关于带服务器的平行机排序问题, 最初的研究主要集中在带有一个装载服务器的排序问题。Kravchenko 等^[5]首先证明了装载时间为单位时间的两台机排序问题是 NP-难的, 并给出了一个伪多项式时间算法。Hall 等^[6]则证明了加工时间为单位时间的两台机问题是多项式可解的, 而当装载时间相等时, 该问题是 NP-难的。当所有工件的加工时间相等时, Burkner 等^[7]证明了该问题是 NP-难的。Jiang 等^[8]则考虑了工件允许中断的情形, 分别对单位装载时间和单位加工时间情形给出了最优算

收稿日期: 2018-06-08 网络出版日期: 2018-10-08

基金项目: 国家自然科学基金项目 (11471286, 11571013)

作者简介: 马春磊 (1990-), 男, 河南叶县人, 硕士研究生, 主要从事运筹与组合优化理论方面的研究。

通信作者: 蒋义伟, E-mail: ywjiang@zjgsu.edu.cn

法。Cheng 等^[9]考虑了更为一般的可中断情形,给出了 m 台机的近似算法。Hamzadayi 等^[10-11]研究了 m 台机器环境下的新模型和智能优化算法。Su^[12]应用 LPT 算法了带服务器问题的在线排序问题。Jiang 等^[13]则分析了两台机情形的 LS(List scheduling)算法并给出了改进的在线算法。

近年来,关于服务器同时具有装载和卸载操作功能的研究开始陆续出现。当装、卸载时间均为单位时间时的两台机排序问题,Jiang 等^[14]证明了 LS 和 LPT 的最坏情况紧界分别为 $12/7$ 和 $4/3$ 。而在可中断情形下,该问题存在 $O(n \log n)$ 的多项式时间算法^[15]。在工件事先分划好的情况下,Xie 等^[16]给出了一个 2 的紧界算法。

在上述的研究中,所有的装卸载操作均由一个服务器来完成。Jiang 等^[17]则首次考虑了带有两个专用服务器的情形,即一个服务器用来装载工件,另一个服务器用来卸载工件,问题可以表示为 $P2, S2 | s_j = t_j = 1 | C_{\max}$ 。他们分别给出了 LS 和 LPT 算法的最坏情况界至多为 $8/5$ 和 $6/5$ 。

本文在以上成果的基础上研究带有一个装载和一个卸载服务器的三台平行机排序问题。具体来说,给定一个包含 n 个工件的工件集 $N = \{J_1, J_2, J_3, \dots, J_n\}$ 和三台机器 M_1, M_2 和 M_3 ,每个工件在加工之前需要由装载服务器将工件安装到一台机器上,在加工完后由卸载服务器进行卸载。工件 J_j 的装载时间工件 s_j 和卸载时间工件 t_j 均为单位时间,即 $s_j = t_j = 1$,加工时间 p_j 为正整数。一旦装载

(卸载)完毕,装载(卸载)服务器就可以装载(卸载)其他工件。当机器在加工工件时,该机器不能再进行装载和卸载操作。工件不允许中断,目标是极小化最大完工时间,用三参数法可表示为 $P3, S2 | s_j = t_j = 1 | C_{\max}$ 。由于机器数量的增加,本文重新定义了块的概念,并分析了 LS 排序的结构特点,最后证明了 LS 算法的最坏情况界至多为 $17/9$ 。

1 符号与预备结论

为方便具体描述带有装卸服务器平行机排序问题,定义如下记号:记 $e_j = s_j + p_j + t_j$ 为工件 J_j 的执行时间,其中 $s_j = t_j = 1$,因此 $e_j = p_j + 2 \geq 3$ 。记 $E = \sum_{i=1}^n e_i$ 和 $e_{\max} = \max_{1 \leq i \leq n} e_i$ 为所有工件的总执行时间和最大执行时间。分别用 C^A 和 C^* 表示在算法 A 下得到的最大完工时间和最优完工时间。

首先,给出两个定义:

定义 1 若三台机器上有两个连续卸载时间且在此时间段内第三台机非空闲,则称这两个连续卸载操作的结束点为结点;若三台机器上存在三个连续卸载时间,也称这三个连续卸载操作的结束点为结点。特别地,称零时刻为第一个结点。

定义 2 a) 两个结点之间的时间段称为一个块。b) 最后一个结点至排序结束称为一个不完全块。

对于上述两个定义中提到的概念,图 1 具体给出了机器 M_1, M_2 和 M_3 结点的位置以及块与不完全块的分布和结构。

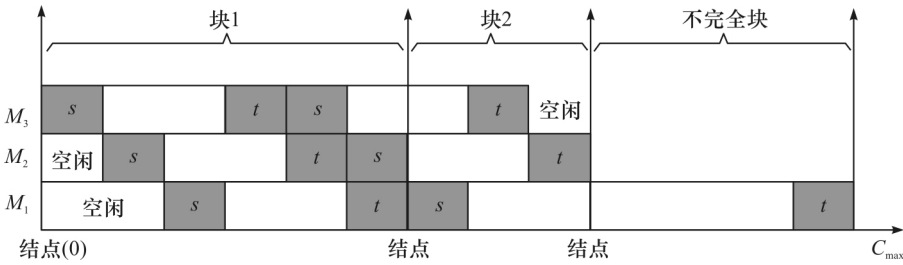


图 1 结点、块与不完全块示意图

与文献^[17]关于两台机情形类似,可以得到关于三台机情形下的最优目标函数值的下界。

引理 1 对于问题 $P3, S2 | s_j = t_j = 1 | C_{\max}$,

$$C^* \geq \left\{ e_{\max}, \frac{E+6}{2} \right\}。$$

下面给出三台机情形下 LS 排序的相关性质。LS 算法的主要思想是把当前工件安排在最早空闲的机器上进行加工,具体到本文所考虑的问题,可以表述如下:

a) 记 $J_j (1 \leq j \leq n)$ 为当前所需加工的工件, l_i 为机器 M_i 的当前完工时间。

b) 令 $l_k = \min\{l_i\}$, 把工件 J_j 分给机器 M_k 。

c) 在保证装载服务器空闲且不会与其他工件的卸载时间冲突的情况下,尽可能早地装载工件 J_j 到机器上并开始加工。

引理 2 按 LS 算法得到的最终排序 σ , 必为下面三种情形之一。

a) 整个排序为一个不完全块, 记作 $\sigma = B'$;

b) 整个排序由连续的块构成,记作 $\sigma = (B_1, B_2, \dots, B_k)$;

c) 整个排序由一些连续的块和一个不完全块构成,记作 $\sigma = (B_1, B_2, \dots, B_k, B')$ 。

为方便叙述,用 $r(B)$ 表示块 B 内部所产生的空闲单位的数量,这里不包括块开始加工所产生的空闲。根据块的定义以及 LS 的规则可以得到以下结论。

引理 3 对任意的一个块 $B, r(B) \leq 3$ 。

推论 1 对于经典的 LS 算法 $\sigma = (B_1, B_2, \dots, B_k, B')$, 有:

a) $r(B_1) = 3, 4, 5, 6$ 。若 $B' = \emptyset$ 且 $k = 1$, 则有 $r(B_1) = 6, 7, 8, 9$;

b) 对于任意 $2 \leq i \leq k$, 有 $r(B_i) = 0, 1, 2, 3$ 。若 $B' = \emptyset$, 则有 $r(B_k) = 3, 4, 5, 6$;

c) 若 $k > 0$, 则有 $r(B') = 0$; 若 $k = 0$, 则有 $r(B') = 3$ 。

证明 a) 块 B_1 在开始加工时至少有三个单位的空闲, 由于卸载时间不允许重叠, 又会导致其内部产生空闲, 由引理 3 知, 至多形成三个单位的空闲, 因此会产生 3~6 个单位空闲时间。如果 $B' = \emptyset$, 块的结尾也会产生三个单位空闲, 故有 6~9 个空闲时间。与 a) 类似可得 b) 和 c) 中的结论。

引理 4 当 $k \geq 2$ 时, 对于块 B_i 有如下结论:

$$\text{a) } e(B_i) \geq \begin{cases} \frac{9}{4}l(B_i) - 2, i=1 \text{ 或 } k \\ \frac{9}{4}l(B_i), 1 < i < k \end{cases}.$$

$$\text{b) } Y \geq \frac{9}{4}X - 4.$$

证明 当 $i = 1$ 时, 由于 $k \geq 2$, 由推论 1 a) 可知, B_1 至多包含 6 个单位的空闲时间。根据块的定义可进行如下讨论:

a) 若 B_1 包含 3 个单位空闲时间, 则 $l(B_1) \geq 4$,

因此可得

$$e(B_1) \geq 3l(B_1) - 3 \geq \frac{9}{4}l(B_1).$$

b) 若 B_1 包含 4 个单位空闲时间, 则有 $l(B_1) \geq 5$,

因此可得

$$e(B_1) \geq 3l(B_1) - 4 \geq \frac{9}{4}l(B_1) - 1.$$

c) 若 B_1 包含 5 个单位空闲时间, 则 $l(B_1) \geq 6$,

因此可得

$$e(B_1) \geq 3l(B_1) - 5 \geq \frac{9}{4}l(B_1) - 2.$$

d) 若 B_1 包含 6 个单位空闲时间, 则 $l(B_1) \geq 6$,

因此可得

$$e(B_1) \geq 3l(B_1) - 6 \geq \frac{9}{4}l(B_1) - 2.$$

综上所述, 有

$$e(B_1) \geq 3l(B_1) - 6 \geq \frac{9}{4}l(B_1) - 2.$$

当 $i = k$ 时, 由推论 1 b) 可知, B_k 至多包含 6 个单位的空闲时间。同样地, 有

$$e(B_i) \geq 3l(B_i) - 6 \geq \frac{9}{4}l(B_i) - 2.$$

当 $1 < i < k$ 时, 由推论 1 b) 可知, B_i 中至多出现 3 个单位空闲时间, 因此有

$$e(B_i) \geq 3l(B_i) - 3.$$

若 $l(B_i) \geq 4$, 则有

$$\frac{e(B_i)}{l(B_i)} \geq 3 - \frac{3}{l(B_i)} \geq \frac{9}{4}.$$

若 $l(B_i) \geq 4$, 不难得到 B_i 是没有空闲的。于是,

$$e(B_i) \geq 3l(B_i) \geq \frac{9}{4}l(B_i).$$

b) 由 a) 直接计算可得。

2 LS 算法界的证明

本节主要证明 LS 算法的最坏情况界为至多为 17/9。

定理 1 LS 算法的最坏情况界为 17/9。

证明 设 J_l 为最后完工的工件, T 为工件 J_l 的开工时间, 即有 $C^{\text{LS}} = T + e_l$ 。根据引理 2 分情形讨论。

若 $\sigma = B'$, 即整个排序由一个不完整的块组成。由推论 1 c) 可知最多在零时刻产生 3 个单位的空闲时间, 故

$$E \geq 3T + e_l - 3,$$

由引理 1 知

$$C^* \geq \max\left\{\frac{3T + e_l + 3}{3}, e_l\right\} \geq \max\left\{\frac{3T + e_l}{3}, e_l\right\}.$$

因此,

$$\frac{C^{\text{LS}}}{C^*} \leq \frac{T + e_l}{\max\left\{\frac{3T + e_l}{3}, e_l\right\}} \leq \begin{cases} \frac{3T + 3e_l}{3}, e_l \leq \frac{3}{2}T \\ \frac{T + e_l}{e_l}, e_l > \frac{3}{2}T \leq \frac{5}{3} < \frac{17}{9} \end{cases}$$

接下来考虑 $\sigma=(B_1,B_2,\cdots,B_k,B')$ 的情形。
若 $B'=\emptyset$,即 $\sigma=(B_1,B_2,\cdots,B_k)$,则有

$$C^{\text{LS}}=X。$$

另外,每个块的长度至少为 3,由此可知 $X\geqslant 3k$ 。据推论 1 a)和 b)可得总的空闲时间至多为 $3k+6$,即有

$$E\geqslant 3X-(3k+6),$$

进而由引理 1 可知

$$C^*\geqslant \frac{3X-(3k+6)+6}{3},$$

因此,

$$\frac{C^{\text{LS}}}{C^*}\leqslant \frac{3X}{3X-3k}\leqslant \frac{3}{2}<\frac{17}{9}。$$

若 $B'\neq\emptyset$,注意到此时最后一个完工的工件 J_l 必在 B' 中。若 J_l 是 B' 中的前 3 个工件之一,则有

$$C^{\text{LS}}=T+e_l\leqslant X+e_l。$$

结合引理 4 可得

$$E\geqslant Y+e_l\geqslant \frac{9}{4}X-4+e_l,$$

故

$$C^*\geqslant \max\left\{\frac{E+6}{3},e_{\max}\right\}\geqslant \max\left\{\frac{\frac{9}{4}X+e_l}{3},e_l\right\},$$

因此,

$$\frac{C^{\text{LS}}}{C^*}\leqslant \frac{X+e_l}{\max\left\{\frac{\frac{9}{4}X+e_l}{3},e_l\right\}}\leqslant \frac{17}{9}。$$

若 J_l 不是 B' 中的前 3 个工件之一,则在 B' 中至少存在其他工件在 J_l 之前加工,记

$$\Delta=T-X>0,$$

即

$$C^{\text{LS}}=T+e_l=X+\Delta+e_l。$$

令 $e(\Delta)$ 表示从 X 到 T 这个时间段内三台机器上总的加工时间。显然在这段时间内不存在空闲时间,即有 $e(\Delta)=3\Delta>\frac{9}{4}\Delta$ 不难得到

$$E\geqslant Y+e(\Delta)+e_l\geqslant \frac{9}{4}(X+\Delta)+e_l,$$

因此,

$$\frac{C^{\text{LS}}}{C^*}\leqslant \frac{X+\Delta+e_l}{\max\left\{\frac{\frac{9}{4}(X+\Delta)+e_l}{3},e_l\right\}}\leqslant \frac{17}{9}。$$

证毕。

表 1 给出了 10 组实例的算法结果和最优解目

标值。

表 1 LS 算法与最优解的比较结果

组别	C^{LS}	C^*	C^{LS}/C^*
1	69	65	69/65
2	64	57	64/57
3	70	61	70/61
4	48	48	1/1
5	80	62	80/62
6	72	62	72/62
7	51	46	51/46
8	64	60	64/60
9	70	64	70/64
10	54	44	54/44

以上 10 组实例由计算机随机产生,每个实例包含 10 个工件,每个工件的执行时间在 3~30 之间随机产生。从结果上来看,性能比均优于本文理论证明的界,一方面说明 LS 算法在解决此类问题的实际效果还不错,另一方面也说明本文所给的界不一定是紧界。

3 结 论

本文研究带有装、卸载服务器的三台平行机排序问题,目标是极小化最大完工时间。在装、卸载时间均为单位时间的情形下,证明了 LS 算法的最坏情况界至多为 17/9。在后续的研究中,将考虑一般 m 台机情形下的 LS 和 LPT 排序,并分析他们的最坏情况紧界。

参考文献:

[1] Koulamas C. Scheduling two parallel semiautomatic machines to minimize machine interference [J]. Computers and Operations Research,1996,23(10):945-56.

[2] Ganesharajah T, Hall N, Sriskandarajah, C. Design and operational issues in AGV-served manufacturing systems[J]. Annals of Operations Research,1998,76(1):109-154.

[3] Dawande M, Geismar H, Sethi S, et al. Sequencing and scheduling in robotic cells: Recent developments [J]. Journal of Scheduling,2005,8(5):387-426.

[4] Kim M Y, Lee Y H. MIP models and hybrid algorithm for minimizing the makespan of parallel machines scheduling problem with a single server[J]. Computers and Operations Research,2012,39(11):2457-2468.

[5] Kravchenko S, Werner F. Parallel machine scheduling problems with a single server [J]. Mathematical and Computer Modelling,1997,26(12):1-11.

[6] Hall N, Potts C, Sriskandarajah C. Parallel machine

- scheduling with a common server[J]. *Discrete Applied Mathematics*, 2000, 102(3): 223-243.
- [7] Brucker P, Dhaenens-Flipo C, Knust S, et al. Complexity results for parallel machine problems with a single server[J]. *Journal of Scheduling*, 2002, 5(6): 429-457.
- [8] Jiang Y, Dong J, Ji M. Preemptive scheduling on two parallel machines with a single server[J]. *Computers & Industrial Engineering*, 2013, 66(2): 514-518.
- [9] Cheng T C E, Kravchenko S A, Lin B M T. Preemptive parallel-machine scheduling with a common server to minimize makespan[J]. *Naval Research Logistics*, 2017, 64(5): 388-398.
- [10] Hamzadayi A, Yildiz G. Event driven strategy based complete rescheduling approaches for dynamic m identical parallel machines scheduling problem with a common server[J]. *Computers & Industrial Engineering*, 2016, 91: 66-84.
- [11] Hamzadayi A, Yildiz G. Modeling and solving static m identical parallel machines scheduling problem with a common server and sequence dependent setup times [J]. *Computers & Industrial Engineering*, 2017, 106: 287-298.
- [12] Su C. Online LPT algorithms for parallel machines scheduling with a single server[J]. *Journal of Combinatorial Optimization*, 2013, 26(3): 480-488.
- [13] Jiang Y, Yu F, Zhou P, et al. Online algorithms for scheduling on two parallel machines with a single server[J]. *International Transactions in Operational Research*, 2015, 22: 913-927.
- [14] Jiang Y, Zhang Q, Hu J. et al. Single-server parallel-machine scheduling with loading and unloading times [J]. *Journal of Combinatorial Optimization*, 2015, 30(2): 201-213.
- [15] Jiang Y, Wang H, Zhou P. An optimal preemptive algorithm for the single-server parallel-machine scheduling with loading and unloading times[J]. *Asia-Pacific Journal of Operational Research*, 2014, 31(5): 1450039.
- [16] Xie X, Zheng Y, Li Y. Scheduling parallel machines with a single server: A dedicated case [C] // Fifth International Joint Conference on Computational Sciences and Optimization (CSO). IEEE, 2012: 146-149.
- [17] Jiang Y, Zhou P, Wang H, et al. Scheduling on two parallel machines with two dedicated servers [J]. *Anziam Journal*, 2017, 58(3-4): 314-323.

(责任编辑: 康 锋)