ELSEVIER

Discrete Applied Mathematics 85 (1998) 251–254

DISCRETE
APPLIED
MATHEMATICS

Note

The exact bound of Lee's MLPT ☆

Guo-Hui Lin *

Institute of Applied Mathematics, Chinese Academy of Sciences, P.O. Box 2734, Beijing 100 080, China

Received 23 January 1996; received in revised form 28 October 1997; accepted 1 December 1997

Abstract

Lee provided a modified version of LPT algorithm to deal with the problem *Parallel Machines Scheduling with Nonsimultaneous Machine Available Times*, and got an upper bound of $\frac{4}{3}$ for its worst-case performance ratio. An open question is then proposed to obtain the exact value of this ratio, which is determined in this paper. The instance which achieves the ratio is also demonstrated. © 1998 Elsevier Science B.V. All rights reserved.

Keywords: MLPT; Initially; Worst-case performance ratio

1. Introduction

The problem considered in this paper is to schedule n independent jobs onto m identical machines in order to minimize the makespan, the total finishing time. At time zero, all the jobs are available while some machines may not be ready. This problem is called *Parallel Machines Scheduling with Nonsimultaneous Machine Available Times* as in [2]. Note that it is a generalization of the classical multiprocessor scheduling problem, in which the n jobs and the m machines are simultaneously available at time zero. For the ease of presentation, the former problem is denoted as GMS and the latter CMS in this paper.

Notice that the *Longest Processing Time* (LPT) algorithm is a simple and practical one to approximate CMS [1]. Lee [2] analyzed the performance of it applied to GMS and showed that its worst-case performance ratio is $\frac{3}{2} - 1/2m$, where m is the number of machines. That is, for any m-machine instance I of GMS, let M be the makespan attained by LPT and M^* the optimal one, then

$$M/M^* \leqslant \frac{3}{2} - 1/2m.$$

☆ Research supported in part by National Natural Science Foundation of China.

* E-mail: ghlin@amath11.amt.ac.cn.

PII S0166-218X(97)00139-X

Lee also provided a modified version of LPT, called MLPT, expecting to get a better guarantee. This algorithm can be described as follows [2]:

Suppose the real job set is $\mathscr{J} = \{J_1, J_2, \ldots, J_n\}$, where job J_j has a processing time p_j, $j = 1, 2, \ldots, n$, and machine m_i is available at time a_i, $i = 1, 2, \ldots, m$. We consider the a_i with $a_i > 0$ as the processing time of a *wide-sense job*, denoted A_i, and then merge $\mathscr{A} = \{A_i \mid a_i > 0\}$ with $\mathscr{J}$ into a wide-sense job set $\mathscr{S}$.

MLPT Algorithm

Step 0: Sort all the jobs in $\mathscr{S}$ into nonincreasing order with respect to processing time, and MLPT always considers the job at the head of this list. Assume now that all machines are ready at time zero.

Step 1: Consider the job at the head of $\mathscr{S}$. If the job is a real job J_j for some j, then assign it to a smallest loaded machine. Let $\mathscr{S} = \mathscr{S} - \{J_j\}$ and go to Step 4. Otherwise, the job is a wide-sense job A_i for some i and go to Step 2.

Step 2: If there is a smallest loaded machine which has not been assigned any wide-sense job, then assign job A_i to it and let $\mathscr{S} = \mathscr{S} - \{A_i\}$ and go to Step 4. Otherwise, go to Step 3.

Step 3: Let $\mathscr{J}'$ denote the set of (real) jobs which have been assigned by MLPT to those machines each of which has not been assigned any wide-sense job. Denote the job which comes latest in the job list in $\mathscr{J}'$ as J_f. Replace J_f by A_i and reassign J_f to a smallest loaded machine. Let $\mathscr{S} = \mathscr{S} - \{A_i\}$ and go to Step 4.

Step 4: If $\mathscr{S} = \emptyset$, go to Step 5. Otherwise, return to Step 1.

Step 5: Move all wide-sense jobs to the head of their respective scheduled machines.

Remarks

(1) In Steps 1, 2 and 3 there may be more than one machine each of which has the smallest load. To break tie, MLPT, always chooses the one with the minimum index.

(2) In Step 3, job J_f has the currently longest processing time among all those unassigned jobs.

(3) No real job can be assigned by the algorithm more than twice because it must end up on a machine with a wide-sense job after the second assignment; and, every wide-sense job A_i can be assigned exactly once.

Lee proved that the MLPT-makespan M is upper bounded by $\frac{4}{3}M^*$ and believed that actually it could be somewhat smaller than $\frac{4}{3}M^*$. However, we have known that the worst-case performance ratio of MLPT cannot be less than $\frac{4}{3} - 1/3m$, which is the worst-case performance ratio of LPT applied to CMS. This gap provides an interesting open question (as stated in [2]).

In the next part of this paper, we complete Lee's analysis by establishing the worst-case performance ratio in the case of two machines, and give an example that shows that Lee's $\frac{4}{3}$-bound is tight in the case of three or more machines.

2. Main results

The main results of this paper are summarized in the following theorem, and the rest of this section is devoted to the proof.

Theorem 1. *Applying MLPT to GMS, the worst-case performance ratio is*

$$R_{\mathrm{MLPT}} = \begin{cases} \leqslant \frac{4}{3}, & \text{when } m \geqslant 3, \\ \leqslant \frac{5}{4}, & \text{when } m = 2, \end{cases}$$

where m denotes the number of machines.

Proof. Given any m-machine instance I of GMS, let M denote the makespan attained by MLPT and M^* the optimal one. Recall that Lee has showed $M \leqslant \frac{4}{3}M^*$ for all m, $m \geqslant 2$. Thus, in the case that $m \geqslant 3$, we need only to give an instance to show the tightness of the ratio. Let us consider the following example:

Example. In this instance, there are m machines, $m \geqslant 3$, and $n = m+1$ jobs. The wide-sense job list after sorting is $\mathscr{S} = \{A_1, A_2, \ldots, A_{m-3}, J_1, A_{m-2}, A_{m-1}, J_2, J_3, \ldots, J_{m-1}, J_m, J_{m+1}, A_m\}$ and the processing times are

$$a_i = 3 + \varepsilon, \quad i = 1, \ldots, m-3;$$

$$a_{m-2} = a_{m-1} = 3 - \varepsilon;$$

$$a_m = 2.$$

$$p_1 = 3;$$

$$p_j = 3 - 2\varepsilon, \quad j = 2, \ldots, m-1;$$

$$p_m = p_{m+1} = 2 + \varepsilon.$$

Trivially, for this instance, we have $M = 8 - 2\varepsilon$ and $M^* = 6 + 2\varepsilon$. It follows that $M/M^* \to \frac{4}{3}$ as $\varepsilon \to 0$.

The remaining part is dedicated to showing the theorem in the case of $m = 2$. Our proof is a modification of that in [2] to fit the case of $m = 2$. For easy presentation, the MLPT-schedule in the following refers to the one obtained in the MLPT procedure just before executing Step 5.

In MLPT-schedule, note that the makespan M should occur on some machine, say m_1. Let job Y denote the last job (either a real or a wide-sense job) assigned to m_1 and its processing time is p. Let s denote the starting time of job Y on m_1. Since the machine to which Y is not assigned must have a load of at least s, the total load on both machines is at least $2s+p$. A simple geometric argument thus shows that $2M^* \geqslant 2s+p$. It follows that

$$M = s + p \leqslant M^* + \tfrac{1}{2}p. \tag{1}$$

Suppose to the contrary we have a counterexample, that is, we have a two-machine instance for which the inequality $M > \frac{5}{4}M^*$ holds. Then, by (1), we derive that

$$p > \tfrac{1}{2}M^*. \tag{2}$$

As we have mentioned in Remark (3) that a real job may be assigned by MLPT algorithm twice, let us consider the following two cases:

Case 1: *Y is "initially" assigned to m_1.*

In this case we know that job Y is just taken from the job list $\mathscr{S}$. Therefore, $s > 0$, for otherwise we have $p = M > M^*$, which contradicts the definition of M^*. Thus, there are at least three jobs in $\mathscr{S}$ each of which has a processing time at least p. It follows immediately that in any schedule of this counterexample there is always a machine with its completion time greater than M^*. This is a contradiction to the definition of M^*.

Case 2: *Y is reassigned to m_1.*

In this case job Y is a real job. Assume job Y is "initially" assigned to m_k, and the starting time of job Y on m_k, is t. We claim that $t > 0$.

By contradiction, suppose $t = 0$. Let A_f denote the wide-sense job which replaces job Y from machine m_k. According to Step 3 of the algorithm, once job Y has been replaced by job A_f on m_k, but before job Y gets reassigned to its new machine, m_k has a load of a_f. Since job Y is then reassigned to a smallest loaded machine, which is exactly m_1, it is clear that $s \leqslant a_f$. Because Step 3 is invoked for A_f, there is not a smallest-loaded machine without a wide-sense job, and thus at least one machine already has a wide-sense job. This shows that the "not-m_k" machine must have a wide-sense job A_e, which (by LPT rule) must have $a_e \geqslant a_f$. Therefore, we have $s = a_f$ and $M = s + p = a_f + p$. Since $\mathscr{S}$ has at least three jobs A_e, A_f and Y, there is a machine with its completion greater than M^* in any schedule of this counterexample. Thus a contradiction occurs.

In that $t > 0$, the same argument as in Case 1 leads to the same contradiction.

We have shown that in both cases the counterexample does not exist. Hence, the MLPT makespan M in the case of two machines is upper bounded by $\frac{5}{4}M^*$. A 2-machine instance with its M/M^* approaching $\frac{5}{4}$ has been presented in [2], thus, we conclude that this worst-case performance ratio is $\frac{5}{4}$. □

Acknowledgements

Thanks go to Professor X.-D. Hu and two referees for their many helpful suggestions.

References

[1] R.L. Graham, Bounds on multiprocessing timing anomalies, SIAM J. Appl. Math. 17 (1969) 416–429.

[2] C.Y. Lee, Parallel machines scheduling with nonsimultaneous machine available time, Discrete Appl. Math. 30 (1991) 53–61.