IE Interfaces
Vol. 16, Special Edition, pp. 132-137, December 2003.

LPT Scheduling for Multipurpose Machines

Hark-Chin Hwang†

Department of Industrial Engineering, Chosun University, Gwangju, 501-709

여러 종류의 작업 처리가 가능한 기계 시스템에 대한 LPT 스케줄링

황 학 진

조선대학교 산업공학과

We consider scheduling jobs on multipurpose machines where jobs can be processed by a subset of the machines operated in parallel with the objective of minimizing makespan. We apply LPT(Longest Processing Time first) algorithm and prove that its posterior worst-case performance ratio is at most $\log_2 4m/(1+\lambda)$, where λ is the number of machines eligible for processing the job with the latest completion time. In general, LPT is shown to always find a schedule with makespan at most $\log_2 4m/3$ times optimum.

Keywords: analysis of algorithm, posterior performance, eligibility

1. Introduction

In multipurpose parallel machine systems, each machine is flexible to server several different kinds of jobs. In the viewpoint of jobs, each job has several eligible machines in which it can be possibly processed. We are given set of machines $M=1,2,\cdots,m$ and set of jobs $J=1,2,\cdots,n$. Each job j has processing time or size p_j and set of eligible machines $E_j \subseteq M$ for $j=1,2,\cdots,n$, each machine in E_j can process the job j. Under the objective function of makespan minimization, the problem can be formally formulated as the following integer program:

Minimize C

Subject to

$$\sum_{i \in E_j} x_{ij} = 1 \quad \text{for} \quad j=1,\cdots,n$$

$$\sum_{j: i \in E_j} p_j x_{ij} \leq C \quad \text{for} \quad i=1,\cdots,m$$

$$x_{ij} \in \{0,1\}.$$

The application comes from assembly lines. In a factory in a manufacturing company in Korea, the capacities of assembly lines are allocated to hundreds of different kinds of displays for computer, TV, an so on. At the beginning of each month, decisions are to be made how to assign demands from around of the world to the lines. The lines are flexible to process several kinds of displays. In order to increase the utilization of the factory, it is necessary to distribute the loads of demands over the lines evenly. Also, in health care industry and semiconductor manufacturing, the problem can be applied: operating rooms in hospital can handle several cases of operations (Vairaktarakis and Cai, 2001) and machines in wafer test workcenters are multipurpose to process

† Corresponding author : Hark-Chin Hwang, Department of Industrial Engineering, Chosun University, Gwangju, 501-709, Korea, Fax : +82-62-230-7128, E-mail : hchwang@chosun.ac.kr

different kinds of jobs (Centeno and Armacost, 1997).

The multipurpose parallel machine scheduling problem is a generalized version of the classical parallel machine scheduling problem, where all the jobs can be assigned to any machines. Since even the classical problem is known to be NP-Complete(Garey and Johnson, 1979), it seems impossible to develop an optimal polynomial time algorithm for the multipurpose scheduling problem.

When every machine is eligible for every job, a schedule with makespan at most 4/3-1/(3m) times optimal is guaranteed by the algorithm LPT (Longest Processing Time first). The first posterior worst-case analysis for this problem has been done by Coman and Sethi (1976). They showed the makespan of the LPT schedule is at most $1+1/k-1/km$ times optimal, where k is the number of jobs placed on the machine to which the job with the latest completion time is assigned. In extending the result, Blocher and Chand (1991) improved the bound and proved that LPT always generates a schedule with makespan at most $1+1/k-1/km$ times π, where π is a lower bound to the optimum makespan.

For the multipurpose machine scheduling, LS(List Scheduling) algorithm is proved to have its worst-case performance ratio $\log_2 2m$ (Azar *et al.*, 1995). Recently, Hwang *et al.* (2003) showed that the posterior performance ratio of LS is $\log_2 \frac{4}{\lambda} m - \frac{1}{\lambda}$, where λ is the number of eligible machines for the job with the latest completion time. Using the LP-based algorithm of Lenstra *et al.* (1990), we can guarantee a schedule with makespan at most twice optimum. In theory, the LP(Linear Programming)-based algorithm produces much better schedules than those produced by algorithms based on simple dispatching rules, for instance, LS and LPT. However, in real manufacturing, it is often requested to schedule jobs in almost real time. Since LP-based algorithm uses binary searches as its major loops with executing LP at each iteration, it takes long time to get final result.

In this paper, we consider LPT scheduling for our problem and prove that its *posterior* performance ratio is at most $\log_2 \frac{4}{1+\lambda} m$, where λ is the number of eligible machines for the job with the latest completion time. Furthermore, we show that the worst-case performance ratio of LPT is at most $\log_2 \frac{4}{3} m$, in general.

In the next section, we present notations and some properties of LPT schedule. Then, in Section 3, we will prove the posterior performance ratio of the algorithm LPT. Finally, concluding remarks are given in Section 4.

2. Notations and Basic Properties of LPT Schedule

For machines $M=1,2,\cdots,m$ and jobs $J=1,2,\cdots,n$, a schedule is thought of as a partition of the n jobs into m disjoint sets, $S=\langle S_1,\cdots,S_m\rangle$ such that $j\in S_i$ if machine i belongs to the eligible set of job j, that is, $i\in E_j$. The makespan of a schedule S, denoted by $z(S)$, is defined as $\max_{1\le i\le m} p(S_i)$, where $p(S_i)=\sum_{j\in S_i} p_j$. And if the schedule S is produced by an algorithm, we also denote its makespan by $z^A(J,M)$. Denote the optimum makespan for (J,M) as $z^*(J,M)=\min z(S)$, where the minimization is over all the schedules S. Then the *worst-case performance ratio* for an algorithm A is defined by

$$R(A)=\sup\left\{\frac{z^A(J,M)}{z^*(J,M)}:\ \text{for all instances }(J,M)\right\}.$$

Then we consider LPT to attack the multipurpose parallel machine scheduling problem. Algorithm LPT can be described as follows:

Algorithm LPT

Step 1. Assign all the jobs with $|E_j|=1$.

Step 2. Pick a job j with the largest size among the unassigned jobs, and assign it to the machine in the set E_j with the least load, while breaking ties by choosing the machine with the least index.

We introduce some notations with respect to the problem instance (J,M) and the schedule generated by LPT in the following.

- $S(S^*)$: the LPT (optimum) schedule for the instance (J,M), respectively.
- $z^{LPT}(z^*)$: for the ease of notation, z^{LPT} and z^* will be used instead of $z^{LPT}(J,M)$ and $z^*(J,M)$ to denote the makespan of the

LPT schedule and the optimum makespan, respectively.

- q : the job with the latest completion time in the LPT schedule.
- S': the schedule produced by LPT just right before job q is assigned.
- r : the machine to which job q is assigned, i.e., $q \in S'_r$ and hence $q \in S_r$.
- f_i : the finishing time of machine i, i.e., the maximum of completion times of jobs on machine $i, i=1,\cdots,m$.
- λ : the cardinality of the eligible set for the job q, i.e., $\lambda=|E_q|$.
- $u, \varDelta$: u is a positive integer and $\varDelta$ is a value in $[0,1)$, satisfying

$$z^{LPT}=uz^*+\varDelta$$

- R_k : the total load of all the machines which should be processed after time kz^*. Formally, we let

$$R_k=\sum_{i=1}^{m}\max\{0, f_i-kz^*\} \text{ for } k=0,1,\cdots,u.$$

Let's consider R_k, the total load to be processed after time kz^*. Note that $R_0=\sum_{j=1}^{n}p_j$. The following lemma is proved by Azar(1995) and we present it without proof here.

Lemma 1: $R_{k-1}\geq 2R_k$ for all $k=1,\cdots,u$.

Noting the way job a_q is assigned by the algorithm LPT, we can observe that $p(S'_i)+p_q\geq z^{LPT}$ for all $i\in E_q$. Thus, $p(S'_i)+z^*\geq z^{LPT}$ for all $i\in E_q$, since $p_q\leq z^*$. Therefore, we conclude that

$$p(S'_i)\geq z^{LPT}-z^* \text{ for all } i\in E_q \tag{1}$$

3. The Posterior Performance Ratio of Algorithm LPT

Now we present the following theorem, which is an extended result of Hwang *et al.*(2002).

Theorem 1: $\frac{z^{LPT}}{z^*}\leq \log_2\frac{4}{1+\lambda}m$.

Before proving Theorem 1, we consider the special case of the theorem when $m=\lambda$.

First of all, note that

$$\sum_{i=1}^{m}p(S'_i)+p_q\leq mz^*. \tag{2}$$

Due to the fact that $\lambda=m$, we also have

$$p(S'_i)+p_q\geq z^{LPT}, \text{ for all } i=1,\cdots,m. \tag{3}$$

Lemma 2: If $\lambda=m$, then we have $\frac{z^{LPT}}{z^*}\leq \frac{3}{2}-\frac{1}{2m}$ and thus $\frac{z^{LPT}}{z^*}\leq\log_2\frac{4}{1+\lambda}m$.

Proof. Now, we assume that the lemma is false, i.e., $z^{LPT}>\left(\frac{3}{2}-\frac{1}{2m}\right)z^*$. Then, from (3), it holds that

$$\sum_{i=1}^{m}p(S'_i)+mp_q\geq mz^{LPT}>m\left(\frac{3}{2}-\frac{1}{2m}\right)z^*. \tag{4}$$

Then, this together with equation (2) implies $p_q>\frac{1}{2}z^*$.

With respect to the incomplete schedule S', we define M' to be a set of machines such that $i\in M'$ if every job in S'_i is designated job(whose eligible set has only one element). Then, for each $i\in M-M'$, S'_i has at least one job having more than one eligible machine. From the rule of LPT we can easily see that if a job is assigned before job q and it has more than one eligible machine, then the job has size greater than or equal to that of q. Hence, for each S'_i, $i\in M-M'$, there exists at least one job whose length is greater than or equal to that of job q.

In optimal schedule S^*, a non-designated job j with $p_j\geq p_q$ cannot be assigned to the machines in M', since $p(S'_i)+p_j\geq p_q+p(S'_i)>z^*$ and $S'_i\subseteq S^*_i$ for all $i\in S$.

If we let $k=|S|$, we know that there are at least $m-k+1$ non-designated jobs (including job q) with size strictly greater than $\frac{1}{2}z^*$ which should be assigned to the $m-k$ machines in $M-M'$. If so, we can see that it is impossible to get a schedule whose makespan is within the optimum makespan z^*. This is a contradiction. Thus, $\frac{z^{LPT}}{z^*}\leq\frac{3}{2}-\frac{1}{2m}$. Next, it is also true that if $\lambda=m$ then $\frac{z^{LPT}}{z^*}\leq\log_2\frac{4}{1+\lambda}m$, since $\frac{3}{2}-$

$\frac{1}{2m} \leq \log_2 \frac{4m}{1+m}$.

Suppose that $u \geq 2$. Then S is not an optimal schedule and this implies

$$\lambda \geq 2. \tag{5}$$

For the machine r to which the job q is assigned, we also have

$$p(S'_q) \geq uz^* + \Delta. \tag{6}$$

By equation (1), for all $i \in E_q$,

$$p(S'_i) \geq z^{LPT} - z^* = (uz^* + \Delta) - z^* = (u-1)z^* + \Delta$$
$$\text{for all } i \in E_q. \tag{7}$$

From the fact that $\lambda = |E_q|$ with equation (6) and equation (7), we obtain that

$$R_{u-2} = \sum_{i \in E_q} (p(S_i) - (u-2)z^*)$$
$$+ \sum_{i \in M - E_q} \max 0,\ p(S_i) - (u-2)z^*$$
$$\geq (1+\lambda)z^* + \lambda\Delta + \sum_{i \in M - E_q} \max 0,\ p(S_i) - (u-2)z^*$$

Hence, if $u \geq 2$, it holds

$$R_{u-2} \geq (1+\lambda)z^* + \lambda\Delta$$
$$+ \sum_{i \in M - E_q} \max 0,\ p(S_i) - (u-2)z^*. \tag{8}$$

Now, we establish the following lemma to prove Theorem 1.

Lemma 3: If $u \geq 2$, then $R_{u-2} \geq (1+\lambda)(\Delta + z^*)$.

Proof. We assume that the lemma is false, i.e.,

$$R_{u-2} < (1+\lambda)(\Delta + z^*) \tag{9}$$

Adding this to equation (8), we obtain

$$(1+\lambda)(\Delta + z^*) > (1+\lambda)z^*$$
$$+ \lambda\Delta + \sum_{i \in M - E_q} \max 0,\ p(S'_i) - (u-2)z^*$$

This implies

$$\Delta > 0 \tag{10}$$

and

$$p(S'_i) < (u-2)z^* + \Delta, \text{ for all } i \in M - E_q. \tag{11}$$

For each $i \in E_q$, we use j_i to denote the last job assigned to S'_i. Then, from equation (7) and equation (11) with the fact that each $p_{j_i} \leq z^*$, we observe that each job in $j_i; i \in E_q \cup q$ cannot be assigned to the machines in the set $M - E_q$. Moreover, note that $p(S'_i) \geq (u-1)z^* + \Delta > z^*$ since $u \geq 2$ and $\Delta > 0$. Thus the jobs j_i have their completion time greater than z^*. This means that they are not designated job, i.e., have at least two eligible machines. Hence, they are assigned by Step 2 of the LPT algorithm and hence their sizes are at least that of job q.

Now, we will obtain lower bound on the size of job q. First, note that

$$R_{u-1} = \sum_{i \in E_q} (p(S_i) - (u-1)z^*)$$
$$+ \sum_{i \in M - E_q} \max 0,\ p(S_i) - (u-1)$$
$$\geq \sum_{i \in E_q} (p(S_i) - (u-1)z^*).$$

Hence,

$$R_{u-1} \geq \sum_{i \in E_q} (p(S_i) - (u-1)z^*)$$
$$= p(S_r) - (u-1)z^*$$
$$+ \sum_{i \in E_q - r} (p(S_i) - (u-1)z^*)$$
$$= p(S'_r) + p_q - (u-1)z^*$$
$$+ \sum_{i \in E_q - r} (p(S_i) - (u-1)z^*)$$
$$\geq p(S'_r) + p_q - (u-1)z^*$$
$$\sum_{i \in E_q - r} (p(S_i) - (u-1)z^*)$$
$$= \sum_{i \in E_q} (p(S'_i) - (u-1)z^*) + p_q$$
$$\geq \lambda(p(S_r') - (u-1)z^*) + p_q,$$

since $p(S'_r) \leq p(S'_i)$ for all $i \in E_q$. That is,

$$R_{u-1} \geq \lambda(p(S'_r) - (u-1)z^*) + p_q \tag{12}$$

Then by Lemma 1 and equation (12), we have

$$R_{u-2} \geq 2\lambda(p(S_r') - (u-1)z^*) + 2p_q. \tag{13}$$

Thus from equation (9) and (13), it is true that

$$(1+\lambda)(\varDelta+z^*)>2\lambda(p(S_r')-(u-1)z^*)+2p_q$$
$$=2\lambda(p(S_r')+p_q-(u-1)z^*)-2(\lambda-1)p_q$$
$$=2\lambda(uz^*+\varDelta-(u-1)z^*)-2(\lambda-1)p_q$$
$$=2\lambda(\varDelta+z^*)-2(\lambda-1)p_q.$$

Therefore,

$$(1+\lambda)(\varDelta+z^*)>2\lambda(\varDelta+z^*)-2(\lambda-1)p_q.$$

Due to equation (5), this implies $p_q>\frac{1}{2}(\varDelta+z^*)>\frac{1}{2}z^*$. Hence, $p_{j_i}\geq p_q>\frac{1}{2}z^*$, for all $i\in E_q$. Thus there are $1+\lambda$ jobs (jobs in j_i: $i\in E_q\cup q$) with size greater than $\frac{1}{2}z^*$, which are only eligible to be processed by some of the machines in E_q. How- ever, it is impossible to schedule $1+\lambda$ jobs with size greater than $\frac{1}{2}z^*$ on λ machines within time z^*. This is a contradiction and the lemma follows.

We are now prepared to prove the last theorem concerning the worst-case bound of the LPT schedule.

Proof of Theorem 1:

If $\lambda=m$, the theorem holds by Lemma 2. If $u<2$ and $\lambda<m$, the theorem also holds, since

$$\frac{z^{LPT}}{z^*} < 2 \leq \log_2\frac{4(1+\lambda)}{1+\lambda} \leq \log_2\frac{4}{1+\lambda}m.$$

Now, we prove the theorem for the case of $m>\lambda$ and $u\geq 2$. When $u\geq 2$, $R_{u-2}\geq(1+\lambda)(\varDelta+z^*)$ by Lemma 3. Also, note that $R_0\geq 2^{u-2}R_{u-2}$ by Lemma 1. Hence, we have

$$mz^*\geq 2^{u-2}(1+\lambda)(\varDelta+z^*).$$

If we let $\delta=\varDelta/z^*$, $0<\delta<1$, we obtain $u\leq\log_2\frac{4m}{(1+\lambda)(1+\delta)}$. Therefore we see that

$$\frac{z^{LPT}}{z^*} < u+\delta$$
$$\leq \log_2\frac{4m}{(1+\lambda)(1+\delta)}+\delta$$
$$\leq \log_2\frac{4}{1+\lambda}m+\delta-\log_2(1+\delta)$$
$$\leq \log_2\frac{4m}{1+\lambda},$$

since $\delta-\log_2(1+\delta)\leq 0$ for all $0<\delta<1$. Hence, the theorem is true

$$z^{LPT}=\log_2\frac{4m}{1+\lambda}.$$

Therefore, we have $\frac{z^{LPT}}{z^*}=\log_2\frac{4m}{1+\lambda}$.

If the LPT schedule is not optimal, the job q has at least two eligible machines. Hence, the following corollary follows directly from Theorem 1.

Corollary 4: $\frac{z^{LPT}}{z^*}=\log_2\frac{4m}{3}$.

6. Conclusions

We considered LPT scheduling for multipurpose parallel machines and proved its posterior performance ratio is at most $\log_2 4m/(1+\lambda)$, where λ is the number of machines eligible for processing the job with the latest completion time. In general, its performance ratio is shown to be at most $\log_2 4m/3$.

This algorithm can be applied to an environment of almost realtime manufacturing. However, it still needs to develop approximation algorithms of better performance ratios with fast running times.

References

Azar Y., Naor J., Rom R. (1995), The competitiveness of On-Line Assignments, *J. Algorithms*, 18, 221-237.

Blocher J. D., Chand S. (1991), Scheduling of Parallel Processors: A Posterior Bound on LPT Sequencing and a Two-Step Algorithm, Naval Research Logistics 38, 273-287.

Centeno G., Armacost R.L. (1997), Parallel machine scheduling with release time and machine eligibility restrictions, *Computers and Industrial Engineering,* 33, 273-276.

Coffman E. G., Sethi R. (1976), A Generalized Bound on LPT Sequencing, Proceedings of the International Symposium on Computer Performance Modeling March, 306-310.

Garey M.R., Johnson D.S. (1979), *Computers and Intractability: A Guide to the theory of NP-Completeness*, Freeman, San Francisco.

Graham R.L. (1969), Bounds on multiprocessor timing anomalies, *SIAM J. Appl. Math.*, 17, 263-269

Hwang H.C., Chang Y.S., Hong Y. (2003), A Posterior Competitiveness for List Scheduling Algorithm on Machines with Eligibility Constraints, accepted by *Asia-Pacific Journal of Operational Research*

Lenstra J.K., Shmoys D.B., Tardos E. (1990), Approximation Algorithms for Scheduling Unrelated Parallel Machines, *Mathematical Programming,* 46, 259-271.

G.L. Vairaktarakis, X. Cai (2001), The Value of Processing Flexibility in Multipurpose Machines, Technical Memorandum Number 744, Weatherhead School of Management, Case Western Reserve University.