Machine covering with combined partial information

Qifan YANG* Yong WU † Yikun HUANG‡

Abstract

Machine scheduling and covering problems may occur in many applications such as load balancing in network communication channel assignment, parallel processing in large-size computing, task arrangement in flexible manufacturing systems, etc. In this paper we study machine covering problems with combined partial information on m parallel identical machines. We analyze the model while optimal value is known in advance, and each job has processing time at most $1/k$ times of the optimal value and the model we only known the processing time of each job is at most $1/k$ times of the optimal value. For those problems, we give lower bounds and present semi-online algorithms which are optimal in some cases.

Keywords: Scheduling, Design and analysis of algorithm, Semi-online, Competitive ratio

1 Introduction

In this paper, we study machine covering problems with combined partial information on m parallel identical machines. Consider a machine set $\mathcal{M}$ of m identical machines $M_1, M_2, \cdots, M_m$ and a job set $\mathcal{J}$ of n independent jobs. We identify jobs by their processing times as $p_j, j = 1, \cdots, n$. Machines and jobs are available at time zero, and no preemption is allowed. The goal is to schedule the jobs to the machines so as to maximize the minimum machine load, where the *load* of a machine is the sum of the processing time of the jobs assigned to that machine. We denote this problem as $Pm||C_{min}$. It has applications in the sequencing of maintenance actions for modular gas turbine aircraft engines and bandwidth allocation on parallel links motivated by issues of QoS.

A scheduling problem is called *offline* if the information of the instance is completely known. If nothing is known about the instance and each job has to be assigned to a machine as soon as it is available, we call the problem *online*. However, in the most realistic situation maybe that some kinds of partial information about future jobs are known in advance. We call such problems as *semi-online*. Algorithms for online (semi-online) problems are called online (semi-online) algorithms. Semi-online scheduling problems have been widely studied in the last decade, different kinds of partial information have been proposed. For example, the total processing time of all jobs $T = \sum_{j=1}^{n} p_j$ is known in advance (denoted by sum), the largest processing time of all jobs $p_{max} = \max\limits_{j=1,\cdots,n} p_j$ is known in advance (denoted by max), the optimal value of the instance is known in advance (denoted by opt), etc. Combined semi-online problems also have been discussed, where two types of different partial information are known in advance in the same time.

*Corresponding author. Department of Mathematics, Zhejiang University, Hangzhou 310027, P. R. China. Tel.:+86 571 88226249; fax:+86 571 88226249; Email: lyang@mail.hz.zj.cn(Q. YANG).

†Department of Mathematics, Zhejiang University, Hangzhou 310027, P. R. China.

‡Department of Mathematics, Linyi Normal University, Linyi, 276000, P. R. China.

The performance of an online (semi-online) algorithm can be measured by its *competitive ratio.* For an instance I and an algorithm A, let $C^A(I)$ (or shortly C^A) denote the objective value produced by A and let $C^*(I)$ (or shortly C^*) denote the optimal value in an offline version. Then the competitive ratio of A is defined as the smallest number c such that for any I, $C^*(I) \leq cC^A(I)$. An online (semi-online) scheduling problem has a *lower bound* ρ if no online (semi-online) algorithm has a competitive ratio smaller than ρ. An online (semi-online) algorithm A is called *optimal* if its competitive ratio matches the lower bound of the problem.

For the online version of $Pm||C_{min}$, Woeginger showed that LS is the optimal algorithm with competitive ratio m, where LS always assign jobs to the machine with the smallest current load. Azar and Epstein proposed an algorithm $Fill$ for $Pm|opt|C_{min}$ with competitive ratio $2-1/m$, which is optimal for $m=2,3,4$, and Ebenlendr shows a lower bound 43/24 and algorithm with competitive ratio 11/6 for any number of machines. For $Pm|sum|C_{min}$ and $Pm|max|C_{min}$, Tan and Wu designed optimal algorithms for both cases with competitive ratio $m-1$. Optimal algorithms for $Pm|sum\ \&\ max|C_{min}$ with competitive ratio $\frac{3}{2}$ when $m=3$ and $m-2$ when $m \geq 4$ are also given in the same paper. For the two machine case, He gave optimal algorithms for four different semi-online problems. Epstein considered a new kind of partial information, where all jobs have processing time at most C^*/k (denoted by $k-bounded$). She proved that LS is the optimal algorithm for $P2|k-bounded|C_{min}$ with competitive ratio $\frac{2k}{2k-1}$ and furthermore designed an optimal algorithm $kFill$ for $P2|opt\ \&\ k-bounded|C_{min}$ with competitive ratio $\frac{2k+1}{2k}$.

In this paper we consider two semi-online problems of $Pm||C_{min}$ with combined partial information. For $Pm|opt\ \&\ k-bounded|C_{min}$, we prove that $\frac{mk+1}{mk}$ is the lower bound and the competitive ratio of $kFillm$, which is a generation of $kFill$, is $\frac{mk+m-1}{mk}$. We also prove that LS is an optimal algorithm for $Pm|k-bounded|C_{min}$ with competitive ratio $\frac{mk}{mk-m+1}$. In section 2 we study the problem $Pm|k-bounded|C_{min}$. Section 3 is devoted to problem and $Pm|opt\ \&\ k-bounded|C_{min}$.

When analyzing a semi-online algorithm A, denote l_i^j be the current load of M_i right before the assignment of p_j, and let L_i be the load of M_i after all the jobs have been assigned, $i=1,\cdots,m$. Thus $C^A = \min_{1\leq i\leq m} L_i$.

2 Optimal algorithm for $Pm|k-bounded|C_{min}$

Before considering the problem $Pm|opt\ \&\ k-bounded|C_{min}$, we will show that LS is the optimal algorithm for $Pm|k-bounded|C_{min}$.

Theorem 2.1 *The competitive ratio of any semi-online algorithm A for $Pm|k-bounded|C_{min}$ is at least $\frac{mk}{mk-m+1}$.*

Proof. The proof will be done by adversary method. The job sequence starts by mk jobs with processing time of 1, at this moment $C^*=k$. If they are not assigned evenly by A, there is at least one machine with load less than $k-1$. Hence, $\frac{C^*}{C^A} \geq \frac{k}{k-1} \geq \frac{mk}{mk-m+1}$. Otherwise there are additional $mk-1$ jobs with processing time of mk. This gives $C^*=mk^2$. However, there is at least one machine with load of at most $k+(k-1)(mk)=mk^2-mk+k$, and we also have $\frac{C^*}{C^A} \geq \frac{mk^2}{mk^2-mk+k} = \frac{mk}{mk-m+1}$. □

Theorem 2.2 *The competitive ratio of LS for $Pm|k-bounded|C_{min}$ is at most $\frac{mk}{mk-m+1}$.*

Proof. Assume $C^{LS} = L_m$. Let p_{j_i} be the last job assigned to $M_i, 1 \le i \le m-1$. By LS rule, we have $L_i - p_{j_i} = l_i^{j_i} \le l_m^{j_i} \le L_m,\ 1 \le i \le m-1$. Together with $p_{j_i} \le \frac{C^*}{k},\ 1 \le i \le m-1$, we have, $C^{LS} = L_m \ge \frac{1}{m}(L_m + \sum_{i=1}^{m-1} l_i^{j_i}) = \frac{T - \sum_{i=1}^{m-1} p_{j_i}}{m} \ge C^* - \frac{m-1}{m} \cdot \frac{C^*}{k} = \frac{mk-m+1}{mk} C^*$. □

From Theorem 2.1 and Theorem 2.2, we know that LS is an optimal semi-online algorithm for $Pm|k-bounded|C_{min}$ with competitive ratio $\frac{mk}{mk-m+1}$.

3 Bounds for $Pm|opt\ \&\ k-bounded|C_{min}$

Theorem 3.1 *The competitive ratio of any semi-online algorithm A for $Pm|opt\ \&\ k-bounded|C_{min}$ is at least* $\begin{cases} \frac{2k+1}{2k} & \text{if } m \le 2k+1, \\ \frac{m}{m-1} & \text{if } m > 2k+1. \end{cases}$

Proof. Let $C^* = 1$ be known in advance, and we will prove the theorem in following two cases.

Case 1 $m \le 2k+1$. Let the first m jobs all with processing time of $\frac{1}{2k+1}$. If they are assigned to less than m machines by A, then mk jobs with processing time of $\frac{2}{2k+1}$ come. Thus we have $C^A \le \frac{2k}{2k+1}$. Otherwise, if the first m jobs with processing time $\frac{1}{2k+1}$ are assigned to m machines, then we finish the job sequence with jobs with processing time of $\frac{1}{k}$ as more as possible and a little job with processing time not greater than $\frac{1}{k}$. The less loaded machine will have load of at most $\max\{(k-1)\frac{1}{k} + \frac{1}{2k+1},\ 1-(m-1)\frac{1}{2k+1}\} \le \frac{2k}{2k+1}$. However, in the optimal schedule, $mk-k$ jobs with processing time $\frac{1}{k}$ are assigned evenly to $m-1$ machines, and the remaining jobs are assigned to one machine.

Case 2 $m > 2k+1$. Let the first m jobs all with processing time of $\frac{1}{m}$. If they are assigned to less than m machines by A, then mk jobs with processing time of $\frac{m-1}{mk}$ come. Thus we have $C^A \le 1 - \frac{1}{m}$. Otherwise, if the first m jobs with processing time $\frac{1}{m}$ are assigned to m machines, then $(m-1)k$ jobs with processing time of $\frac{1}{k}$ come. The less loaded machine will have load of at most $1 - \frac{1}{m}$.

Since we have $C^* = 1$ in all cases above, we are done. □

Now we present a semi-online algorithm $kFillm$ for this problem, which is a generalization of $kFill$. Let $0 < \beta = \frac{mk}{mk+m-1} < 1$. We call p_j as a *tiny* job if $p_j \le \frac{\beta}{k} C^*$, otherwise we call it as a *regular* job. We call a machine is *mixed* if it processes both regular jobs and tiny jobs.

Algorithm $kFillm$

Let p_j be the current job, and

$$\begin{aligned} X^j &= \{M_i | l_i^j < \beta C^*, \text{and there are only regular jobs on } M_i, 1 \le i \le m\}, \\ Y^j &= \{M_i | l_i^j < \beta C^*, \text{and there are only tiny jobs on } M_i, 1 \le i \le m\}, \\ Z^j &= \{M_i | l_i^j = 0, 1 \le i \le m\}. \end{aligned}$$

1. if p_j is a regular job.

   (1.1) if $X^j \ne \emptyset$, assign p_j to the machine with the largest load in X^j.

   (1.2) if $X^j = \emptyset$, $Z^j \ne \emptyset$, assign p_j to an arbitrary machine in Z^j.

   (1.3) if $X^j = Z^j = \emptyset$, $Y^j \ne \emptyset$ assign p_j to the machine with the largest load in Y^j.

(1.4) if $X^j = Z^j = Y^j = \emptyset$, assign p_j by LS rule.

2. if p_j is a tiny job.

   (2.1) if $Y^j \neq \emptyset$, assign p_j to the machine with the largest load in Y^j.

   (2.2) if $Y^j = \emptyset$, $Z^j \neq \emptyset$, assign p_j to an arbitrary machine in Z^j.

   (2.3) if $Y^j = Z^j = \emptyset$, $X^j \neq \emptyset$, assign p_j to the machine with the largest load in X^j.

   (2.4) if $Y^j = Z^j = X^j = \emptyset$, assign p_j by LS rule.

Theorem 3.2 *The competitive ratio of kFillm for $Pm|opt$ & $k-bounded|C_{min}$ is at most* $\frac{1}{\beta} = \frac{mk+m-1}{mk}$.

Proof. We show that when the algorithm terminates, $L_i \geq \beta C^*$, $1 \leq i \leq m$. This implies that the competitive ratio is at most $\frac{1}{\beta}$.

Assume that there is a machine $M_{i'}$ with load less than βC^*. Let

$$\begin{aligned}
\mathcal{M}_1 &= \{M_i | L_i \geq \beta C^*, \text{and there are only regular jobs on } M_i,\ 1 \leq i \leq m\},\\
\mathcal{M}_2 &= \{M_i | L_i \geq \beta C^*, \text{and there are only tiny jobs on } M_i,\ 1 \leq i \leq m\},\\
\mathcal{M}_3 &= \mathcal{M} \backslash (\mathcal{M}_1 \cup \mathcal{M}_2),
\end{aligned}$$

and $m_j = |\mathcal{M}_j|,\ j = 1, 2, 3$. Obviously, $M_{i'} \in \mathcal{M}_3$ and $m_3 \geq 1$.

From the description of *kFillm*, we can verify that for any $1 \leq j \leq n$, $|X^j| \leq 1$ and $|Y^j| \leq 1$. If there exist mixed machines, let M_t be the first one during the scheduling process. Suppose p_r is the first regular (tiny) job assigned to M_t, which previously processing only tiny (regular) jobs, then $X^r = Z^r = \emptyset$ ($Y^r = Z^r = \emptyset$) and $Y^r = \{M_t\}$ ($X^r = \{M_t\}$). As machines in $\mathcal{M} \backslash \{M_t\}$ are not mixed at that time, we must have $l_i^r \geq \beta C^*, \forall M_i \in \mathcal{M} \backslash \{M_t\}$, and in fact $\mathcal{M}_3 = \{M_t\} = \{M_{i'}\}$. If there is no mixed machine, then the loads of machines in $\mathcal{M}_3$ are clearly less than βC^*. Hence, for both cases, we have

$$L_i < \beta C^*, \forall M_i \in \mathcal{M}_3. \tag{1}$$

Moreover, for both cases, no jobs will be assigned to machines in $\mathcal{M}_1 \cup \mathcal{M}_2$ by LS rule since $L_{i'} < \beta C^*$, Therefore, for any machine $M_i \in \mathcal{M}_1$, there are at most k regular jobs on M_i,

$$\beta C^* = k \cdot \frac{\beta}{k} C^* \leq L_i \leq k \cdot \frac{1}{k} C^* = C^*,\ \forall M_i \in \mathcal{M}_1, \tag{2}$$

and

$$\beta C^* \leq L_i < \beta C^* + \frac{\beta}{k} C^*,\ \forall M_i \in \mathcal{M}_2. \tag{3}$$

Since $m_3 \geq 1$ and $m = m_1 + m_2 + m_3$, together with (1), (2) and (3),we have

$$\begin{aligned}
T &= \sum_{i=1}^{m} L_i = \sum_{M_i \in \mathcal{M}_1} L_i + \sum_{M_i \in \mathcal{M}_2} L_i + \sum_{M_i \in \mathcal{M}_3} L_i < m_1 C^* + m_2(\beta + \frac{\beta}{k}) C^* + m_3 \beta C^*\\
&= m_1 C^* + m_2 \cdot \frac{m(k+1)}{mk+m-1} C^* + \frac{m_3 mk}{mk+m-1} C^* = \left(m_1 + m_2 + \frac{m_2 + m_3 mk}{mk+m-1}\right) C^*\\
&= \left(m_1 + m_2 + m_3 + \frac{m - m_1 - m_3 m}{mk+m-1}\right) C^* \leq (m_1 + m_2 + m_3) C^* = mC^*,
\end{aligned}$$

which is a contradiction. □