A lower bound for the on-line preemptive machine scheduling with $\ell_p$ norm

Tianping Shuai$^*$ and Donglei Du$^†$

$^1$School of Sciences, Beijing University of Posts and Telecommunication
$^2$Faculty of Business Administration, University of New Brunswick

June 4, 2008

Abstract

We consider the on-line version of the preemptive scheduling problem that minimizes the machine completion time vector in the $\ell_p$ norm (a direct extension of the $l_\infty$ norm: the makespan) on $m$ parallel identical machines. We present a lower bound on the competitive ratio of any randomized on-line algorithm with respect to the general $\ell_p$ norm. This lower bound amounts to calculating a (non-convex) mathematical program and generalizes the existing result on makespan. While similar technique has been utilized to provide the best possible lower bound for makespan, the proposed lower bound failed to achieve the best possible lower bound for general $\ell_p$ norm (though very close), and hence revealing intricate and essential difference between the general $\ell_p$ norm and the makespan.

Keywords On-line algorithm; Scheduling; Preemption; $\ell_p$ norm

$^*$This work was done while the author was visiting the Faculty of Business Administration, University of New Brunswick. (shuaitp@gmail.com)
$^†$Supported in part by the Natural Science and Engineering Research Council of Canada (NSERC) grant 283103. (ddu@unb.ca)
1 Introduction

We consider the following scheduling problem. We are given $m$ machines and a sequence of jobs. In the variation with preemption any job may be divided into several pieces that may be processed on several machines; in addition, the time slots assigned to different pieces must be disjoint. The objective is to find a schedule which minimizes the $\ell_p$ norm of the machines’ completion times, that is, $\sqrt[\ell_p]{L_1^p + L_2^p + \cdots + L_m^p}$, where $L_i, i = 1, 2, ..., m$ are the completion times of the last job on $M_i$. We denote this problem as $Pm|\text{pmtn, online-list}|\ell_p$. The off-line version of problem, where the full information (number of jobs and sizes of jobs) on all the jobs is known in advance, can be denoted as $Pm|\text{pmtn}|\ell_p$ using the three-field notation in [15].

The makespan, the most studied objective function in scheduling, is just a special $\ell_p$ norm—that is—the $\ell_\infty$. Scheduling in the general $\ell_p$ norm has also been widely studied in the literature [1, 2, 4, 5, 6, 7, 8, 9, 13, 14, 16, 17, 19], and the $\ell_p$ norm of the completion time vector is one of the basic and fundamental objectives investigated in scheduling theory. While the makespan only characterizes the latest completion time among all machines, the general $\ell_p$ norm is more appropriate when we are interested in the average behavior of the machine completion times rather than the worst-case scenario. A particular application of $\ell_p$ norm scheduling in disk storage allocation problem is illustrated in [8].

The main focus of this work is on the lower bound analysis of the competitive ratio of the on-line scheduling problem introduced in the beginning. The quality of an on-line algorithm is measured by its competitive ratio $r_{ON}$, which is defined to be the supremum of ratio $C_{ON}/C_{OFF}$ over all problem instances, where $C_{ON}$ and $C_{OFF}$ denote respectively the $\ell_p$ norm of machine completion time vector of the on-line schedule constructed by $ON$ and that of the corresponding (off-line) optimal schedule.

Related work. Scheduling in the $\ell_p$ norm has been investigated from both on-line (and semi-on-line) [2, 8, 9, 17, 16, 19] and off-line [1, 4, 5, 6, 7, 13, 14] points of view. But we only review the results related to the on-line case as it is the main focus of this work. Readers who are interested in the off-line case are referred to the references listed here and further pointers therein.

Directly related to the current work is [9], in which a best possible on-line algorithm for $P2|\text{pmtn, online-list}|\ell_p$ is developed. Moreover, several semi-online models of $P2||\ell_p$, and $Q2||\ell_p$ are studied (both preemptive and non-preemptive versions) in [8, 16, 19].

Our results. The main contribution of this work is to provide a lower bound for the competitive ratio of any randomized on-line algorithm for the problem $Pm|\text{pmtn, online-list}|\ell_p$, generalizing
existing result for makespan [3]. We adopt a similar lower bounding technique developed in [12, 10] for the uniform machines scheduling \(Qm|\text{pmtn}, \text{online-list}|C_{\text{max}}\). The proposed lower bound amounts to calculating a (non-convex) mathematical program. While the technique of [12, 10] has been utilized to provide the best possible lower bound for makespan, our lower bound derived from the similar technique (with more involved technical development) fails to achieve the best possible lower bound for general \(\ell_p\) norm as shown in this paper (though very close), and hence revealing intricate and essential difference between the general \(\ell_p\) norm and the makespan. This leaves the obvious open question on finding the best possible lower bound for the \(\ell_p\) norm problem by either extending the existing lower bounding techniques or developing more powerful new methods.

The rest of the paper is organized as follows. After some preliminary results in Section 2, we present our lower bound in Section 3.

## 2 Preliminaries and notations

We will need the following closed-form formula of the off-line optimal objective value for the problem \(Pm|\text{pmtn}|\ell_p\). This formula is explicitly implied by a general result in [13]. For the special case of the identical parallel machines, a much simplified proof is possible, which is offered here for completeness.

**Lemma 1** [13] For any given instance of the problem \(Pm|\text{pmtn}|\ell_p\) with \(m\) machines and \(n\) jobs \(J = \{p_n, \ldots, p_1\}\) such that \(p_n \leq \ldots \leq p_1\), the optimal objective value \(\text{OPT}(m,J)\) satisfies:

\[
\text{OPT}(m,J) = \left( \max \left\{ p_1, \frac{\sum_{i=1}^{n} p_i}{m} \right\} \right)^p + \max \left\{ \min \left\{ \frac{\sum_{i=1}^{n} p_i}{m}, \frac{\sum_{i=2}^{n} p_i}{m-1}, \ldots, \frac{\sum_{i=m-1}^{n} p_i}{2} \right\} \right\}^p
\]

\[
+ \ldots + \max \left\{ \min \left\{ \frac{\sum_{i=1}^{n} p_i}{m}, \frac{\sum_{i=2}^{n} p_i}{m-1}, \ldots, \frac{\sum_{i=m-1}^{n} p_i}{2} \right\} \right\}^p
\]

\[
+ \min \left\{ \frac{\sum_{i=1}^{n} p_i}{m}, \frac{\sum_{i=2}^{n} p_i}{m-1}, \ldots, \frac{\sum_{i=m-1}^{n} p_i}{2}, \sum_{i=m}^{n} p_i \right\}^{1/p} \right)(1)
\]

**Proof.** The proof is by induction on \(m\), the number of machines. The formula is obviously true for \(m = 1\). For \(m = 2\), if \(p_1 \leq \sum_{i=1}^{n} p_i/2\), then the optimal solution given by McNaughton’s Warp-around algorithm [18] for makespan is also an optimal solution in the \(\ell_p\) norm due to the convexity of \(\ell_p\) norm. For this case, all the machine loads are equal to \(\sum_{i=1}^{n} p_i/2\), implying that \(\sqrt[2]{2(\sum_{i=1}^{n} p_i/2)}\) and

\[
\text{OPT}(2,J) = \left( \max \left\{ \frac{\sum_{i=1}^{n} p_i/2}{2} \right\} \right)^{p} + \left( \min \left\{ \frac{\sum_{i=1}^{n} p_i/2}{2} \right\} \right)^{p} = \sqrt[2]{2} \left( \sum_{i=1}^{n} p_i/2 \right).
\]
If \( p_1 > \sum_{i=1}^{n} p_i / 2 \), then in any optimal solution, \( p_1 \) must be assigned entirely to one machine and all remaining jobs are assigned to another machine, implying that

\[
\left( \frac{p_1^p}{\left( \sum_{i=2}^{n} p_i \right)^p} \right) = p^p \left( \max \left\{ p_1, \sum_{i=2}^{n} p_i \right\} \right)^p + \left( \min \left\{ p_1, \sum_{i=2}^{n} p_i \right\} \right)^p.
\]

Next suppose that the formula were true for \( m - 1 \) machines, we shall show that it is also true for \( m \) machines.

(i). If \( p_1 \leq \sum_{i=1}^{n} p_i / m \), then the optimal solution given by McNaughton for makespan is also the optimal solution in the \( l_p \) norm by the convexity of the objective function. For this case, all the machine loads are equal to \( \sum_{i=1}^{n} p_i / m \), implying that the objective value is \( \sum_{i=1}^{n} p_i / m \). Next, consider each term in the formula (1). It is easy to prove that \( \sum_{i=1}^{n} p_i / m \geq p_i, i = 1, 2, \ldots, n, \) and \( \sum_{i=1}^{n} p_i / m \leq \sum_{i=k}^{n} p_i / (m - k + 1) \), \( k = 1, 2, \ldots, m - 1 \) since \( p_1 \leq \sum_{i=1}^{n} p_i / m \) and \( p_1 \geq p_2 \geq \cdots \geq p_n \). So each term in (1) is equal to \( (\sum_{i=1}^{n} p_i / m)^p \), implying that (1) is just equal to \( \sqrt[m]{\sum_{i=1}^{n} p_i / m} \) and we are done.

(ii). If \( p_1 > \sum_{i=1}^{n} p_i / m \), then in any optimal solution, \( p_1 \) must be assigned entirely on one machine, and \( \text{opt}^b(J) = p_1^p + \text{opt}^b(m-1, J_1) \), where \( J_1 = J \setminus \{ p_1 \} \) and \( \text{opt}^b(m-1, J_1) \) is an optimal value for \( m - 1 \) machines. The rest of argument is similar to (i). Consider each term in the formula (1). The first term is \( p_1^p \), and deleting \( \sum_{i=1}^{n} p_i / m \) from each term results in the sum of the last \( m - 1 \) terms being equal to \( \text{opt}^b(m-1, J_1) \) due to \( \sum_{i=1}^{n} p_i / m \geq \sum_{i=k}^{n} p_i / (m - k + 1) \), \( k = 1, 2, \ldots, m - 1 \). We are done.

3 Lower bound

Consider the problem \( Pm|\text{pmtn},\text{online-list}|l_p \). Let \( J = (p_n, \ldots, p_1) \) be a sequence of jobs arriving in that order. Denote \( \text{opt}(J) \) as the optimal (off-line) objective value for input \( J \). For a given randomized algorithm \( A \), let \( A(J) \) be the objective value of the schedule generated on input \( J \) by \( A \). Algorithm \( A \) is \( \sigma \)-competitive if, \( E[A(J)] \leq \sigma \text{opt}(J) \) for any sequence \( J \), where \( E[A(J)] \) denotes the expected objective value of the schedule generated by \( A \). The best possible competitive ratio among all randomized algorithm will be denoted as \( R \) for the problem \( Pm|\text{pmtn},\text{online-list}|l_p \).

For any \( k = 1, \ldots, n \), we denote \( J_k = (p_n, \ldots, p_k) \). Let \( \text{opt}(J_k) \) be the optimal objective value of the instance of the problem \( Pm|\text{pmtn},\text{online-list}|l_p \) with job sequence \( J_k \). Let \( L = \sum_{i=1}^{n} p_i \).
Theorem 2 For any job sequence \( J_n = \{p_n, \ldots, p_1\} \) such that \( p_{m-1} \leq \ldots \leq p_1 \), the best possible competitive ratio \( R \) among all randomized algorithm must satisfy
\[
\sqrt[p]{L^p + (L - p_1)^p + (L - p_1 - p_2)^p + \cdots + \left( L - \sum_{i=1}^{m-1} p_i \right)^p} \leq R \sum_{k=1}^{m} \text{OPT}(J_k) \tag{2}
\]

Proof. Fix a sequence of random bits used by random algorithm \( A \). Since we are considering the best possible randomized algorithm, we can assume that there is no unenforced idle time before the last job in the schedule generated by \( A \). Suppose that \( L_i \) (\( i = 1, \ldots, m \)) is the load on machine \( i \) after all jobs are scheduled by algorithm \( A \), where \( L_i \leq L_{i+1} \). Let \( T_i (k = 1, \ldots, m) \) be the resultant objective value by algorithm \( A \) on job sequence \( J_i \).

First we claim that, for any \( i (i = 1, \ldots, m) \),
\[
\sqrt[p]{L_1^p + L_2^p + \cdots + L_{m-i}^p + (L_{m-i+1} - p_{i-1})^p + (L_{m-i+2} - p_{i-2})^p + \cdots + (L_m - p_1)^p} \leq T_i. \tag{3}
\]
Since the algorithm is on-line, the schedule for \( J_i \) is obtained from the schedule for \( J_n \) by removing the last \( i - 1 \) jobs. After all jobs are assigned, the machines loads are \( L_1 \leq L_2 \leq \cdots \leq L_m \). The last \( m - 1 \) jobs being nondecreasing and the objective function being convex imply that the objective value produced by the algorithm is at least \( L_1^p + L_2^p + \cdots + L_{m-i}^p + (L_{m-i+1} - p_{i-1})^p + (L_{m-i+2} - p_{i-2})^p + \cdots + (L_m - p_1)^p \), and hence (3) holds.

By (3), we have
\[
\sum_{i=1}^{m} \sqrt[p]{L_1^p + L_2^p + \cdots + L_{m-i}^p + (L_{m-i+1} - p_{i-1})^p + (L_{m-i+2} - p_{i-2})^p + \cdots + (L_m - p_1)^p} \leq \sum_{i=1}^{m} T_i.
\]
Using Minkowski’s inequality, we have
\[
\sum_{i=1}^{m} \sqrt[p]{L_1^p + L_2^p + \cdots + L_{m-i}^p + (L_{m-i+1} - p_{i-1})^p + (L_{m-i+2} - p_{i-2})^p + \cdots + (L_m - p_1)^p} \geq \sqrt[p]{L^p + (L - p_1)^p + (L - p_1 - p_2)^p + \cdots + (L - \sum_{i=1}^{m-1} p_i)^p}.
\]
Algorithm \( A \) being \( R \)-competitive implies that \( E[A(J)] \leq R \cdot \text{OPT}(J) \) and the theorem follows. \( \blacksquare \)

Theorem 2 implies that a lower bound on \( R \) can be obtained by formulating a mathematic program, which is explained below. Note that
\[
R \geq \frac{\sqrt[p]{L^p + (L - p_1)^p + (L - p_1 - p_2)^p + \cdots + (L - \sum_{i=1}^{m-1} p_i)^p}}{\sum_{i=1}^{m} \text{OPT}(J_i)}
\]
We can normalize the jobs such that \( \sum_{i=1}^{m} \text{OPT}(J_i) = 1 \), and therefore
\[
R^p \geq L^p + (L - p_1)^p + (L - p_1 - p_2)^p + \cdots + \left( L - \sum_{i=1}^{m-1} p_i \right)^p
\]
Now it is easy to given a mathematical program to compute the above lower bound. The program
has variables \( q_1, q_2, \ldots, q_m, O_1, O_2, \ldots, O_m \). Variable \( q_1 \) corresponds to the sum of all processing times
in \( J_m \), variables \( q_2, \ldots, q_m \) to the processing times of the last \( m - 1 \) jobs, and variables \( O_i \) correspond
to \( \text{OPT}(J_{m+i-1}) \).

**Definition 1** Let \( r \) be the value of the objective function of the optimal solution of the following
program:

\[
\begin{align*}
\max & \quad r^p = q_1^p + (q_1 + q_2)^p + \cdots + (q_1 + q_2 + \cdots + q_m)^p \\
\max & \quad \left\{ \frac{q_1 + q_2}{m}, q_2 \right\}^p + \min \left\{ \frac{q_1 + q_2}{m}, \frac{q_1}{m-1} \right\}^p (m-1) \leq O_2^p \\
\max & \quad \left\{ \frac{q_1 + q_2 + q_3}{m}, q_3 \right\}^p + \max \left\{ \min \left\{ \frac{q_1 + q_2 + q_3}{m}, \frac{q_1 + q_2}{m-1} \right\}, q_2 \right\}^p \\
& \quad + \min \left\{ \frac{q_1 + q_2 + q_3}{m}, \frac{q_1 + q_2}{m-1}, \frac{q_1}{m-2} \right\}^p (m-2) \leq O_3^p \\
& \quad \vdots \\
\max & \quad \left\{ \frac{q_1 + \cdots + q_m}{m}, q_m \right\}^p + \max \left\{ \min \left\{ \frac{q_1 + \cdots + q_m}{m}, \frac{q_1 + \cdots + q_{m-1}}{m-1} \right\}, q_{m-1} \right\}^p \\
& \quad + \cdots + \max \left\{ \min \left\{ \frac{q_1 + \cdots + q_m}{m}, \frac{q_1 + \cdots + q_{m-1}}{m-1}, \ldots, \frac{q_1 + q_2}{2} \right\}, q_2 \right\}^p \\
& \quad + \min \left\{ \frac{q_1 + \cdots + q_m}{m}, \frac{q_1 + \cdots + q_{m-1}}{m-1}, \ldots, \frac{q_1 + q_2}{2}, q_1 \right\}^p \leq O_m^p \\
& \quad O_1 + O_2 + \cdots + O_m = 1 \\
& \quad O_i \geq 0, i = 1, 2, \ldots, m \\
& \quad q_i \leq q_{i+1}, i = 2, 3, \ldots, m \\
& \quad q_i \geq 0, i = 1, 2, \ldots, m
\end{align*}
\]

The above program has a feasible solution with the only non-zero variable \( O_m = 1 \). It is also easy to
see that the objective function is bounded. The constraints imply that \( (q_1 + q_2 + \cdots + q_i)^p \leq m^{p-1}O_i^p \)
for each \( i \) and \( q_1^p + (q_1 + q_2)^p + \cdots + (q_1 + q_2 + \cdots + q_m)^p \leq m^{p-1} \sum_{i=1}^{m} O_i^p \leq m^{p-1}(O_1 + O_2 + \cdots + O_m)^p = m^{p-1} \). Thus the value \( r \) is well defined.

**Theorem 3** Any randomized on-line algorithm for \( m \) identical machines has competitive ratio at
least \( r \)
Proof. There exist optimal solutions $q_1^*, q_2^*, ..., q_m^*, O_1^*, O_2^*, ..., O_m^*$. We create instance $I$ as follows:

The first $m$ jobs have processing times $p_1 = p_2 = \cdots = p_m = q_1^*/m$. The remaining $m-1$ jobs have processing times $p_{m+1} = q_2^*, p_{m+2} = q_3^*, \cdots, p_{2m-1} = q_m^*$. By lemma 1,

$$\text{OPT}(I_{m-i+1}) = \max \left\{ \frac{p_1 + \cdots + p_{m+i}}{m}, \frac{p_1 + \cdots + p_{m+i-1}}{m-1} \right\}^p + \cdots + \max \left\{ \min \left\{ \frac{p_1 + \cdots + p_{m+i}}{m}, \frac{p_1 + \cdots + p_{m+i-1}}{m-1} \right\}^p, \frac{p_1 + \cdots + p_m}{m-i} \right\}^p + \min \left\{ \frac{p_1 + \cdots + p_{m+i}}{m}, \frac{p_1 + \cdots + p_m}{m-i} \right\}^p (m-i)$$

and the constraints of the program implies that $O_i^* \geq \text{OPT}(I_{m-i+1})$. Hence, by Theorem 1, we have

$$\frac{\sqrt{q_1^* p_1^p + q_2^* p_2^p + \cdots + q_m^* p_m^p}}{\sum_{i=1}^{m} C_{\text{OPT}}(I_{m-i+1})} \geq \frac{\sqrt{q_1^* p_1^p + q_2^* p_2^p + \cdots + q_m^* p_m^p}}{\sum_{i=1}^{m} O_i^*} = r$$

$\blacksquare$

4 Solutions for $m = 2$

The mathematical program in Definition 1 is non-convex for general $m$. Below we shall show that, when $m = 2$, we can decompose the original program into two convex sub-programs, and hence the best bound is obtained by $r = \max \{ r_1, r_2 \}$:

$$\max r_1^p = q_1^p + (q_1 + q_2)^p$$
$$q_1^p \leq 2^{p-1}O_1^p$$
$$(q_1 + q_2)^p \leq 2^{p-1}O_2^p$$
$$O_1 + O_2 = 1$$
$$q_1 \geq q_2$$
$$O_i \geq 0, \quad i = 1, 2.$$  

$$\max r_2^p = q_1^p + (q_1 + q_2)^p$$
$$q_1^p \leq 2^{p-1}O_1^p$$
$$q_1^p + q_2^p \leq O_2^p$$
$$O_1 + O_2 = 1$$
$$q_1 \leq q_2$$
$$O_i \geq 0, \quad i = 1, 2.$$  

Note that for each of the above two programs, the first two inequalities must be tight (satisfied as equality) in any optimal solutions, otherwise we can increase the objective value by either increasing $q_2$ when the second inequality is non-tight, or decreasing $O_1$, increasing $q_2$ and $O_2$ when the first inequality is non-tight. So, the above programs can be reformulate as follows.
\[
\begin{align*}
\max r_1^p &= q_1^p + (q_1 + q_2)^p \\
q_1^p &= 2^{p-1}O_1^p \\
(q_1 + q_2)^p &= 2^{p-1}O_2^p \\
O_1 + O_2 &= 1 \\
q_1 &\geq q_2 \\
O_i &\geq 0, \quad i = 1, 2. \\
q_i &\geq 0, \quad i = 1, 2.
\end{align*}
\]

For the first program, it is easy to see that the optimal solution is \( q_1 = 1, q_2 = 0, O_1 = O_2 = 1/2 \) and the objective value is 1. For the second program, letting \( t = O_1 \), leads to a two-stage problem

\[
\max_{t \in [0, 1]} f(t),
\]

where \( f(t) \) is the objective function of the following parameterized problem:

\[
\begin{align*}
f(t) &= \max q_1^p + (q_1 + q_2)^p \\
q_1^p &= 2^{p-1}t^p \\
q_1^p + q_2^p &= (1 - t)^p \\
q_1 &\leq q_2 \\
q_i &\geq 0, \quad i = 1, 2.
\end{align*}
\]

**Case 1:** \( q_1 = q_2 \). By direct computation, \( t = 1/3 \) and \( f(t) = (2^p + 1)2^{p-1}/3^p \).

**Case 2:** \( q_1 < q_2 \). Then \( q_1 = \frac{\sqrt[2p-1]{1-t} - 2^p t^p}{\sqrt[2p-1]{1-t} + 2^p t^p} \). So

\[
f(t) = 2^{p-1}t^p + (2^{p-1}t^p + \sqrt[2p-1]{1-t}^p - 2^{p-1}t^p)^p.
\]

Consider the equation \( f'(t) = 0 \); that is,

\[
(2t)^{p-1} \left( (1-t)^p - 2^{p-1}t^p \right) \left( 1 + \left( 2^{p-1}t^p + \sqrt[2p-1]{1-t}^p - 2^{p-1}t^p \right)^{p-1} \right) \]

\[
= \left( 2^{p-1}t^p + \sqrt[2p-1]{1-t}^p - 2^{p-1}t^p \right)^{p-1} \left( (1-t)^p - 2^{p-1}t^p - (2t)^p \right) \frac{\sqrt[2p-1]{1-t^p} - 2^{p-1}t^p}{\sqrt[2p-1]{1-t}^p - 2^{p-1}t^p}.
\]

The solution of the desired problem (4) is obtained by first calculating the roots of \( f'(t) = 0 \) and choose the best one.

As an example, when \( p = 2 \), the equation (5) is

\[
2(3t - 1) + \frac{2\sqrt{3}(1 - 3t - 2t^2)}{\sqrt{1 - 2t - t^2}} = 0,
\]

8
which can be simplified as follows:

\[ 17t^4 + 36t^3 - 10t^2 - 4t + 1 = 0, \]

resulting in four roots: \( t_1 = -0.3413, t_2 = 0.2535, t_3 = 0.2927, t_4 = -2.3226 \). Among which, only \( t_1 \) and \( t_2 \) are the solution to the equation (6), and \( t_2 \) is the maximizer of the problem (4).

For \( p = 2, 3, 4, 5 \), we obtain the following results. Compared with optimal competitive ratio from [9], we found that this bound is very close to the optimal bound and approaches the best bound when \( p \to \infty \).

\[
\begin{array}{|c|c|c|c|c|}
\hline
 & p = 2 & p = 3 & p = 4 & p = 5 \\
\hline
This paper & 1.07483 & 1.1267 & 1.16358 & 1.19055 \\
\hline
\end{array}
\]

Table 1: Results on lower bounds and optimal bounds.

From the above analysis, we know that the bound given by the above program is not tight in general, and hence revealing intricate and essential difference between the general \( \ell_p \) norm and the makespan, and moreover, leaving the obvious open question on finding the best possible lower bound for the \( \ell_p \) norm problem by either extending the existing lower bounding techniques or developing more powerful new methods.

References


