Lower Bounds for the Classical Scheduling Problem

Guochuan Zhang

Joint with Lin Chen (TU Berlin), Klaus Jansen (Kiel), and Deshi Ye (Zhejiang)

April 1, 2014, Aussois
The classical scheduling problem

\[ M_1: \]
\[ M_2: \]
\[ ... \]
\[ M_m: \]

Machines

Makespan
Identical machines $P//C_{\text{max}}$

Machine 1

Machine 2

Machine 3
Related machines $Q//C_{\text{max}}$

- Machine 1 (speed=1/2)
- Machine 2 (speed=1)
- Machine 3 (speed=2)
Unrelated machines $R//C_{max}$

Machine 1

Machine 2

Machine 3
Known results
Approximation results

- **LPT (Longest Procoessing Time):** $\frac{4}{3} - \frac{1}{3m}$ for P//Cmax

- **PTAS** for Q//Cmax

- **FPTAS** for Rm//Cmax

- $(2 - \frac{1}{m})$-approximation for R//Cmax

- $(\frac{3}{2} - \varepsilon)$- approximation for R//Cmax implies P=NP
Two topics

• Improving running time for approximation schemes

• Improving performance rations for R//Cmax
Lower bounds on running time
Approximation schemes for $P||C_{max}$

- $(n/\epsilon)^{O(1/\epsilon^2)}$ (Hochbaum and Shmoys, JACM 1987)

- $f(1/\epsilon) + O(n)$ doubly exponential in $1/\epsilon$ (Alon et al., SODA 1998)

- $2^{O(1/\epsilon^2 \log^3 1/\epsilon)} + n^{O(1)}$ (Jansen, SIAM J. Discrete Mathematics 2010)
Approximation schemes for $Pm\|C_{\text{max}}$

- $O(nm(nm/\epsilon)^{m-1})$ (Horowitz and Sahni, JACM 1976)

- $(n + 1)^{m/\epsilon} \text{poly}(|I|)$ (Lenstra, Shmoys, Tardos, Math. Prog 1990)

- $n(m/\epsilon)^{O(m)}$ (Jansen and Porkolab, MOR 2001)

- $O(n) + (1/\epsilon)^{O(m)}$ when $1/\epsilon > m$ (Jansen and Mastrolilli, 2010)
Approximation schemes for $Rm||C_{max}$

- $O(nm(nm/\epsilon)^{m-1})$ (Horowitz and Sahni, JACM 1976)
- $(n + 1)^{m/\epsilon} \text{poly}(|I|)$ (Lenstra, Shmoys, Tardos, Math. Prog 1990)
- $n(m/\epsilon)^{O(m)}$ (Jansen and Porkolab, MOR 2001)
- $O(n) + (1/\epsilon)^{O(m)}$ when $1/\epsilon > m$ (Jansen and Mastrolilli, 2010)
Our contribution

<table>
<thead>
<tr>
<th>Algorithms</th>
<th>Upper bounds</th>
<th>Lower bounds</th>
</tr>
</thead>
<tbody>
<tr>
<td>Approximation schemes</td>
<td>$2^O(\frac{1}{\epsilon^2 \log^3 \frac{1}{\epsilon}}) + n^O(1)$</td>
<td>$2^O((\frac{1}{\epsilon})^{1-\delta}) + n^O(1)$</td>
</tr>
<tr>
<td>Approximation schemes</td>
<td>$(\frac{1}{\epsilon})^O(m) + O(n)$</td>
<td>$(\frac{1}{\epsilon})^O(m^{1-\delta}) + n^O(1)$</td>
</tr>
<tr>
<td>Exact algorithm</td>
<td>$2^O(\sqrt{m</td>
<td>I</td>
</tr>
<tr>
<td>Exact algorithm</td>
<td>$2^O(n)$</td>
<td>$2^O(n^{1-\delta})$</td>
</tr>
</tbody>
</table>

Based on Exponential Time Hypothesis (ETH), assuming that 3SAT could not be solved in $2^{sn}$ time for some $s>0$ (Impagliazzo, Paturi, Zane, 2001)
A small gap for $P||C_{max}$

- $2^O(1/\varepsilon^2 \log^3 1/\varepsilon) + n^O(1)$ vs. $2^O((1/\varepsilon)^{1-\delta}) + n^O(1)$

There exists a $2^O(1/\varepsilon \log^2 1/\varepsilon) + n^O(1)$ - time algorithm,

1. if every machine can accommodate at most a constant number of jobs;

2. if a certain conjecture holds (Jansen and Robenek, WAOA 2011)
Reduction for $P||C_{max}$

- $2^{sn}$ for 3SAT $\rightarrow 2^{O((1/\epsilon)^{1-\delta})}$ for Scheduling

- 3SAT instance of $O(n)$ variables and clauses
  $\rightarrow$ A scheduling instance of makespan $L = O(n^k)$

Goal

A lower bound of $2^{O((1/\epsilon)^{1/k-\delta})}$ by taking $1/\epsilon = L + 1 = O(n^k)$

(A $(1+\epsilon)$-alg. is able to distinguish the makespan of $L$ and $L+1$)

$k = 16$ in the traditional reduction
Reduction for $Pm\|C_{max}$

- $2^{sn}$ for 3SAT $\rightarrow (1/\epsilon)^{O(m^{1-\delta})}$ for Scheduling

- 3SAT instance of $O(n)$ variables and clauses

- A scheduling instance of makespan $L = 2^{O\left(\frac{n}{m} \log^2 m\right)}$

Taking $1/\epsilon = L + 1$, a $(1 + \epsilon)$-algorithm is able to distinguish the makespan of $L$ and $L+1$, running in $2^{\delta m n}$ time with $\delta_m \rightarrow 0$ when $m \rightarrow \infty$
Open problems

• Close the gap between $2^O(1/\epsilon^2 \log^3 1/\epsilon) + n^O(1)$ and $2^O((1/\epsilon)^{1-\delta}) + n^O(1)$ for $P||C_{max}$

• Influence of additional constraints: $P|r_j|C_{max}$

• Lower bounds on running times of approximation schemes for other classical objectives: $P||\sum w_j C_j$, $Q||\sum w_j C_j$, $P||\sum C_i^p$
Lower bound for lower rank scheduling
Matrix of job processing times

\[ P = \begin{bmatrix}
p_{11} & p_{12} & \cdots & p_{1,m} \\
p_{21} & p_{22} & \cdots & p_{2,m} \\
\vdots & \vdots & \ddots & \vdots \\
p_{n,1} & p_{n,2} & \cdots & p_{n,m}
\end{bmatrix} \]
More precisely, the rank \(d\) comes from:

\[
\begin{bmatrix}
    p_{11} & p_{12} & \cdots & p_{1,m} \\
    p_{21} & p_{22} & \cdots & p_{2,m} \\
    \vdots & \vdots & \ddots & \vdots \\
    p_{n,1} & p_{n,2} & \cdots & p_{n,m}
\end{bmatrix}
\]

\[
= \begin{bmatrix}
    \ast & \ast \\
    \ast & \ast \\
    \vdots & \vdots \\
    \ast & \ast
\end{bmatrix}
\begin{bmatrix}
    \ast & \ast & \cdots & \ast
\end{bmatrix}
\]

\[
\text{n} \times \text{d} \quad \text{d} \times \text{m}
\]
Rank of scheduling

• Identical machines: Rank=1
• Related machines: Rank=1
• Unrelated machines: Rank could be arbitrary

The hardness of the problem and the rank of matrix?
<table>
<thead>
<tr>
<th>Rank of Matrix</th>
<th>Hardness</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>PTAS (Hauchbaum and Shmoys)</td>
</tr>
<tr>
<td>2</td>
<td>QPTAS</td>
</tr>
<tr>
<td>3</td>
<td>open</td>
</tr>
<tr>
<td>4</td>
<td>APX-hard</td>
</tr>
<tr>
<td>[\geq 7]</td>
<td>((3/2-\epsilon))-inapproximate</td>
</tr>
</tbody>
</table>
New lower bounds

<table>
<thead>
<tr>
<th>Rank of Matrix</th>
<th>Hardness</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>APX-hard</td>
</tr>
<tr>
<td>≥4</td>
<td>(3/2-(\epsilon))-inapproximate</td>
</tr>
</tbody>
</table>
Rank 4 scheduling
Reduction from 3DM

• **Element sets:** $W \cup X \cup Y$, $|W| = |X| = |Y|$  
• **Match set:** $T \subseteq W \times X \times Y$  
• **Perfect matching:** $T' \subseteq T$, *every element appears once*

NP-hard to decide whether there exists a perfect matching
Rank 7 scheduling

Bhaskara, et al. 2013

• One machine for every match:
  \((w_i, x_j, y_k) \rightarrow (N^i, N^{-i}, N^j, N^{-j}, N^k, N^{-k}, 1)\)

• Real jobs:
  \(x_j \rightarrow (0, 0, \epsilon N^{-j}, \epsilon N^j, 0, 0, 1)\)
  \(y_k \rightarrow (0, 0, 0, 0, \epsilon N^{-k}, \epsilon N^k, 1)\)

• Dummy jobs: \(d(w_i) - 1\) copies
  \(w_i \rightarrow (\epsilon N^{-i}, \epsilon N^i, 0, 0, 0, 0, 2)\)
Rank 7 scheduling

Bhaskara, et al. 2013

- The processing time of each real job is at least 1
- The processing time of a dummy job is at least 2
  -- Ensured by the 7-th coordinate
- A w-job has to be scheduled on a machine (match) containing element w
  -- Ensured by the first 2 coordinates
Rank 7 scheduling

Bhaskara, et al. 2013

A schedule of length $2 + O(\epsilon)$

$(w_1, *, *)$  $w_1 - \text{dummy}$

$(w_1, *, *)$  $w_1 - \text{dummy}$

...  $w_1 - \text{dummy}$

$(w_1, x_j, y_k)$  $x_j - \text{job}, y_k - \text{job}$

$(w_2, *, *)$  $w_2 - \text{dummy}$

...
Rank 4 scheduling

$LRS(7) \leftarrow (\ast, \ast, \ast, \ast, \ast, \ast)"

Let $w, x, y$ share coordinates

However, in the 3DM problem, for a match $(w_i, x_j, y_k)$, the indices $(i,j,k)$ could be arbitrary
A variant of 3DM

Reduction from a special 3DM (still NP-complete)

- Element sets: $W = \{w_i, \bar{w}_i | i \in [3n]\}$, $X = \{s_i, a_i | i \in [3n]\}$, $Y = \{s'_i, b_i | i \in [3n]\}$
- Match set: $T_1 \subseteq \{(w_i, s_j, s'_j), (\bar{w}_i, s_j, s'_j) | w_i \in W, s_j \in X, s'_j \in Y\}$
  $T_2 = \{(w_i, a_i, b_i), (\bar{w}_i, a_i, b_{\zeta(i)}) | i \in [3n]\}$

$\zeta: \zeta(3k + 1) = 3k + 2, \zeta(3k + 2) = 3k + 3, \zeta(3k + 3) = 3k + 1$
Tied indices

\[(w_i, s_j, s'_j), \quad (\bar{w}_i, s_j, s'_j), \quad (w_i, a_i, b_i), \quad (\bar{w}_i, a_i, b_{\zeta(i)})\]

Indices are tied
Overall structure

**Machines**

- \((w_i, s_j, s_j')\): \((N^i, N^{-i}, N^{j+N}, N^{-j-N})\)
- \((\bar{w}_i, s_j, s_j')\): \((N^{-i}, N^i, N^{j+N}, N^{-j-N})\)
- \((w_i, a_i, \ast)\): \((N^i, N^{-i}, N^{-i}, \ast)\)
- \((\bar{w}_i, a_i, \ast)\): \((N^{-i}, N^i, N^{-i}, \ast)\)

**Jobs**

- \(w_i\): \((N^{-i}, N^i, 0, 0)\)
- \(\bar{w}_i\): \((N^i, N^{-i}, 0, 0)\)
- \(a_i\): \((N^{-i}, N^{-i}, N^i, 0)/2\)
- \(s_j(s_j')\): \((0, 0, N^{-j-N}, N^{j+N})/2\)
Overall structure

**Machines**
- \((w_{3i}, a_{3i}, b_{3i}):\) \((N^{3i}, N^{-3i}, N^{-3i}, N^{-3i-1})\)
- \((w_{3i+1}, a_{3i+1}, b_{3i+1}):\) \((N^{3i+1}, N^{-3i-1}, N^{-3i-1}, N^{-3i})\)
- \((w_{3i+2}, a_{3i+2}, b_{3i+2}):\) \((N^{3i+2}, N^{-3i-2}, N^{-3i-2}, \frac{1}{\epsilon}N^{-3i-1})\)
- \((\bar{w}_{3i}, a_{3i}, b_{3i+1}):\) \((N^{-3i}, N^{3i}, \epsilon N^{-3i}, N^{-3i})\)
- \((\bar{w}_{3i+1}, a_{3i+1}, b_{3i+2}):\) \((N^{-3i-1}, N^{3i+1}, N^{-3i-1}, \frac{1}{\epsilon}N^{-3i-1})\)
- \((\bar{w}_{3i+2}, a_{3i+2}, b_{3i}):\) \((N^{-3i-2}, N^{3i+2}, \epsilon N^{-3i-2}, N^{-3i-1})\)

**Jobs**
- \(b_{3i}:\) \((\epsilon N^{-3i}, \epsilon N^{-3i-2}, N^{3i}, N^{3i+1})/2\)
- \(b_{3i+1}:\) \((\epsilon N^{-3i-1}, \frac{1}{\epsilon}N^{-3i-1}, N^{3i}, N^{3i})/2\)
- \(b_{3i+2}:\) \((\epsilon N^{-3i-2}, \epsilon N^{-3i-1}, N^{3i+1}, \epsilon N^{3i+1})/2\)
Gap

• If the instance of the special 3DM has a perfect matching, then there exists a schedule with a makespan of $2+O(\varepsilon)$

• If the scheduling instance has a feasible schedule of makespan smaller than 3, then the 3DM has a perfect matching

• This gives a lower bound of $3/2 - \varepsilon$
Open problems

. Approximation algorithms for scheduling problem of a constant rank.
. Is rank 2 scheduling harder than rank 1 (in terms of schemes)?
. Is rank 4 scheduling harder than rank 3 (in terms of approx ratios)?
Thank You!