Scheduling unreliable jobs on parallel machines

Alessandro Agnetis, Paolo Detti, Marco Pranzo
Università di Siena

Manbir S. Sodhi
University of Rhode Island

Assois, May 18-21, 2008
Overview

• The Unreliable Jobs Scheduling Problem
• Related problems and applications
• Complexity analysis
• Approximation analysis
• Computational results
Stochastic activities

• There are a number of activities (projects, jobs) to execute, and there is a probability \( p_i \) to successfully carry out activity \( i \)

• Upon failure of a job, the unit (\textit{machine}) currently in charge of its execution can’t perform any other activity
The problem

- The problem is to arrange the jobs in an order that maximizes an utility index, accounting for the possibility of machine breakdowns (due to job failure)
Application context: Unsupervised systems

- In certain highly automated systems, production (at least during the night shift) is unattended.
- When planning unattended activities, the chances of failures leading to machine breakdown should be taken into account.
- If a machine stops, no immediate intervention is possible.
The Unreliable Jobs Scheduling Problem (UJSP)

We are given:

- $J = \{1, \ldots, n\}$ job set
- a single machine or a set of $m$ identical parallel machines
- $p_i$ success probability of job $i$
- $r_i$ reward for job $i$, if completed
• Single machine case

Let $\sigma$ be a schedule of the $n$ jobs on the machine, and $\sigma(j)$ be the job in $j$-th position

the total expected reward is

$$ER(\sigma) = p_{\sigma(1)} r_{\sigma(1)} + p_{\sigma(1)} p_{\sigma(2)} r_{\sigma(2)} + \ldots + p_{\sigma(1)} p_{\sigma(2)} \ldots p_{\sigma(n-1)} p_{\sigma(n)} r_{\sigma(n)}$$
UJSP

- Parallel machine case

Let $\phi=\{\sigma_1, \ldots, \sigma_m\}$ be a schedule of the $n$ jobs on a
the $m$ machines, where $\sigma_k$ is the schedule of the jobs assigned to the $k$-th machine

the total expected reward is

$$ER(\phi) = ER(\sigma_1) + ER(\sigma_2) + \ldots + ER(\sigma_m)$$
A related problem

Total Weighted Discounted Completion Time (TWDCT)

\[ \text{P/} \sum w_i e^{-rC_i} \]

- \( n \) jobs must be scheduled on \( m \) parallel machines
- \( t_i \) processing time of job \( i \)
- \( w_i \) reward of job \( i \)
- \( r > 0 \) a fixed discount rate
- \( w_i e^{-rt} \) reward of job \( i \) if completed at time \( t \)
- \( C_i \) completion time of job \( i \) in some schedule \( \phi \)

The present value of the total reward (to maximize) is

\[ \text{PV}(\phi) = \sum w_i e^{-rC_i} \]
Equivalence between TWDCT and UJSP

Given an instance of TWDCT with \( n \) jobs and \( m \) parallel machines

we build an instance of UJSP with jobs \( n \) and \( m \) machines setting:

\[
p_i = e^{-r_i}
\]

\[
r_i = w_i
\]

The expected reward to maximize is

\[
ER(\phi) = \sum_{k=1}^{m} ER(\sigma_k) = PV(\phi)
\]
The Weighted Sum Completion Time (WSCT)

\[ P | \mid \sum_i w_i C_i \]

\( n \) jobs must be scheduled on \( m \) parallel machines
\( t_i \) processing time of job \( i \)
\( w_i \) weight of job \( i \)
\( C_i \) completion time of job \( i \) in some schedule \( \phi \)

The function to minimize is

\[ WC(\phi) = \sum_i w_i C_i \]
WSCT is a special case of TWDCT

\[
\left( P \mid \mid \sum_i w_i C_i \right) \quad \left( P \mid \mid \sum_i w_i e^{-rC_i} \right)
\]

When \( r \ll 1 \) we have

\[
e^{-rC_i} = 1 - rC_i + \frac{(rC_i)^2}{2!} - \frac{(rC_i)^3}{3!} + \frac{(rC_i)^4}{4!} + ... \approx 1 - rC_i
\]

Hence

\[
\max PV(\phi) \approx \max \sum_i w_i (1 - C_i) \equiv \sum_i w_i - \min \sum_i w_i C_i \equiv \min WC(\phi)
\]
**Single machine case**

**UJSP, TWDCT and WSCT problems can be optimally solved by the following ordering rules**

<table>
<thead>
<tr>
<th>Problem</th>
<th>Index</th>
<th>Ordering</th>
<th>Ref.</th>
</tr>
</thead>
<tbody>
<tr>
<td>UJSP</td>
<td>$Z_i = \frac{p_i r_i}{1 - p_i}$</td>
<td>nonincreasing</td>
<td>Mitten 1960</td>
</tr>
<tr>
<td>TWDCT</td>
<td>$w_i \frac{1}{1 - e^{-rt_i}}$</td>
<td>nondecreasing</td>
<td>Rothkopf 1966</td>
</tr>
<tr>
<td>WSCT</td>
<td>$\frac{w_i}{t_i}$</td>
<td>nonincreasing</td>
<td>Smith 1956</td>
</tr>
</tbody>
</table>
Related problems and application contexts (single machine case)

• Component testing (Monma and Sidney, 1979)
• Data acquisition and processing problems in sensor networks (Srivastava et al., 2005).
• Management of queries in databases (Hellerstein and Stonebraker, 1993)
UJSP with identical parallel machines

UJSP with 2 parallel machine is strongly NP-hard

Approximation results for two simple heuristics when \( m=2 \):

- Round Robin heuristic (RR)
- “Highest probability” heuristic (HP)
Round robin heuristic (RR)

- Order the jobs according to the ratio: \( Z_i = \frac{p_i r_i}{1 - p_i} \)

- Assign jobs to the machines in a round robin way: To machine \( h \) are assigned jobs \( i < n \)

\[
i = m \times k + h \quad \text{for} \quad k = 0, 1, \ldots, \left\lfloor \frac{n}{m} \right\rfloor
\]
Highest probability heuristic (HP)

- Order the jobs according to the ratio $Z_i = \frac{p_i r_i}{1 - p_i}$ if jobs have the same $Z$-ratio sequence first the job with the smallest success probability

- Assign the next job to the machine having the highest cumulative probability, i.e., the highest product of the probabilities of the jobs already assigned to it.
HP heuristic

jobs

machines

\[
\begin{align*}
p_1 &= 0.9 & r_1 &= 20 \\
p_2 &= 0.8 & r_2 &= 10 \\
p_3 &= 0.7 & r_3 &= 15 \\
p_4 &= 0.9 & r_4 &= 30 \\
p_5 &= 0.8 & r_5 &= 40 \\
\end{align*}
\]

\[
\begin{align*}
Z_{t_1} &= 180 & Z_{t_2} &= 180 & Z_3 &= 160 \\
Z_{t_2} &= 270 & Z_3 &= 160
\end{align*}
\]

Cumulative Probability

- \( \text{M}_1 \) 0.881
- \( \text{M}_2 \) 0.8
- \( \text{M}_3 \) 0.7

- 4
- 2
- 3
- 1
- 5
An approximation result

In problem **UJSP** with \( m \) parallel machines, any schedule \( \phi \) in which the jobs are sequenced according to the ratios on each machine is at least \( \frac{1}{m} \)-approximate.

\[
Z_i = \frac{p_i r_i}{1 - p_i}
\]

\[
\frac{ER(\phi)}{ER^*} \geq \frac{1}{m}
\]
How bad is RR heuristic?

A 3-job 2-machine instance

<table>
<thead>
<tr>
<th></th>
<th>$p_i$</th>
<th>$r_i$</th>
<th>$Z_i$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>$\varepsilon$</td>
<td>$1/\varepsilon$</td>
<td>$1/(1-\varepsilon)$</td>
</tr>
<tr>
<td>2</td>
<td>$1-\varepsilon$</td>
<td>$\varepsilon/(1-\varepsilon)$</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>$\varepsilon$</td>
<td>$(1-\varepsilon)/\varepsilon$</td>
<td>1</td>
</tr>
</tbody>
</table>
Round robin solution

<table>
<thead>
<tr>
<th>1</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td></td>
</tr>
</tbody>
</table>

\[
ER(\sigma_1^{RR}) = p_1 r_1 + p_1 p_3 r_3 = 1 + \varepsilon (1 - \varepsilon) = 1 + \varepsilon - \varepsilon \\
ER(\sigma_2^{RR}) = p_2 r_2 = \varepsilon \\
\]

\[
ER^{RR} = ER(\sigma_1^{RR}) + ER(\sigma_2^{RR}) = 1 + 2\varepsilon - \varepsilon^2
\]
Optimal solution

\[ ER(\sigma_1^*) = p_1 r_1 = 1 \]

\[ ER(\sigma_2^*) = p_2 r_2 + p_2 p_3 r_3 = \varepsilon + (1 - \varepsilon)^2 = \]

\[ = \varepsilon + 1 + \varepsilon^2 - 2\varepsilon \]

\[ ER^* = ER(\sigma_1^*) + ER(\sigma_2^*) = 2 - \varepsilon + \varepsilon^2 \]
Approximation ratio

\[ \frac{ER^{RR}}{ER^*} = \frac{1 + 2\varepsilon - \varepsilon^2}{2 - \varepsilon + \varepsilon^2} \rightarrow 0.5 \]

**RR** is 1/2-approximate.
How bad is HP heuristic?

When $m=2$ HP is 0.81-approximate

$$\frac{ER_{HP}}{ER^*} \geq 0.81$$

A first (high multiplicity) upper bound for UJSP

- A special case is when there exist several identical jobs of few different types
- If exactly $m$ copies exist of each job, then the optimal solution consists in assigning one job of each type on each of the $m$ machines, and sequencing according to the $Z$-ratio
A first (high multiplicity) upper bound for UJSP

- The HM case can be exploited to devise an upper bound for the general case
- Given an instance of UJSP, replace each job \( i \) with \( m \) identical jobs \( k \):

\[
p_k = m\sqrt{p_i} \quad \quad \quad r_k = \frac{(1 - m\sqrt{p_i})p_i}{(1 - p_i)^m\sqrt{p_i}}r_i
\]

\[
UB_{HM} = m\sum_{i=1}^{n} \frac{1 - m\sqrt{p_i}}{1 - p_i} p_i r_i \sqrt{i-1} \prod_{k=1}^{i} p_k
\]
A first lower bound for UJSP

- The HP heuristic schedules at each step a job on the machine having the maximum cumulative probability. It can be proved that in HP the contribution of the $i$-th job is at least

$$p_i r_i^m \sqrt{\prod_{k=1}^{i-1} p_k}$$

- A lower bound for HP solution is then

$$LB_1 = \sum_{i=1}^{n} p_i r_i^m \sqrt{\prod_{k=1}^{i-1} p_k}$$
A first ratio

• W.l.o.g. let 1 be the job with the smallest probability \( p_1 \), we have

\[
\frac{ER_{HP}}{ER^*} \geq \frac{LB_1}{UB_{HM}} = \frac{\sum_{i=1}^{n} p_i r_i m \prod_{k=1}^{i-1} p_k}{m \sum_{i=1}^{n} \frac{1 - m \sqrt{p_i}}{1 - p_i} p_i r_i m \prod_{k=1}^{i-1} p_k} \geq \frac{1}{m(1 - m \sqrt{p_i})} \cdot \frac{1}{1 - p_1}
\]
New upper and lower bounds when all jobs have the same $Z$-ratio and $m=2$

- Consider a UJSP instance in which all jobs have the same ratio $Z$
- Let $S_k$ be the set of jobs scheduled on machine $k$
- The expected reward on machine $k$ is then

$$ER_k = Z(1 - \prod_{i \in S_k} p_i)$$
Expected reward of a schedule \( (m=2) \)
(all jobs have the same ratio \( Z \))

- Given a schedule \( \phi=\{\sigma_1, \sigma_2\} \) for \( m=2 \), w.l.o.g. suppose that job 1 (with the smallest probability) is assigned to machine 1.
- Let \( p_1 P_A \) and \( P_B \) be the cumulative probabilities of jobs assigned to machine 1 and 2, respectively.
- If all jobs have the same ratio \( Z \), the total expected reward of \( \phi \) is

\[
ER(\phi) = Z(1-p_1 P_A) + Z(1-P_B)
\]
A second lower bound \((m=2)\) (all jobs have the same ratio \(Z\))

- Consider the schedule \(\phi' = \{\sigma_1, \sigma_2\}\) in which job 1 is assigned to machine 1 and all other jobs are assigned to machine 2.
- Let \(P'_B\) the product of the probabilities of all other jobs (assigned to machine 2). Hence, \(P'_B = P_A P_B\)
- Note that the HP heuristic produces a schedule not worse than \(\phi'\) (\(\phi'\) provides a lower bound to HP sol.)
- If all jobs have the same ratio \(Z\), a lower bound to the solution provided by HP is

\[ LB_2 = Z(1 - p_1) + Z(1 - P'_B) \]
A second ratio \( m=2 \)

- By definition \( P'_B=P_A P_B \), hence we have:

\[
\frac{ER_{HP}}{ER^*} \geq \frac{LB_2}{ER(\phi)} \geq \frac{Z(1-p_1)+Z(1-P_A P_B)}{Z(1-p_1 P_A)+Z(1-P_B)}
\]

- Which is minimized when \( P_A=P_B=0 \)

- Hence

\[
\frac{ER_{HP}}{ER^*} \geq \frac{LB2}{ER(\phi)} \geq 1-p_1/2
\]

This result holds even when jobs have different \( Z \)-values
An approximation result for HP when m=2

- The minimum of maximum between

\[
\frac{ER_{HP}}{ER^*} \geq \frac{LB1}{UB_{HM}} \geq \frac{1 - p_1}{2(1 - \sqrt{p_1})}
\]

and

\[
\frac{ER_{HP}}{ER^*} \geq \frac{LB2}{ER^*} \geq 1 - p_1 / 2
\]

is about 0.81 (for \(p_1=0.37\))

HP is 0.81-approximate
Experimental results

- $n = 50, 100, 500$
- $m = 2, 5, 10, 20$
- $p_i \sim U[0.7, 0.99]$
- $p_i \sim U[0.3, 0.99]$
- $p_i \sim U[0.3, 0.7]$
- $r_i \sim U[10, 40]$
- 100 randomly generated instances for each setting
Average gap of RR  \[
100 \times \frac{UB_{HM} - ER_{RR}}{UB_{HM}}
\]

<table>
<thead>
<tr>
<th>Machines</th>
<th>(U[0.7, 0.99])</th>
<th>(U[0.3, 0.99])</th>
<th>(U[0.3, 0.7])</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>(n=50)</td>
<td>(n=100)</td>
<td>(n=500)</td>
</tr>
<tr>
<td>2</td>
<td>0.126</td>
<td>0.072</td>
<td>0.027</td>
</tr>
<tr>
<td>5</td>
<td>0.428</td>
<td>0.235</td>
<td>0.074</td>
</tr>
<tr>
<td>10</td>
<td>0.760</td>
<td>0.457</td>
<td>0.123</td>
</tr>
<tr>
<td>20</td>
<td>1.362</td>
<td>0.801</td>
<td>0.215</td>
</tr>
</tbody>
</table>
**Average gap of HP**

\[ \text{Average gap of HP} = 100 \times \frac{UB_{HM} - ER_{HP}}{UB_{HM}} \]

<table>
<thead>
<tr>
<th>Machines</th>
<th>jobs</th>
<th>( U[0.7,0.99] )</th>
<th>( U[0.3,0.99] )</th>
<th>( U[0.3,0.7] )</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>n=50</td>
<td>n=100</td>
<td>n=500</td>
<td>n=50</td>
</tr>
<tr>
<td>2</td>
<td>0.069</td>
<td>0.034</td>
<td>0.007</td>
<td>0.162</td>
</tr>
<tr>
<td>5</td>
<td>0.251</td>
<td>0.114</td>
<td>0.022</td>
<td>0.557</td>
</tr>
<tr>
<td>10</td>
<td>0.547</td>
<td>0.256</td>
<td>0.045</td>
<td>1.272</td>
</tr>
<tr>
<td>20</td>
<td>1.133</td>
<td>0.552</td>
<td>0.091</td>
<td>2.904</td>
</tr>
</tbody>
</table>