NP-complete Partitioning Problems

Subset Sum: Given a list of \( t \) positive integers \( S = \{x_1, x_2, \ldots, x_t\} \) and an integer \( B \), is there a subset \( S' \subseteq S \) s.t. \( \sum_{x_i \in S'} x_i = B \).

- Yes instance: \( S = \{1, 2, 5, 7, 8, 10, 11\} \), \( B = 22 \).
- No instance: \( S = \{4, 10, 11, 12, 15\} \), \( B = 28 \).

Note: It is still NP-complete if \( B = \sum_i x_i / 2 \).

3-Partition: Given a list of \( 3t \) positive integers \( S = \{x_1, x_2, \ldots, x_{3t}\} \) with \( \sum_{x_i \in S} x_i = tB \), and each \( x_i \) satisfying \( B/4 < x_i < B/2 \), can you partition \( S \) into \( t \) groups of size 3, such that each group sums to exactly \( B \).

- Yes instance: \( S = \{26, 26, 27, 28, 29, 29, 31, 33, 39, 40, 45, 47\} \)
- No instance: \( S = \{26, 26, 27, 28, 29, 29, 31, 33, 38, 40, 45, 48\} \) (I think)

4 groups of 100
Problem: Given \( n \) jobs with processing times \( p_j \), schedule them on \( m \) machines so as to minimize the makespan.

Decision version: Given \( n \) jobs with processing times \( p_j \) and a number \( D \), can you schedule them on \( m \) machines so as to complete by time \( D \).

Sample inputs:
- Jobs are \( \{1, 2, 5, 7, 8, 10, 11\} \), 2 machines, \( D = 22 \).
- Jobs are \( S = \{4, 4, 10, 11, 12, 15\} \), 3 machines \( D = 20 \).

Reduction: Subset sum reduces to \( P||C_{\text{max}} \).

Idea of reduction: Given a subset sum instance, create a 2-machine instance of \( P||C_{\text{max}} \), with \( p_j = x_j \) and \( D = B \). Now there is a feasible schedule iff there is a subset summing to \( B \).
Given a input to subset sum \( S = \{x_1, \ldots, x_n\} \)

\( B \), with \( B = \sum x_i \)

- Form an input to \( \text{Pll}\text{Cmax} \) with
  - \( n \) jobs, \( P_i = x_i \), 2 machines \( D = B \).
  - Solve \( \text{Pll}\text{Cmax} \) output yes/no

Show Subset Sum outputs yes

(\( \Rightarrow \)) \( \text{Pll}\text{Cmax} \) outputs yes

PF \( \Rightarrow \) If subset:sum:yes, then there are two subsets of jobs \( S_1, S_2 \) each summing to \( B \), \( \sum \) the jobs on each machine sum to \( B = D \), so the answer is yes.
If $\text{AllCmax}$ is yes, then $C_{\text{max}} \leq D$, but $E_P = 2D$, so $C_{\text{max}} \neq D$, so the schedule gives 2 sets of jobs, each of total size D, therefore it gives a partition of $S$ into 2 sets of size $\frac{D}{2}$.

Also, the reduction is just copying the input to polynomial time.
Reduction: Reduce 3-partition to $1|r_j|L_{\max}$.

3-Partition Given a list of $3t$ positive integers $S = \{x_1, x_2, \ldots, x_{3t}\}$ with $\sum_{x_i \in S} x_i = tB$, can you partition $S$ into $t$ groups of size 3, such that each group sums to exactly $B$.

Given a 3-partition instance, we will create a $1|r_j|L_{\max}$ instance in the following way:

Jobs: $n = 4t - 1$ jobs, $t - 1$ of which are dummy jobs

<table>
<thead>
<tr>
<th>$j$</th>
<th>$r_j$</th>
<th>$p_j$</th>
<th>$d_j$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>B</td>
<td>1</td>
<td>B+1</td>
</tr>
<tr>
<td>2</td>
<td>2B +1</td>
<td>1</td>
<td>2B+2</td>
</tr>
<tr>
<td>3</td>
<td>3B + 2</td>
<td>1</td>
<td>3B+3</td>
</tr>
<tr>
<td>⋮</td>
<td>⋮</td>
<td>⋮</td>
<td>⋮</td>
</tr>
<tr>
<td>$t-1$</td>
<td>$(t-1)B + (t-2)$</td>
<td>1</td>
<td>$(t-1)B + (t-1)$</td>
</tr>
</tbody>
</table>

Dummy Jobs:

Real Jobs:
- indexed $t$ through $4t - 1$
- All have $r_j = 0$
• All have  \[ d_j = tb + (t - 1) \]
• \[ p_j = x_{j-(t-1)} \]
3 partition yes \( \Rightarrow \ \lambda_{\text{max}} \geq 0 \)
Reduction so that all jobs meet their deadlines iff 3-partition has a solution
(12 jobs 4 groups of 3 each sum to 100)

\[
\begin{align*}
&c=100 \quad \phi_1 = 1 \quad d_1 = 101 \\
&g=201 \quad \phi_2 = 1 \quad d_2 = 202
\end{align*}
\]
- All have $d_j = tb + (t - 1)$
- $p_j = x_{j-(t-1)}$

<table>
<thead>
<tr>
<th>$j$</th>
<th>$f_j$</th>
<th>$x_j$</th>
<th>$p_j$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>100</td>
<td>20</td>
<td>10</td>
</tr>
<tr>
<td>2</td>
<td>20</td>
<td>302</td>
<td>202</td>
</tr>
<tr>
<td>3</td>
<td>302</td>
<td>0</td>
<td>303</td>
</tr>
<tr>
<td>4</td>
<td>0</td>
<td>0</td>
<td>403</td>
</tr>
<tr>
<td>5</td>
<td>0</td>
<td>40</td>
<td></td>
</tr>
</tbody>
</table>
• All have \( d_j = tb + (t - 1) \)
• \( p_j = x_{j-(t-1)} \)

Show

3-petition is yes

(\( \max = 0 \))

\[
\Rightarrow \text{if 3-petition is yes, schedule follows the 3-petition policy.}
\]

\[
\Leftarrow \text{if 3 petition is no, then one group is } B \text{ and that forces C to miss its deadline.}
\]
- All have $d_j = tb + (t - 1)$
- $p_j = x_{j-(t-1)}$
Proof

<table>
<thead>
<tr>
<th></th>
<th>$r_j$</th>
<th>$p_j$</th>
<th>$d_j$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>B</td>
<td>1</td>
<td>B+1</td>
</tr>
<tr>
<td>2</td>
<td>2B+1</td>
<td>1</td>
<td>2B+2</td>
</tr>
<tr>
<td>3</td>
<td>3B+2</td>
<td>1</td>
<td>3B+3</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>$t-1$</td>
<td>$(t-1)B+(t-2)$</td>
<td>1</td>
<td>$(t-1)B+(t-1)$</td>
</tr>
</tbody>
</table>

**Dummy Jobs:**

- Indexed $t$ through $4t-1$.
- All have $r_j = 0$.
- All have $d_j = tb + (t-1)$.
- $p_j = x_{j-(t-1)}$

**Real Jobs:**

- Indexed $t$ through $4t-1$.
- All have $r_j = 0$.
- All have $d_j = tb + (t-1)$.
- $p_j = x_{j-(t-1)}$

**Idea of Proof:** Argue that there is a schedule with $L_{\text{max}} = 0$ iff the partition instance is yes.