1. Minimising maximum lateness

In this section we develop an algorithm that finds an optimal schedule for problem 1||Lmax, where

\[ L_{\text{max}} = \max \{ L_j = C_j - d_j \mid j=1,\ldots,n \} \]

is the maximum lateness.

Assume that the jobs are numbered in non-decreasing order of their due dates \( d_j \). In order to minimize the maximum lateness it is quite natural to process the jobs in this order. This dispatching rule is known as EDD (Earliest Due Date), sometimes it is called Jackson’s rule due to R. Jackson who studied it in 1954.

We can prove

**Theorem 1.** For 1|Lmax the EDD-rule is optimal.

**Proof** (adjacent pairwise interchange argument)

Suppose a schedule \( S \), which violates EDD, is optimal. In this schedule there must be at least two adjacent jobs \( i \) and \( k \) such that \( d_i > d_k \) and job \( i \) precedes job \( k \).

Swapping jobs \( i \) and \( k \) leads to a schedule \( S' \) such that

\[
\begin{align*}
L'_k &= C'_k - d_k < C_k - d_k \\
L'_i &= C'_i - d_i = C_k - d_i < C_k - d_k \\
L'_j &= C'_j - d_j = C_j - d_j = L_j \quad \text{for } j \neq i,k
\end{align*}
\]

which implies that \( L_{\text{max}}(S') = \max \{ L_i, L_k, \max_{j \neq i,k} \{ L_j \} \} \leq \max \{ L_i, L_k, \max_{j \neq i,k} \{ L_j \} \} = L_{\text{max}}(S) \).

If \( L_{\text{max}}(S') < L_{\text{max}}(S) \), then we have obtained a contradiction to the assumption that \( S \) is optimal.

If \( L_{\text{max}}(S') = L_{\text{max}}(S) \), then schedule \( S \) can be modified without increasing the objective function value so that jobs \( i \) and \( k \) do not violate EDD.

EDD rule can be implemented in …………………. time.
2. Minimising the number of late jobs: $1||\Sigma U_j$

In this section we develop an algorithm that finds an optimal schedule for problem $1||\Sigma U_j$, where $U_j$ is a unit penalty for completing job $j$ after its due date:

$$U_j = \begin{cases} 0, & \text{if } C_j \leq d_j \\ 1, & \text{otherwise.} \end{cases}$$

Consider the following example.

<table>
<thead>
<tr>
<th>Job</th>
<th>$p_j$</th>
<th>$d_j$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>7</td>
<td>9</td>
</tr>
<tr>
<td>2</td>
<td>8</td>
<td>17</td>
</tr>
<tr>
<td>3</td>
<td>4</td>
<td>18</td>
</tr>
<tr>
<td>4</td>
<td>6</td>
<td>19</td>
</tr>
<tr>
<td>5</td>
<td>6</td>
<td>20</td>
</tr>
</tbody>
</table>

The jobs are in the EDD order. The first two jobs can be scheduled on-time while job 3 is late.

The optimal job sequence is (___________), where the first ............ jobs are on-time and the last ............ jobs are late.

The value of the objective function for this schedule is

$$\Sigma U_j = ........$$
The algorithm that solves problem $1\|\Sigma j$ is known as Moore’s algorithm, due to J.M. Moore who designed it in 1968.

**Moore’s algorithm.**

- The algorithm repeatedly adds jobs in the EDD order to the end of a partial schedule of on-time jobs.
- If the addition of job $j$ results in this job being completed after its due date $d_j$, then a job in the partial schedule with the largest processing time is removed and declared late.
- All late jobs are scheduled in an arbitrary order after on-time jobs.

The correctness of the algorithm can be proved by induction. By removing the job with the largest processing time, we guarantee that the total processing time of on-time jobs is as small as possible.

The time complexity of Moore’s algorithm is ....................