Candidatos a Solución Greedy en el Scheduling Problem con Minimum Maximum Lateness

En este problema las tareas están representadas por dos datos: la duración l_k y la deadline d_k.

Candidato 1

Primero se considera el escenario en que la deadline es la misma para todas las tareas.

Considerando la definición de lateness:

$$\lambda_k(\sigma) = max\{0, C_k(\sigma) - d_k\}$$

Si la deadline es un valor constante para todas las tareas:

$$\lambda_k(\sigma) = max\{0, C_k(\sigma) - d\}$$

Bajo este escenario, dado que se le restaría lo mismo a todos los costos, lo que se vuelve más importante es que cada tarea espere la menor cantidad de tiempo posible para poder completarse.

La mejor manera de lograr lo anterior es ordenar las tareas ascendentemente por duración, por lo que el **Candidato a Criterio de Selección #1** es simplemente:

$$l_k \quad \text{, en orden ascendente}$$

Candidato 2

Ahora se considera el escenario en que todas las duraciones son iguales.

Considerando la definición de lateness:

$$\lambda_k(\sigma) = max\{0, C_k(\sigma) - d_k\}$$

Si la duración es igual a un valor l para todas las tareas, entonces los costos son múltiplos de este valor l:

$$\lambda_k(\sigma) = max\{0, kl - d_k\}$$

Bajo este escenario, lo que se vuelve más importante es que los sustraendos vayan creciendo junto con los costos, para que el resultado de la resta sea lo más pequeño posible.

La mejor manera de lograr lo anterior es ordenar las tareas ascendentemente por deadline, por lo que el **Candidato a Criterio de Selección #2** es simplemente:

$$d_k \quad \text{, en orden ascendente}$$

En este caso no existe una expresión que considere ambos datos y que cumpla las dos observaciones, es decir, que provea mayor "puntaje" tanto al disminuir la duración como al disminuir la deadline, por lo que no habría más candidatos siguiendo esta línea de análisis.

Sin embargo, cuando esto ocurre es conveniente considerar criterios adicionales que pueden tener sentido en el contexto del problema, y en este escenario se tiene al menos un criterio más que se puede identificar.

Candidato 3

Definición: Llamaremos "*slack*" o "*tiempo de sobra*" a la diferencia entre la deadline de una tarea y su tiempo de duración.

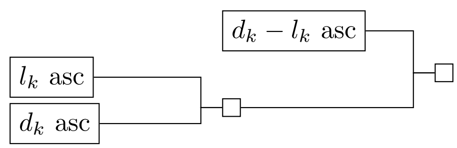
$$slack := d_k - l_k$$

Y se puede justificar que lo conveniente es entonces priorizar las tareas con el menor tiempo de sobra, las "más urgentes". Por lo que el **Candidato a Criterio de Selección #3**:

$$d_k - l_k \quad \text{, en orden ascendente}$$

VS: Combate entre los candidatos para determinar cuál es mejor

Dado que tenemos 3 candidatos, podemos organizar un pequeño torneo:



l_k VS d_k

Necesitamos definir dos tareas tal que, al utilizar l_k como criterio de selección gane una de ellas, y que al utilizar d_k como criterio de selección gane la otra:

$$T_1 = (1, 5)$$
$$T_2 = (4, 2)$$

De acuerdo a ambos criterios de selección, las agendas correspondientes serían:

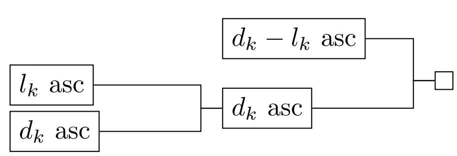
$$\sigma_{l_k} = (1, 5), (4, 2)$$
$$\sigma_{d_k} = (4, 2), (1, 5)$$

Calculemos la Tardanza Máxima de ambas agendas:

$$\begin{aligned} max\{\lambda_1(\sigma_{l_k}), \lambda_2(\sigma_{l_k})\} &= max\{max\{0, C_1(\sigma_{l_k}) - d_1\}, max\{0, C_2(\sigma_{l_k}) - d_2\}\} \\ &= max\{max\{0, 1 - 5\}, max\{0, (1 + 4) - 2\}\} \\ &= max\{max\{0, -4\}, max\{0, 3\}\} = max\{0, 3\} = 3 \end{aligned}$$

$$\begin{aligned} max\{\lambda_1(\sigma_{d_k}), \lambda_2(\sigma_{d_k})\} &= max\{max\{0, C_1(\sigma_{d_k}) - d_1\}, max\{0, C_2(\sigma_{d_k}) - d_2\}\} \\ &= max\{max\{0, 4 - 2\}, max\{0, (4 + 1) - 5\}\} \\ &= max\{max\{0, 2\}, max\{0, 0\}\} = max\{2, 0\} = 2 \end{aligned}$$

La Tardanza Máxima Mínima es 2, por lo que tenemos un ganador:



$d_k - l_k$ **VS** d_k

Nuevamente, primero definimos dos tareas tal que, al utilizar $d_k - l_k$ como criterio de selección gane una de ellas, y que al utilizar d_k como criterio de selección gane la otra:

$$
\begin{aligned}
T_1 &= (1,4) \\
T_2 &= (4,6)
\end{aligned}
$$

De acuerdo a ambos criterios de selección, las agendas correspondientes serían:

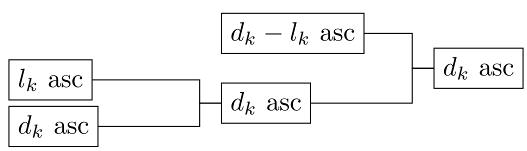
$$
\begin{aligned}
\sigma_{d_k - l_k} &= (4,6),(1,4) \\
\sigma_{d_k} &= (1,4),(4,6)
\end{aligned}
$$

Calculemos la Tardanza Máxima de ambas agendas:

$$
\begin{aligned}
max\{\lambda_1(\sigma_{d_k-l_k}), \lambda_2(\sigma_{d_k-l_k})\} &= max\{max\{0, C_1(\sigma_{d_k-l_k}) - d_1\}, max\{0, C_2(\sigma_{d_k-l_k}) - d_2\}\} \\
&= max\{max\{0, 4-6\}, max\{0, (4+1)-4\}\} \\
&= max\{max\{0,-2\}, max\{0,1\}\} = max\{0,1\} = 1
\end{aligned}
$$

$$
\begin{aligned}
max\{\lambda_1(\sigma_{d_k}), \lambda_2(\sigma_{d_k})\} &= max\{max\{0, C_1(\sigma_{d_k}) - d_1\}, max\{0, C_2(\sigma_{d_k}) - d_2\}\} \\
&= max\{max\{0, 1-4\}, max\{0, (1+4)-6\}\} \\
&= max\{max\{0,-3\}, max\{0,-1\}\} = max\{0,0\} = 0
\end{aligned}
$$

La Tardanza Máxima Mínima es 0, por lo que tenemos un ganador:



Por lo tanto, la **Solución Greedy** utilizará como criterio d_k en orden ascendente.