P1 Modelar $1||\sum C$

Sea $I = \{1...N\}$ el conjunto de trabajos

Definamos la variable

$$y_{ij} \begin{cases} 1 \text{si trabajo } i \text{ se hace en la posicion } j \\ 0 \text{sino} \end{cases}$$

$$\left[\begin{array}{l} \text{En la clase se definió otra variable:} \\ \\ x_{ij} \begin{cases} 1\text{si trabajo } i \text{ se hace antes que el trabajo } j \\ 0\text{sino} \end{cases} \\ \\ \text{eso no esta correcto} \end{array} \right]$$

Las restricciones son

- El trabajo debe hacerse de una (en una sola posicion)

$$\sum_j y_{ij} = 1$$

- En cada posicion, solo puede asignarse un trabajo

$$\sum_i y_{ij} = 1$$

Si el tiempos de procesamiento del trabajo i lo notamos por p_i
el tiempo de termino del trabajo en la k-esima posicion es

$$\sum_{j=1}^{k} \sum_{i=1}^{N} p_i y_{ij}$$

entonces la funcion objetivo es (el promedio de todos los tiempos de termino)

$$\sum_{k=1}^{N} \sum_{j=1}^{k} \sum_{i=1}^{N} p_i y_{ij}$$

Si resolvemos en AMPL, el problema para $p = (7, 8, 9, 10, 6.9, 8.1, 5, 6, 7.2, 11)$

```
data;
param N:=10;
param p:= 1 7 2 8 3 9 4 10 5 6.9 6 8.1 7 5 8 6 9 7.2 10 11;
```

obtenemos el siguiente orden de ejecución: 7,8,5,1,9,2,6,3,4,10 que coincide con la estrategia que vimos en clases (glotón con el criterio SPT). El modelo de AMPL es

```
param N;
set POS := 1..N; set JOB := 1..N;
param p {JOB};
var y {JOB,POS} binary;
minimize obj: sum {k in POS} sum {j in 1..k} sum {i in JOB} p[i]*y[i,j];
c1{j in JOB}: sum{i in POS} y[i,j]=1; c2{i in POS}: sum{j in JOB} y[i,j]=1;
```

P2 Modelar $P2|r, d| \sum C$

Sea $I = \{1...N\}$ el conjunto de trabajos

Definamos la variable

$$y_{ijk} \begin{cases} 1 \text{ si trabajo } i \text{ se hace en la posicion } j \text{ en la máquina k} \\ 0 \text{ sino} \end{cases}$$

$k = 1, 2$ y en el peor de los casos, todos los trabajos se procesan en una sóla máquina, por lo que $j = 1..N$.
Las restricciones son

- El trabajo debe hacerse de una tanda y en una sola posición, y en una sóla máquina

$$\sum_k \sum_j y_{ijk} = 1$$

- En cada posicion, puede asignarse a lo más un trabajo (puede que una posicion este vacía)

$$\sum_i y_{ijk} \leq 1$$

- el tiempo de termino del trabajo en la $(n-1)$-esima posicion en la maquina k es

$$\sum_{j=1}^{n-1} \sum_{i=1}^{N} p_i y_{ijk}$$

  que es tambien el tiempo de inicio del trabajo en la n-esima posicion en la maquina k. Ese tiempo debe ser mayor que el tiempo de llegada del pedido (r).

$$\sum_i r_i y_{ink} \leq \sum_{j=1}^{n-1} \sum_{i=1}^{N} p_i y_{ijk}$$

- Los trabajos deben estar terminados para su fecha de entrega (d)

$$\sum_{j=1}^{n} \sum_{i=1}^{N} p_i y_{ijk} \leq \sum_i r_i y_{ink}$$

La función objetivo en este caso es minimar la suma de todas las tardanzas.
El tiempo de término del trabajo n-esimo de la máquina k es

$$\sum_{j=1}^{n} \sum_{i=1}^{N} p_i y_{ijk}$$

y su tardanza es entonces

$$\max(0, \sum_{j=1}^{n} \sum_{i=1}^{N} p_i y_{ijk} - \sum_i d_i y_{ink})$$

y la sumatoria de las tardanzas de todos los trabajos de las dos maquinas es

$$Obj := \sum_{k=1,2} \sum_{n=1}^{N} \max(0, \sum_{j=1}^{n} \sum_{i=1}^{N} p_i y_{ijk} - \sum_i d_i y_{ink})$$

Si resolvemos para $p = (7, 8, 9, 10, 6,9, 8,1, 5, 6, 7,2, 11)$, $r = (0, 5, 20, 5, 10, 5, 15, 0, 5, 10)$, $d = (50, 55, 70, 55, 60, 55, 65, 50, 55, 60)$
agregar al anterior

```
param r:= 1 0 2 5 3 20 4 5 5 10 6 5 7 15 8 0 9 5 10 10;
param d:= 1 50 2 55 3 70 4 55 5 60 6 55 7 65 8 50 9 55 10 60;
```

obtenemos el siguiente horario para cada maquina:

Máquina	Orden de Ejecucin
1	8, 9, 5, 6, 10
2	1, 2, 7, 3, 4

El modelo de AMPL es

```
param N; param M;
set POS := 1..N; set JOB := 1..N; set MAQ := 1..M;
param p {JOB}; param r {JOB}; param d {JOB};
var y {JOB,POS,MAQ} binary;
minimize obj: sum {b in MAQ} sum {k in POS} sum {j in 1..k} sum {i in JOB} p[i]*y[i,j,b];
c1{i in JOB}: sum{k in MAQ} sum{j in POS} y[i,j,k] = 1;
c2{j in POS, k in MAQ}: sum{i in JOB} y[i,j,k] <= 1;
c3{j in POS, k in MAQ}: sum{i in JOB} r[i]*y[i,j,k] <= sum {a in 1..j-1} sum{i in JOB} p[i]*y[i,a,k];
c4{j in POS, k in MAQ}: sum{a in 1..j} sum{i in JOB} p[i]*y[i,a,k] <= sum{b in JOB} d[b]*y[b,j,k];
```