TP Final - Optimización

Lupi, Matta & Raigorodsky

Modelado del problema:

Parámetros:

g_o: Ganancia de la orden o

T_o: Trabajadores necesarios para concretar la orden o

Variables:

$X_{i,d,t,o}$: Variable binaria que vale 1 si la persona *t* fue asignada a la orden *o*, en el turno *i* del día *d*. Vale 0 en caso contrario

D_o: Variable binaria que vale 1 si la orden *o* fue realizada, o 0 en caso contrario

$K_{i,d,o}$: Variable binaria que vale 1 si la orden *o* fue realizada el turno *i* del día *d*, o 0 en caso contrario

$L_{d,t}$: Variable binaria que vale 1 si el trabajador *t* trabajó el día *d*, o 0 en caso contrario

E_t: Variable entera no negativa que representa la cantidad de órdenes totales que toma el trabajador *t* en la semana

X_t^{0-5}: Variable entera no negativa que representa la cantidad de órdenes tomadas por el trabajador *t* dentro de las primeras 5 órdenes

X_t^{6-10}: Variable entera no negativa que representa la cantidad de órdenes tomadas por el trabajador *t* dentro de las siguientes 5 órdenes

X_t^{11-15}: Variable entera no negativa que representa la cantidad de órdenes tomadas por el trabajador *t* dentro de las siguientes 5 órdenes

X_t^{16-}: Variable entera no negativa que representa la cantidad de órdenes tomadas por el trabajador *t* luego de las primeras 15 órdenes y en adelante

Función Objetivo:

$$\text{Max} \sum_o D_o * g_o - \sum_t \left(1000\, X_t^{0-5} + 1200\, X_t^{6-10} + 1400\, X_t^{11-15} + 1500\, X_t^{16-}\right)$$

Sujeto a:

R1: Un trabajador en un turno de un día solo puede estar como mucho en una órden a la vez:

$$\sum_o X_{i,d,t,o} \leq 1 \; \forall \; i, d, t$$

R2: Si no están los trabajadores necesarios para una orden, esta no se cobra

$$\sum_t X_{t,o} \geq T_o D_o \; \forall \; o$$

R3 - R5: Los trabajadores que realizan una misma orden, deben hacerlo en un mismo turno y día (tienen que estar todos juntos)

R3: Una orden no se puede hacer en más de un turno

$$\sum_{i,d} K_{i,d,o} \leq 1 \; \forall \; o$$

R4: La variable $K_{i,d,o}$ se prende en el turno *i* del día *d*, si hay algún trabajador asignado a la orden *o* en ese momento

$$K_{i,d,o} \geq X_{i,d,t,o} \; \forall \; i, d, t, o$$

R5: Si ningún trabajador está asignado a la orden *o* en el turno *i* del día *d*, en ese momento la orden no está tomada

$$K_{i,d,o} \leq \sum_t X_{i,d,t,o} \; \forall \; i, d, o$$

R6: Ningún trabajador puede trabajar los 5 turnos de un día

$$\sum_{i,o} X_{i,d,t,o} \leq 4 \; \forall \; d, t$$

R7-R9: Ningún trabajador puede trabajar los 6 días de la semana

R7: Si el trabajador *t* tuvo alguna orden el día *d*, trabajó ese día

$$L_{t,d} \geq \sum_{o} X_{i,d,t,o} \ \forall \ i, d, t$$

R8: Si el trabajador no tuvo ninguna orden el día *d*, ese día no trabajó

$$L_{t,d} \leq \sum_{i,o} X_{i,d,t,o} \ \forall \ d, t$$

R9: Cada trabajador trabaja como mucho 5 días en la semana

$$\sum_{d} L_{t,d} \leq 5 \ \forall \ t$$

R10 & R11: Órdenes Correlativas (*a*, *b*)

R10: Si dos órdenes son correlativas, o se hacen ambas, o no se hace ninguna

$$D_a = D_b \ \forall \ (a, b)$$

R11: Si se hacen las correlativas, tiene que ser una inmediatamente luego de la otra. Además, por cada par ordenado de órdenes correlativas, la primera de estas (la *a*) no puede hacerse en el último turno de un día (el turno 5)

$$K_{i,d,a} = K_{(i+1),d,b} \ \forall \ (a, b),\ d,\ i = 1, \dots, 4$$
$$K_{5,d,a} = 0 \ \forall \ d, a$$

R12 & R13: Órdenes Conflictivas (*a*, *b*) - Un trabajador no puede hacer dos órdenes conflictivas consecutivamente

$$X_{i,d,t,a} + X_{i+1,d,t,b} \leq 1$$
$$X_{i,d,t,b} + X_{i+1,d,t,a} \leq 1 \ \forall \ (a, b),\ d,\ t,\ i = 1, \dots, 4$$

R14 & R15: Un trabajador no puede tener más de 10 órdenes de diferencia con otro

R14: Las E_t representan la cantidad de órdenes tomadas por el trabajador *t*

$$E_t = \sum_{i,d,o} X_{i,d,t,o} \ \forall \ t$$

R15: La diferencia entre trabajadores no puede superar las 10 órdenes

$$E_a - E_b \leq 10 \;\forall\, (a, b),\; a \neq b$$

R16: La cantidad de órdenes totales de cada trabajador se descompone en la suma de órdenes tomadas dentro de cada franja de salario

$$E_t = X_t^{0-5} + X_t^{6-10} + X_t^{11-15} + X_t^{16-}$$
$$X_t^{0-5} \leq 5$$
$$X_t^{6-10} \leq 5$$
$$X_t^{11-15} \leq 5$$
$$X_t^{16-} \leq 15$$

<u>Restricciones Deseables</u>:

R17: Trabajadores Conflictivos (*a, b*) - Dos trabajadores conflictivos no deberían trabajar juntos en la misma orden

$$\sum_{id} X_{i,d,a,o} + \sum_{id} X_{i,d,b,o} \leq 1 \;\forall\, (a,\; b),\; o$$

R18: Órdenes Repetitivas (*a*, *b*) - Un trabajador no debería realizar dos órdenes repetitivas en una misma semana

$$\sum_{id} X_{i,d,t,a} + \sum_{id} X_{i,d,t,b} \leq 1 \;\forall\, t,\; (a,\; b)$$

Además, tenemos que:

$$X_{i,d,t,o},\; D_o,\; K_{i,d,o},\; L_{d,t} \in \{0,\; 1\}$$

$$E_t,\; X_t^{0-5},\; X_t^{6-10},\; X_t^{11-15},\; X_t^{16-} \;\geq\; 0,\; \in Z$$

Implementación:

La carpeta src/ contiene el código, que debe ser corrido como *python field_service.py <input.txt>*

La implementación con las restricciones deseables se encuentra en src_deseables/, y se corre de igual manera.

Luego de correr una instancia se generan en el directorio actual los archivos balanced_assignment.lp y log.txt. El primero contiene la formulación del problema matemáticamente, y el segundo reporta las variables utilizadas y sus valores.

Chequeos de restricciones:

En la carpeta tests/ se incluyen instancias que permiten chequear de manera aislada cada una de las restricciones. Cada una de las instancias incluye los comentarios pertinentes. Notar que las deseables deberán ser corridas con la implementación que las tiene en cuenta.

<u>Comparación de modelos con instancia “real”:</u>

Corremos ambas implementaciones (la que tiene en cuenta las restricciones deseables y la que no) sobre una misma instancia un poco más grande que las de testeo.

La instancia que creamos tiene órdenes que requieren más de un trabajador. Hicimos que un trabajador esté en conflicto con todo el resto, y que algunas órdenes sean repetitivas.

Cuando ignoramos las restricciones deseables, la corrida toma 0.65 segundos aproximadamente y llega a un beneficio de 27.000. Por otro lado, al tener en cuenta todas las restricciones, el tiempo de ejecución aumenta a 45 segundos aproximadamente, y el beneficio disminuye a 22.000.

Al agregar restricciones la función objetivo será menor o igual a un caso menos restringido, y además en este caso lo complejiza a la hora de resolver.