Trabajo Final de Análisis Prescriptivo y Optimización

Paula Luvini, Facundo Marconi, Florencia Ludueña

A partir de la implementación preliminar del problema de Field Service Management vamos a formular un modelo que cumpla todas las restricciones para realizar una asignación y con la función objetivo enunciada. Dentro de estas restricciones también vamos a agregar un grupo de "restricciones deseables" que no son necesarias para que el problema sea factible pero sí esperamos contar con ellas.

En primer lugar desarrollaremos este modelo teóricamente en los siguientes apartados del informe, luego lo implementaremos en CPLEX. Por último realizaremos algunas experimentaciones cambiando las variables de entrada del modelo para analizar el funcionamiento del mismo.

1 Entrada

Como datos de entrada tenemos:

- t: Cantidad de trabajadores
- j: Cantidad de órdenes
- T_{0j}: Trabajadores necesarios para realizar la orden j.
- Beneficio: b_j es el beneficio obtenido por la resolución de cada una de las órdenes.
- Cantidad conflictos trabajadores
- Cantidad ordenes conflictivas
- Ordenes conflictivas O_c
- Ordenes correlativas O_{cr}
- Trabajadores conflictivos, que son pares de trabajadores que se prefiere no asignar en la misma tarea. L_c

Tenemos información de días laborales $d \setminus d = 1, \ldots, 6$, horarios $h \setminus h = 1, \ldots, 5$ y salario de tramos $S_r \setminus r = 1, 2, 3, 4$ de acuerdo a la cantidad ordenes trabajadas.

2 Variables:

Las variables que planteamos en el problema son:

- Trabajadores: T_{ijhd} es una variable binaria que representa si un trabajador fue asignado o no a una determinada orden, de un determinado día y horario. A modo de subíndices:
  - i es un subíndice que indica a cada trabajador tal que $i = 1, \ldots,$ Cantidad de trabajadores
  - h el subíndice horario o turno tal que $t = 1, \ldots, 5$, correspondiente a los 5 turnos diarios
  - d es un subíndice que indica a cada día tal que $d = 1, , 6$ correspondiente a los días lunes, martes, miércoles, jueves, viernes y sábado.
  - j es un subíndice que indica cada orden.
- W_t es la cantidad de turnos trabajados por el trabajador t.

$$\sum_j \sum_d \sum_h T_{jdh} = R_t \quad \forall i$$

- O_j es ordenes, una variable binaria que toma valor 1 si realiza la orden y 0 en caso contrario.

- H_{jtd} Es una variable binaria que toma valor 1 cuando la orden j se realiza en un determinado horario de un determinado día d

Variables auxiliares:

S_i^s

- S_i^1 el trabajador recibe una remuneración de 1000 para una cantidad de entre 0 y 5 órdenes.
- S_i^2 el trabajador recibe una remuneración de 1200 para una cantidad de entre 6 y 10 órdenes.
- S_i^3 el trabajador recibe una remuneración de 1400 para una cantidad de entre 11 y 15 órdenes.
- S_i^4 el trabajador recibe una remuneración de 1500 para una cantidad mayor de 15 de órdenes.

Turnosimposibles La segunda orden correlativa no puede estar en el primer turno del día, la primera puede estar en la última.

3 Modelo

La solución debe indicar:

- Cómo asignar los trabajadores a las órdenes.
- Cómo asignar las ordenes a los turnos.

De manera de realizar una asignación que maximice el beneficio. El beneficio final obtenido por cada orden es el beneficio asociado a la orden o "ingreso" menos los costos incurridos de realizar la orden. En este el caso el costo está compuesto enteramente por la remuneración que se le realiza al trabajador. El salario del trabajador depende de la cantidad de turnos trabajados por lo que la función de costo se compone en tramos salariales.

$$Max \quad \underbrace{\sum_{j=1}^{O} o_j b_j}_{ingreso} - \underbrace{\sum_{i=1}^{T}\sum_{j=1}^{O}(W1_{ij}1000 + W2_{ij}1200 + W3_{ij}1400 + W4_{ij}1500)}_{costo}$$

Sujeto a:

3.1 Restricciones obligatorias

- $O_j \in \{0, 1\}$ Para ser satisfecha la orden debe tener asignado una cantidad mínima de trabajador T_0, los trabajadores necesarios. Si no se asignan la cantidad suficientes de trabajadores la orden no se realiza. No se asigna más trabajadores de los trabajadores necesarios ya que se debe remunerar a estos por turnos trabajadas, por lo que no sería una forma de maximizar beneficio.

$$O_j \begin{cases} 1 & si \quad \sum_{i=1}^{T} t_{ijhd} = T_o \\ 0 & \quad\quad \sum_{i=1}^{T} t_{ijhd} < T_o \end{cases}$$

- Una orden sólo puede realizarse en un horario, esto es día y turno. Esto asimismo tiene que estar conectado con los trabajadores que trabajen en esa orden. Realizamos para ello una discretización.

$$\sum_t \sum_d H_{jtd} \leq 1 \quad \forall j = 1, \ldots, \text{cantidad de órdenes}$$
$$\sum_i T_{ijtd} = T_{oj} * H_{jtd} \quad \forall j = 1, \ldots, \text{cantidad de órdenes}$$

- Un trabajador solo puede trabajar en una orden j en un horario determinado, por lo que suma de los trabajadores asignados a las ordenes es igual o menor a 1.

$$\sum_j t_{ijhd} \leq 1 \quad \forall i \text{con un h y d dado}$$

- Ningún trabajador puede trabajar los 6 dıas de la planificación ni los 5 turnos de un día.

$$\sum_j \sum_d t_{ijhd} \leq 5$$
$$\sum_j \sum_h t_{ijhd} \leq 4$$

- Hay ordenes que no pueden ser satisfechas en forma simultánea: órdenes conflictivas. Ej:Orden 1 y 2 no puede ser resulta en el siguiente turno porque los trabajadores no pueden concurrir al mismo.

$$O_1 + O_2 \leq 1$$

Generalizando, en la implementación restringimos a los trabajadores creando una lista de turnos consecutivos T_c = [[0,1],[1,2],[2,3],[3,4],[1,0],[2,1],[3,2],[4,3]] que no podían ocurrir, dada la lista de órdenes conflictivas O_{conf}. Si O_{conf} = [1,2]

$$\sum (T_{i1hd} + T_{i2hd}) \leq 1$$
$$\sum (T_{ijhd}) \leq 1$$

Dado:

$$j \in O_{conf} \quad y \quad h \in T_c$$

- Hay pares de ordenes correlativas (lista O_{cr} Usamos XNOR, es decir, si pasa una pasan las dos o ninguna). Dada una lista de turnos correlativos: T_corr = [[0,1],[1,2],[2,3],[3,4]], forzamos a que las órdenes correlativas (dentro de una lista O_corr) sucedan de esa manera simultánea.

$$\sum_{j1} \sum_d \sum_h H_{jhd} = \sum_{j2} \sum_d \sum_h H_{jhd}$$

$$j \in O_{corr} \quad y \quad h \in T_{corr}$$

Además, debemos agregar la restricción de que las órdenes no pueden sucederse en los llamados turnos imposibles: T_i = [4,0].

$$\sum_{j1} \sum_d \sum_h H_{jhd} + \sum_{j2} \sum_d \sum_h H_{jhd} = 0$$

$$j1yj2 \in O_{corr} \quad y \quad t \in T_i$$

- La diferencia entre el trabajador con más ordenes asignadas y el trabajador con menos ordenes no puede ser mayor a 10.

$$Max\Big(R_t\Big) - min\Big(R_t\Big) \leq 10$$
$$\forall j \neq i \quad |R_i - R_j| - 10 \leq 0$$

- Función de remuneración a trozos:

$$\mathrm{W}_1 t \begin{cases} Rt & si \quad 0 \leq Rt < 5 \\ 5 & Rt \geq 5 \end{cases} \qquad \mathrm{S}_t^1 \begin{cases} 1 & si \quad Rt \geq 5 \\ 0 & \text{caso contrario} \end{cases}$$

$$\mathrm{W}_2 t \begin{cases} 0 & si \quad R_t < 5 \\ Rt - 5 & si \quad 5 \leq Rt < 10 \\ 10 & Rt \geq 10 \end{cases} \qquad \mathrm{S}_t^2 \begin{cases} 1 & si \quad Rt \geq 10 \\ 0 & \text{caso contrario} \end{cases}$$

$$\mathrm{W}_3 t \begin{cases} 0 & si \quad Rt < 10 \\ Rt - 10 & si \quad 15 \leq Rt \end{cases} \qquad \mathrm{S}_t^3 \begin{cases} 1 & si \quad Rt \geq 15 \\ 0 & \text{caso contrario} \end{cases}$$

$$\mathrm{W}_4 t \begin{cases} 0 & si \quad R_t < 15 \\ 15 & si \quad Rt \geq 15 \end{cases}$$

Se debe cumplir que:

$$W_{1t} + W_{2t} + W_{3t} + W_{4t} = R_t$$

$$\begin{aligned} 5S_t^1 \leq & W_1 t < 5 \\ (10-5)S_t^2 \leq & W_2 t < (10-5)S_t^2 \\ (15-10)S_t^3 \leq & W_3 t < (15-10) \\ 0S_t^4 \leq & W_4 t < (30^1 - 15) \end{aligned}$$

$$S_t^1, S_t^2, S_t^3, S_t^4 \in \{0, 1\}$$

3.2 Restricciones deseables

- Hay conflictor entre trabajadores que no quieren ser asignador a la misma orden de trabajo. Lista de trabajadores conflictivos L_c

$$\sum_i \sum_j T_{ijhd} \leq 1 \quad \forall i \in L_c$$

- Pares de ordenes repetitivas: que el mismo trabajador no sea asignado a ambas. OR lista de ordenes repetitivos

$$\sum_i \sum_j T_{ijhd} \leq 1 \quad j \in O_R \quad \forall i = i$$

Ejemplo: orden 3 y 4

4 Organización de archivos

Para poder resolver el problema en CPLEX comenzamos a modificar la implementación preliminar que se encontraba en la carpeta *src* llamado *field_service.py*. Mantenemos en este archivo las tener las clases que nos permiten leer los inputs así como la solución del problema. Para modularizar este código y que sea más fácil identificar la fuente de errores agregamos otros dos, *add_variables.py* y *restriction_tools.py* de manera separada. En *add_variables* vamos a encontrar la función que permite agregar todas las variables al modelo y en *restriction_tools* las restricciones obligatorias y deseables del modelo.

En la carpeta *data* podemos encontrar también un código para generar nuevos escenarios para testear el modelo llamado *create_data.py*. También están los dos archivos con los que probamos los escenarios de las secciones siguientes: *input_field_service_original.txt* y *input_field_service_escenario1.txt*.

4.1 Revisión de restricciones

- Órdenes consecutivas de un trabajador: En este caso, los pares de órdenes no pueden realizarse en 2 turnos consecutivos. Sin embargo, no hay restricción para que una sea después de la otra. Realizamos una prueba con 1 trabajador, los 5 turnos y sólo 1 día de trabajo. Pongo de órdenes conflictivas:[(1,2), (2,3), (3,6)].

Ordenes conflictivas:
O_1
O_2
O_3
O_4

Trabajador 0 asignado a orden j, en el turno h, del día d.
T-0-1-0-0
T-0-2-2-0
T-0-3-4-0
T-0-4-1-0

Horarios
H1-0-0
H4-1-0
H2-2-0
H3-4-0

Horas trabajadas
W-0-1-0-0
W-0-4-1-0
W-0-2-2-0
W-0-3-4-0

Función objetivo: 16.0

- Órdenes correlativas: En este caso pruebo que las órdenes 1 y 2 son correlativas. Reviso cómo se asignaron los horarios y confirmo que la primera orden sucede en el turno 1 del día 1 y la segunda orden sucede en el turno 2 del día 1.

Horarios asignado a órdenes:
H3-0-0
H7-0-1
H6-0-5
H1-1-1
H9-1-1
H0-1-5
H8-1-5
H2-2-1
H5-2-1
H4-2-3

- Conflicto entre trabajadores: [(0,1),(1,2),(2,3),(3,4)]
  Genero un escenario con 7 trabajadores (del 0 al 6) y todas las órdenes requiriendo 5 trabajadores a excepción de la orden 9 que sólo necesita 2. Esta orden es efectivamente la única que se lleva a cabo, empleando a los trabajadores 1 y 3 que no tienen problemas entre sí.
- Órdenes repetitivas:

  Ahora miro las órdenes repetitivas, considero un total de 8 trabajadores y todas las órdenes requiriendo 5 trabajadores, exceptuando a la 1 y la 5 que requieren 4. Las órdenes repetitivas son:[(1,5),(2,6),(3,7),(4,8)]

Órden realizada
O9
T1-9-3-1
T3-9-3-1
H9-3-1

Finalmente trabajan los siguientes trabajadores, donde confirmamos que no hay un mismo trabajador realizando las órdenes 1 y 5.

Trabajadores asignados a orden, día y turno
T0-2-4-3
T0-4-1-3
T0-5-1-1
T0-7-0-2
T1-1-3-3
T2-2-4-3
T2-4-1-3
T2-5-1-1
T2-7-0-2
T3-1-3-3
T4-2-4-2
T4-4-1-3
T4-5-1-1
T4-7-0-2
T5-1-3-3
T5-2-4-3
T5-4-1-3
T5-7-0-2
T6-5-1-1
T7-1-3-3
T7-2-4-3
T7-4-1-3
T7-7-0-2

5 Experimentación

A modo de experimentación planteamos dos escenarios que contrastamos con el período base.

5.1 Escenario 1: Muchas órdenes correlativas

Vamos a analizar cuánto influye añadir restricciones más fuertes. El escenario de partida tiene 10 trabajadores, 40 órdenes y sólo 1 par de órdenes correlativas. El escenario de comparación es el mismo problema, la misma cantidad de trabajadores, la misma cantidad de órdenes y mismos conflictos entre trabajadores y órdenes, pero le agregamos más órdenes correlativas hasta totalizar las 20. Es decir, el problema no es complejo en cuanto a cantidad de órdenes y de trabajadores sino en las restricciones. A continuación esbozamos los resultados:

En primer lugar, es llamativo el tiempo de procesamiento que lleva a agregar más restricciones sin cambiar las variables de entrada: El tiempo es 60 veces mayor al tiempo que lleva procesar el problema con 20 pares de órdenes correlativas. En comparación, si bien el beneficio es menor, no se ve perjudicado significativamente: un 8.5% menos. Es decir, si bien el problema ahora tiene muchas restricciones de órdenes correlativas, el mayor costo no es monetario sino de procesamiento. Y esto en un caso de un problema con pocas variables e inputs: sólo hay 40 órdenes que asignar.

Table 1: Comparación entre modelos

Variables	1 par correlativo	20 pares correlativos
Trabajadores empleados	10	10
Ordenes realizadas	35	32
Beneficio	757.025	692.646
Tiempo empleado	1.92 seg	117.78 seg
Nodos restantes al final	0	3
Gap Final	0%	0.39%

Por otro lado, podemos observar que al modificar las restricciones y con ello el poliedro el gap final entre la mejor solución y la mejor cota pasa de ser 0% a 0.39%. De la misma manera, al final del procesamiento quedan 3 nodos restantes en el problema más restrictivo respecto al otro.

Dentro de la carpeta de *data* se puede encontrar el archivo *input_field_service_original.txt* correspondiente al escenario original con 1 par correlativo y el archivo *input_field_service_escenario1.txt* con las 20 pares de órdenes correlativas.

5.2 Escenario 2: Tramos por salario más escalonados

Vamos a analizar cuánto influye añadir tramos salariales más escalonados. Para ello partimos por un salario más bajo para el tramo inicial pero va aumentando de forma más acelerada (un aumento del 40% en cada tramo). El escenario de partida tiene 10 trabajadores, 40 órdenes, 1 solo par de órdenes correlativas, y el tramo salarial original. El escenario de comparación es el mismo problema que el anterior con 1 par de órdenes correlativas, la misma cantidad de trabajadores, la misma cantidad de órdenes y mismos conflictos entre trabajadores y órdenes, pero le modificamos la remuneración de los trabajadores en los siguientes tramos:

- Si realizan entre 0 y 5 órdenes: Obtienen una remuneración de 500 por cada orden.
- Si realizan entre 6 y 10 órdenes: Obtienen una remuneración de 700 por cada orden.
- Si realizan entre 11 y 15 órdenes: Obtienen una remuneración de 980 por cada orden.
- Si realizan más de 15 órdenes: Obtienen una remuneración de 1372 por cada orden.

Table 2: Comparación entre modelos

Variables	Salarios Originales	Salarios modificados
Trabajadores empleados	10	10
Ordenes realizadas	35	36
Beneficio	757.025	830.591
Tiempo empleado	1.92 seg	3.78 seg
Nodos restantes al final	0	0
Gap Final	0%	0%

Observamos que la baja de punto de partida de salarios genera que se produzca una orden más, y el beneficio aumente en un 9%. Teniendo en cuenta que se bajaron los sueldos un 50% para el primer tramo, un 41% en el segundo tramo, un 30% en el tercer tramo, y un 8,53% el último tramo, no observamos el beneficio aumenten en igual magnitud, por lo que no se sugiere ir en esta dirección en caso de afrontar posible huelgas o complicaciones gremiales.