Universidad de San Andrés

Análisis Prescriptivo y Optimización
2021
Trabajo Final

Field Service Management

Problema

Muchas empresas ofrecen productos y servicios que se utilizan en una locación ajena a la misma. Muchas veces, estos productos o servicios dependen de ciertos trabajos como instalación, mantenimiento o inspección que deben ser realizados por cuadrillas que son asignadas especificamente a los diferentes trabajos. En las empresas donde este tipo de necesidades es moneda corriente (telcos, proveedores de energía o gas, entre otros) suele existir un área, *Field Service Management*, que enmarca todas estas actividades.

En particular en este trabajo nos interesa la planificación semanal de las cuadrillas de trabajo a las diferentes órdenes enlistadas. En este problema vamos a contar con una lista de trabajadores y una lista de trabajos a realizar. El objetivo es maximizar la ganancia total, teniendo el cuenta el beneficio que nos otorga resolver un trabajo y considerando los pagos a los trabajadores.

El problema formalmente es el siguiente. Contamos con una cantidad T de trabajadores para realizar los trabajos. Por otro lado, tenemos una lista de O órdenes de trabajo a realizar. Cada orden requiere de una cantidad prefijada de trabajadores, denominada T_o. La semana a planificar tiene 6 días, ya que planificaremos de Lunes a Sábado. Cada día tiene 5 *Turnos* de 2 horas. Se asume que cada órden de trabajo se puede realizar utilizando un turno, y no se pueden realizar varias órdenes en un mismo turno si comparten trabajadores. La solución debe indicar cómo asignar los trabajadores a las órdenes, y a su vez cómo asignar las órdenes a los turnos, cumpliendo las siguientes restricciones necesarias para que la asignación sea factible:

- No toda orden de trabajo tiene que ser resuelta.
- Ningún trabajador puede trabajar los 6 días de la planificación.
- Ningún trabajador puede trabajar los 5 turnos de un día.
- Hay pares de órdenes de trabajo que no pueden ser satisfechas en turnos consecutivos de un trabajador (Si bien en este problema no nos preocupamos por el ruteo sino solo por la asignación, hay órdenes tan lejanas geográficamente que no se podrían satisfacer consecutivamente).
- Una órden de trabajo debe tener asignada sus T_o trabajadores en un mismo turno para poder ser resuelta.

- Existen algunos pares de órdenes de trabajo correlativas. Un par ordenado de órdenes correlativas A y B, nos indica que si se satisface A, entonces debe satisfacerse B ese mismo día en el turno consecutivo.
- Los trabajadores son remunerados según la cantidad de órdenes asignadas por lo que la diferencia entre el trabajador con más órdenes asignadas y el trabajador con menos órdenes no puede ser mayor a 10. Para esto se consideran todos los trabajadores, aún los que no tienen ninguna tarea asignada esta semana.

Si se cumplen las anteriores restricciones, la asignación es factible. De todas maneras, existen dos restricciones más que son *deseables*. Es decir, vamos a querer evaluar si se pueden intentar cumplir o estar cerca de ello. Estas dos restricción son:

- Hay conflictos entre algunos trabajadores que hacen que prefieran no ser asignados a una misma orden de trabajo.
- Hay pares de órdenes de trabajo que son repetitivas por lo que sería bueno que un mismo trabajador no sea asignado a ambas.

Una vez cumplidas las restricciones necesarias, el objetivo de nuestra solución es maximizar la ganancia de la asignación. La ganancia de la asignación se define como la suma de los beneficios de las órdenes satisfechas menos la remuneración otorgada a cada trabajador. Los trabajadores son remunerados según el siguiente esquema:

- Si realizan entre 0 y 5 órdenes: Obtienen una remuneración de 1000 por cada orden.
- Si realizan entre 6 y 10 órdenes: Obtienen una remuneración de 1200 por cada orden.
- Si realizan entre 11 y 15 órdenes: Obtienen una remuneración de 1400 por cada orden.
- Si realizan más de 15 órdenes: Obtienen una remuneración de 1500 por cada orden.

Contexto

Contamos con una implementación preliminar para nuestro problema con la lectura de los datos necesarios de una instancia. En estos datos se encuentran los mencionados anteriormente, particularmente la cantidad de trabajadores, las órdenes de trabajo con su respectivo beneficio y su cantidad necesaria de trabajadores, la lista de órdenes correlativas, la lista de órdenes conflictivas, la lista de órdenes con tareas repetitivas y la lista de trabajadores con conflictos. Luego de realizar la lectura de los datos, estos se encuentran guardados en una instancia de la clase FieldWorkAssignment que contiene los siguientes campos:

- $cantidad\_trabajadores$: un entero que representa la cantidad de trabajadores.
- $ordenes$: una lista que contiene todas las órdenes de la semana.
- $conflictos\_trabajadores$: una lista con pares de trabajadores que conflictúan entre sí.

- *ordenes_correlativas*: una lista con pares ordenados de identificadores de órdenes correlativas.
- *ordenes_conflictivas*: una lista con pares de identificadores de órdenes conflictivas.
- *ordenes_repetitivas*: una lista con pares de identificadores de órdenes repetitivas.

A su vez, una orden de trabajo contiene los siguientes campos:

- *id*: un entero identificador de la orden
- *beneficio*: un entero que representa el beneficio que nos da una orden si la satisfacemos.
- *trabajadores_necesarios*: un entero que representa la cantidad de trabajadores necesarios para satisfacerla.

Enunciado

La resolución del trabajo consiste en la realización de modelos que se adecuen al problema, completar los códigos provistos para poder resolver diferentes instancias del problema y la entrega de un informe donde se detallen los puntos a continuación. Se pide:

1. Modelo: Formular un modelo que cumpla todas las restricciones para realizar una asignación y que tenga en cuenta la función objetivo enunciada.
2. Restricciones deseables: Reformular el modelo para tener en cuenta las restricciones deseables. Se deben tener en cuenta tanto como restricciones nuevas, como términos de la función objetivo.
3. Implementación: Realizar la implementación de las alternativas propuestas utilizando CPLEX.
4. Experimentación: Realizar experimentación sobre las diferentes alternativas para lograr una discusión tanto cuantitativa como cualitativa de los resultados obtenidos.

Se debe entregar un informe que desarrolle los puntos propuestos junto con el código implementado. El código debe contener comentarios y presentar una modularización adecuada para que sea entendible por alguien que no realizó la implementación.

Fechas de entrega

- *Formato Electrónico:* Domingo 17 de Octubre de 2021, hasta las **23:59 hs**, enviando el trabajo (informe + código) a la dirección fpousa@udesa.edu.ar. El subject del email debe comenzar con el texto `[TPFinal]` la lista de apellidos de los alumnos. Todos los integrantes del grupo deben estar copiados en el mail.