

Balanceando la felicidad

Problema

Las empresas de charters empresariales realizan diferentes tipos de traslados de un cierto punto hasta la empresa cliente o viceversa. Uno de los servicios que se venden para empresas de tamaño mediano o grande es la contratación de una unidad entera (como por ejemplo un micro) para realizar un viaje entre dos puntos en ciertos horarios prefijados. Uno de los problemas que surge en este contexto es la asignación de los servicios a las diferentes unidades con las que cuenta la empresa. Este problema de asignación tiene varias características. Por un lado, la asignación tiene que cumplir ciertas restricciones de factibilidad como para que sea realmente posible cubrir todos los viajes con la asignación propuesta. Por otro lado, dado que los choferes facturan por kilómetro recorrido, no queremos realizar una asignación que otorgue una cantidad de kilómetros muy dispar a las diferentes unidades porque eso generaría un descontento con los choferes.

El problema formalmente es el siguiente. Contamos con una cantidad U de unidades indistinguibles para realizar los traslados. Por otro lado, tenemos una lista de servicios a realizar. Cada servicio requiere exactamente de una unidad para ser satisfecho. Los servicios cuentan con la siguiente información: un horario de salida, un horario de llegada, la cantidad de kilómetros a recorrer, la *zona* en donde se encuentra el punto de salida y la zona en donde se encuentra el punto de llegada. Las zonas son simplemente 7 puntos que segmentan a la Ciudad de Buenos Aires y sus alrededores para poder saber una vez que se termina un servicio en cierto lugar, cuanto tiempo llevará aproximadamente moverse al punto de salida del próximo servicio asignado a una unidad. Para poder realizar esta estimación, contamos también con una matriz de distancias entre las 7 zonas.

Los servicios que tendremos que asignar a las diferentes unidades comprenden el transcurso de un día hábil para nuestra empresa. En este contexto una unidad nunca podrá realizar demasiados viajes. De esta manera, una posible forma de modelar este problema es mediante las variables binarias x_{ijk} que indican si el servicio i es asignado a la unidad j como su k -ésimo viaje.

Una vez definidas dichas variables se debe modelar el problema para producir una asignación factible y útil en la realidad. Para esto se deben modelar las siguientes características:

- La función objetivo debe ayudar en lo posible a que la cantidad de kilómetros asignados a cada unidad sea *balanceada*.
- La asignación debe ser realizable. Es decir, se deben cumplir restricciones lógicas como que los servicios de una unidad tengan un orden cronológico razonable, no se le asignen dos servicios *incompatibles* a una misma unidad, etc.
- La asignación debe contemplar restricciones deseables pero no necesarias como evitar que una unidad esté demasiado tiempo esperando entre un servicio y otro.

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 lista de servicios y la matriz de distancia entre zonas. Luego de realizar la lectura de los datos, estos se encuentran guardados en una instancia de la clase `BalancedAssignmentInstance` que contiene los siguientes campos:

- *cantidad_servicios*: un entero que representa la cantidad de servicios.
- *servicios*: una lista que contiene todos los servicios de un día.
- *cantidad_zonas*: un entero que representa la cantidad de zonas.
- *distancias*: una matriz que contiene las distancias entre las diferentes zonas.

A su vez, un servicio contiene los siguientes campos:

- *horario_salida*: un entero que representa el minuto en el que sale el servicio. Considerando el minuto 0, como las 00:00hs del día.
- *horario_llegada*: un entero que representa el minuto en el que llega el servicio.
- *punto_salida*: un string que indica el punto de salida.
- *punto_llegada*: un string que indica el punto de llegada.
- *zona_salida*: un entero que representa la zona a la que pertenece el punto de salida.
- *zona_llegada*: un entero que representa la zona a la que pertenece el punto de llegada.

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 detalle lo que se indica en la próxima sección. Se pide:

1. Función objetivo: Formular de al menos dos maneras diferentes una función objetivo para nuestro modelo que ayude a que la facturación de los choferos no sea demasiado desbalanceada. Es posible que se necesiten agregar variables auxiliares al modelo.
2. Restricciones *hard*: Formular todas las restricciones que consideren necesarias para que la asignación conseguida sea realizable. Desarrollar y explicar las diferentes opciones analizadas.
3. Restricciones *soft*: Formular al menos dos restricciones deseables pero no necesarias para nuestro problema.
4. Implementación: Realizar la implementación de las alternativas propuestas utilizando CPLEX.

5. 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:* Lunes 26 de Noviembre de 2018, hasta las **23:59 hs**, enviando el trabajo (informe + código) a la dirección **fedepousa@gmail.com**. El subject del email debe comenzar con el texto [TP1] la lista de apellidos de los alumnos. Todos los integrantes del grupo deben estar copiados en el mail.

Importante: El horario es estricto. No se considerarán los correos recibidos después de la hora indicada.