Universidad de Buenos Aires
Facultad de Ingeniería
Año 2019 - 2do Cuatrimestre

71.14 MODELOS Y OPTIMIZACIÓN - CURSO 4

TRABAJO PRÁCTICO: Segunda entrega
FECHA DE ENTREGA: 12 de Octubre de 2019

INTEGRANTES:
PEREZ ONDARTS, Flavio - 96786
<perezflavio94@gmail.com>
FLOREZ DEL CARPIO, Christian - 91011
<chris.florez.d.c@gmail.com>

Índice

1. Introducción

En este informe se continúa con la resolución del problema que comenzó a tratarse en la primer entrega del TP.

Para esta segunda entrega hemos pasado el modelo propuesto en la entrega anterior al formato requerido para su resolución mediante el software de programación lineal GLPK. Además hemos conseguido correr ese modelo y obtener una solución óptima para el subconjunto de datos reducido que provisto por la cátedra.

Dado el formato de los datos, hemos también desarrollado un modelo alternativo al que propusimos originalmente en la primer entrega. La diferencia de este modelo con el original es que prescinde de la táctica de agrupar a los votantes en 'manzanas', y por lo tanto, su resolución nos permite saber que centro le corresponde a cada votante.

Podríamos decir que el segundo modelo es más preciso, ya que el resultado del primero nos dice para cada manzana, cuantos votantes deben ir a cada centro, pero no hace el trabajo de asignar individuamente a cada votante. Además, al agrupar a los votantes por manzana y decir que todos los votantes tienen la misma ubicación, estamos introduciendo cierto error, más allá de que podamos considerarlo insignificante. El segundo modelo considera la ubicación exacta de cada votante, y por lo tanto no introduce el error mencionado.

Obviamente, no todo es a favor del segundo modelo. El agrupamiento de votantes en manzanas genera una reducción considerable de la cantidad de variables enteras que tiene el modelo, y por lo tanto hace que su tiempo de resolución sea mucho menor.

Otra cosa que podemos agregar es que dado a

2. Equilibrio entre eficiencia y equidad

En el enunciado de la segunda entrega se indica que debemos considerar un equilibrio entre la solución más eficiente y la más equitativa.

En un principio nosotros pensamos plasmar esto mediante una restricción, que limitara la distancia máxima que un usuario debe recorrer para votar, o que estableciera un límite en la relación entre el promedio de la distancia recorrida por votante, con la distancia máxima recorrida por un votante. Sin embargo, creemos que esto podría llegar a ser perjudicial para el modelo, ya que simplemente podría ocurrir que la disponibilidad y la ubicación de los centros me impidan cumplir dichas restricciones y por lo tanto tengamos un problema sin solución. Algo que también podría pasar, es que si establecemos una restricción para la relación entre la distancia máxima de un votante con el promedio de las distancias, y dadas las condiciones del problema el modelo no puede reducir la distancia máxima, el modelo busque una solución poco eficiente solo para cumplir con esta restricción.

Como solución a esto lo que pensamos fue modificar un poco el funcional propuesto en la entrega anterior, para que no contemple únicamente reducir el promedio de las distancias recorridas por cada uno de los votantes, sino que le de también importancia a la distancia máxima recorrida por un votante, y de esta forma se llegue a una solución más equilibrada. El funcional sería:

$$Z(min) = Dist_{Avg} + Dist_{Max}$$

Siendo $Dist_{Avg}$ la distancia promedio recorrida por los votantes y $Dist_{Max}$ la distancia máxima recorrida, es decir, la distancia que recorre el votante que más distancia debe recorrer. De esta forma le estamos dando, en el funcional, la misma 'penalidad' o el mismo 'peso' a la distancia máxima y al promedio de las distancias.

3. Distancia utilizada

Dado a que los datos de la ubicación de los votantes vienen dados en coordenadas (latitud y longitud) hemos tenido que utilizar la formula de Haversine para llegar a las distancias en metros entre los votantes y los centros. Esta fórmula nos devuelve la longitud en metros del segmento de la recta que une dos puntos en el circuito.

Para que esta distancia obtenida nos sea útil, debemos tomar una hipótesis bastante fuerte: la distancia que recorrerá un votante para ir a votar será directamente proporcional a la distancia de Haversine entre un votante y su centro. Es necesario tomar esta hipótesis ya que con los datos que tenemos, no podemos obtener la distancia exacta entre un votante y un centro, ya que no tenemos idea de las calles o caminos que forman parte del circuito. Por la misma razón, tampoco tiene mucho sentido considerar otro tipo de distancias, como la distancia Manhattan.

4. Primer modelo (*manzanas*)

En esta sección desarrollamos nuevamente el modelo que habíamos propuesto ya en la primer entrega. Dado que debimos realizar algunos ajustes para esta segunda entrega para como por ejemplo la introducción de variables binarias, concepto que no manejabamos todavía al realizar la primer entrega, presentamos aquí la reformulación del modelo, incluyendo la declaración de constantes, variables, restricciones y función objetivo.

4.1. División en manzanas

Dado que no contamos con el dato exacto de a que manzana corresponde cada votante, lo que hicimos fue dividir el circuito en porciones de 100m x 100m, y hemos mapeado a cada votante a una de estas porciones. A partir de ahora, llamaremos *manzanas* a estas porciones del circuito. Al realizar esta división del circuito, podemos ver al mismo como una matriz con un número FIL de filas y COL de columnas, y podemos ver a cada manzana como un elemento f, c (f de fila, c de columna) de dicha matríz. A cada manzana le corresponderá una cantidad de votantes $V_{f,c}$ y una distancia a cada centro j dada por $D[f, c, j]$

4.2. Constantes

En primer lugar, antes de declarar las variables que estarán involucradas en nuestro modelo, debemos también presentar algunas constantes, que representarán datos conocidos para el modelo.

Identificaremos cada centro con un número j $(j = 1, 2, .., n)$ siendo n la cantidad total de centros de votación. Entonces, se representará la capacidad de cada uno de esos centros con la constante:

$$C_1, C_2, ..., C_n$$

Es decir, la constante C_j representa la cantidad de votantes que puede recibir el centro j.

Representaremos al número de votantes mínimo que debe asignarse a cada centro como la constante:

$$C_{min} \tag{1}$$

En el enunciado de esta entrega, se indicó que para el set de datos reducido, C_{min} tiene un valor de 30, y para el set completo, 120. El valor es el mismo para todos los centros.

Por otro lado, tenemos la matriz de las manzanas del circuito. Identificaremos cada una de esas manzanas con un par (f, c), siendo FIL y COL la cantidad de filas y columnas de la matriz

$$V_{1,1}, V_{1,2}, ..., V_{FIL,COL}$$

Cada variable $V_{f,c}$ representa la cantidad de votantes que residen en la manzana i.

También tenemos la cantidad total de votantes del circuito, que llamaremos TV

Otro dato que aparece en nuestro modelo, es la distancia de cada manzana del circuito a cada uno de los centros de votación. Ya habiendo identificado a cada una de las manzanas y los centros con un número, definimos

$$D[f, c, j]$$

como la distancia entre la manzana (f, c) y el centro de votación j, con $f = 1, 2, ..., FIL$, $c = 1, 2, ..., COL$ y $j = 1, 2, .., n$

Esta distancia corresponde a la distancia de Haversine entre el punto centrad de la manzana y el centro de votación.

4.3. Variables

La primer variable que definimos es

$$X_{f,c,j}(integer) : Cantidaddevotantesdelamanzana(f, c)queseasignanalcentroj$$

para $f = 1, 2, ..., FIL$, $c = 1, 2, ..., COL$ y $j = 1, 2, ..., n$.

Tendremos la variable

$$abre_j = \begin{cases} 1 & \text{el centro j abre} \\ 0 & \text{sino} \end{cases}$$

Tendremos la variable

$$y_{f,c,j} = \begin{cases} 1 & \text{si asignamos algun votante de (f,c) al centro j} \\ 0 & \text{sino} \end{cases}$$

Por último:

$$X_{max} : \text{Distancia máxima recorrida por un votante}$$

4.4. Restricciones

Teniendo definidas las constantes del modelo y las variables, podemos definir restricciones sobres las segundas, utilizando las primeras.

Las restricciones que se plantean de forma genérica para una manzana (f,c) o un centro j, deben entenderse que valen para todas las manzanas (f,c) con $f = 1, 2, .., FIL$, $c = 1, 2, .., COL$ y centros $j = 1, 2, ..n$

4.4.1. Cantidad de votantes de una manzana

La suma de los votantes de una manzana asignados a los distintos centros de votación, debe ser igual al total de votantes que residen en la manzana.

$$X_{f,c,1} + X_{f,c,2} + ... + X_{f,c,n} = V_{f,c}$$

4.4.2. Capacidad de un centro - no asignar a un centro cerrado

La cantidad de votantes asignados a un centro no puede exceder la capacidad del mismo. Además, al mismo tiempo le prohibimos al modelo que asigne votantes a un centro si el mismo no abre.

$$X_{1,j} + X_{2,j} + ... + X_{m,j} \leq C_j * abre_j$$

4.4.3. Asignación mínima a cada centro

La cantidad de votantes asignados a un centro no puede ser menor a una cantidad mínima.

$$X_{1,j} + X_{2,j} + ... + X_{m,j} \geq C_{min}$$

4.4.4. Envío algún votante de (f,c) a j

Si envío votantes de (f,c) al centro j, debo enviar al menos uno sino, la cantidad enviada debe ser cero.

$$X_{f,c,j} \geq 1 \times y_{f,c,j}$$

$$X_{f,c,j} \leq M \times y_{f,c,j}$$

4.4.5. Distancia máxima

La distancia máxima debe ser mayor o igual que todas las distancias recorridas.

$$X_{Max} \geq D[f,c,j] * y_{f,c,j}$$

4.5. Funcional

Minimizaremos la suma distancias que recorran el total de los votantes.

$$\begin{aligned} Z(min) &= \sum_{f=1}^{FIL} \sum_{c=1}^{COL} \sum_{j=1}^{n} \frac{D[f,c,j] X_{f,c,j}}{TV} + X_{max} \\ &= Dist_{Avg} + Dist_{Max} \end{aligned}$$

4.6. Modelo en GLPK

Hemos generado 2 archivos: manzanas.mod y manzanas-red.mod, el primero corresponde al modelo para el set de datos completo y el segundo para el set de datos reducido. Estos modelos solo difieren en la restricción para la cantidad mínima de votantes por centro, pero son en esencia el mismo modelo.

Hemos llegado a la solución óptima con el set de datos reducido.

No hemos podido hacer una ejecución lo suficientemente larga de glpsol para el set de datos completo como para llegar a la solución final, incluyendo la optimización entera.

4.6.1. manzanas.mod/manzanas-red.mod

Ver archivos adjuntos.

4.6.2. manzanas-red.sol

Ver archivos adjuntos

4.6.3. Interpretación del resultado

Se ha obtenido un funcional con valor de 2613 (metros) y además, la distancia máxima que recorre un votante es de aproximadamente 2102 metros. Esto quiere decir que el promedio que recorren los votantes es de aproximadamente de 510 metros.

En el archivo manzanas-red.sol figura también el valor de cada variable x[f,c,j], que corresponde a la cantidad de votantes de la manzana (f,c) que debemos asignar a cada centro.

La ejecución de glpsol sobre este modelo genera también un archivo *matriz.csv*, donde cada registro represtenta a que manzana se asigna cada votante. Se adjunta a este informe un script de bash *proceso-manzanas.sh*, que al ejecutarse en el directorio donde se encuentra *matriz.csv*, genera un archivo .csv por cada manzana del circuito, donde se indican los votantes que pertenecen a la manzana en cuestión (se adjunta a este informe el resultado de correr dicho script para el set de datos reducido). Utilizando estos archivos y los valores de las variables x[f,c,j], el usuario de este modelo podría decidir que votantes de cada manzana se asignan a cada centro.

Lamentablemente no hemos llegado a generar más scripts que depuren los resultados obtenidos por este modelo y le den un formato más amigable para personas que no tengan ningún conocimiento en GLPK o investigación operativa.

5. Segundo modelo (sin *manzanas*)

En este modelo consideraremos cada votante del set de datos de forma individual. Cada votante está identificado con número i.

5.1. Constantes

Tenemos las mismas variables C_j del modelo anterior, indicando la capacidad de cada centro j. También tenemos la misma constante C_{min}.

Otro dato que aparece en nuestro modelo, es la distancia de cada votante i a cada uno de los centros de votación.

$$D[i,j] : \text{Distancia que existe entre el votante i y el centro j}$$

Esta distancia corresponde a la distancia de Haversine.

5.2. Variables

Tendremos la variable

$$abre_j = \begin{cases} 1 & \text{el centro j abre} \\ 0 & \text{sino} \end{cases}$$

Tendremos la variable

$$y_{i,j} = \begin{cases} 1 & \text{si asignamos al votante i al centro j} \\ 0 & \text{sino} \end{cases}$$

Tenemos también la distancia que recorre cada votante dada por la variable

$$X_i \text{ : Distancia que debe recorrer el votante i para votar.}$$

Por último:

$$X_{max} \text{ : Distancia máxima recorrida por un votante}$$

5.3. Restricciones

Teniendo definidas las constantes del modelo y las variables, podemos definir restricciones sobres las segundas, utilizando las primeras.

Las restricciones que se plantean de forma genérica para un votante i o un centro j, deben entenderse que valen para todos los votantes $i = 1, 2, .., TV$ y centros $j = 1, 2, ..n$

5.3.1. Votante se asigna a un centro

Todos los votantes deben ser asignados a un centro, ni más ni menos.

$$\sum_{j=1}^{n} y_{i,j} = 1$$

5.3.2. Capacidad de un centro - no asignar a un centro cerrado

La cantidad de votantes asignados a un centro no puede exceder la capacidad del mismo. Además, al mismo tiempo le prohibimos al modelo que asigne votantes a un centro si el mismo no abre.

$$\sum_{i=1}^{TV} y_{i,j} \leq C_j \times abre_j$$

5.3.3. Asignación mínima a cada centro

Si un centro abre, la cantidad de votantes asignados a un centro no puede ser menor a una cantidad mínima.

$$\sum_{i=1}^{TV} y_{i,j} \geq C_{min} \times abre_j$$

5.3.4. Distancia del votante

La distancia que recorre el votante debe ser igual a la distancia entre el mismo y su centro.

$$X_i = \sum_{j=1}^{n} D[i, j] \times y_{i,j}$$

5.3.5. Distancia máxima

La distancia máxima debe ser mayor o igual que todas las distancias recorridas. El funcional se encargará de acotar el valor de esta variable.

$$X_{Max} \geq X_i$$

5.4. Funcional

$$\begin{aligned} Z(min) &= \sum_{i=1}^{TV} \frac{X_i}{TV} + X_{Max} \\ &= Dist_{Avg} + Dist_{Max} \end{aligned}$$

5.5. Modelo en GLPK

Hemos generado 2 archivos para este modelo: 2.mod que tiene como input el set completo de datos, y 2-red.mod que tiene el set reducido. Estos modelos solo difieren en la restricción para la cantidad mínima de votantes por centro, pero son en esencia el mismo modelo.

Al igual que para el modelo anterior, solo hemos llegado al resultado final incluyendo la refinación entera para el set de datos reducido

5.5.1. 2.mod/2-red.mod

Ver archivos adjuntos

5.5.2. 2-red.sol

Ver archivo adjunto

5.5.3. Interpretación del resultado

Estos son los resultados que se obtuvieron después de correr dicho modelo:

Centros de votación elegidos

Centro	Cantidad de votantes
20	43
18	30
14	66
25	30
9	295
10	30
30	30
27	30
3	75
4	40
24	30
16	45
1	83
8	102
12	31
28	47
29	30
7	30
15	37
22	30

También se parsearon los resultados arrojados por glpk sobre a qué centro de votación fue asignado cada votante dicha información se encuentra en el archivo analisis_de_votantes_por_centro.csv

Como vemos todos los votantes fueron asignados a un centro de votacion, y solo se abrieron los centros que tuvieron igual o mas de 30 votantes (cantidad minima de votantes para el set reducido de datos).

El funcional obtenido es igual a 2643.473943 metros, y el valor de la distancia máxima fue de 2036 metros aproximadamente. Por lo tanto, el valor promedio de distancia recorrida fue de aproximadamente 600 metros, un valor similar al obtenido por el primer modelo.

Herramientas utilizadas para el análisis

Para la resolución de este modelo, se ultizo GLPK, y para poder interpretar los resultados arrojados por este, desarrollamos scripts en python los cuales nos ayudaron a filtrar los datos relevantes y transformarlos de manera tal de poder ver los centros que fueron elegidos, y a que centro fue asignado cada votante.

Para ejecutar los scripts sobre los resultados de este modelo, se le debe dar al archivo output de GLPK para este modelo el nombre "tp.sol".