

25 de Setiembre de 2017

En un proyecto independiente a éste, se prevé conectar esos tres puntos por una línea de metro de alta velocidad, así que asumiremos que llegar al Centro es indiferente del punto en términos de tiempo. Se han identificado además cuatro agregadores zonales, las terminales: Cerro (CRO), Colón (TCO), Carrasco (CAR) y Pocitos (POC). Se supone que los pasajeros con destino al Centro, llegan a alguna de estas terminales por un medio tradicional (un ómnibus por ejemplo), para luego tomar un LRT que lo alcance a PIN, XXX o PAM en cuestión de pocos minutos.

A efectos ilustrativos se esquematizan en el plano los tramos posibles a ser usados por las líneas del LRT. Los nodos del Centro están distinguidos en azul, mientras que: CAR, CRO, POC y TCO están en rojo. Las restantes estaciones a construir surgirán del resultado de este diseño y los nodos potenciales son los coloreados con amarillo. Los archivos *nodos.txt* y *aristas.txt* contienen los datos del problema en la forma de un grafo no-dirigido, conteniendo *nodos.txt* los índices y coordenadas, mientras que *aristas.txt* contiene: índice nodo1, índice nodo2, costo y demora esperada (en segundos) de cada arista.

1. En una primera aproximación se busca el trazado de costo mínimo para conectar: CAR, CRO, POC y TCO con el Centro. Las únicas restricciones son que deben existir dos líneas (caminos) para conectar tanto: CAR, como CRO y TCO con los nodos del Centro. En el caso de POC, por la población que concentra, este número es tres. Además, como prevención ante contingencias, se requiere que ninguno de los tramos del plano sea usado por más de una línea (las estaciones sí pueden ser parte de más de una línea). Plantee y resuelva el problema como uno de programación lineal (LP), inspirándose en el modelo LP para el MFP visto en clase, considerando aquí que los flujos son conocidos y que se busca minimizar el costo.

*Sugerencia:* Usar el *linprog* de Matlab o el *glpk* (<https://www.gnu.org/software/glpk/>).

¿Cuál es la complejidad del problema?

Vuelva a resolver el problema buscando optimizar la *demora* agregada en toda la red. Compare los resultados en ambas versiones. Compare los tiempos de recorrido de las líneas con los óptimos (si hubiera un único camino desde cada concentrador).

2. Suponga ahora las mismas condiciones que la parte anterior, sólo que los tramos pueden ser usados por líneas de distinto origen (no por las de un mismo origen). Proponga una formulación MIP para este problema ¿Es la nueva versión una relajación de la anterior? ¿Es computacionalmente más simple o más compleja?

*Opcional:* Resuelva el problema en forma exacta usando *intlinprog* o *glpk*. En cualquier caso, se presentará en clase la solución óptima de esta parte para el análisis.

Calcule cuál es el tiempo de recorrido en cada línea (desde su concentrador al centro), y compare la performance (costo vs calidad) de ambas versiones.

3. Se busca agregar restricciones a la variante de la parte 2), con el fin de mantener el nivel de servicio dentro de límites razonables. Se apunta a no superar el tiempo máximo de viaje para las líneas encontradas en la parte 1), desde el concentrador correspondiente. Demuestre que la versión general del problema aquí planteado es NP-Hard.

*Sugerencia:* Pensar en el CSP (Constrained Shortest Path) y buscar reducciones polinomiales a partir de KSP (KnapSack Problem).

4. Proponga e implemente una Metaheurística para encontrar soluciones de buena calidad para el problema de la parte 3).

*Nota:* Se distribuirán artículos donde se presentan Metaheurísticas exitosamente aplicadas a problemas relacionados.

*Opcional:* Estime cotas para el error cometido.

5. A efectos de valorar la performance del algoritmo implementado en la parte 4), se proveerán instancias equivalentes aunque de mayor tamaño, junto con los valores de las mejores soluciones conocidas, y los tiempos de cómputo asociados.