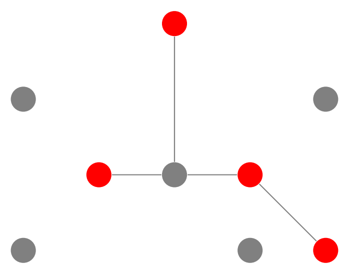
1158065 - Laboratorio de Optimización
Tarea 2 - 2025-I.

Considérese una gráfica completa $G = (V, E)$ en el plano. Dado un subconjunto no vacío $V' \subset V$, se desea construir un árbol que abarque V' y que sí puede incluir vértices que no están en V'. Se desea además que el árbol sea de costo mínimo, definiendo éste como la suma de las longitudes de sus aristas. A continuación se muestra un ejemplo de instancia y su solución, indicando en rojo los vértices de V' y en gris el resto de los vértices de G.



Ejemplo de instancia y su solución (árbol de Steiner).

Un modelo de programación entera para el problema descrito es el siguiente[1]. Sea $V = \{v_1, v_2, \ldots, v_n\}$ y $V' = \{v_1, v_2, \ldots, v_m\}$ con $m \leq n$. La distancia entre v_i y v_j se denotará como $d_{i,j}$ y será igual a su distancia euclidiana. Sea x_i una variable binaria que indica si v_i se incluye en la solución. Sea $y_{i,j}$ una variable binaria que indica si el arco de v_i a v_j se incluye en la solución. Sea $w_{i,j}$ una variable entera que denota el flujo no negativo que transfiere el arco respectivo.

- Función objetivo que minimiza el costo de la solución.

$$\text{minimizar} \quad \sum_{i=1}^{n} \sum_{j=1}^{n} d_{i,j}\, y_{i,j}$$

- Restricciones que obligan a incluir los vértices de V' en la solución.

$$x_i = 1 \qquad \text{para } 1 \leq i \leq m$$

- Restricciones que crean un arco saliente en cada vértice usado en la solución, excepto en v_1.

$$\sum_{j=1}^{n} y_{i,j} = 0 \qquad \text{para } i = 1$$

$$\sum_{j=1}^{n} y_{i,j} = x_i \qquad \text{para } 2 \leq i \leq n$$

[1] https://wseas.com/journals/computers/2022/a625105-028(2022).pdf

- Restricciones que generan una unidad de flujo en cada vértice usado en la solución, excepto en v_1.

$$\sum_{j=1}^{n} w_{i,j} - \sum_{j=1}^{n} w_{j,i} = x_i \qquad \text{para } 2 \leq i \leq n$$

- Restricciones que evitan que un arco no usado transfiera flujo.

$$0 \leq w_{i,j} \leq (n-1)\, y_{i,j} \qquad \text{para } 1 \leq i, j \leq n$$

Escribe un programa en C++ que lea las coordenadas de los n vértices de V y que genere y resuelva el modelo de programación lineal entera descrito anteriormente usando el API de programación de Gurobi. En la página del curso se encuentra una instancia que describe una gráfica para el problema[2]. La primera línea contiene el valor de n. La segunda línea contiene el valor de m. Cada una de las siguientes n líneas contiene dos reales separados por espacios que denotan las coordenadas de un vértice de V. Los vértices se numeran implícitamente de 1 a n según el orden de la entrada. Las variables de tu modelo deberán llamarse `x_`i, `y_`i`_`j y `w_`i`_`j respetando la numeración dada en el modelo previo.

Deberás enviar a racc@azc.uam.mx dos archivos adjuntos: el código fuente de tu programa con el nombre *matricula*`t2.cpp` y el archivo *matricula*`t2.sol` que se obtiene de resolver el modelo mediante Gurobi. Tu programa deberá funcionar también para gráficas distintas a la del ejemplo, pero cuya descripción de entrada siga el mismo formato.

[2] `https://racc.mx/uam/trimestres-anteriores/2025-i/lo/tarea2_entrada.txt`