Optimización Combinatoria y Teoría de Grafos

Docentes: Graciela Nasini, Daniel Severin

Práctica nº 1

En esta práctica asumimos que todos los grafos son simples (sin lazos ni aristas múltiples).

1. Sea $G = (V, E)$ un grafo y $\overline{G}$ su complemento. Si G es isomorfo a $\overline{G}$, ¿cuál es el cardinal de E?

2. Determine, salvo isomorfimos, la cantidad total de grafos de 5 vértices. ¿Y si se contasen también los isomorfos, a cuánto ascendería esta cantidad?

3. Demuestre que la cantidad de vértices de grado impar de todo grafo es par.
Sugerencia: Utilice la identidad $\sum_{v \in V} d(v) = 2.|E|$, donde $d(v)$ es el grado del vértice v y V, E son los conjuntos de vértices y aristas respectivamente.

4. Encuentre una condición suficiente y necesaria para que un digrafo sin vértices aislados tenga un circuito euleriano dirigido. *Sugerencia:* Ver Teorema 11.3 sobre circuitos eulerianos en grafos no dirigidos en [LMDC, pág. 552].

5. Dado un grafo $G = (V, E)$, demuestre que las siguientes proposiciones son equivalentes entre sí: a) G es un árbol, b) $|E| = |V| - 1$ y G es conexo, c) $|E| = |V| - 1$ y G es acíclico.

6. Determine la cantidad de matchings perfectos [LMDC, Def. 13.5, pág. 684] en un grafo completo de $2n$ vértices. *Sugerencia:* Por inducción.

7. a) Pruebe que si un grafo G admite dos matchings M y M' entonces toda componente del grafo $G[(M \backslash M') \cup (M' \backslash M)]$ es un ciclo par o un camino.
b) Pruebe que todo árbol tiene a lo sumo un matching perfecto.
Sugerencia: Puede hacerlo por inducción o puede usar el ítem (a).

8. a) Para $n \geq 3$, demuestre que la cantidad de ciclos hamiltonianos en un grafo completo de n vértices es $\dfrac{(n-1)!}{2}$.
b) Suponga que tiene una computadora capaz de evaluar en un nanosegundo (10^{-9} segundos) el costo de recorrer un ciclo hamiltoniano. ¿Qué tan grande puede ser la instancia del *Problema del Viajante de Comercio Euclidiano* [LCO, pág. 1] si empleamos 24 horas de tiempo de cómputo? ¿Y si la computadora fuera 10 veces más rápida? ¿Y si fuera 100 veces más rápida?

9. Demuestre que existe una instancia del *Problema del Viajante de Comercio en grafos* con 4 ciudades con solución óptima de costo 7 y tal que, para todo $M \geq 8$ la solución provista por el *Algoritmo del Vecino más Cercano* tiene costo M (en otras palabras, la solución que genera el algoritmo se puede alejar de la óptima tanto como se quiera).

10. Determine el número de operaciones elementales que realiza el Algoritmo del Vecino más Cercano para el Problema del Viajante Euclídeo sobre una instancia de n puntos en el plano.

11. Pruebe que existen instancias del *Problema de Matching Euclídeo (de Peso Mínimo)* [LCO, pág. 4] tales que la solución obtenida por el algoritmo *goloso* puede acercarse tanto como se quiera al triple de la solución óptima.

12. Pruebe que el *Problema de Matching Perfecto de Peso Mínimo* puede considerarse un caso particular del *Problema de Matching de Peso Máximo*, es decir para toda instancia del Problema de Matching Perfecto de Peso Mínimo es posible generar una instancia del Problema de Matching de Peso Máximo tal que, su resolución permite resolver el primer problema. *Nota:* En ambos problemas, una instancia viene dada por un grafo $G = (V, E)$ y costos $c \in \mathbb{R}^E$ pero son diferentes los objetivos.

    Problema de Matching Perfecto de Peso Mínimo

    Objetivo: hallar (si existe) un matching perfecto $M \subset E$ de G tal que $\sum_{e \in M} c_e$ es mínimo.

    Problema de Matching de Peso Máximo

    Objetivo: hallar un matching $M \subset E$ (no necesariamente perfecto) tal que $\sum_{e \in M} c_e$ es máximo.

13. El *Problema del Camino más Corto* consiste en, dado un digrafo $D = (V, A)$, costos $c \in \mathbb{R}^A$ y un vértice inicial $a \in V$, hallar los caminos dirigidos de costo mínimo entre a y el resto de los vértices de D, si tales caminos existen [LMDC, pág. 657; LCO, pág. 20]. Considere las siguientes dos variaciones de este problema:

    a) *Entrada*: Un grafo $G = (V, E)$, un vértice $a \in V$ y un vector de costos $c \in \mathbb{R}^E$.
    Objetivo: Encontrar los caminos que van desde a a todo otro vértice $v \in V \backslash \{a\}$ de costo mínimo.

    b) *Entrada*: Un grafo $G = (V, E)$, un vértice $a \in V$ y un vector de costos $c \in \mathbb{R}^V$.
    Objetivo: Encontrar los caminos que van desde a a todo otro vértice $v \in V \backslash \{a\}$ de costo mínimo, donde el costo de un camino es la suma de los costos de cada uno de sus vértices.

    Reduzca estos problemas alternativos al Problema de Camino más Corto original (es decir, describa cómo construir una instancia del Problema de Camino más corto que resuelva una instancia del problema alternativo). Demuestre que la reducción planteada efectivamente obtiene una solución óptima al problema alternativo.

14. Si una instancia del Problema del Camino más Corto tiene todos sus costos positivos, entonces puede resolverse con el *Algoritmo de Dijkstra* [LMDC, pág. 660; LCO, pág. 32]:

Paso 1. Etiquetamos el vértice a con $(0, -)$ y el resto de los vértices de D con $(+\infty, -)$. Para un vértice $v \in V$ nos referimos al primer valor de la etiqueta con $L(v)$ y que indica el costo de ir desde a hasta v (por el momento), mientras que el segundo valor de la etiqueta corresponde al vértice anterior que hay que visitar antes de alcanzar v.

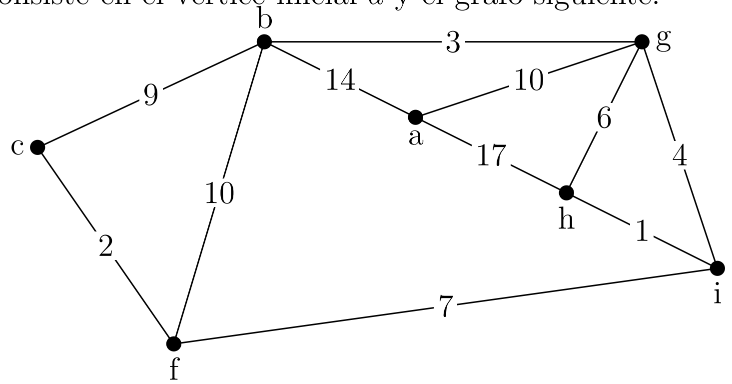
Paso 2. Hacemos $V_1 \leftarrow \{a\}$ y $V_2 \leftarrow \varnothing$. El primer conjunto contiene los vértices que faltan evaluar mientras que el segundo son los vértices ya evaluados.

Paso 3. Tomamos el vértice $v \in V_1$ tal que $L(v)$ es mínimo.

Paso 4. Para cada vecino no visitado $w \in N^+(v) \setminus V_2$, si el costo de alcanzar a w pasando por v es menor al costo actual, es decir $L(v) + c_{vw} < L(w)$, entonces actualizamos la etiqueta de w con $(L(v) + c_{vw}, v)$ y agregamos w a V_1.

Paso 5. Eliminamos v de V_1 y lo agregamos en V_2. Si $V_1 \neq \varnothing$, volvemos al Paso 3.

Usando este algoritmo (y lo visto en el Ejercicio anterior) resuelva la siguiente instancia que consiste en el vértice inicial a y el grafo siguiente:



15. De una instancia que muestre que el Algoritmo de Dijkstra devuelve una solución incorrecta cuando son permitidos costos negativos en los arcos.

16. En el contexto del Problema del Camino más Corto, decimos que $L(v)$ para todo $v \in V$ es un *potencial factible* cuando $L(v) + c_{vw} \geq L(w)$ para todo $vw \in A$ [LCO, pág. 20]. Pruebe que si $P = a, e_1, v_1, e_2, v_2, \ldots, e_r, b$ es un camino dirigido de a a b entonces $L(b)$ es una cota inferior del costo de P, $\sum_{i=1}^{r} c_{e_i}$.

17. Resuelva las instancias consideradas en los Ejercicios 14 y 15 del Problema del Camino más Corto con el *Algoritmo de Ford-Bellman* [LCO, pág. 29]:

Paso 1. Etiquetamos el vértice a con $(0, -)$ y el resto de los vértices de D con $(+\infty, -)$.

Paso 2. Hacemos $i \leftarrow 0$.

Paso 3. Hacemos $i \leftarrow i + 1$ y hubo_actualización $\leftarrow Falso$.

Paso 4. Para cada arco $vw \in A$, si $L(v) + c_{vw} < L(w)$ entonces actualizamos la etiqueta de w con $(L(v) + c_{vw}, v)$ y hacemos hubo_actualización $\leftarrow Verdadero$.

Paso 5. Si $i < |V|$ y hubo_actualización, volver al Paso 3.

Paso 6. En caso que $i = |V|$, existe un ciclo con costo negativo. Si no, L es un potencial factible que define los caminos más cortos.

18. Pruebe que el Algoritmo de Ford-Bellman es correcto (utilice lo visto en el Ejercicio 16).

19. Determine la complejidad de una implementación del Algoritmo de Ford-Bellman que asume que las siguientes operaciones tienen complejidad $O(1)$:

- Inicializar/actualizar la etiqueta de un vértice.
- Evaluar si un arco satisface $L(v) + c_{vw} < L(w)$.

Bibligrafía:

[LMDC] R. Grimaldi. *Matemática Discreta y Combinatoria.* 3ra. edición. Pearson.

[LCO] W. Cook, W. Cunningham, W. Pulleyblank A. Schrijver. *Combinatorial Optimization.* Wiley-Interscience.