Algoritmos de Aproximación

Clase 1
Introducción al curso y repaso de teoría de grafos

Pablo Romero

Lunes 7 de agosto de 2023. Montevideo, Uruguay.

Modo de Aprobación

Aprobación

- Asistir a clases y evaluar presentaciones (20 puntos de 100).
- Presentar un tema a elección (30 puntos de 100).
- Resolver la lista de problemas (50 puntos de 100).

Fechas importantes

- Seleccionar un tema para el oral antes del jueves 5 de octubre (durante el curso comentaremos posibles temas para elegir).
- Entregar los ejercicios resueltos antes del lunes 4 de diciembre (se darán sugerencias en clase para resolver los ejercicios).

Contenidos del Curso

Bloques

- Introducción al curso y repaso de teoría de grafos (Clase 1)
- Teorema fundamental de la combinatoria (Clase 2)
- Concepto de algoritmo de aproximación (Clase 3)
- Estructuras combinatorias (Clases 4 a 6).
- Esquema dual en programación lineal (Clases 7 a 10).
- Problemas abiertos y sugerencias para ejercicios (Clases 11-12)
- Presentaciones Orales.

Bibliografía

Seguiremos de cerca el contenido del libro de Vijay Vazirani, titulado *Approximation Algorithms*. En el Programa del curso se indica bibliografía complementaria.

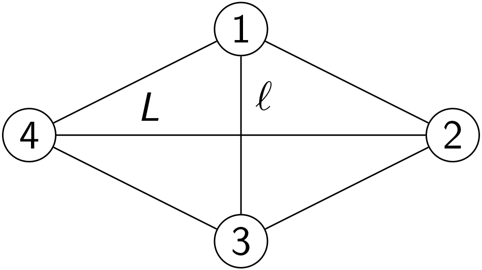
Motivación

Motivación

- Muchos problemas prácticos de ingeniería son problemas de optimización combinatoria (COPs) de alta complejidad.
- En pocos casos, el problema admite una resolución exacta (ej. relajación y programación lineal, exhaustividad).
- En otros problemas con estructuras similares a TSP, Knapsack, K-center u otros, se pueden emplear metaheurísticas o *algoritmos de aproximación*.
- Los algoritmos de aproximación *garantizan un factor de proximidad al óptimo global*.
- Tal factor de proximidad es el *peor caso*. En muchas situaciones, el algoritmo de aproximación va a producir una aproximación satisfactoria al problema de estudio.

Ejemplo de Problema de Optimización Combinatoria

Un delivery debe entregar 3 pedidos en tres hogares diferentes ubicados en los puntos 2, 3, y 4 y volver al local inicial, que se halla en el punto 1. Se pueden utilizar cualquiera de los 6 segmentos que determinan un rombo con diagonales ℓ y L tales que $\ell < L$. ¿Cómo debe ordenar la entrega de sus pedidos?



Ejemplo de Problema de Optimización Combinatoria

Veinte familias que integran una cooperativa estrenarán sus hogares. Cada familia $i \in \{1, 2, \ldots, 20\}$ va a puntuar del 1 al 20 a cada vivienda $\pi_i : \{V_1, V_2, \ldots, V_{20}\} \to \{1, 2, \ldots, 20\}$.
Se quiere hallar la asignación que maximiza la satisfacción total:

$$\max_{x_{ij}} s(x) = \sum_{i=1}^{10} \pi_i(j) x_{ij}$$

$$s.a. \sum_{j=1}^{20} x_{ij} = 1, \forall i \in \{1, 2, \ldots, 20\}$$

$$\sum_{i=1}^{20} x_{ij} = 1, \forall j \in \{1, 2, \ldots, 20\}$$

$$x_{ij} \in \{0, 1\} \forall i, j \in \{1, 2, \ldots, 20\}$$

¿Cuántas soluciones óptimas existen? ¿Y si repartimos al azar?

Grafos y caminos

Grafos y caminos

- Un *grafo* es un par $G = (V, E)$, donde V es finito y $E \subseteq \binom{V}{2}$.
- Dos vértices u y v de V son *adyacentes* si $\{u, v\} \in E$. La arista $e = \{u, v\}$ *incide* en los vértices u y v.
- La cantidad de aristas de G que inciden en un vértice v es el *grado de v en G* y se denota $gr_G(v)$.
- Un *camino* en G es una secuencia ordenada $v_0, v_1, \ldots, v_n$ de vértices de G tales que $\{v_i, v_{i+1}\} \in E$ para cada $i \in \{0, 1, \ldots, n-1\}$. Si $v_0 = v_n$ el camino es *cerrado*; en caso contrario es *abierto*. Un *camino simple* es un camino que no repite vértices. Un *recorrido* es un camino que no repite aristas. Un *circuito* es un recorrido cerrado.
- Un circuito es *euleriano* si contiene a todas las aristas de G.

Grafos conexos

Sea $G = (V, E)$ un grafo. Decimos que el vértice u alcanza a v en G si existe un camino en G con vértice inicial u y vértice final v.

Grafos conexos

- La relación de alcanzabilidad entre los vértices de G es de equivalencia.
- La clases de equivalencia de la relación de alcanzabilidad en G se llaman *componentes conexas de* G.
- Un grafo es *conexo* si tiene una sola componente conexa.
- La *distancia* $d_G(u, v)$ entre u y v de G es la cantidad de aristas del camino más corto entre u y v.
- El *diámetro* $d(G)$ de un grafo G es la máxima distancia entre todo par de vértices de G: $d(G) = \max_{u,v \in V}\{d(u, v)\}$.

Subgrafos e isomorfismos

Sean $G = (V, E)$ y $G' = (V', E')$ dos grafos.

Sugrafos e isomorfismos

- Los grafos G y G' son *isomorfos* si existe una función biyectiva $\varphi : V \to V'$ tal que $x, y \in E$ si y sólo si $\{\varphi(x), \varphi(y)\} \in E'$.
- El grafo G' es un *subgrafo de* G si $V' \subseteq V$ y $E' \subseteq E$.
- El subgrafo G' de G es *generador* si $V' = V$.
- Dado un vértice v de G, el subgrafo $G - v$ se obtiene de G quitando el vértice v y todas las aristas incidentes a v.
- El subgrafo G' es *inducido* si se obtiene de G quitando una cantidad finita de vértices de V.

Grafos hamiltonianos

Sugrafos e isomorfismos

- Sea n un entero positivo tal que $n \geq 3$. El *n-ciclo* es el grafo cuyos vértices son $\{v_1, v_2, \ldots, v_n\}$ y sus aristas son $\{v_1, v_n\}$ y $\{v_i, v_{i+1}\}$ para cada $i \in \{1, 2, \ldots, n-1\}$.
- Un grafo es un *ciclo* si es un n-ciclo para algún entero positivo n tal que $n \geq 3$.
- Un grafo es *acíclico* si no tiene a ningún ciclo como subgrafo.
- Un grafo G con n vértices es *hamiltoniano* si tiene al menos un n-ciclo.

Caracterización de Árboles

Defínción 1.

Si G es un grafo conexo, entonces una arista e de G es un puente si $G - e$ no es conexo.

Defínción 2.

Un grafo T es un árbol si es conexo y no tiene ciclos.

Teorema 1.

Sea $G = (V, E)$ un grafo. Cada una de las siguientes afirmaciones son equivalentes:

1. *G es un árbol.*
2. *G es un grafo conexo tal que $|E| = |V| - 1$.*
3. *G es un grafo sin ciclos tal que $|E| = |V| - 1$.*
4. *G es un grafo conexo donde todas sus aristas son puentes.*

Caracterización de Grafos Bipartitos

Defición 3.

Un grafo $G = (V, E)$ es bipartito si existen dos subconjuntos V_1 y V_2 tales que $V_1 \cup V_2 = V$, $V_1 \cap V_2 = \emptyset$ y cada arista e de E tiene exactamente un extremo en V_1 y extactamente un extremo en V_2.

Teorema 2.

Un grafo es bipartito si y sólo si no posee ciclos impares.

Ejemplos

- Todos los árboles son grafos bipartitos.
- El grafo C_n es bipartito si y sólo si n es par.
- El grafo P_n es todo él un camino simple de n vértices. Como es un árbol, es bipatito.

Caracterización de Grafos Eulerianos

Defninción 4.

Un multigrafo es un par $G = (V, E)$ donde E es un multiconjunto de pares de elementos de V.

Defninción 5.

Un grafo es euleriano si tiene algún circuito euleriano.

Teorema 3.

Un grafo o multigrafo conexo G es euleriano si y sólo si todos sus vértices tienen grado par.

Observación

Si a partir de grafo conexo $G = (V, E)$ construimos un multigrafo $G' = (V, E')$ *duplicando* cada una de las aristas de E, entonces el multigrafo resultante es euleriano.

Problemas en Grafos

Problemas en $\mathcal{P}$ que utilizaremos en el curso

1. Camino mínimo entre dos vértices de un grafo (Dijkstra).
2. Árbol generador de costo mínimo (Kruskal, 1956).
3. Reconocimiento de grafos bipartitos (DFS).
4. Construcción de circuitos eulerianos (Euler, 1736).
5. Emparejamiento de peso mínimo en grafos (Edmonds, 1965).

Otros problemas históricos en $\mathcal{P}$

- Flujo máximo entre dos vértices en una red (FF, 1956).
- Cantidad de árboles generadores de un grafo (Kirchhoff,1847).
- Emparejamiento de peso mínimo en bipartitos (Kuhn, 1955).

Problemas que estudiaremos en el curso

Los siguientes problemas se seleccionan especialmente para ilustrar conceptos y aplicaciones de algoritmos de aproximación.

Problemas que estudiaremos en el curso

1. Cubrimiento de vértices y de conjuntos (MCVC y SC).
2. Problema de Steiner en árboles (STP).
3. Problema del vendedor ambulante (TSP).
4. Diseño de subred 2-conexa de mínimo costo (MWTCSN)
5. Algunas generalizaciones del problema de Steiner.

Temas adicionales

Los siguientes temas son aspectos a profundizar que bien pueden ser elegidos para que presenten al final del curso.

Temas adicionales

1. Rendimiento del algoritmo goloso.
2. Problema del mochilero y aproximabilidad polinomial.
3. Método de capas y aplicaciones (secuenciación de ADN).
4. Corte de máxima capacidad en redes (MAX-CUT).
5. Aplicaciones del esquema primal-dual en PL.