MA4701. Optimización Combinatorial. 2013.
Profesor: José Soto
Escriba(s): Francisco Arana y Felipe Muñoz.
Fecha: 19 de Agosto de 2013.

Cátedra 1

1. Introducción

1.1. Problemas en Optimización Combinatorial

Entendemos por *Optimización Combinatorial* a la rama de la matemática centrada en el diseño y análisis de algoritmos con el fin de encontrar objetos óptimos en conjuntos finitos. Más formalmente se considera un conjunto finito OBJ y una función $f\colon \mathrm{OBJ} \to \mathbb{R}$, y se buscan los elementos $x \in argmax(f)$ ó $x \in argmin(f)$.

Considerando problemas de la forma anterior, es claro que estos siempre tiene solución dada la finitud del conjunto OBJ. Pese a esto, un problema clave que aparece es el hecho que $|\mathrm{OBJ}|$ es enorme. El gran tamaño del conjunto OBJ nos llevará a describirlo de forma implícita.

Ejemplo 1 (Problema de emparejamiento o matching de peso máximo)**.** De manera intuitiva se considera un hotel con n habitaciones y $2n$ huéspedes, donde el conjunto de huéspedes se denotará por A. Consideramos a su vez una función de peso $w\colon \binom{A}{2} \to \mathbb{R}^+$ positiva, esto es $w(\{a,b\}) \geq 0$ para todo $\{a,b\} \in \binom{A}{2}$. En este contexto se define el concepto de matching como sigue.

Definición 1 (Matching)**.** En el contexto anterior, se dice que $M \subseteq \binom{A}{2}$ es un *matching* en A ssi M es un conjunto de pares disjuntos de elementos de A.

El objetivo del presente problema es encontrar un matching de tamaño n que maximice el peso total, esto es resolver

$$\max_{M \text{ matching}, |M|=n} w(M),$$

donde se define la función de peso total del matching M como

$$w(M) = \sum_{\{i,j\} \in M} w(\{i,j\}).$$

Ejemplo 2 (Problema del Vendedor Viajero)**.** Intuitivamente consideramos un conjunto de ciudades $A \subseteq \mathbb{Q}^2$. Consideramos a su vez un vendedor, ubicado en alguna ciudad de A, que quiere visitar todas las ciudades. Para ir desde la ciudad a a la ciudad b con $a, b \in A$, el vendedor debe recorrer una distancia $d(a,b)$. El objetivo del problema es encontrar el *tour* que visite cada ciudad y vuelva al punto de origen, de largo total mínimo.

Ejemplo 3 (Problema de Conectividad de Costo Mínimo)**.** Intuitivamente consideramos un conjunto de servidores A, y una función de costos $c\colon \binom{A}{2} \to \mathbb{R}^+$ positiva. El objetivo del presente problema es encontrar una configuración de cables que contenga un camino entre cada par de servidores, de costo total mínimo.

Durante el curso se trabajará constantemente con uniones de conjuntos, intersecciones de conjuntos y otras operaciones por lo cual resulta útil introducir notación abreviada.

Definición 2 (Notación)**.** Se expone a continuación la notación a usar durante el curso

1. $[n] = \{1, 2, \ldots, n\}$.
2. $\binom{A}{k} = \{X \subseteq A \mid |X| = k\}$.
3. $2^A = \mathcal{P}(A)$.
4. $A + x = A \cup \{x\}$.
5. $A - x = A \backslash \{x\}$.

1.2. Fuerza Bruta

Como primer ejemplo, en el contexto del tipo de problemas a resolver discutido en la sección anterior, entederemos por *fuerza bruta* al método de solución de problemas que consiste en probar todas las posibles soluciones. Si bien este método siempre se puede emplear para encontrar soluciones a los problemas de estudio, presenta claras dificultades en cuanto puede tardarse mucho tiempo en encontrar las soluciones, pues como se mencionó anteriormente, los conjuntos estudiados son en general de gran tamaño. Observemos este hecho con un par de ejemplos.

En el contexto del problema de emparejamiento de peso máximo ¿Cuántos matching M en A con $|M| = n$ existen? Consideremos que $|A| = 2n$. Para codificar los matching podemos usar la siguiente representación matricial

$$M = \begin{pmatrix} a_1 & a_2 & \cdots & a_n \\ b_1 & b_2 & \cdots & b_n \end{pmatrix}.$$

Notemos que existen $(2n)!$ matrices de la forma anterior. Más aún cada matching se puede representar matricialmente de $n! \cdot 2^n$ formas distintas, donde el factor $n!$ emana de las posibles permutaciones entre columnas de la matriz, y el factor 2^n emana de las posibles permutaciones de elementos en una columna. Con esto el número total de matchings es

$$\#_{\text{matchings}} = \frac{(2n)!}{n! \cdot 2n} = \frac{(2n)(2n-1)\cdots(n+1)}{2^n} \geq \left(\frac{n}{2}\right)^n.$$

En particular tomando $n = 20$ obtenemos que

$$\#_{\text{matchings}} \geq 10^{20}.$$

Si revisamos 10^9 matchings por segundo nos demoraremos

$$T = 10^{11} \text{ seg} \approx 3000 \text{ años}.$$

Como segundo ejemplo, en el contexto del problema del vendedor viajero ¿Cuántos tour existen? Para responder esta pregunta basta biyectar los posibles tours con las permutaciones de $[n]$ con lo cual

$$\#_{\text{tours}} = n!$$

Esto siempre y cuando distingamos los tours por punto de inicio. Utilizando la aproximación de Strirling para valores grandes de n dada por

$$n! \approx \sqrt{2\pi n}\left(\frac{n}{e}\right)^n$$

obtenemos que para $n = 22$ el número total de posibles tours es

$$\#_{\text{tours}} \approx 10^{20}.$$

1.3. Programación Lineal y Relajación

Otra forma de abordar los problemas de Optimización Combinatorial es transformar estos problemas a problemas de programación lineal equivalentes y aplicar relajaciones posteriores al problema obtenido. Sin embargo, las relajaciones aplicadas pueden introducir distorciones en cuanto las soluciones obtenidas no cumplen con la restricciones naturales de ciertas variables que deben tomar valores enteros. Se expone a continuación un ejemplo de este procedimiento y de los problemas que surgen de aplicarlo.

Ejemplo 4 (Planteamiento del Problema de Matching como PL)**.** Tenemos como objetivo plantear el problema de matching como un PL equivalente. Para esto consideramos para cada $\{i,j\} \in \binom{A}{2}$ la indicatriz

$$X_{i,j} = \begin{cases} 1 & \text{si } \{i,j\} \in M, \\ 0 & \text{si } \{i,j\} \notin M \end{cases}$$

Luego planteamos el problema lineal equivalente

$$\begin{array}{lll} (P) & \max\limits_{(X_{i,j}) \in \mathbb{R}^{\binom{2n}{2}}} & \sum\limits_{\{i,j\} \in \binom{A}{2}} X_{i,j} \cdot w_{i,j} \\ & \text{s.a} & (\forall i \in A) \colon \sum\limits_{j \neq i} X_{i,j} = 1 \\ & & \left(\forall \{i,j\} \in \begin{pmatrix} A \\ 2 \end{pmatrix}\right) : X_{i,j} \in \{0,1\} \end{array}$$

Donde la primera restricción representa el hecho que todo elemento de A puede estar apareado con un único elemento, distinto de él, en el matching y la segunda restricción representa el hecho de que la indicatriz de que dos elementos de A estén apareados puede tomar sólo los valores 0 ó 1. Es posible relajar este problema para obtener un problema continuo cambiando la última restricción por

$$\left(\forall \{i,j\} \in \begin{pmatrix} A \\ 2 \end{pmatrix}\right) : 0 \leq X_{i,j} \leq 1.$$

Si bien esta relajación puede resultar útil, existen casos patológicos donde las soluciones obtenidas no tienen sentido en el contexto original del problema. Como ejemplo consideremos la representación para un problema de Matching en base a la figura expuesta más abajo, donde los nodos corresponden a los *huéspedes* y los pesos de emparejamiento para los pares conectados por linea continua es 1 mientras que para los que no están conectados por linea continua es 0.

En este contexto es facil notar que cualquier solución óptima de emparejamiento, para el problema entero original, alcanza un peso total igual a 2. Pese a esto, una solución del problema relajado consiste en dar valor $\frac{1}{2}$ a cada una de las variables $X_{i,j}$ para pares que están conectados con linea punteada. En este caso se obtiene un peso total de 2,5; mejorando la solución anterior. Pese a ello, esta solución carece de sentido en el problema original pues las variables $X_{i,j}$ deben tomar valores 0 ó 1.

2. Grafos

2.1. Definiciones

Antes de comenzar a trabajar debemos formalizar las estructuras que manejaremos en el curso.

Definición 3 (Grafo simple)**.** Un grafo simple G es un par (V,E), donde $E \subseteq \binom{V}{2} = \{\{u,v\} : u,v \in V, u \neq v\}$.
Los elementos $v \in V$ se denominan vértices y los $e \in E$ aristas. Denotaremos una arista por $e = \{u,v\}$ o bien $e = uv = vu$. Además diremos que los vértices u y v de una arista e se denominan extremos de e.

De manera natural podemos extender esta definición a un contexto más general.

Definición 4 (Multigrafo)**.** Un grafo G es un par (V,E), donde a cada arista $e \in E$ se le asigna un conjunto de extremos que puede ser $\{u,v\} \subseteq \binom{V}{2}$ o un síngleton $\{v\}$, con $v \in V$.

Diremos que las aristas de un solo extremo serán llamadas *bucles* o *loops* y las aristas con extremos iguales se denominarán *paralelas*. Por comodidad, pero abusando de la notación, escribiremos $e = \{u\}$ para el bucle de extremo u y $e = uv$ para una arista de extremos u y v.

Observación 1. Un grafo sin bucles ni aristas paralelas es un grafo simple.

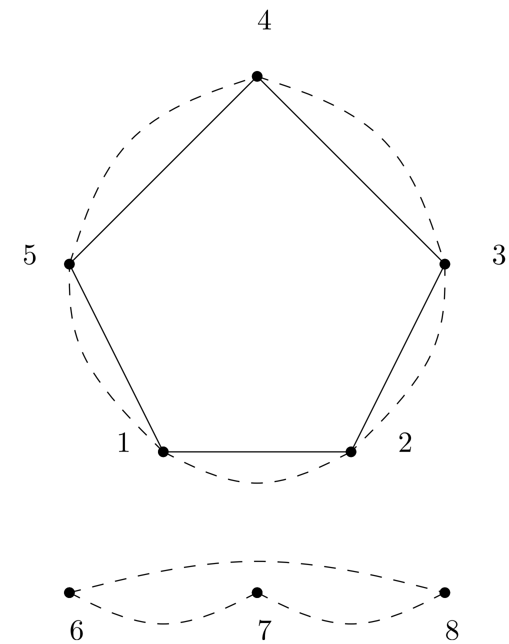

Figura 1: Representación de Problema de Matching

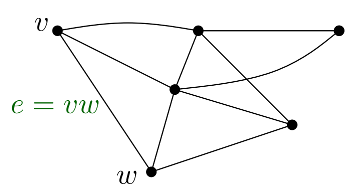

Figura 2: Un grafo simple

Dado un grafo G podemos asociar los siguientes conceptos a él:

- Una arista e es *incidente* a un vértice v, si v es extremo de e.
- Dos aristas son *adyacentes* si son incidentes a un vértice común.
- Dos vértices u y v son *adyacentes* o *vecinos* si existe una arista de extremos u y v.
- El *órden* de un grafo es $|V|$ y solemos denotar por $n = |V|$.
- El *tamaño* de un grafo es $|E|$ y solemos denotar por $m = |E|$.

Antes de continuar es necesario establecer algunas convenciones que se utilizarán a lo largo del curso. A no ser que se diga lo contrario, cuando decimos “grafo” nos referimos a un grafo simple. Por otra parte, si $G = (V, E)$ escribimos $V = V(G)$ y $E = E(G)$, por lo que decimos que G es un grafo con vertices en V y aristas en R. Finalmente dentro del contexto que nos interesa estudiar, asumiremos que $|E|, |V| < \infty$.

A continuación listaremos una serie de ejemplos para clarificar la idea de grafo.

Ejemplo 5.

- El grafo trivial: $(\emptyset, \emptyset)$

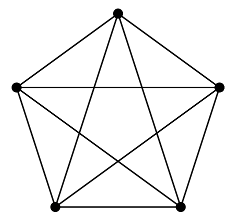

Figura 3: El grafo completo K_5

- El grafo completo: $K_n = \left([n], \binom{[n]}{2}\right) = (\{1,\ldots,n\}, \{ij : i < j\})\,.$
- El complemento del grafo completo: $\overline{K_n} = ([n], \emptyset)$.
- Dada una relación simétrica $\sim$ en V, podemos definir:

$$G_\sim = (V, \{uv : u \neq v, u \sim v\})\,.$$

2.2. Conjuntos notables en un grafo

En lo que sigue definiremos algunos conjuntos especiales que surgen al estudiar los grafos y que nos serán de gran utilidad.

Definición 5. Dados $U, W \subseteq V$, $U \cap W = \emptyset$, $F \subseteq E$ podemos definir:

$$F[U,W] = \{e = uv \in F : u \in U, v \in W\}.$$

Definición 6. Si $v \in V, U \subseteq V$ y $F \subseteq E$ definimos:

- $\delta_F(v) = \{e \in F : e \text{ es incidente a } v\} = F[\{v\}, V\backslash\{v\}]$.
- $\delta_F(U) = \{e \in F : e \text{ es incidente a algún } v \in U \text{ y a algún } v \notin U\} = F[U, V\backslash U]$.
- $N_F(v) = \{w \in V\backslash\{v\} : \exists e \in F, e = vw\}$.
- $N_F(U) = \{w \in V\backslash U : \exists v \in U, \exists e \in F, e = vw\}$.

Cuando $F = E$ omitiremos el subíndice en las definiciones anteriores.

La cantidad $d(v) = |\delta(v)| = |N(v)|$ se conoce como el *grado* de v.

Definición 7 (Subgrafo)**.** Un grafo $G' = (V', E')$ es *subgrafo* de $G = (V, E)$ si $V' \subseteq V$ y $E' \subseteq E$. En tal caso escribiremos $G' \subseteq G$.

Observación 2. Como G' es grafo, se deduce también que $E' \subseteq E \cap \binom{V'}{2}$.

Definición 8. Dado $U \subseteq V$, el subgrafo de G *inducido* por U, es el grafo dado por:

$$G[U] = (U, \{e = uv \in E : u, v \in U\})\,.$$

Definición 9. Dado $F \subseteq E$, el grafo obtenido al borrar F de G es:

$$G\backslash F = (V, E\backslash F)\,.$$

De manera similar, dado $U \subseteq V$, el grafo obtenido al borrar U de V es:

$$G\backslash U = G[V\backslash U].$$

2.3. Paseos, caminos y ciclos

Definición 10. Un *paseo* es una secuencia:

$$P = v_0 e_1 v_1 e_2 v_2 e_3 v_3 \cdots e_k v_k,$$

donde $v_i \in V, e_i \in E, e_i = v_{i-1}v_i$. Diremos que el largo del paseo es el número de aristas que este contiene.

Observación 3. Un paseo se dice *arista-simple* si todas las aristas que contiene son distintas.

Definición 11 (Camino/ciclo)**.** Dado un paseo P, si todos los vértices v_i que lo componen son distintos, entonces se dice que P es un camino entre v_0 y v_k. En el caso en que son distintos salvo sus extremos, es decir $v_0 = v_k$, diremos que P es un *ciclo.*

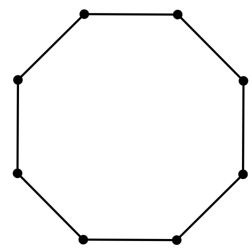

Figura 4: Un ciclo de largo 8

Observación 4. Por abuso de notación, al conjunto $\{e_i\}_{i=1}^k$ también se le llama camino o ciclo. De la misma manera, llamamos camino o ciclo al grafo $\left(\{v_i\}_{i=0}^k, \{e_i\}_{i=1}^k\right)$.

2.4. Conectividad y generación

Definición 12 (Conexidad)**.** Un grafo $G = (V, E)$ se dice conexo si para cada $u, v \in V$ existe un paseo entre u y v, o de manera equivalente un camino.

Definición 13. Una *componente conexa* de G es un conjunto maximal de vértices W tal que $G[W]$ es conexo.

Definición 14. Sean $F \in E$ y $U \in V$. Decimos que F *genera* a U si para todo vértice $u \in U$, existe una arista $e \in F$ incidente a u. También decimos que un grafo G' genera a U si $E(G')$ genera a U.

Observación 5. Un subgrafo $G' \subseteq G$ es generador si G' genera a $V(G)$.