MA3705. Algoritmos Combinatoriales. 2014.
Profesor: José Soto
Escriba(s): Juan Pablo Donoso, Felipe Garrido y Juan Granier.
Fecha: 27 de Octubre 2014.

Cátedra 22

1. Recuerdo

Estabamos resolviendo el problema siguiente: Dado un grafo no dirigido conexo $G = (V, E)$, dotado de una función de peso $w : E \to \mathbb{R}_+$, queremos encontrar un conjunto de aristas F de peso mínimo que desconecte el grafo, es decir:

$$F \in \arg\min\{w(F) : (V, E \setminus F) \text{ es disconexo}\} \qquad (1)$$

Si bien se podía resolver con Edmond-Karp, esto nos llevó a una solución $O(n^2m^2)$. Para encontrar una solución más eficiente, se introdujo el concepto de función submodular y s-t separador.

Definición 1 (Separador de dos elementos)**.** $S \subset V$ es un s-t separador si $|S \cap \{s, t\}| = 1$

Definición 2 (Trío factible)**.** Sean $s, t \in V$ y $S \subset V$ un s-t separador. El trío (S, s, t) se dirá *trío factible* si es el mejor separador de s y t, es decir, si $f(S) \leq f(X)$ para todo X s-t separador.

Definición 3 (Función submodular)**.** $f : 2^V \to \mathbb{R}_+$ se dice *submodular* si para todo $A, B \subset V$

$$f(A) + f(B) \geq f(A \cup B) + f(A \cap B)$$

diremos además que f es *simétrica* si para $A \subset V$

$$f(A) = f(V \setminus A)$$

Ejemplo 1. *Dado $G = (V, E)$ y w como antes, podemos tomar $f : 2^V \to \mathbb{R}_+$ como*

$$f(A) = w(\delta(A))$$

Por último, para resolver el problema enunciado arriba mediante funciones submodulares simétricas, necesitamos resolver el problema siguiente

$$\min\{f(X) : \emptyset \subsetneq X \subsetneq V\} \qquad (2)$$

para f una función submodular y simétrica.

Observación 1. *Notemos que el problema interesante es tomar inclusiones estrictas, pues siempre se tiene que, para todo $X \subset V$ y f submodular simétrica*

$$\begin{aligned} 2f(X) &= f(X) + f(X) \\ &\geq f(\emptyset) + f(V \setminus X) \\ &= 2f(\emptyset) \end{aligned}$$

La última cátedra vimos que, asumiendo que somos capaces de encontrar un trío factible, entonces el siguiente algoritmo resuelve (2).

Algorithm 1 Minimizar f submodular y simétrica

1: $U \leftarrow \emptyset$
2: $f' \leftarrow f$
3: **while** (existe (S, s, t) trío factible para f') **do**
4: $\quad U \leftarrow U + S'$ donde S' se obtiene expandiendo los elementos de S.
5: $\quad f' \leftarrow$ función fusión de s y t
6: **end while**
7: **return** $S \in \arg\min\{f(X) : X \in U\}$

Donde en el algoritmo anterior usamos la fusión de elementos:

Definición 4 (Fusión de elemtentos)**.** Sea $f : 2^V \to \mathbb{R}$ submodular, y sean s y t en V. La función $f_{st} : 2^{V-s-t+st} \to \mathbb{R}$ es la *fusión* de los dos elementos s y t, dada por:

$$f_{st}(X) = \begin{cases} f(X) & \text{si } st \notin X \\ f(X \cup \{s,t\} - st) & \text{si } st \in X \end{cases}$$

para todo $X \subset V - s - t + st$.

Ahora veremos una manera de encontrar un trío factible, mediante el concepto de par colgante y el Teorema de Queyranne.

2. Par colgante y Teorema de Queyranne

2.1. Par Colgante

En adelante consideraremos f una función submodular y simétrica.

Definición 5 (Par colgante de f)**.** Un par ordenado (s,t) tal que $(\{s\}, s, t)$ es trío factible se dice *par colgante*.

Mader (1978) demostró que toda función submodular simetrica tiene un par colgante. Queyranne (1998) dió un algoritmo para encontrarlo. Así, en vez de demostrar Mader, demostraremos Queyranne. Para esto necesitaremos la *bifunción de adyacencia*.

Definición 6 (Bifunción de adyacencia)**.** Dada $f : 2^V \to \mathbb{R}_+$, la *bifunción de adyacencia* $d : 2^V \times 2^V \to \mathbb{R}_+$ esta dada por

$$d(A,B) = \frac{1}{2}(f(A) + f(B) - f(A \cup B))$$

Ejemplo 2. *Si $f = w(\delta(\cdot))$, entonces $d(A,B)$ es el peso de todas las aristas que van desde A hasta B.*

Lema 1. *$d(\cdot,\cdot)$ satisface lo siguiente*

1. *Es simétrica, i.e. para todo $A, B \subset V$:*
$$d(A,B) = d(B,A).$$
2. *$f(A) = d(A, V \setminus A) + \frac{1}{2} f(\emptyset)$.*
3. *d es monótona en el siguiente sentido: Para $A, B, C \subset V$ disjuntos*
$$d(A,B) \leq d(A, B \cup C).$$
4. *d es consistente: Para $A, B, C \subset V$ disjuntos*
$$d(A,B) \leq d(A,C) \implies d(A \cup C, B) \leq d(A \cup B, C).$$

Demostración Lema1.

1. Directo.
2. Sea $A \subset V$.
$$\begin{aligned} d(A, V \setminus A) &= \frac{1}{2}(f(A) + f(V \setminus A) - f(V)) \\ &= f(A) - \frac{1}{2} f(\emptyset) \qquad (f(A) = f(V \setminus A) \ \wedge \ f(V) = f(\emptyset)) \end{aligned}$$
de donde se concluye.

3. En efecto:

$$\begin{aligned}2(d(A,B\cup C)-d(A,B)) &= f(A)+f(B\cup C)-f(A\cup B\cup C)-(f(A)+f(B)-f(A\cup B))\\ &= f(B\cup C)+f(A\cup B)-f(B)-f(A\cup B\cup C)\\ &\geq 0\end{aligned}$$

donde el último paso se tiene por submodularidad.

4. Hacemos la misma idea que en (3)

$$\begin{aligned}2(d(A\cup B,C)-d(A\cup C,B)) &= f(A\cup B)+f(C)-\cancel{f(A\cup B\cup C)}-f(A\cup C)-f(B)+\cancel{f(A\cup B\cup C)}\\ &= f(C)-f(A\cup C)-f(B)+f(A\cup B)\\ &= f(A)+f(C)-f(A\cup C)-(f(A)+f(B)-f(A\cup B))\\ &= 2(d(A,C)-d(A,B))\\ &\geq 0\end{aligned}$$

□

Para poder utilizar la bifunción de adyacencia, es necesario saber que significa que (s,t) sea un par colgante de f. Tenemos lo siguiente:

$$\begin{aligned}(s,t) \text{ es par colgante de } f &\iff \forall\ S\ s\text{-}t \text{ separador, } f(\{s\})\leq f(S)\\ &\iff \forall\ S\ s\text{-}t \text{ separador, } d(\{s\},V-s)\leq d(S,V\setminus S)\end{aligned}$$

2.2. Teorema de Queyranne

Mader (1978) probó que toda función submodular y simétrica tiene un par colgante, de manera no constructiva. El año 1998, Queyranne dio un algoritmo para encontrar un par colgante.

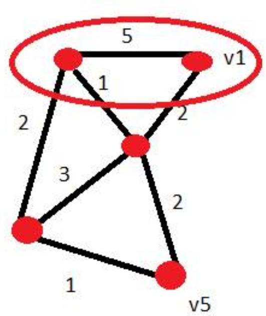
Teorema 1. *Supongamos que V se ordena de forma $\{v_1,...v_n\}$, de modo que para todo $i\geq 2$*

$$d(\{v_1,...,v_{i-1}\},\{v_i\})\geq d(\{v_1,...,v_{i-1}\},\{v_j\}),\forall j>i$$

Entonces, el último par, (v_n,v_{n-1}) corresponde a un par colgante.

Observación 2. *Este orden se llama Orden de Máxima Adyacencia.*

Ejemplo 3. *Consideremos el siguiente grafo con pesos.*



Consideremos v_1 arbitrario. Se elige v_2 de modo que $d(v_1,v_2)$ sea máximo. Luego se elegirá v_3 de modo que $d(\{v_1,v_2\},\{v_3\})$ sea máximo, y así sucesivamente.

Queyranne nos dice entonces, que (v_5,v_4) es par colgante.

Demostremos el Teorema de Queyranne.

Demostración Teorema de Queyranne. Utilizamos inducción sobre $|V| = n$.

Caso $n = 2$: Tenemos que $V = \{v_1, v_2\}$. Luego, los singletons son los únicos $v_1 - v_2$ separadores pues notemos que

$$d(\{v_1\}, V \setminus \{v_2\}) = d(\{v_1\}, \{v_2\}) = d(\{v_2\}, \{v_1\})$$

Luego, no hay nada que probar.

Caso $n = 3$: Sea v_1, v_2, v_3 un orden de máxima adyacencia. Debemos ver que (v_3, v_2) es par colgante. Sea X un v_3-v_2 separador, cualquiera. Probaremos que

$$d(\{v_3\}, \{v_1, v_2\}) \leq d(X, V \setminus X)$$

Pero en realidad, basta probarlo para $X = \{v_3, v_1\}$, pues es el único v_1-v_3 separador distinto a (v_3, v_2).

Veamos entonces que

$$d(\{v_3\}, \{v_1, v_2\}) \leq d(\{v_3, v_1\}, \{v_2\})$$

En efecto, como el orden es de máxima adyacencia, $d(v_1, v_2) \geq d(v_1, v_3)$, y luego se concluye por consistencia.

Caso $n \geq 4$: Sea $v_1, ..., v_n$ orden de máxima adyacencia. Sea X un v_n-v_{n-1} separador. Separaremos en 3 casos.

Caso 3.1: X no separa v_1 de v_2. Definamos entonces $V' = V - v_1 - v_2 + v_1v_2$, la contracción de los vértices v_1 y v_2. Dado $X' \subseteq V'$, definimos

$$expandir(X') = \begin{cases} X \text{ si } v_1v_2 \notin X \\ X - v_1v_2 + v_1 + v_2 \text{ si } v_1v_2 \in X \end{cases}$$

De este modo, definimos

$$f'(X') = f(expandir(X'))$$

Es fácil ver que f' es simétrica y submodular. Además, su bifunción de adyacencia d' viene dada por

$$d'(A, B) = d(expandir(A), expandir(B))$$

Como el orden inicial era de máxima adyacencia para d, el orden $v_1v_2, v_3, ..., v_n$ es de máxima adyacencia para d' y tenemos que la cantidad de vértices es $n-1$. Luego, por hipótesis inductiva, (v_n, v_{n-1}) es par colgante de d'. Pero notemos que

$$d'(v_n, V' - v_n) \leq d'(X', V' \setminus X')$$

y $expandir(X') = X$, luego se obtiene, por definición de expandir, que

$$d(v_n, V - v_n) \leq d(X, V \setminus X)$$

y se concluye el primer caso.

Caso 3.2: X no separa v_2 de v_3. Hacemos la contracción de v_2 y v_3 y definimos $V' = V - v_2 - v_3 + v_2v_3$ además de definir f' y d' de forma análoga al caso anterior.

Veamos que $v_1, v_2v_3, v_4, ..., v_n$ es de máxima adyacencia para d'. Aprovechándose de que el antiguo orden era de máxima adyacencia para d y de la definición de expandir, solo falta chequear que

$$d'(v_1, v_2v_3) \geq d'(v_1, v_k), \forall k \geq 4$$

En efecto, por la monotonía de d y por su máxima adyacencia,

$$d'(v_1, v_2v_3) = d(v_1, \{v_2, v_3\}) \geq d(v_1, v_2) \geq d(v_1, v_k) = d'(v_1, v_k), \forall k \geq 4$$

Luego, de la misma forma que en el caso 3.1 se concluye que (v_n, v_{n-1}) es par colgante de d.

Caso 3.3: X no separa v_1 de v_3. Fusionamos entonces a v_1 con v_3 como lo hemos hecho en los casos anteriores. Veamos que $v_2, v_1v_3, v_4, ..., v_n$ es orden de máxima adyacencia para d'.

Debemos ver que

$$d'(v_2, v_1v_3) \geq d'(v_2, v_k), \forall k \geq 4$$

O equivalentemente

$$d(v_2, \{v_1, v_3\}) \geq d(v_2, v_k), \forall k \geq 4$$

En efecto, por máxima adyacencia de d, $d(v_1, v_2) \geq d(v_1, v_3)$, luego por consistencia

$$\begin{aligned} d(\{v_1, v_3\}, v_2) &\geq d(\{v_1, v_2\}, v_3) \\ &\geq d(\{v_1, v_2\}, v_k), \forall k \geq 4 \\ &\geq d(v_2, v_k), \forall k \geq 4 \end{aligned}$$

donde la segunda desigualdad se obtiene por máxima adyacencia y la tercera por monotonía.

Se concluye entonces al igual que en los casos anteriores que (v_n, v_{n-1}) es par colgante de d', y luego, es par colgante de d.

Gracias a que los casos 3.1, 3.2 y 3.3 cubren todos los escenarios posibles de X, se concluye que (v_n, v_{n-1}) es par colgante. $\square$

A continuación escribimos el algoritmo de Queyranne en su forma recursiva e iterativa.

3. Algoritmo de Queyranne

Algorithm 2 ALG (Queyrranne) para minimizar f (Versión Recursiva)

1: **if** $|V| = 2, (V = \{\, s, t\})$ **then**
2:     devolver s
3: **else**
4:     **if** $|V| \geq 3$ **then**
5:         Encontrar par colgante (s, t) de f
6:         Fusionar s y t para obtener f'
7:         $X' \leftarrow \text{ALG}(f')$
8:     **end if**
9: **end if**

Presentamos también una versión iterariva del algoritmo.

Algorithm 3 ALG (Queyranne) para minimizar f (Versión Iterativa)

1: cand $\leftarrow \emptyset$
2: $V' \leftarrow V, f' \leftarrow f$
3: **while do** $|V'| \geq 2)$
4:     Encontrar par adyacente (s', t') de f'
5:     cand $\leftarrow$ cand $\cup \{\text{expandir}(S')\}$
6:     Fusionar s' y t' y actualizar f' y v'
7: **end while**
8: Devolver el mejor candidato, $X \in \arg\min\{f(X) : X \in \text{cand}\}$

Estudiamos la complejidad del Algoritmo de Queyranne
Complejidad: En total es $O(n^3)$ operaciones y $O(n^3)$ llamadas a un oráculo que calcule f. Pues

- De la linea 1 a 2 toma $O(n)$.
- La linea 4 toma $O(n^2)$.

- De la 5 a la 6 toma $O(n)$.
- Devolver el mejor candidato toma $O(n) + O(nT(|V|))$, donde hacemos el supuesto de que $f(X)$ se puede calcular en $T(|V|)$.

Así, la complejidad total es $O(n^3)$ operaciones y $O(n^3)$ llamadas al oráculo f, es decir $O(n^3(1 + T(n)))$. La proxima clase se verá que, para corte mínimo, efectivamente esto es $O(n^3)$, y se puede mejorar un poco a $O(n \log n + m)$.