Clase 09 MA6150. 20 de septiembre 2021.

Escribe

Felipe Valdevenito

Tabla.

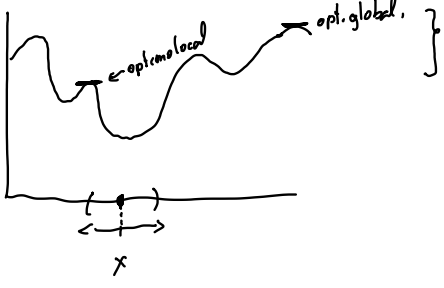
- Búsqueda local
- Matchings rápidos.

Búsqueda local

Paradigma simple en el que cada solución factible tiene asociada una vecindad polinomial de soluciones (usualmente obtenibles mediante pequeños cambios).

Óptimo local = Solución que es la mejor de su vecindad.

Algoritmo para encontrar óptimo local: Dada solución actual, moverse a la mejor solución de la vecindad. Hacer esto hasta que no haya mejora.



Ejemplo:

Max-Cut (sin pesos)

Dado un grafo $G = (V, E)$ encontrar un subconjunto S que maximize $|\delta(S)|$. (equivalente, encontrar partición $(S, \bar{S})$ que maximice $|E(S : \bar{S})|$).

Algoritmo

Partir con partición $(S, \bar{S})$ arbitraria.
Si existe $u \in V$ tal que cambiar u de lado mejora la partición, moverlo.

¿Aproximación:?

$2|\delta(S)| = \sum_{v \in V} |\delta(S) \cap \delta(v)| \geq \frac{1}{2} \sum_{v \in V} |\delta(v)| = |E| \geq opt.$

Nota: $|E(S : \bar{S})| \in \{0, 1, \ldots, m\}$ ~~luego se puede llegar a un óptimo local en tiempo...~~

$|\delta(S) \cap \delta(v)| \geq \frac{1}{2}|\delta(v)|$

Como en cada iteración alg aumenta en al menos 1

$\Rightarrow$ # iteraciones es $\leq m$.

2apx . búsqueda local.

Ejemplo 2:

Max-Cut (con pesos)

Dado un grafo $G = (V, E)$ encontrar un subconjunto S que maximize $w(\delta(S))$. (equivalente, encontrar partición $(S, \bar{S})$ que maximice $w(E(S : \bar{S}))$).

Algoritmo

Partir con partición $(S, \bar{S})$ arbitraria.
Repetir: Si existe $u \in V$ tal que cambiar u de lado mejora la partición, moverlo, si no, parar.

¿Aproximación:?
$2w(\delta(S)) = \sum_{v \in V} w(\delta(S) \cap \delta(v)) \geq \sum_{v \in V} w(\delta(v))/2 = w(E) \geq \text{opt}.$

Pero: ¡Podría tomar tiempo exponencial llegar a un óptimo local!

Óptimo local aproximado

Basta moverse si la mejora es sustancial.
Algoritmo para encontrar óptimo local aproximado: Dada solución actual X, moverse a la mejor solución Y de la vecindad. Si $w(Y) < (1 + f(\varepsilon, n))w(X)$ parar.

Idea: Elegir f de modo que el número de iteraciones sea poco y que sea posible adaptar demostración de garantía a un óptimo aproximado.

Algoritmo para max-cut

$v \leftarrow \arg\max\{w(\delta(v))\}$
Partir con partición $(S, \bar{S}) = (\{v\}, V \setminus \{v\})$.
Repetir: Si existe $u \in V$ tal que cambiar u de lado mejora el valor en un factor al menor $(1 + \varepsilon/n)$, moverlo. Si no, parar.

¿Iteraciones? $w(\delta(v)) \geq \frac{2w(E)}{n}$. $\quad opt \in \left[\frac{2w(E)}{n}, w(E)\right]$

pues $\sum_X w(\delta(X)) = 2w(E)$

$w(E) \geq alg_{\text{iteración } j} \geq \frac{2w(E)}{n} \cdot (1 + \varepsilon/n)^j \rightsquigarrow$ $\quad (1 + \varepsilon/n)^j \leq \frac{n}{2}$

$j \leq \log_{1+\varepsilon/n} n/2 = \frac{\log n/2}{\log(1+\varepsilon/n)} \sim \varepsilon/n$

$= O\left(\frac{n}{\varepsilon} \log n\right)$

¿Garantía de aproximación?

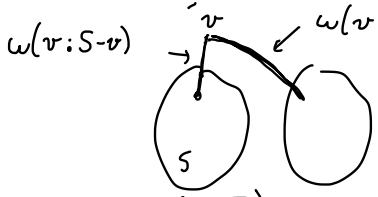
Algoritmo para max-cut

$v \leftarrow \arg\max\{w(\delta(v))\}$
Partir con partición $(S, \bar{S}) = (\{v\}, V \setminus \{v\})$.
Repetir: Si existe $u \in V$ tal que cambiar u de lado mejora el valor en un factor al menor $(1 + \varepsilon/n)$, moverlo. Si no, parar.

$w(\delta(S)) \geq opt\left(\frac{1-\varepsilon/2}{2}\right) \cdot \frac{2}{1-\varepsilon/2}$ aprox.

Nota: Si S es óptimo local aproximado. salida del algoritmo

$$2w(\delta(S) \cap \delta(v)) > w(\delta(v)) - w(\delta(S))\varepsilon/n \quad (*)$$



Δsolución $= w(v : S-v) - w(v : \bar{S}) < \varepsilon/n \; w(\delta(S))$

$\Rightarrow$ Nueva $< \left(1 + \frac{\varepsilon}{n}\right)$ Antigua

$2w(\delta(S) \cap \delta(v)) = 2w(v : \bar{S})$

$w(\delta(v)) = w(v : \bar{S}) + w(v : S - v)$

$\rightarrow \varepsilon/n \; w(\delta(S)) < w(v : \bar{S}) - w(v : S - v)$ $\forall v$ +

$w(\delta(v)) - w(\delta(S))\frac{\varepsilon}{n} < 2w(\delta(S) \cap \delta(v))$

$$\begin{aligned} 2w(\delta(S)) &= \sum_{v \in V} w(\delta(S) \cap \delta(v)) \\ &> \frac{1}{2}\left[\sum_{v \in V}\left[w(\delta(v)) - w(\delta(S)) \cdot \frac{\varepsilon}{n}\right]\right] \\ &= w(E) - \frac{\varepsilon}{2} w(\delta(S)) \\ &\geq w(E)\left[1 - \frac{\varepsilon}{2}\right] = w(E)\left(1 - \frac{\varepsilon}{2}\right) \end{aligned}$$

Matchings

Matchings

Sea M un matching en un grafo $G = (V, E)$.

- Un vértice v es M-cubierto si v toca una arista de M (si no, es M-expuesto)
- Un ciclo/camino C es M-alternante si sus aristas alternan entre M y $E \setminus M$
- Un camino M-alternante con ambos extremos M-expuestos se llama M-aumentante.
- Si $P_1, \ldots, P_k$ son caminos M-aumentante vértice disjuntos entonces $M' = M \Delta P_1 \Delta \cdots \Delta P_k$ es matching con $|M'| = |M| + k$.

Lema

M, M' matchings con $|M'| = |M| + k$, entonces existen k caminos M-aumentantes vértice-disjuntos.

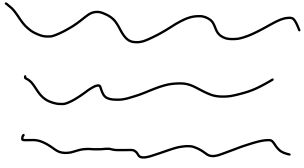
Dem: $M \Delta M'$ se puede descomponer en caminos y ciclos M-alternantes M-alternantes

$\Rightarrow$ $\exists$ al menos k caminos P

$|P \cap M'| = |P \cap M| + 1$

$\hookrightarrow$ M-aumentante

$|M \Delta P| = |M| + 1.$



Matching de cardinalidad máxima

$\ell = 1$

Corolarios para M matching, $\ell \geq 1$

No hay caminos M-aumentantes $\iff$ $|M| = \text{opt}$.

No hay caminos M-aumentantes de largo $\leq 2\ell - 1$ $\implies$ $|M| \geq \text{opt}(1 - \frac{1}{\ell+1})$.

$\ell = 2$

$\ell = 1 \Rightarrow$ Matching Maximal $\Rightarrow |M| \geq \text{opt}\left(1 - \frac{1}{2}\right)$ 2apx

$\ell = 2 \Rightarrow$ No hay  agregar $\Rightarrow |M| \geq \text{opt}\left(1 - \frac{1}{3}\right) \Rightarrow$ 1.5 apx.

Dem Sea k tq

$\text{opt} - |M| = k$.

$\Rightarrow$ Hay k caminos M-aumentantes vértice disjuntos

Lema

OPT cubre a todo $|V(P_1) \cup V(P_2) \cup \dots \cup V(P_k)|$

$2\text{opt} \geq \sum_{i=1}^{k} |V(P_i)| \geq k(2\ell+2) \Rightarrow k \leq \frac{2\text{opt}}{2\ell+2} = \frac{\text{opt}}{\ell+1}$

largo $\geq 2\ell + 1 \Rightarrow$ toca $2\ell+2$ vertice o más

$\Rightarrow \text{opt} - |M| \leq \frac{\text{opt}}{\ell+1}$

$\Rightarrow \text{opt}\left(1 - \frac{1}{\ell+1}\right) \leq |M|$

Búsqueda local

$\text{ALG} \leftarrow \emptyset$

Mientras exista camino P M-aumentante de largo $\leq 2\ell - 1$

$\text{ALG} \leftarrow \text{ALG} \Delta P$.

¿Por qué es búsqueda local si hay algoritmos polinomiales?

1. Cambios de una iteración a la siguiente son pequeños (toma $O(\ell)$ intercambiar un camino aumentante de largo $\leq 2\ell + 1$). Como son $\leq m$ iteraciones, existe esperanza de obtener un algoritmo de aproximación a tiempo lineal $O(m\ell)$. (Lo anterior es cierto solo si somos capaces de encontrar un camino P-aumentante rápido)
2. Idea para los mejores algoritmos de aproximación existentes para algunos sistemas de independencia.
3. Buenos algoritmos para matching hacen uso de esta idea.

Algoritmos exactos para matching de cardinalidad máxima en grafos bipartitos $n = O(m)$.

- (1931) König / (1956) Ford-Fulkerson $O(nm)$. Mediante flujo máximo.
- (1973) Hopcroft-Karp $O(m\sqrt{n})$.
- (1981) Ibarra-Morán $O(n^\omega)$ (con alta probabilidad). Usando multiplicación rápida de matrices.
- (2013) Madry $\tilde{O}(m^{10/7})$ (con alta probabilidad). Usando flujos eléctricos.

Algoritmos exactos para matching de cardinalidad máxima en grafos arbitrarios $n = O(m)$.

- (1965) Edmonds $O(n^2 m)$.
- (1980) Micali-Vazirani, (1990) de Blum, (1991) Gabow-Tarjan $O(m\sqrt{n})$.
- (2004) Mucha-Sankowski $O(n^\omega)$ (con alta probabilidad). Usando multiplicación rápida de matrices.

Caminos aumentantes crecen

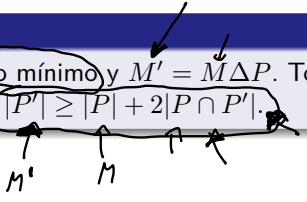
$2\ell-1$

Glotón (caso $\ell = 1$) es especial: una vez que una arista es analizada no es necesaria volverla a mirar. Luego glotón es una 2-aproximación a tiempo $O(m)$.

Idea para mayores ℓ: Largo mínimo de un camino ALG-aumentante solo puede aumentar.

Lema

Sea M matching, P camino M-aumentante de largo mínimo y $M' = M \Delta P$. Todo camino M'-aumentante P' tiene largo $\geq |P|$. De hecho $|P'| \geq |P| + 2|P \cap P'|$.



Dem:

M

$M' = M \Delta P$

$M'' = M' \Delta P' = M \Delta P \Delta P'$

$|M''| = |M| + 2$

$\hookrightarrow \exists Q_1, Q_2$ vértice disjuntos M-aumentantes.

Lema $\ell(Q_1), \ell(Q_2) \geq \ell(P)$

$|M \Delta M''| \geq |Q_1| + |Q_2| \geq 2|P|$

Q_1, Q_2

$M \Delta M'' = (M \Delta M) \Delta P \Delta P'$

$\Rightarrow 2|P| \leq |P \Delta P'| = |P| + |P'| - 2|P \cap P'|$

Consecuencia

Teorema

Sea M matching; $P_1, \ldots, P_k$ colección de caminos M-aumentantes vértice-disjuntos de largo $|P_1| = \cdots = |P_k| = q$ mínimo y con k maximal.
Si $M' = M \Delta P_1 \Delta \cdots \Delta P_k$, entonces el grafo no tiene caminos M'-aumentantes de largo $\leq q$

(llamemos $P_1, \ldots, P_k$ un conjunto bloqueador maximal.)