Parcial2AA-Problema8

Oriol Miró López-Feliu

May 2023

Problema 8

Consider the Max Clique problem. Given an undirected graph $G = (V, E)$, compute a set of vértices that induce a complete subgraph with maximum size.

For each $k \geq 1$, define G^k to be the undirected graph (V^k, E^k), where V^k is the set of all ordered k-tuples of vértices from V. E^k is defined so that $(v_1, \ldots, v_k)$ is adjacent to $(u_1, \ldots, u_k)$ if and only if, for $1 \leq i \leq k$, either $(v_i, u_i) \in E$ or $v_i = u_i$.

(a) Prove that the size of a maximum size clique in G^k is the k-th power of the corresponding size in G.

(b) Argue that if there is a constant approximation algorithm for Max Clique, then there is a polynomial time approximation schema for the problem.

Part (a)

Para comprender el problema, empecemos con un ejemplo sencillo:

Consideremos un grafo $G = (V, E)$ donde el Max Clique tenga tamaño 2 (una arista $e = (a, b)$ que conecta dos vértices $a, b \in V$). En G^2, todos los vértices que consisten en combinaciones de a y b estarán conectados, ya que para cualesquiera dos vértices $x = (x_1, x_2), y = (y_1, y_2) \in G^2$, o $x_j = y_j$ o $x_j \neq y_j \implies$ una es a i la otra es b, i como $\exists (x, y) \in E$, esto implica que $\exists (x, y) \in E^2 \ \forall x, y \in V^2$ tales que $x = (x_1, x_2), y = (y_1, y_2)$ y $\forall j \in \{1, 2\}\ x_j, y_j \in \{a, b\}$.

Extendamos ahora este razonamiento para un clique de tamaño n en G. Supongamos que los vértices que forman el clique en G son $C = \{a_1, a_2, ..., a_n\}$, y por tanto $\forall i, j\ i \neq j, \exists (a_i, a_j) \in E$. En G^2, estos estos n vértices generarán n^2 vértices de la forma:

$$C^2 = \{a_1a_1, a_1a_2, ..., a_1a_n, a_2a_1, a_2a_2, ..., a_2a_n, ..., a_na_1, a_na_2, ..., a_na_n\}$$

Como C es una clique, $\forall x, y \in C^2, x = (x_1, x_2), y = (y_1, y_2), \forall j \in \{1, 2\}$, sabemos que $x_j, y_j \in C$, y o bien $x_j = y_j$ o bien $x_j \neq y_j \implies \exists (x_j, y_j) \in E$, igual que antes. Concluimos pues que $\forall x, y \in C^2 \exists (x, y) \in E^2$, y por tanto C^2 es una clique de tamaño n^2.

Finalmente, extendemos este razonamiento para cualquier G^k. Igual que antes, consideremos todos los vértices que forman la Max Clique de tamaño n en G como $C = \{a_1, a_2, ..., a_n\}$. En G^k, para cualquier $k \in \mathbb{N}$, estos vértices generarán n^k vértices de la forma:

$$C^k = \{(c_1, c_2, ..., c_k) \text{ tal que } \forall j \in [1, k], j \in \mathbb{N}, c_j \in C$$

Note: Quizá poner $j \in \mathbb{N}$ no es necesario y demasiado estricto, como queráis.

Una vez más, podemos razonar que $\forall x, y \in C^k$ tales que $x = (x_1, x_2, ..., x_k), y = (y_1, y_2, ..., y_k)$ y $j \in [1, k], x_j, y_j \in C$, entonces o bien $x_j = y_j$ o bien $x_j \neq y_j \implies \exists (x_j, y_j) \in E$, ya que C es una clique, concluyendo que C^k tiene una clique de tamaño n^k.

Por tanto, hemos demostrado que si tenemos una Max Clique de tamaño n en G, tendremos una Max Clique de tamaño **al menos** n^k en G^k. Ahora demostraremos que esta Max Clique no puede ser mayor que n^k.

Si tuviéramos una clique de tamaño $> n^k$, implicaría que hay vértices en G^k conectados a todo vértice de C^k que no pertenecen a C^k. Por simplicidad, asumamos que existe 1 vértice que cumple estas condiciones, y por tanto la Max Clique en G^k tiene tamaño $n^k + 1$. Nombremos este vértice w. Para que $w = (w_1, w_2, ..., w_k)$ sea adyacente a todos los vértices de C^k, se debe cumplir que $\forall j \in [1, k]$ o bien $\exists i \in [1, n]$ tal que $w_k = a_i$ (lo que querría decir que $w \in C$), o bien $\forall i \in [1, n] \exists (w_j, a_i) \in E$. Como hemos restringido $w \notin C^k$, eso implica que $\exists w_j \notin C$ (de otra manera w estaría en C^k). Esto implica que existe un vértice $w_j \in V, w_j \notin C$ que es adyacente a todos los vértices en C. Pero esto es una contradiccion, porque si ese vértice w_j existe, sería adyacente a todos los n vértices del Max Clique en G, con lo que el Max Clique real sería $C \cup w_j$, con tamaño $n + 1$. Pero como habiamos fijado inicialmente que el Max Clique en G tenía tamaño n, esto es una contradiccion, con lo que concluímos que el Max Clique en G^k no puede tener un tamaño mayor que n^k, y por tanto, queda demostrado que el Max Clique en G^k tiene tamaño **exactamente** n^k.

Part (b)

Si existe un algoritmo de aproximación constante para Max Clique, eso implica que existe un algoritmo A que nos dá una k-aproximación, es decir, sea OPT el la respuesta óptima al Max Clique en G, tenemos:

$$\frac{OPT}{A(G)} \leq k$$

Un *polynomial time aproximation schema (PTAS)* es tal que dado un $\epsilon > 0$, podemos computar una solución al problema en un factor de $1 + \epsilon$ de ser la óptima. En este caso, la existencia de un algoritmo de aproximación constante para Max Clique implica que existe un $PTAS$ para el problema, como demostraremos ahora.

Empezaremos con un ejemplo simple. Supongamos que, dada una instancia $G = (V, E)$ de Max Clique, construimos G^2 y aplicamos el algoritmo de aproximación constante A. Sabemos que su resultado, $A(G^2)$, estará fitado por $OPT(G^2) \leq k \cdot A(G^2)$. Aplicando lo que hemos visto en la pregunta a), sabemos que la Max Clique de G^2 será exactamente el cuadrado de la Max Clique en G. Por tanto, $OPT(G^2) = OPT(G)^2$, y $A(G^2) = A(G)^2$, lo que nos lleva a $OPT(G)^2 \leq k \cdot A(G)^2 \implies OPT(G) \leq \sqrt{k} \cdot A(G)$

Construyendo G^2 hemos mejorado nuestro algoritmo de aproximación constante A por un factor de $\sqrt{k}$. Podemos aplicar esto tantas veces como queramos, construyendo G^l para cualquier l, lo que dará mejores aproximaciones, resultando en $OPT(G) \leq \sqrt[l]{k} \cdot A(G)$. De hecho,

$$\lim_{l \to \infty} \sqrt[l]{k} = 1, \forall k > 1$$

lo que nos asegura poder obtener una aproximación tan ajustada como queramos.

Dada pues una $\epsilon > 0$, escogemos l tal que $\sqrt[l]{k} < 1 + \epsilon$ y construimos G^l, corremos nuestro algoritmo de aproximación constante A en G^l, y obtenemos una aproximación $A(G)$ dentro de un factor de $\sqrt[l]{k}$ del problema.

La construcción de G^l se puede hacer en tiempo $O(l \cdot |V|^{2l})$, dado que la construcción de V^l recibe $O(|V|^l)$ y para construir E^l necesitamos realizar l comprobaciónes por cada pareja de vértices en V^l, resultando en $O(l \cdot |V|^{2l})$, lo cual es polinómico en el tamaño de la entrada (el grafo G dado). Suponemos que estamos usando una estructura de datos óptima donde comprobar la existencia de una arista en E se hace en tiempo $O(1)$, tal y como una matriz de adyacencia.