EL VIAJE

MARCOS M. TIRADOR AND LEANDRO RODRÍGUEZ LLOSA

1. Preliminares

Se define un grafo no dirigido G como un par (V, E) de conjuntos tales que el segundo es una relación ($E \subseteq V \times V$) anti-reflexiva y asimétrica definida sobre el primero. A los elementos de V se les suele llamar vértices o nodos y a los elementos de E aristas. Además usaremos la notación $V(G)$ y $E(G)$ para referirnos respectivamente al conjunto de vértices y al conjunto de aristas de un grafo G dado; aunque en caso de que no haya ambigüedad usaremos simplemente V y E. Sea (x, y) un elemento de E, usaremos también la notación $\langle x, y \rangle$ para referirnos al mismo.

Dado un grafo $G = (V, E)$ se dice que $G' = (V', E')$ es un subgrafo de G si $V' \subset V$, $E' \subset E$ y todas las aristas en E' inciden solo sobre elementos de V'. En este caso escribimos $G' \leq G$. Dado un subconjunto $X \subset V$ definimos el grafo inducido por X en G y lo denotamos por $G[X]$ al grafo $G' = (X, E')$, donde E' consiste de todas las aristas en E que no inciden sobre ningún vértice en $V \setminus X$.

Definimos el grado de un vértice $v \in V(G)$ como la cantidad de aristas que inciden sobre este y escribimos $\deg_G(v)$ (o $\deg(v)$ cuando no haya ambigüedad). Decimos que un grafo es regular si todos sus nodos tienen el mismo grado. Un grafo será entonces k-regular si todos sus nodos tienen grado k. A los grafos k-regulares con $k = 3$ también suele llamárseles grafos cúbicos.

Definimos un problema abstracto o simplemente problema como una relación binaria Q sobre un conjunto de instancias del problema I y un conjunto S de soluciones del problema. Se dice que un problema es decisión si $S = \{0, 1\}$. Decimos que un problema es polinomial o pertenece a la clase de complejidad P si existe un algoritmo que con complejidad temporal polinomial (o dicho más simple un algoritmo polinomial) puede encontrar para un elemento de I un elemento de S que esté relacionado con este.

Dado un problema de decisión se define informalmente un certificado como un camino que puede ser seguido para verificar si la respuesta del problema es que "si" o que "no". Decimos que un problema es NP si existe un algoritmo polinomial A tal que para cada instancia del problema x con solución 1, existe un certificado y que el algoritmo recibe como entrada junto con x y tiene salida 1, o sea $A(x, y) = 1$.

Se dice que un problema Q puede ser reducido a otro problema Q' si cualquier instancia de Q puede ser transformada en una instancia de Q' tal que la solución de Q' provea una solución para Q. Se dice que una reducción es polinomial si existe un algoritmo polinomial que tranforma cada instancia de Q en una intancia de Q'. Se dice entonces que un problema Q pertenece a la clase de complejidad NP-duro si existe una reducción polinomial de todo problema NP al problema Q (en términos informales se dice que cualquier problema NP es cuándo más tan difícil como Q). Por último se dice que un problema es NP-completo si este es NP y NP-duro. Es fácil ver que demostrar que un problema NP-completo Q' puede ser reducido a un problema Q es suficiente para afirmar que Q es NP-duro, dado que cualquier problema NP puede ser reducido a Q' en tiempo polinomial y luego este a Q.

Date: June 16, 2023.

2. Introducción al Problema

Problema 2.1. La venganza de Alejandra

Por algún motivo, a Alejandra no le gustaba la paz y le irritaba que sus compañeros de aula se llevaran tan bien. Ella quería ver el mundo arder. Un día un demonio se le acercó y le propuso un trato: "A cambio de un cachito de tu alma, te voy a dar el poder para romper relaciones de amistad de tus compañeros, con la única condición de que no puedes romperlas todas". Sin pensarlo mucho (Qué más da un pequeño trocito de alma), Alejandra aceptó y se puso a trabajar. Ella conocía, dados sus k compañeros de aula, quiénes eran mutuamente amigos.

Como no podía eliminar todas las relaciones de amistad, pensó en qué era lo siguiente que podía hacer más daño. Si una persona quedaba con uno o dos amigos, podría hacer proyectos en parejas o tríos (casi todos los de la carrera son así), pero si tenía exactamente tres amigos, cuando llegara un proyecto de tres personas, uno de sus amigos debería quedar afuera y se formaría el caos.

Ayude a Alejandra a saber si puede sembrar la discordia en su aula eliminando relaciones de amistad entre sus compañeros de forma que todos queden, o bien sin amigos, o con exactamente tres amigos.

Podemos representar a los compañeros de Alejandra mediante un grafo no dirigido, donde las aristas indican las relaciones de amistad existentes. Como Alejandra quiere que cada persona termine siendo amigo de cero o tres personas, sin eliminar todas las relaciones, básicamente Alejandra quiere obtener un subgrafo del grafo de amistad tal que todos sus nodos tengan grado 0 o 3, y haya al menos uno de grado 3. Luego de retirar los nodos de grado 0, podemos pensar en el problema de la siguiente forma.

Problema 2.2. Dado un grafo no dirigido G determinar si existe un subgrafo $G' \leq G$ tal que G' es cúbico.

3. NP completitud

En esta sección demostraremos que el Problema 2.2 es NP-completo, al demostrar que es NP y que se puede reducir en tiempo polinomial el conocido problema de 3-coloración al nuestro. Definamos primeramente el problema de 3-coloración.

Problema 3.1. Problema de 3-coloración

Dado un grafo no dirigido $G = (V, E)$ se quiere determinar si existe una partición de V de la forma $\{V^1, V^2, V^3\}$, tal que no existe $\langle x, y\rangle \in E$ tal que $x, y \in V^i$ para algún $i \in \{1, 2, 3\}$.

Teorema 3.2. *El problema de* 3*-coloración es* NP*-completo.*

Demostración: Ver [3]. □

Ya estamos preparados para demostrar el resultado principal de esta sección. Este esta basado en la demostración hecha en [2], en el apéndice titulado *NP-completness.*

Teorema 3.3. *El problema de reconocer si un grafo tiene algún subgrafo cúbico es NP-completo.*

Demostración: Primero es fácil ver que el problema es NP dado que podemos crear fácilmente un algoritmo que dado un subgrafo G' del grafo original, compruebe en tiempo polinomial si este es cúbico, solamente comprobando los grados de todos los vértices. Entonces para cada grafo G que es una instancia del problema con solución afirmativa podemos tomar como certificado algún subgrafo $G' \leq G$ cúbico y el algoritmo descrito decidirá que la solución en este caso es afirmativa.

Ahora para demostrar que el problema de decisión en cuestión es NP-completo reduciremos el problema de 3-coloración a nuestro problema. Sea $G = (V, E)$ un grafo que se recibe como entrada al problema de 3-coloración. Del mismo queremos determinar si sus vértices se pueden dividir en 3 conjuntos disjuntos tal que no existen aristas entre los vértices de un mismo conjunto. Para proceder a demostrar que este problema se puede reducir al problema de determinar si un grafo posee subgrafos cúbicos, transformemos la entrada G del problema original en otra entrada G' para el segundo problema.

Denotemos por $v_1, v_2, \ldots, v_n$ los nodos en V y por $e_1, e_2, \ldots, e_m$ las aristas. Además introducimos $d_i := \deg(v_i)$ por cada nodo $v_i \in V$. Por último para cada nodo $v_i \in V$ definimos $\rho_i := \{j \mid e_j$ **incide en** $v_i\}$, fijaremos un orden $\rho_{i1}, \rho_{i2}, \ldots, \rho_{id_i}$ de los elementos de ρ_i. El conjunto V' estará formado de la siguiente forma:

- Por cada $v_i \in V$ tendremos el vértice $v_{i,j}^h$ en V', para cada $1 \leq j \leq 3d_i + 2$ y $1 \leq h \leq 3$.
- Por cada arista $e_j \in E$ tendremos el vértice a_j^h en V', para cada $1 \leq h \leq 3$.
- Por cada vértice $v_i \in V$ tendremos otros dos vértices $x_i, s_i \in V'$. A los vértices de la forma s_i los llamaremos especiales.

En el resto de la demostración será importante el nombre de cada nodo en V', pues dependiendo de este serán las aristas que incidan en el nodo y la función que cumpla este en el grafo. El conjunto de aristas E' estará definido como sigue:

- Por cada $1 \leq i \leq n$ tendremos una arista entre los vértices $v_{i,j}^h$ y $v_{i,j+1}^h$ (donde se toma el segundo subíndice módulo $3d_i + 2$), para cada $1 \leq h \leq 3$. Como podemos ver se formarán así $3n$ ciclos de tamaño $3d_i + 2$ a los que denotaremos por C_i^h.
- Para cada nodo v_i, por cada $1 \leq h \leq 3$, conectaremos los nodos $v_{i,3j}^h$, $v_{i,3j+1}^h$ y $v_{i,3j+2}^h$ con el nodo $a_{\rho_{ij}}^h$, para cada $1 \leq j \leq d_i$.
- Por cada vértice $v_i \in V$ conectamos el vértice $x_i \in V'$, con $v_{i,3d_i+1}^h$ y $v_{i,3d_i+2}^h$, para cada $1 \leq h \leq 3$.
- Cada vértice s_i lo conectamos con s_{i+1} (módulo n) formando así un ciclo que denotaremos por S.
- Por cada $1 \leq i \leq n$ conectamos x_i con s_i.

Una vez definida la transformación del grafo G que se da como entrada del problema original en el grafo G' solo resta demostrar que G es 3 coloreable si y solo si G' tiene un subgrafo regular de grado 3.

Primero supongamos que G es 3-coloreable y por tanto podemos tomar una partición de sus vértices $\{V^1, V^2, V^3\}$ en tres conjuntos correspondientes a la 3-coloración. Tomaremos G'' como el subgrafo inducido en G' por los siguientes vértices:

- El vértice s_i, para $1 \leq i \leq n$.
- El vértice x_i, para $1 \leq i \leq n$.
- El vértice a_i^h, para $1 \leq i \leq m$ y $1 \leq h \leq 3$.
- Por cada vértice $v_i \in V^j$, añadimos todos los vértices en C_i^j, para $j \in \{1, 2, 3\}$.

Demostremos que G'' contiene solo vértices de grado 0 o 3, por lo que luego de eliminar los vértices de grado 0 tendremos el grafo cúbico buscado. Es fácil ver que todos los vértices s_i tienen grado 3. Para cada vértice x_i notemos que este está conectado a s_i y además a exactamente a uno de los siguientes pares de vértices: $(v_{i,3d_i+1}^1, v_{i,3d_i+2}^1), (v_{i,3d_i+1}^2, v_{i,3d_i+2}^2), (v_{i,3d_i+1}^3, v_{i,3d_i+2}^3)$, ya que solo los vértices en uno de esos pares pertenecerá a G'', en dependencia de a qué conjunto de V^1, V^2, V^3 pertenezca v_i.

Sea $v_{t,y}^k$ un vértice arbitrario de G'' (con la forma descrita). Nótese que para cada vértice $v_i \in V^h$, todos los vértices $v_{i,j}^h$ pertenecen G'' y entonces el ciclo C_i^h está en G''. Por tanto, el vértice $v_{t,y}^k$ tendrá arista en G'' con sus dos adyacentes en el ciclo C_t^k. Además, como $v_{t,y}^k$ estaba conectado en G' con exactamente un vértice de la forma x_t o $a_{\rho_{tp}}^k$ para algún $1 \leq p \leq d_t$, y cada vértice con alguna de estas formas pertenece a G'', tenemos que v_{ty}^k tiene al menos grado 3. Sin embargo, $v_{t,y}^k$ tenía exactamente grado 3 en G', por lo que concluimos que este tiene grado 3 en G'' también.

Ahora tomemos un vértice a_i^h cualquiera (con la forma descrita), supongamos que $e_i = \langle v_t, v_y \rangle$ (donde el índice de e_i fue tomado de igual forma que en la definición de a_i^h). Nótese que a_i^h está en G' conectado a los 3 vértices $v_{t,3j}^h, v_{t,3j+1}^h$ y $v_{t,3j+2}^h$ para algún $1 \leq j \leq d_t$, y a los 3 vértices $v_{y,3k}^h, v_{y,3k+1}^h$ y $v_{y,3k+2}^h$ para algún $1 \leq k \leq d_y$. Sin embargo, como v_t y v_y son adyacentes, entonces estos están en conjuntos de $\{V^1, V^2, V^3\}$ diferentes. Por tanto, a lo sumo uno de los ciclos C_t^h y C_y^h estará en G'' de donde tenemos que a_i^h tendrá o bien grado 3, o 0. Por tanto podemos concluir que G' tiene un subgrafo cúbico.

Ahora supongamos que G' tiene un subgrafo cúbico G'' y demostremos que G es 3-coloreable. Para ello construiremos los conjuntos $V^k := \{v_i \mid v_{i,j}^k \in V(G'') \textbf{ para algún } j\}$, para $k = 1, 2, 3$, y demostraremos que estos conforman una partición de V. Además, tendríamos que demostrar que no existen aristas entre vértices en un mismo conjunto V^k.

Denotemos por $V'' := V(G'')$. Empecemos por notar que si $v_{i,j}^h \in V''$, entonces C_i^h está en G''. Esto se debe a que $v_{i,j}^h$ tiene solamente un adyacente fuera de C_i^h en G'. Supongamos que el vértice a_i^h está en G''. Sean v_t y v_y los vértices que e_i conecta en G. Si a_i^h tuviera adyacentes en G'' tanto en C_t^h como en C_y^h, entonces por lo expuesto anteriormente ambos ciclos estarían en G'' y el grado de a_i^h sería 6, lo cual es absurdo. Entonces si $a_i^h \in V''$ entonces exactamente uno de los ciclos C_t^h o C_y^h está en G'', ya que $\deg(a_i^h) = 3$. Podemos concluir de aquí que no existen en G aristas entre vértices de un mismo conjunto V^k. Analizando en el otro sentido, si el ciclo C_t^h está en G'', entonces para cada arista $e_{\rho_{tk}}$ incidente en v_t en G, tenemos que para que el nodo $v_{t,3i}^h$ tenga grado 3 en G'', el vértice $a_{\rho_{tk}}^h$ debe estar en V''.

Ahora veamos que si $C_i^h \in G''$ entonces $v_{i,3d_i+1}^h \in V''$ y para que su grado sea 3, debe pasar que $x_i \in V''$. Ahora si $x_i \in V''$, como este vértice es adyacente en G' solamente a s_i y a dos vértices en cada ciclo C_i^h, entonces tenemos que $s_i \in V''$ es la única forma de que $\deg_{G''}(x_i)$ sea impar. Además, en ese caso exactamente uno de los ciclos C_i^1, C_i^2 y C_i^3 estará en G''. Es fácil ver además que si $s_i \in V''$ para algún i, entonces $s_i, x_i \in V''$ para todo $1 \leq i \leq n$, dado que s_i tiene en G' exactamente un adyacente fuera de S, que es x_i, y dos en S.

Hemos demostrado que si un vértice v cualquiera pertenece a G'', entonces x_i y s_i pertenecen a V'' para todo $1 \leq i \leq n$. Además, si $x_i \in V''$, entonces $C_i^h \in G''$ para exactamente un valor de h. Por tanto, podemos concluir que $\{V^1, V^2, V^3\}$ es una partición de V.

$\square$

4. Aproximación

Para poder encontrar una solución aproximada de un problema de decisión lo que podemos hacer es transformarlo en un problema de optimización, cuya solución óptima nos permita encontrar la solución del problema original. Para el caso del Problema 2.2 un posible problema de optimización correspondiente sería el siguiente.

Problema 4.1. Dado un grafo no dirigido G, encuentre un subgrafo $G' \leq G$, con al menos un vértice de grado 3, que maximice el número de nodos con grado 0 o 3.

Sea C^* la solución óptima del problema para una instancia dada y sea C una solución hayada por un algoritmo de aproximación. Se dice que el algoritmo es una $\rho(n)$-aproximación si

$$\max\left(\frac{C}{C^*}, \frac{C^*}{C}\right) \leq \rho(n).$$

En el caso de un problema de maximizar es suficiente con $C^*/C \leq \rho(n)$.

Sea p_1 la función de costo para el Problema 4.1. Construyamos una solución aproximada del problema. Notemos que en este problema para que haya al menos un nodo con grado 3 en G', tiene que haber al menos un nodo con grado mayor o igual que 3 en G. Sea v este nodo, y x_1, x_2 y x_3 tres de sus adyacentes. Entonces si tomamos el subgrafo G' inducido por $\{v, x_1, x_2, x_3\}$ tenemos que $p_1(G') \geq n - 3$. Por tanto si $G* \leq G$ es tal que $p_1(G^*)$ es máximo, tenemos que

$$\frac{C^*}{C} = \frac{p_1(G^*)}{p_1(G')} = \frac{p_1(G^*)}{n-3} \leq \frac{n}{n-3} = 1 + \frac{3}{n-3}.$$

Por tanto si tomamos $\rho(n) = 1 + 3/(n+3)$, tenemos que esta solución es una $\rho(n)$-aproximación. Podemos observar que para $n \geq 6$ esta es una 2-aproximación. En general para todo $k > 1$ existe $N \in \mathbb{N}$ tal que para todo $n \geq N$ esta solución es una k-aproximación del óptimo.

Veamos otra posible transformación en problema de optimización.

Problema 4.2. Dado un grafo no dirigido G y un entero positivos $1 \leq k \leq |V|$, encuentre el subgrafo $G' \leq G$ de k vértices con el mayor número posible de nodos con grado 3.

Véase que resolviendo este problema, para cada valor de k, podemos encontrar la solución del problema de decisión 2.2. Dado un grafo G, si para algún valor de k entre 1 y n, el óptimo del Problema 4.2 con entrada G y k, es precisamente k, esto significa que G tiene un subgrafo cúbico de k nodos. En caso contrario, podemos afirmar que G no contiene subgrafo cúbico.

Veamos un algoritmo polinomial que constituye una 4-aproximación para el Problema 4.2.

Algoritmo 4.3. Tomemos un grafo $G' = (V', E')$ y dos conjuntos de nodos S y T inicialmente vacíos. En cada momento en el algoritmo garantizamos que se cumplan las siguientes invariantes.

1. S y T constituyen una partición de V'.
2. Todos los nodos de S tienen grado 3 en G'.
3. $|V'| \leq k$.

Decimos que un vértice de S no está saturado si tiene al menos una arista con uno de T en G'. En cada iteración del algoritmo tomamos el vértice v de menor grado de $V \setminus V'$ tal que su grado sea mayor o igual que 3. Pueden darse 3 casos:

Case 1: v tiene al menos 3 adyacentes que no están en S. En este caso añadimos v a S y a tres de tales adyacentes a T (si no estaban ya), y las 3 aristas correspondientes a E'.

Case 2: v tiene a lo sumo $t < 3$ adyacentes que no están en S, pero tiene al menos $3-t$ adyacentes en S que no están saturados. En este caso, al igual que en el anterior añadimos a v a S, los mencionados t adyacentes a T y las respectivas aristas. Además tomaremos $3 - t$ de los adyacentes a v en S que no están saturados, los conectamos con v y eliminamos de cada uno una de sus aristas que van hacia nodos de T, las cuales sabemos que existen porque los nodos no están saturados.

Case 3: En caso de que no ocurra ninguno de los casos anteriores tomamos el conjunto X de todos los nodos de $V \setminus V'$ que no cumplan ni uno ni 2, y añadimos todos sus elementos a T. Luego eliminamos de G todas las aristas que van de algún nodo de $V - V'$ hacia alguno de los nodos saturados de S que son adyacentes a algún nodo de X (esto se hace con el fin de obtener un nuevo orden de los nodos de $V - V'$).

Repetimos entonces esta iteración del algoritmo hasta que se deje de cumplir la invariante 3, en cuyo caso devolvemos el estado de G' antes de hacer la última iteración.

Es fácil ver que en cada iteración del algoritmo se mantienen las 3 invariantes definidas en el mismo.

Teorema 4.4. *La complejidad temporal del algoritmo es $O(kn^2)$.*

Demostración: En cada iteración del algoritmo aumenta en al menos 1 el tamaño de V', por lo tanto este hace k iteraciones a lo sumo. Los casos 1 y 2 en una iteración se ejecutan en tiempo $O(n)$, dado que hay que explorar a lo sumo todos los adyacentes de un nodo, que son cuando más $n-1$. En el caso 3 tenemos que explorar todos los nodos que quedan en $V - V'$ y por cada uno, pasar por todos sus adyacentes para ver si no se cumplen 1 o 2, por lo que en este caso la complejidad de una iteración es $O(n^2)$ (la complejidad de la ordenación es entonces despreciable). Por tanto en el caso peor nuestro algoritmo hace $O(kn^2)$ iteraciones. □

Corolario 4.5. *la solución aproximada 4.3 es polinomial.*

Teorema 4.6. *La solución 4.3 es una 5-aproximación.*

Demostración: Sean $X_1, X_2, \ldots, X_t$ los conjuntos de nodos añadidos a T después de cada iteración del algoritmo que pasó por el caso 3 y sea Y_i, con $1 \leq i \leq t$ el conjunto de nodosde S que eran adyacentes a algún nodo de X y que estaban saturados en el momento que se añadió X a T. Como luego de ser añadido X_i a T se eliminan todas las aristas de Y_i a $V \setminus V'$, entonces los conjuntos Y_i son todos disjuntos dos a dos. Además en el momento que se adiciona X_i a T, se cumple que el número de aristas que va de X_i a Y_i es menor que $d-2$ donde d es el menor grado de algún vértice de X_i. Sin embargo el número de aristas que van de Y_i a X_i es mayor que $d'-3$, donde d' es mayor grado de algún vértice de Y_i. Sin embargo como en todo momento tomamos los vértices en orden creciente de su grado, tenemos que $d \geq d'$. Se e el número de aristas entre Y_i y X_i, tenemos entonces que $|X_i|(d-2) \leq e \leq |Y_i|(d'-3) \leq |Y_i|(d'-2) \leq |Y_i|(d-2)$, de donde tenemos que $|X_i| \leq |Y_i|$. Por tanto $X^* := \sum_{i=0}^{t} |X_i| \leq \sum_{i=0}^{t} |Y_i| \leq |S|$. Finalmente en cada iteración que cae en el caso 1 o 2, se cumple que se agregan a T a lo sumo 3 nodos por cada nodo que se agrega a S. Por tanto tenemos finalmente que $|V'| = |T| + |S| = |S| + |X^*| + |T \setminus X^*| \leq |S| + |S| + 3|S| = 5|S|$, de donde se cumple lo pedido. □

References

[1] T. H. Cormen, C. E. Leiserson, R. L. Rivest y C. Stein: *Introduction to Algorithms.* Third Edition.

[2] V. Chvátal, H. Fleischner, J. Sheehan, C. Thomassen: *Three-Regular Subgraphs of Four-Regular Graphs.* Journal of Graph Theory, Vol. 3 (1979) pp. 371 - 386.

[3] R. M. Karp: *Reducibility among combinatorial problems.* R. E. Miller et al. (eds.), Complexity of Computer Computations (1972), pp 85-103..

Facultad de Matemática y Computación, Universidad de La Habana, Ciudad de La Habana, Cuba
Email address: marcosmath44@gmail.com

Facultad de Matemática y Computación, Universidad de La Habana, Ciudad de La Habana, Cuba
Email address: leandro_driguez@outlook.com