PRÁCTICA 4
MAC

Problemas NP - Completos

Juan Antonio Velasco Gómez
Miguel Sánchez Maldonado

1. **Camino más largo** [LONGEST PATH]: Dado un grafo $G = (V, E)$ y un entero positivo $K \leq |V|$, ¿contiene G un camino simple (que no pase dos veces por el mismo sitio) con K o más arcos?

Solución

Vamos a buscar reducir desde el problema del circuito Hamiltoniano, para ello dado un grafo G=(V,E) con nodos, construimos el siguiente procedimiento

- Si G no tiene un camino largo con n-1 arcos, devolvemos falso. En caso contrario, ordenamos los vértices según el camino encontrado, esto es, tenemos v1,...,vn
- Añadimos un vértice adicional v0 que esté conectado a v1 y vn exclusivamente.
- Si G tiene un camino largo con n-arcos ⇒ Devuelve verdadero y si no, falso.

Por tanto,

G tiene un circuito Hamiltoniano ⇔ El procedimiento nos da verdadero.

2. **Coloreado de grafos con 3 colores** [GRAPH 3-COLORING]: Dado un grafo $G = (V, E)$, ¿existe una función $f : V \to \{1, 2, 3\}$, tal que $f(u) \neq f(v)$ para todos los arcos $\{u, v\} \in E$?

Solución

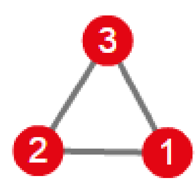
Ya vimos que con dos colores siempre se puede resolver (puesto que es un grafo bipartido). Veamos si es posible resolverlo con 3 colores, para ello buscamos que sea un problema NPC (completo). Tomamos un problema que sabemos ya cumple esta condición, en este caso será:

3SAT → Problema NP-Completo
$(x_1 \vee x_2 \vee x_3) \wedge (x_2 \vee \overline{x_3} \vee x_4)$

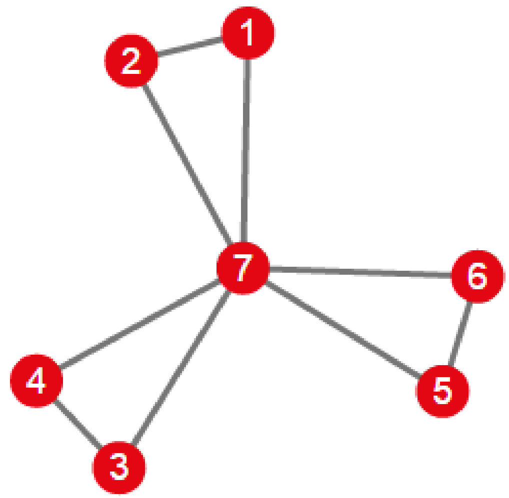
Lo reducimos a nuestro problema G3C, es decir, 3SAT $\leq$ G3C mediante

3SAT $\leq$ NAE 3SAT $\leq$ G3C

Pensamos en un nodo x_1 que sea verdadero, entonces su negación sería falso. Como buscamos 3 colores, buscamos hacer triángulos. Para ello utilizaremos un nodo artificial o auxiliar y los unimos. De forma que unimos por una parte x_1 y su negación (cada uno de un color) y el nodo artificial del tercer color.



Todas las parejas de x_i y su negación se unen a ese nodo auxiliar. De forma que para cada cláusula que tengamos (que será un triángulo), podamos unir cada x_i de la cláusula, con su correspondiente negación en el grafo que consideramos como inicial.



El problema viene al reducir las cláusulas de 4 variables a cláusulas de 3 variables. Para ello se hace la siguiente reducción:

$(x_1 \vee x_2 \vee x_3) \Rightarrow (x_1 \vee x_2 \vee x_3 \vee z) \Rightarrow (x_1\ x_2\ x_3\ z) \Rightarrow (x_1\ x_2\ y_i)(x_3\ z\ \overline{y_i})$

Llegando a la conclusión de que:

$3SAT \leq$ NAE-4SAT $\leq$ NAE-3SAT $\leq$ G3C

Y por tanto G3C es NPC.

3. **Recubrimiento de un grafo** [VERTEX COVER]: Dado un grafo G y un entero k, ¿tiene G una cobertura de tamaño k o menor? Un conjunto de vértices $S \subseteq V$ es una cobertura del grafo G si, para toda arista $e \in E$, al menos uno de los extremos de e está en S.

   Dado que encontrar un recubrimiento mínimo de un grafo es equivalente a encontrar un conjunto independiente [INDEPENDENT SET] o un clique de tamaño máximo [CLIQUE], descubrir que cualquiera de estos tres problemas es NP-completo sirve para demostrar que los tres lo son.

Solución

Para demostrar que el problema de recubrimiento de un grafo es NP completo buscamos reducir un problema NPC al problema de "Independent Set" puesto que ya sabemos por la práctica anterior que:

INDEPENDENT SET ⇔ VERTEX COVER

Por tanto, lo que buscamos es reducir el problema que acabamos de ver que es NPC para grafos, es decir, el del coloreado de grafos, a uno de los 3 problemas que sabemos son equivalentes, aunque como ya hemos dicho, buscamos reducirlo mejor al problema de "Independent Set"

Podemos realizar la siguiente reducción:

$$3SAT \leq CLIQUE = VC$$

La idea es unir cada vértice del grafo con todas las demás cláusulas salvo en los que él mismo salga negado. Se van añadiendo aristas nuevas mientras el grafo siga siendo compatible, puesto que de ser así, todas se consideran verdaderas. (No puede ocurrir que sea cierto x_1 y x_1 negado).

Para reducir 3SAT al problema del CLIQUE tiene que tener cada cláusula uno de los nodos verdaderos. Dentro de la misma cláusula no se unen y por tanto el clique sería formado por un nodo de cada cláusula.

4. **Subgrafo común maximal** [SUBGRAPH ISOMORPHISM]: Dados los grafos $G_1 = (V_1, E_1), G_2 = (V_2, E_2)$, y un entero positivo K, ¿existen subconjuntos $E_1' \subseteq E_1$ y $E_2' \subseteq E_2$ tales que $|E_1'| = |E_2'| \geq K$ y tal que los dos subgrafos $G_1' = (V_1, E_1')$ y $G_2' = (V_2, E_2')$ son isomorfos?

Solución

Se busca reducir el problema del CLIQUE al problema de SUBGRAPH ISOMORPHISM. Para ello se utiliza lo siguiente:

$$3SAT \leq CLIQUE \leq SUBGRAPH\ ISOMORPHISM$$

La idea es determinar si tengo un clique de tamaño K.

```
CLIKE (G,K) {
        return IS(G,K_K)
}
```

donde K_k es un subgrafo completo de k-nodos.

Podemos ver esto construyendo una máquina de Turing, M, que toma como entrada dos grafos $G_1 = (V_1, E_1)$, $G_2 = (V_2, E_2)$, y una función no determinística h que va desde los nodos de G_1 hasta los nodos de G_2. La comprobación de que G_1 es un subgrafo de G_2, podemos hacerlo en tiempo polinomial: para cada (i, j) ∈ E1, vemos que (h(i), h(j)) ∈ E2.

Podemos reducir el problema del Clique al subgrafo común minimal:
dado un grafo G y un número entero k, decidir si el grafo G tiene un clique con k nodos (cualesquiera dos nodos son adyacentes).

Dicha reducción la hacemos de la siguiente manera: Para un grafo dado G y un número k, generamos un grafo completo, G_0, con k nodos.
G tiene un clique ⇔ G_0 es un subgrafo de G.
La reducción no necesita un espacio adicional, sino que puede escribirse la descripción de G0 directamente sobre la cinta de salida y así, la limitación de espacio es conocida.

- Probamos la implicación a la derecha: si G tiene un k-clique, entonces G tiene un subgrafo completo con k nodos, que es isomorfo al grafo generado G_0.
- Probamos la implicación a la izquierda: Si G_0 es isomorfo al subgrafo de G, entonces, esto significa que G tiene un subgrafo completo de tamaño k, que es un k-clique.

Por tanto, podemos concluir que el problema dado también es NP-completo.

5. **Corte máximo** [MAX CUT]: Dado un grafo ponderado G, dos vértices s y t y un entero positivo k, ¿existe un corte en G que separe s de t y tenga al menos peso k?

Solución

Buscamos reducirlo al problema NAE-3-SAT

Dado un problema de 3-sat con instancia ϕ, lo que vamos a construir será un máximo equivalente con instancia (G,c). Para cada variable xi que pertenezca a ϕ, lo que haremos será añadir dos vértices a G que serán etiquetados con xi y ¬xi y los conectaremos por una arista. Asignamos la capacidad M=10 · m para cada una de esas aristas. (m es el número de cláusulas en ϕ y n será el número de variables.) Para cláusula C en ϕ añadimos un triángulo entre los vértices correspondientes a los términos de C. Además cada uno de estos será arista.

Se puede afirmar que G contiene un corte con la capacidad al menos de n · M + 2 · m si y sólo si φ es satisfiable.

Suponemos que ϕ es satisfacible y consideramos cualquier asignación que sea correcta. Esto corresponde a un corte con G (Uno de los lados del corte se compone de todos los vértices etiquetados por términos que evalúa a 1 en esa tarea. El otro lado del corte consiste en todos los vértices etiquetados por términos que evalúan a 0 en la asignación) Uno de los términos xi y ¬xi evaluados en 1 en una tarea, toda variables de los bordes pasan a través del corte, que contribuyen a n · M a la capacidad del corte. Dado que la tarea satisface ϕ exactamente dos bordes en cada triángulo , lo que contribuye a 2 · m la capacidad del cote. En total la capacidad del corte es igual a n · M + 2 · m.

Por otro lado supongamos que G contiene un corte con capacidad del menos n · M + 2 · m. Primero, afirmamos que todos las las variables de las aristas van a través de este corte. La razón es que cualquier corte que no alcanza al menos uno de las aristas tiene la capacidad en la mayoría de los casos n − 1) · M + 3 · m = n · M + 3 · m − 10m , que es estrictamente más pequeño que n · M + 2m. A continuación, afirmamos que exactamente dos aristas de la cláusula de cada triángulo van a través del corte. La razón es que el no corte puede separar las 3 aristas del triángulo y por otro lado si un corte separa menos de dos bordes en un triángulo, esta capacidad será menor que n · M + 2m.

Además, puesto que el corte separa exactamente dos aristas por triángulo, esto corresponde a la asignación satisfactoria de todas las cláusulas de φ.

6. **Empaquetado de conjuntos** [SET PACKING]: Dada una colección C de conjuntos finitos y un entero $K \leq |C|$, ¿existen K conjuntos disjuntos en C?

Solución

Partimos de que el problema INDSET = {(G,K) /G es un grado que contiene un conjunto independiente de tamaño k } es NP-completo, he intentamos reducir este al SET PACKING

En este problema del empaquetado de conjuntos, tenemos una lista C de n conjuntos con k <= C. La meta es ver si hay una colección de k conjuntos tales que no hay un elemento que está contenido en dos de estos k conjuntos.

Para ver que SETPACK es NP-completo, lo reducimos al problema INDSET. Dado un grafo G = (V,E), construimos una familia de conjuntos cuyos elementos son los bordes de G.
De esta forma, habrá un conjunto para cada vértice del grafo.
Esto es, para cada nodo V_i V, creamos un conjunto $S_i = \{ \{v_i,v_j\} / \{v_i,v_j\} \in E\}$.

El conjunto asociado con el vértice vi es el conjunto de todos los lados que van a parar a v_i.
Por tanto INDSET $\leq_p$SETPACK.

7. **Partición de conjuntos** [SET COVER]: Dada una familia C de subconjuntos de un conjunto finito S ¿existe un subconjunto de C de tamaño k que cubre S? En otras palabras, ¿existe $C' \subset C$ con $|C'| = k$ tal que $\cup_{C_i \in C'} C_i = S$? S_2).

Solución

Dada una instancia del problema SET COVER, U = x1, x2,...,xn, una colección de m subconjuntos: $S_i \subseteq U$ y de un entero k. La pregunta es si se puede seleccionar una colección C con k de esos subconjuntos para cubrir U. Es decir, buscar un conjunto C tal que:

$$\bigcup_{i\, e\, C} Si = U.$$

Veamos que SET COVER es NP-Completo. Para ello primero vamos a discutir si es NP, dada la colección de conjuntos C, queremos ver que:

$$Vertex\ Cover \leq p\ Set\ Cover$$

Dada una instancia del problema Vertex cover, vamos a construir otra para el problema del Set Cover. La construcción puede hacerse en tiempo polinómico y para ver si es correcta tenemos que comprobar que la instancia de ambos es afirmativa.

Supongamos que G tiene un vertex cover de tamaño j. Y sea S el conjunto de nodos, por construcción, S corresponde a la colección C de subconjuntos U. Haciendo k=j, tenemos como máximo k subconjuntos. Por tanto, los subconjuntos de C cubren a U.

Ahora supongamos que existe un set cover C de tamaño k en nuestra instancia, como está asociado al vértice en G, podemos llamar S al conjunto de vértices. Se cumple que tienen S y C el mismo número de elementos. Por tanto, consideramos un ‘edge’ e. Como está en U y C es un set cover, C tiene que incluir al menos un conjunto que tenga a e dentro. Entonces, S tiene que contener al menos uno de los e. Como nuestra construcción tiene tiempo polinómico, y sabemos que Set cover es NP ⇒ Set cover es NP - Completo.

Por ello, Vertex Cover es un caso particular del problema Set Cover.

8. **Partición de conjuntos (bis)** [Set splitting]: Dada una familia C de subconjuntos de un conjunto finito S ¿existe una partición de S en dos partes S_1 y S_2 tales ningún elemento $A \in C$ esté contenido ni en S_1 o ni en S_2 (todo $A \in C$ debe de tener intersección no vacía con S_1 y con S_2). *Pista: Reducir desde 3-SAT.*

Solución

El problema de Set-Splitting es NP puesto que se puede reducir desde el problema NAE-3-SAT. Sabemos que la entrada a ese problema es una función booleana f de m cláusulas
$\{C_1,...,C_m\}$ sobre n variables $\{x_1,...,x_n\}$. Se puede crear de la siguiente manera.

Consideramos una instancia del problema Set-Splitting (F,S,S_1,S_2). Consideramos S el conjunto que contiene todas las variables del problema NAE-3-SAT y sus negaciones. La familia F de subconjuntos de un conjunto finito S está formada por:

1) Para cada par de parejas, (variable, negación variable) se crea un subconjunto para la colección 1.
2) Para cada cláusula de tres elementos se crea un subconjunto en la colección 2.

El subconjunto S se divide en dos subconjuntos según sean las variables verdaderas o falsas. Cada subconjunto debe contener al menos una variable verdadera y otra falsa, justo lo que se requiere para el problema NAE-3-SAT.

9. **Estrella de Steiner en grafos** [STEINER TREE]: Dado un grafo $G = (V, E)$, y un subconjunto $R \subseteq V$, y un entero positivo $K \leq |V| - 1$, ¿existe un subárbol de G que contiene todos los vértices de R y que no contiene más de K arcos?

Solución

Buscamos reducirlo al problema de “Set cover”

Dada una instancia del problema SET COVER, construimos una “estrella” con el vértice central v. Cada punta de esa estrella se corresponde con un subconjunto del total, cuyo peso será igual que el coste de ese conjunto.

Para cada elemento i = 1...N, tenemos un grupo gi en el problema de GROUP STEINER TREE. Cada uno de esos gi consiste en todos los vértices incidentes que contienen al elemento i.
Esta reducción nos permite utilizar este problema para resolver el del SET COVER.

10. **Matching 3D** [PERFECT 3-MATCHING]: Dados tres conjuntos U, V, W de tamaño n y un conjunto $S \subseteq U \times V \times W$, ¿existe un subconjunto $T \subseteq S$ de tamaño n que incluya cada elemento de $U \cup V \cup W$ exactamente una vez?

NOTA: El matching bidimensional está en P (p.ej. se puede resolver como un problema de flujo en redes).

Solución

El problema del Matching 3D es un problema NP ya que un algoritmo no determinista que elige un subconjunto M y comprueba si hay en cada elemento del subconjunto uno y sólo uno de los elementos de cada conjunto tiene tiempo polinómico.

Para demostrar que es completo para NP, vamos a reducir el problema 3SAT a MATCHING 3D. Supongamos un ejemplo de 3SAT con los siguientes datos

$$U = \{u1, u2, ...un\},\ C = \{c1, c2, ..., cm]\}$$

Vamos a construir un ejemplo de MATCHING 3D con la misma solución. Construiremos los conjuntos W,X,Y y el subconjunto de tripletas M $\subseteq$ WxXxY. Para cada variable u_i vamos a introducir en W los elementos $u_i[j]$, $\overline{ui}[j]$ donde j=1,...,m, en X los elementos $a_i[j]$ y en Y los elementos $b_i[j]$.

En M se introducen las tripletas

$T_i^t = \{(u_i[j], a_i[j], b_i[j]) : 1 \le j \le m\}$ y

$T_i^f = \{(u_i[j], a_i[j+1], b_i[j]) : 1 \le j < m\} \cup \{(u_i[m], a_i[1], b_i[m])\}$.

Los valores a_i , b_i no aparecen en otras tripletas. El efecto es que en S′ deberán de estar todos los elementos $u_i[1]$, $u_i[2]$, $u_i[3]$, $u_i[4]$ o todos los elementos $u^{-}{}_i[1]$, $u^{-}{}_i[2]$, $u^{-}{}_i[3]$, $u^{-}{}_i[4]$.

Luego, podemos ver que este problema es NP-Completo.

11. **Partición en subgrafos hamiltonianos** [HAMILTONIAN SUBGRAPH PARTITION]: Dado un grafo $G = (V, E)$, y un entero positivo $K \leq |V|$, ¿pueden particionarse los vértices de G en $k \leq K$ conjuntos disjuntos $V_1, V_2, \ldots, V_k$ de tal forma que para todo $1 \leq i \leq k$, V_i contiene un circuito hamiltoniano?

Solución

Reducimos este problema al de ciclo hamiltoniano no dirigido ya que el caso k=1 es precisamente ese, por lo que podemos hacerlo. Sabemos entonces que es NP-hard, veamos que es NP.

Sea T una solución al problema. Como podemos comprobarla en tiempo polinómico entonces hemos probado que el problema es NP-Completo.

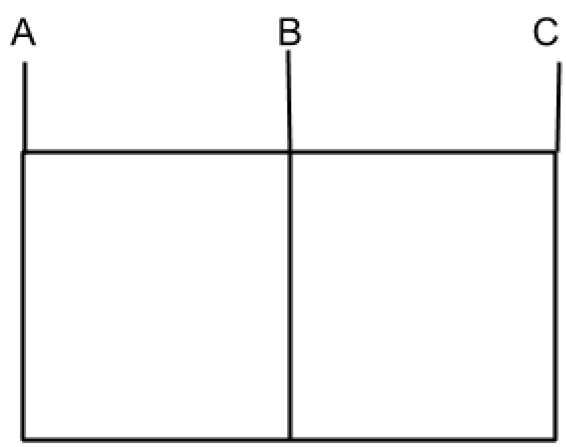
12. **Partición en caminos de longitud 2** [PATH PARTITION]: Dado un grafo $G = (V, E)$, con $|V| = 3q$ donde q es un entero positivo, ¿existe una partición de V en q conjuntos disjuntos $V_1, V_2, \ldots, V_q$ de tal forma que para cada $V_i = \{v_{i[1]}, v_{i[2]}, v_{i[3]}\}$, al menos dos de los tres posibles arcos, $\{v_{i[1]}, v_{i[2]}\}, \{v_{i[1]}, v_{i[3]}\}, \{v_{i[2]}, v_{i[3]}\}$ está en E. *Pista: Reducir PERFECT 3-MATCHING...*

Solución

La partición en caminos de longitud 2 es un problema equivalente al “Edge-Cover”, el problema de la **cobertura mínima de arista** que trata de encontrar una cobertura de aristas de tamaño mínimo.

Este es un problema de optimización que pertenece a la clase de problemas de cobertura y puede resolverse en tiempo polinomial.

Vamos a tomar las combinaciones posibles de 3 elementos aceptados por el 3-matching, y para cada combinación vamos a modificar el grafo añadiendo la siguiente estructura



Esta estructura se da en caso de que tengamos 2 ligaduras, si tuviéramos n, tendríamos que añadir 2n+1 vértices de esa forma, dejando uno entre medias de cada 2, menos una ligadura que tendría 2 en medio.

Una vez tenemos hecha esta transformación, le aplicamos el Perfect 3-matching y tenemos el resultado, por lo que este problema es reducible al Perfect 3-marching, y por tanto es NP-Hard, ahora veamos que es NP.

Sea T una solución al problema. Como podemos comprobarlo en tiempo polinómico $\Rightarrow$ es NPC

13. **Conjunto de vértices de realimentación** [FEEDBACK VERTEX SET]: Dado un grafo dirigido $G = (V, A)$, y un entero positivo $K \leq |V|$, ¿existe un subconjunto $V' \subseteq V$ tal que $|V'| \leq K$ y tal que todo circuito dirigido en G incluya al menos un vértice de V'?

Solución

Como ya sabemos el problema Vertex Cover es NP-completo, por tanto lo reduciremos a este problema y por tanto ya tendremos la solución. Sabemos que un instancia del Vertex Cover consiste en un grafo G=(V,E) y un número k.

Nuestro problema de decisión será determinar si existe un recubrimiento de vértices que tenga al menos tamaño k en G. Vamos a definir un nuevo grado H en un conjunto de vértices U_v= V correspondiente a los vértices y U_e=E correspondiente a las aristas de G.

Para cada arista a, formada por los vértices v_1 y $v_2 \in$ E, tendremos tres aristas en H, una que estará entre los vértices v_1 y v_2 , la segunda entre $v_1 \in U_v$ y a $\in U_e$, y la última entre $v_2 \in U_v$ y a $\in U_e$.

Por tanto ya tenemos que nuestro grado H tiene un Feedback Vertex Set de tamaño menor o igual que k si y solamente si G tiene el recubrimiento de vértices de tamaño menor o igual que k.

Luego nuestro problema del **Feedback Vertex** es NP-completo porque hemos conseguido reducir el Vertex Cover que sabemos que es NP-completo a él.

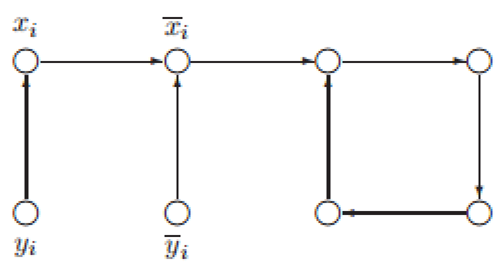
14. **Numeración en grafos** [GRAPH GRUNDY NUMBERING]: Dado un grafo dirigido $G = (V, A)$, ¿existe una numeración $L : V \to \mathbb{N}$, donde el mismo número puede asignarse a más de un vértice y tal que cada $L(u)$ es igual al mínimo de todos los valores enteros que no están en $\{L(v) : (u, v) \in A\}$.

Solución

El problema **GRUNDY** es claramente NP, porque uno puede simplemente imaginar una asignación de números no negativos, y verificar en un tiempo polinomial que para cada vértice la condición para un Grundy-numbering se mantiene.

Para demostrar que 3-SAT es p-reducible a GRUNDY, expongamos los (sub)grados H1,...,Hn y continuamos la construcción de G(α) como sigue: Para cada cláusula Cj de α (1 > j > m), expongamos un subgrafo cumpliendo:

- Para cada x_i añadimos una estructura H_i a D con la siguiente forma:



- Para cada cláusula C_j añadimos los vértices A_j, B_j, C_j y los lados (A_j, B_j), (B_j, C_j), (C_j, A_j) a D.
- Para cada aparición de la variable x_i en C_j añadimos a D los lados (C_j, y_i)

Ahora podemos ver fácilmente que podemos aplicar la reducción, por lo tanto:

$$Cj \to Aj \to Bj \to Cj$$

15. **Conjunto dominante** [DOMINATING SET]: Dado un grafo $G = (V, E)$ y un entero positivo $K \leq |V|$, ¿existe un subconjunto $V' \subseteq V$ tal que $|V'| \leq K$ y tal que todo vértice $v \in V - V'$ está conectado con al menos un vértice de V'?

Solución

El problema del conjunto dominante (DOMINATING SET) es NP, ya que dada una solución podemos verificarla en tiempo polinomial. Esto se puede hacer cogiendo cada vértice y comprobando si está en el Set o en uno de sus lados.

Para ver que es NP-completo, lo primero que debemos observar es que un Dominating Set tiene que incluir todos los vértices aislados, por lo que suponemos que nuestro grafo no tiene vértices aislados. Para demostrar que este problema es NP-completo vamos a reducirlo al Vertex Cover.

Dado un grafo G, vamos a construir G_0 como sigue:

G_0 posee todos los vértices y lados de G. Además, para cada lado (u,v) perteneciente a G, añadimos un nodo intermedio para generar un camino paralelo en G_0.

Ahora vamos a ver que G tiene un Vertex Cover de tamaño k $\Leftrightarrow$ G_0 tiene un Dominating Set del mismo tamaño.

Si S es un Vertex Cover en G, veremos que S es un Dominating Set en G_0. Sea S una Vertex Cover, por lo que cada lado en G tiene al menos uno de sus vértices en S.

Consideremos $v \in G_0$. Si v es un nodo de G, entonces o $v \in S$ o hay algún lado conectando v a algún otro vértice u. Como S es un Vertex Cover, $v \notin S$, entonces u debe pertenecer a S y por tanto hay un vértice adyacente a v en S. Por tanto v es cubierto por algún elemento de S. En caso de que w es un nodo adicional en G_0, entonces w tiene 2 vértices adyacentes pertenecientes a G y usando el argumento previo al menos uno de ellos está en S, por lo que los nodos adicionales también están cubiertos por S. Por lo que si G tiene un Vertex Cover, entonces G_0 tiene un Dominating Set de al menos el mismo tamaño (podría ser incluso el mismo). Si G_0 tuviera un Dominating Set D de tamaño k, entonces todos los vértices adicionales pertenecen a D. Fijémonos en que cada vértice de D tiene que estar conectado exactamente a 2 vértices de G. Podemos ver ahora que podemos simplemente reemplazar cada vértice w de D por uno de G, u o v. El vértice w en D nos ayudará a dominar solo a u, v, $w \in G_0$. Pero estos 3 forman un 3-ciclo, y por tanto podemos tomar u o v y seguir dominando todos los vértices que w dominaba. Por lo que podemos eliminar todos los vértices adicionales gracias a esto. Como todos los vértices adicionales corresponden con uno de los lados de G, y como todos los vértices están cubiertos por el D ya modificado, vemos que todos los caminos de G están cubiertos por el Set. Finalmente vemos la equivalencia que buscábamos, demostrando así que es NP-completo.

16. **Circuito hamiltoniano alternativo** [ANOTHER HAMILTONIAN CYCLE]: Dado un grafo y un circuito hamiltoniano, determinar si el grafo tiene otro circuito hamiltoniano.

Solución

Sabemos que un grafo Hamiltoniano cúbico tiene al menos 2 ciclos Hamiltonianos. Por tanto, cómo encontrar un ciclo ya es de por sí NP-hard en general, y no conocemos ningún algoritmo para encontrar un segundo ciclo Hamiltoniano cuando ya se ha dado uno por entrada.

17. **Suma de cuadrados mínima** [MINIMUM SUM OF SQUARES]: Dado un conjunto finito A y un tamaño $s(a) > 0$ para todo $a \in A$ y dos enteros positivos K y J, ¿pueden particionarse los elementos de A en K conjuntos disjuntos, $A_1, \ldots, A_k$, de tal forma que $\sum_{i=1}^{K} \left(\sum_{a \in A_i} s(a)\right)^2 \leq J$?

Solución

Vamos a reducir este problema al PARTITION de la práctica anterior.

Sea $K = 2$ y $J = B^2/2$.

Si particionar fuese posible $\Rightarrow$ $(B/2)^2 + (B/2)^2 = B^2/2 = J$.
Si no fuera posible $\Rightarrow$ Para cada A_1 y A_2 ,tenemos $\Sigma A_1 = B/2 - \varepsilon$ donde $\varepsilon > 0$
$\Rightarrow$ la suma será $B^2/2 + 2^* \varepsilon^2$.

Por tanto queda definida la reducción y probado que Minimum Sum of Squares es NP-Completo..

18. **Equivalencia de expresiones regulares sin estrella** [STAR-FREE REGULAR EXPRESSION EQUIVALENCE]: Dadas dos expresiones regulares E_1 y E_2 sobre el alfabeto A que no contienen el operador de clausura $*$, ¿representan estas expresiones regulares lenguajes distintos sobre A?

Solución

Dadas dos expresiones regulares E_1 y E_2 sobre el alfabeto A que no contienen el operador de clausura *, podemos darnos cuenta de que cualquier cadena en L(R), para alguna expresión regular R, tiene longitud |R|. De modo que, para determinar la pertenencia al lenguaje, sólo necesitamos ver que una cadena es $L(R_1)$ y no es $L(R_2)$ o viceversa.

Un NTM puede recibir una cadena para probar en cada dirección y entonces dar un resultado en tiempo polinomial, por lo que este problema es NP.

Vamos a reducir **START-FREE REGULAR EXPRESSION EQUIVALENCE** al problema SAT que sabemos ya es NP-Completo

Suponemos que hay variables $x_1, \ldots, x_n$ y cláusulas $C_1, \ldots, C_m$ en la expresión booleana de partida. Podemos convertir esto en una expresión regular sobre el alfabeto {0,1}. La primera expresión regular (R_1) será la unión de las asignaciones inválidas de las variables. R_2, por tanto, consiste en simplificar la expresión regular (0+1)(0+1)....(0+1), la cual representa las asignaciones posibles de las n variables. Fijémonos en que si $L(R_1) = L(R_2)$, entonces todas las asignaciones fallan y las correspondiente fórmula del SAT no puede ser satisfecha. Si no son iguales, entonces existen algunas asignaciones que no son fallidas. Por tanto, la fórmula original sería satisfacible, y la reducción estaría completa. Lo único que queda es decir cómo está hecho R_1. Una asignación puede fallar si toda cláusula de la expresión booleana devuelve falso. Para una cláusula dada, podemos generar una expresión regular $(a_1, \ldots, a_n)$ que corresponda a una asignación fallida asegurándonos de que todos los literales son falsos de la siguiente manera:

- Asignamos el valor 0 a a_i si x_i aparece en la cláusula en cuestión
- Asignamos el valor 1 a a_i si el complemento de x_i aparece en la cláusula en cuestión
- En otro caso, asignamos a a_i el valor (0+1), que corresponde a que x_i no aparece en la cláusula en cuestión

Entonces, R_1 es formado tomando la unión de estas expresiones regulares para cada cláusula $C_1, \ldots, C_m$.

Y el problema queda demostrado que es NP-completo.

19. **Red distribuida de telefonía móvil**: Tenemos un grafo no dirigido G=(V,E) en el que los vértices son personas y las aristas nos indican si dos personas están dentro del alcance de las señales que emiten sus móviles. Cuando dos personas están hablando entre sí, sus vecinos no pueden utilizar la misma frecuencia para evitar interferencias en la conversación que está teniendo lugar. Por tanto, un conjunto de conversaciones consiste en un conjuno de aristas $C \subset E$ en el que los vértices de las distintas aristas de C no pueden ser vecinos entre sí.

    La capacidad de la red dada por G es el número máximode conversaciones simultáneas que pueden tener lugar en una misma frecuencia (el tamaño del mayor conjunto posible C). Dado el siguiente problema de decisión, demuestre que es NP-completo:

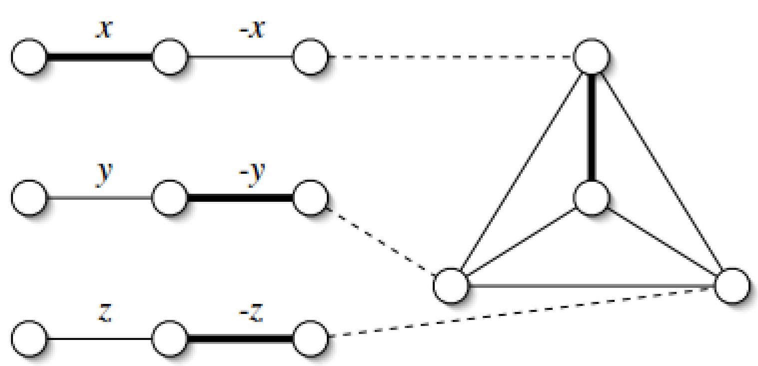
    > Dado un grafo G y un entero k, ¿existe un conjunto C de conversaciones con $|C| \geq k$?

    [Moore & Mertens: "The Nature of Computation", problem 5.7, pp. 164]

Solución

Para probar que es NP tomamos un conjunto que sea comprobable en tiempo polinómico S, se comprueba que $|S| \geq k$ con S c E y que cumple la condición dada en el enunciado.

Vamos a reducirlo a partir del problema 3SAT. Dada una instancia de n variables y m cláusulas, convertimos S a un grafo G de la siguiente manera:



Cada variable tiene un vértice central que puede hablar con uno, pero no con el resto de sus vecinos. Esto se convierte en una especie de triángulo donde cada vértice está conectado a otra variable más un vértice central que está conectado a las otras 3. Definimos ahora k=n+m.

La idea está en ver que uno de los extremos puede estar en S si uno o varios vecinos no están en S. En el caso de estar los tres vecinos en S, ninguno de los extremos de la cláusula podría estarlo también. Estos dos casos se corresponden con que la cláusula sea satisfacible o no.

Ahora supongamos que es satisfacible. S contiene uno de los extremos de cada variable y cada cláusula. Por tanto tenemos $|S| = k = n + m$.

Por otro lado, supongamos que C(G) $\geq$ k. Notamos que ninguno de los extremos conectados entre las cláusulas y las variables puede estar en S puesto que en ese caso ninguno de los extremos en esas cláusulas puede estar en S. Definimos una asignación cierta para la cual un extremo de una variable está en S, entonces la observación anterior implica que la asignación que sabíamos cierta satisface la fórmula y por tanto quedaría así demostrado.

20. **Spin glass** (`http://en.wikipedia.org/wiki/Spin_glass`): Un "vidrio de espín" es un sistema magnético en el que el acoplamiento entre los momentos magnéticos de los distintos átomos es aleatorio (algo así como un imán desordenado). Formalmente, puede verse como un grafo en el que el vértice i-ésimo tiene un espín $s_i = \pm 1$ que

representa si el campo magnético del átomo i-ésimo apunta hacia arriba o hacia abajo. Cada arista (i, j) del grafo tiene asociada una fuerza de interacción J_{ij} que indica hasta qué punto interactúan s_i y s_j. Además, cada vértice i está sometido a un campo externo h_i. Dada una configuración del sistema (un conjunto de valores para los s_i), su energía es

$$E(\{s_i\}) = -\sum_{ij} J_{ij} s_i s_j - \sum_i h_i s_i$$

La arista (i, j) es ferromagnética si $J_{ij} > 0$ y antiferromagnética si si $J_{ij} < 0$; esto es, el ferromagnetismo tiende a alinear los espines. El campo externo es positivo o negativo en función de si queremos que s_i sea $+1$ o -1, respectivamente.

Ya vimos que, en el caso ferromagnético, el problema está en P. Sin embargo, en cuanto algunas aristas son antiferromagéticas, el tema se complica, incluso en ausencia de un campo magnético externo. Demuestre que la siguiente versión del problema es NP-completa:

> Dado un grafo G con interacciones J_{ij} y un nivel de energia E, ¿existe un estado $\{s_i\}$ tal que $-\sum_{ij} J_{ij} s_i s_j \leq E$?

Pista: Muestre que el problema es NP-completo en ausencia de un campo externo cuando todas las aristas son antiferromagnéticas con $J_i j = -1$. Observe que, en este caso, buscaremos un corte maximo (MAX CUT)...

[Moore & Mertens: "The Nature of Computation", problem 5.30, pp. 167-168]

Solución

Para ver que es NP-Completo, vamos primero a ver si dada una solución podemos verificarla en tiempo polinómico y después vamos a reducir el MAX CUT a este problema.

Sea P una solución, podemos observar fácilmente que comprobar si esta solución verifica o no la desigualdad se puede hacer en tiempo polinómico. Procedamos a la reducción:

Dado un problema de MAX-CUT ponderado con un grafo G donde cada arista (v_i,v_j) tiene un peso w_{ij}. Consideramos ahora que para cada vértice, si no existe la arista (v_i,v_j), entonces $J_{ij}=0$ y si existe, entonces $J_{ij}= - w_{ij}$, además $h_i=0$.

Si ahora consideramos 2 conjuntos de vértices V_1 y V_2 podemos considerar que V_1 tiene un spin +1 y V_2 tiene un spin -1.

Sobre los vértices de un mismo conjunto sumamos $-J_{ij}$ y en los de distinto sumamos J_{ij}. Así, si el valor del corte es K, entonces el valor de la energía es $-\Sigma_{ij} J_{ij} - 2K$. Ahora, si en el MAX-CUT tenemos un límite de K, en este tomaremos como límite $E = sum_{ij} w_{ij} - 2K$.

En estas condiciones, la reducción es inmediata, por lo que queda demostrado.

21. La autoridad portuaria del Puerto de Motril está preocupada porque sus ingresos se ven limitados por la velocidad a la que se pueden descargar los contenedores de los barcos que llegan al puerto. El manejo de sustancias peligrosas le añade complejidad adicional al problema, que ya les resulta complicado de por sí. Como consultor, le piden que analice el siguiente problema:

    Supongamos que llega un convoy de barcos por la mañana con un total de n contenedores, cada uno de los cuales contiene distintas sustancias (algunas de ellas peligrosas). En el muelle, hay m camiones disponibles, cada uno de los cuales puede transportar hasta k contenedores. Cualquier contenedor se puede transportar en cualquier camión, pero existen ciertos pares de sustancias químicas que no pueden cargarse en el mismo camión (podrían provocar una explosión si entrasen en contacto). ¿Existe alguna forma de cargar los n contenedores en los m camiones de forma que ningún camión esté sobrecargado y no se carguen en el mismo camión sustancias que no deban transportarse juntas? Demuestre que se trata de un problema NP-completo.

Solución

Sabemos que el problema es NP ya que si distribuimos los n contenedores en los m camiones se puede probar en tiempo polinómico si ningún camión está sobrecargado y que no hay pares de sustancias que no puedan ir juntas en un mismo camión.

Vamos a reducir el Matching 3D al problema de los contenedores, para ello consideramos un subconjunto UxVxW que lo llamamos S, donde U,V,W son conjuntos finitos de tamaño n. Denotamos m=|S|, que será el número de camiones disponibles. Lo que queremos es l=m - n .
Pongamos que cada camión puede transportar como máximo 4 contenedores.Vamos a denotar;
a_i a la coordenada de U ,b_i a la coordenada de V y c_i a la coordenada de W que en nuestro caso representarán a los contenedores. Por cada elemento de nuestro conjunto S consideramos el contenedor x_i , x_j han de ir en distintos camiones diferentes $\forall$ i != j , por lo que cada x_i irá en un camión diferente. Ahora añadimos lo mismo para contenedores y_i , donde dos de ellos no podrán ir en el mismo camión pero los x_i y los y_j si pueden ir en el mismo.
x_i es compatible con a_i , b_i y c_i e incompatible con los demás.

Por tanto podemos encontrar otro subconjunto de S que podemos llamar T que será de tamaño n y que contenga a cada elemento de U $\cup$ V$\cup$ W una y solo una vez si y solo sí se pueden transportar todos los contenedores.

Supongamos S' $\subseteq$ S una solución del matching 3D, colocamos cada x_i en un camión distinto y los y_1...y_l van en los camiones en los que van los elementos S\ S'.

Los elementos U,V,W van acompañando a los x_i que pertenecen a S'. Por tanto cada $x_i \in S'$ lleva a los elementos a_i , b_i , c_i. Y se cargaran con un límite de 4 contenedores por camión.

Recíprocamente, si hay una carga de los camiones, cada xi tiene que ir en un camino distinto. Cada yj ir en un camino acompañando a un xi. El matching 3-D viene dado por los xi que no llevan ningún yj y que tendrán que llevar a sus elementos. Como cada contenedor ai, bi, ci va en uno y un sólo camino.

Entonces, cada elemento estar en uno y sólo uno de los vectores seleccionados.

Como el problema Matching 3D es NP-Completo (ejercicio 10) tenemos que también lo es el problema que nos dan. Si cambiamos la formulación del problema, de forma que, para cada sustancia química dispongamos de un subconjunto de camiones en los que es seguro transportarla, entonces para demostrar que el problema está en la clase P lo planteamos como un problema de flujo máximo.

El grafo estará formado del siguiente modo: Un nodo s, un nodo por cada contenedor, otro nodo por cada camión y un último nodo que llamamos destino. Por otra parte tenemos por cada contenedor una arista de peso 1 que va del nodo s al nodo correspondiente al contenedor.

Ahora, por cada contenedor ponemos arcos que van del nodo correspondiente a ese contenedor a cada camión donde es seguro que se transporte, cada arco de peso 1.

Finalmente, por cada camión, añadimos una arista que va del nodo correspondiente a ese camión al nodo destino de peso el número máximo de contenedores que puede transportar cada camión. El flujo máximo de esa red es igual al número máximo de contenedores que se pueden transportar en los camiones.