Relación de Problemas Nº 4
Problemas NP-completos

Serafín Moral Callejón
Cristina Zuheros Montes

6 de junio de 2015

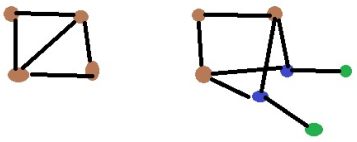
Nota. No vamos a detallarlo, pero es importante hacer notar que todas las reducciones que vamos a dar son polinómicas, aunque no se diga. Todas conllevan cálculos elementales que se pueden realizar con algoritmos polinómicos.

1. Dado un camino en un grafo, podemos comprobar que es un camino (que no repite vértices), y su longitud en tiempo polinómico, por lo que el problema es NP.

   Ahora veamos que es NP-completo.

   Empezamos reduciendo el problema del ciclo Hamiltoniano al problema del camino Hamiltoniano.

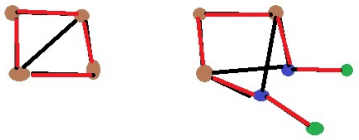
   Si tenemos un grafo G con n vértices y un conjunto de aristas, contruimos un nuevo grafo G' en el cual sustituimos un vértice por dos copias suyas (dos nuevos vértices), los cuales conectamos con todos los vértices con los que estaba conectado el vértice original en el grafo original, pero no conectamos las copias entre ellas. Además, añadimos dos nodos nuevos (satélites) que estarán conectados sólo conectamos con las dos copias que hemos añadido. Es claro que nuevo grafo G' contiene un camino Hamiltoniano si y sólo si, el original contiene un ciclo Hamiltoniano.

   Si el original tiene un circuito Hamiltoniano, el camino Hamiltoniano comienza en un satélite va a la copia del nodo duplicado con el que está conectado, sique por el circuito Hamiltoniano hasta la segunda copia y va al segundo satélite.

Recíprocamente, si hay un camino Hamiltoniano, entonces empieza y termina en los satélites. Si quitamos los satélites y volvemos a juntar los nodos, obtenemos un circuito Hamiltoniano en el grafo original.

La correspondencia puede verse en el siguiente dibujo.



Ahora reducimos el problema del camino Hamiltoniano al problema del camino más largo. En efecto, si tenemos un grafo G con n vértices, el hecho de que el grafo tenga un camino Hamiltoniano es equivalente a que el grafo tenga un camino simple cn $k-1$ arcos. Si pasa por todos los vértices una sóla vez entonces el número de aristas que recorre es $k-1$. Y recíprocamente, si recorre $k-1$ aristas sin repetir vértices, entonces pasa por todos los vértices sin repetir ninguno, luego el grafo tiene un camino Hamiltoniano.

Como sabemos que el problema del circuito Hamiltoniano es NP-completo y se puede reducir al problema del camino más largo concluimos que este último es NP-completo.

Como los cálculos consisten en duplicar un nodo y algunos enlaces y añadir dos nodos y enlaces, todos los cálculos se pueden hacer en tiempo polinómico.

2. Dada una función $f : V \to \{1, 2, 3\}$, podemos comprobar en tiempo polinómico que $f(u) \neq f(v)$ para cada arista (u, v) de G, por lo que el problema está en la clase NP.

   Veamos ahora que es NP-Completo

   En primer lugar vamos a reducir el 3-Sat al NAE-3-SAT.

   Consideramos una instancia del problema 3SAT. Añadimos una nueva variable z (la idea es que sea siempre falsa en NAE-3-SAT). Sea $(x_1 \vee x_2 \vee x_3)$ una cláusula del problema de 3SAT (los x_1, x_2, x_3 pueden ser literales positivos o negativos). Añadimos una variable y (distinta de cada cláusula) y las clásulas:

   $x_3 \vee y \vee z$, $x_1 \vee x_2 \vee \neg y$, $\neg x_1 \vee y \vee z$, $\neg x_2 \vee y \vee z$

   Es inmediato comprobar que las clásulas originales se pueden satisfacer si y solo si el problema NAE-3-SAT tienes solución con los valores de verdad de las variables de 3-SAT, z falso e y con el valor de verdad de $x_\vee x_2$.

   Recíprocamente, si hay una solución al NAE-3-SAT, entonces si z es verdadero, se cambia el valor de verdad de todas las variables y sigue siendo

una solución con z falso. El valor de verdad de las variables del problema original hace que 3-SAT se satisfaga.

Como todo consiste en añadir variables y crear un número fijo de clásulas por cada una de 3-SAT, los cálculos se pueden hacer en tiempo polinómico.

Ahora reducimos el NAE-3-SAT al Graph 3-Coloring:

Para cada variable que aparezca en nuestro problema NAE-3-SAT vamos a considerar dos vértices, que presentan variable y su negación. Establecemos una unión dos a dos (el vértice del literal y el vértice del literal negado) mediante una arista. Para crear el grafo base $G = (V, E)$, falta añadir un vértice extra, sea v, que conectaremos con una arista a cada uno de los vértices que hemos creado. Este grafo inicial se puede colorear sin problema alguno mediante 3 colores: primer color para v, segundo color para los literales de nuestro problema NAE-3-SAT, tercer color para los literales negados. Ya tenemos el grafo con los literales, tenemos que representar las cláusulas para que la reducción sea correcta. Supongamos la cláusula $(x1 \vee \neg x2 \vee x3)$. Creamos un grafo aparte S donde los vértices sean los literales de la cláusula, y conectamos los tres vértices mediante aristas. Vamos a conectar este grafo S creado a partir de la cláusula con el grafo G: Unimos mediante una arista cada vértice de S con su correspondiente literal negado del grafo G.

Todo coloreado de un grafo con 3 colores corresponde con una solución del problema NAE-3-SAT:

Si tenemos una solución para NAE-3-SAT, vamos a suponer que los tres colores del grafo son C_1, C_2, C_3.

Entonces le asignamos C_1 al vértice extra v, después para cada arista que une una variable y su negación, le asignamos C_2 a los literales que sean verdaderos y C_3 a los falsos (Si u_i es cierto, entonces u_i recibe el color C_2 y $\neg u_i$ el C_3. Finalmente en los triángulos de las cláusulas, asignamos C_2 a un literal cierto (que está unido con un nodo de color C_3), C_3 a un literal falso (que está unido con un nodo con color C_2) y C_1 al literal restante.

Recíprocamente, si tenemos una coloración por 3 colores del grafo, podemos suponer que el nodo extra v tiene color C_1 (en otro caso una permutación de los colores sigue siendo una solución). Entonces, haciendo u_i verdadero si recibe el color C_2 y falso si recibe el color C_3, se consigue una asignación de valores de verdad tal que para cada cláusula hay un literal verdadero (el que recibe el color C_2 en el triángulo) y otro falso (el que recibe el color $C_2 3$).

Todas las transformaciones se pueden hacer en tiempo polinómico ya que consisten en crear un nodo, un enlace por cada variable, un triángulo por cada cláusula, y tres enlaces adiciones por cada clásula.

3. Dada una cobertura de un grafo G, se puede comprobar en tiempo polinómico su tamaño y que ningún par de vértices que están en la cobertura forman un par de vértices adyacentes, luego el Vertex Cover es NP.

Vamos a reducir el 3-SAT al Vertex Cover.

Consideramos un conjunto de cláusulas para el problma 3-SAT que contenga m variables y n cláusulas. Vamos a construir un grafo G del siguiente modo: Por cada cláusula construimos un subgrafo completo con tres nodos, cada uno de ellos .^{et}iquetándolosçon el nombre de uno de los literales de la cláusula. Por otra parte, por cada variable, ponemos dos vértices, uno que se corresponde con ella misma y otro con la negación, y conectamos esos dos vértices con una arista. En cada uno de los triángulos, unimos cada vértice con otro que tenga "su misma etiqueta". Tomamos $k = m + 2 * n$ y veamos que el grafo G tiene una cobertura de tamaño k o menor si y sólo si la fórmula dada es satisfacible.

Supongamos que la fórmula es satisfascible. Contruimos un conjunto C así: Para línea que une una variable y su negada, añadimos a C el nodo correspondiente al literal para el cual la fórmula es cierta. Por cada triángulo añadimos dos vértices a C, que se corresponderán con los dos nodos restantes del etiquetado con el literal que hace cierta la cláusula (en caso de haber más de 1 se descarta cualquiera). Se ve fácilmente que C es una cobertura del grafo G.

Recíprocamente, Si G tiene una cobertura C de tamaño $m+2*n$, entonces en esa cobertura hay 2 vértices por cada triángulo y un vértice por cada arista que une una variable y su negada, pues para que C cumpla con la definición de cobertura para cada arista al menos uno de los extremos ha de pertenecer a la cobertura. Si en C aparece un nodo correspondiente a una línea que une una variable y su negada, hacemos el literal correspondiente a la etiqueta de dicho nodo verdadera. Si una cláusula contiene ese literal está claro que será verdadera. En caso contrario, en esa cláusula el nodo correspondiente al literal negado ha de pertenecer a C por definición de cobertura. Por tanto, el otro de los otros dos literales que aparecen en la cláusula lo hacemos falso y el otro verdadero y esto nos da la fórmula para la cual es satisfascible el problema.

4. Si tenemos dos grafos G_1 y G_2, dados dos subgrafos G_1' y G_2' de G_1 y de G_2 respectivamente y una aplicación biyectiva entre los subgrafos, nosotros podemos comprobar en tiempo polinómico si los dos subgrafos son isomorfos, de donde deducimos que el Subgraph Isomorphism está en la clase NP.

   Sabemos que el problema del Clique es NP-Completo, luego para concluir que el Subgraph Isomorphism también lo es reducimos el problema del clique al Subgraph Isomorphism.

   Sea G un grafo y k' un entero. Pretendemos saber si existe un clique de tamaño k'. Sea n el número de vértices del grafo.

   Construimos un ejemplo del problema del isomorfismo de subgrafos considerando que el primer grafo G_1 es G y el segundo G_2 es un grafo que contiene un clique de tamaños k' (k' primeros nodos conectados entre si)

y ninguna arista más. El número de aristas en común de los subgrafos se hace igual a $K = k' * (k' - 1)/2$ (el número de aristas de un clique de tamaño k'). Es decir exigimos que todas las aristas del segundo subgrafo estén presentes.

En el segundo no se pueden quitar aristas porque tiene exactamente K aristas. Si se pueden quitar aristas del primer grafo hasta que sea isomorfo al segundo, es porque tiene un clique tamaño k'. Luego los dos problemas son equivalentes.

La construcción consiste en copiar un grafo y en crear un grafo de n nodos en el que haya sólo un clique de tamaño k' y esto se puede hacer en tiempo polinómico.

5. Dado un corte en un grafo G, podemos comprobar en tiempo polinómico que es un corte que separa los vértics s y t y su tamaño, por lo que el problema MAX CUT es NP.

   Vamos a reducir el problema de la partición al MAX CUT.

   Supongamos que tenemos un conjunto $S = \{u_1, \ldots, u_n\}$ con u_i numeros enteros positivos $I = \{1, \ldots, n\}$. Denotamos $M = \sum_{i=1}^{n} s(u_i)$. Construimos un grafo G de la siguiente manera: Un vértice por cada u_i y todos los vértices del grafo conectados entre sí mediante aristas. El peso de cada arista (u_i, u_j) será igual a $s(u_i) * s(u_j)$. Tomamos $w = (1/4) * M^2$.

   Si podemos particionar el conjunto S en dos conjuntos S_1 y S_2 cuyos elementos sumen lo mismo, es decir, $M/2$, entonces tomando el subconjunto de vértices de G correspondientes a los nodos etiquetados como u_i tales que $u_i \in S_1$ se cumple que la suma de los $s(u_i) * s(u_j)$ tales que $u_i \in S_1$ y $u_j \in S_2$ es igual a $\sum_{u_i \ inS_1} s(u_i) * \sum_{u_j \ inS_2} s(u_j) = M^2/4$, es decir, el tamaño del corte sería $M^2 * (1/4)$.

   Recíprocamente, supongamos un corte del grafo G en un conjunto de vértices V_1 y V_2. Consideramos la partición de S en dos subconjuntos S_1 y S_2 de modo que $S_1 = \{u_i | u_i \in V_1\}$ y $S_2 = S \setminus S_1$. Supongamos que la suma de los elementos que están en S_1 no es $M/2$. Sin pérdida de generalidad podemos suponer que la suma de los elementos de S_1 es $M/2 + \epsilon$ y que la suma de los elementos de S_2 es $M/2 - \epsilon$ para cierto $\epsilon > 0$. Entonces se verifica que la suma de los $s(u_i) * s(u_j)$ tales que $u_i \in V_1$ y $u_j \in V_2$ es igual a $\sum_{u_i \ inS_1} u_i * \sum_{u_j \ inS_2} = (M/2 + \epsilon) * (M/2 - \epsilon) = M^2/4 + \epsilon^2 - 2 * M * \epsilon < M^2/4$. Por tanto, hemos comprobado que si existe un corte del grafo de peso mayor o igual a $M^2/4$ entonces existe una partición de S en dos subconjuntos disjuntos cuyos elementos suman $M/2$.

   Como la transformación consiste en construir un nodo por cada elemento y una arista por cada pareja de elementos con un peso que es una multiplicación, la transformación se puede hacer en tiempo polinómico.

   Como sabemos que el problema de la partición es NP-Completo concluimos que también lo es el MAX CUT.

6. Sea C una colección de conjuntos finitos. Dada una subfamilia de conjuntos podemos comprobar en tiempo polinómico su tamaño, que son conjuntos disjuntos y que todos ellos pertenecen a C, de donde se sigue que el Set Packing pertenece a la clase NP.

   Sabemos por la relación anterior que el Independent Set es equivalente al Clique, por lo que si reducimos el al Independent Set al Set Packing, concluimos que el Set Packing es NP-Completo.

   Dado un grafo G y un entero k, para cada vértice del nodo consideramos el conjunto formado por las aristas adyacentes a dicho vértice. Tenemos así una colección C de conjuntos, uno por cada vértice de G. Los subconjuntos de la familia tal que son disjuntos dos a dos se corresponden a conjuntos de nodos del grafo que no son adyacentes entre sí. Por consiguiente, que haya en el grafo un conjunto independiente de tamaño k es lo mismo que decir que en la familia C haya k conjuntos disjuntos.

   La transformación se hace en tiempo polinómico ya que es crear un conjunto por cada vértice con las aristas que inciden en él.

7. El Set Cover es NP, ya que, si tenemos un conjunto finito S y una familia C de subconjuntos de S, dada una subfamilia de subconjuntos de S podemos comprobar en tiempo polinómico que está contenida en C, que cubre a S y su tamaño.

   Vamos a reducir el Vertex Cover al SET-COVER, y, como sabemos que el Vertex Cover es NP-Completo concluiríamos que el SET-COVER también es NP-Completo.

   Sea G un grafo y k un entero. Consideramos el conjunto S cuyos elementos son las aristas del grafo G y construimos la familia de subconjuntos de S del siguiente modo: Para cada vértice del grafo G, consideramos el subconjunto de S cuyos elementos son las aristas que inciden en dicho vértice. Decir que el grafo tiene una cobertura de tamaño k es lo mismo que decir que S tiene una subfamilia de subconjuntos que cubren a S. En efecto, si tenemos una cobertura del grafo G entonces para cada arista del grafo al menos uno de los dos vértices está en la cobertura. Entonces si consideramos la familia de subconjuntos cuyos subconjuntos son los que se corresponden a los vértices de la cobertura entonces esa familia de subconjuntos recubre a S. El razonamiento inverso es análogo.

   La transformación se hace en tiempo polinómico como en el caso anterior.

8. Sea S un conjunto finito y C una familia de subconjuntos de S. Dados dos subconjuntos de S, S_1 y S_2, se puede comprobar en tiempo polinómico que son subconjuntos disjuntos de S y que ningún elemento de C está ni en S_1 ni en S_2, por lo que el problema está en la clase NP

   Vamos a reducir el 3-SAT al Set Splitting.

   Sea V el conjunto formado por las variables del problema 3-Sat y sea ϕ una fórmula bien formada del problema. Llamamos $V^c = \{x^c | x \in V\}$ Construimos la siguiente instancia del problema Set Splitting:

Tomamos $S = V \cup V^c \cup \{z\}$, siendo z una variable que no está ni en V ni en V^c. Formamos la familia de subconjuntos C de la siguiente manera: Para cada variable x en ϕ consideramos el conjunto $S_x = \{x, x^c\}$. Para cada cláusula cl_i de ϕ consideramos el conjunto $S_{cl_i} = \{\text{variables de la cláusula}\} \cup \{z\}$. Donde suponemos que si la variable x aparece negada, lo que se añade es x^c.

Los problemas son equivalentes, suponiendo que en S_1 se introducen todas las variables ciertas (x si es cierta y x^c si es falta) y en S_2 las faltas (x si es falsa y x^c si es cierta) más z.

Para transformar una solución del segundo problema en una solución de 3-SAT, hay que tener en cuenta que si $z \in S_1$, entonces se intercambian S_1 y S_2 y sigue siendo una solución. Entonces se hacen ciertas las variables de S_1 y falsas las de S_2. Esa asignación de valores satisface todas las cláusulas.

La transformación es en tiempo polinómico ya que la construcción de los conjuntos es lineal en las cláusulas.

9. Sea $G = (V, E)$ un grafo y un subconjunto R de V. Si tenemos un subgrafo de G podemos comprobar en tiempo polinómico que es subárbol, que contiene a todos los vértices de R y el número de arcos que contiene, lo que implica que el Steiner Tree pertenece a la clase NP.

   Vamos a reducir el Exact Cover al Steiner Tree.

   Dada una familia de subconjuntos S de un conjunto C vamos a contruir un grafo G como sigue:

   Añadimos un vértice por cada elemento del conjunto C, así como un vértice por cada subconjunto de la familia S. Conectamos los nodos referentes a los elementos con los nodos correspondientes a los subconjuntos que verifiquen la relación de pertenencia. Le añadimos un vértice adicional que le vamos a llamar n. Por cada nodo del grafo que corresponda a un subconjunto $|S_j|$ insertamos un camino de longitud $|S_j|$ que va del nodo correspondiente al subconjunto $|S_j|$ a n. Como R consideramos el conjunto de vértices formado por cada uno de los vértices que se corresponden con elementos del conjunto junto con el nodo n. Finalmente tomamos $K = 2 * |C|$.

   Si podemos cubrir el conjunto C mediante una familia de subconjuntos disjuntos entonces el subgrafo formado por el nodo n, los nodos correspondientes a los elementos del conjunto, y los nodos correspondientes a los subconjuntos que están en la familia tiene K arcos: Uno por cada elemento del conjunto, que iría de ese nodo al subconjunto al que pertenecen (el subgrafo es conexo porque la familia cubre a C), así salen $|C|$ arcos por ser los subconjuntos disjuntos, y $|C|$ arcos que van de los nodos correspondientes a los subconjuntos al nodo n, pues la suma de las cardinalidades de los $|S_j|$ ha de ser igual a $|C|$ al ser los $|S_j|$ disjuntos.

   Recíprocamente, si el grafo contiene un subárbol A conteniendo a n y a los nodos correspondientes a los elementos del conjunto, entonces claramente ha de contener a algún S_j. Es claro que el subárbol ha de ser conexo, luego

los S_j han de cubrir al conjunto C. Si los S_j no son disjuntos, entonces la suma de las cardinalidades de los S_j es mayor que C, y además habría más de C arcos que irían de los nodos correspondientes a los elementos de C a los nodos correspondientes a los subconjuntos a los que pertenece. Por tanto, el grafo tendría más de K arcos. Así, queda verificado que si si el grafo contiene un subárbol A conteniendo a n y a los nodos correspondientes a los elementos del conjunto, entonces se puede particionar C en una familia de subconjuntos disjuntos.

Hemos reducido el Exact Cover al Steiner Tree y, como sabemos que el primer problema es NP-Completo deducimos que también lo es el Steiner Tree.

10. (del libro de Garey-Jhonson) Comprobar si un subconjunto S de T cumple las condiciones de que cada elemento esté en uno y sólo uno de los vectores se puede comprobar en tiempo polinómico, luego el problema es NP. Para demostrar que es completo, vamos a reducir 3SAT a Matching 3D.

    Sea un problema 3SAT con variables l_i que englobamos en un conjunto $L = \{l_i \,|\, i = 1, \ldots, n\}$ y un conjunto de cláusulas $C = \{c_j \,|\, talquej = 1, \ldots, m\}$.

    Construimos los conjuntos U, V y W: Para cada variable l_i vamos a introducir en U los elementos $l_i[j]$, $l_ic[j]$, en V cierto elementos $a_i[j]$ y en W ciertos elementos $b_i[j], j = 1, \ldots, m$ y $i = 1, ..., n$.

    En el conjunto S introduzco los vectores $T_it = \{(l_ic[j], a_i[j], b_i[j]) \,|\, j = 1, \ldots, m\}$ y $T_if = \{(l_i[j], a_i[j{+}1], b_i[j]) : j = 1, \ldots, m{-}1\} \cup \{(l_i[m], a_i[1], b_i[m])\}$

    Con esta asignación de conjuntos conseguimos que cualquier subconjunto T tenga que contener los conjuntos T_it o T_if. El primero corresponde a l_i verdadero y en el segundo caso a l_i falso.

    Ahora tenemos que tener en cuenta las cláusulas del conjunto C. Para cada cláusula c_j

    Añadimos los elementos $s1[j]$ a V

    Añadimos los elementos $s2[j]$ a W. Añadimos a S los vectores:

    $\{(l_i[j], s1[j], s2[j])$ tal que l_i está en $c_j\} \cup \{(l_ic[j], s1[j], s2[j])$ tal que $\neg l_i$ está en $c_j\}$

    Los subconjuntos T tendrán que contener alguna de estos vectores. Por lo tanto, en cada cláusula habrá un literal que sea cierto.

    Finalmente añadimos los siguientes elementos:

    Añadimos $g1[k]$ con $1 \leq k \leq m(n-1)$ a V.

    Añadimos $g2[k]$ con $1 \leq k \leq m(n-1)$ a W. Añadimos a S los vectores:

    $(l_i[j], g1[k], g2[k]), (l_ic[j], g1[k], g2[k]), 1 \leq k \leq m(n-1), 1 \leq i \leq n, 1 \leq j \leq m$.

    Para ver que los problemas son equivalentes. Supongamos una asignación de valores de verdad que satisface todas las cláusulas, entonces se cogen los

vectores de T_it si u_i es verdadera y las de T_if si es falsa. A continuación por cada cláusula se selecciona $(l_i[j], s1[j], s2[j])$ si u_i aparece es la variable que aparece como positiva en la cláusula y es cierta y $(l_ic[j], s1[j], s2[j])$ si u_i aparece es la variable que aparece como negada en la cláusula y es falsa. De los vectores $(l_i[j], g1[k], g2[k]), (l_ic[j], g1[k], g2[k]), 1 \leq k \leq m(n-1), 1 \leq i \leq n, 1 \leq j \leq m$. se elige unopor cada k con los elementos $l_i[j]$ y $l_ic[j]$ no elegidos anteriormente.

Recíprocamente, si tenemos una solución del PERFECT 3D-Matching, entonces para cada variable se tiene que haber elegido T_it o T_if. En el primer caso hacemos u_i verdadera y en el segundo falsa. Como para cada cláusula hay que elegir uno de los vectores $\{(l_i[j], s1[j], s2[j])$ tal que l_i está en $c_j\} \cup \{(l_ic[j], s1[j], s2[j])$ tal que $\neg l_i$ está en $c_j\}$, el que se elija es el que hace verdadera la cláusula y entonces esa es una solución de 3-SAT.

Como la construcción se basa en crear vectores de longitud 3 que dependen de varios índices que varían entre valores polinómicos, la transformación se hace en tiempo polinómico.

11. Sea $G = (V, E)$ un grafo y K un entero menor o igual ue $|V|$. Dados k subconjuntos de vértices $V_1, \ldots, V_k$ con $k \geq K$ y un circuito Hamiltoniano C_i correspondiente a cada V_i podemos comprobar en tiempo polinómico que esos subconjuntos de vértices forman una partición del conjunto V y que cada C_i es un circuito Hamitoniamp de V_i correspondiente, por lo que el problema dado es NP.

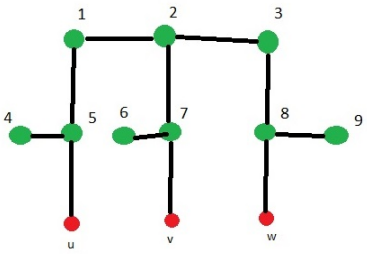
    Vamos a reducir el problema de circuito Hamiltoniano al Hamiltoniam Subgraph Partition. Se trata de una reducción trivial, ya que dado un grafo G y su conjunto de vértices v, tomando $k = 1$, se tiene V puede particionarse una partición de conjuntos disjuntos, con el tamaño de la partición igual a 1 de tal manera que cada elemento en la partición tena un circuito Hamiltoniano si y solo si G tiene un circuito Hamiltoniano.

12. Es un problema NP, ya que el problema consiste en encontrar una partición en subconjuntos de tamaño 3 que cumpla una condición y la condición es fácilmente verificable en tiempo polinómico.

    Vamos a reducir PERFECT 3-MATCHING a este problema.

    Comenzamos con un problema de PERTECT 3-MATCHING con $U \times V \times W$ y un conjunto $S \subseteq U \times V \times W$.

    Si $(u, v, w) \in S$, construimos un grafo como el de la figura:



En el que los nodos 1,2,3,4,5, 6, 7, 8 y 9 son nodos nuevos artificiales para este vector. Los nodos u, v, w son nodos que corresponde a los elementos de $U \cup V \cup W$. Estos nodos están conectados con los nodos artificiales correspondientes a todas las tripletas a las que pernencen.

La equivalencia de los dos problemas se basa en el hecho de que hay sólo dos formas de partir los nodos artificiales en conjuntos de 3, de tal manera que para cada conjunto haya dos aristas:

- Se cogen los nodos $\{1, 2, 3\}, \{4, 5, u\}, \{6, 7, v\}, \{8, 9, w\}$
- Se cogen los nodos $\{1, 4, 5\}, \{2, 6, 7\}, \{3, 8, 9\}$

El primer caso corresponde a elegir el vector (u, v, w) en el conjunto T que sea solución al PERFECT 3-D MATCHING y el segundo a no elegirlo.

Se puede comprobar que los dos problemas tienen la misma solución, sin más que transformar un subconjunto T solución de PERTECT 3-MATCHING en una selección de subconjuntos de 3 elementos de acuerdo con el criterio anterior.

Recíprocamente, dada una solución al problema de los caminos de longitud 2, cogemos como elementos de T los vectores que corresponden a la primera forma de particionar los nodos artificiales. Como cada elemento de $U \cup V \cup W$ estará en un sólo triángulo, entonces estará en un sólo vector.

13. Sea $G = (V, E)$ un grafo dirigido y $K \leq |V|$ un entero positivo. Dado un subconjunto V' de V, como los circuitos dirigos del grafo G se pueden determinar en tiempo polinómico, podemos verificar en tiempo polinómico que todo circuito dirigido de G incluye al menos un vértice de V', así como su tamaño, de donde deducimos que el Feedback Vertex Set pertenece a la clase NP.

    Vamos a reducir el Vertex Cover al Feedback Vertex Set. Como el Vertex Cover es NP-Completo deduciríamos que el Feedback Vertex Set es también NP-Completo.

    Sea un grafo G y k un entero positivo. Construimos un grafo G' con los mismos vértices y dos aristas por cada arista (u, v) del grafo G: una que va

de u hacia v y otra que va de v hacia u. E G' hay un circuito dirigido por cada arista, más los circuitos no dirigidos que había en el grafo original. Vamos a suponer un entero K para el Feedback Vertex Set igual al k del Vertex Cover.

Si en el grafo G hay una cobertura C de tamaño k entonces por cada arista de G al menos uno de los dos extremos en C. Es claro que entonces para cada circuito dirigido que hay en G' al menos un vértice en C.

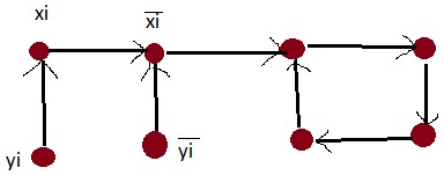
Recíprocamente, si hay un conjunto de vértices V en G' que verifican que para cada circuito dirigido en G' al menos uno de los vértices de V está en el circuito, en particular esta propiedad se verifica para los circuitos de longitud 2, es decir, para aquellos circuitos que van de un vértice a otro y luego van de ese otro vértice al original. Por consiguiente, para cada arista del grafo G al menos uno de los dos vértices incidentes está en V, o lo que es lo mismo, V forma una cobertura de G.

14. Este problema es NP, ya que pide encontrar una numeración tal que comprobar si se cumple la condición equivale a visitar todos los nodos del grafo y para cada uno de ellos las aristas que inciden en ese nodo y los números de los extremos y todo eso se puede hacer en tiempo polinómico.

    Para comprobar que es NP-completo, vamos a reducir 3-SAT a este problema.

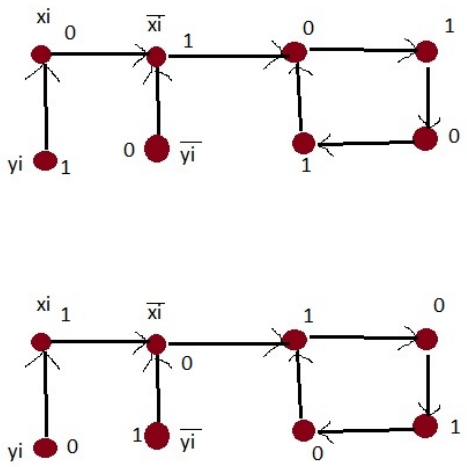
    Para ello si tenemos un problema de 3-SAT con variables $\{u_1, \ldots, u_n\}$ y cláusulas $C1, \ldots, C_m$ construimos un grafo de la siguiente forma:

    - Para cada variable u_i construimos la estructura:



      Donde la idea es que x_i y $\overline{y}_i$ tengan el mismo valor de verdad que u_i (0 si es falso y 1 si verdadero) y y_i y $\overline{x}_i$ el valor contrario (1 si es falso y 0 si verdadero).

      Hay dos formas de numerar el grafo de manera que se cumplan las condiciones con dos valores distintos que supondremos 0 y 1



El primero corresponde a u_i falso y el segundo a u_i verdadero.

A continuación para cada cláusula C_j que contiene las variables u_i, u_k, u_l se construye un triángulo dirigido con tres vértices A_j, B_j, C_j: $A_j \to B_j \to C_j \to A_j$. Finalmente, se añaden arcos desde la variable C_j a cada nodo y_i si u_i aparece en C_j en positivo y a $\overline{y}_i$ si u_i aparece en C_j en positivo.

Si las clásulas se pueden satisfacer, entonces se numeran los grafos de las variables de acuerdo con si u_i es verdadero o falso. Entonces C_j se le asigna un 2, a B_j un 0 y a A_j un 1.

Recíprocamente si hay una numeración, entonces C_j no puede conectarse a nodos y_i o $\overline{y}_i$ con valor 1, ya que entonces es imposible encontrar una numeración del triángulo. Entonces, C_j tiene que estar conectado a un y_i o $\overline{y}_i$ con valor 0, el correspondiente x_i o $\overline{x}_i$ tiene un valor 1, y el valor de verdad de la variable u_i de acuerdo con los dos casos posible hace cierta la cláusula y las cláusulas se pueden satisfacer.

15. Si tenemos un grafo $G = (V, E)$ un grafo y un entero positivo $K \leq |V|$. Dado un subconjunto V' de V podemos verificar, tanto su tamaño como que todo vértice que no pertenece a ese conjunto está conectado con al menos un vértie de V', en tiempo polinómico, por lo que el Dominating Set es Np.

    Vamos a reducir vertex cover a Dominating Set.

    Sea G un grafo y k un entero. Construimos un grafo G' como sigue: Por cada arista en el grafo G ponemos un triángulo en el grafo G', es decir, añadimos un vértice al grafo por cada arista, conectando ese nuevo vértice

con los otros dos vértices de la arista. Seleccionamos como K el mismo del Vertex Cover.

Si G tiene una cobertura de tamaño k, entonces ese mismo conjunto de vértices forma el conjunto en G' que se busca en el problema del conjunto dominante. En efecto, cada vértice v que no pertenece a la cobertura tiene al menos una arista incidente en dicho vértice (suponiendo que el grafo es conexo), llamémosle (u, v), u ha de pertenecer a la cobertura. Como u y v son adyacentes tenemos que v tiene un vértice adyacente que pertenece a la cobertura, u.

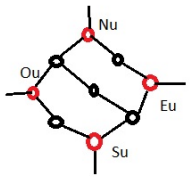
Recíprocamente, si G' tiene un conjunto dominante C (un conjunto que cumple la condición del enunciado) de tamaño k. Podemos suponer que C está contenido en V, ya que en caso contrario, esto es, en caso de que exista un vértice v' que no pertenezca a V, lo podemos sustituir por cualquiera de los otros dos vértices de V con los que forma el triángulo, sin que esto aumente la cardinalidad de C y sigue siendo un conjunto dominante. Este conjunto de vértices forma una cobertura del grafo G' : Dada una arista e, consideramos el vértice v_e que se ha añadido para formar el triángulo correspondiente a esa arista. Se tiene que v_e tiene un vértice adyacente que pertenece a C, pues dicha cobertura ha de contener a uno de los dos vértices de la arista que provenía del grafo G por la definición de cobertura. Este vértice será el vértice que pertenezca a C y que contenga a la arista. Tenemos así que C es una cobertura del grafo G'.

Como vertex cover es NP-Completo y se ha reducido a Dominating Set deducimos que este último problema también es NP-Completo

16. El problema del circuito Hamiltoniano Alternativo es NP-Completo, pues dados un grafo y dos circuitos del grafo podemos comprobar que son circuitos Hamiltonianos y que son diferentes en tiempo polinómico.

    Vamos a reducir el problema del Circuito Hamiltoniano al problema del circuito Hamiltoniano Alternativo (basado en una reducción de un problema similar del libro Papadimitriou, Steiglitz, Combinatorial Optimization. Algorithms and Complexity).

    Supongamos un grafo $G = (V, E)$ para el cual queremos saber si existe un circuito hamiltoniano. Construimos el grafo $G' = (V', E')$ para el circuito hamiltoniano alternativo de la siguiente forma:



Para cada vértice u de V construimos la siguiente estructura.

En ella sólo los nodos N_u, S_u, E_u, O_u están conectamos con otros nodos.

Sólo hay dos formas en las que un circuito hamiltoniamo puede pasar por estos grafos:

Ahora: suponiendo un orden de los nodos de $V = \{u_1, \ldots, u_n\}$, conectamos S_{u_i} con $N_{u_{i+1}}$ y S_{u_n} con N_{u_1}

Estos arcos junto con el recorrido de las estructuras de norte a sur proporcionan un circuito hamiltoniano que siempre existe.

Ahora si $(u, v) \in E$, conectamos E_u con O_v y E_v con O_u. Estos arcos son los que simulan los arcos del grafo original.

La equivalencia entre los dos problemas, se obtiene teniendo en cuenta que todo circuito hamiltoniano del grafo original es equivalente a un circuito alternativo al que recorre las estructuras de norte a sur y que recorre las estructuras de este a oeste. La equivalencia se obtiene teniendo en cuenta que si (u, v) es un enlace de este circuito hamiltoniano del primer grafo, entonces en el segundo vamos de O_u a E_v y entonces recorremos la estructura de v entrando por E_v y saliendo por E_u. Además si existe un camino hamiltoniano alternativo en el segundo grafo al que recorre las estructuras de norte a sur, tiene que recorrer las estructuras de oeste a este (o de este a oeste si consideramos el circuito inverso). En ese caso irá de una estructura a otra por enlaces que corresponden a enlaces en el grafo original y estos enlaces compondrán un circuito hamiltoniamo en el grafo original.

17. Sea A un conjunto y $s(a) > 0$ una función de tamaño $\forall a \in A$. Sea J un entero positivo. Dada una familia de subconjuntos de A podemos verificar el número de subconjuntos de la familia, si son disjuntos y si se cumple la desigualdad que nos da el problema, todo en tiempo polinómico, de donde se sigue que el problema de la suma mínima de cuadrados pertenece a la clase NP.

    Vamos a reducir el problema de la partición al problema de la suma mínima de cuadrados.

    Dado un conjunto de objetos A' que definen el conjunto de la partición. Si $a \in A$‘, sea $s(a)$ su tamaño. Consideramos $s = \sum_{a \in A'} s(a)$ tomamos $A = A'$, $J = s^2/2$ y $K = 2$. I

Si existe una partición de A' en 2 conjuntos B y C cuyos elementos sumen lo mismo, esos dos conjuntos van a verificar:

$\sum_{a\in B} s(a)^2 + \sum_{a\in C} s(a)^2 = 2 * \sum_{a\in B} s(a)^2 = 2 * s^2/4 = s^2/2$

Recóprocamente, si no existe una partición de A en dos conjuntos B y C que no verifiquen que $\sum_{a\in C} s(a) = \sum_{a\in B} s(a)$. Si tenemos una partición cualquiera de A en dos conjuntos B y C, llamamos $a = \sum_{a\in C} s(a)$ y $b = \sum_{a\in B} s(a)$. Como no suman lo mismo, podemos suponer que existe $\epsilon > 0$ tal que $a = s/2 + \epsilon$. En este caso $b = s/2 - \epsilon$, pues $a + b = s$. Entonces se verifica que $s/2 + \epsilon^2 + s/2 - \epsilon^2 = s^2/2 + 2 * \epsilon^2 > s^2/2$. Por consiguiente, hemos demostrado que si existe una partición de A en dos subconjuntos que verifiquen que $\sum_{x\in C} x^2 + \sum_{x\in B} x^2 \leq s^2/s$ entonces la suma de los elementos de B han de sumar $s/2$, al igual que los elementos de C

Como sabemos que el problema de la partición es NP-Completo y lo hemos reducido al de la suma mínima de cuadrados, podemos afirmar que el problema de la suma mínima de cuadrados es NP-Completo.

18. Sea A un alfabeto y E_1 y E_2 dos expresiones regulares sobre A que no contienen el operador $*$. Dada una palabra sobre el alfabeto, podemos comprobar en tiempo polinómico si pertenece al lenguaje generado por la expresión regular E_1 y no al lenguaje generado por la expresión regular E_2, (o viceversa), lo que implicaría que los lenguajes generados por ambas expresiones son distintos. Por consiguiente, el problema dado es NP.

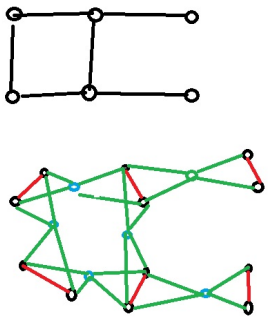
    Sabemos que el 3-Sat es NP-Completo. Vamos a reducir el 3-SAT al Star-Free Regular Expresion Equivalence para concluir que este último problema es también NP-Completo.

    Sea ϕ un conjunto de cláusulas $C_1, \ldots, C_m$ sobre las variables $x_1, \ldots, x_n$. Formamos la expresión regular r del siguiente modo: $r = (r_1 + r_2 + \ldots + r_m)$, donde $r_i = \alpha_{i_1}, \ldots, \alpha_{i_n}$, con $\alpha_{i_j} = 0$ si $x_j \in C_i$, $\alpha_{i_j} = 1$ si $x_j^c \in C_i$ y $\alpha_{i_j} = (0 + 1)$ en otro caso. Que el conjunto de cláusulas sea satisfacible es lo mismo que decir que el lenguaje aceptado por la expresión regular r sea distinto al lenguaje aceptado por la expresión regular $(0 + 1)^n$ ya que por cada cláusula C_i hemos creado la expresión r_i que corresponde a las palabras de $(0+1)^n$ que corresponden a asignaciones de valores de verdad a las variables (el símbolo número j de una palabra $u \in (0+1)^n$ es el valor de verdad de la variable u_j) que hacen que la cláusula C_j no se satisfaga. Entonces r corresponde a las palabras de $(0 + 1)^n$ que corresponden a asignaciones de valores de verdad a las variables que hacen que no se satisfaga alguna cláusula. Si esta expresión regular no representa $(0 + 1)^n$ entones hay una asignación de valores de verdad que satisface todas las cláusulas.

19. El problema de la red de telefonía móvil pertenece a la clase NP, ya que, dado un conjunto de aristas de un grafo $G = (V, E)$, podemos verificar en tiempo polinómico que esas aristas no tienen ningún vecino en común.

Sabemos de la relación anterior que el Independent Set es equivalente al Clique y también sabemos que este último es NP-Completo. Así pues, vamos a reducir el Independent Set al problema que nos dan para concluir que el problema dado es NP-Completo.

Dado un grafo G construimos el siguiente grafo G': Por cada nodo de G añadimos dos nodos al grafo G' y los conectamos mediante una arista. Ahora, por cada arista del grafo original añadimos al grafo G' un nodo entre los 4 que correspondían a los 2 del grafo G que estaban enlazados. Este nodo añadido lo enlazamos con cada uno de los 4 que habíamos añadido antes. Aquí podemos ver gráficamente la idea:



Si en el grafo G encontramos un conjunto independiente de tamaño k, entonces formamos el siguiente conjunto de aristas: Por cada nodo de G que esté en el conjunto independiente insertamos en nuestro conjunto de aristas la arista que enlaza a los dos nodos que hemos añadido a G' correspondientes al nodo que está en el conjunto independiente. Se puede ver que este conjunto de aristas así construido no tiene aristas con vecinos en común. De esta manera, si tenemos un conjunto independiente en el grafo G tenemos un conjunto de aristas en G' tales que no hay dos de ellas que tengan vecinos en común.

Recíprocamente, supongamos que en el grafo G' encontramos un conjunto de aristas de tamaño k que no tienen vecinos en común. Si una de esas aristas tiene como nodo adyacente a un nodo intermedio, la podemos cambiar por la que une el otro vértice adyacente con el nodo que junto con ese otro vértice adyacente corresponden a un nodo del grafo original, que será otra solución del problema dado, porque los vecinos de la segunda arista están incluidos en los vecinos de la primera. Por consiguiente, si consideramos los nodos de G correspondientes a cada una de las aristas que están en nuestro conjunto de aristas obtenemos un conjunto independiente de vértices de tamaño k del grafo G.

20. El problema es NP ya que se pide un nivel de energía que cumpla una

condición y el cumplimiento de esa condición supone unas operaciones que se pueden comprobar en tiempo polinómico.

Para demostrar que es NP-Completo, vamos a reducir el MAX CUT a este problema.

Dado un problema de MAX-CUT ponderado (Ejercicio 5) con un grafo $G = (V, A)$ y donde cada arista $(v_i, v_j) \in A$ tiene un peso w_{ij} construimos un ejemplo de este problema en el que hay un vértice por cada elemento de V y en el que si no hay arista de v_i a v_j consideramos que $J_{ij} = 0$ y si $(v_i, v_j) \in A$, hacemos $J_{ij} = -w_{ij}$. Además $h_i = 0$.

Un corte en el primer problema V_1 V_2 corresponde a un estado en los que los elementos de V_1 tienen un spin $+1$ y los de V_2 un spin -1.

En los vértices del mismo lado de la partición se suma $-J_{ij}$ y en los de distinto se suma J_{ij}. Así, si el valor del corte es K, entonces el valor de energía es

$-\sum_{ij} J_{ij} - 2K$.

Como el valor $-\sum_{ij} J_{ij} = sum_{ij} w_{ij}$ no depende de la partición y se puede calcular en tiempo polinómico. Si en el problema de MAX-CUT tenemos un límite de K, en este tomaremos como límite $E = sum_{ij} w_{ij} - 2K$.

Con estas condiciones es inmediato que los dos problemas son equivalentes.

21. Dada una manera de distribuir los n contenedores en los m camiones se puede comprobar en tiempo polinómico si ningún contenedor está sobrecargado y que en ningún camión hay pares de sustancias que no pueden ir juntas, por lo que el problema dado es NP.

    Vamos a reducir el Matching 3-D al problema dado.

    Sea S un subconjunto de $UxVxW$, donde U, V y W son conjuntos finitos de tamaño n.

    Denotamos $m = |S|$, que será el número de camiones disponibles en la instancia del problema dado que queremos construir, $l = m - n$. Cada camión podrá transportar a lo más 4 contenedores. Denotamos como a_i a la coordenada de U del i-ésimo elemento de S, como b_i a la coordenada de V del i-ésimo elemento, y como c_i a la coordenada de W del i-ésimo elemento, que serán contenedores de nuestro problema. Por cada elemento del conjunto S consideramos el contenedor x_i. Suponemos que los contenedores x_i , x_j han de ir en distintos contenedores $\forall i \neq j$, por lo que cada x_i va a un camión diferente. Añadimos también los contenedores $y_1, \ldots, y_l$ de tal manera que dos de ellos tampoco pueden ir en el mismo camión. Los x_i y los y_j pueden ir en el mismo camión $\forall i, j$. Si $x_i = (a_i, b_i, c_i)$, entonces x_i es compatible con a_i, b_i y c_i. Esta condición de compatibilidad la hacemos para cada x_i. Los demás contenedores son incompatibles.

    De este modo, se pueden encontrar un subconjunto T de S de tamaño n que contenga a cada elemento de $U \cup V \cup W$ una y sólo una vez si

y solo si se pueden transportar todos los contenedores citados en los m contenedores.

Supongamos $S' \subseteq S$ una solución al matching 3-D. Entonces, se coloca cada x_i en un camión distinto, los $y_1, \ldots, y_l$ van en los camiones en los que van los elementos $S \setminus S'$. Los elementos de U, V, W van acompañando a los x_i con $x_i \in S'$. Cada $x_i = (a_i, b_i, c_i)$ con $x_i \in S'$ lleva a los elementos a_i, b_i, c_i. Con esto todos los contenedores se cargan en camiones con un límite de carga 4.

Recíprocamente, si hay una carga de los camiones, cada x_i tiene que ir en un camión distinto. Cada y_j irá en un camión acompañando a un x_i. El matching 3-D viene dado por los x_i que no llevan ningún y_j y que tendrán que llevar a sus elementos. Como cada contenedor a_i, b_i, c_i va en uno y un sólo camión. Entonces, cada elemento estará en uno y sólo uno de los vectores seleccionados.

Como el problema Matching 3-D 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 estaría 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. Como el flujo máximo se puede calcular en tiempo polinómico, entonces el número máximo de camiones también.