Más problemas NP-completos

Horst H. von Brand
vonbrand@inf.utfsm.cl

Departamento de Informática
Universidad Técnica Federico Santa María

Contenido

El problema Set Cover

El problema Set Cover da un conjunto universo $U = \{x_1, x_2, \ldots, x_n\}$, un conjunto de subconjuntos $S_i \subseteq U$, y un entero k. La pregunta es si hay una colección C de k subconjuntos tal que «cubren» U, o sea $|C| = k$ y:

$$\bigcup_{i \in C} S_i = U$$

Teorema

Set Cover *es* NP*-completo.*

Set Cover

Demostración.

Un certificado es una colección de k conjuntos que cubren U.
Reducimos Vertex Cover $\leq_p$ Set Cover.
Sea $G = (V, E)$, tomamos $U = E$ y los conjuntos S_v como los arcos que inciden en v. Un *vertex cover* de G es un *set cover* de U. $\square$

El problema Feedback Vertex Set

Sea $G = (V, E)$ un grafo dirigido. Un *feedback vertex set* de G es un conjunto de vértices tal que si se eliminan de G el grafo resultante no tiene ciclos dirigidos. El problema Feedback Vertex Set (abreviado FVS) da un digrafo G y un natural k, y pregunta si G tiene un *feedback vertex set* de tamaño k.

Teorema

Feedback Vertex Set *es* NP-*completo.*

Feedback Vertex Set

Demostración.

Un certificado es un *feedback vertex set* de tamaño k.
Reducimos Vertex Cover $\leq_p$ FVS.
Dados el grafo $G = (V, E)$ y k construimos el digrafo $G' = (V, E')$ con:

$$E' = \{(u, v) : \{u, v\} \in E\}$$

Un *vertex cover* de G son vértices que están en todos los arcos de G, eliminarlos rompe todos los ciclos dirigidos en G'. Romper todos los ciclos dirigidos de G' puede hacerse solo incluyendo uno de u, v para cada $\{u, v\} \in E$. $\square$

El problema Dominating Set

Dado un grafo $G = (V, E)$, un *dominating set* $S \subseteq V$ es tal que todo vértice en $V \smallsetminus S$ tiene a un vértice en S de vecino. El problema Dominating Set (abreviado DS) pregunta si el grafo G tiene un *dominating set* de tamaño k.

Teorema

Dominating Set *es* NP-*completo.*

Dominating Set

Demostración.

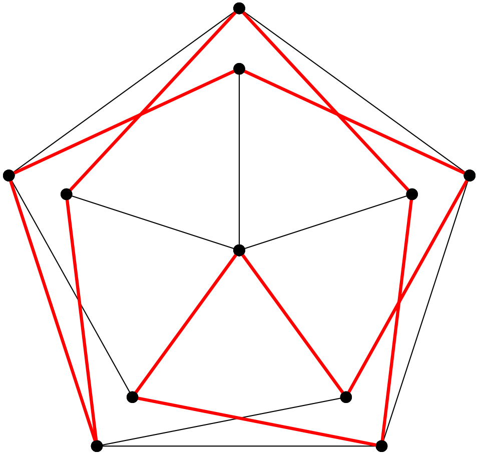
Un certificado es un *dominating set* de tamaño k.
Reducimos Vertex Cover $\leq_p$ DS.
Dado $G = (V, E)$ construimos el grafo G' con vértices $V \cup E$. Para cada arco $e = uv \in E$ creamos un triángulo, con arcos uv, ue y ve. Un *vertex cover* de G contiene un vértice de cada arco de G, con lo que esos vértices son vecinos de todos los vértices de G'. Al revés, si un *dominating set* de G' incluye el arco $e = uv$ de G, podemos reemplazarlo por u o por v (solo está conectado a esos vértices en G'), y el conjunto resultante es un *vertex cover* de G. $\square$

Caminos y ciclos hamiltonianos

Dado un grafo $G = (V, E)$, un *camino hamiltoniano* visita cada vértice exactamente una vez. Un *ciclo hamiltoniano* visitas cada vértice exactamente una vez. Básicamente las mismas definiciones se aplican a digrafos (con caminos y ciclos dirigidos). Las preguntas si un grafo o digrafo tienen un camino o ciclo hamiltoniano son todos problemas NP-completos.

El grafo de Grötzsch

La figura muestra un ciclo hamiltoniano del grafo de Grötzsch, de número cromático 4 sin tener triángulos.



El problema D-Hamiltonian

El problema D-Hamiltonian (abreviado DH) pregunta si el digrafo G tiene un ciclo hamiltoniano dirigido.

Teorema

D-Hamiltonian *es* NP-*completo.*

D-Hamiltonian

Demostración.

Un certificado es un ciclo hamiltoniano dirigido.

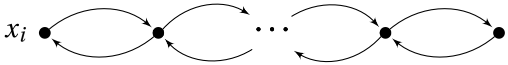
Reducimos 3SAT $\leq_p$ D-Hamiltonian.

D-Hamiltonian

Para cada variable x_i definimos una cadena de vértices conectados hacia la izquierda y derecha. La única forma de visitar cada vértice una sola vez es atravesar la cadena de izquierda a derecha (lo interpretamos como x_i toma el valor verdadero) o de derecha a izquierda (interpretado como x_i falso).

D-Hamiltonian

Gráficamente, es:



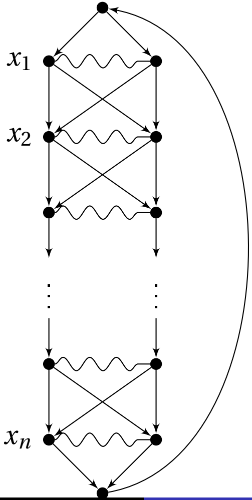
Dirección si x_i verdadero $\rightarrow$
Dirección si x_i falso $\leftarrow$

D-Hamiltonian

Conectamos ambos extremos de la cadena para x_i con arcos dirigidos a ambos extremos de la cadena para x_{i+1}, de forma que estamos obligados a atravesar las cadenas en orden.
Agregamos un vértice inicial, conectado hacia los extremos de la cadena para x_1, y otro vértice final conectado desde los extremos de la cadena x_n. Completamos el grafo con un arco del vértice final al inicial.

D-Hamiltonian

La conexión entre cadenas queda como:

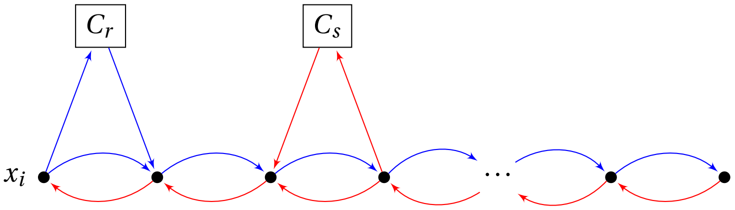


D-Hamiltonian

Falta representar cláusulas. Creamos un vértice para cada cláusula. Si aparece el literal x_i en la cláusula c, creamos dos vértices consecutivos en la cadena para x_i y los conectamos de derecha a izquierda con el vértice para c (con esto, si x_i es verdadera, tenemos la opción de dar un desvío a través de c al marchar de derecha a izquierda por la cadena). Si aparece $\overline{x}_i$, las conexiones son de izquierda a derecha.

D-Hamiltonian

La conexión del literal x_i a la cláusula C_r y de $\overline{x}_i$ a C_s es como muestra el diagrama:



D-Hamiltonian

Este digrafo tiene un ciclo hamiltoniano dirigido si y solo si ϕ puede satisfacerse. $\square$

Si se revisa la demostración, se ve que la construcción en realidad sirve para CSAT.

El problema D-Hamiltonian Path

El problema D-Hamiltonian Path pregunta si el digrafo G tiene un camino hamiltoniano dirigido.

Corolario

D-Hamiltonian Path *es* NP*-completo.*

Demostración.

Basta eliminar el arco del vértice final al inicial en la demostración precedente para crear un digrafo que tiene un camino hamiltoniano dirigido si y solo si ϕ se puede satisfacer. $\square$

El problema Hamiltonian

El problema Hamiltonian es similar a D-Hamiltonian, pero para un grafo.

Teorema

Hamiltonian *es* NP-*completo.*

Hamiltonian

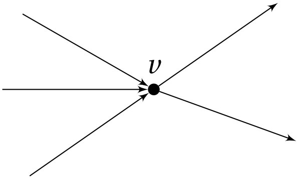
Demostración.

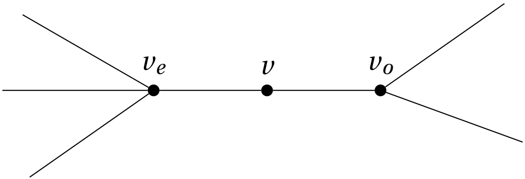
Certificado es un ciclo hamiltoniano.

Reducimos D-Hamiltonian $\leq_p$ Hamiltonian. Para ello debemos forzar un orden de recorrido, lo que logramos dividiendo el vértice v en tres: v_i de entrada, los arcos hacia v se dirigen acá; v_o de salida, los arcos desde v salen de acá. Conectamos $v_i v$ y $v v_o$. Un camino que pase por los vértices deberá visitarlos todos siempre en el orden $v_e - v - v_o$ (o al revés), fijando la dirección de los arcos. $\square$

Hamiltonian

El vértice v se divide en v_i, v, v_o:





El problema del vendedor viajero

El problema del vendedor viajero (en inglés *Traveling Salesman Problem*, sexistamente correcto *Traveling Salesperson Problem*, abreviado TSP) considera un conjunto de clientes a visitar, dados costos de viaje entre los clientes i y j dados por $d(i,j)$ (note que puede ser $d(i,j)=\infty$, no hay manera de ir directamente de i a j; y nada impide que $d(i,j) \neq d(j,i)$, por ejemplo uno es en bajada, el otro en subida), volviendo a su origen. Obviamente interesa el viaje de costo total mínimo.

Trataremos con el problema de decisión, que llamaremos D-TSP, ¿hay un circuito de costo menor o igual a L que visita todos los vértices en el grafo dirigido?

D-TSP

Teorema

D-TSP *es* NP*-completo.*

Demostración.

Un certificado que demuestra que D-TSP está en NP es un circuito que cumple lo pedido.
Para demostrar que es NP-duro, reducimos D-Hamiltonian $\le_p$ D-TSP. Dada una instancia $G = (V, E)$ de D-Hamiltonian, creamos un grafo dirigido completo. Al arco $uv \in E$ le asignamos costo $d(u, v) = 1$, al arco $uv \neq E$ le asignamos $d(v, u) = 2$. Es claro que si y solo si hay un ciclo hamiltoniano en G, hay un circuito para el vendedor de costo $|V|$. □

El problema del vendedor viajero

Es un problema de inmensa importancia práctica, intensamente estudiado. Hay múltiples variantes, como TSP con costos simétricos; en TSP euclidiano los vértices son puntos en el plano y los costos son las distancias entre ellos; la variante TSP_Δ es TSP en que los costos cumplen la desigualdad triangular $d(r,t) \le d(r,s) + d(s,t)$.

Para algunos es el símbolo de los problemas NP-completos.

El problema de la mochila (Knapsack)

El problema Knapsack (mochila) presenta un conjunto de objetos de pesos positivos w_i con valores positivos v_i, una capacidad de la mochila b, y un valor c. ¿Se puede llenar la mochila (sin sobrepasar su capacidad), logrando al menos el valor c? Suponemos que los valores son todos enteros.

Teorema

Knapsack *es* NP-*completo.*

Knapsack

Demostración.

Un certificado es un conjunto de objetos de peso a lo más b cuyos valores suman al menos c.
Para demostrar que es NP-duro, reducimos
Subset Sum $\le_p$ Knapsack. Para la instancia de Subset Sum dada por $\mathscr{A} = \{a_1, \ldots, a_n\}$ y S, creamos la instancia de Knapsack con $w_i = v_i = a_i$, $b = c = S$. □

El problema de partición (Partition)

El problema Partition pone un multiconjunto de enteros que suman $2B$, y pregunta si puede dividirse en dos conjuntos que suman B.

Teorema

Partition *es* NP-*completo.*

Partition

Demostración.

Un certificado es un multisubconjunto que suma B.
Para demostrar que es NP-duro, reducimos Subset Sum $\leq_p$ Partition. Para la instancia de Subset Sum dada por $\mathscr{A} = \{a_1, \dots, a_n\}$ y suma S, creamos la instancia de Partition como sigue: sea m la suma de los elementos de $\mathscr{A}$, agregamos nuevos elementos $a_{n+1} = 2m + S, a_{n+2} = 3m - S$, dando el multiconjunto $\mathscr{B} = \mathscr{A} \cup \{2m + S, 3m - S\}$. La suma total de los elementos de $\mathscr{B}$ es $6m$, a_{n+1} y a_{n+2} no pueden pertenecer a la misma partición de $\mathscr{B}$. Si podemos elegir elementos de $\mathscr{A}$ que sumen S, sumando a_{n+2} con estos da $S + 3m - S = 3m$, y hemos construido una partición. Al revés, sabemos que cualquier partición de $\mathscr{B}$ tiene a_{n+1} y a_{n+2} en particiones distintas, los demás elementos de la partición que contiene $a_{n+2} = 3m - S$ suman S. $\square$

El problema Job Shop Scheduling

Muchos problemas de asignación de recursos son NP-completos. Una variante muy simple es Job Shop Scheduling.
El problema Job Shop Scheduling da un conjunto de tareas, la tarea i es de duración t_i, y un número m de máquinas idénticas. Las tareas se ejecutan una tras otra en una máquina sin interrupción, y se busca determinar si hay una distribución de tareas a máquinas de manera de cumplir un plazo fatal d.

Teorema

Job Shop Scheduling *es* NP-*completo.*

Job Shop Scheduling

Demostración.

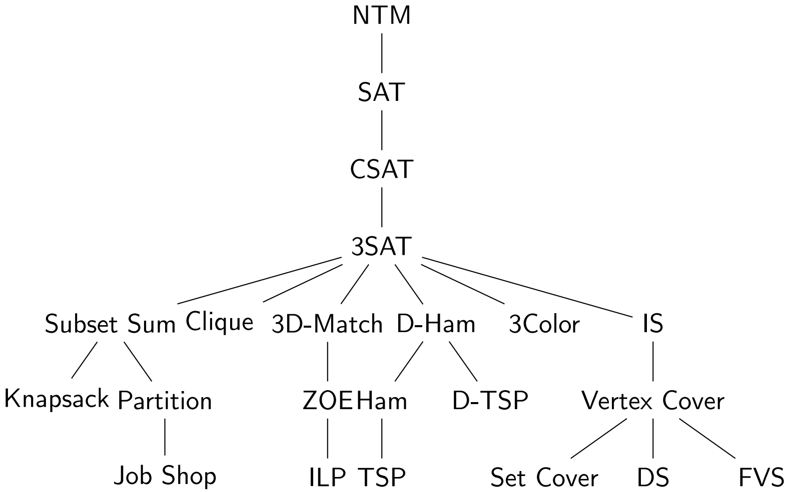
Un certificado es una asignación de tareas a las máquinas que cumple el plazo fatal.
Reducimos Partition $\leq_p$ Job Shop Scheduling. Dada una instancia de Partition, damos como tiempos de ejecución los elementos y $m = 2$, especificando plazo fatal B si la suma de los elementos es $2B$. □

¿Por qué hacemos estas reducciones?

Lo que logramos es pasar de demostraciones con máquinas de Turing a otras más concretas. Muchos problemas prácticos se pueden asimilar a estos ejemplos, problemas reales generalmente serán más complejos. Note eso sí que si hay restricciones adicionales estas pueden simplificar el problema, con lo que puede dejar de ser NP-completo.

Las reducciones *no* son formas prácticas de resolver los problemas, se buscan algoritmos especializados para resolverlos (u obtener soluciones aproximadas).

Reducciones discutidas



Resumen

- Ampliamos la lista de problemas NP-completos, ilustrando algunas nuevas técnicas de demostración en el proceso.
- Vemos que hay problemas en una diversidad de áreas que son NP-completos. Muchos problemas de asignación de recursos son NP-completos. En casos concretos, vea si hay restricciones adicionales que simplifican el problema.