Ensemble Computation

Rodrigo Alejandro Sánchez Morales

Universidad Nacional Autónoma de México
Facultad de Ciencias

Complejidad Computacional, 2019-I

Octubre de 2018

Tabla de contenidos

¿Qué es una reducción?

Definición:

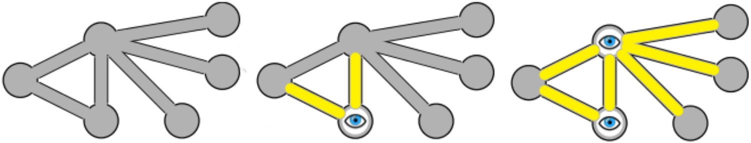
Una reducción es una transformación de un problema a otro problema.

Técnicas para probar la NP-Completez

- Restriction.
- Component Design.
- **Local Replacement:** Todo lo que hacemos es elegir algún aspecto del ejemplar del problema NP-Completo conocido para hacer una colección de unidades básicas, y obtenemos el ejemplar correspondiente al problema objetivo reemplazando cada unidad básica de manera uniforme con una estructura diferente.

¿Qué es el Vertex Cover?

Es un conjunto de vértices tales que cada arista del grafo es incidente a al menos un vértice del conjunto.



¿Cómo probar NP-Completez?

Requisitos

Para probar que un problema es NP-Completo debemos probar que está en NP, buscar un problema NP-Completo y luego reducirlo a nuestro problema en tiempo polinomial.

Representación del problema

Ejemplar:

Una colección C de subconjuntos de un conjunto finito A y un entero positivo J.

Pregunta:

¿Hay una secuencia

$$< z_1 = x_1 \cup y_1, z_2 = x_2 \cup y_2, \ldots, z_j = x_j \cup y_j >$$

de $j \leq J$ operaciones de unión, donde cada x_i y y_i es $\{a\}$ para algún $a \in A$ o z_k para algún $k < i$, tal que x_i y y_i son disjuntos a $1 \leq i \leq j$ y que para cad tu subconjunto $c \in C$ hay algún z_j, $1 \leq i \leq j$, que es idéntico a c?

Ensemble Computation es NP-Completo

Ensemble Computation $\in$ NP

Dado un ejemplar particular del problema, es posible especificar un algoritmo que en la primera fase *adivinadora* asigne un elemento de la secuencia z_i y en la segunda fase *verificadora* compruebe si la asignación resuelve el problema.

$\therefore$ *Ensemble Computation* $\in NP$

Idea principal:

Transformamos *Vertex Cover* a *Ensemble Computation*.

$$\textit{Vertex Cover} \propto_p \textit{Ensemble Computation}$$

Sea la gráfica $G = (V, E)$ y un entero positivo $K \leq |V|$ que constituye un ejemplar arbitrario de *Vertex Cover*.

La unidad básica del ejemplar de *Vertex Cover* son las aristas de G.

Sea a_0 un nuevo elemento no en V.

El **reemplazo local** sólo sustituye para cada arista $\{u, v\} \in E$, el subconjunto $\{a_0, u, v\} \in C$.

El ejemplar de *Ensemble Computation* está completamente especificado por:

$$A = V \cup a_0$$
$$C = \{\{a_0, u, v\} : \{u, v\} \in E\}$$
$$J = K + |E|$$

Es fácil ver que este ejemplar puede ser construido en tiempo polinomial.

Afirmamos que G tiene una cubierta de vértices de tamaño K o menos si y sólo si la secuencia deseada de $j \leq J$ operaciones existe para C.

$$G \text{ tiene cubierta de vértices, } |K|$$
$$\Leftrightarrow$$
$$\exists \text{ secuencia } j \leq J \text{ operaciones } \in C$$

$\Rightarrow$

Primero, supongamos que V' es una cubierta de vértices para G de tamaño K o menor.

Ya que podemos agregar vértices adicionales a V' y seguirá siendo una cubierta de vértices, no hay pérdida de generalidad en asumir que $|V'| = K$.

Etiquetamos los elementos de V' como $v_1, v_2, \ldots, v_k$ y etiquetamos las aristas en E como $e_1, e_2, \ldots, e_m$ donde $m = |E|$.

Ya que V' es una cubierta de vértices, cada e_j contiene al menos un elemento de V'.

Entonces podemos escribir cada e_j como $e_j = \{u_j, v_{r[j]}\}$, donde $r[j]$ es un entero que satisface $1 \leq r[j] \leq K$.

La siguiente secuencia de $K + |E| = J$ operaciones es facilmente ver que tiene todas las propiedades requeridas.

$$< z_1 = \{a_0\} \cup \{v_1\}, z_2 = \{a_0\} \cup \{v_2\}, \ldots, z_j = \{a_0\} \cup \{v_k\},$$
$$z_{K+1} = \{u_1\} \cup z_{r[1]}, z_{K+2} = \{u_2\} \cup z_{r[2]}, \ldots, z_J = \{u_m\} \cup z_{r[m]} >$$

$\Leftarrow$

Por el contrario, supongamos $S =< z_1 = x_1 \cup y_1, \ldots, z_j = x_j \cup y_j >$ es la secuencia deseada de $j \leq J$ operaciones para el ejemplar de *Ensemble Computation*.

Además asumimos que S es la secuencia más corta para este ejemplar y que, entre todas esas secuencias mínimas, S contiene el menor número posible de operaciones de la forma $z_i = \{u\} \cup \{v\}$ para $u, v \in V$.

Nuestra primera afirmación es que S no puede contener operaciones de esta última forma. Supongamos que $z_i = \{u\} \cup \{v\}$ con $u, v \in V$ es incluido.

Ya que $\{u, v\}$ no está en C y ya que S tiene longitud mínima, debemos tener $\{u, v\} \in E$ y $\{a_0, u, v\} = \{a_0\} \cup z_j$ (o $z_j \cup \{a_0\}$) debe ocurrir más tarde en S.

Sin embargo ya que $\{u, v\}$ es un conjunto de solamente un miembro de C, z_j no puede ser usada en cualquier otra operación en esta secuencia de longitud mínima.

Permite que podamos reemplazar dos operaciones.

$$z_i = \{u\} \cup \{v\} \text{ y } \{a_0, u, v\} = \{a_0\} \cup z_j$$

por

$$z_i = \{a_0\} \cup \{u\} \text{ y } \{a_0, u, v\} = \{v\} \cup z_j$$

Reduciendo así el número de operaciones prohibidas sin alargar la secuencia global, lo que contradice la elección de S.

Por lo tanto, S consiste solamente de operaciones teniendo una de las dos formas $z_j = \{a_0\} \cup \{u\}$ para $u \in V$ o $\{a_0, u, v\} = \{v\} \cup z_j$ para $\{u, v\} \in E$ (donde ignoramos el orden relativo de los dos operandos en cada caso).

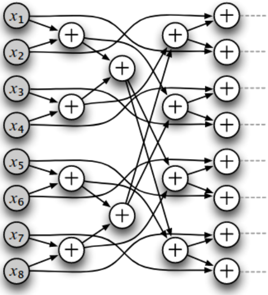
Porque $|C| = |E|$ y porque cada miembro de C contiene tres elementos, S debe contener exactamente $|E|$ operaciones de esta última forma y exactamente $j - |E| \leq J - |E| = K$ de los primeros.

Entonces el conjunto

$$V' = \{u \in V : z_j = \{a_0\} \cup \{u\} \text{ es una operación en } S\}$$

Contiene al menos K vértices de V y, como puede ser verificada fácilmente de la construcción de C, debe ser una cubierta de vértices para G. □

Buscando circuitos eficientes



$x_2 + x_3 + x_4 + x_5 + x_6 + x_7 + x_8$

$x_1 + x_3 + x_4 + x_5 + x_6 + x_7 + x_8$

$x_1 + x_2 + x_4 + x_5 + x_6 + x_7 + x_8$

$x_1 + x_2 + x_3 + x_5 + x_6 + x_7 + x_8$

$x_1 + x_2 + x_3 + x_4 + x_6 + x_7 + x_8$

$x_1 + x_2 + x_3 + x_4 + x_5 + x_7 + x_8$

$x_1 + x_2 + x_3 + x_4 + x_5 + x_6 + x_8$

$x_1 + x_2 + x_3 + x_4 + x_5 + x_6 + x_7$

Figura: Un ejemplar de Ensemble Computing (derecha) y un circuito que lo resuelve (izquierda).

Referencias

Garey & Johnson,
Computers and Intractactability, A Guide to the Theory of NP-Completeness.
(1979). Estados Unidos de América: Freeman.
https://sites.google.com/a/ciencias.unam.mx/cco-2019-1/recursos/%5BGarey-Johnson%5D%20Computers%20and%20Intractability.djvu?attredirects=0&d=1

Gracias por su atención.