



UNIVERSIDAD DE GRANADA

E.T.S. de Ingenierías Informática y de Telecomunicación

Departamento de Ciencias de la Computación e Inteligencia Artificial

Modelos Avanzados de Computación

Demostración de NP-Complejidad del Problema de la Contratación Eficiente

Autor:

Sánchez Varbas, José Manuel josemsv@correo.ugr.es

Curso 2025-2026

ETSIIT

Escuela Técnica Superior
de Ingenierías Informática
y de Telecomunicación



Índice

1. Introducción y Definición del Problema	3
2. Demostración	3
2.1. El problema está en NP	3
2.2. El problema es NP-completo	8
3. Bibliografía	13

1. Introducción y Definición del Problema

Estudiamos el problema 10ñ, es decir, el **Problema de la Contratación Eficiente**. Este es un problema de decisión, por lo que su respuesta es siempre *Sí* o *No*.

Definición 1.1 (Problema de la Contratación Eficiente (CE)). Tenemos un conjunto de n deportes $D = \{d_1, \dots, d_n\}$ y un conjunto de m especialistas $E = \{e_1, \dots, e_m\}$, tal que cada especialista e_j conoce un subconjunto de deportes $D_j \subseteq D$. Si tenemos un límite de contratos $K \leq m$, determinar si existe un subconjunto $E' \subseteq E$ de tamaño menor o igual a K y tal que para cada deporte $d_i \in D$ existe algún especialista $e_j \in E'$ con $d_i \in D_j$.

Es decir, queremos decidir si es posible contratar como mucho K especialistas de forma que, entre todos los especialistas contratados, se conozcan todos los deportes.

Para demostrar que este problema es NP-completo seguiremos el esquema visto en teoría. Por tanto, debemos probar:

1. $CE \in \mathbf{NP}$.
2. Existe un problema NP-completo conocido P' tal que

$$P' \propto CE.$$

En nuestro caso para el segundo punto haremos la reducción desde **Cubrimiento por Vértices**, que aparece en la lista proporcionada de problemas NP-completos básicos para reducciones. Recordaremos su definición en el momento de probar la completitud.

2. Demostración

2.1. El problema está en NP

Veamos que $CE \in \mathbf{NP}$ usando la definición alternativa verificador-certificado vista en teoría.

Fijamos un alfabeto de codificación $A = \{0, 1\}$. En teoría se ha visto que toda instancia de un problema computacional se representa mediante una palabra de A^* . Para evitar ambigüedades, podemos usar la siguiente codificación concreta para las instancias de CE .

Una instancia de Contratación Eficiente viene dada por $I = (D, E, \{D_j\}_{j=1}^m, K)$, donde $D = \{d_1, \dots, d_n\}$ es el conjunto de deportes, $E = \{e_1, \dots, e_m\}$ es el conjunto de especialistas, cada $D_j \subseteq D$ es el conjunto de deportes que conoce el especialista e_j , y $K \leq m$ es el número máximo de especialistas que se pueden contratar.

Como los nombres concretos de los deportes y de los especialistas no son relevantes, sino únicamente qué especialista conoce qué deporte, una instancia queda completamente descrita por los números n, m, K y por la matriz binaria

$$M_I = (a_{ij})_{1 \leq i \leq n, 1 \leq j \leq m},$$

donde

$$a_{ij} = \begin{cases} 1, & \text{si } d_i \in D_j, \\ 0, & \text{si } d_i \notin D_j. \end{cases}$$

Por tanto, podemos codificar la instancia I mediante una palabra

$$\langle I \rangle \in A^*$$

que contiene las codificaciones binarias de n , m y K , junto con los nm bits de la matriz M_I , usando una codificación binaria autodelimitada para cada bloque. Por ejemplo, podemos codificar cada palabra binaria w como $1^{|w|}0w$. De esta forma se puede recuperar inequívocamente dónde termina cada bloque y dónde empieza el siguiente, usando solo símbolos de $A = \{0, 1\}$. Esta es una codificación relativamente sencilla y, lo más importante, tiene tamaño polinómico en la descripción de la instancia.

En lo que sigue, si $x = \langle I \rangle$, diremos (abusando de la notación) que

$$x = (D, E, \{D_j\}_{j=1}^m, K),$$

entendiendo siempre que x es la palabra de A^* que codifica dicha instancia. Si una palabra $x \in A^*$ no codifica correctamente una instancia de CE , definiremos $R(x, y) = 0$ para todo $y \in A^*$ más adelante.

Teniendo lo anterior en cuenta, según la caracterización de **NP** por verificador y certificado, debemos encontrar una relación binaria

$$R : A^* \times A^* \longrightarrow \{0, 1\}$$

calculable en tiempo polinómico, y un polinomio p , tales que, para toda palabra $x \in A^*$,

$$x \in CE \iff \exists y \in A^* \text{ con } |y| \leq p(|x|) \text{ y } R(x, y) = 1.$$

Tomamos como certificado una palabra $y \in A^*$ que codifica un vector binario de longitud m :

$$(b_1, \dots, b_m) \in \{0, 1\}^m.$$

Como $A = \{0, 1\}$, podemos codificar este vector simplemente mediante la palabra

$$y = b_1 b_2 \cdots b_m.$$

La interpretación es la siguiente:

$$b_j = \begin{cases} 1, & \text{si contratamos al especialista } e_j, \\ 0, & \text{si no contratamos al especialista } e_j. \end{cases}$$

Por tanto, el certificado y representa el subconjunto de especialistas

$$E_y = \{e_j \in E : b_j = 1\}.$$

Esta forma de certificado es suficiente porque cualquier posible solución del problema consiste precisamente en elegir un subconjunto de especialistas. Dado cualquier subconjunto $E' \subseteq E$, podemos representarlo mediante el vector binario $(b_1, \dots, b_m)$ definido por

$$b_j = 1 \iff e_j \in E'.$$

Recíprocamente, cualquier vector binario $(b_1, \dots, b_m) \in \{0, 1\}^m$ determina el subconjunto de especialistas

$$E_y = \{e_j \in E : b_j = 1\}.$$

Además, la longitud del certificado está acotada polinómicamente por la longitud de la entrada. En efecto, con la codificación anterior, el certificado contiene exactamente un bit por cada especialista, luego $|y| = m$. Por otra parte, la palabra $x = \langle I \rangle$ codifica el conjunto de especialistas $E = \{e_1, \dots, e_m\}$ y, con la codificación mediante matriz de incidencia que estamos considerando, codifica también la información necesaria para identificar las m columnas correspondientes a

dichos especialistas. En particular, la entrada x debe contener información suficiente para saber que hay m columnas. Esto implica que el tamaño de la codificación está acotado inferiormente por una cantidad proporcional a m . Es decir, existe una constante $c > 0$, dependiente solo de la codificación elegida, tal que

$$m \leq c|x|$$

para toda instancia x correctamente codificada. Si $m = 0$, la desigualdad es trivial. Así,

$$|y| = m \leq c|x|.$$

Tomando $p(t) = ct$, obtenemos

$$|y| \leq p(|x|).$$

Definimos ahora la relación binaria

$$R : A^* \times A^* \longrightarrow \{0, 1\}.$$

Dadas dos palabras $x, y \in A^*$, definimos

$$R(x, y) = 1$$

si y solo si se cumplen todas las siguientes condiciones:

1. x codifica correctamente una instancia

$$(D, E, \{D_j\}_{j=1}^m, K)$$

del problema CE .

2. y codifica un vector binario de longitud m , es decir,

$$y = b_1 b_2 \cdots b_m$$

con

$$b_j \in \{0, 1\}$$

para todo $j \in \{1, \dots, m\}$.

3. El número de especialistas seleccionados por y es menor o igual que K , es decir,

$$\sum_{j=1}^m b_j \leq K.$$

4. Todo deporte queda cubierto por algún especialista seleccionado. Es decir, para cada deporte $d_i \in D$, existe algún índice $j \in \{1, \dots, m\}$ tal que

$$b_j = 1 \quad \text{y} \quad d_i \in D_j.$$

En cualquier otro caso, definimos

$$R(x, y) = 0.$$

Nótese que en la última condición debe aparecer $d_i \in D_j$, ya que D_j es el conjunto de deportes que conoce el especialista e_j . Por tanto, los elementos de D_j son deportes, no especialistas.

Veamos que la relación R se puede calcular en tiempo polinómico. Consideramos el Algoritmo 1, que calcula exactamente la relación $R(x, y)$.

Algoritmo 1 Verificador para Contratación Eficiente

Entrada: Dos palabras $x, y \in A^*$.

Comprobar si x codifica correctamente una instancia $(D, E, \{D_j\}_{j=1}^m, K)$ de CE.

Si x no codifica correctamente una instancia del problema **entonces**

Devolver 0.

Fin Si

Comprobar si y codifica una palabra binaria $b_1b_2 \cdots b_m$ de longitud m .

Si y no codifica una palabra binaria de longitud m **entonces**

Devolver 0.

Fin Si

Inicializar un contador $c = 0$.

Para $j = 1, \dots, m$ **hacer**

Si $b_j = 1$ **entonces**

Incrementar c en una unidad.

Fin Si

Fin Para

Si $c > K$ **entonces**

Devolver 0.

Fin Si

Para $i = 1, \dots, n$ **hacer**

Inicializar una variable booleana *cubierto* = Falso.

Para $j = 1, \dots, m$ **hacer**

Si $b_j = 1$ y $d_i \in D_j$ **entonces**

Poner *cubierto* = Verdadero.

Fin Si

Fin Para

Si *cubierto* = Falso **entonces**

Devolver 0.

Fin Si

Fin Para

Devolver 1.

Primero, el algoritmo verifica que x codifica una instancia válida del problema. Después comprueba que y codifica un vector binario de longitud m . Luego cuenta cuántos especialistas han sido seleccionados y devuelve 0 si son más de K (porque rebasaríamos el límite de contratos K). Finalmente, para cada deporte d_i , comprueba si existe algún especialista seleccionado e_j que conozca dicho deporte, es decir, si existe $j \in \{1, \dots, m\}$ con

$$b_j = 1 \quad \text{y} \quad d_i \in D_j.$$

Este algoritmo tiene tiempo polinómico en la longitud de x y de y . En efecto, la comprobación de que x e y están bien codificados se hace recorriendo sus codificaciones. Después, el algoritmo recorre el vector y para contar los especialistas seleccionados. El bucle principal recorre, para cada deporte d_i , todos los especialistas e_j y comprueba si $d_i \in D_j$. Esta última comprobación puede realizarse recorriendo la lista D_j , o equivalentemente consultando la entrada correspondiente de la matriz de incidencia de la instancia x .

Por tanto, el número total de operaciones está acotado por una cantidad polinómica en $|x| + |y|$. Como ya hemos probado que existe un polinomio p tal que

$$|y| \leq p(|x|),$$

cualquier tiempo polinómico en $|x| + |y|$ queda también acotado por un polinomio en $|x|$.

En efecto, si el algoritmo tarda, para cierto $q \in \mathbb{N}$, un tiempo de orden

$$O((|x| + |y|)^q),$$

entonces

$$(|x| + |y|)^q \leq (|x| + p(|x|))^q.$$

Como p es un polinomio, escribimos

$$p(t) = a_0 + a_1 t + \cdots + a_r t^r.$$

Entonces

$$t + p(t) = a_0 + (a_1 + 1)t + a_2 t^2 + \cdots + a_r t^r,$$

que sigue siendo un polinomio en t . Por el binomio de Newton, elevarlo a la potencia q vuelve a dar un polinomio:

$$(t + p(t))^q = \sum_{\ell=0}^q \binom{q}{\ell} t^{q-\ell} p(t)^\ell.$$

Cada término

$$t^{q-\ell} p(t)^\ell$$

es un polinomio en t , y la suma finita de polinomios es un polinomio, así como el producto de una constante (en este caso, $\binom{q}{\ell}$) por un polinomio. Por tanto,

$$(|x| + p(|x|))^q$$

está acotado por un polinomio en $|x|$. Así, R es calculable en tiempo polinómico.

Veamos ahora que esta relación caracteriza exactamente las instancias positivas de CE . Supongamos primero que x codifica una instancia positiva de CE . Entonces existe un subconjunto de especialistas

$$E' \subseteq E$$

tal que

$$|E'| \leq K$$

y tal que todos los deportes quedan cubiertos. Esto significa que para cada deporte $d_i \in D$ existe algún especialista $e_j \in E'$ verificando

$$d_i \in D_j.$$

Construimos el certificado $y = b_1 b_2 \cdots b_m$ definiendo, para cada $j \in \{1, \dots, m\}$,

$$b_j = \begin{cases} 1, & \text{si } e_j \in E', \\ 0, & \text{si } e_j \notin E'. \end{cases}$$

Entonces y codifica un vector binario de longitud m . Además, como $|E'| \leq K$, se tiene

$$\sum_{j=1}^m b_j \leq K.$$

Por otra parte, como E' cubre todos los deportes, para cada $d_i \in D$ existe algún $e_j \in E'$ con $d_i \in D_j$. Por la definición de y , esto equivale a que existe $j \in \{1, \dots, m\}$ tal que

$$b_j = 1 \quad \text{y} \quad d_i \in D_j.$$

Consecuentemente,

$$R(x, y) = 1.$$

Recíprocamente, supongamos que existe un certificado $y \in A^*$ tal que

$$R(x, y) = 1.$$

Por definición de R , la palabra y codifica un vector binario de longitud m :

$$y = b_1 b_2 \cdots b_m.$$

Definimos

$$E_y = \{e_j \in E : b_j = 1\}.$$

Como $R(x, y) = 1$, tenemos

$$\sum_{j=1}^m b_j \leq K.$$

Luego

$$|E_y| \leq K.$$

Además, también por la definición de R , para cada deporte $d_i \in D$ existe $j \in \{1, \dots, m\}$ tal que

$$b_j = 1 \quad \text{y} \quad d_i \in D_j.$$

Como $b_j = 1$ implica $e_j \in E_y$ (por definición de E_y), se deduce que cada deporte d_i es conocido por algún especialista de E_y . Por tanto, E_y es un conjunto de como mucho K especialistas que cubre todos los deportes.

Necesariamente entonces x codifica una instancia positiva de CE . Así pues,

$$x \in CE \iff \exists y \in A^*, |y| \leq p(|x|), R(x, y) = 1.$$

Como R es calculable en tiempo polinómico, por la definición alternativa de **NP** mediante verificador y certificado, concluimos que

$$CE \in \mathbf{NP}.$$

Con esto queda comprobada la primera parte de la demostración.

2.2. El problema es NP-completo

Como ya hemos probado que $CE \in \mathbf{NP}$, para demostrar que CE es NP-completo basta probar que podemos reducir desde un problema NP-completo P' a CE . Para ello, como ya adelantamos en la introducción, reduciremos desde Cubrimiento por Vértices (CV), que es uno de los problemas NP-completos conocidos de la lista proporcionada. Recordemos su definición.

Definición 2.1 (Cubrimiento por Vértices (CV)). Dado un grafo $G = (V, \mathcal{A})$ y un número natural $J \leq |V|$, determinar si existe un cubrimiento por vértices C de tamaño menor o igual que J , donde $C \subseteq V$ es un cubrimiento por vértices si todo enlace del grafo tiene, al menos, uno de sus extremos en C .

Usamos la notación $\mathcal{A}$ para el conjunto de aristas, para no confundirlo ni con el alfabeto de codificación A ni con el conjunto de especialistas del problema CE . Queremos construir una reducción

$$CV \propto CE.$$

Para ello, según lo que se ha visto en teoría, debemos dar un algoritmo determinista que, en espacio logarítmico, calcule una función

$$REDU : X_{CV} \longrightarrow X_{CE}$$

tal que, para toda instancia x de CV ,

$$CV(x) = CE(REDU(x)).$$

Equivalentemente, como estamos usando codificaciones sobre el alfabeto $A = \{0, 1\}$, dada una palabra

$$x = \langle G, J \rangle$$

que codifica una instancia de Cubrimiento por Vértices, construiremos una palabra

$$REDU(x) = \langle I' \rangle$$

que codifica una instancia I' de Contratación Eficiente, de forma que

$$\langle G, J \rangle \in CV \iff \langle I' \rangle \in CE.$$

Construcción de la reducción

Sea $x = \langle G, J \rangle$ una instancia cualquiera de Cubrimiento por Vértices, donde

$$G = (V, \mathcal{A})$$

es un grafo, $\mathcal{A}$ es su conjunto de aristas y $J \leq |V|$. Supondremos, para nuestra construcción, que los vértices y las aristas de G vienen dados por

$$V = \{v_1, \dots, v_m\}$$

y

$$\mathcal{A} = \{a_1, \dots, a_n\}.$$

La instancia de CE que construiremos tendrá m especialistas, uno por cada vértice, y n deportes, uno por cada arista. Cada arista $a_i \in \mathcal{A}$ tiene la forma $a_i = \{u, v\}$ con $u, v \in V$. Construimos la instancia

$$I' = (D, E, \{D_j\}_{j=1}^m, K)$$

de Contratación Eficiente como sigue.

Por cada arista $a_i \in \mathcal{A}$ creamos un deporte d_i . Por tanto,

$$D = \{d_1, \dots, d_n\}.$$

Por cada vértice $v_j \in V$ creamos un especialista e_j . Por tanto,

$$E = \{e_1, \dots, e_m\}.$$

Definimos ahora qué deportes conoce cada especialista. Para cada $j \in \{1, \dots, m\}$, el especialista e_j conoce exactamente los deportes correspondientes a las aristas incidentes con el vértice v_j . Es decir,

$$D_j = \{d_i \in D : v_j \text{ es extremo de la arista } a_i\}. \quad (1)$$

Finalmente, tomamos como límite de contratos el mismo número que en la instancia de Cubrimiento por Vértices:

$$K = J.$$

Por tanto, la reducción queda definida como

$$REDU(\langle G, J \rangle) = \langle D, E, \{D_j\}_{j=1}^m, K \rangle.$$

Intuitivamente, las aristas del grafo pasan a ser deportes, los vértices pasan a ser especialistas, y elegir un vértice para cubrir aristas equivale a contratar al especialista correspondiente para cubrir los deportes asociados a dichas aristas.

El algoritmo *REDU* es logarítmico en espacio

Veamos que *REDU* puede calcularse en espacio logarítmico. La entrada de la reducción es una codificación

$$x = \langle G, J \rangle$$

de una instancia de Cubrimiento por Vértices, donde $G = (V, \mathcal{A})$. La salida debe ser la codificación

$$REDU(x) = \langle I' \rangle$$

de la instancia

$$I' = (D, E, \{D_j\}_{j=1}^m, K)$$

de Contratación Eficiente construida anteriormente. Usando la codificación mediante matriz de incidencia descrita en el apartado en que se comprobó que $CE \in \mathbf{NP}$, basta escribir:

1. El número de deportes de la nueva instancia: $n' = |\mathcal{A}|$.
2. El número de especialistas de la nueva instancia: $m' = |V|$.
3. El límite de contratos: $K = J$.
4. La matriz binaria de incidencia $M' = (\alpha_{ij})_{1 \leq i \leq n', 1 \leq j \leq m'}$, donde

$$\alpha_{ij} = \begin{cases} 1, & \text{si el vértice } v_j \text{ es extremo de la arista } a_i, \\ 0, & \text{en otro caso.} \end{cases}$$

Algoritmo 2 Reducción de Cubrimiento por Vértices a Contratación Eficiente

Entrada: Una palabra $x = \langle G, J \rangle$ que codifica una instancia de *CV*, con $G = (V, \mathcal{A})$.

Comprobar que x codifica correctamente una instancia de *CV*.

Si x no codifica correctamente una instancia de *CV* **entonces**

Escribir una instancia (negativa) fija de *CE*. P. ej: $D = \{d_1\}$, $E = \{e_1\}$, $D_1 = \emptyset$ y $K = 0$. Terminar.

Fin Si

Enumerar los vértices como

$$V = \{v_1, \dots, v_m\}$$

y las aristas como

$$\mathcal{A} = \{a_1, \dots, a_n\}.$$

Escribir en la salida la codificación del número de deportes $n' = |\mathcal{A}|$.

Escribir en la salida la codificación del número de especialistas $m' = |V|$.

Escribir en la salida la codificación del límite de contratos $K = J$.

Para $i = 1, \dots, n'$ **hacer**

Localizar en la entrada la arista $a_i = \{u, v\}$.

Para $j = 1, \dots, m'$ **hacer**

Localizar en la entrada el vértice v_j .

Si $v_j = u$ o $v_j = v$ **entonces**

Escribir 1 en la posición (i, j) de la matriz de incidencia de salida.

Si no

Escribir 0 en la posición (i, j) de la matriz de incidencia de salida.

Fin Si

Fin Para

Fin Para

El Algoritmo 2 calcula exactamente la instancia $REDU(\langle G, J \rangle)$. En efecto, escribe una instancia de CE con un deporte por cada arista de G , un especialista por cada vértice de G , el mismo límite $K = J$, y una matriz de incidencia que indica que el especialista e_j conoce el deporte d_i si y solo si el vértice v_j es extremo de la arista a_i .

Veamos ahora que usa espacio logarítmico. El algoritmo escribe la salida de izquierda a derecha, de modo que no necesita almacenar la instancia completa de salida, sino únicamente calcular cada símbolo en el momento de escribirlo.

Para calcular la matriz M' , el algoritmo recorre los pares de índices

$$(i, j), \quad 1 \leq i \leq |\mathcal{A}|, \quad 1 \leq j \leq |V|.$$

Estos índices se almacenan mediante contadores binarios. Como la entrada $x = \langle G, J \rangle$ codifica explícitamente el grafo $G = (V, \mathcal{A})$, su longitud crece al menos linealmente con el número de vértices y aristas. Por tanto, $|V|$ y $|\mathcal{A}|$ están acotados por una constante por $|x|$, y cada contador ocupa $O(\log |x|)$ espacio.

Para calcular el valor α_{ij} , el algoritmo localiza en la entrada la arista $a_i = \{u, v\}$ y comprueba si el vértice v_j coincide con alguno de sus extremos. Es decir, comprueba si

$$v_j = u \quad \vee \quad v_j = v.$$

Esta comprobación solo requiere guardar los índices i y j , recorrer la entrada y comparar identificadores de vértices. Cada identificador de vértice puede almacenarse con $O(\log |V|)$ bits, y por tanto con $O(\log |x|)$ bits. Todo ello puede hacerse usando espacio logarítmico.

Por tanto, durante todo el cálculo solo se almacenan contadores e índices de tamaño logarítmico, junto con una cantidad constante de información auxiliar. La cinta de salida no cuenta como espacio de trabajo y se escribe de izquierda a derecha. Así, el algoritmo $REDU$ es determinista y obtiene la transformación en espacio logarítmico.

Si CV tiene solución positiva, entonces CE también

Supongamos que

$$\langle G, J \rangle \in CV.$$

Entonces, por definición de CV , existe un cubrimiento por vértices $C \subseteq V$ tal que $|C| \leq J$ y tal que toda arista de G tiene al menos uno de sus extremos en C . A partir de C , construimos el siguiente conjunto de especialistas:

$$E' = \{e_j \in E : v_j \in C\}.$$

Como la aplicación $v_j \mapsto e_j$ es biyectiva entre V y E (por construcción), al seleccionar los especialistas correspondientes a los vértices de C se conserva el cardinal. Por tanto,

$$|E'| = |C| \leq J = K.$$

Veamos ahora que E' cubre todos los deportes. Sea $d_i \in D$ un deporte cualquiera. Nuevamente por construcción, d_i corresponde a la arista $a_i \in \mathcal{A}$. Escribimos $a_i = \{u, v\}$. Como C es un cubrimiento por vértices, la arista a_i tiene al menos un extremo en C . Por tanto,

$$u \in C \quad \vee \quad v \in C.$$

Si $u \in C$, entonces existe un índice $j \in \{1, \dots, m\}$ tal que $u = v_j$. Por definición de E' , se tiene $e_j \in E'$. Además, como v_j es extremo de la arista a_i , por definición de D_j (1) se tiene que

$d_i \in D_j$. Por ende el deporte d_i queda cubierto por el especialista contratado e_j .

La otra alternativa es que $v \in C$, y el razonamiento es completamente análogo: existe un índice $j \in \{1, \dots, m\}$ tal que $v = v_j$, entonces $e_j \in E'$ y, como v_j es extremo de a_i , se tiene $d_i \in D_j$.

En cualquier caso, el deporte d_i queda cubierto por algún especialista de E' . Como d_i era arbitrario, concluimos que todos los deportes de D quedan cubiertos por especialistas de E' . Además, ya hemos visto que $|E'| \leq K$. Por tanto,

$$REDU(\langle G, J \rangle) \in CE.$$

Hemos demostrado la implicación hacia la derecha

$$\langle G, J \rangle \in CV \implies REDU(\langle G, J \rangle) \in CE.$$

Si CE tiene solución positiva, entonces CV también

Supongamos ahora que

$$REDU(\langle G, J \rangle) \in CE.$$

Esto significa que la instancia construida de Contratación Eficiente es positiva. Por tanto, existe un subconjunto de especialistas $E' \subseteq E$ tal que $|E'| \leq K$ y tal que para todo deporte $d_i \in D$ existe algún especialista $e_j \in E'$ con $d_i \in D_j$. A partir de E' , construimos el siguiente conjunto de vértices:

$$C = \{v_j \in V : e_j \in E'\}.$$

Como hay exactamente un vértice v_j por cada especialista e_j , repitiendo un argumento anterior, se tiene

$$|C| = |E'| \leq K = J.$$

Veamos que C es un cubrimiento por vértices de G (es decir, veamos que tenemos una instancia positiva de CV). Sea

$$a_i = \{u, v\} \in \mathcal{A}$$

una arista cualquiera de G , para cierto $i \in \{1, \dots, n\}$. Por construcción, a esta arista le corresponde el deporte $d_i \in D$. Como E' cubre todos los deportes (porque suponemos que $REDU(\langle G, J \rangle) \in CE$), existe algún especialista $e_j \in E'$ tal que $d_i \in D_j$. Por definición de D_j (1), que $d_i \in D_j$ significa que el vértice v_j es extremo de la arista a_i . Por tanto,

$$v_j = u \quad \vee \quad v_j = v.$$

Además, como $e_j \in E'$, por definición de C tenemos $v_j \in C$. Luego la arista a_i tiene al menos un extremo en C . Como a_i era una arista arbitraria, toda arista de G tiene al menos un extremo en C . Por tanto, C es un cubrimiento por vértices de G . Además, ya hemos visto en repetidas ocasiones que $|C| \leq J$. Así,

$$\langle G, J \rangle \in CV.$$

Hemos demostrado la implicación que faltaba

$$REDU(\langle G, J \rangle) \in CE \implies \langle G, J \rangle \in CV.$$

Conclusión

Por las dos implicaciones anteriores, para toda instancia $\langle G, J \rangle$ de Cubrimiento por Vértices se cumple

$$\langle G, J \rangle \in CV \iff REDU(\langle G, J \rangle) \in CE.$$

Además, $REDU$ es un algoritmo determinista y logarítmico en espacio. Por tanto,

$$CV \propto CE.$$

Como CV es NP-completo, y hemos demostrado que $CE \in \mathbf{NP}$, concluimos que

$$CE \text{ es NP-completo}$$

como se pedía demostrar.

3. Bibliografía

1. OpenAI, *ChatGPT* (versión GPT-5.5), modelo de lenguaje. Disponible en <https://chat.openai.com>, consultado en junio de 2026.
2. Material de PRADO. Especialmente Tema 3 y Tema 4 Parte 1.