

# Análisis de Problemas NP Clásicos.

## Número Cromático

¿Cuál es la complejidad de  $k$ -Coloreo? Primero que todo, el caso  $k = 2$  es un problema conocido. Determinar si un grafo  $G$  es bipartito, es equivalente a la siguiente pregunta: ¿Se pueden colorear los nodos de  $G$  en rojo y azul de manera que cada arista tenga un extremo rojo y un extremo azul?

Esta última pregunta es precisamente el problema de Coloreo de Grafos en el caso en que hay  $k = 2$  colores (es decir, rojo y azul) disponibles. Así que:

Un grafo  $G$  es 2-coloreable si y solo si es bipartito.

Y chequear que un grafo es bipartito se puede hacer eficientemente en  $O(n + m)$  con un BFS, ya que necesitamos que detecte si hay ciclos impares, y este detectaría si hubiera alguno. Tan pronto como pasamos a  $k = 3$  colores, las cosas se vuelven mucho más difíciles.

## Demostración de que 3-Coloreo es NP-Completo

El problema pertenece a NP porque, dados  $G$  y  $k$ , un certificado de que la respuesta es *sí* es simplemente un  $k$ -coloreo: uno puede verificar en tiempo polinómico que se usan como máximo  $k$  colores, y que ningún par de nodos unidos por una arista reciben el mismo color.

3-Coloreo es difícil de relacionar con otros problemas NP-completos que hemos visto. Por lo tanto, vamos a retroceder hasta 3-SAT. Dada una instancia arbitraria de 3-SAT con variables  $x_1, \dots, x_n$  y cláusulas  $C_1, \dots, C_k$ , lo resolveremos usando una "caja negra" para 3-Coloreo.

El inicio de la reducción es bastante intuitivo. Definimos nodos  $v_i$  y  $\bar{v}_i$  correspondientes a cada variable  $x_i$  y su negación  $\bar{x}_i$ . También definimos tres "nodos especiales"  $T$ ,  $F$ , y  $B$ , que llamamos Verdadero, Falso y Base.

Para construir el grafo  $G$ , conectamos cada par de nodos  $v_i$  y  $\bar{v}_i$  entre sí con una arista, y ambos se conectan al nodo Base. También conectamos los nodos Verdadero, Falso y Base en un triángulo. Estas conexiones aseguran que los nodos  $v_i$  y  $\bar{v}_i$  tengan colores diferentes, y que uno de ellos sea Verdadero y el otro Falso. Por último, para cada cláusula, adjuntamos un subgrafo como el de la Figura 2.

Al final, mostramos que la instancia de 3-SAT es satisfactible si y solo si  $G$  es 3-coloreable.

El grafo  $G$  que hemos definido hasta ahora ya tiene algunas propiedades útiles:

- En cualquier 3-coloreo de  $G$ , los nodos  $v_i$  y  $\bar{v}_i$  deben recibir colores diferentes, y ambos deben ser distintos del color asignado a la Base.

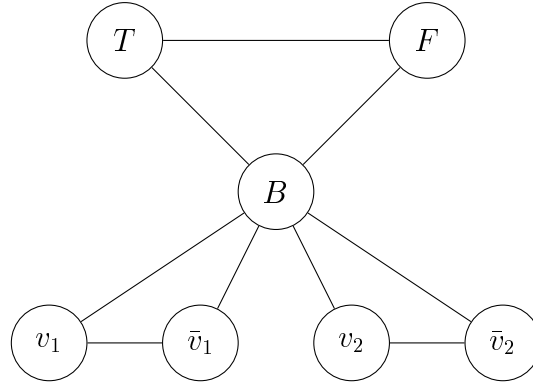


Figura 1: Ejemplo del principio de reducción para 3-Coloreo.

- En cualquier 3-coloreo de  $G$ , los nodos Verdadero, Falso y Base deben recibir los tres colores en alguna permutación. Por lo tanto, podemos referirnos a los tres colores como el color Verdadero, el color Falso y el color Base, dependiendo de cuál de estos tres nodos recibe cada color. En particular, esto significa que para cada  $i$ , uno de  $v_i$  o  $\bar{v}_i$  recibe el color Verdadero, y el otro recibe el color Falso. Para el resto de la construcción, consideraremos que la variable  $x_i$  se establece en 1 en la instancia dada de 3-SAT si y solo si el nodo  $v_i$  recibe el color Verdadero.

En resumen, ahora tenemos un grafo  $G$  en el que cualquier 3-coloreo determina implícitamente una asignación de valores de verdad para las variables en la instancia de 3-SAT. Ahora necesitamos expandir  $G$  para que solo las asignaciones que satisfacen las cláusulas puedan extenderse a 3-coloreos del grafo completo. ¿Cómo podemos hacer esto?

Como en otras reducciones de 3-SAT, consideremos una cláusula como  $x_1 \vee \bar{x}_2 \vee x_3$ . En el lenguaje de 3-coloreos de  $G$ , esto significa: 'Al menos uno de los nodos  $v_1$ ,  $\bar{v}_2$  o  $v_3$  debe recibir el color Verdadero'. Por lo tanto, necesitamos un pequeño subgrafo que podamos conectar a  $G$ , de modo que cualquier 3-coloreo que se extienda a este subgrafo tenga la propiedad de asignar el color Verdadero a al menos uno de  $v_1$ ,  $\bar{v}_2$  o  $v_3$ .

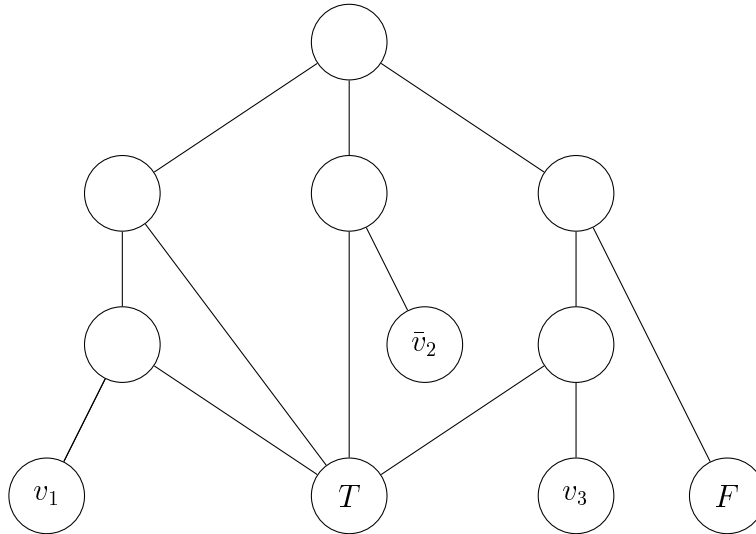


Figura 2: Un subgrafo que representa la cláusula  $x_1 \vee \bar{x}_2 \vee x_3$ .

Este subgrafo de seis nodos se “conecta” al resto de  $G$  en cinco nodos existentes: Verdadero, Falso y aquellos correspondientes a los tres términos en la cláusula que estamos

tratando de representar (en este caso,  $v_1$ ,  $\bar{v}_2$  y  $v_3$ ). Ahora supongamos que en algún 3-coloreo de  $G$ , los tres nodos  $v_1$ ,  $\bar{v}_2$  y  $v_3$  son asignados con el color Falso. Entonces, los dos nodos sombreados más bajos en el subgrafo deben recibir el color Base, y los tres nodos sombreados por encima de ellos deben recibir, respectivamente, los colores Falso, Base y Verdadero. Por lo tanto, no hay un color que se pueda asignar al nodo sombreado superior. En otras palabras, un 3-coloreo en el que ninguno de  $v_1$ ,  $v_2$  o  $v_3$  reciba el color Verdadero no puede extenderse a un 3-coloreo de este subgrafo.

Finalmente, y de forma inversa, una revisión manual de los casos muestra que siempre y cuando uno de  $v_1$ ,  $\bar{v}_2$  o  $v_3$  reciba el color Verdadero, el subgrafo completo puede ser 3-coloreado.

A partir de esto, podemos completar la construcción: Comenzamos con el grafo  $G$  definido anteriormente y, para cada cláusula en la instancia de 3-SAT, conectamos un subgrafo de seis nodos como se muestra en la Figura. Llamemos al grafo resultante  $G'$ .

Afirmamos ahora que la instancia dada de 3-SAT es satisfactible si y solo si  $G$  tiene un 3-coloreo. Primero, supongamos que existe una asignación satisfactoria para la instancia de 3-SAT. Definimos un coloreo de  $G$  coloreando primero los nodos Base, Verdadero y Falso arbitrariamente con los tres colores disponibles. Luego, para cada  $i$ , asignamos el color Verdadero a  $v_i$  si  $x_i = 1$ , y el color Falso si  $x_i = 0$ . Después asignamos a  $\bar{v}_i$  el único color disponible. Finalmente, como se argumentó anteriormente, ahora es posible extender este 3-coloreo a cada subgrafo de seis nodos correspondiente a una cláusula, lo que da como resultado un 3-coloreo para todo  $G$ .

En sentido inverso, supongamos que  $G$  tiene un 3-coloreo. En este coloreo, cada nodo  $v_i$  está asignado al color Verdadero o al color Falso; establecemos la variable  $x_i$  correspondientemente. Ahora afirmamos que, en cada cláusula de la instancia de 3-SAT, al menos uno de los términos en la cláusula tiene el valor de verdad 1. Si no, entonces los tres nodos correspondientes tienen el color Falso en el 3-coloreo de  $G$  y, como vimos anteriormente, no es posible 3-colorear el subgrafo correspondiente a la cláusula de manera consistente, lo cual es una contradicción.

Cuando  $k > 3$ , es muy fácil reducir el problema de 3-Coloreo al problema de  $k$ -Coloreo. Básicamente, lo que hacemos es tomar una instancia del problema de 3-Coloreo, representada por un grafo  $G$ , agregar  $k - 3$  nuevos nodos y unir estos nuevos nodos entre sí y con todos los nodos en  $G$ . El grafo resultante es  $k$ -coloreable si y solo si el grafo original  $G$  es 3-coloreable. Por lo tanto, el problema de  $k$ -Coloreo para cualquier  $k > 3$  también es NP-completo.

## EXACT-COVER

Dado un conjunto finito  $A$  y una familia  $F$  de subconjuntos de  $A$ , determinar si existe una subfamilia  $F'$  de  $F$  verificando la siguiente propiedad:

Para cada  $x \in A$  existe un único conjunto  $A_x \in F'$  tal que  $x \in A_x$ .

En el contexto anterior, se dice que  $F'$  es un *recubrimiento exacto* de  $(A, F)$ .

## Reducción polinomial desde 3-COL a EXACT-COVER

Probar que existe una reducibilidad en tiempo polinomial del problema 3-COL en el problema EXACT-COVER, concluyendo que el problema EXACT-COVER es NP-

completo.

## Indicaciones

Considérese la aplicación  $F : E_{3-COL} \rightarrow E_{\text{EXACT-COVER}}$  definida como sigue: sea  $G = (V, E)$  un grafo no dirigido,

■ **Conjunto  $A$ :**

- Cada nodo  $u$  del grafo proporciona los elementos  $u, p_1^u, p_2^u, p_3^u$ . Esto representa las posibles asignaciones de colores al nodo  $u$ , donde cada elemento  $p_i^u$  corresponde a un color.
- Cada arista  $\{u, w\}$  del grafo proporciona los elementos  $q_1^{uw}, q_2^{uw}, q_3^{uw}, q_1^{wu}, q_2^{wu}, q_3^{wu}$ . Esto asegura que las restricciones de colores entre nodos adyacentes se cumplan.

■ **Familia  $F$  de subconjuntos de  $A$ :**

$$F = \bigcup_{u \in V} \{\{u, p_1^u\}, \{u, p_2^u\}, \{u, p_3^u\}, A_1^u, A_2^u, A_3^u\} \cup \bigcup_{\{u, w\} \in E} \{\{q_i^{uw}, q_j^{wu}\} : 1 \leq i < j \leq 3\}$$

donde, para cada  $u \in V$  y cada  $j$  ( $1 \leq j \leq 3$ ):

$$A_j^u = \{q_j^{uw} : \{u, w\} \in E\} \cup \{p_j^u\}.$$

Estos subconjuntos conectan las posibles asignaciones de colores con las restricciones de las aristas incidentes.

## Probar la equivalencia

Probar que un grafo no dirigido  $G$  es coloreable con tres colores si y sólo si  $(A, F)$  posee un recubrimiento exacto.

### Si $f$ es una coloración válida

Construyamos la familia:

$$F' = \bigcup_{u \in V} \{\{u, p_{f(u)}^u\}\} \cup \bigcup_{u \in V} \{A_j^u : j \neq f(u)\} \cup \bigcup_{\{u, w\} \in E} \{\{q_{f(u)}^{uw}, q_{f(w)}^{wu}\}\}.$$

Esto asegura que cada nodo tiene un color asignado y que las restricciones de las aristas se cumplen.

### Si $F'$ es un recubrimiento exacto

- Para cada  $x \in V$ , existe un único  $c_x \in \{1, 2, 3\}$  tal que  $\{x, p_{c_x}^x\} \in F'$ .
- Considérese la coloración  $f$  definida como:  $f(x) = c_x$ . Probar que dicha coloración de  $G$  es válida.

## Motivación de la Construcción

La construcción divide el problema en partes:

- Representar las posibles asignaciones de colores a los nodos.
- Manejar las restricciones impuestas por las aristas.
- Garantizar que la solución final sea válida tanto para EXACT-COVER como para 3-COL.

Esta correspondencia asegura que cualquier solución al problema EXACT-COVER también resuelve el problema 3-COL, y viceversa, completando la reducción polinomial.

## Clique

Dado un grafo  $G = (V, E)$  y un entero  $k$ , determinar si existe un subgrafo completo (clique) de tamaño al menos  $k$  en  $G$ .

## Demostración de que Clique es NP-Hard

Para demostrar que el problema de decisión de Clique es NP-Hard, realizamos una reducción polinomial desde un problema conocido como NP-completo, en este caso, 3-SAT (Satisfiabilidad Booleana con cláusulas de 3 literales).

## Reducción de 3-SAT a Clique

Sea dado un problema de 3-SAT con  $n$  variables  $x_1, x_2, \dots, x_n$  y  $m$  cláusulas  $C_1, C_2, \dots, C_m$ , donde cada cláusula  $C_i$  tiene exactamente 3 literales.

### Construcción del Grafo

Construimos un grafo  $G = (V, E)$  como sigue:

- Por cada literal en cada cláusula, añadimos un nodo al conjunto de vértices  $V$ . Es decir,  $V$  contiene un nodo por cada literal de cada cláusula.
- Añadimos una arista entre dos nodos  $u$  y  $v$  si:
  - $u$  y  $v$  pertenecen a cláusulas diferentes, y
  - Los literales correspondientes a  $u$  y  $v$  no se contradicen (por ejemplo, no añadimos una arista entre  $x_i$  y  $\neg x_i$ ).

### Parámetro $k$

Fijamos  $k = m$ , el número de cláusulas en el problema 3-SAT.

## Equivalencia de las Soluciones

- **Si el problema 3-SAT es satisfacible:** Existe una asignación de valores de verdad a las variables que satisface todas las cláusulas. Para cada cláusula  $C_i$ , seleccionamos un literal que sea verdadero bajo esta asignación. Los nodos correspondientes a estos literales forman un clique de tamaño  $m$  en el grafo  $G$ , ya que:
  - Los nodos seleccionados pertenecen a cláusulas diferentes.
  - Los literales seleccionados no se contradicen, por lo que las aristas correspondientes están presentes en  $G$ .
- **Si existe un clique de tamaño  $m$  en  $G$ :** Cada nodo del clique corresponde a un literal de una cláusula diferente. Dado que las aristas conectan sólo nodos cuyos literales no se contradicen, podemos asignar valores de verdad a las variables correspondientes para satisfacer todas las cláusulas del problema 3-SAT.

Dado que podemos transformar cualquier instancia de 3-SAT en una instancia equivalente del problema de decisión de Clique en tiempo polinomial, y dado que 3-SAT es NP-completo, el problema de decisión de Clique es NP-Hard.

Luego el problema de optimización de Clique no es chequeable en tiempo polinomial, porque tendríamos que revisar todos los posibles subgrafos de tamaño mayor por lo tanto no es NP.

## Demostración de que el problema de Máximo Corte es NP-Completo

### El problema está en NP

Dada una partición de los vértices  $V$  en dos subconjuntos  $S$  y  $T$ , podemos calcular la suma de los pesos de las aristas que cruzan entre  $S$  y  $T$  en tiempo polinomial, simplemente sumando los pesos de dichas aristas. Por lo tanto, podemos verificar en tiempo polinomial si existe un corte con un costo al menos  $k$ . Esto ubica al problema en NP.

### El problema es NP-Duro

Reduciremos desde el problema NAE-3SAT (Not-All-Equal 3SAT), que es NP-Completo.

**Reducción:** Dada una instancia de NAE-3SAT con variables  $x_1, x_2, \dots, x_n$  y cláusulas  $C_1, C_2, \dots, C_m$ , construiremos un grafo  $G = (V, E)$  de la siguiente manera:

- **Vértices:** Para cada variable  $x_i$ , creamos un vértice que la representa. También creamos un vértice para su negación  $\neg x_i$ .
- **Aristas de variables:** Conectamos  $x_i$  y  $\neg x_i$  con una arista de peso alto  $W$ , de manera que en el corte óptimo forzamos a que estén en lados opuestos.
- **Aristas de cláusulas:** Para cada cláusula  $C_j = (l_{j1} \vee l_{j2} \vee l_{j3})$ , donde  $l_{ji}$  es un literal (una variable o su negación), añadimos un triángulo completo conectando los vértices correspondientes a  $l_{j1}, l_{j2}$  y  $l_{j3}$ , con aristas de peso 1.

**Objetivo:** El peso total máximo del corte se logra si y sólo si, en cada cláusula, no todos los literales están asignados al mismo lado del corte, respetando la condición NAE (*Not All Equal*).

**Justificación:**

- Las aristas de peso alto  $W$  fuerzan a que cada variable  $x_i$  y su negación  $\neg x_i$  se coloquen en lados opuestos del corte.
- Las aristas de las cláusulas aseguran que, para maximizar el corte, no todos los literales de una cláusula estén en el mismo lado, cumpliendo así la condición NAE.

Dado que la construcción puede realizarse en tiempo polinomial y el problema NAE-3SAT es NP-Completo, concluimos que Máximo Corte es NP-Duro.

## Conclusión

Como el problema está en NP y es NP-Duro, concluimos que el problema de Máximo Corte es NP-Completo.

---

## Demostración de que el problema de Subgrafo Máximo Bipartito es NP-Completo

### El problema está en NP

Dado un subgrafo  $H = (V, E')$  de  $G = (V, E)$ , donde  $E' \subseteq E$ , podemos verificar en tiempo polinomial si  $H$  es bipartito utilizando un recorrido (BFS o DFS) para colorear los vértices con dos colores. Si en algún momento detectamos un conflicto (una arista cuyos extremos requieren el mismo color), sabremos que no es bipartito. Además, podemos contar el número de aristas en  $E'$  en tiempo polinomial. Por lo tanto, el problema pertenece a NP.

### El problema es NP-Duro

Reduciremos desde el problema de Máximo Corte en grafos no ponderados, que es conocido por ser NP-Duro.

**Reducción:** Dado un grafo no ponderado  $G = (V, E)$ , el problema de Máximo Corte busca una partición de  $V$  en dos subconjuntos  $S$  y  $T$  que maximice el número de aristas que cruzan entre  $S$  y  $T$ .

En un grafo bipartito, todas las aristas conectan vértices de distintos conjuntos en la partición bipartita. Por lo tanto, encontrar un subgrafo bipartito máximo es equivalente a encontrar la partición de  $V$  que maximiza el número de aristas entre  $S$  y  $T$ . Esto muestra que resolver el problema de Subgrafo Máximo Bipartito permitiría resolver Máximo Corte en grafos no ponderados. Dado que Máximo Corte es NP-Duro, concluimos que Subgrafo Máximo Bipartito también es NP-Duro.

## Conclusión

Dado que el problema está en NP y es NP-Duro, concluimos que el problema de Subgrafo Máximo Bipartito es NP-Completo.

## Problema Retroalimentación de Vértices

**Definición del problema:** Dado un grafo  $G = (V, E)$ , un conjunto de retroalimentación de vértices es un subconjunto de vértices  $F \subseteq V$  tal que, al eliminar todos los vértices de  $F$  (y sus aristas incidentes), el grafo resultante no contiene ciclos (es un grafo acíclico o un bosque, si es no dirigido). El objetivo del problema es encontrar el conjunto de retroalimentación de vértices de tamaño mínimo.

## Solución

Vamos a demostrar que el problema es NP-Hard. Para ello, primero demostraremos que el problema de retroalimentación de vértices de tamaño  $k$  es NP-Completo.

### Demostración de que es NP

Dada una instancia de retroalimentación de vértices y un conjunto solución  $D$ , basta verificar que el tamaño de  $D$  es menor o igual que  $k$  y que, al eliminar esos vértices del grafo, éste se vuelve acíclico. Para comprobar que el grafo es acíclico, se puede realizar un recorrido (por ejemplo, DFS o BFS) y verificar que no existe ningún ciclo. Esto se hace en tiempo polinomial, por lo tanto el problema está en NP.

### Demostración de que es NP-Hard

Para esta parte, reduciremos el problema *Vertex Cover* (del cual sabemos que es NP-Completo) al problema de retroalimentación de vértices.

**Reducción:** Dada una instancia de *Vertex Cover* con  $G = (V, E)$ , crearemos un grafo dirigido  $G' = (V', E')$  de la siguiente forma:

- Sea  $V' = V$ , es decir, los mismos vértices que en  $G$ .
- Por cada arista  $(a, b) \in E$  en el grafo original no dirigido, crearemos en  $G'$  las aristas dirigidas  $(a, b)$  y  $(b, a)$ , de manera que se genere un ciclo entre  $a$  y  $b$  en  $G'$ .

### Relación entre ambos problemas:

En *Vertex Cover*, necesitamos un conjunto mínimo de vértices tal que cada arista del grafo sea cubierta. En retroalimentación de vértices, necesitamos un conjunto mínimo de vértices tal que, si lo eliminamos del grafo, éste no contenga ciclos.

Ahora, dado que cada arista en  $G$  de *Vertex Cover* se transforma en dos aristas dirigidas en  $G'$ , formando un ciclo entre  $a$  y  $b$ , entonces el conjunto seleccionado para la retroalimentación de vértices (denominado  $F$ ) debe incluir al menos uno de los dos vértices para romper dicho ciclo.



Si tenemos un conjunto  $F$  en  $G$ , dicho conjunto debe cubrir cada arista en  $G$ , lo que significa que para cada ciclo dirigido correspondiente en  $G'$ ,  $F$  debe contener al menos uno de los vértices de ese ciclo para romperlo. Al eliminar  $F$  en  $G'$ , todos los ciclos dirigidos se eliminan, haciendo que  $G'$  sea acíclico.

Por tanto, si en retroalimentación de vértices hay una solución de tamaño  $k$ , entonces también la hay en *Vertex Cover*, y viceversa.

Habiendo demostrado que este problema es NP-Completo, podemos reducirlo al problema de hallar el mínimo conjunto que satisfaga retroalimentación de vértices. Pues si el mínimo es menor que  $k$ , entonces hay una solución de longitud  $k$ ; si no lo es, pues no existe tal solución. En consecuencia, el problema de hallar el mínimo conjunto que satisfaga retroalimentación de vértices es NP-Hard.

## Problema Retroalimentación de Arcos

**Definición del problema:** Dado un grafo  $G = (V, E)$ , un *conjunto de retroalimentación de arcos* es un subconjunto de arcos  $F \subseteq E$  tal que, al eliminar todos los arcos en  $F$ , el grafo resultante no contiene ciclos. El objetivo del problema es encontrar el conjunto de retroalimentación de arcos de **tamaño mínimo**.

## Solución

Al igual que en el problema anterior, demostraremos que el problema de encontrar un conjunto de arcos de longitud  $k$  tal que, al quitarlos, no queden ciclos en  $G$  es NP-Completo. Para ello, reduciremos el problema de *Retroalimentación de Vértices* a este en tiempo polinomial.

**Reducción:** Dada una instancia  $(G, k)$  de *Retroalimentación de Vértices* con  $G = (V, E)$ , crearemos una instancia  $(G', k')$  de *Retroalimentación de Arcos* de la siguiente forma:

- Por cada nodo  $v \in V$ , creamos **dos nodos**  $v_1$  y  $v_2$  en  $G'$  y añadimos el arco  $(v_1, v_2)$ . A estos arcos se les llamará *arcos internos*.
- Luego, por cada arco  $(u, v) \in E$  en el grafo original, añadimos en  $G'$  el arco externo  $(u_2, v_1)$ .

Con esto, cada vértice en  $G$  se **duplica** en  $G'$ , creando dos copias de cada vértice. Las aristas (arcos) del grafo original se dividen en:

- Arcos internos: entre las dos copias de un mismo vértice  $(v_1 \rightarrow v_2)$ .
- Arcos externos: entre las copias de vértices diferentes  $(u_2 \rightarrow v_1)$ .

Al resolver *RA* (Retroalimentación de Arcos) en  $G'$ , buscamos un conjunto de arcos cuyo borrado rompa todos los ciclos en  $G'$ . La clave está en que cualquier ciclo que involucre arcos externos también debe pasar por al menos un arco interno; por lo tanto, **eliminar sólo arcos internos** (que corresponden a eliminar vértices en  $G$ ) es suficiente para romper los ciclos.

**Correspondencia de soluciones:** Cualquier conjunto de arcos internos que forme una solución para  $RA$  en  $G'$  corresponde a un conjunto de vértices que forman una solución para *Retroalimentación de Vértices* ( $RV$ ) en  $G$ . Esto se debe a que **romper los ciclos en  $G'$  equivale a romper los ciclos en  $G$** , y viceversa.

**Conclusión:** De manera análoga al problema anterior, se concluye que hallar el **mínimo conjunto** de arcos que satisfaga  $RA$  en  $G'$  es NP-Hard, ya que se corresponde con hallar el **mínimo conjunto** de vértices en  $RV$  en  $G$ . Por lo tanto, *Retroalimentación de Arcos* es NP-Hard.