MAC-ARRONES.pdf

Jiuxe

Modelos Avanzados de Computacion

3º Grado en Ingeniería Informática

Escuela Técnica Superior de Ingenierías Informática y de Telecomunicación
Universidad de Granada

INNJOO| Tienda de productos tecnológicos

INNJOO

Granada

Auriculares

Tablet voom tab pro + teclado

Patinete

Llévate Un **patinete**, unos **auriculares** o **una tablet** voom tab pro+teclado.

Todos los estudiantes que presenten unos apuntes de WUOLAH en tienda se les aplicará un **10% de descuento** en la compra de cualquiera de nuestros productos.

G R A N A D A

Calle Puentezuelas, 49

www.innjoo.es

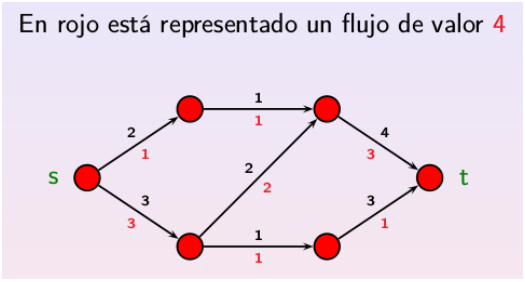
1 Problema Flujo Máximo

Tenemos un grafo dirigido en el que los arcos estan etiquetados por su capacidad. Hay un origen (s) y un destino (t).

Un flujo es una asignación de un valor a cada arco que no supere su capacidad, de forma que la suma de lo que entra a cada nodo intermedio es igual a la suma de lo que sale.

El valor de un flujo es la suma de lo que sale del origen (que es igual a la suma de lo que llega al destino).

Se trata de calcular el flujo de valor máximo.

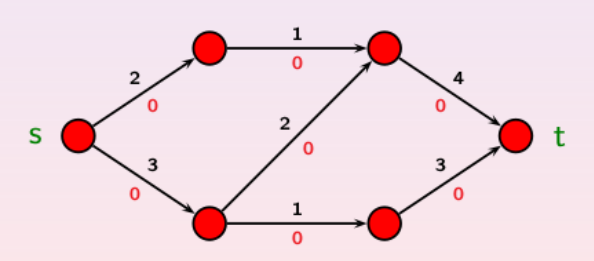


Partimos del problema (V, E, s, t, c) donde:

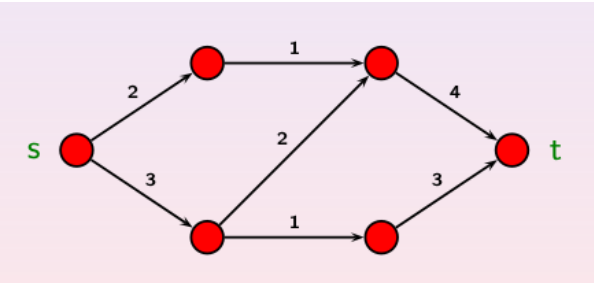
- **V**: Conjunto de vértices
- **E**: Conjunto de aristas
- **s:** Nodo inicial
- **t:** Nodo final
- **c:** Función que asigna a cada pareja de nodos (x, y) su capacidad $[c(x, y)]$.

1.1 Algoritmo Ford-Fulkenson

El **flujo** se representara con la función **f**:
1. **INICIALIZACIÓN:**Se comienza con un flujo cualquiera, $f(x, y) = 0$, por ejemplo:

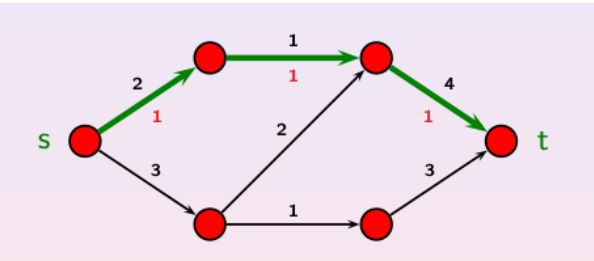


2. **GRAFO DIFERENCIA:** Restamos el flujo inicializado a la capacidad de las aristas del grafo, esto da como resultado el grafo diferencia $(c'(x,y) = c(x,y) - f(x,y), si(x,y) \in E$ o $c'(x,y) = f(x,y) si(x,y) \notin E)$, en este ejemplo se queda igual porque es 0

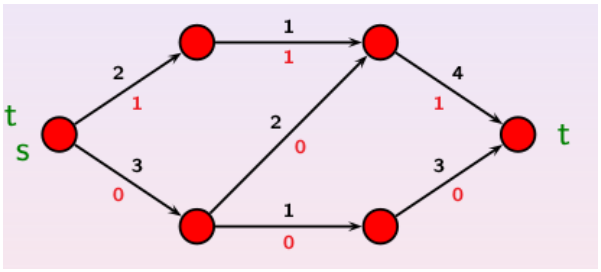


3. **BUSCAR CAMINO:** Se calcula un camino de s a t en el nuevo grafo. Se le asigna un flujo al nuevo camino que es valor mínimo de c' (en este caso es 1) en todas las aristas que lo componen.

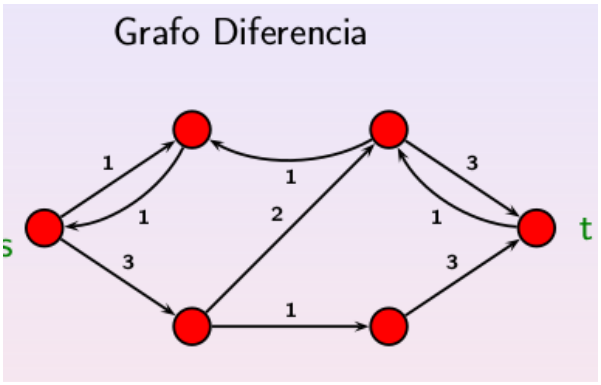
Si el camino no existe se termina y f contiene el flujo máximo.



4. **ASIGNAR FLUJO:** Asignamos el flujo al grafo por el camino que hemos elegido.

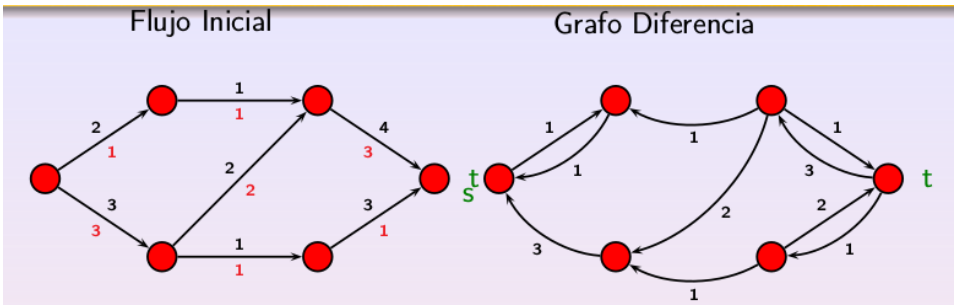


Si la capacidad de una arista es superior a la del flujo, restamos. De modo que si capacidad es 2 de la arista y el flujo 1 nos quedaría en ese arco la capacidad de 1 en el grafo de diferencia.



5. **REPETIMOS** el proceso desde el paso 2 hasta que no exista mas caminos en el grafo de diferencias.

SOLUCIÓN:
El flujo máximo es 4 ya que del nodo S la cantidad de flujo total que sale es esa $(3+1)$



2 Flujo Máximo: Complejidad (Elección del camino mínimo)

En respuesta a la pregunta del examen de *2017 Septiembre: ¿Bajo que condición se garantiza que el algoritmo tenga complejidad polinomica?*, debemos de responder:

Para obtener una complejidad polinomica dentro del problema de flujo máximo basta con elegir en cada caso el camino mas corto entre el origen y el destino, entonces el numero de iteraciones esta limitado por n^3, donde n es el numero de nodos.

Hay que tener en cuenta que un camino mínimo entre el origen y el destino es también camino mínimo entre cada dos nodos que aparezcan en dicho camino

Un arco no puede ser primer cuello de botella mas de n veces (**Cuello de botella:** es el arco de un camino con el mínimo coste). Luego el numero máximo de iteraciones es n^3.

Si aparece una vez como cuello de botella, la siguiente vez que puede ser está a más distancia del origen.

Como en cada iteración hay un primer cuello de botella, el número de iteraciones esta limitado por el numero de arcos (n^2) multiplicado por el numero máximo de veces que pueden ser primer cuello de botella (n).

La complejidad total es del orden $O(n^5)$

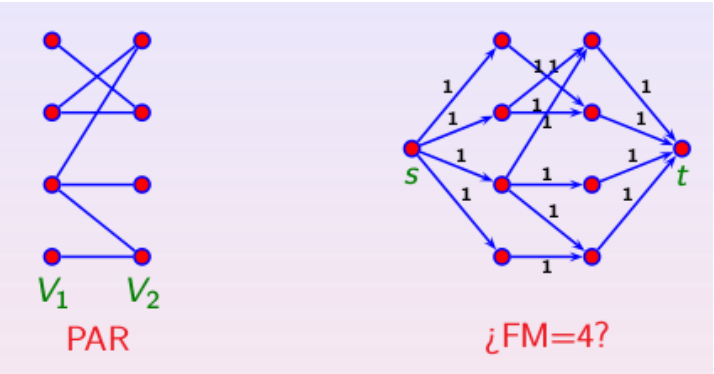
3 Problema de las Parejas

Tenemos dos conjuntos del mismo tamaño $V_1 y V_2$ y un subconjunto $A \in V_1$ X V_2 (representa las compatibilidades de elementos de V_1 con elementos de V_2). El problema consiste en decidir si existe un subconjunto $A' \in A$ tal que cada elemento de $v_1 \in V_1$ aparece en uno y solo en uno de los pares $(v_1, y) \in A'$ y cada elemento de $v_2 \in V_2$ aparece en uno y solo en uno de los pares $(x, v_2) \in A'$

3.1 Reducción de Flujo Máximo a Parejas

Para cada ejemplo de las parejas podemos construir un problema de flujo máximo equivalente:

- Añadir un nodo s y arcos dirigidos desde este nodo a todos los de V_1
- Dirigir los arcos originales desde V_1 a V_2
- Añadir un nodo final t y arcos dirigidos desde los nodos de V_2 a este nodo
- Asignar una capacidad de 1 a todos los arcos



Si m es el numero de elementos en V_1 y V_2, la pregunta equivalente al problema de las parejas es:

¿Existe un flujo máximo de tamaño m?

4 Problema de la Partición

Datos: Un conjunto A y un tamaño para cada uno de sus elementos:

$$s : A \to N$$

Pregunta: Determinista si existe un $A' \subseteq A$ tal que se verifique:

$$\sum_{a \in A'} s(a) = \sum_{a \in A/A'} s(a)$$

Este es claramente un problema de NP-Completo: se eligen de forma no determinista los elementos de A' y en tiempo polinomico se determina si los tamaños totales de los conjuntos A' y A/A' son iguales.

Para demostrar que es completo para NP, vamos a reducir el cubrimiento por tripletas (ACTRI) a este problema.

4.1 Reducción

Sea W, X, Y con $|W| = |X| = |Y| = q$ y un subconjunto $M \subseteq X \times Y \times Z$ un ejemplo del problema ACTRI, vamos a construir una partición equivalente.

Consideramos:

$$W = w_1, ..., w_q, X = x_1, ..., x_q, Y = y_1, ..., y_q, M = m_1, ..., m_k$$

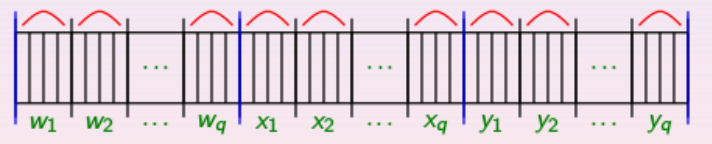
El conjunto de la partición, A, va a contener k+2 elementos. Hay k elementos $a_1, ..., a_k$ que corresponda a las k tripletas de M.

Para cada tripleta $m_i = (w_{f(i)}, x_{g(i)}, y_{h(i)})$, se considera el elemento a_i con un peso:

$$s(a_i) = 2^{p(3q-f(i))} + 2^{p(2q-g(i))} + 2^{p(q-h(i))}$$

donde $p = [log_2(k)] + 1$

En binario podemos ver el numero en grupo de p posiciones. Cada grupo corresponde a un elemento de W, X o Y:



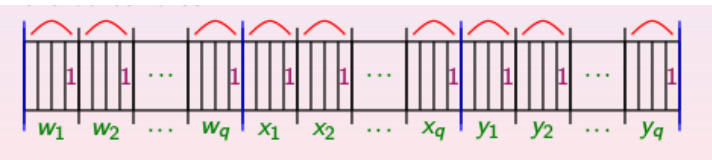
Cada tripleta $m_i = (w_{f(i)}, x_{g(i)}, y_{h(i)})$ se corresponde con un numero con un 1 a la derecha de las zonas de $w_{f(i)}, x_{g(i)}, y_{h(i)}$ y 0 en el resto

Ejemplo: Tripleta (w_2, x_1, y_q)

Si $p = [log_2(k) + 1]$, entonces sumando k o menos unos nunca obtenemos un numero con un 1 mas allá de la posición p+1.

Sumando k o menos numeros de los asociados a la tripleta, nunca pasa un 1 de la zona de un elemento a la zona de otro

Sea $B = \sum_{3q-1}^{j=0} 2^{p.j}$ este numero tiene un 1 a la derecha de cada una de las zonas:



Como una consecuencia, para cada $A' \subseteq a_1, ..., a_k$, tenemos que:

$$\sum_{a \in A'} s(a) = B$$

si y solo si $M' = m_i : a_i \in A'$ es un recubrimiento por tripletas.

Añadimos dos nuevos elementos a $A : b_1$ y b_2 con pesos:

$$s(b_1) = \sum_{i=1}^{k} s(a_i) s(b_2) = 2B$$

Cada uno de estos pesos necesita, a lo mas, $(3pq)$ bits. Se pueden calcular zona a zona de forma consecutiva.

4.2 Conjunto A

El conjunto A es igual a $a_1, ..., a_k, b_1, b_2$.

La suma de los pesos de sus elementos es:

$$\sum_{x \in A} s(x) = \sum_{i=1}^{k} s(a_i) + s(b_1) + s(b_2) =$$
$$\sum_{i=1}^{k} s(a_i) + \sum_{i=1}^{k} s(a_i) + 2B = 2 \sum_{i=1}^{k} s(a_i) + 2B$$

Este conjunto se parte en dos mitades cuando cada una pesa:

$$\sum_{i=1}^{k} s(a_i) + B$$

Si la solución es positiva al problema de la partición, entonces es positiva a ACTRI.

Supongamos $A' \subseteq A$, tal que

$$\sum_{a \in A'} s(a) = \sum_{a \in A/A'} s(a)$$

Entonces cada una de estas sumas es $\sum_{i=1}^{k} s(a_i) + B$. Uno de los conjuntos (supongamos que es A') debe de contener b_1 y no b_2. La suma de los pesos de los elementos de A' distintos de b_1 tiene que ser B.

Por tanto, las tripletas asociadas a estos elementos son un cubrimiento por tripletas del conjunto original.

Si $M' \subseteq M$ es un cubrimiento por tripletas. Entonces, el conjunto $A' = b_1 \cup a_i : m_i \in M'$ representa una respuesta positiva al problema de la partición, ya que, al ser M' una partición, la suma de los pesos de estos elementos es $\sum_{m_i \in M'} s(a_i) = B$ y, por tanto, la suma de los pesos de A' es:

$$\sum_{i=1}^{k} s(a) + B$$

que es la mitad del total.

5 Clases de Complejidad

- **Clase polinomica (tiempo):** $P = \cup TIEMPO(n^j)$ con $j > 0$
- **Clase polinomica no determinista (tiempo):** $NP = \cup NTIEMPO(n^j)$ con $j > 0$
- **Clase polinomica(espacio):** $PESPACIO = \cup ESPACIO(n^j)$ con $j > 0$
- **Clase polinomica no determinista (espacio):** $NPESPACIO(n^j)$ con $j > 0$
- **Clase de espacio logarítmico determinista:** $L = ESPACIO(log(n))$
- **Clase de espacio logaritmo no determinista:** $NL = NESPACIO(log(n))$. Ejemplo: Reconocer un palindromo y 2-SAT.
- **Clase exponencial en tiempo:** $EXP = \cup(2^{n^j})$

Relaciones:

$$L \subseteq NL \subseteq P \subseteq NP \subseteq PESPACIO$$

No se sabe si alguna de estas inclusiones son igualdades, pero si se sabe que **NL** esta incluida estrictamente en **PESPACIO**.

Definiciones de TIEMPO y ESPACIO:

- **TIEMPO(f):** Todos los lenguajes aceptados por una maquina de Turing determinista en tiempo $O(f(n))$
- **ESPACIO(f):** Todos los lenguajes aceptados por una maquina de Turing determinista en espacio $O(f(n))$
- **NTIEMPO(f):** Todos los lenguajes aceptados por una maquina de Turing n determinista en tiempo $O(f(n))$
- **NESPACIO(f):** Todos los lenguajes aceptados por una maquina de Turing n determinista en espacio $O(f(n))$

Relaciones:

a) $\text{ESPACIO}(f(n)) \subseteq \text{NESPACIO}(f(n))$
$\text{TIEMPO}(f(n)) \subseteq \text{NTIEMPO}(f(n))$
b) $\text{NTIEMPO}(f(n)) \subseteq \text{ESPACIO}(f(n))$
c) $\text{NESPACIO}(f(n)) \subseteq \text{TIEMPO}(k^{\log(n)+f(n)})$

6 Reconocer Palindromos

Se hace con una Maquina de Turing con las siguientes estructura de cintas:

1	2	3	3	2	1	□	Entrada
1	0	1	□	□	Posición que se está comprobando (binario)		
1	1	□	□	□	Contador en binario para encontrar posiciones		

7 Búsqueda de existencia de caminos (Espacio Determinista y No Determinista) y su Complejidad

Enunciado: Tenemos un grafo dirigido G y dos nodos x_i y x_j. ¿Existe un camino entre dos nodos?.

Los algoritmos de búsqueda en profundidad o en anchura son del orden $O(n^2)$ en tiempo y del orden $O(n)$ en espacio, donde n es el número de nodos.

7.1 Espacio Determinista:

Teorema de Savitch: Existencia de Caminos. La complejidad en espacio de existencia de caminos en grafos dirigidos es del orden de $O(\log^2(n))$, donde n es el numero de nodos del grafo.

La demostración se basa en una resolución ordenada del problema $CAMINO(x, y, i)$: Existencia de un camino de longitud $\leq 2^i$, y la relación

$$CAMINO(x, y, i) \iff \exists z, CAMINO(x, z, i-1) \qquad CAMINO(z, y, i-1)$$

Se van colocando las tripletas necesarias para resolver el problema de forma recursiva en una cinta. En cada llamada recursiva solo hay que colocar un tripleta. Para saber si estamos en la primera o segunda tripleta, solo hay que comprobar la tripleta anterior (si el x y el y son iguales a los dos del

nivel anterior no es necesario saber si es la primera o la segunda llamada, se puede proceder como si fuese la segunda llamada siempre).

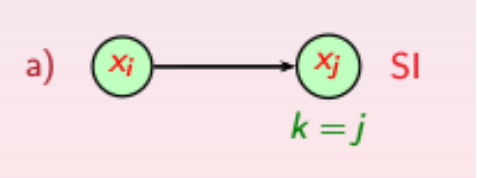
Cada tripleta es de longitud $O(log(n))$ y el numero de tripletas es de orden $O(log(n))$

7.2 Espacio No Determinista

En espacio no-determinista el problema se resuelve en espacio $O(log(n))$ donde n es el número de vértices, y la misma complejidad.

Supongamos que queremos determinar si existe un camino entre el nodo x_i y el nodo x_j.
Escribimos un numero $m = n$ (el numero de nodos).
Si $m = 0$ solo acepta si $x_i = x_k$. En otro caso:
El algoritmo funciona escribiendo el identificador de un nodo x_k de forma no-determinista.

a) Si existe un arco de x_i a x_k y x_k es igual a x_j, entonces para y acepta.



b) Si existe el arco y x_k es distinto de x_j, vuelve a ejecutar el algoritmo con x_k en el lugar de x_i y m decrementado en uno.

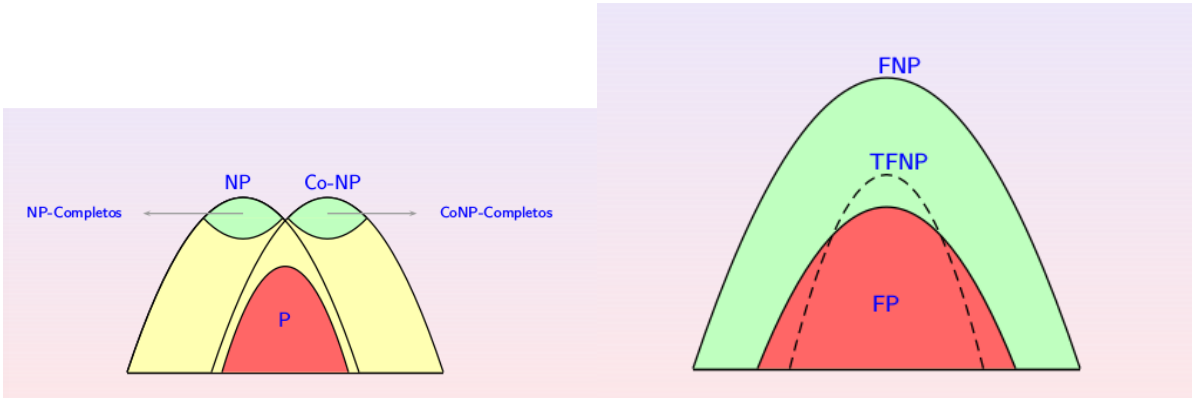
c) Si no existe el arco de x_i a x_k para con rechazo

8 Clases de Complejidades (NP, NP-Completo, FNP, ...)

Un problema A es NP-completo si y solo si existe un problema B NP-Completo que se puede reducir a A.

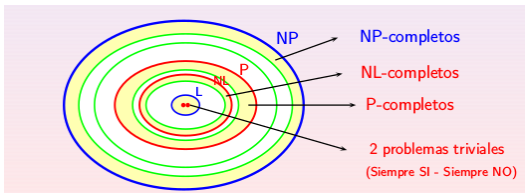
- **Clase Co-NP:** Complementarios de los problemas de NP.

- **Clase FNP:** es la clase de complejidad que extiende la clase NP (la cual incluye exclusivamente problemas de decisión), hacia problemas computacionales de tipo funcional, es decir, aquellos que obtienen como salidas valores distintos de SÍ o NO.
- **Clase FP:** corresponde al conjunto de problemas funcionales que pueden ser resueltos por una Máquina de Turing determinista en tiempo polinomial, lo cual en la práctica significa que incluye a todos los problemas que pueden ser resueltos eficientemente en los computadores clásicos, sin necesidad de aleatoriedad.
- **Clase FNPT:** Clase de problemas de FNP totales. Ejemplos: Red Feliz: Dado un grafo con pesos, un nodo es feliz si $\sum_{(i,j)\in E} s(j).w(i,j) \geq 0$ y el problema de, dado un numero p, este puede descomponerse en números primos.



Teoremas:

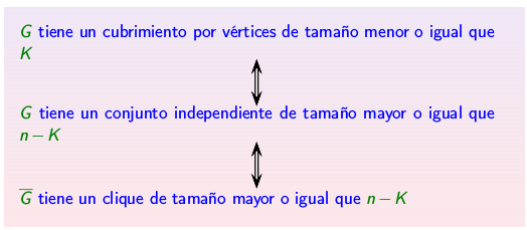
- L es NP-Completo $\iff$ L-complementario es CoNP-Completo.
- Si un problema CoNP-completo esta en NP, entonces CoNP = NP
- Si P = NP, entonces CoNP = NP



9 Equivalencia de Problemas NP-Completo

Respuesta a la pregunta de si Clique máximo, Cubrimiento por Vertices y Conjunto Independiente basta con demostrar que uno solo es NP-Completo, los demás también lo son:

Los tres problemas son equivalentes. Si n es el numero de nodos de G, entonces:



10 Complejidades NAESAT MAXSAT 3-SAT ...

Respuestas a la pregunta *Define los siguientes problemas y especifica sobre su complejidad:*

- **Flujo Máximo:** Este problema es de una complejidad total de $O(n^5)$, es decir, polinomico (**P**). El problema consiste en obtener el valor del flujo máximo que puede salir del origen, por ejemplo cantidad de información que se puede transportar por una red.
- **Reconocer palindromo:** Este problema es de una complejidad $O(log(n))$, es decir, complejidad logarítmica (**L**). El problema consiste en determinar si dada una entrada, esta es un palindromo.
- **2-SAT:** Este problema esta en NL y, por tanto, en **P**. El problema consiste en saber si, dada una expresión booleana con variables y sin cuantificadores, hay alguna asignación de valores para sus variables que hace que la expresión sea verdadera cuando el número máximo de literales por cláusula es 2.
- **MAX2SAT:** Este problema es **NP-Completo**. Tenemos un conjunto de clausulas con dos literales y un valor $K \geq 0$.¿Pueden satisfacerse, al menos, K clausulas?(Generalización de 2-SAT). *Reducción desde 3-SAT*

- **NAESAT:** Este problema es **NP-Completo**. Dado un conjunto de clausulas de longitud 3, determinar si existe una asignación de valores de verdad tal que para cada clausula, alguno, pero no todos los literales sean ciertos. *Reducción desde 3-SAT*

- **Cubrimiento por Vértices:** Complejidad del problema: **NP-Completo**. Dado un grafo $G = (V, E)$ y un numero natura $K \leq |V|$, determinar si existe un cubrimiento de vértices de tamaño menor o igual que K. *Usamos 3-SAT para reducirlo*

- **Problema de la Partición:** Complejidad del problema: **NP-Completo**. Dado un conjunto A y un tamaño para cada uno de sus elementos. Determinar si existe un $A' \subseteq A$ tal que se verifique: $\sum_{a \in A'} s(a) = \sum_{a \in A\ A'} s(a)$. *Se Reduce desde ACTRI.*

- **Circuito Hamiltoniano:** Complejidad del problema: **NP-Completo**. Dado un grafo no dirigido. ¿Existe un circuito Hamiltoniano?.*Reducción desde Cubrimiento de Vértices.*

- **Problema de la mochila:** Complejidad del problema: **NP-Completo**. Dado un conjunto finito de objetos, cada objeto tiene asignado un peso y un valor.¿Existe un valor cuya suma de valores sea mayor a un mínimo y el peso no supere peso máximo?. *Reducción desde el problema de la partición, ya que este es un caso particular de este problema.*

- **Acoplamiento por tripletas:** Complejidad del problema: **NP-Completo**. Dado tres conjuntos disjuntos del mismo tamaño y un conjunto M de compatibles.¿Contiene M un subconjunto $M' \subseteq M$ con q elementos , tal que para cada tripleta $\in M'$, no hay ninguna tripleta igual ni sus componentes son iguales?.*Reducción desde 3-SAT.*

- **Isomorfismos de Grafos:** Complejidad del problema es desconocido, se ha definido una nueva complejidad para problemas que se reducen desde este problema llamado **GI-Completo**. Dados dos grafos no dirigidos $G_1 = (V_1, E_1)$ y $G_2 = (V_2, E_2)$, determinar si existe un isomorfismo de uno a otro, es decir, una aplicación biyectiva de $f : V_1 \to V_2$, tal que $(u, v) \in E_1 \iff (f(u), f(v)) \in E_2$. *Sin Reducción.*

- **Viajante de comercio:** Complejidad del problema: **NP-Completo**. Dado un conjunto finito de ciudades y una cota B. ¿Existe un circuito que visite todas las ciudades una sola vez y de coste no superior a B?. *Reducción desde el circuito Hamiltoniano*

- **Clique Máximo:** Complejidad del problema: **NP-Completo**.Dado un grafo $G = (V, E)$ y un número natural $J \leq |V|$, determinar si existe un clique de tamaño mayor o igual que J.

- **Conjunto Independiente:** Complejidad del problema: **NP-Completo**. Dado un grafo $G = (V, E)$ y un numero natural $J \leq |V|$, determinar si existe un conjunto independiente de tamaño mayor o igual que J.