Resumen de Algoritmos y Estructuras de Datos III

Martín Arjovsky

Febrero 2013

1 Técnicas Algorítmicas

- Principio de optimalidad: un problema de optimizacón satisface el principio de optimalidad de Bellman si en una sucesión óptima de decisiones o elecciones, cada subsucesión es a su vez óptima. Es decir, si miramos una subsolución de la solución óptima, debe ser solución del subproblema asociado a esa subsolución. Es condición necesaria para poder usar la técnica de programación dinámica.

2 Grafos

Theorem 1.

$$\sum_{i=1}^{n} d(v_i) = 2m$$

Proof. Inducción en m
$m = 1$ trivial
$m > 1$ sacarle una arista, usar HI y listo. $\square$

Theorem 2. *Un grafo G con 2 o más nodos es bipartito si y sólo si no tiene circuitos simples de longitud impar.*

Proof. $\Rightarrow$
$G = (V, X)$ es bipartito, por ende $V = V_1 \cup V_2$ y $V_1 \cap V_2 = \emptyset$. Sea $v_1, ..., v_n, v_1$ un circuito impar ($n \equiv 1 (\mathrm{mod}\ 2)$), entonces sin perdida de generalidad $v_1 \in V_1$, y por inducción en i, $v_i \in V_1 \Leftrightarrow i \equiv 1 (\mathrm{mod}\ 2)$. Luego, $v_n \in V_1 \Rightarrow v_1 \notin V_1 \Rightarrow$ Abs!
$\Leftarrow$
Si G es conexo, agarro un vértice cualquiera v y lo pongo en V_1, a todos sus vecinos en V_2, a todos los vecinos de estos en V_1 y así. Como el grafo es conexo y finito, tiene diametro finito y este proceso termina. Es facil ver que si $\exists w \in V_1$ y $w \in V_2$ entonces forma parte de un ciclo impar, lo que es absurdo.

Si G es disconexo, con C componentes conexas aplico el mismo proceso a cada componente conexa y hago $V_1 = \bigcup_{i=1}^{c} V_{1_i}$ y $V_2 = \bigcup_{i=1}^{c} V_{2_i}$ □

3 Árboles

Lemma 1. *Sea $G = (V, X)$ un grafo conexo y $e \in X$. $G \setminus e$ es conexo si y solo si e pertenece a un circuito simple de G*

Proof. $\Rightarrow$
Sea $e = (v, w)$ y $G \setminus e$ es conexo, por lo que en $G \setminus e$ hay un camino simple $e_1, ..., e_n$ entre v y w con $e \neq e_i \forall i$, luego $e_1, ..., e_n, e$ es un circuito simple que contiene a e.
$\Leftarrow$
Sea $e = (v, w)$, tal que existe un circuito simple $e, e_1, ..., e_n$ en G. Sean $x, y \in V$. Como G es conexo, existe un camino simple $e'_1, ..., e'_m$ entre x e y. Si $e'_i \neq e \forall i$ entonces el camino existe en $G \setminus e$. Si, por el contrario, $e = e'_i$ para algún i, entonces $e'_1, ..., e'_{i-1}, e_1, ..., e_n, e'_{i+1}, ..., e'_m$ o bien $e'_1, ..., e'_{i-1}, e_n, ..., e_1, e'_{i+1}, ..., e'_m$ es un camino entre x e y en $G \setminus e$. Luego, $\forall x, y \in V$ hay un camino que los une en $G \setminus e$, por lo que $G \setminus e$ es conexo. □

Lemma 2. *La concatenación de dos caminos distintos entre un par de vértices contiene un circuito simple.*

Proof. Sean $e_1, ..., e_n$ y $e'_1, ..., e'_m$ caminos distintos de v a w, luego la concatenación $e_1, ..., e_n, e'_1, ..., e'_m$ es un circuito. Por inducción (o mínimo contraejemplo) se prueba que todo circuito contiene un circuito simple. □

Theorem 3. *Dado un grafo $G = (V, X)$, son equivalentes:*

1. *G es un árbol.*
2. *G es un grafo sin circuitos simples, pero si se agrega una arista e a G resulta un grafo con exactamente un circuito simple que pasa por e.*
3. *Existe exactamente un camino simple entre todo par de nodos.*
4. *G es conexo, pero si se quita cualquier arista a G queda un grafo no conexo.*

Proof. $1 \Rightarrow 2$
$G = G \cup e \setminus e$ con $e = (v, w)$ es conexo y sin circuitos simples, por lo que usando el Lema 1, $G \cup e$ es conexo y e pertenece a un circuito simple de $G \cup e$. Si e perteneciera a dos o más circuito en $G \cup e$, entonces habría dos caminos distintos, de v a w que no usan e (por lo que estan contenidos en G) y usando el Lema 2, su concatenación, que está contenida en G, contendría un ciclo, lo que es absurdo.
$2 \Rightarrow 3$

Sean $v, w \in V$ un par de nodos cualquiera. Si hay mas de un camino simple entre v y w entonces su concatenación existe en G y contiene un circuito simple, lo que es absurdo. Si no hay ningún camino, entonces $e = (v, w) \notin G$ y $G \cup e$ tiene exactamente un circuito simple que pasa por e $e, e_1, ..., e_n$, pero luego $e_1, ..., e_n$ es un camino entre v y w, lo que es absurdo. Luego, para todo par de nodos v, w hay exactamente un camino entre ellos.
$3 \Rightarrow 4$
Hay un camino simple entre todo par de nodos, por lo que G es conexo. Si se saca una arista $e = (v, w)$ a G y sigue habiendo un camino $e_1, ..., e_n$ entre v y w entonces este y e son dos caminos distintos entre v y w en G, lo que es absurdo.
$4 \Rightarrow 1$
G es conexo. Supongamos que existe un ciclo $e, e_1 ..., e_n$ en G, con $e = (v, w)$. Luego, si sacamos e queda un grafo disconexo, por lo que hay dos nodos x e y tal que ahora no hay ningun camino y antes sí. Sea el camino en G entre x e y $e'_1, ..., e'_m$ entonces tiene que haber un i tal que $e'_i = e$ ya que sino este camino esta en $G \setminus e$. Pero de esta manera sigue existiendo el camino $e'_1, ..., e'_{i-1}, e_1, ..., e_n, e'_{i+1}, ..., e'_m$ o $e'_1, ..., e'_{i-1}, e_n, ..., e_1, e'_{i+1}, ..., e'_m$ entre x e y en $G \setminus e$, lo que es absurdo. Luego, G es conexo y sin ciclos, por lo que es un arbol. □

Lemma 3. *Todo árbol no trivial tiene al menos dos hojas.*

Proof. Inducción en n
$n = 2$ trivial
$n > 2$ sacarle una hoja, usar HI y listo. □

Lemma 4. *Sea $G = (V, X)$ arbol, entonces $m = n - 1$.*

Proof. Inducción en n
$n = 1$ trivial
$n > 1$ sacarle una hoja, usar HI y listo. □

Corolary 1. *Sea $G = (V, X)$ sin circuitos simples y c componentes conexas, entonces $m = n - c$*

Proof. $m = |X| = |\dot{\bigcup}_{i=1}^{c} X_i| = \sum_{i=1}^{c} |X_i| = \sum_{i=1}^{c} (|V_i| - 1) = \sum_{i=1}^{c} |V_i| - c =$
$|\dot{\bigcup}_{i=1}^{c} V_i| - c = |V| - c = n - c$. En la cuarta igualdad se usa el Lema 4. □

Corolary 2. *Sea $G = (V, X)$ con c componentes conexas, entonces $m \geq n - c$ y la desigualdad es estricta si hay ciclos.*

Proof. Sacarle aristas de los ciclos hasta que quede un arbol y usar el Corolario 1. □

Theorem 4. *Dado un grafo $G = (V, X)$, son equivalentes:*

1. *G es un árbol.*

2. G *es un grafo sin circuitos simples y* $m = n - 1$

3. G *es conexo y* $m = n - 1$

Proof. $1 \Rightarrow 2$
G es conexo por definición de árbol y $m = n - 1$ por el Lema 4.
$2 \Rightarrow 3$
Usando el Corolario 1, $m = n - c$ y $m = n - 1$ por hipótesis. Luego, $c = 1$, por lo que G es conexo.
$3 \Rightarrow 1$
Por el Corolario 2, si hay ciclos entonces $m > n - c$ y $c = 1$ por hipótesis. Luego, pasaría que $m > n - 1$ y $m = n - 1$, lo que es absurdo, por lo que no hay ciclos en G. Además, G es conexo por hipótesis, por lo que es un árbol. □

Lemma 5. *Sea* $T = (V, X_T)$ *un árbol generador de* $G = (V, X)$*. Si* $e \in X$ *y* $e \notin X_T$ *y* $f \in X_T$ *es una arista del ciclo de* $T \cup e$*, entonces* $T' = (V, X_T \cup \{e\} \setminus \{f\}$ *es un arbol generador de* G*.*

Proof. Como T' tiene el mismo conjunto de nodos que G, solo hace falta probar que es un árbol. Además, $m_{T'} = m_T + 1 - 1 = m_T = n - 1$, por lo que solo falta probar que T' es conexo. Sean x, y dos nodos de G cualesquiera y $f = (v, w)$. Como T es un árbol, es conexo. Sea $e_1, ..., e_n$ el camino en T que une a x y a y. Si $f \neq e_i \forall i$ entonces este camino sigue existiendo en T', sino $e_i = f$. Sea $f, e'_1, ..., e, ..., e'_m$ el ciclo generado al agregar e, entonces $e_1, ..., e_{i-1}, e'_1, ..., e, ..., e'_m, e_{i+1}, ..., e_n$ o $e_1, ..., e_{i-1}, e'_m, ..., e, ..., e'_1, e_{i+1}, ..., e_n$ es un camino entre x e y en T', por lo que T' es conexo y $m_T = n - 1$, lo que implica que T' es un árbol generador de G. □

Theorem 5. *Sea* $G = (V, X)$ *un grafo conexo. Sea* $T_k = (V_k, X_k)$ *el árbol que el algoritmo de Prim determina en la iteración* k*, entonces* T_k *es un subgrafo de un arbol generador de* G*.*

Proof. Inducción en k
$k = 1$ trivial
$k > 1$
Por hipótesis inductiva, supongamos que T_k es subgrafo de un árbol generador T. En la iteración $k + 1$ el algoritmo añade la arista $e = (v, w)$, que por el algoritmo es la arista tal que tiene un extremo en V_k y otro en $X_k \setminus V_k$ (v y w respectivamente) de distancia mínima. Supongamos que $e \notin T$. Sin embargo, como T es un árbol, $T \cup e$ tiene que tener un circuito simple que pase por v y w. Luego, en $T \cup e$ hay dos caminos entre v y w, uno es e y el otro $e_1, ..., e_n$, con $e_i \neq e \forall i$. Como $e_1, ..., e_n$ empieza en V_k y termina en $X_k \setminus V_k$ tiene que haber una arista e_i que tenga un extremo en V_k y otro en $X_k \setminus V_k$, por lo que $|e| \leq |e_i|$. Además, por el lema, $T \cup e \setminus e_i$ es un arbol generador, y tiene tamaño menor o igual a T, por lo que es un árbol generador mínimo. Finalmente, T_{k+1} es subgrafo de $T \cup e \setminus e_i$ ya que tiene a e y todas las otras aristas, que por la hipótesis inductiva estaban en T y son todas distintas de e_i. Luego, T_{k+1} es subgrafo de un arbol generador mínimo. □

Corolary 3. *El algoritmo de Prim es correcto.*

Theorem 6. *El algoritmo de Kruskal es correcto.*

Proof. Por inducción en k, T_k trivialmente no tiene ciclos y tiene $k-1$ aristas. El algoritmo termina porque, como el grafo es conexo, si hay un vértice v que no esta en V_k entonces hay una arista con un extremo en v que no forma ciclo. $V_k = V$ y el algoritmo todavía no terminó, entonces T_k todavía no es conexo ya que $m < n-1$ y T_k es acílico, por lo que hay una arista que conecta dos componentes conexas, y por ende no puede formar ciclo. Como cuando termina, luego de la iteración $n-1$, T tiene $n-1$ aristas, $m_T = n-1$ y es acíclico, lo que implica que es un árbol. Además, como tiene la misma cantidad de nodos que G y es subgrafo de G, es un árbol generador. Para probar que T es árbol generador mínimo se procede igual que en la demostración del Teorema 5, con inducción en k con la hipótesis inductiva de que T_k es subgrafo de un árbol generador mínimo. □

4 Camino Mínimo

Theorem 7. *Dado un grafo orientado G con pesos positivos en las aristas, al finalizar la iteración k el algoritmo de Dijkstra determina el camino mínimo entre el nodo v y los nodos de S_k (donde S_k es el conjunto S al finalizar la iteración k*

Proof. Inducción en k
$k = 1$ trivial
$k > 1$
Sea w el nodo agregado a S_k en la iteración $k+1$, entonces w no está en S_k. Obviamente, por definición del algoritmo, w tiene un padre $z \in S_k$. Sea C otro camino entre v y w, $v = v_0, v_1, ..., v_n = w$. Como este camino empieza en S_k y termina afuera de S_k, $\exists i$ tal que $v_i \in S_k \wedge v_{i+1} \notin S_k$. Llamamos a estos dos nodos x e y respectivamente. Luego, $|C_1| \geq d(v,x) + e(x,y) + d(y,w) = \pi_k(v,x) + e(x,y) + d(y,w) \geq \pi_k(v,x) + e(x,y) \geq \pi_k(y) \geq \pi_k(w)$. En la primera desigualdad partimos a C en tramos $(v,x),(x,y),(y,w)$ y por definición de d el primer y el último tramo tienen tamaño mayor o igual a d. En la igualdad siguiente usamos la hipótesis inductiva y el hecho de que $x \in S_k$. En la desigualdad siguiente usamos que los pesos en las aristas son no negativos, por lo que $d(y,w) \geq 0$. En el siguiente paso usamos que $x \in S_k$, que y es adyacente a x y la definición de π_k. Finalmente, en el último paso se usó que si $\pi_k(y) < \pi_k(w)$ entonces el algoritmo hubiera elegido a y y no a w. Es trivial probar por inducción en k que el algoritmo devuelve un camino válido entre v y w. □

Corolary 4. *El algoritmo de Dijkstra es correcto*

Theorem 8. *El algoritmo de Bellman-Ford es correcto*

Proof. Trivial, hacer inducción en la iteración k para la minimalidad del camino. Usar que los caminos mínimos tienen como máximo n nodos, por lo que la iteración $n-1$ es la última si y solo si no hay ciclos negativos. $\square$

Theorem 9. *En la iteración k, el algoritmo de Floyd, entre todo par de nodos v y w el camino de longitud mínima dentro de los que usan solo los nodos $v_1, ..., v_k$ como nodos intermedios*

Proof. Trivial, inducción en k $\square$

Corolary 5. *El algoritmo de Floyd es correcto*

Theorem 10. *En la iteración k, el algoritmo de Dantzig, computa los caminos mínimos entre todo par de nodos del subgrafo inducido por los nodos $v_1, ..., v_k$.*

Proof. Trivial, inducción en k $\square$

Corolary 6. *El algoritmo de Dantzig es correcto*

5 Grafos Eulerianos y Hamiltonianos - Heurísticas

Theorem 11. *El algoritmo para encontrar circuitos eulerianos devuelve un ciclo euleriano de $G = (V, X)$ si y solo si G es conexo y todos sus nodos tienen grado par.*

Proof. El algoritmo en un nodo v_0, y agrega aristas a la pila, removiendolas de G hasta que no hay mas aristas en el nodo actual. En ese momento en la pila hay un camino simple $e_1 ...., e_n$ con v, como nodo final, por lo que v no tiene más aristas. Pero v tiene dos aristas por cada vez que aparece en el medio del camino y una por cada extremo en el que aparece, por lo que si $v \neq v_0$ entonces $d(v) = 1 + \sum_{j=1}^{l} 2 \equiv 1 (\text{mod } 2)$, lo que es absurdo. Luego, hay un ciclo en la pila. Tras esto el algoritmo vuelve hasta el primer nodo w empezando desde el tope de la pila que tenga aristas libres. Siguiendo el mismo argumento, encuentra un ciclo que empiece y termine en v_i y así sucesivamente. Como el grafo es conexo, toda arista que no haya sido recorrida tiene que poder ser accesible desde v_i, y entonces va a ser recorrida antes de que v_i se quede sin aristas. Luego, el circuito devuelto cubre todas las aristas exáctamente una vez (ya que se aniquilan al ponerlas en la cola. Por ende, el circuito devuelto es euleriano. La ida es trivial, ya que si devuelve un circuito euleriano $v_1, ..., v_n, v_1$ entonces G es obviamente conexo, todos los nodos internos tienen dos aristas incidentes por aparición (grado par) y v_1 una por cada extremo más dos por las interiores (también grado par). $\square$

Theorem 12. *Un digrafo tiene un circuito euleriano si y solo si para todo nodo v, $d_{in}(v) = d_{out}(v)$.*

Proof. Idéntica a la anterior. $\square$

Corolary 7. *Un grafo tiene un camino euleriano si y solo si todos sus nodos salvo dos tienen grado par.*

Proof. Sean v y w los nodos con grado impar, luego $G \cup e$ con $e = (v, w)$ tiene todos sus nodos con grado par, por lo que hay un circuito euleriano $e_1, ..., e_m, e$. Luego, $e_1, ..., e_m$ es un camino euleriano en G. □

Theorem 13. *Sea $G = (V, X)$ un grafo conexo. Si existe $W \subset G$ tal que $G \setminus W$ tiene c componentes conexas con $c > |W|$ entonces G no es hamiltoniano.*

Proof. Primero es importante notar que, por definición de componente conexa, en todo camino que vaya de una componente conexa a otra, tiene que haber un nodo de W. Supongamos que hay un ciclo hamiltoniano $v_1, ..., v_n, v_1$ Luego, existen en el ciclo $v_{i_1}, ..., v_{i_c}$ que pertenecen a distintas componentes conexas con $v_{i_1} = v_1$. Luego, entre cada uno de ellos hay un $w_i \in W$ distinto ($c - 1$ en total) entre cada uno de ellos. Pero además, la componente conexa de v_{i_c} es distinta que la de v_1, por lo que entre estos dos (al final del ciclo) hay otro w_i distinto. Pero entonces tiene que haber por lo menos $c - 1 + 1 = c$ miembros en W, lo que es absurdo. □

Theorem 14. *Sea $G = (V, X)$ un grafo tal que $n \geq 3$ y $d(v) \geq \frac{n}{2} \forall v \in V$. Entonces el grafo es hamiltoniano.*

Proof. Sea $C = (v_1, ..., v_k)$ el camino más largo. Si v_1 es adyacente a un vértice v que no está en el camino entonces $v, v_1, ..., v_k$ es un camino más largo, lo que es absurdo. Lo mismo para v_n, por lo que todos sus vecinos están en el camino. Veamos que existe $i < n$ tal que v_i es adyacente a v_k y v_{i+1} lo es a v_1. Supongamos que esto es falso, pero entonces v_k tiene al menos $\frac{n}{2}$ nodos adyacentes en C y ninguno de los que les sigue es adyacente a v_1. Por ende, si $A = \{v_{i+1}/v_i \text{ es adyacente a } v_k\}$ entonces $v_1 \uplus A \uplus n(v_1) \subseteq V$ pero $|v_1 \uplus A \uplus n(v_1)| = 1 + |A| + |n(v_1)| = 1 + |n(v_k)| + |n(v_1)| \geq 1 + \frac{n}{2} + \frac{n}{2} = 1 + n > n = |V|$, lo que es absurdo. Por ende, hay un i tal que v_i es adyacente a v_k y v_{i+1} lo es a v_1. Esto nos da un ciclo $C' = v_1, ..., v_i, v_k, ..., v_{i+1}$. Supongamos que $G \setminus C'$ es no vacío, entonces hay un $v \in G \setminus C'$, pero esto implica que hay algún $v_j \in C'$ tal que v es adyacente a v_j ya que $d(v) \geq \frac{n}{2}$ y $|C'| \geq \frac{n}{2}$. Luego, $v, v_j, v_{j+1}, ..., v_{j-1}$ es un camino más largo que C, lo que es imposible. Por ende, $G \setminus C' = \varnothing$, lo que implica que C' es un ciclo hamiltoniano. □

6 Planaridad

Theorem 15. *Si un grafo contiene un grafo no planar, es no planar.*

Proof. El contrarrecíproco es trivial, si un grafo es planar, un subgrafo de él va a ser planar porque al sacarle aristas y/o nodos en la representación planar original, las aristas no se van a intersecar y queda una representación planar del subgrafo. □

Theorem 16. *Si G es un grafo conexo planar entonces cualquier representación planar de G determina $f = m - n + 2$ regiones en el plano.*

Proof. Inducción en m, trivial. □

Theorem 17. *Si G es conexo, planar, con $n \geq 3$ entonces $m \leq 3n - 6$.*

Proof. Cada región está encerrada por al menos tres aristas del plano, y cada arista es adyacente a lo sumo a dos regiones. Por ende, $f \geq \frac{2}{3}m$. Luego, usando el Teorema 16 tenemos $m - n + 2 = f \leq \frac{2}{3}m \Rightarrow \frac{1}{3}m \leq n - 2 \Rightarrow m \leq 3n - 6$ □

Corolary 8. *K_5 es no planar.*

Theorem 18. *Si G es conexo, planar, con $n \geq 3$ y libre de triángulos entonces $m \leq 2n - 2$.*

Proof. Cada región está encerrada por al menos cuatro aristas del plano, y cada arista es adyacente a lo sumo a dos regiones. Por ende, $f \geq \frac{2}{4}m = \frac{1}{2}m$. Luego, usando el Teorema 16 tenemos $m - n + 2 = f \leq \frac{1}{2}m \Rightarrow \frac{1}{2}m \leq n - 2 \Rightarrow m \leq 2n - 4$ □

Corolary 9. *$K_{3,3}$ es no planar.*

7 Coloreo

Lemma 6. *Si H es un subgrafo de G entonces $\chi(G) \geq \chi(H)$*

Proof. Trivial, todo coloreo en G tiene que ser válido en H y el subcoloreo no puede tener más colores. □

Corolary 10. $\chi(G) \geq \omega(G)$

Theorem 19. $\chi(G) \leq \Delta(G) + 1$

Proof. Cualquier coloreo greedy devuelve a lo sumo $\Delta(G) + 1)$ □

Theorem 20. *Si G es conexo y no es un circuito ni es completo entonces $\chi(G) \leq \Delta(G)$*

Theorem 21. *Si $G = (V, X)$ es planar, $\chi(G) \leq 5$*

Proof. Primero suponemos que $d(v) \geq 6 \forall v \in V$. Pero por el teorema 17 $m \leq 3n - 6 \Rightarrow 2m \leq 6n - 12 \Rightarrow \sum_{i=1}^{n} d(v_i) \leq 6n - 12 \Rightarrow 6n \leq 6n - 12 \Rightarrow$ *Abs*!. Luego, podemos suponer que existe un vértice de grado a lo sumo 5 que llamamos x. Si removemos x, por inducción $G \setminus x$ queda 5-coloreable. Si entre sus nodos adyacentes no se usan los 5 colores, entonces ponemos a x uno de los colores sobrantes y ya estamos. Por ende, consideramos el caso en que x tiene 5 vecinos y_1, y_2, y_3, y_4, y_5 en sentido horario en alguna representación planar de G, con colores $1, 2, 3, 4, 5$ respectivamente. Consideramos ahora el subgrafo inducido

$G_{1,3}$ compuesto por los nodos coloreados con los colores $1, 3$. Si en este grafo no hay un camino entre y_1, y_3 entonces invertimos el coloreo de la componente conexa en $G_{1,3}$ de y_1, cambiando el color de los nodos de 1 a 3 y de 3 a 1. Luego, x no tiene un nodo adyacente con el color 1 por lo que podemos usar este color en él, 5-coloreando G. Análogamente, podemos hacer lo mismo con y_2 e y_4, por lo que el caso restante es en el que existe un camino con colores 1 y 3 entre y_1, y_3 y uno con colores 2 y 4 entre y_2 e y_4. Pero estos caminos forman ciclos con x y porque los y_i están en sentido horario, estos ciclos se cruzan. Finalmente, como estos ciclos tienen caminos distintos no pueden tener nodos en común, por lo que se cruzan las aristas, haciendo que la representación elegida no sea planar, lo que es absurdo por como fue elegida. ☐

Theorem 22. $\Delta(G) \leq \chi'(G) \leq \Delta(G) + 1$

8 Matching

Lemma 7. *$S \subseteq V$ es un conjunto independiente si y solo si $V \setminus S$ es un cubrimiento de aristas.*

Proof. $\Rightarrow$
Si $V \setminus S$ no es un cubrimiento de aristas, entonces existe $e = (v, w)$ una arista tal que $v, w \notin V \setminus S$ por lo que $v, w \in S$ y comparten una arista, lo que implica que S no es independiente, absurdo.
$\Leftarrow$
Si S no es independiente, entonces existe $e = (v, w)$ una arista tal que $v, w \in S$ por lo que $v, w \notin V \setminus S$ y entonces e no esta cubierta por $V \setminus S$, lo que es absurdo. ☐

Lemma 8. *Sean M_0 y M_1 dos matchings en $G = (V, X)$ y sea $G' = (V, X')$ con $X' = M_0 \Delta M_1$. Entonces las componentes conexas de G' son de alguno de los siguientes tipos:*

- *Nodo aislado.*
- *Circuito simple con aristas alternadas entre M_0 y M_1*
- *Camino simple con aristas alternadas entre M_0 y M_1*

Proof. Sea C una componente conexa de G. Si hay un nodo$v \in C$ y $d(v) \geq 3$ entonces hay tres aristas que comparten a v como extremo e_1, e_2, e_3. Pero esto es imposible, ya que tienen que estar repartidas entre M_0 y M_1 por lo que dos de ellas tendrían que estar en el mismo matching y compartir una arista, lo que es absurdo. Luego, C es conexo y tiene sus nodos con grado menor o igual a 2. Esto implica trivialmente por inducción que C es un ciclo, un camino o un nodo aislado. Por inducción también se prueba trivialmente que si C es un ciclo o camino tiene sus aristas alternadas. ☐

Theorem 23. *M es un matching máximo si y solo si no hay un camino de aumento en G con respecto a M.*

Proof. $\Rightarrow$
Sea M máximo y C un camino de aumento, entonces $M' = M \setminus (C \cap M) \cup (C \cap M^c)$ es un matching con $|M'| = M + 1$, lo que es imposible.
$\Leftarrow$
Probemos el contrarrecíproco. M no es máximo, por lo que existe M' un matching con $|M'| > |M|$. Sea $G' = (V, X')$ con $X' = M \Delta M'$. Luego, las componentes conexas de G' son descriptas como en el Lema 8. Como $|M'| > |M|$ hay una componente conexa $C \subseteq G'$ en donde $|C \cap M'| > |C \cap M|$ Como tiene aristas, C no es un nodo aislado, y para ser un ciclo con aristas alternadas tendría que tener la misma cantidad de aristas de M' y de M, por lo que es un camino con aristas alternadas. Luego, como tiene más aristas de M' que de M, la arista inicial y la final tienen que ser de M', lo que hace que sea un camino de aumento para M. □

Theorem 24. *Dado un grafo G sin nodos aislados, si M es un matching máximo de G y R_e un recubrimiento mínimo de los nodos de G, entonces $|M| + |R_e| = n$.*

Proof. Sea M un matching máximo, y S el conjunto de nodos no cubiertos por M. Obviamente, S es un conjunto independiente, ya que sino, existe una arista $e = (v, w)$ con $v, w \in S$ y $M \cup e$ es un matching mas grande que M, lo que es absurdo. Como no hay nodos aislados, cada nodo $v_i \in S$ tiene una arista e_i con borde en v_i. Luego, $|R| = M \cup \bigcup_{i=1}^{|S|} e_i$ es un cubrimiento de nodos y $|R_e| \leq |R| \leq |M| + |S| = |M| + (n - 2|M|) = n - |M| \Rightarrow |R_s| + |M| \leq n$
Análogamente, sea R_e un cubrimiento de nodos mínimo. Entonces, R_e no puede tener caminos de tamaño mayor o igual a 3 e_1, e_2, e_3 ya que entonce $R_e \setminus e_2$ sería un cubrimiento de nodos más chico que R_e. Luego, $R_e = \dot{\bigcup}_{i=1}^{l} S_i$, donde S_i son las aristas de estrellas con n_i nodos cada una. Luego, $|R_e| = \sum_{i=1}^{l} |S_i| = \sum_{i=1}^{l} (n_i - 1) = n - l$ ya que las estrellas son árboles y $\sum_{i=1}^{l} n_i = n$ porque las S_i cubren nodos. Notar que se puede construir un matching M' de tamaño l tomando una arista de cada estrella, por lo que $|R_e| + |M| \geq |R_e| + |M'| = n - l + l = n$. Luego, tenemos que $|R_e| + |M| \leq n$ y $|R_e| + |M| \geq n$, lo que implica que $|R_e| + |M| = n$. □

Theorem 25. *Dado un grafo G sin nodos aislados, si I es un conjunto independiente máximo de G y R_n un recubrimiento mínimo de las aristas de G, entonces $|I| + |R_n| = n$.*

Proof. Usando el Lema 7, si I es máximo entonces $V \setminus I$ es un cubrimiento de aristas mínimo, por lo que $|R_n| = |V \setminus I|$. Entonces $|I| + |R_n| = |I| + |V \setminus I| = |V| = n$. □

9 Flujo en Redes

Theorem 26. *Sea f un flujo determinado en una red $N = (V, X)$ y sea S un corte, entonces*

$$F = \sum_{e \in S\overline{S}} f(e) - \sum_{e \in \overline{S}S} f(e)$$

Proof. Haciendo inducción top-down en la cantidad de nodos en el corte.
Caso base, $S = V \setminus \{t\}$
En este caso, $S\overline{S} = In(t)$ y $\overline{S}S = Out(t)$, por lo que

$$\sum_{e \in S\overline{S}} f(e) - \sum_{e \in \overline{S}S} f(e) = \sum_{e \in In(t)} f(e) - \sum_{e \in Out(t)} f(e) = F$$

En el caso general, sea $S \neq V \setminus \{t\}$ un corte. Sea $x \notin S$ con $x \neq t$ y $S' = S \cup \{x\}$, entonces $S'\overline{S'} = S\overline{S} \setminus S\{x\} \cup \{x\}\overline{S} \setminus \{x\}\{x\}$ y $\overline{S'}S' = \overline{S}S \setminus \{x\}S \cup \overline{S}\{x\} \setminus \{x\}\{x\}$
Luego, usando la hipótesis inductiva

$$F = \sum_{e \in S'\overline{S'}} f(e) - \sum_{e \in \overline{S'}S'} f(e) =$$

$$\sum_{e \in S\overline{S}} f(e) - \sum_{e \in S\{x\}} f(e) + \sum_{e \in \{x\}\overline{S}} f(e) - \sum_{e \in \{x\}\{x\}} f(e)$$

$$- \sum_{e \in \overline{S}S} f(e) + \sum_{e \in \{x\}S} f(e) - \sum_{e \in \overline{S}\{x\}} f(e) + \sum_{e \in \{x\}\{x\}} f(e) =$$

$$\sum_{e \in S\overline{S}} f(e) - \sum_{e \in \overline{S}S} f(e) + \sum_{e \in Out(x)} f(e) - \sum_{e \in In(x)} f(e) =$$

$$\sum_{e \in S\overline{S}} f(e) - \sum_{e \in \overline{S}S} f(e)$$

Donde usamos conservación del flujo en x en la última igualdad. □

Lemma 9. *Si f es una función de flujo con valor F y S un corte entonces*

$$F \leq c(S)$$

Proof. Usanco el Teorema 26 y que $0 \leq f(e) \leq c(e) \forall e \in X$

$$F = \sum_{e \in S\overline{S}} f(e) - \sum_{e \in \overline{S}S} f(e) \leq \sum_{e \in S\overline{S}} c(e) - \sum_{e \in \overline{S}S} 0 = c(S)$$

□

Corolary 11. *Si F es el valor de un flujo f y S un corte en N tal que $F = c(S)$ entonces f define un flujo máximo y S un corte de capacidad mínima.*

Theorem 27. *El algoritmo de camino de aumento determina un camino de aumento si existe, y si no llega a incorporar t en S es porque no lo hay.*

Proof. Trivial, por propiedades de BFS, DFS o el algoritmo que se quiera usar. ☐

Theorem 28. *Sea f un flujo definido sobre una red $N = (V, X)$ con valor F y sea P un camino de aumento en $R(N, f)$. Entonces el flujo $\overline{f}$, definido por*

$$\overline{f}(v \to w) = \begin{cases} f(v \to w) & si\ (v \to w) \notin P \\ f(v \to w) + \Delta(P) & si\ (v \to w) \in P \\ f(v \to w) - \Delta(P) & si\ (w \to v) \in P \end{cases}$$

Es un flujo factible sobre N con valor $\overline{F} = F + \Delta(P)$

Proof. Por definición de $\Delta(P)$, $0 \leq \overline{f}(e) \leq c(e) \forall e \in X$. Veamos que sigue valiendo la conservación del flujo. Sea $v \in V \setminus \{s, t\}$, si $v \notin P$ entonces vale por conservación de f. Si por el contrario, $v \in P$ con $(x, v) \in P \wedge (v, y) \in P$ entonces hay cuatro casos.

- $(x, v) \in X \wedge (v, y) \in X$. En este caso

$$\begin{aligned} \sum_{e \in In(t)} \overline{f}(e) - \sum_{e \in Out(t)} \overline{f}(e) &= \left(\sum_{e \in In(t)} f(e) + \Delta(P) \right) - \left(\sum_{e \in Out(t)} f(e) + \Delta(P) \right) \\ &= \sum_{e \in In(t)} f(e) - \sum_{e \in Out(t)} f(e) \end{aligned}$$

- $(x, v) \in X \wedge (y, v) \in X$. En este caso

$$\begin{aligned} \sum_{e \in In(t)} \overline{f}(e) - \sum_{e \in Out(t)} \overline{f}(e) &= \left(\sum_{e \in In(t)} f(e) \right) - \left(\sum_{e \in Out(t)} f(e) + \Delta(P) - \Delta(P) \right) \\ &= \sum_{e \in In(t)} f(e) - \sum_{e \in Out(t)} f(e) \end{aligned}$$

- $(v, x) \in X \wedge (v, y) \in X$. En este caso

$$\begin{aligned} \sum_{e \in In(t)} \overline{f}(e) - \sum_{e \in Out(t)} \overline{f}(e) &= \left(\sum_{e \in In(t)} f(e) + \Delta(P) - \Delta(P) \right) - \left(\sum_{e \in Out(t)} f(e) \right) \\ &= \sum_{e \in In(t)} f(e) - \sum_{e \in Out(t)} f(e) \end{aligned}$$

- $(v, x) \in X \wedge (y, v) \in X$. En este caso

$$\begin{aligned} \sum_{e \in In(t)} \overline{f}(e) - \sum_{e \in Out(t)} \overline{f}(e) &= \left(\sum_{e \in In(t)} f(e) - \Delta(P) \right) - \left(\sum_{e \in Out(t)} f(e) - \Delta(P) \right) \\ &= \sum_{e \in In(t)} f(e) - \sum_{e \in Out(t)} f(e) \end{aligned}$$

Luego, el flujo se conserva. Finalmente, basta elegir un corte cualquiera S que solo tome una parte conexa de P para ver que

$$\overline{F} = \sum_{e \in S\overline{S}} \overline{f}(e) - \sum_{e \in \overline{S}S} \overline{f}(e) = \sum_{e \in S\overline{S}} f(e) - \sum_{e \in \overline{S}S} f(e) + \Delta(P) = F + \Delta(P)$$

Ya sea porque se agregó $\Delta(P)$ a $S\overline{S}$ o $-\Delta(P)$ a $\overline{S}S$. □

Theorem 29. *Sea f un flujo definido sobre una red N. Entonces f es un flujo máximo $\Leftrightarrow$ no existe camino de aumento en $R(N, f)$.*

Proof. $\Rightarrow$
Sea f máxmino con valor F y P camino de aumento en $R(N, f)$, entonces por el Teorema 28 existe un flujo $\overline{f}$ con valor $\overline{F} = F + \Delta(P)$, lo que es absurdo ya que por definición $\Delta(P) \geq 0$ y f es máximo.
$\Leftarrow$
Como no hay camino de aumento, tiene que haber un corte S tal que $S\overline{S} = \varnothing$ en $R(N, f)$, esto implica que (por definición de red residual), para toda arista $(v \to w)$ en $S\overline{S}$ en N, $f(v \to w) = c(v \to w)$ y para toda arista $(w \to v)$ en $\overline{S}S$ en N, tiene que valer que $f(w \to v) = 0$. Luego, por el Teorema 26

$$F = \sum_{e \in S\overline{S}} f(e) - \sum_{e \in \overline{S}S} f(e) = \sum_{e \in S\overline{S}} c(e) - \sum_{e \in \overline{S}S} 0 = c(S)$$

lo que marca que hay un corte S tal que $c(S) = F$. Esto, por el corolario 11 implica que f es un flujo máximo. □

10 Complejidad

Theorem 30. *3-SAT es NP-Completo*

Proof. Vamos a reducir SAT a 3-SAT. Sea $\phi = C_1 \wedge ... \wedge C_m$. Vamos a reducir cada C_i a una conyuncion de disyunciones ϕ'_i donde cada clausula tenga tres literales y $\phi \Leftrightarrow \phi'$.

- Si C_i tiene tres literales entonces $\phi'_i = C_i$.
- Si C_i tiene dos literales, x_1 y x_2 agregamos una nueva variable y tal que $\phi'_i = (x_1 \vee x_2 \vee y) \wedge (x_1 \vee x_2 \vee \neg y)$
- Si C_i tiene $k \leq 4$ literales, entonces agregamos $k - 3$ variables tal que $\phi'_i = (x_1 \vee x_2 \vee y_1) \wedge ... \wedge (\neg y_{r-2} \vee x_r \vee y_{r-1}) ... \wedge (\neg y_{k-3} \vee x_{k-1} \vee x_k)$

□

Theorem 31. *Coloreo es NP-Completo*

Proof. Vamos a reducir SAT a coloreo. Para ello, a partir de una formula $\phi = C_1 \wedge ... \wedge C_m$ armamos un grafo $G = (V, X)$, con $V = V_1 \uplus V_2 \uplus V_3$ y donde $|V_1|$ = cantidad de literales en ϕ, $|V_2| = m$ y $|V_3|$ = la cantidad de variables en ϕ. Además, V_1 es completo, V_2 tiene una arista a cada nodo de V_1 por cada literal que aparece en la cláusula correspondiente. También hay una arista de cada nodo de V_2 a cada variable de V_3 por cada variable que aparece en la cláusula. Finalmente, hay una arista entre cada nodo de V_3 y todos los literales que no correspondan a la variable correspondiente. Este grafo es $|V_1|$-coloreable si y solo si ϕ es satisfacible. $\square$

11 Problemas Conocidos

11.1 P

Los siguientes problemas pertenecen a la clase de complejidad P:

- Recorrido de grafos: DFS, BFS: $\mathcal{O}(n+m)$.
- Encontrar componentes conexas: DFS, BFS: $\mathcal{O}(n+m)$.
- Encontrar componentes fuertemente conexas: Tarjan: $\mathcal{O}(n+m)$.
- Árbol generador mínimo. Prim, Kruskal: $\mathcal{O}(m\log(n))$.
- Camino mínimo, variante con único origen. Dijkstra (no puede haber pesos negativos): $\mathcal{O}(m\log(n))$. Bellman-Ford: $\mathcal{O}(mn)$.
- Camino mínimo, variante con múltiples orígenes. Floyd, Dantzig. $\mathcal{O}(n^3)$.
- Encontrar circuito o camino euleriano. Algoritmo de Hierholzer (DFS con el truco del stack): $\mathcal{O}(n+m)$.
- Reconocer planaridad. Hopcroft-Tarjan en $\mathcal{O}(n+m)$
- Flujo máximo. Edmonds-Karp: $\mathcal{O}(nm^2)$
- Matching máximo bipartito. Edmonds-Karp: $\mathcal{O}(nm^2)$
- Problema del cartero chino en grafos dirigidos o no dirigidos.
- Flujo mínimo.
- Mínimo cubrimiento de aristas.

11.2 NP-C

Los siguientes problemas son NP-completos:

- Encontrar un camino hamiltoniano.
- Problema del cartero chino en grafos mixtos.
- Travelling salesman problem (encontrar circuito hamiltoniano de costo mínimo en un grafo completo).
- Coloreo de nodos.
- 3-Coloreo.
- SAT.
- 3-SAT.
- Coloreo de aristas (decidir si $\Delta(G) = \chi'(G)$ o $\chi'(G) = \Delta(G) + 1$).
- Homomorfismo de grafos.
- Camino máximo.
- Conjunto máximo independiente.
- Clique máximo.
- Decidir si H es menor de G.
- Cubrimiento de nodos.
- Knapsack

11.3 NP

Los siguientes problemas son NP pero no se sabe si estan en P, son NP-completos o ninguno de los dos:

- Isomorfismo de grafos.
- Factorización de numeros naturales.