# Flujo Máximo y Corte Mínimo

Colectivo Estructuras de Datos y Algoritmos

Noviembre 2020

1. Responda V o F. Justifique en cada caso.

   (a) Sea $G = \langle V, E, c \rangle$ una red de flujo con fuente y receptor $s, t$ respectivamente y sea $f^*$ un flujo máxino en dicha red. Si selecciona una arista específica $e \in E$ y se incrementa su capacidad en una unidad, entonces el valor del flujo máximo en la nueva red $G'$ es $|f^*|$ o $|f^*| + 1$.

   **Respuesta**

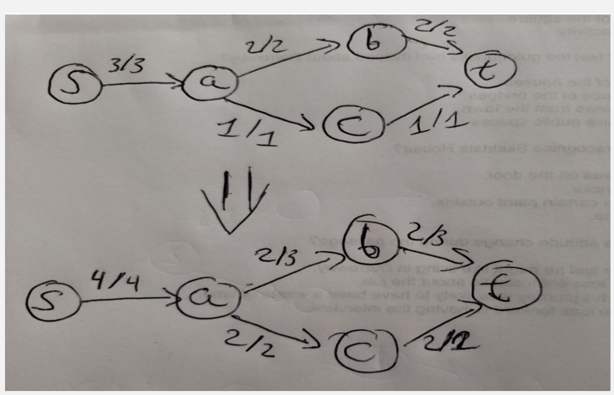
   (V). **Demostración:** Sea $G'$ la red de flujo resultante de incrementar en uno a la capacidad de una arista $e \in E$ en $G$. Sabemos que el valor del flujo máximo en $G'$ es al menos $|f^*|$, dado que $f^*$ sigue siendo un flujo factible en $G'$, pues al aumentar la capacidad de una arista no se afecta en nada las condiciones necesarias para que el flujo $f^*$ exista en $G'$. Así que es suficiente con demostrar que el valor del flujo máximo en $G'$ es a lo sumo $|f^*| + 1$.

   Por el *Teorema de Max-Flow Min Cut* sabemos que existe un corte $(A, B)$ con respecto a $s, t$ en la red de flujo original $G$ de capacidad $|f^*|$. Ahora veamos cuál sería la capacidad de dicho corte en $G'$. Todas las aristas que cruzan el corte $(A, B)$ tienen la misma capacidad en $G'$ que en $G$, con la posible excepción de $e$ (en caso que $e$ cruce el corte $(A, B)$). Pero la capacidad de $e$ solo aumento en 1, así que la capacidad del corte $(A, B)$ en la red de flujo $G'$ es a lo sumo $|f^*| + 1$ y por lo tanto el valor de todo flujo en $G'$ está acotado por la capacidad de todo corte, en particular $(A, B)$, por lo que todo flujo incluido el máximo en $G'$ estará acotado por $|f^*| + 1$.

   (b) Sea $G = \langle V, E, c \rangle$ una red de flujo con fuente y receptor $s, t$ respectivamente. Sea $(A, B)$ un corte mínimo con respecto a las capacidades de la función $c$. Ahora supongamos que se añade uno a cada capacidad, o sea tendríamos una red de flujo $G' = \langle V, E, c' \rangle$, donde para toda arista $e \in E$ se cumple que $c'(e) = c(e) + 1$. Entonces el corte $(A, B)$ también será un corte mínimo ahora en la red de flujo $G'$ con estas nuevas capacidades.

   **Respuesta**

   (F). Contraejemplo. En la imagen se muestra la red de flujo original arriba y la red de flujo tras incrementar en uno la capacidad a cada arista:




   Inicialmente el corte $(A, B)$ donde $A = \{s, a\}$ y $B = \{b, c, t\}$ es un corte de capacidad mínima en dicha red de flujo pues el flujo mostrado es máximo dado que $|f^*| = \sum_{v \in V} c(s, v)$ la cual es una cota superior de todo flujo y como $c(A, B) = |f^*|$ que es una cota inferior entonces el corte $(A, B)$ es mínimo.

Sin embargo al añadir 1 a la capacidad de toda arista ya el corte $(A, B)$ no es mínimo pues $c(A, B) = 5$ mientras que el valor del flujo máximo es 4, así como la capacidad del corte que es mínimo en la red modificada que es $(A', B')$ donde $A' = \{s\}$ y $B' = \{a, b, c, t\}$.

2. Diseñe un algoritmo que dada una red de flujo $G = \langle V, E, c\rangle$ y un flujo máximo $f^*$ en dicha red, devuelva un corte mínimo con respecto a capacidades de las aristas en $G$. La complejidad temporal de su algoritmo debe ser de $O(|V| + |E|)$.

Respuesta

El objetivo de este ejercicio es que refresquen la demostración de conferencia del *Teorema de Max Flow Min Cut* la cual brinda implícitamente un algoritmo para la extracción de un corte de capacidad mínima en una red de flujo a partir de su flujo máximo.

El algoritmo consiste en realizar un $BFS$ o $DFS$ a partir de $s$ en la red residual $G_{f^*}$, todos los vértices alcanzados por $s$, incluyendo $s$ conformarán una parte del corte y los no alcanzables junto a $t$ la otra parte del corte. Demostremos ahora la correctitud de dicho algoritmo, o sea demostremos que el corte obtenido siguiendo este algoritmo sobre la red de flujo $G$ con flujo máximo $f^*$ es de capacidad mínima.

**Demostración:** Sea el corte $(S, T)$ formado por $S = \{s, v_1, v_2, ..., v_k\}$ y $T = \{v_{k+1}, v_{k+2}, ..., v_{n-2}, t\}$ donde todo vértice $v_i \in S$ es alcanzable por $s$ a través de un camino en la red residual $G_{f^*}$ y todo vértice en $T$ serían los no alcanzables por $s$ bajo este criterio. Sabemos que ambos conjuntos son no vacíos, pues a $S$ al menos pertenece el vértice $s$ y a $T$ tiene que pertenecer el vértice $t$, pues si no perteneciera significa que existe un camino de $s$ hacia $t$ en $G_{f^*}$ lo que implica que sería un camino aumentativo y sería posible incrementar el flujo utilizando dicho camino lo cual entra en contradicción con el hecho de que $f^*$ es un flujo máximo en $G$.

Luego, toda arista $< u, v >$ que cruza el corte $(S, T)$ de $S$ hacia $T$, donde $u \in S$ y $v \in T$ cumple que $f(< u, v >) = c(< u, v >)$, pues si existiese $< u, v >$ que cruza el corte tal que $f(< u, v >) < c(< u, v >)$ entonces la arista $< u, v >$ no estaría saturada y existiera en $G_{f^*}$ por lo que el camino de $s$ hacia $u$ de aristas no saturadas, adicionando la arista $< u, v >$ forman un camino $s$ hacia $v$ en $G_{f^*}$ lo cual es una contradicción pues $v \in T$.

Además toda arista $< v, u >$ que cruza el corte desde $T$ hacia $S$ donde $u \in S$ y $v \in T$ cumple que $f(< u, v >) = 0$, pues si existiese $< v, u >$ que cruza el corte tal que $f(< v, u >) > 0)$ entonces dicha arista da origen a una arista inversa en $G_{f^*}$, denotémosla por $< u, v >$ y como $u \in S$, entonces existe un camino desde $s$ hacia $u$ en $G_{f^*}$ que al unirse con la arista $< u, v >$ daría lugar a un camino de $s$ hacia $v$ en $G_{f^*}$ lo cual es una contradicción con el hecho de que $v \in T$.

Por lo tanto todas las aristas que salen de $S$ están saturadas y todas las que entran tienen flujo 0. Demostremos entonces que la capacidad del corte $(S, T)$ es igual al valor del flujo máximo $f^*$ en $G$. Sabemos que $|f^*| = f(S, T)$ por el *Lemma* 26.4 visto en conferencia. Luego:

$$|f^*| = f(S, T) = \sum_{u \in S} \sum_{v \in T} f(u, v) - \sum_{u \in S} \sum_{v \in T} f(v, u) \tag{1}$$

Pero como demostramos anteriormente que $f(u, v) = c(u, v)$ y $f(v, u) = 0$ donde $u \in S$ y $v \in T$, entonces:

$$|f^*| = f(S, T) = \sum_{u \in S} \sum_{v \in T} c(u, v) - \sum_{u \in S} \sum_{v \in T} 0 = \sum_{u \in S} \sum_{v \in T} c(u, v) = C(S, T) \tag{2}$$

Luego la capacidad del corte $(S, T)$ es igual al valor del flujo máximo $f^*$ en $G$ por lo que su capacidad es mínima ya que $|f^*|$ es una cota mínima de la capacidad de todo corte por *Corolario* 26.5 visto en conferencia:

$$f(S,T) = |f| \leq C(S,T) \tag{3}$$

**Pseudocódigo**

```
visited[] ← DFS(s)
S ← []
T ← []
for v ∈ V do
    if visited[v] == TRUE then
        S.Add(v)
    else
        T.Add(v)
    end if
end for
return (S, T)
```

Como se puede observar la complejidad temporal sería la complejidad del $DFS$ que sabemos que es $O(|V| + |E|)$ y la complejidad del ciclo de tipo $for$ la cual es $O(V)$. Luego por regla de la suma la complejidad temporal es $O(|V| + |E|)$.

3. Diseñe un algoritmo que dado un grafo dirigido $G = \langle V, E \rangle$, y dos vértices $x, y \in V$ devuelva la menor cantidad de aristas que es necesario remover del grafo $G$ para que no exista ningún camino desde $u$ hacia $v$. La complejidad temporal de su algoritmo debe ser $O(|E| * k)$, donde $k$ es la menor cantidad de aristas necesarias a remover para lograr el objetivo.

Respuesta

La idea en este problema es que como queremos desconectar a $x$ de $y$ quitando la menor cantidad de aristas posibles, una forma de ver el problema es encontrar de todos los cortes $(A, B)$ donde $x \in A$ y $y \in B$, el de menor cantidad de aristas que cruza el corte y devolver esa cantidad. Ahora qué problema tiene esa solución? El problema está en que pudiera existir una arista en ese corte de menor cantidad de aristas que al removerla en realidad no se esté destruyendo ningún camino desde $x$ a $y$, pues poder pasar de $A$ a $B$ no significa que existe un camino seguro a $y$. Sin embargo podemos inducir esto, quedándonos con el grafo que solo tiene caminos desde $x$ hacia $y$ en $G$, ya que toda arista que no pertenezca a un camino de $x$ a $y$ no influye en nuestro objetivo. De esta forma todo corte $(A, B)$ cumplirá que para todo vértice $u \in A$ existe un camino de $x$ a $u$ y para todo vértice $v \in B$ existe un camino de $v$ a $y$.

Este razonamiento da la idea de una posible solución donde a partir del grafo $G$ nos quedamos con un subgrafo $G_c$ con solamente aristas que estén en caminos de $x$ hacia $y$. Luego se puede tomar $G_c$ como una red de flujo, y como lo que nos interesa es los cortes, siguiendo la idea anterior, y queremos que el corte de capacidad mínima coincida con un corte de menor cantidad de aristas y por supuesto que el valor de la capacidad mínima del corte coincida con la cantidad de aristas y esto podemos lograrlo si a todas las aristas se les pone una capacidad de 1.

Sin embargo, aunque esta idea ayuda a entender mejor el problema y los detalles en el mismo, si se piensa más a fondo, dicha transformación de $G$ en $G_c$ no es necesaria. Y el algoritmo de tomar $G$ como una red de flujo con función de capacidad que otorga el valor de una unidad a la capacidad de toda arista y devolver la capacidad del corte de capacidad mínima o flujo máximo, también funciona. Pues las unidades de flujo solo pueden llegar desde $x$ hasta $y$ a través de caminos de $x$ a $y$ en $G$, por lo que solo estarán con unidades de flujo y saturadas, dado que la capacidad es 1 para toda arista, las aristas que pertenezcan a caminos de $x$ hacia $y$. Y como sabemos además que existe un corte por el ejercicio 2-) de capacidad mínima donde todas las aristas que cruzan el corte están saturadas entonces en ese corte todas las aristas que cruzan el corte están en caminos de $x$ hacia $y$ y además si se remueven todas se desconecta $x$ de $y$ y no puede existir un corte con estas condiciones como las del ejercicio 2-) que además tenga menos aristas.

Realicemos ahora la construcción formal de la red de flujo y la demostración de la correctitud de dicho algoritmo.

**Construcción:** Sea $G' = \langle V', E', c\rangle$ la red de flujo tal que $V' = V$, $E' = E$, los vértices fuente y receptor son $x, y$ respectivamente y la función de capacidad $c$ se define como $c(e) = 1$ para toda $e \in E'$.

**Demostración:** Demostremos ahora que la cantidad mínima de aristas que se necesitan quitar para que no exista ningún camino desde $x$ hacia $y$ es $k$ si y solo si el valor del flujo máximo $f^*$ en $G'$ es $k$ ($|f^*| = k$).

($(k \geq |f^*|) \longrightarrow$) Sabemos por ejercicio 5-) de la Clase Práctica anterior de flujo máximo que en la red de flujo $G'$ el valor del flujo máximo coincide con la cantidad máxima de caminos de $x$ hacia $y$ disjuntos en aristas. Sea $K = \{e_1, e_2, ..., e_k\}$ un conjunto mínimo de $k$ aristas que al removerlas no existe camino de $x$ hacia $y$ en $G$ y por tanto en $G'$. Si se remueven todas las aristas en $K$ entonces se separan $x$ de $y$, por lo tanto cada camino de $x$ hacia $y$ en $G$ debe tener como mínimo una arista en $K$ y por lo tanto el número de caminos disjuntos en aristas desde $x$ hacia $y$ en $G$ es a lo sumo $k$. Lo cual implica $k \geq |f^*|$.

($(k \leq |f^*|) \longleftarrow$) Sea $f^*$ un flujo máximo en $G'$. Por el *Teorema de Max flow Min Cut* existe un corte $(A, B)$ con capacidad $|f^*|$. Sea $K$ el conjunto de aristas que van desde $A$ hacia $B$, cada arista tiene capacidad 1 por lo que $|K| = |f^*|$. Luego por definición de corte remover las $|f^*|$ aristas separa $x$ de $y$ lo cual implica $k \leq |f^*|$).

**Pseudocódigo**

```
1: G′ ← Build_Network_From(G)
2: |f*| ← Ford-Fulkerson(G′)
3: return |f*|
```

Como se puede observar la complejidad temporal depende de los métodos *Build_Network_From(G)* y del algoritmo de *Ford-Fulkerson*. Construir la red de flujo es básicamente crear un grafo donde $V' = V$ lo cual es $O(|V|)$ y $E' = E$ lo cual es $O(|E|)$. Luego el método *Build_Network_From(V)* es $O(|V| + |E|)$. Y el algoritmo de *Ford-Fulkerson* tal y como vimos en la conferencia sería $O(|E| * k)$ donde $k$ es la cantidad mínima de aristas a remover para sepaarar $x$ de $y$. Donde finalmente por regla de la suma resulta en $O(|E| * k)$

**Lista de Estudios Independientes:**

(a) Pseudocódigo de un algoritmo que extraiga el conjunto $K$ de aristas mínimo para separar $x$ de $y$.

4. Diseñe un algoritmo que dada una red de flujo $G = \langle V, E, c\rangle$ encuentre entre todos los cortes de capacidad mínima el de menor cantidad de aristas, si existe más de uno, devolver cualquiera. La complejidad temporal de su algoritmo debe ser de $O(|E|^2 * \sum_{e \in E} c(e))$. **Hint:** Modifique $G$ de tal forma que en la nueva red cualquier corte de capacidad mínima sea un corte de capacidad y cantidad de aristas mínimas en la red original.

Respuesta

Ya el **Hint** nos da un buen lugar donde comenzar. Pues sin modificar $G$ esta claro que es extremadamente complicado, habría que pasar por todos los cortes posibles chequear si es de capacidad mínima y luego quedarnos con el de menor cantidad de aristas lo cual es bastante ineficiente. Entonces una alternativa en este tipo de problemas es adaptar el problema a otro (en este caso la red) o al mismo problema, donde se fuerce a que una solución en el nuevo problema cumpla con los requerimientos para se solución en el problema original. En este caso buscar una alteración a la red de flujo $G$ para que todos los cortes de capacidad mínima sean un corte de capacidad mínima y de cantidad de aristas mínimas en $G$.

Ahora la pregunta es cuál sería la modificación. Cómo lograr que de todos los cortes de capacidad mínima en $G$, solo sean de capacidad mínima en la nueva red, los de menor cantidad de aristas. Una primera idea es pensar en penalizar a esos cortes de más aristas y esto se puede lograr si a todas las aristas se les suma 1 en la capacidad, pues los cortes que tengan más aristas tendrán entonces más capacidad que los de menor cantidad de aristas. Sin embargo, aunque parece como si ya essa fuese la solución, esto tiene un problema grave y es que al hacer eso pudiera suceder que un corte que no era de capacidad mínima pero tenía menor cantidad de aristas que todos los cortes de capacidad mínima sería beneficiado, beneficiado al punto que podría terminar siendo un corte de capacidad mínima en la nueva red y terminemos devolviendo ese, el cual no es un corte de capacidad mínima en la red original. Se deja de estudio independiente construir un ejemplo en el que esto suceda.

Para resolver este problema, es necesario reescalar las capacidades antes de hacer la penalización. La razón por la que pudiera pasar el problema que describimos anteriormente es precisamente porque la suma de las capacidades de un corte puede, potencialmente, ser menor que la cantidad de aristas. Que quiere decir esto, sea el corte $(A, B)$, si sucede que:

$$\sum_{u \in A, v \in B} c(< u, v >) < |E| \tag{4}$$

Entonces al sumar 1 a la capacidad de todas las aristas, dado que $c(< u, v >)$ puede ser cualquier número entero, lo que añades en capacidad a los cortes depende de la cantidad de aristas del corte. Luego sea $(A, B)$ un corte de capacidad mínima, entonces sea $(A', B')$ que no es de capacidad mínima y sea la función $E(A', B')$ la cantidad de aristas que cruzan el corte $(A', B')$ tal que:

$$\sum_{u \in A', v \in B'} c(< u, v >) - \sum_{u \in A, v \in B} c(< u, v >) < E(A', B') \tag{5}$$

Si esto sucede para algún corte entonces al sumar 1 a las capacidades de todas las aristas el corte $(A', B')$ pasaría a ser un corte de capacidad mínima o en un peor caso el único corte de capacidad mínima. Luego, qué hacemos para que esto no suceda y ahí entra la idea de reescalar los valores de las capacidades, o sea, en vez de sumar 1 sumar o multiplicar algo lo suficientemente grande como para que la condición anterior no pueda suceder al sumar 1 y penalizar los cortes de más aristas. Particularmente una buena idea es multiplicar por $|E|$ el valor de todas las capacidades de esta forma la condición de arriba nunca se daría.

Ahora planteemos la construcción formal y hagamos la demostración de la solución.

**Construcción:** Construyamos una nueva red de flujo $G' = \langle V', E', c' \rangle$, donde $V' = V$ y $E' = E$, y la nueva función de capacidad $c'$ se define como:

$$\text{c'(e)=} \begin{cases} \text{c(e)*}|E|\text{+ 1} & e \in E' \\ 0 & \text{en otro caso} \end{cases}$$

**Demostración:** Para demostrar la correctitud de dicha solución demostremos la siguiente proposición:

**Proposición:** Un corte $(A, B)$ es de capacidad mínima en la red de flujo $G'$ si y solo si es un corte de capacidad mínima y mínima cantidad de aristas en la red de flujo $G$.

$(\longrightarrow)$ Supongamos que tenemos un corte $(A, B)$ de capacidad mínima en la red de flujo $G'$. Demostremos que es de capacidad mínima y mínima cantidad de aristas en $G$. Hagámoslo por reducción al absurdo, supongamos que el corte $(A, B)$ no es de capacidad mínima y mínima cantidad de aristas en $G$. Veamos primero que pasa si asumimos que no es de capacidad mínima:

Si el corte $(A, B)$ no es de capacidad mínima en $G$, entonces sea $(A', B')$ un corte de capacidad mínima en $G$, entonces la capacidad del corte $(A', B')$ en la red de flujo $G'$ sería:

$$\sum_{u \in A', v \in B'} c'(< u, v >) = \sum_{u \in A', v \in B'} |E| * c(< u, v >) + 1 \tag{6}$$

Si definimos a la función $E(A, B)$ a la cantidad de aristas que cruzan el corte $(A, B)$ y abrimos la fórmula nos queda:

$$|E| \sum_{u \in A', v \in B'} c(< u, v >) + E(A', B') \tag{7}$$

Como el corte $(A, B)$ es de capacidad mínima en $G'$, entonces:

$$|E| \sum_{u \in A', v \in B'} c(< u, v >) + E(A', B') - |E| \sum_{u \in A, v \in B} c(< u, v >) - E(A, B) > 0 \tag{8}$$

Donde al despejar:

$$\sum_{u \in A', v \in B'} c(< u, v >) - \sum_{u \in A, v \in B} c(< u, v >) > \frac{E(A, B) - E(A', B')}{|E|} \tag{9}$$

Lo cual, debido a que el corte de $(A', B')$ es de capacidad mínima en $G$, sucede que:

$$\sum_{u \in A', v \in B'} c(< u, v >) - \sum_{u \in A, v \in B} c(< u, v >) < 0 \tag{10}$$

Pero como las capacidades son numeros enteros, la diferencia entre el corte $(A', B')$ y $(A, B)$ es estricta dado que asumimos que $(A, B)$ no es de capacidad mínima en $G$, pero $(A', B')$ sí, entonces en realidad:

$$\sum_{u \in A', v \in B'} c(< u, v >) - \sum_{u \in A, v \in B} c(< u, v >) <= -1 \tag{11}$$

Esto implica que:

$$-1 > \frac{E(A, B) - E(A', B')}{|E|} \tag{12}$$

Lo cual significa que en primer lugar el término $\frac{E(A,B)-E(A',B')}{|E|}$ tiene que ser negativo y su módulo un valor mayor que 1. Lo primero puede suceder si $E(A, B) - E(A', B') < 0$ pero lo segundo nunca pudiera ocurrir, pues dicha diferencia nunca sería mayor que $|E|$. Por lo cual llegamos a una contradicción.

Ahora veamos que sucede si asumimos que sí es de capacidad mínima en $G$ pero no de mínima cantidad de aristas:

Si el corte $(A, B)$ que es de capacidad mínima en $G'$ es de capacidad mínima en $G$ pero no es de mínima cantidad de aristas, entonces sea el corte $(A', B')$ un corte de capacidad mínima y mínima cantidad de aristas en $G$. Entonces:

$$\sum_{u \in A', v \in B'} c'(< u, v >) = \sum_{u \in A', v \in B'} |E| * c(< u, v >) + 1 \tag{13}$$

Si definimos a la función $E(A, B)$ a la cantidad de aristas que cruzan el corte $(A, B)$ y abrimos la fórmula nos queda:

$$|E| \sum_{u \in A', v \in B'} c(< u, v >) + E(A', B') \tag{14}$$

Como el corte $(A,B)$ es de capacidad mínima en $G'$, entonces:

$$|E| \sum_{u\in A', v\in B'} c(<u,v>) + E(A',B') - |E| \sum_{u\in A, v\in B} c(<u,v>) - E(A,B) > 0 \tag{15}$$

Donde al despejar:

$$\sum_{u\in A', v\in B'} c(<u,v>) - \sum_{u\in A, v\in B} c(<u,v>) > \frac{E(A,B) - E(A',B')}{|E|} \tag{16}$$

Lo cual, debido a que el corte de $(A',B')$ es de capacidad mínima en $G$ y también lo es $(A,B)$, sucede que:

$$\sum_{u\in A', v\in B'} c(<u,v>) - \sum_{u\in A, v\in B} c(<u,v>) = 0 \tag{17}$$

Esto implica que:

$$0 > \frac{E(A,B) - E(A',B')}{|E|} <=> E(A',B') > E(A,B) \tag{18}$$

Lo cual es una contradicción pues $(A',B')$ es un corte de capacidad mínima y mínimo en cantidad de aristas donde además habíamos asumido que $(A,B)$ no lo era.

**Pseudocódigo**

```
1: G' ← Build_Network_From(G)
2: |f*| ← Ford-Fulkerson(G')
3: C ← GET_MIN_CUT(G')
4: return C
```

Como se puede observar la complejidad temporal depende de los métodos *Build_Network_From(G)*, *GET_MIN_CUT(G′)* y del algoritmo de *Ford-Fulkerson*. Construir la red de flujo es básicamente crear un grafo donde $V' = V$ lo cual es $O(|V|)$ y $E' = E$ lo cual es $O(|E|)$, pero con una función de capacidad diferente que no afecta a la complejidad final. Luego el método *Build_Network_From(V)* es $O(|V| + |E|)$. El método de *GET_MIN_CUT(G′)* sería ejecutar el algoritmo que vimos en el ejercicio 2 el cual ya demostramos que es $O(|V| + |E|)$. Y el algoritmo de *Ford-Fulkerson* tal y como vimos en la conferencia sería $O(|E| * |f^*|)$ donde el valor del flujo máximo en este caso sería en el peor caso $|f^*| = |E| * \sum_{e\in E} c(e) + |E|$ lo cual nos lleva a una complejidad temporal de $O(|E|^2 * \sum_{e\in E} c(e) + |E|)$ que por regla de la suma sería $O(|E|^2 * \sum_{e\in E} c(e))$.

**Lista de Estudios Independientes:**

(a) Diseñar un ejemplo donde sumar solamente 1 a las capacidades de todas las aristas sea una solución incorrecta.

5. El **Edge Connectivity** de un grafo no dirigido se define como el número $k$ mínimo de aristas que se deben quitar para desconectar el grafo. Por ejemplo, el **Edge Connectivity** de un árbol es 1 y de un ciclo es 2. Diseñe un algoritmo que determine el **Edge Connectivity** de un grafo no dirigido $G = \langle V, E\rangle$. La complejidad temporal de su algoritmo debe ser de $O(|V||E|^2)$. **Hint:** Utilice $|V|$ redes de flujo construidas todas a partir del grafo $G$ cada una con $O(V)$ vértices y $O(E)$ aristas.

Respuesta

Ya en el ejercicio 3 vimos como encontrar la cantidad mínima de aristas para desconectar un par de vértices. En este caso queremos la menor cantidad de aristas para desconectar un grafo no dirigido. Sin embargo, el problema puede verse de otra forma que es la idea clave para la solución y es que desconectar un grafo no dirigido $G$ con la menor cantidad de aristas posibles es lo mismo que desconectar al menos un par de vértices con la menor cantidad de aristas posibles. Lo cual le da un enfoque ya mas semejante al ejercicio 3 y luego al ver el **Hint** es que puede surgir la idea de elegir un vérrtice $u \in V$ y resolver el ejercicio 3 para $u$ con cada otro de los vértices $v \in V$ en $G$ por lo cual tendríamos $O(|V|)$ redes de flujo donde $u$ siempre es la fuente y lo único que cambia es quién es el receptor. Y precisamente demostraremos que la menor cantidad de aristas para desconectar a $u$ de alguno del resto de los vértice $v$ es también el **Edge Conectivity** del grafo $G$.

Recordar que como estamos en un grafo no dirigido y vamos a usar el ejercicio 3 lo cual sería hallar el flujo máximo en cada una de las redes es necesario susutituir las aristas por dos aristas dirigidas en sentidos contrarios para cada par de vértices $u, v$ tal que $< u, v > \in E$. Y luego lidiar con el problema de las aristas antiparalelas como ya vimos en ejercicios anteriores.

Ahora realicemos la construcción formal de las $O(|V|)$ redes de flujo y demostremos la correctitud de nuestra solución:

**Construcción:** Sea un vértice $s \in V$, sean $v_1, v_2, ..., v_{|V|-1}$ el resto de los vértices en $V$ que no son $s$ y sean $G_1 = \langle V', E', c \rangle, G_2 = \langle V', E', c \rangle, ..., G_{|V|-1} = \langle V', E', c \rangle$ redes de flujo donde $V' = V$ y $E' = \{< u, v >, < v, u > \mid u, v \in V\ y < u, v > \in E\}$, donde en la red de flujo $G_i$ $(1 \leq i < |V|)$ la fuente es el vértice $s$ y el receptor es el vértice $v_i$. La función de capacidad es la misma en todas las redes de flujo y sería:

$$\text{c(e)=} \begin{cases} 1 & e =< u, v > \in E' \\ 0 & \text{en otro caso} \end{cases}$$

**Demostración:** Si denotamos como $f^*_{sv_i}$ como el flujo máximo en la red de flujo $G_i$. Habíamos dicho que nuestra solución sería el valor del mínimo de todos los $f^*_{sv_i}$. Sea ese mínimo $f^*_{sv}$, demostremos que $k$ es el valor del **Edge Conectivity** de $G$ si y solo si $k = |f^*_{sv}|$.

Demostremos primero que $k \geq |f^*_{sv}|$. Supongamos que se puede desconectar el grafo quitando $|f^*_{sv}| - 1$ aristas de $G$. Para todo $v \in V - \{s\}$, por el *Teorema de Max-Flow Min-Cut*, los vértices $s$ y $v$ siguen conectados, pues el flujo máximo en la red que tiene a $s$ como fuente y a $v$ como receptor es al menos $|f^*_{sv}|$ (dado que este valor es el mínimo de todos los flujos máximos), lo que significa que al menos $|f^*_{sv}|$ aristas son necesarias para separar a $s$ de $v$ por ejercicio 3, luego todos los vértices están conectados con $s$ dado que el grafo original es no dirigido y si todos los vértices están conectados con $s$ entonces el grafo $G$ se mantiene conexo. Por lo tanto $k \geq |f^*_{sv}|$ .

Demostremos ahora que $k \leq |f^*_{sv}|$. Sea el vértice $v$ el receptor de la red de flujo de menor valor de flujo máximo, luego por el *Teorema de Max-Flow Min-Cut* existe un corte que separa a $s$ de $v$ de capacidad $|f^*_{sv}|$. Luego exactamente $|f^*_{sv}|$ aristas es el mínimo a quitar para poder desconectar $s$ de $v$, nuevamente por el ejercicio 3. Por lo que $k \leq |f^*_{sv}|$.

**Pseudocódigo**

```
G_1, ..., G_{|V|-1} ← Build_Networks_From(G)
for i = 1 to |V| - 1 do
    f*_{sv_i} ← Ford-Fulkerson(G_i)
end for
m = MIN(f*_{sv_1}, ..., f*_{sv_{|V|-1}})
return m
```

La complejidad temporal de la primera línea depende del llamado a la función *Build_Networks_From(G)* la cual construye cada una de las redes de flujo antes descritas donde solo cambia el receptor, lo cual sería $O(|V|(|V|+|E))$ lo cual es $O(|V||E|)$. Luego tenemos un ciclo de tipo $for$ hallando el valor del flujo máximo en cada una de las redes de flujo usando el algoritmo de *Ford-Fulkerson(G)* el cual es $O(|E||f^*|)$ lo cual es en este caso $O(|E|^2)$, pero como es un $for$ por cada una de las redes, la complejidad del ciclo sería $O(|V||E|^2)$. Y finalmente se halla el mínimo lo cual es $O(|V|)$ en este caso, por lo que finalmente por regla de la suma la complejidad temporal final sería $O(|V||E|^2)$.