Árbol abarcador de costo mínimo

Colectivo Estructuras de Datos y Algoritmos

2021

Hint

Tengan como referencia el siguiente lema, que agrupa un conjunto de propiedades que son trasversales a esta Clase Práctica, y de las que podrán auxiliarse en el futuro.

Lema 1 *Sea $G =< V, E >$ un grafo no dirigido y conexo. Sea T un árbol abarcador de G. Sea $e = (u, v)$ una arista de T.*

(a) Si se elimina e de T, esto resulta en un grafo con dos componentes conexas S y $V - S$. Noten que como T es abarcador, $(S, V - S)$ es un corte en G, y e cruza dicho corte. Denominemos a ese corte $C_{T,e}$.

(b) Sea C un camino entre u y v en G. Entonces, existe una arista $e' \in E$ tal que e' cruza el corte $C_{T,e}$.

(c) Sea e' una arista que cruza el corte $C_{T,e}$. Entonces el grafo $T' = T - e + e'$ (resultante de eliminar la arista e y agregar la arista e' en el T), es un árbol abarcador de G.

Demostración

(a) Esto se deriva directamente de la definición de árbol, y el hecho de que eliminar una arista de un árbol lo divide en dos componentes conexas.

(b) Noten que en $C_{T,e} = (S, V - S)$, se cumple los vértices de e quedan uno a cada lado del corte. Sin pérdida de generalidad digamos que $u \in S$ y $v \in (V - S)$. Sea $C = [x_0, x_1, ..., x_k]$ con $u = x_0, v = x_k$ y k posiblemente igual a 2 (o lo que es lo mismo, que solo exista un vértice además de u y v en C). Sea i el mayor entero tal que $\forall_{j \leq i}\ x_j \in S$. Nótese que $i < k$ (o lo que es lo mismo, x_i no puede ser v, ya que $v \notin S$). Entonces, la arista (x_i, x_{i+1}) está en el camino C y cruza el corte, ya que $x_{i+1} \in V - S$ por la forma en que se escogió i.

(c) Los grafos inducidos por los vértices de S y $V - S$, son las dos componentes conexas que resultan de eliminar la arista e del árbol T. Este nuevo grafo T' tiene dos componentes conexas y $|V| - 2$ aristas. Si se inserta la arista $e' = (u', v')$ que cruza el corte $C_{T,e}$, el grafo T'' que resulta es conexo, ya que u' y v' pertenecen a componentes conexas distintas de T'. Además, el grafo T'' tiene $|V(T'')| - 1$ aristas. Luego, es un árbol, por definición. Como los vértices no cambiaron, entonces es a su vez, un árbol abarcador de G.

1. Diga Verdadero (V) o Falso (F) según corresponda. Justifique su respuesta en cada caso.

   (a) Cuando a un grafo completo, no dirigido y ponderado se le quiere hallar el árbol abarcador de costo mínimo, el algoritmo más eficiente en tiempo es el de *Kruskal*.

   (b) Si la arista (u, v) es una arista de costo mínimo en un grafo, entonces (u, v) pertenece a algún **AACM**.

   (c) El conjunto de aristas $T = \{(u, v) \mid \exists (S, V - S)$ tal que (u, v) es una arista liviana que cruza $(S, V - S)\}$ forma un **AACM**.

   (d) Si la arista (u, v) está contenida en un **AACM** de un grafo , entonces en existe $(S, V - S)$ un corte para el cual la arista (u, v) cruza el corte y es liviana.

(e) Sea T el árbol abarcador de costo mínimo del grafo conexo y ponderado $G =< V, E >$ y sean $a, b \in V$. El camino simple en T que conecta los vértices a y b se corresponde con el camino de costo mínimo en G que los conecta.

2. Demuestre los siguientes enunciados:

   (a) Sea G un subgrafo de G, y T' un **AACM** de G' , entonces $\exists T$ **AACM** de G tal que, $\forall e \in T$ que esté en G' también está en T'.

   (b) Una arista e pertenece a todo **AACM** de un grafo G si y solo si existe un corte de los vértices de G tal que e es la única arista liviana que cruza el corte.

3. Suponga que se define el costo de un árbol como el producto del costo de las aristas contenidas en el árbol en lugar de la suma. Implemente un algoritmo que dado un grafo $G =< V, E >$, cuyas aristas tienen costo positivo, permita determinar el **AACM** de G de forma eficiente. Justifique el orden y la correctitud de su algoritmo.

4. Sea $G =< V, E >$ un grafo no dirigido, ponderado y conexo. Se desea implementar un algoritmo que permita determinar si una arista $e \in E$ pertenece a algún **AACM** de G.

   (a) Demuestre que una arista (u, v) no pertenece a ningún **AACM** de G si y solo si en existe un camino de u a v formado por aristas cuyos costos son todos menores que el de (u, v)

Respuesta

($\Leftarrow$) Supongamos que existe un camino C entre u y v formado por aristas cuyos costos son todos menores que el de $e = (u, v)$. Además, supongamos que e pertenece a algún **AACM** T.

Por el Lema 1 sabemos que existe una arista $e' \in C$ que cruza el corte $C_{T,e}$. Además, sabemos que el árbol $T' = T - e + e'$ es un árbol abarcador de G.

El costo de T' está dado por:

$$\omega(T') = \omega(T) - \omega(e) + \omega(e')$$

Pero como todas las aristas en C tienen costo menor estricto que que e, particularmente $\omega(e')$ ¡ $\omega(e')$. De lo que:

$$\omega(T') < \omega(T)$$

Luego T' es un árbol abarcador de costo menor que T, por lo que T no era un **AACM** de G. Contradicción! ∎.

($\Rightarrow$) Supongamos en este caso que e no pertenece a ningún **AACM** de G. Además, supongamos que para todo camino en G entre u y v existe una arista que tenga costo mayor o igual que e.

Sea T un **AACM** de G. Sabemos por nuestra suposición que $e \notin T$. Pero en T existe un camino C entre u y v. Además, $\exists e' \in C$ tal que $\omega(e') \geq \omega(e)$.

Sea $T' = T + e - e'$ el grafo que resulta de eliminar la arista e' de T y agregar la arista e. Al añadir la arista e al árbol T se forma el ciclo $C + e$. Al eliminar la arista e', el grafo que resulta sigue siendo conexo y tiene $|V| - 1$ aristas, por lo que sigue siendo un árbol, y como tiene los mismos vértices que T, es un árbol abarcador de G.

El costo de T' está dado por:

$$\omega(T') = \omega(T) + \omega(e) - \omega(e')$$

Pero como $\omega(e') \geq \omega(e)$, entonces

$$\omega(T') \leq \omega(T) \tag{1}$$

Como T es un **AACM** de G entonces

$$\omega(T') \geq \omega(T) \tag{2}$$

Finalmente, de las desigualdades 1 y 2 se obtiene que

$$\omega(T) = \omega(T')$$

Luego, T' es también un **AACM** de G, y contiene a la arista e. Contradicción! $\blacksquare$.

(b) Diseñe un algoritmo que en $O(|E|)$ dado un grafo $G =< V, E >$ y una arista $e \in E$ determine si e pertenece a algún **AACM** de G. Justifique el orden y la correctitud de su algoritmo.

Respuesta

Dada la propiedad demostrada en el inciso anterior, demostrar que $e = (u, v)$ pertenece a algún **AACM** de G es equivalente a probar que para cualquier camino entre u y v existe una arista cuyo costo es mayor o igual que e.

Una forma sencilla de verificar si esto se cumple o no, es construir el grafo G' que resulta de eliminar de G todas las aristas cuyo costo sea mayor o igual que el de e. Y comprobar si u y v están conectados en dicho grafo. Si están conectados significa que en G existe un camino entre u y v formado por aristas de costo menor que e. Si no están conectados, entonces cualquier camino en G que una a u y v tiene necesariamente una arista con costo mayor o igual que el costo de e.

El siguiente pseudocódigo ilustra el algoritmo.

```
1 def Solve(G, e, ω):
2     G' = (V(G), {e' | e' ∈ E(G) ∧ ω(e') < ω(e)})
3     d, pi = BFS(G,u)
4     return d[v] == ∞
```

Las líneas 2 y 3 tienen complejidad temporal $O(|V| + |E|)$ cada una, mientras que la línea 4 tiene costo constante. Luego, por regla de la suma, la complejidad temporal del algoritmo es $O(|V|+|E|)$.

5. Sea $G =< V, E >$ un grafo no dirigido, ponderado y conexo. Diseñe un algoritmo eficiente que permita determinar si una arista $e \in E$ dada, aparece necesariamente en todo **AACM** de G. Justifique el orden y la correctitud de su algoritmo.

Hint

No utilicen el resultado obtenido en el ejercicio 2b

6. Sea $G =< V, E >$ un grafo no dirigido, conexo y ponderado por una función $w : E \to \{1, 2\}$. Implemente un algoritmo que en $O(|E|)$ halle el árbol abarcador de costo mínimo de G. Justifique la correctitud y el orden de su algoritmo.

7. Sea G un grafo conexo y no dirigido y sea T un **AACM** de G. Suponga que se realiza una de las siguientes modificaciones a G:

(a) Decrecer el peso de una de las aristas de G que no está en T.

Respuesta

Para resolver este ejercicio nos apoyaremos en la propiedad enunciada en el ejercicio 2a.
Sea $G =< V, E >$, $e \in E$ la arista a la que se quiere hacer decrecer su peso. Definamos el grafo $G' =< V, E - e >$, resultante de quitar la arista e de G. Claramente, G' es subgrafo de G.

Noten además que T también es un **AACM** de G' ya que, como todas las aristas de G' están en G y tienen el mismo conjunto de vértices, si existiera un árbol abarcador de G' de menor costo que T, este también sería árbol abarcador de G, y eso contradice el hecho de que T es **AACM** de G.

De acuerdo al resultado del ejercicio 2a, existe un T'' **AACM** de G, tal que cualquier arista que tenga que esté en G' también estará en T. Esto significa que existe un **AACM** de G que solamente utiliza las aristas de T y la arista e. Calcular este **AACM** de G será mucho más eficiente que recalcular el **AACM** en todo el grafo.

Como T es un árbol, el grafo $T + e$ tiene n aristas y un ciclo que contiene a la arista e. El **AACM** de $T + e$ tiene que tener a toda arista que no pertenezca a dicho ciclo, porque cualquiera de ellas es necesaria para que el grafo se mantenga conexo. Descartar la mayor arista que pertenezca a ese ciclo resulta en el **AACM** de $T + e$. Este procedimiento se puede realizar en $O(|V|)$.

(b) Añadir un nuevo vértice a G con sus respectivas aristas.

Proponga para cada caso un algoritmo que recalcule de forma eficiente el **AACM** de G luego de modificarlo. El costo de sus algoritmos debe ser asintóticamente menor que el costo de recalcular desde cero el **AACM**. Justifique la correctitud y el orden de los mismos