Árbol de expansión de un grafo

Ronald Mas,
Angel Ramirez

11 de junio de 2021

Contenido

Algoritmo de Kruskal

La entrada es un grafo conexo $G = (V, E)$ con función de peso w para las aristas. Sea las aristas $e_1, e_1, \cdots, e_m$ tal que:

$$w(e_1) \leq w(e_2) \leq \cdots \leq w(e_n).$$

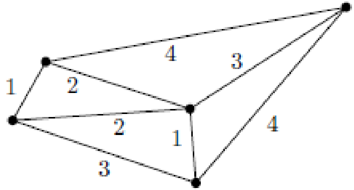
Para el ordenamiento de aristas ejecutar el algoritmo de expansión:

1) Se crea un bosque B (un conjunto de árboles), donde cada vértice del grafo es un árbol separado.
2) Se crea un conjunto C que contenga a todas las aristas del grafo.
3) Mientras C es no vacío, suprimir una arista de peso mínimo de C.
4) Si esa arista conecta dos árboles diferentes se añade al bosque, combinando los dos árboles en un solo árbol.
5) En caso contrario, se desecha la arista.

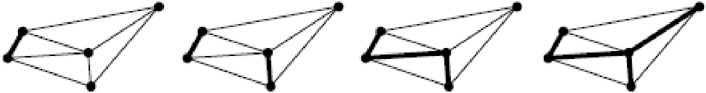
Al acabar el algoritmo, el bosque tiene un solo componente, el cual forma un árbol de expansión mínimo del grafo.

Ejemplo

Dada la siguiente red o malla:



Una posible ejecución del algoritmo de Kruskal se muestra en el siguiente diagrama:



Proposición

El algoritmo de Kruskal resuelve el problema de árbol expansión mínima.

Prueba:
Sea T un árbol de expansión encontrado por el algoritmo y sea $\check{T}$ otro árbol de expansión del grafo $G = (V, E)$. Por demostrar que $w(E(T)) \leq w(E(\check{T}))$. Denotemos las aristas de T por $e_1', e_2', \cdots, e_{n-1}'$ tal que $w(e_1') \leq w(e_2') \leq \cdots \leq w(e_{n-1}')$ (la arista $e_i = e_j$ para algún j). De igual modo, sea $\check{e}_1, \cdots, \check{e}_{n-1}$ las aristas de $\check{T}$ ordenados en orden creciente por pesos. Vamos a probar que para $i = 1, \cdots, n-1$ se tiene que:

$$w(e_i') \leq w(\check{e}_i).$$

Esto prueba que T es un árbol de expansión mínima, procedamos por contradicción supongamos que la desigualdad anterior no se cumple, entonces $w(e_i') > w(\check{e}_i)$.

Continua prueba

Al considerar los conjuntos

$$\begin{array}{rcl} E' & = & \{e'_1, e'_2, \cdots, e'_{i-1}\}, \\ \check{E} & = & \{\check{e}_1, \check{e}_2, \cdots, \check{e}_i\}. \end{array}$$

Los grafos (V, E') y $(V, \check{E})$ no contienen ciclos y $\mid E' \mid = i - 1$, $\mid \check{E} \mid = i$. Para llegar a una contradicción es suficiente probar que existe una arista $e \in \check{E}$ tal que el grafo $(V, E' \cup \{e\})$ no contiene ciclos, así obtenemos que $w(e) \leq w(\check{e}_i) < w(e'_i)$ y esto significa que al elegir la arista e en el algoritmo nosotros cometimos un error, pero no hay necesidad de eliminar la arista e ya que podemos seleccionar a e'_i en su reemplazo. Por tanto, es suficiente probar que, si $E', E \subseteq \binom{V}{2}$ son dos conjuntos de aristas tal que el grafo $(V, \check{E})$ no tiene ciclos y $\mid E' \mid < \mid \check{E} \mid$, entonces alguna arista $e \in \check{E}$ conecta vértices de componentes distintas del grafo V, E'.

Continua prueba

Esto se puede hacer mediante un simple argumento de conteo. Sea $V_1, \cdots V_s$ los conjuntos de vértices de las componentes del grafo (V, E'), luego se tiene que:

$$| E' \cap \binom{V_j}{2} | \geq | V_j | -1,$$

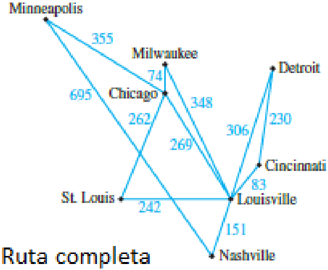
y al sumar estas desigualdades sobre j se tiene que $E' \geq n - s$. Por otro lado, como $\check{E}$ no tiene ciclos, se tiene:

$$| \check{E} \cap \binom{V_j}{2} | \leq | V_j | -1,$$

y por lo tanto a lo más $n - s$ aristas de $\check{E}$ estan contenidos en alguna de las componentes V_j, pero como asumimos que $| \check{E} | > | E' |$, existiría una arista $e \in \check{E}$ que pertenece a dos componentes distintas.

El camino mas corto

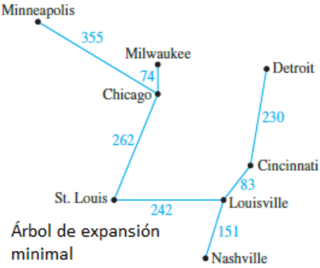
Aunque los árboles producidos por el algoritmo de Kruskal tiene el menor peso posible total en comparación con todos los demás árboles de expansión para los grafos dados, no siempre muestra la distancia más corta entre dos puntos en el grafo. Por ejemplo la siguiente red muestra la ruta completa de viajes entre ciudades con sus respectivas distancias en millas



Ruta completa

Continua

Se puede probar que al aplicacar del algoritmo de Kruskal en la red anterior se tiene el siguiente árbol de expansión minimal:



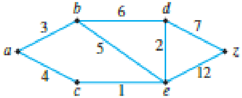
Árbol de expansión minimal

De acuerdo con el sistema de ruta completo, se puede volar directamente de Nashville a Minneapolis por una distancia de 695 millas, mientras que si usa el árbol de expansión mínimal la única forma de volar desde Nashville a Minneapolis es pasando por Louisville, St. Louis y Chicago, lo que da distancia total de 151+242+262+355=1 010 millas.

ALGORITHM 1 Dijkstra's Algorithm.

procedure *Dijkstra*(G: weighted connected simple graph, with
all weights positive)
{G has vertices $a = v_0, v_1, \ldots, v_n = z$ and lengths $w(v_i, v_j)$
where $w(v_i, v_j) = \infty$ if $\{v_i, v_j\}$ is not an edge in G}
for $i := 1$ **to** n
    $L(v_i) := \infty$
$L(a) := 0$
$S := \emptyset$
{the labels are now initialized so that the label of a is 0 and all
other labels are ∞, and S is the empty set}
while $z \notin S$
    $u :=$ a vertex not in S with $L(u)$ minimal
    $S := S \cup \{u\}$
    for all vertices v not in S
        if $L(u) + w(u, v) < L(v)$ **then** $L(v) := L(u) + w(u, v)$
        {this adds a vertex to S with minimal label and updates the
        labels of vertices not in S}
return $L(z)$ {$L(z) =$ length of a shortest path from a to z}

Dada la red



Veamos los pasos en la ejecución del algoritmo de Dijkstra para hallar la ruta más corta de a hacia z.

Step	$V(T)$	$E(T)$	F	$L(a)$	$L(b)$	$L(c)$	$L(d)$	$L(e)$	$L(z)$
0	$\{a\}$	$\varnothing$	$\{a\}$	0	∞	∞	∞	∞	∞
1	$\{a\}$	$\varnothing$	$\{b, c\}$	0	3	4	∞	∞	∞
2	$\{a, b\}$	$\{\{a, b\}\}$	$\{c, d, e\}$	0	3	4	9	8	∞
3	$\{a, b, c\}$	$\{\{a, b\}, \{a, c\}\}$	$\{d, e\}$	0	3	4	9	5	∞
4	$\{a, b, c, e\}$	$\{\{a, b\}, \{a, c\}, \{c, e\}\}$	$\{d, z\}$	0	3	4	7	5	17
5	$\{a, b, c, e, d\}$	$\{\{a, b\}, \{a, c\}, \{c, e\}, \{e, d\}\}$	$\{z\}$	0	3	4	7	5	14
6	$\{a, b, c, e, d, z\}$	$\{\{a, b\}, \{a, c\}, \{c, e\}, \{e, d\}, \{e, z\}\}$							