Matemática Discreta

Elementos de Teoria dos Grafos

Universidade de Aveiro 2017/2018

http://moodle.ua.pt

Algoritmo de Kruskal

Algoritmo de Kruskal

- **Entrada**: grafo $G = (V, E, W)$;
- **Saída**: T árvore abrangente de custo mínimo;

1. Ordenar as arestas $a_1, \ldots, a_m$ por ordem não decrescente do seu custo;
2. $E' \leftarrow \emptyset$ e $i \leftarrow 1$;
3. **Enquanto** $T = (V, E')$ não é conexo **faz**
   - **Se** $E' \cup \{a_i\}$ não contém um ciclo **então** $E' \leftarrow E' \cup \{a_i\}$;
   - $i \leftarrow i + 1$;
   - **Fim faz**;

- Devolver $T = (V, E')$.

Convergência do algoritmo de Kruskal

Teorema

Se G é um grafo conexo, então o algoritmo de Kruskal determina uma árvore abrangente de custo mínimo.

Prova: Uma vez que é imediato que (para grafos conexos) o algoritmo de Kruskal determina uma árvore abrangente, vamos mostrar, por redução ao absurdo, que esta árvore tem custo mínimo. Primeiro, vamos ordenar as arestas de acordo com os respectivos custo, obtendo-se

$$w(e_1) \leq w(e_2) \leq \cdots \leq w(e_\varepsilon).$$

Suponhamos que a árvore $\widetilde{T}$, determinada pelo algoritmo de Kruskal, não é optima. Seja T uma árvore abrangente óptima tal que T e $\widetilde{T}$ têm as mesmas arestas com índices não superiores a $k-1$ e k é o maior índice nesta condições.

Convergência do algoritmo de Kruskal (cont.)

Então, e_k é a próxima aresta a inserir no conjunto de arestas que vão formar $\widetilde{T}$ e é tal que $e_k \notin E(T)$. Logo, $E(T) \cup \{e_k\}$ contém um ciclo C que, necessariamente, contém uma aresta e que não pertence a $E(\widetilde{T})$ e é tal que $w(e) \geq w(e_k)$. Logo, substituindo $E(T)$ por $(E(T) \setminus \{e\}) \cup \{e_k\}$ obtém-se uma árvore abrangente T' de custo não superior ao custo de T.

1. Se $w(e) > w(e_k)$, então T' tem custo inferior ao de T o que é absurdo, uma vez que, por hipótese, T é uma árvore óptima.
2. Se $w(e) = w(e_k)$, então T' tem custo igual ao de T o que é absurdo, uma vez que T' tem pelo menos as k primeiras arestas coincidentes com as de $\widetilde{T}$ (o que contraria a definição de T).

Como consequência, podemos concluir que $\widetilde{T}$ é uma árvore abrangente óptima.

Algoritmo de Prim

Algoritmo de Prim

- **Entrada**: grafo $G = (V, E, W)$;
- **Saída**: T árvore abrangente de custo mínimo;

1. Escolher um vértice $v \in V$;
2. **Fazer** $V' \leftarrow \{v\}$; $E' \leftarrow \emptyset$;
3. **Enquanto** $V' \neq V$ **fazer**
   - De todas as arestas $e = (v_i, v_j)$, tais que $v_i \in V'$, $v_j \in V \setminus V'$ determinar a de menor custo $e^* = (v_i^*, v_j^*)$
   - **Fazer** $V' \leftarrow V' \cup \{v_j^*\}$; $E' \leftarrow E' \cup \{e'\}$.

- Devolver $T = (V', E')$.

Exercício

Determinar a árvore abrangente de custo mínimo do grafo G com custos nas arestas, definido pela seguinte matriz de custos:

$$W = \begin{pmatrix} 0 & 1 & \infty & 10 & 8 & 3 \\ 1 & 0 & 13 & 10 & 6 & 4 \\ \infty & 13 & 0 & 15 & \infty & 4 \\ 10 & 10 & 15 & 0 & 9 & \infty \\ 8 & 6 & \infty & 9 & 0 & 7 \\ 3 & 4 & 4 & \infty & 7 & 0 \end{pmatrix}$$