Minimal spanning trees, Travelling salespeople, and together

Steph van Willigenburg
Math 442-201 2019WT2

26 March 2020
Minimal spanning tree (MST)

**Definition**

Let $G$ be a connected weighted graph. A spanning tree of $G$ with smallest sum of edge weights is called a minimal spanning tree of $G$.

**Example**

$G =$

The minimal spanning tree of $G$ is
Kruskal’s algorithm for MST

Given a connected weighted graph $G$.

1. Write down the vertices of $G$. Add an edge of lowest weight.
2. Continue to add edges of lowest weight, never making a cycle.
3. Stop when all vertices are connected.

Note: Break ties arbitrarily.

Example
**Travelling Salesperson Problem (TSP)**

Given a network of $n$ connected cities, to visit once and only once and then return home, what is the minimum distance to travel?

**OR**

Given a weighted complete graph $K_n$, which Hamiltonian cycle has the sum of edge weights being minimal?

**Note:** No algorithm exists that does better than check all cycles.

**Example**

A minimal weight of is given by
Computing a lower bound: TSP meets MST

Given a weighted complete graph, $K_n$.

1. Choose a vertex $v$ and delete it.
2. Compute an MST for $K_{n-1} = K_n - v$.
3. Note the two smallest edge weights on edges coming from $v$.
4. Sum the edge weights found in 2 and 3.

Example 1 + 2: Delete $D$ and get MST

3: Two smallest weights coming from $D$:

4: Minimal weight $\geq$
Why does this work?

If we take any Hamiltonian cycle in a weighted $K_n$ and delete a vertex $v$, then we are left with a spanning tree of $K_{n-1} = K_n - v$.

So

\[
\begin{align*}
(Sum \ of \ weights \ in \ MST \ for \ K_{n-1}) & \\
+ \ (2 \ smallest \ edge \ weights \ meeting \ at \ v) & \\
\leq & \\
(Sum \ of \ weights \ in \ span. \ tree \ for \ K_{n-1}, \ part \ of \ TSP \ solution) & \\
+ \ (2 \ edge \ weights \ meeting \ at \ v, \ part \ of \ TSP \ solution).
\end{align*}
\]
**Definition**

A directed graph (or digraph) \( D \) has vertices \( V(D) \) and directed arcs \( A(D) \) connecting the vertices. It is simple if

- finite vertices
- no arc joining vertex to itself
- not multiple arcs going in same direction.

**Note:** Take directions from arcs in \( D \) gives underlying graph \( U(D) \). Add directions to edges in graph \( G \) gives orientation of \( G, D_G \).

**Example**

\[ D = \quad \text{is a simple digraph with} \quad U(D) = \]
**Bonus - directed graphs**

Many concepts are similar to graphs, e.g. a directed path/cycle is a path/cycle whose directions you follow.

**Definition**

Let $D$ be a digraph. The number of arc ends meeting at a vertex $v$ is called the **degree**, $\text{deg}(v)$. Arcs coming in is the **indegree**, $\text{indeg}(v)$. Arcs going out is the **outdegree**, $\text{outdeg}(v)$.

**Note:** $\text{indeg}(v) = 0$ is a **source**, $\text{outdeg}(v) = 0$ is a **sink**.

**Example**

$$D = \quad \text{has } \text{indeg}(v) = \quad , \quad \text{outdeg}(v) = \quad , \quad \text{deg}(v) = \quad$$
In summary

- Minimal spanning tree
- Kruskal’s algorithm
- Travelling salesperson problem
- Applying minimal spanning trees to find a lower bound for the travelling salesperson problem.
- Bonus - primer on directed graphs

Thanks! See you next time!