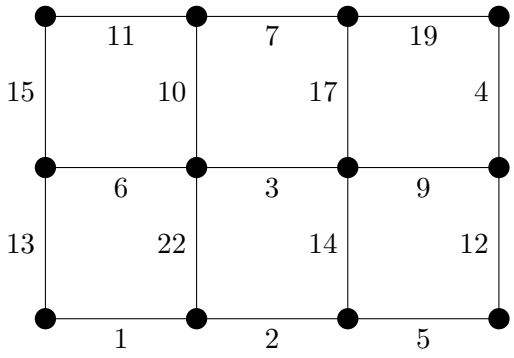
Fall 2016, MATH-566

Spanning Trees

Source: Chapter 2.1 (Bills), Chapter 6.1 of Combinatorial Optimization (Korte)

Problem: Connect cities V with optic cable. For every pair of cities, it is known if the cable can be built and the cost of building it: $c : V^2 \to \mathbb{R}$. Which connections to build to minimize the building cost and make the network connected? (no isolated city)

1: Solve the cities and cable problem for the following diagram of cities:



A *graph* $G = (V, E)$ is a pair of vertices V and edges E, where E consists of pairs of vertices.

Recall definitions of *circuit/cycle*, *tree*, *forest*, *spanning tree*, *connected components*, *path*

Cut for $X \subset V$ is the set of edges with exactly one endpoint in X.

Formal definition of our problem: **Minimum spanning tree problem**
Input: Graph $G = (V, E)$ and costs $c : E \to \mathbb{R}$.
Output: Spanning tree T of minimum cost.

2: If T is a spanning, then the following are equivalent:

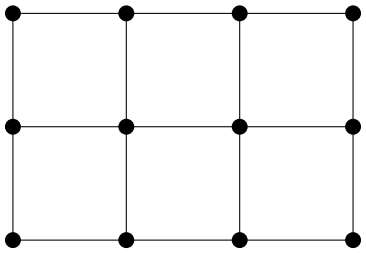
1. T is minimum spanning tree
2. For every $e = \{x, y\} \in E(G) \setminus E(T)$, no edge on the x-y-path in T has higher cost than e.
3. For every $e \in E(T)$, e is a minimum cost edge of the *cut* between the connected components in $T - e$
4. We can order $E(T) = \{e_1, \ldots, e_{n-1}\}$ such that for each i there exists a set $X_i \subset V(G)$ such that e_i is the min cost edge of cut X_i and no previous edge is in the cut X_i.

Show $1 \to 2 \to 3 \to 4$. Try $4 \to 1$ by taking T that satisfies 4 and $T^\star$ satisfying (1) and check how they can differ.

Kruskal's (greedy) algorithm [1956]

1. sort edges of G such that $c(e_1) \leq c(e_2) \leq \cdots \leq c(m)$
2. set $T = (V, \emptyset)$
3. for i in 1 to m:
   if $T + e_i$ does not contain a circuit, then $T := T + e$.

3: Do steps of the algorithm on the graph with cities (note that the algorithm has 11 iterations where edge is added since the tree has 11 edges). Denote the order of edges as they enter the spanning three.

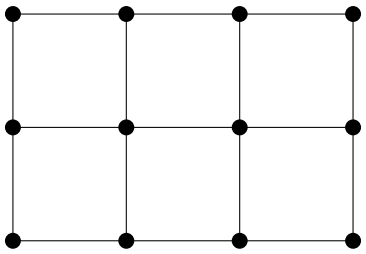


4: Why is the output of the algorithm correct?

Jarník's [1930] and Prims [1957] algorithm

1. choose any $v \in V$ and $T = (\{v\}, \emptyset)$
2. while T does not contain all vertices:
   pick e of minimum cost that has exactly one endpoint in T and $T := T + e$

5: Do steps of the algorithm on the graph with cities (note that the algorithm has 11 iterations since the tree has 11 edges). Denote the order of edges as they enter the spanning three. Start with v being the left top vertex.

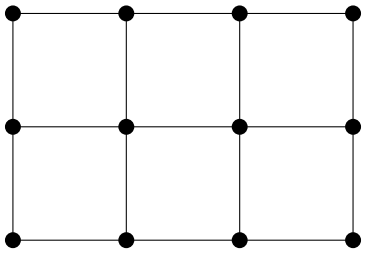


6: Why is the output of the algorithm correct?

Borůvka's [1928] algorithm

1. Let $T = (V, \emptyset)$
2. while T has more than one connected component:
   in parallel, for every connected component C in T, pick e of minimum cost that has exactly one endpoint in C and do $T := T + e$

7: Do steps of the algorithm on the graph with cities. Note that one iteration always gives several edges in. The number of iterations is not clear at the beginning. Denote the order of edges as they enter the spanning three.



8: Why is the output of the algorithm correct?

Algorithmic note: Which algorithm is fastest?

Complexity of algorithm counted in number of operations of CPU.

Count how many times every vertex and edge is used, constant does not matter, we use $O(.)$ notation.

Simple Kruskal's complexity for graph with m edges and n vertices:

- sorting takes $O(m \log(m))$
- for cycles is done m times
- test for circuit can be done in $O(n)$ time

Total time: $O(m \log(m) + mn)$. Better implementation can do $O(m \log(n))$.

Jarník's algorithm can be implemented in $O(m + n \log(n))$.

More effective algorithms exists if weights are integers, graph is *planar*, ...

9: Let $G = (V, E)$ be a a graph and $c : E \to \mathbb{R}$ cost function on the edges. Formulate the minimum spanning tree problem using linear programming.
(Don't be afraid that there are many constraints. Try to make constraint that graph has no cycles.)

Theorem [Edmonds 1970] The set of feasible solutions is *integral* polytope - i.e. - every vertex of the polytope has all coordinates integers. The polytope is called **spanning tree polytope**.

10: Draw the spanning tree polytope for K_3, where K_3 is the complete graph on 3 vertices.