Logistics systems planning I
Optimization of logistics systems
Transportation planning – Minimum spanning tree problem

Univ.-Prof. Dr. Michael Schneider

Deutsche Post Chair – Optimization of Distribution Networks (DPO)
RWTH Aachen University

schneider@dpo.rwth-aachen.de

Deutsche Post
Chair - Optimization of
Distribution Networks
RWTH AACHEN
UNIVERSITY

Course agenda

1. Fundamentals
2. Transportation planning
   - Introduction
   - Shortest path problems
   - **Minimum spanning tree problem**
   - Traveling salesman problem
   - Vehicle routing problems
   - Arc routing problems
3. Warehouse planning
4. Introduction to location planning

Goals of the section:

- Define the problem
- Identify situations in which minimum spanning trees are sought
- Understand optimality conditions
- Understand and manually carry out algorithms of Kruskal and Prim

Agenda

1. Minimum spanning tree problem
   - Definition, applications, optimality conditions
   - Kruskal's algorithm
   - Prim's algorithm

Spanning trees

Spanning tree

In a connected graph (V, E), a spanning tree is a connected acyclic subgraph (W, T) with $W = V$ and $T \subseteq E$.

(V, T) is a spanning tree of (V, E)

$\Leftrightarrow$ (V, T) is a connected subgraph with $|V| - 1$ edges

$\Leftrightarrow$ (V, T) is an acyclic subgraph with $|V| - 1$ edges

$\Leftrightarrow$ for each pair of nodes in V, there exists a unique path in (V, T) connecting the nodes

$\Leftrightarrow$ adding an arbitrary edge to (V, T) generates exactly one elementary cycle

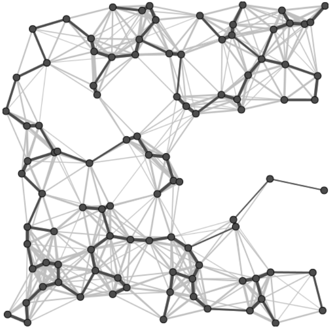
$\Leftrightarrow$ (V, T) decomposes into exactly two components if an arbitrary edge from T is removed

The minimum spanning tree problem

Minimum spanning tree (MST)

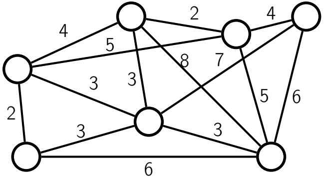
In a connected weighted graph (V, E, c_{ij}), a spanning tree (V, T) is called minimum spanning tree if $c(T) = \sum_{\{i,j\} \in T} c_{ij}$ is minimum for all spanning trees.

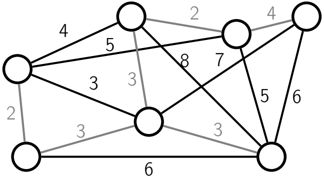
Minimum spanning tree problem – applications



- Applications: Construction of distance or cost-minimal networks, e.g.,
  - Pipeline and other supply networks (gas, water, electricity, oil, ...) connecting the source with the consumers
  - Communication networks, e.g., wiring in computer networks
  - Shortest street network for connecting a set of locations

The minimum spanning tree problem – Example





$\Rightarrow \; c(T) = 2 + 2 + 3 + 3 + 3 + 4 = 17$

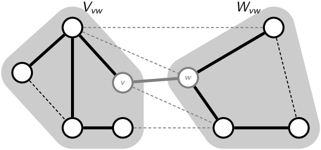
The minimum spanning tree problem

When determining shortest paths of a node s to all other nodes in a graph, the result was a so-called *shortest paths tree*.

- Can you solve the shortest path problem in an undirected graph by determining an MST?
- Can you solve the MST problem by determining the shortest paths tree?
- Are the objective functions of MST and shortest path problem identical?
- Does it matter that we assume an undirected graph for the MST problem?

Cut optimality conditions I

- Assume we remove an edge $\{v, w\} \in T$ from a spanning tree (V, T)



- The tree decomposes into two components
  - the node sets of the components are V_{vw} and W_{vw}
  - the cut-set S_{vw} consists of all edges with one end node in V_{vw} and the other one in W_{vw}:

$$S_{vw} = \{\{i,j\} \in E | i \in V_{vw} \text{ and } j \in W_{vw}\}$$

- The spanning tree can only be an MST, if no other edge $\{i,j\} \in S_{vw}$ is shorter than $\{v, w\}$ (if a shorter edge existed, we could use it to replace $\{v, w\}$). It necessarily holds:

$$c_{ij} \geq c_{vw} \quad \text{for all } \{i,j\} \in S_{vw}$$

Cut optimality conditions II

The conditions are also sufficient.

Cut optimality conditions

Given a weighted graph $G = (V, E, c_{ij})$ and a spanning tree (V, T). (V, T) is an MST

if and only if

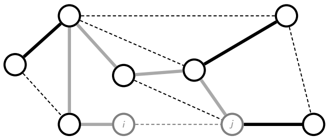
for each edge $\{v, w\} \in T$ it holds

$$c_{ij} \geq c_{vw}$$

for all edges $\{i, j\} \in S_{vw}$.

Path optimality conditions

- Assume we add an edge $\{i, j\} \notin T$ into a spanning tree (V, T)



- This leads to an elementary cycle
  - the nodes i and j are connected through a unique path in T
  - this path P_{ij} and the edge $\{i, j\}$ form an elementary cycle
- The spanning tree can only be an MST, if no edge $\{v, w\} \in P_{ij}$ is longer than $\{i, j\}$ (if there was a longer edge, we could exchange it with $\{i, j\}$). Necessarily, it has to hold:

$$c_{vw} \leq c_{ij} \quad \text{for all } \{v, w\} \in P_{ij}$$

Path optimality conditions

These conditions are also sufficient.

Path optimality conditions

Given a weighted graph $G = (V, E, c_{ij})$ and a spanning tree (V, T). (V, T) is an MST

if and only if

for each edge $\{i, j\} \notin T$

$$c_{vw} \leq c_{ij}$$

holds for all edges $\{v, w\}$ of the unique path in T from i to j.

Agenda

1. Minimum spanning tree problem
   - Definition, applications, optimality conditions
   - Kruskal's algorithm
   - Prim's algorithm

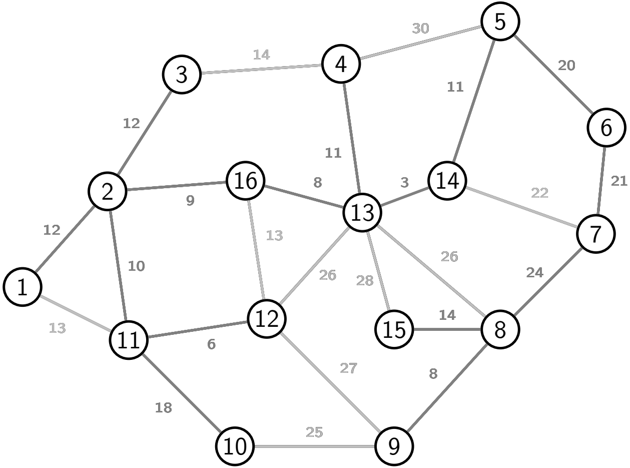
Kruskal's algorithm

- The MST problem can be solved efficiently with the following greedy algorithm

Algorithm 1: Kruskal's algorithm

```
// connected weighted Graph (V, E, c_ij)
SET T := ∅
SORT edges by increasing cost c_ij
for {i,j} ∈ E according to sorting do
    if inserting {i,j} doesn't lead to a cycle in (V, T) then
        T := T ∪ {{i,j}}
    if |T| = |V| − 1 then
        STOP
// Output: edges T of the MST (V, T)
```

Kruskal's algorithm – Example



Kruskal's algorithm

Why is the spanning tree (V, T) constructed with Kruskal's algorithm minimum?

- Consider any edge $\{i, j\} \notin T$
  - $\{i, j\}$ was not added to the graph because it would have formed a cycle with the previously added edges
  - because of the sorting of the edges, all of theses edges are shorter (or of the same length)

  $\rightarrow$ the path optimality conditions are satisfied for $\{i, j\}$
- Because this argument holds for all edges $\{i, j\} \notin T$, path optimality conditions are entirely satisfied

Kruskal's algorithm

Note:

- How to check efficiently whether adding an edge leads to a cycle
- Idea:
  - save the connected components of T as sets of nodes (note that they form a forest). Example: for a graph with 6 vertices and $T = \{\{1,5\},\{3,5\},\{4,6\}\}$, the connected components are $\{\{1,3,5\},\{2\},\{4,6\}\}$
  - if i and j are in different node sets, adding the edge $\{i,j\}$ does not lead to a cycle
- Efficient implementations use so-called *union-find algorithms*, which are able to carry out the required operations (merging sets and testing whether two elements are in the same set) very fast

Agenda

1. Minimum spanning tree problem
   - Definition, applications, optimality conditions
   - Kruskal's algorithm
   - Prim's algorithm

Prim's algorithm

- The MST problem can be solved efficiently with the following greedy algorithm

---

Algorithm 2: Prim's algorithm

`// Input: connected weighted Graph` (V, E, c_{ij})

SET $S := \{v_0\}$ for any node $v_0 \in V$

while $S \neq V$ **do**

  DETERMINE $\{i, j\} \in \delta(S)$ $(i \in S,\, j \notin S)$

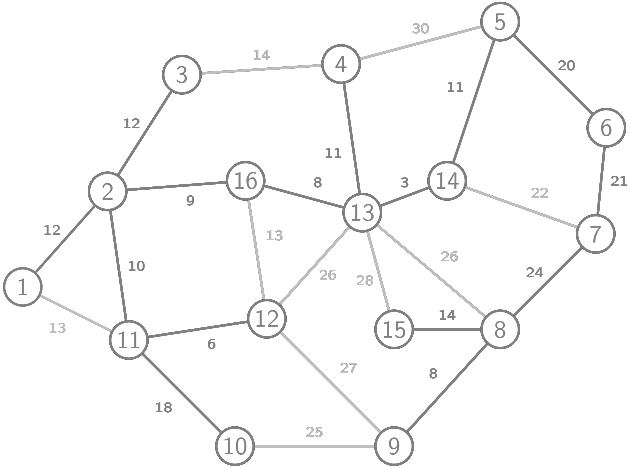
    with $c_{ij} = \min_{\{v,w\} \in \delta(S)} c_{vw}$

  SET $S := S \cup \{j\}$

`// Output: Edges` T `of MST` (V, T)

- The correctness of Prim's algorithm can be proven based on cut optimality conditions (Ahuja et al. (1993), p.519 and p.523)

Prim's algorithm – Example



Want to learn more?

Kruskal's algorithm:

- https://www-m9.ma.tum.de/graph-algorithms/mst-kruskal/index_de.html
- https://courses.cs.washington.edu/courses/cse373/16wi/Hashing/visualization/Kruskal.html

Prim's algorithm:

- https://www-m9.ma.tum.de/graph-algorithms/mst-prim/index_de.html
- https://visualgo.net/en/mst

Descriptions:

- https://www.youtube.com/watch?v=GJ17vvqY6aE
- http://de.wikipedia.org/wiki/Algorithmus_von_Kruskal

Outlook

MST problems are often used in algorithms for other optimization problems:

- as a relaxation of the TSP
- as a component in Christofides tree heuristic for the TSP
- as a component of a heuristic for the rural postman problem

R. K. Ahuja, T. L. Magnanti, and J. B. Orlin. *Network Flows: Theory, Algorithms, and Applications*. Prentice Hall, Englewood Cliffs, New Jersey, 1993. ISBN 0-13-617549-X.