GEO INFORMATION

Methods in Spatial Analysis
PS | LV.Nr. 856.141

Paris-Lodron University Salzburg
Department of Geoinformatics – Z_GIS

Johannes Scholz

TU Graz, Institute of Geodesy
Research Group Geoinformation

johannes.scholz@tugraz.at | johannes.scholz@plus.ac.at
ifg.tugraz.at || www.johannesscholz.net
@Joe_GISc @Joe_GISc@mastodon.online

PS Overview and Contents

Date	Where & When	Content
10-Oct-2023	GI-Lecture | 0900-1500	Administration, Introduction to Spatial Analysis, Basics
17-Oct-2023	GI-Lecture | 0900-1500	Distance Analysis Selection & Aggregation
24-Oct-2023	Online | 0900-1100	Network Analysis
07-Nov-2023	Online | 0900-1100	Spatial Interpolation
14-Nov-2023	Online | 0900-1100	Terrain Analysis
21-Nov-2023	Online | 0900-1100	Visibility Analysis
28-Nov-2023	Online | 0900-1100	Calculating Solar Potential
05-Dec-2023	Online | 0900-1100	Hydrological Modelling (Surface Runoff)
12-Dec-2023	Online | 0900-1100	Weighted Overlay
09-Jan-2024	GI-Lecture | 0900-1500	Cost Surface Analysis, Concluding Remarks

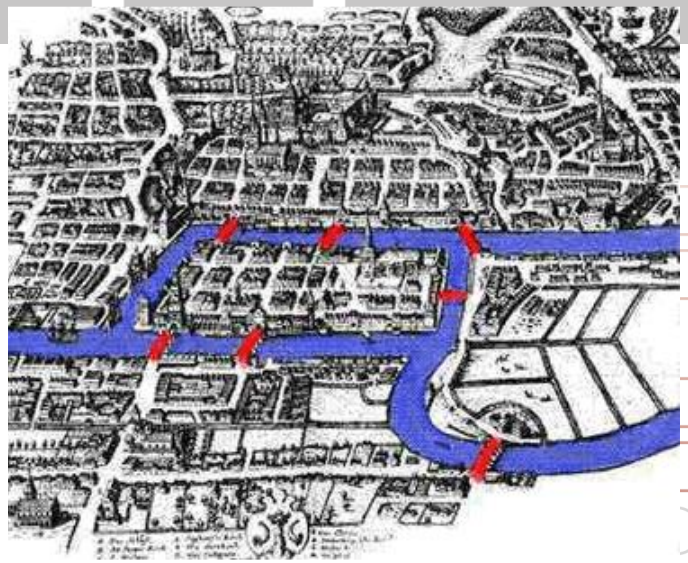
Networks

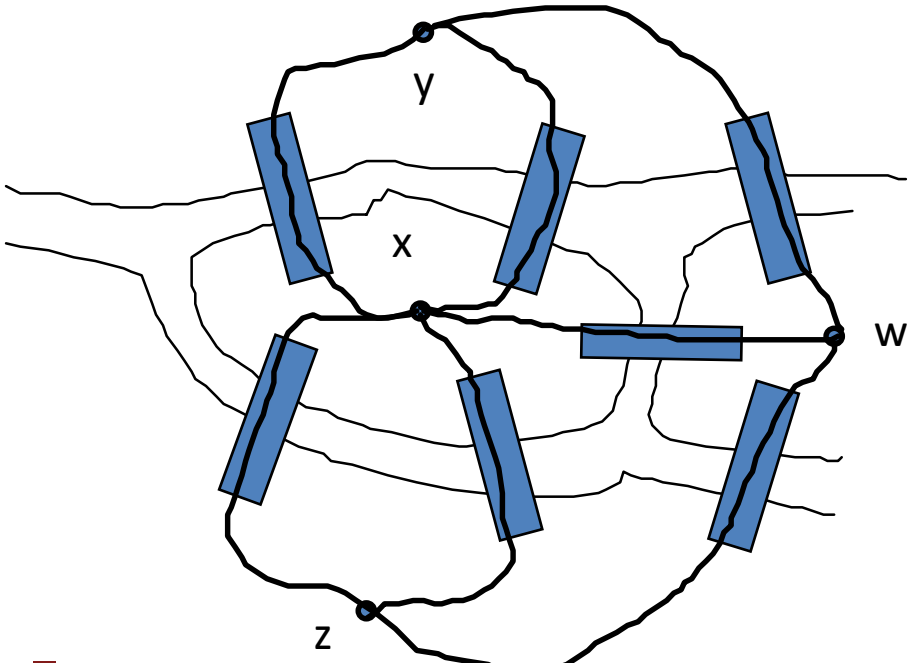
- Important for the following application areas:
  - Supply infrastructure: water, electricity, oil/gas pipeline, sewer system, ….
  - Transport infrastructure
  - Telecommunication networks

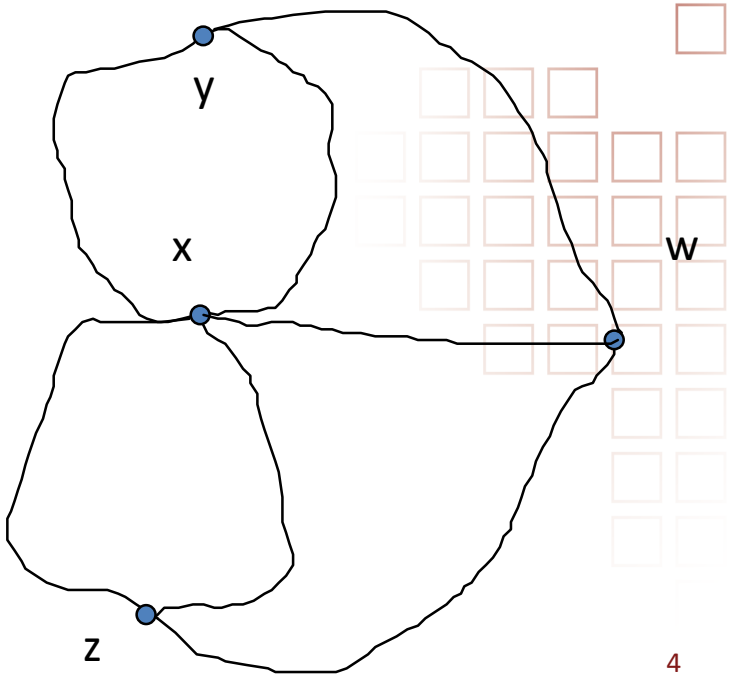
→ Collection of simple geometries (point, line) requires a theoretical concept: **GRAPHS**

Network :: Graphs

- Leonard **Euler**, solved the **Seven Bridges of Königsberg problem in 1736**;
  Euler laid the foundations of graph theory and topology.







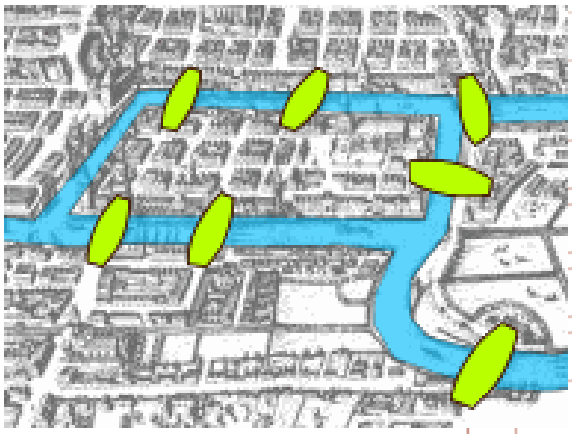
Allgemeines

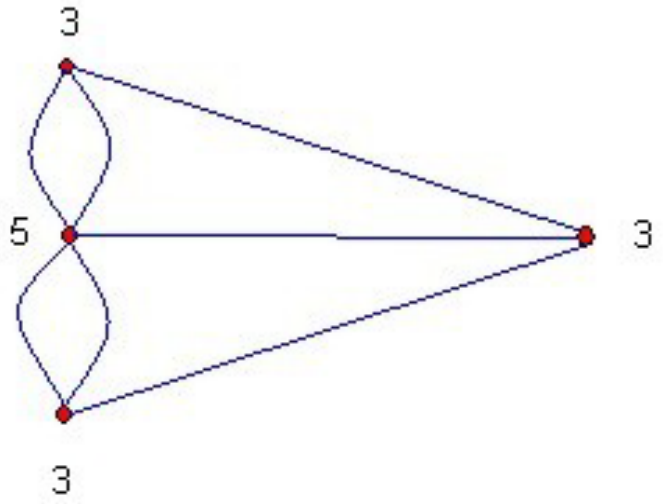
- **Königsberger Brückenproblem cont‘d**

  - Warum ist ein Euler‘scher Weg nicht möglich?

  >> zu allen Ufergebieten / Inseln (i.e. Knoten) führt eine ungerade Anzahl an Brückenl

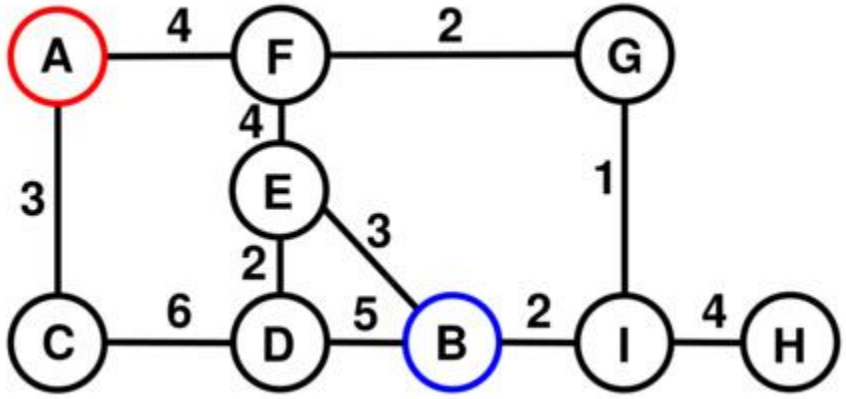
  >> es darf maximal zwei Ufer/Inseln/Knoten mit einer ungeradem Anzahl an Brücken geben – der Rest muss gerade sein!





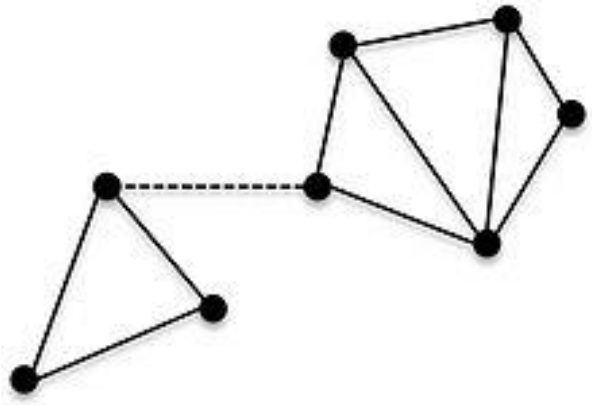
Important Definitions

- Definitions:
  - Vertex/Node
  - Edge/Arc
    - Directed vs. undirected
  - Graph
    - Contains of nodes & vertices
  - Network:
    - Ordered pairs of nodes and edges with attributes (e.g. weights)
  - Weights: distance, driving time, energy consumpotion, …
- Examples:
  - Road
  - Infrastructure (electricity, water, ….)
  - Railway
  - …



Prerequisites

- Topological „clean“ Graph (Arc-Node Topology)
  - Nb! Connectivity



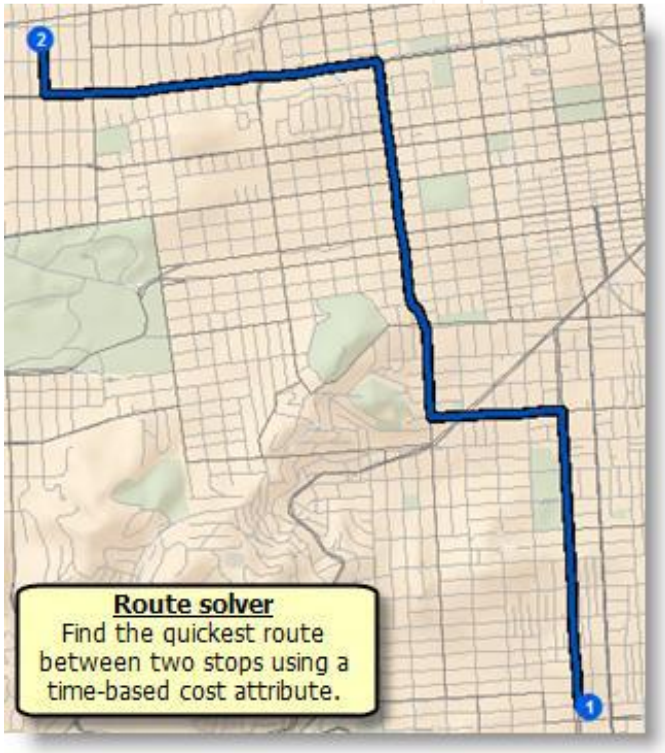
- Definition of any problem in a mathematical language!
- Algorithm to solve a given problem

Network analysis

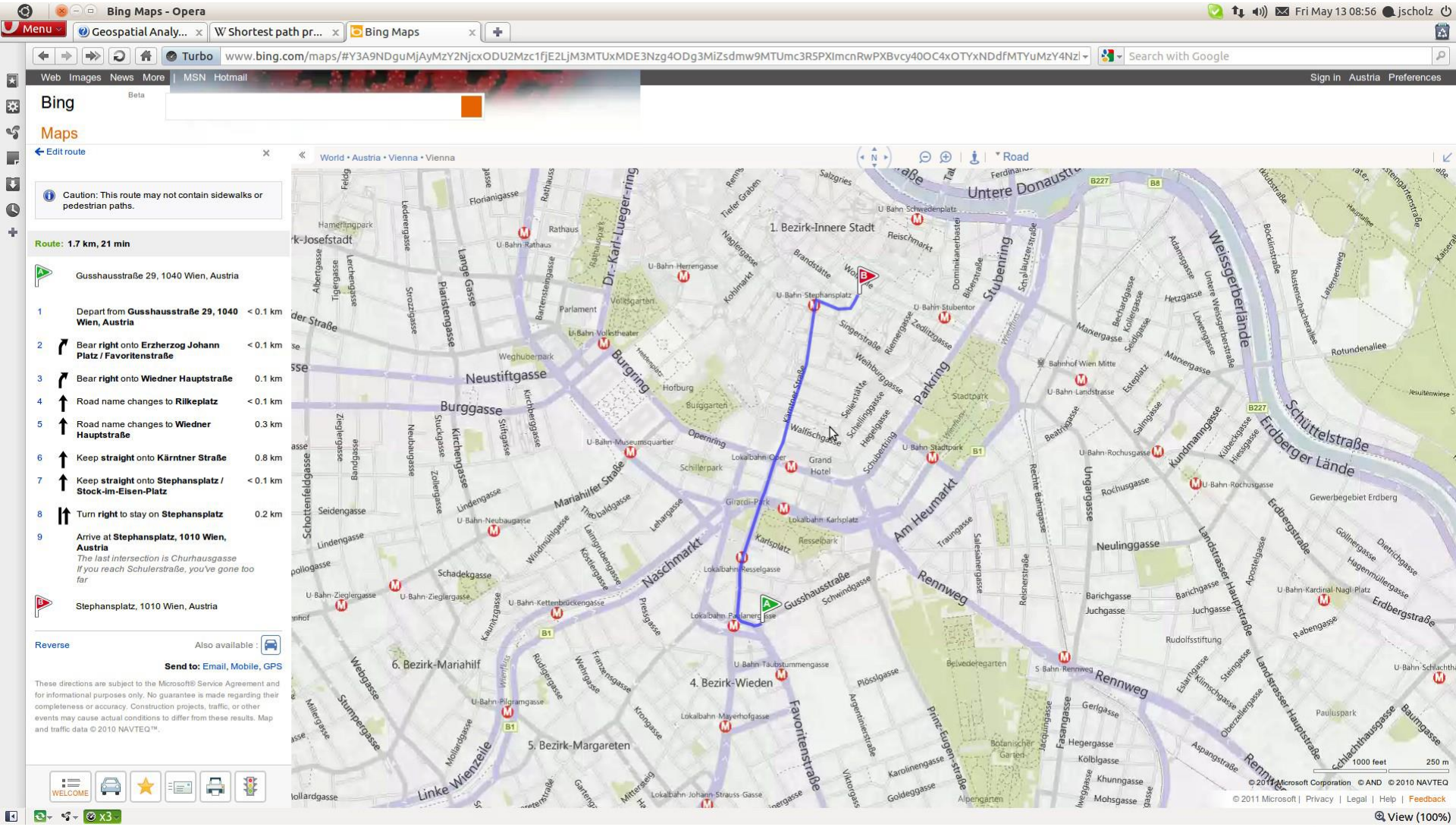
- Routing/shortest Path Problem
- Traveling Salesman Problem (TSP)
- Vehicle Routing Problem (VRP)
- OD-Cost Matrix
- Location and Service Area Problems
  - Service Area
  - Closest Facility

Routing/shortest Paths

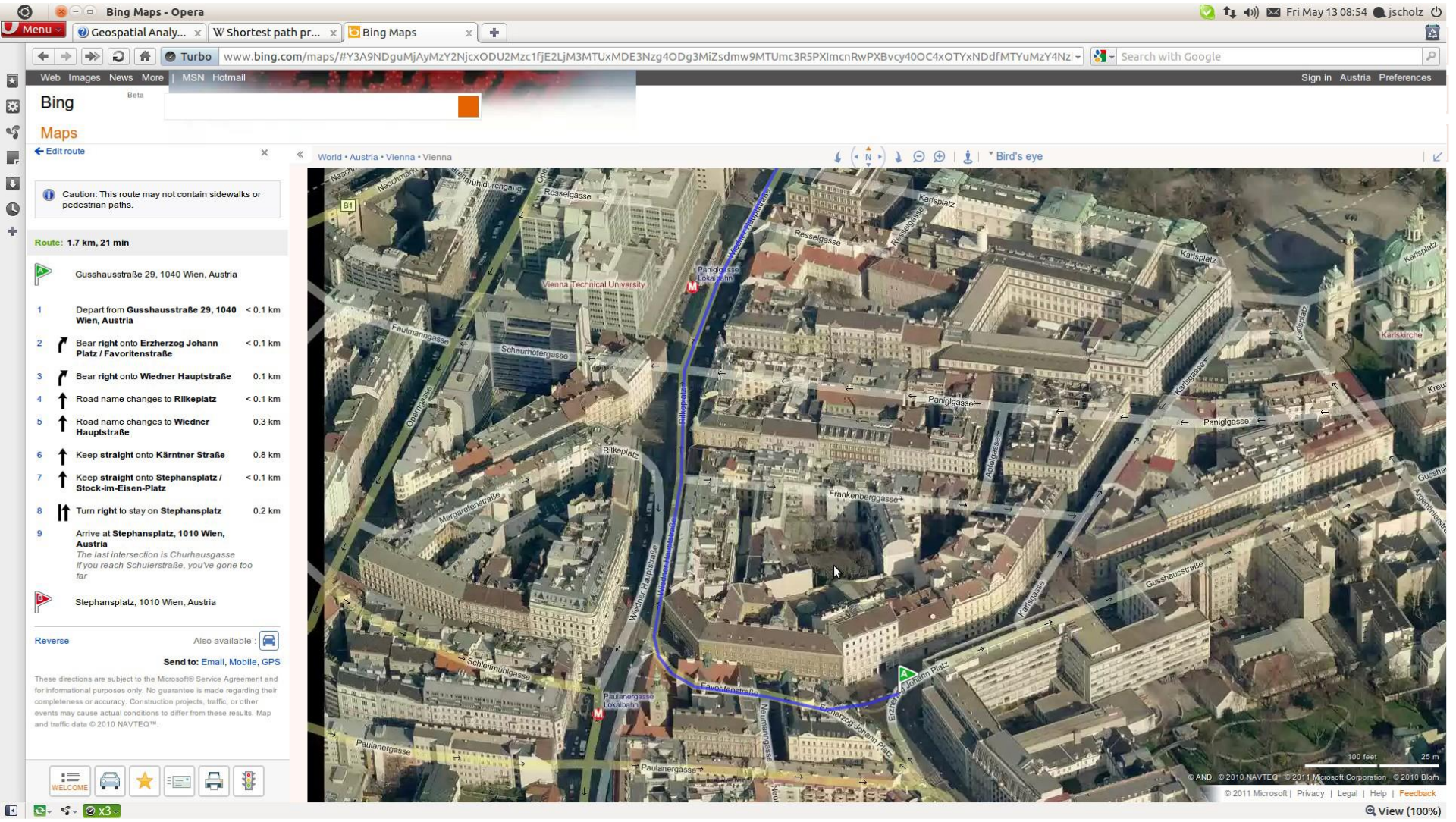
- Objective: Calculation of a path between start node and target node with respect to a metric
  - time
  - distance
  - …
- example:
  - SatNav
  - Route planning (Google, Bing, …)



Routing/shortest Paths



Routing/shortest Paths



Routing/shortest Paths

- Solution Algorithms:
  - Dijkstra Algorithmus (Dijkstra 1956)
  - A* Algorithm
  - Bellman Ford Algorithm

- Dijkstra Algorithm (an overview)
  1. Assign to every node a distance value: set it to zero for our initial node and to infinity for all other nodes.
  2. Mark all nodes as unvisited. Set initial node as current.
  3. For current node, consider all its unvisited neighbors and calculate their tentative distance. For example, if current node (A) has distance of 6, and an edge connecting it with another node (B) is 2, the distance to B through A will be 6+2=8. If this distance is less than the previously recorded distance, overwrite the distance.
  4. When we are done considering all neighbors of the current node, mark it as visited. A visited node will not be checked ever again; its distance recorded now is final and minimal.
  5. If all nodes have been visited, finish. Otherwise, set the unvisited node with the smallest distance (from the initial node, considering all nodes in graph) as the next "current node" and continue from step 3.

Routing/shortest Paths

- **Dijkstra Algorithm**
  - Solution to single source shortest path problem
  - Graph shall not have negative weights!

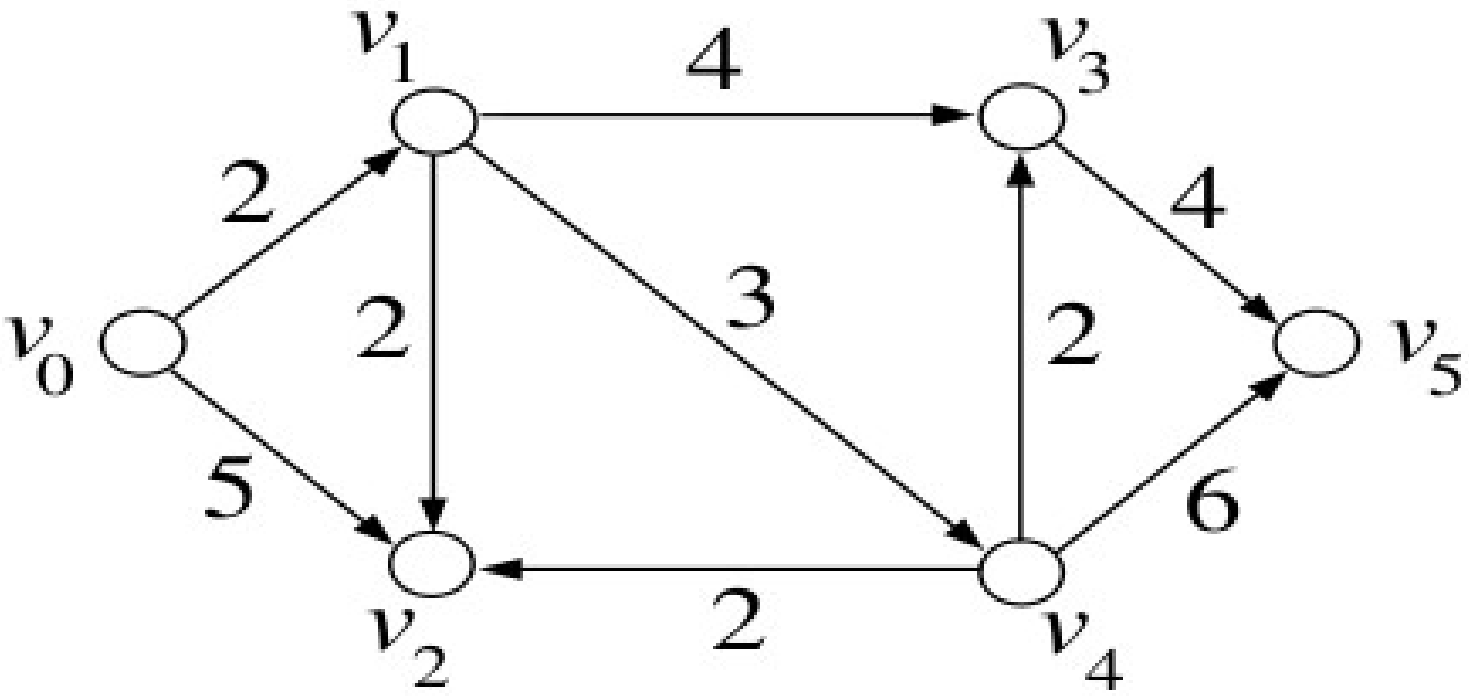
1. $l(v_0) := 0$; for $v \in V \setminus \{v_0\}$ do $l(v) := \infty$ end; $U := \{v_0\}$; $u := v_0$;

2. for $v \in V \setminus U$ do:
   if $(u, v) \in E$ and $l(v) > l(u) + w(u, v)$ then
   $p(v) := u$;
   $l(v) := l(u) + w(u, v)$;
   end if;

3. $m := \min_{v \in V \setminus U} l(v)$, wähle Knoten $z \in V \setminus U$ mit $l(z) = m$;
   $U := U \cup \{z\}$;
   $u := z$;

4. if $U = V$ or $\forall v \in V \setminus U : l(v) = \infty$ then END
   else goto 2;

Routing/shortest Paths



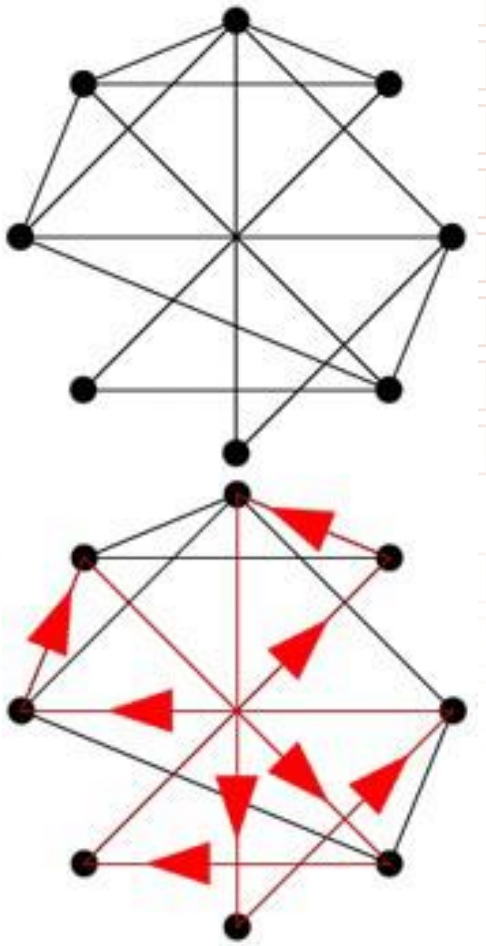
Routing/shortest Paths

- **A* Algorithm**
  - Solution to the single source shortest path problem.
  - Graph shall not have negative weights!
  - Basic Idea:
    - Investigation only of those nodes that seem successful/fruitful (e.g. that lead to faster solution time - heuristics)
    - Other than that similar to Dijkstra

- **Bellman-Ford Algorithm**
  - „copes“ with **negative weights**

Traveling Salesman Problem

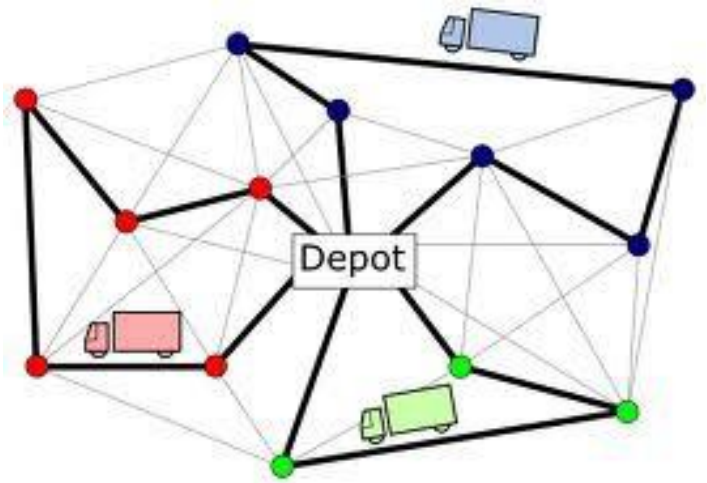
- Problem:
  - 1 traveling salesman with a given hometown
  - Several cities need to be visited
- Objective:
  - Determination of a path that visits all cities exactly once
  - Optimality: shortest total travel distance or time

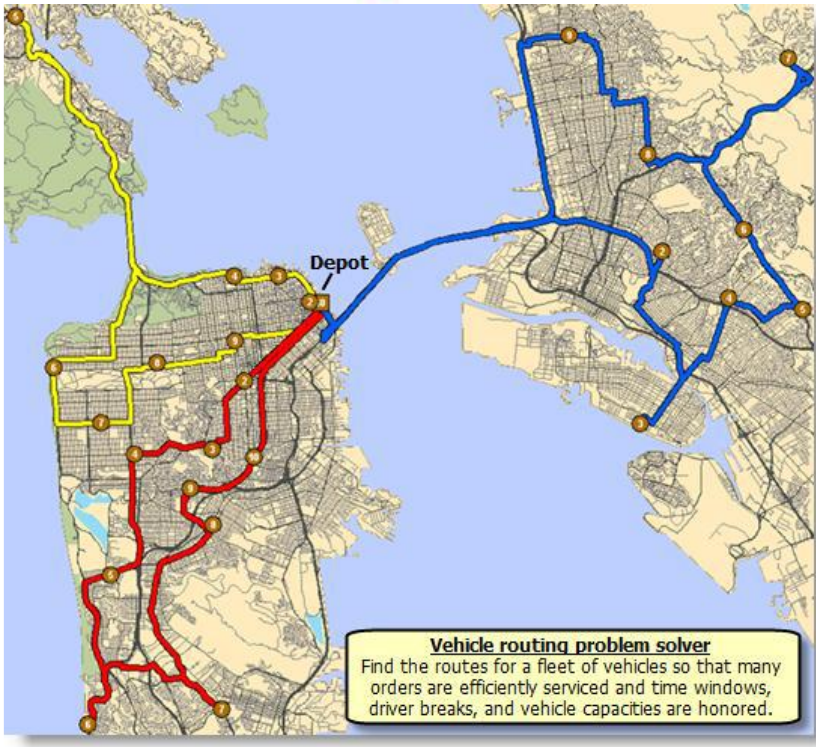


Vehicle Routing Problem

- Problem:
  - Several customers that need to be visited
  - Several vehicles available that visit the customers

  >> each customer is visited by exactly one vehicle exactly once
- Objective
  - Determination of routes for each vehicle
  - Optimality: Sum of travel distances or travel time is minimal
  - … each customer is visited

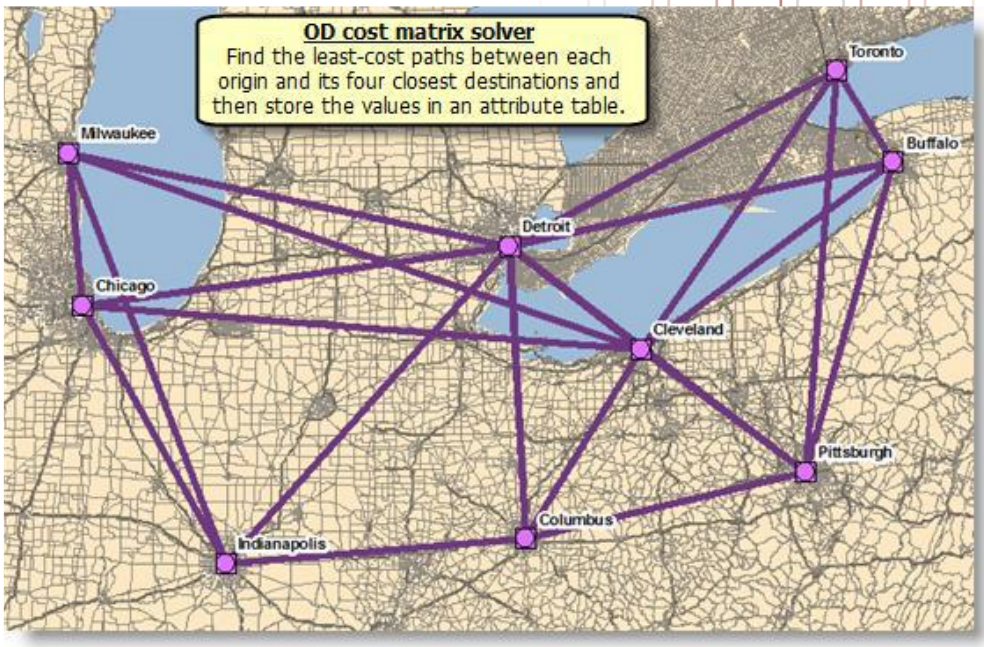




OD Cost Matrix

- Origin-Destination (OD) Cost Matrix
- Problem:
  - Huge number of origins & destinations >> high complexity

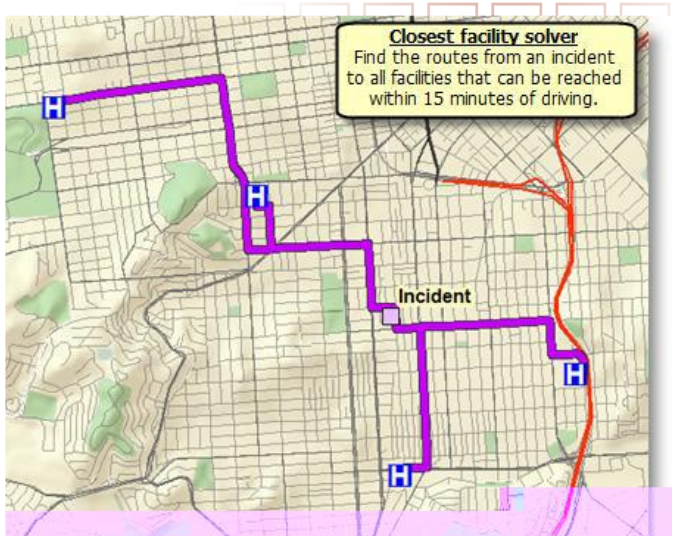
- Objective:
  - Calculation of the shortest path between each OD pair



Location Problems & Service Areas

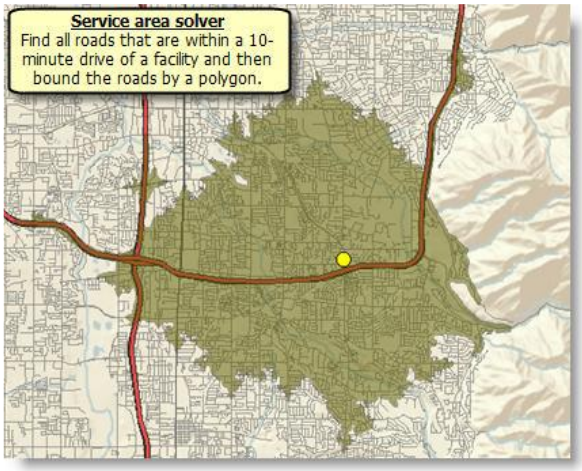
- Location Problems
  - Try to compute the optimal location for an object in space
  - >> Decision Support
  - zB depot, service points, supermarket, antenna locations …

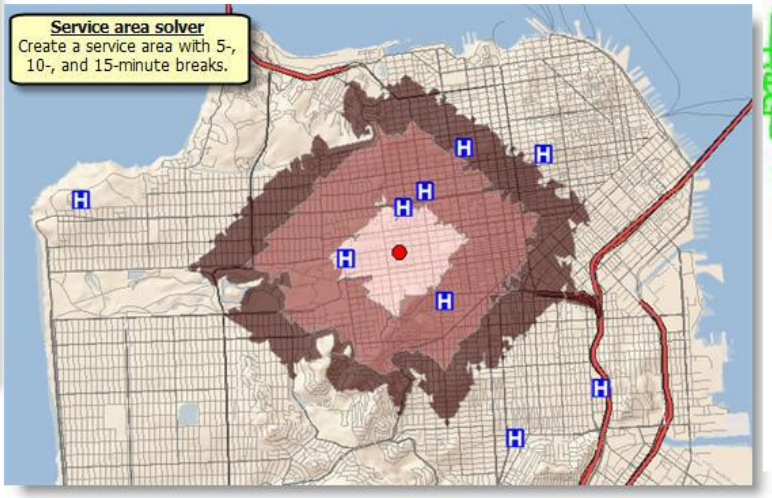
- Closest Facility
  - Calculates which objects are reachable from a single point in space
  - Often with respect to a defined distance/time



Location Problems & Service Areas

- Service Area
  - Definition of zones that are reachable from defined points in space
  - Often with respect to a time or distance value
  - E.g. houses that are reachable within 15min of the fire station





GEO INFORMATION

Methods in Spatial Analysis
PS | LV.Nr. 856.141

Paris-Lodron University Salzburg
Department of Geoinformatics – Z_GIS

Johannes Scholz

TU Graz, Institute of Geodesy
Research Group Geoinformation

johannes.scholz@tugraz.at | johannes.scholz@plus.ac.at
ifg.tugraz.at || www.johannesscholz.net
@Joe_GISc @Joe_GISc@mastodon.online