Geographic Information Technology Training Alliance (GITTA) presents:

Accessibility (Network Analysis)

Responsible persons: Helmut Flitter, Thomas Grossmann

Table Of Content

1. Accessibility (Network Analysis)

The properties of objects along with the relationships between these objects are of interest in spatial analysis. As discussed in the "Spatial Queries" lesson, various relationships between objects can be reviewed. As a basis, thematic (or semantic), spatial or temporal relationships can be detected. Spatial relationships can be further divided into: topological, distance, and directional relationships. In this lesson, the main focus will be on distance relationships. Using methods designed for calculating distances or proximities, one can answer questions such as:

- What is the nearest railway station?
- How many pharmacies are within 300m of a specific location?
- What is the best residential area where the entire route between nursery, school and shopping facilities will be minimal?
- How many residents live in the catchment area of a shopping center?

The measures of distance we have discussed so far were unhindered in their extent and unrestricted in their direction. However, most movements in geographical space are limited to linear networks. In many cases, uninhibited movement is not possible. Even flight paths are limited to corridors. Most movement follows fixed channels: transportation (see figure), pipelines, telephone wires, rivers etc. Networks are of general importance for all areas of spatial science. The analysis of network structures is an important task particularly in the planning area. Network analysis could be about:

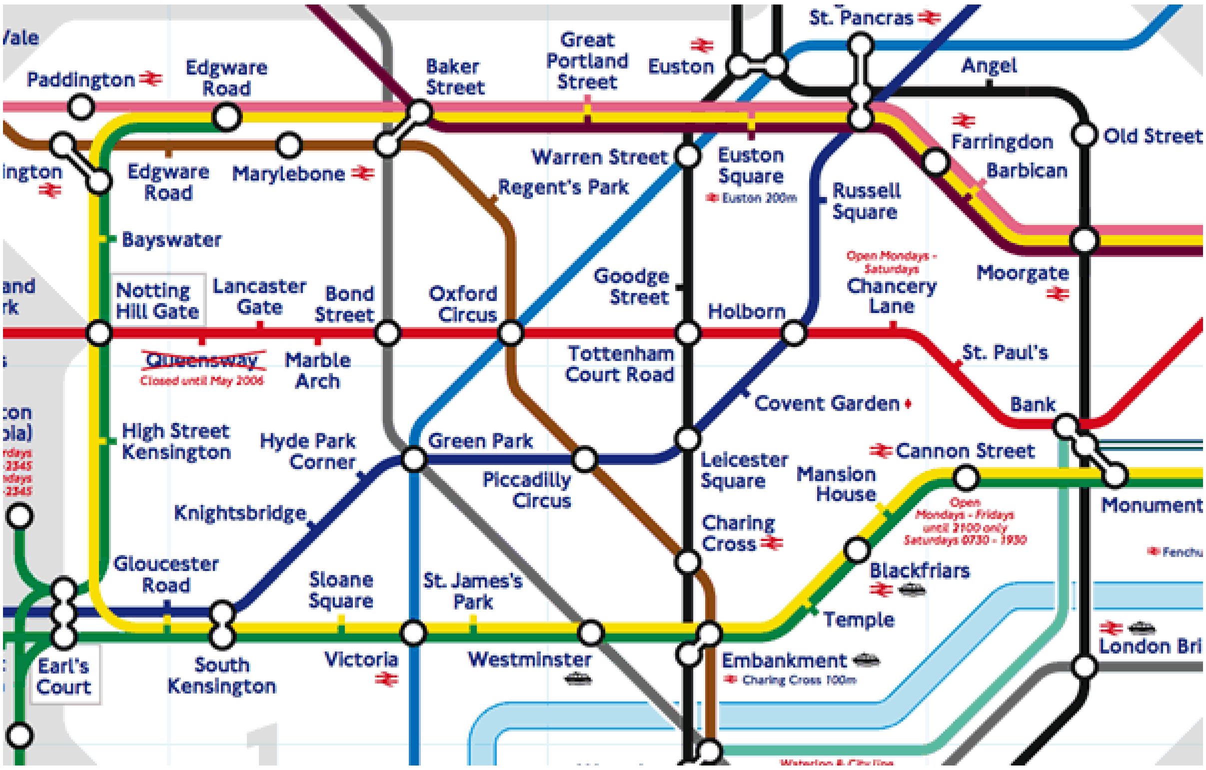
- Optimization of networks: improving the transport infrastructure through additional routes in the suburban rail network
- Best route selection: planning trash collecting rounds; resource scheduling for emergency services
- Classification of catchment areas: delineation of fire districts based on accessibility to road network
- Ideal placement in network: optimally positioning supply centers in the network, ie. Locating and allocating for supply and demand

A prerequisite for the analysis of networks is the analytical description and understanding of network structures. Generally, it is about accessibility of objects. You can find answers to questions like:

- Structural properties of a network: how dense (well connected) is a network?
- Accessibility of places: how well connected is place x compared to place y (how often do you need to transfer)?
- Location in the network: what are the central places (i.e. appropriate transfer stations)?

Network analysis and description in based on graph theory. With graph theory, networks can be described in a more abstract and general way as graphs.

Example: For many geographical problems, or even in everyday life, it is not necessary to know exact coordinates (*xi*, *yi*). To get from one node to another node in a network, it is most important to know the connections between them. The map (detail) of the London subway system contains all the useful information to get from station *i* to station *j*. This topologial representation allows us to see how easy it is to travel between stations that are not adjacent and to find the stops where we need to change.



Detail of the map of the London tube. Click on map to see the whole map (Transport for London 2005)

Learning Objectives

- You know the essential concepts to characterize a network or a graph
- You are able to list simple measures for topological and geometrical description of networks, to explain them, and to give examples for their application
- You know the most used and famous algorithm, which calculates the shortest path between two points
- You know the problem of the traveling salesman and can explain a heuristic solution
- You can describe the different steps of both algorithms and manually calculate a route for simple cases.

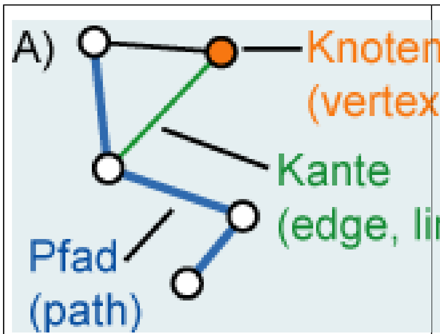
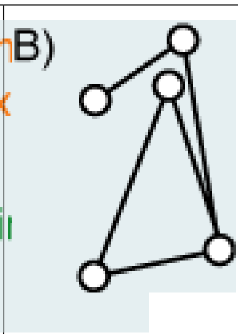
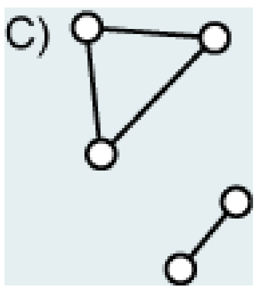
1.1. What are networks

To understand the theory of networks, background knowledge about basic elements and various characteristics of networks is needed. In this unit, we will show how networks are composed and what kind of networks there are, respectively. The following concepts are introduced:

- Graphs
- Elements of graphs: nodes and edges
- Directed (asymmetric) or undirected (symmetric) edges
- Planar and nonplanar graphs
- Adjacency matrix

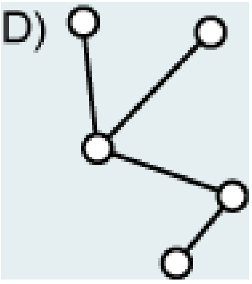
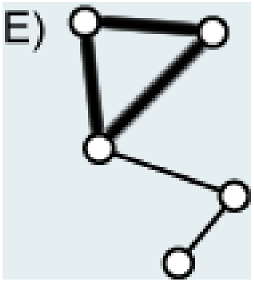
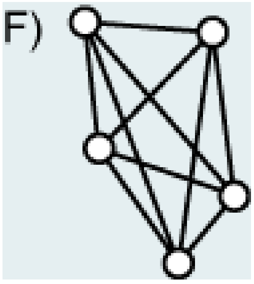
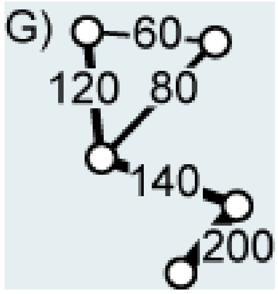
1.1.1. Primitives of a Network

Terms of graph theory and networks
In graph theory, distinct terminology may apply. However, most terms have simple definitions. A graph consists of two types of elements, namely vertices (*V*) and edges (*E*) (e.g.: Graph A). Vertices represent objects that can have names and other attributes such as, for example, transfer points in a subway system. Every edge has two endpoints in the set of vertices, and is said to connect or join the two endpoints (they are adjacent), e.g. a flight connection between Zurich and Berlin. Graphs are represented by drawing a dot or circle for every vertex, and drawing an arc between two vertices if they are connected by an edge (Graphs A-K). A graph is defined independently from its visualisation. Both figures A and B display the same graph.
A path from vertex *s* to vertex *e* in a graph is a list of adjacent vertices. A simple path is a path where no vertex is used more than once. In a connected graph there is an edge from each vertex to every other vertex. In a disconnected graph, there are disconnected parts of edges and nodes (Graph C).

		
Elements of a graph	Connected graph	Disconnected graph

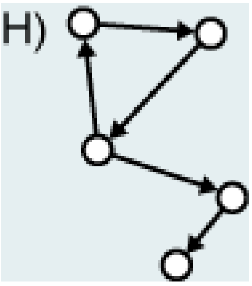
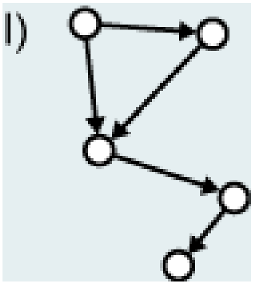
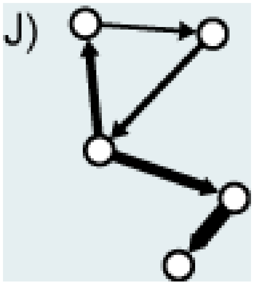
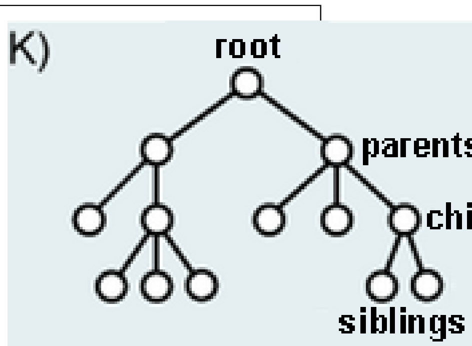
A cycle is a simple path where the first and last vertex is the same (Graph E). A graph without cycles is called a tree (Graph K). Trees have a hierarchical structure and are often explicitly displayed.
The complete graph is a simple graph in which each vertex is adjacent to every other vertex (Graph F).
Graphs are considered as sparse when they have a relatively small number of edges; graphs with a small number of possible edges missing are described as dense. In weighted graphs (Graph G), a number or a weight is assigned to each edge of the graph to represent distance (temporal or geometric) or cost. That way, more information is linked to the graph.

Undirected Graphs:

			
	Cycle	Complete graph	Weighted graph

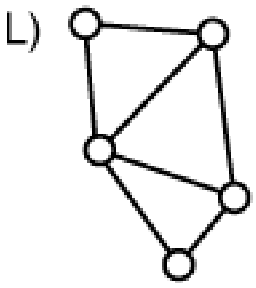
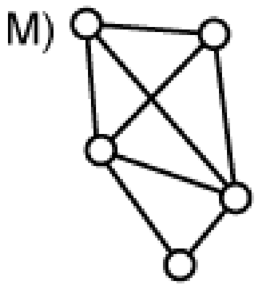
In directed graphs, edges are "one-way streets" (Graph H); an edge can lead from *x* to *y* but not from *y* to *x*. Directed graphs without directed cycles (directed cycles: all edges point in the same direction) are called directed acyclic graphs (Graph I). Directed and weighted graphs are called networks (Graph J). In colloquial language and in the field of geography the term net or network is often used for all kinds of graphs.

Directed Graphs:

			
Graph with a cycle	Acyclic	Directed and weighted = Network	Hierarchic

There is another categorization: **planar** and **non-planar** graphs. A planar graph is one which can be drawn on the plane without any lines crossing (Graphs L and M).

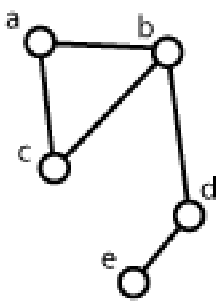
Planar vs. non-planar graphs:

	
Planar	Non-planar

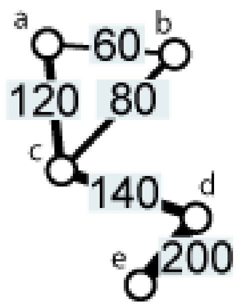
Another simple form of representation for graphs, which can also be processed by a digital computer, is an adjacency matrix. A matrix of size *VxV* is designed, where *V* is the number of nodes. The fields are set to 1 if an edge between the nodes exists, for example, *a* and *b*, and to 0 if no such edge exists. In the example we assume that an edge from each node to itself exists. Whether you set the size of the diagonal to 1 or 0, depends on the intended purpose. In some cases it is better to set the diagonal to 0. This matrix is called adjacency matrix because its structure indicates which nodes are neighbouring (i.e. are adjacent). Something similar is possible with weights: the weights (instead of the value 1) are written into the fields.

	from				
to	a	b	c	d	e
a	1	1	1	0	0
b	1	1	1	1	0
c	1	1	1	0	0
d	0	1	0	1	1
e	0	0	0	1	1

	from				
to	a	b	c	d	e
a	0	80	60	0	0
b	80	0	120	140	0
c	60	120	0	0	0
d	0	140	0	0	200
e	0	0	0	200	0

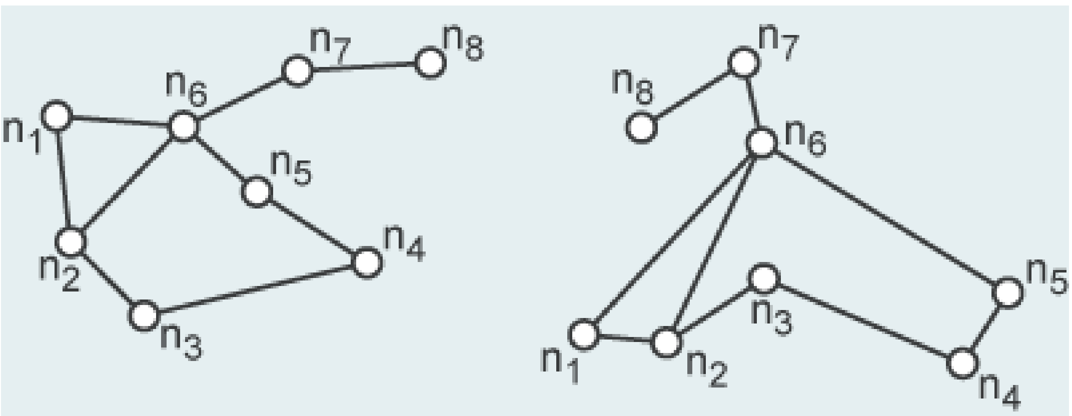


Adjacency matrix



Matrix with weights

The notion of topology is often used in assocition with the description and analysis of networks. Topology and its concepts are discussed in detail in the module "Spatial Modeling". The networks shown in the figure below have topologically equivalent compounds, thus they are topologically equivalent graphs.



Two topologically equivalent graphs

1.2. Structural Properties of a Network

After having discussed the basic building blocks of networks in detail, let us now deal with ways to capture and describe the structure of networks. The following measures are available for these tasks:

- Connectivity (Beta-Index)
- Diameter of a graph
- Accessability of nodes and places
- Centrality / location in the network
- Hierarchies in trees

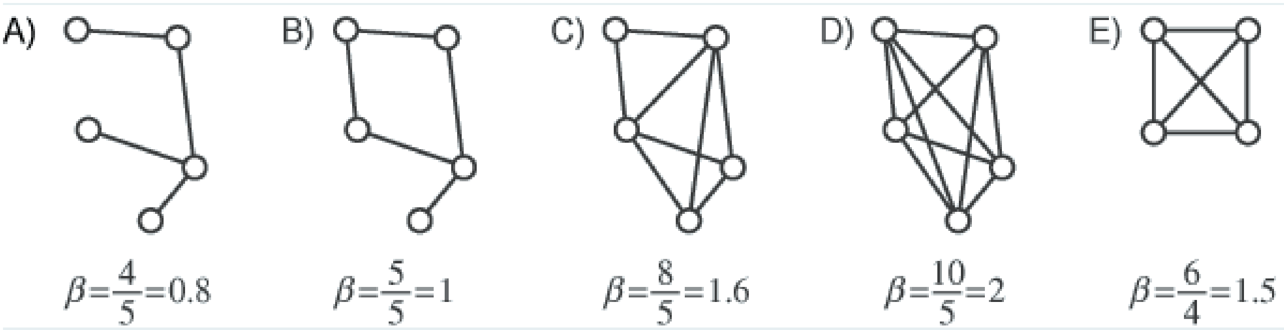
For most of these measures we will present one unweighted and one weighted (metric) case.

1.2.1. Connectivity (Beta index)

The simplest measure of the degree of connectivity of a graph is given by the Beta index (#). It measures the density of connections and is defined as:

$$\beta = \frac{E}{V}$$

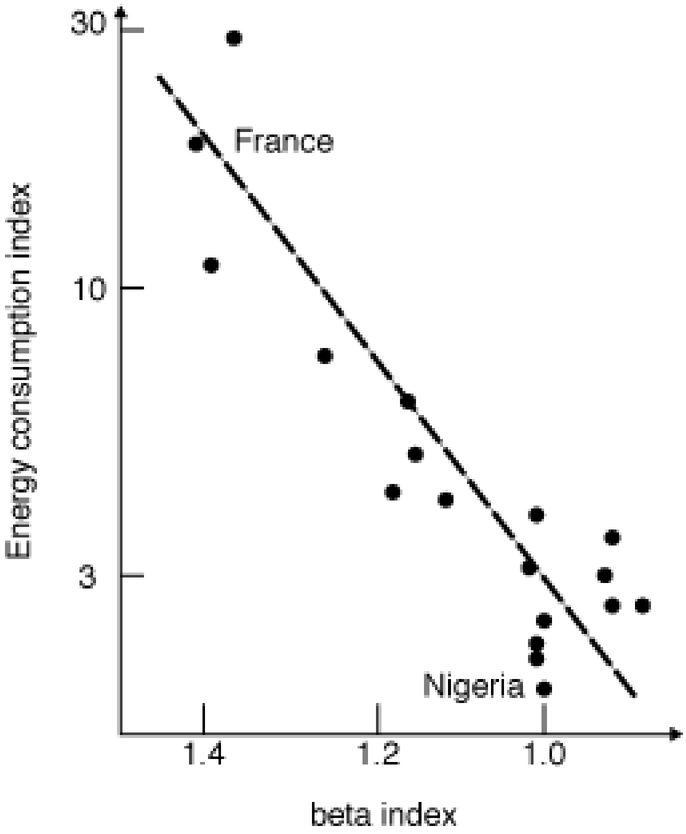
where *E* is the total number of edges and *V* is the total number of vertices in the network.



Beta index, calculated for different graphs

In the figure above, the number of vertices remains constant in A, B, C and D, while the number of connecting edges is progressively increased from four to ten (until the graph is complete). As the number of edges increases, the connectivity between the vertices rises and the Beta index changes progressively from 0.8 to 2. Values for the index start at zero and are open-ended, with values below one indicating trees and disconnected graphs (A), and values of one indicating a network which has only one circuit (B). Thus, the larger the index, the higher the density.

With the help of this index, regional disparities can be described, for example. In the figure below, the railway networks of selected countries are compared to general economic develompent (using the energy consumption-index of the 1960s). Energy consumption is plotted on the *y*-axis and the Beta index on the *x*-axis. Where connectivity is high, the economic development is high as well.



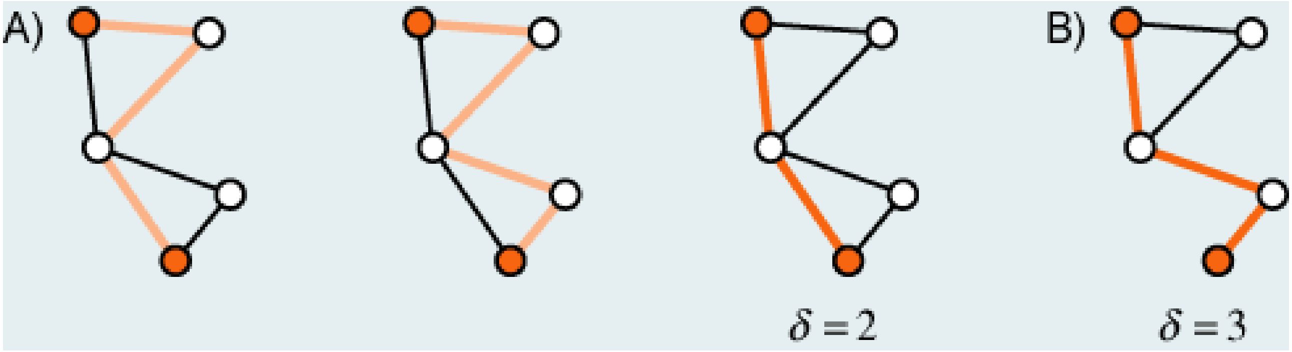
Energy consumption compared to measure of connectivity (Haggett et al. 1977)

1.2.2. Diameter of a graph

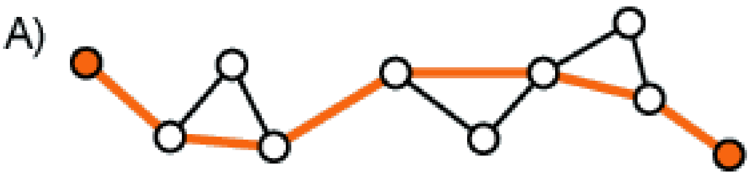
Another measure for the structure of a graph is its diameter. Diameter # is an index measuring the topological length or extent of a graph by counting the number of edges in the shortest path between the most distant vertices. It is:

$$\delta = \max_{ij} \{s(i, j)\}$$

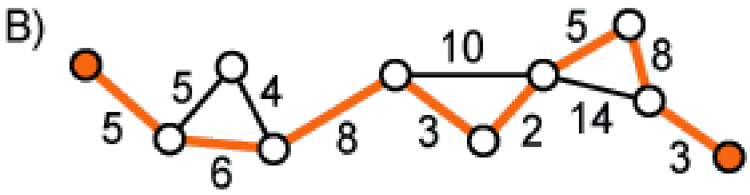
where $s(i, j)$ is the number of edges in the shortest path from vertex i to vertex j. With this formula, first, all the shortest paths between all the vertices are searched; then, the longest path is chosen. This measure therefore describes the longest shortest path between two random vertices of a graph.



The first two figures in graph A show possible paths but not the shortest paths. The third figure and figure B show the longest shortest path.



$$\delta = 6$$



$$m_\delta = 5+6+8+3+2+5+8+3 = 40$$

$$\pi = \frac{m_T}{m_\delta} = \frac{73}{40} = 1.825$$

In addition to the purely topological application, actual track lenths or any other weight (e.g. travel time) can be assigned to the edges. This suggests a more complex measurement based on the metric of the network. The resulting index is *# = mT/m#*, where *mT* is the total mileage of the network and *m#* is the total mileage of the network's diameter. The higher # is, the denser the network.

1.2.3. Accessibility of vertices and places

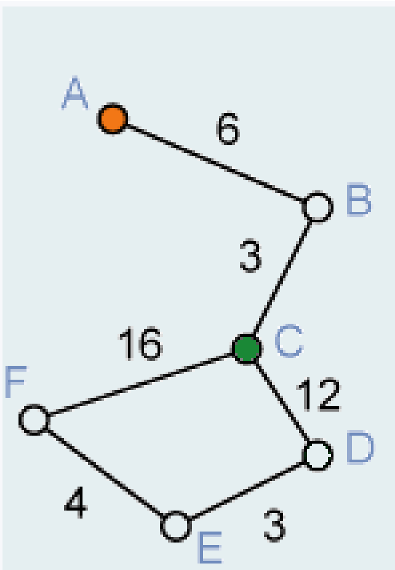
A frequent type of analysis in transport networks is the investigation of the accessibility of certain traffic nodes and the developed areas around them. A measure of accessibility can be determined by the method shown in the animation. The accessibility of a vertex *i* is calculated by:

$$E_i = \sum_{j=1, j\neq i}^{v} \mathrm{n}(i,j)$$

where v = the number of vertices in the network and *n* (*i*, *j*) = the shortest node distance (i.e. number of nodes along a path) between vertex *i* and vertex *j*. Therefore, for each node i the sum of all the shortest node distances *n*(*i*, *j*) are calculated, which can efficiently be done with a matrix. The node distance between two nodes *i* and *j* is the number of intermediate nodes. For every node the sum is formed. The higher the sum (node A), the lower the accessibility and the lower the sum (node C), the better the accessibility.

Only pictures can be viewed in this version! For Flash, animations, movies etc. see online version. Only screenshots of animations will be displayed. [link]

The importance of the node distance lies in the fact that nodes may also be transfer stations, transfer points for goods, or subway stations. Therefore, a large node distance hinders travel through the network.



	A	B	C	D	E	F
A	0	6	9	21	24	25
B	6	0	3	15	18	19
C	9	3	0	12	15	16
D	21	15	12	0	3	7
E	24	18	15	3	0	4
F	25	19	16	7	4	0
Σ	85	61	55	58	64	71

Calculation of the accessibility Ei

As with the diameter of a network, a weighted edge distance can also be used along with the pure topological node distance. Examples of possible weighting factors are: distance in miles or travel time as well as transportation cost. For this weighted measure, however, the edge distance is used and not the node distance.

$$E_i = \sum_{j=1}^{e} s(i,j)$$

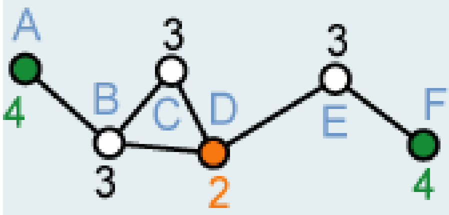
where e is the number of edges and *s*(*i*, *j*) the shortest weighted path between two nodes.

1.2.4. Centrality / Location in the network

The first measure of centrality was developed by König in 1936 and is called the König number *Ki*. Let *s*(*i*, *j*) denote the number of edges in the shortest path from vertex *i* to vertex *j*. Then the König number for vertex *i* is defined as:

$$K_i = \max_{j \neq i}\{s(i,j)\}$$

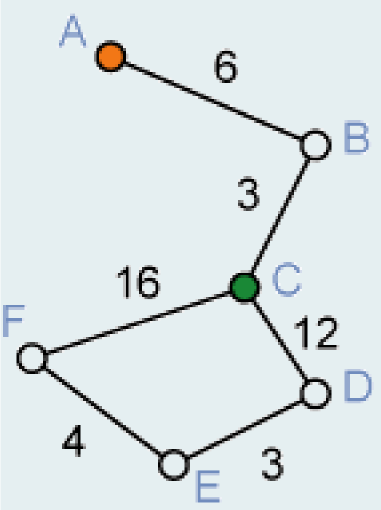
where *s*(*i*, *j*) is the shortest edge distance between vertex *i* and vertex *j*. Therefore, *Ki* is the longest shortest path originating from vertex *i*. It is a measure of topological distance in terms of edges and suggests that vertices with a low König numbers occupy a central place in the network.



	A	B	C	D	E	F
A	0	1	2	2	3	4
B	1	0	1	1	2	3
C	2	1	0	1	2	3
D	2	1	1	0	1	2
E	3	2	2	1	0	1
F	4	3	3	2	1	0
max	4	3	3	2	3	4

If you have determined the shortest edge distance between the nodes, then the largest value in a column in the König number (blue). In the example, the orange node is centrally located and the two green nodes are peripheral.

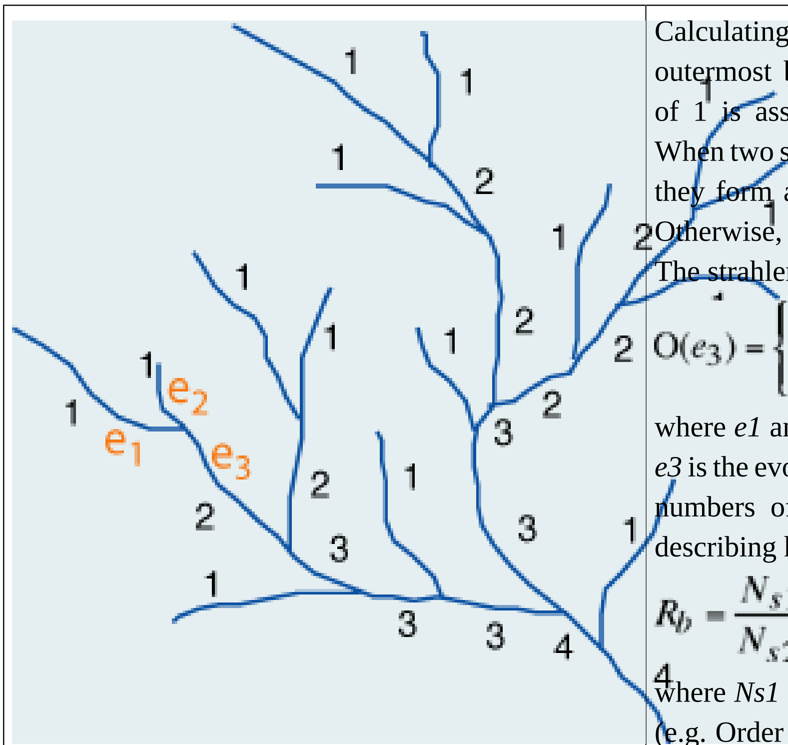
The method for determining the König number is also applicable to a distance matrix. The example of accessibility is shown again in the figure below. This time the matrix is used with the same values to calculate the König number.



	A	B	C	D	E	F
A	0	6	9	21	24	25
B	6	0	3	15	18	19
C	9	3	0	12	15	16
D	21	15	12	0	3	7
E	24	18	15	3	0	4
F	25	19	16	7	4	0
max	25	19	16	21	24	25

1.2.5. Hierarchies in trees

In quantitative geomorphology, more specificall in the field of fluvial morphology, different methods for structuring and order of hierarchical stream networks have been developed. Thus, different networks can be compared with each other (e.g. due to the highest occurence order or the relative frequencies of the unique levels), and sub-catchments can be segregated easily. Of the four ordering schemes in the following figure, only three are topologically defined. The Horton scheme is the only one that takes the metric component into account as well.



Strahler stream order

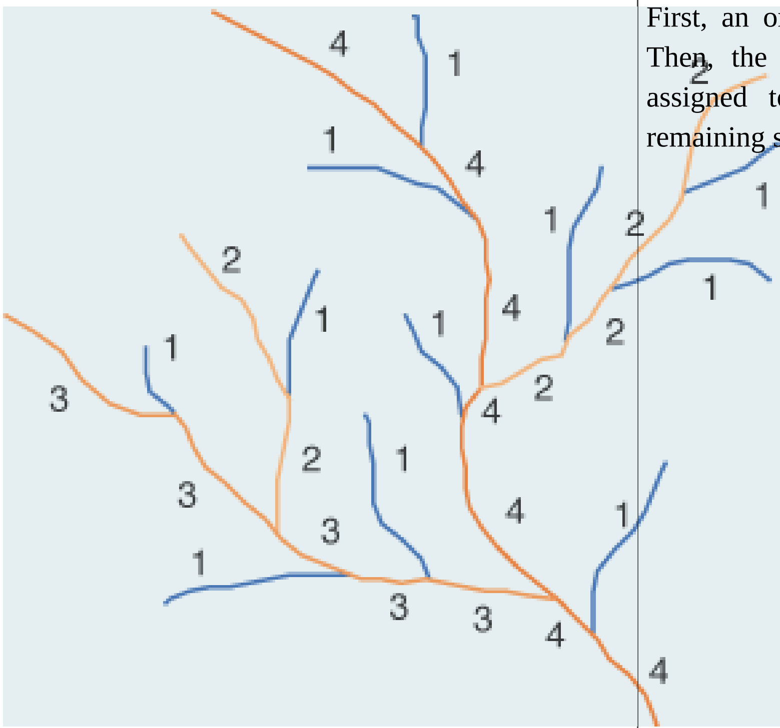
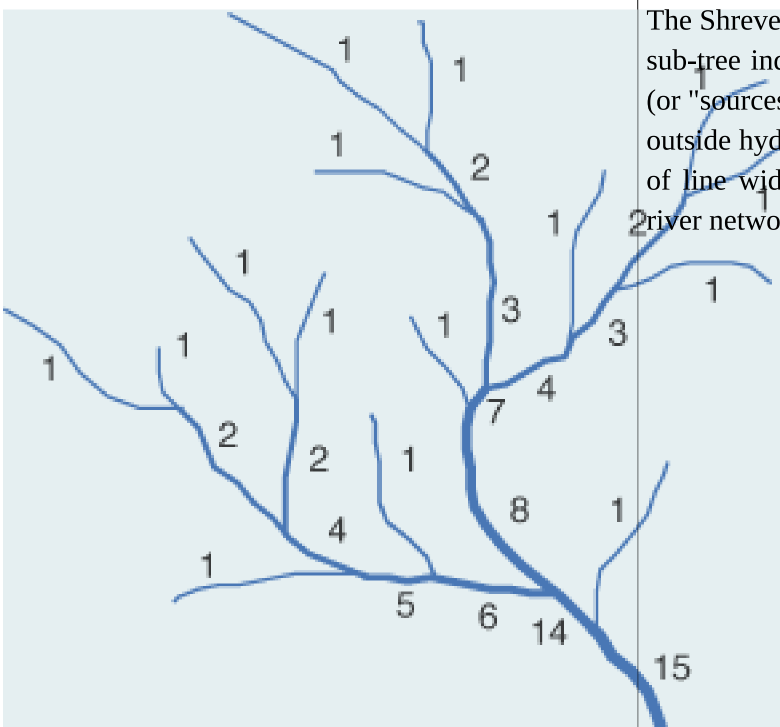
Calculating the strahler number, we start with the outermost branches of the tree. The ordering value of 1 is assigned to those segments of the stream. When two streams with the same order come together, they form a stream with their order value plus one. Otherwise, the higher order of the two streams is used. The strahler number is formally defined as:

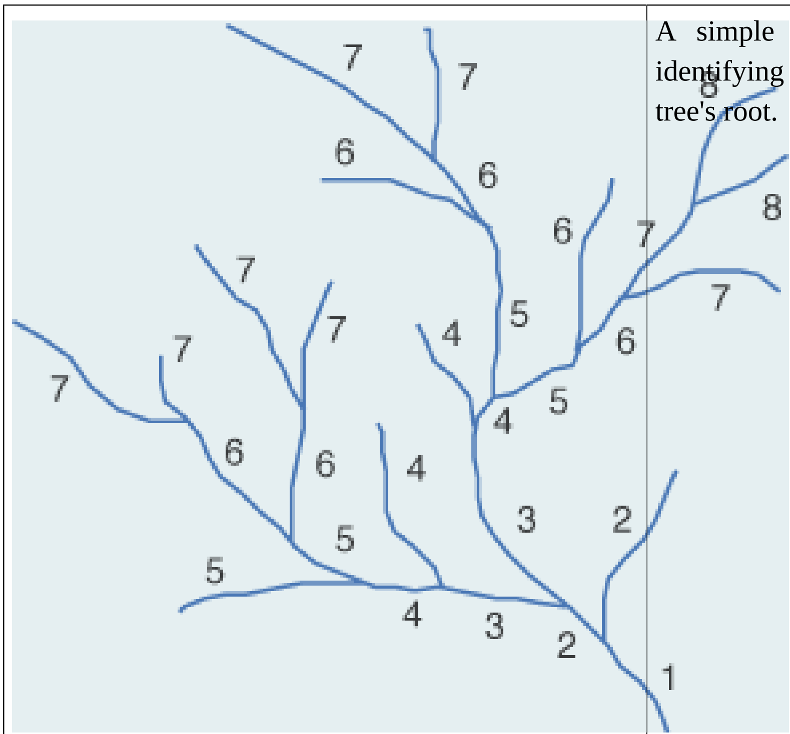
$$O(e_3) = \begin{cases} O(e_1)+1 & O(e_1) = O(e_2) \\ \max(O(e_1), O(e_2)) & \text{sonst} \end{cases}$$

where *e1* and *e2* are the joining stream segments and *e3* is the evolving stream. Associated with the Strahler numbers of a tree are bifurcation ratios, numbers describing how close to balanced a tree is:

$$R_b = \frac{N_{s1}}{N_{s2}}$$

where *Ns1* is the number of edges of a specific order (e.g. Order 1) and *Ns2* is the number of edges of the next higher order. In the example on the left, *Ns1* = 15 and *Ns2* = 7. This results in the bifurcation ration of:

	$R_b = \frac{N_{s1}}{N_{s2}}$
 *Horton stream order*	First, an order according to Strahler is calculated. Then, the highest current order larger than 2 is assigned to the longest (metric) branch in the remaining sub-trees.
 *Shreve stream order*	The Shreve stream order (also called magnitude) of a sub-tree indicates how many segments of first order (or "sources") are upstream. One possible application outside hydrology or geomorphology is in the choice of line widths in the cartographic representation of river networks.



Order by path length

A simple order by path length is achieved by identifying the length of the paths by starting at the tree's root.

1.3. Dijkstra Algorithm

A common application in the field of network analysis is the calculation of shortest paths. There are various algorithms available to solve this problem. A very common algorithm for calculating the shortest distance between two nodes in a network is the Dijkstra algorithm. It is used in edge-weighted graphs, and calls exclusively for positive values used in the weights.
In this unit, we will explain the functionality of the Dijkstra algorithm. We present the pseudocode for the implementation of the algorithm. The principles of the algorithm are shown with the help of an animation.

1.3.1. Dijkstra Algorithm: Short terms and Pseudocode

Using the Dijkstra algorithm, it is possible to determine the shortest distance (or the least effort / lowest cost) between a start node and any other node in a graph. The idea of the algorithm is to continiously calculate the shortest distance beginning from a starting point, and to exclude longer distances when making an update. It consists of the following steps:

1. Initialization of all nodes with distance "infinite"; initialization of the starting node with 0
2. Marking of the distance of the starting node as permanent, all other distances as temporarily.
3. Setting of starting node as active.
4. Calculation of the temporary distances of all neighbour nodes of the active node by summing up its distance with the weights of the edges.
5. If such a calculated distance of a node is smaller as the current one, update the distance and set the current node as antecessor. This step is also called update and is Dijkstra's central idea.
6. Setting of the node with the minimal temporary distance as active. Mark its distance as permanent.
7. Repeating of steps 4 to 7 until there aren't any nodes left with a permanent distance, which neighbours still have temporary distances.

Only pictures can be viewed in this version! For Flash, animations, movies etc. see online version. Only screenshots of animations will be displayed. [link]

1.3.2. Dijkstra Algorithm: Step by Step

The following animation shows the prinicple of the Dijkstra algorithm step by step with the help of a practical example. A person is considering which route from Bucheggplatz to Stauffacher by tram in Zurich might be the shortest…

Only pictures can be viewed in this version! For Flash, animations, movies etc. see online version. Only screenshots of animations will be displayed. [link]

1.3.3. Applications, extensions, and alternatives

There are different applications and special cases where the Dijkstra algorithm can be applied. In addition to routing, the calculation of distances can also be used for other areas where the euclidian distance is not the basis, but time or cost is.
In addition to the weighting of the edges, weights to the node can also be specified. This could be of use when the process of changing within the node "main station" is associated with high costs. In this case, the node receives its own weight, which has to be taken into account while computing the shortest paths.
The Dijkstra algorithm does not work with negative edge weights. If you want to run a shortest path calculation with negative egde weights, other algorithms such as the Bellman-Ford-algorithm must be used.
The Dijkstra algorithm is not suitable for all applications or types of graphs. Other algorithms include the Kruskal or Borùvka which are used to compute minimum spanning trees in undirected graphs.

1.4. Traveling Salesman Problem

The problem of the traveling salesman is a often discussed problem when it comes to routing. It involves optimizing the order of visits to several places in a way that the total route is as short as possible. The total route includes the journey from the last visited place to the place of departure. The computation cannot be managed by using simple methods - in particular if the distance is non-euclidian, but time or cost serves as basis for route calculation (Haggett et al. 1977).

Note: particularly in relation to optimization for the planning of delivery routes, we also refer to the calculation as a vehicle routing problem. This requires additional criteria in delivery planning.

This unit introduces the traveling salesman problem, demonstrates different scenarios that will show its complexity (particularly considering the vehicle routing problem), and presents various approaches that can be used to solve the problem. These include (Bott et al. 1986):

- Exact solution methods
- Heuristic solution methods
- Interactive solution methods
- Combinded solution methods

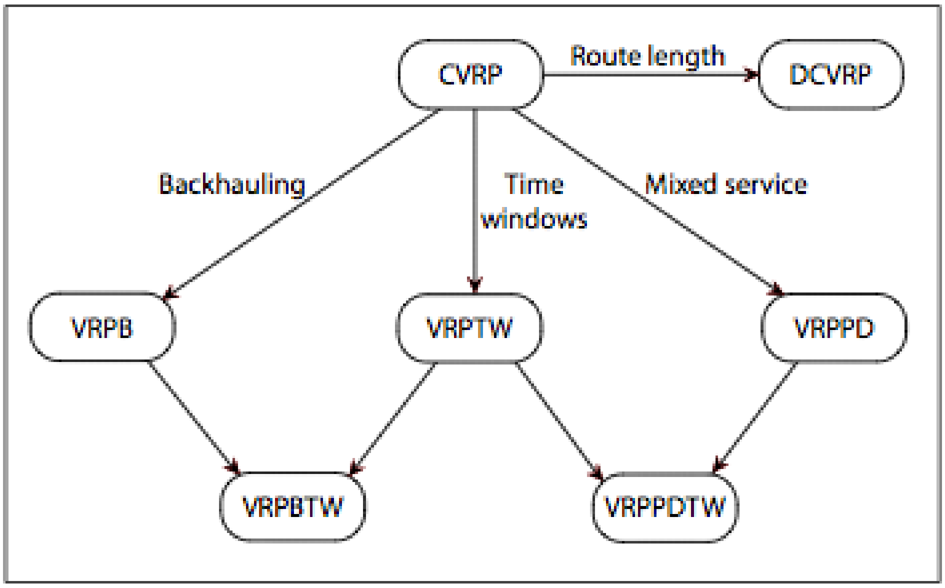
1.4.1. Kriterien des Vehicle Routings (not translated yet)

Neben der Optimierung der Planung im Sinne der Berechnug des minimalen Gesamtreisewegs, kommen beim Vehicle Routing auch spezifische Faktoren wie die Minimierung der globalen Transportkosten (Fahrtkosten + Fixkosten der Fahrzeuge), Minimierung der Anzahl der benutzten Fahrzeuge und der Ausgleich der Touren im Hinblick auf Fahrzeiten und Lademengen hinzu.

Je nach lieferspezifischen Bedingungen kann das Vehicle Routing Problem in unterschiedliche Varianten untergliedert werden (Toth 2001):

- **Capacited VRP (CVRP):** Waren und Fahrzeuge mit vorab festgelegter Kapazität. Waren können nicht aufgeteilt werden, es gibt nur ein Depot und alle Fahrzeuge haben die gleiche Kapazität.
- **VRP with Time Windows (VRPTW):** Kunden sollen innerhalb festgelegter Zeitfenster mit Waren beliefert werden.
- **VRP with Pickup and Delivery(VRPPD):** Waren können beim Kunden sowohl ab- als auch aufgeladen werden.
- **VRP with Backhaul (VRPB):** Waren können entweder ab- oder aufgeladen werden.
- **Distance-Constrained CVRP (DCVRP):** Es ist eine maximale Distanz- oder Zeitdauer vorgeben.
- **Multi Depot VRP (MDVRP):** Es existieren mehrere Depots.

Die folgende Abbildung zeigt die Basisvarianten des Vehicle Routings und deren Mischformen.



Basisvarianten des Vehicle Routings (Toth 2001)

1.4.2. Approaches for the Vehicle Routing Problem

Of the approaches mentioned in the introduction to this unit, the first two will be explained, that is the exact and heuristic methods.

Exact solution methods
Exact methods calculate the path lengths of every possible round trip, and in doing so, to choose the smallest path length. The traveling salesman problem and the vehicle routing problem are often a non-deterministic polynomial problem (NP-hard problem). The cost of computing every possible route is too high (Solomon 1987). Particularly in delivery route planning, there are often too many stations (costumers) to be supplied.

Heuristic solution methods
Heuristic solution methods are essential if there is a large amount of data to be processed (e.g. customers to be supplied). These methods can be divided into opening procedure (also called design procedure) and improvement procedure, depending on whether new routes are constructed or existing routes are improved. However, they lack the ability to make a quality assessment of the identified routes.
Opening procedures use the nearest neighbour algorithm that let the salesmen choose the nearest (according to the length of the path) unvisited city as his next move (see Dijkstra). The nearest insertion algorithm additionally tests whether there are stations located near the connecting lines between two stations. If such a station is found it is interposed.
Improvement procedures try shorten existing routes with the help of minor modifications.
Bott und Ballue (1986) identify four different categories of heuristic approaches to solve the vehicle routing problem (Bott et al. 1986):

- Cluster first, route second approaches
- Route first, cluster second approaches
- Savings or insertion approaches
- Exchange or improvement approaches

It is important to note that in practice, often combined methods such as "cluster first, route second" approaches are used. With these appraoches, the routing starts only after customers are assigned to a depot. Other approaches assume that an inadequate assignment leads to poor routing. Therefore, the first arrangement of routes is gradually improved by modification (exchange or improvement procedures) (Foulds 1997).

1.5. Summary

In this lesson network analysis was discussed. Network analysis is primarily concerned with the relationships between objects. To investigate these (distance) relationships, we need knowledge about the existing primitives (see 1.1.1 Primitives of a Network) and measurements. This is possible with the help of a description about the structure of networks (see 1.2 Structural Properties of a Network).
The computation of shortest paths is a common application of network analysis and graph theory. In this context, the Dijkstra algorithm which is the most commonly used algorithm for computing shortest paths in edge-weighted graphs, has been presented. The pseudocode of the algorithm was presented and explained by means of a step-by-step animation of the algorithm. It was also stressed that for special applications and special types of graphs (weighting of nodes, negative edge weights) modifications or alternative algorithms are required to calculate distance.
Furthermore, the Travelling Salesman Problem (TSP) was presented which dealt with the question of how route calculation for several stops can be optimized for minimal distance travelled. In this context, the vehicle routing problem was introduced. It extends the TSP to the effect that minimizing transportation cost and the number of vehicles used was also considered. Exact and heuristic algorithms and also interactive and combined solution methods provide solutions to such problems. It was shown that heuristic approaches are preferred when using large data dets since exact algorithms are computationally complex.

1.6. Glossary

Algorithmus von Bor#vka:
Algorithmus von Dijkstra:
Algorithmus von Kruskal:
Bellman-Ford Algorithmus:
Graph:
Besteht aus einer Menge an Knoten und Kanten, die auf unterschiedliche Art und Weise miteinander verknüpft sein können. Je nachdem werden ungerichtete von gerichteten Graphen unterschieden, planare von nicht-planaren Graphen. Bei ungerichteten Graphen unterscheidet man zudem zwischen zyklischen, vollständigen oder gewichteten Graphen. Bei gerichteten Graphen zudem zwischen
Heuristik:
Kante:
kantengewichtet:
Knoten:
minimaler Spannbaum:
Pfad:
Routenplanung (Routing):
Shortest Path:
Travelling Salesman Problem:
Vehicle Routing Problem:
Vertex:

1.7. Bibliography

- **Bartelme, Norbert**, 2000. *Geoinformatik - Modelle, Strukturen, Funktionen*. Berlin: Springer-Verlag.
- **Bott, Kevin; Ballue, Ronald H.**, 1986. Research perspectives in vehicle routing and scheduling.. *TRANSPORT. RES.*, 20(3), 239-243.
- **Foulds, L. and J. Wilson**, 1997. A variation of the generalized assignment problem arising in the New Zealand dairy industry.. *Annals of Operations Research*, 69, 105–114.
- **Garrison, William L.**, 1968. Connectivity of the interstate highway system. *In:* **Berry, Brian J.L.; Marble, Duane F.**, ed. *Spatial analysis: A reader in statistical geography*. Englewood Cliffs: Prentice-Hall.
- **Gatrell, A.C.**, 1991. Concepts of space and geographical data. *Geographical Information Systems*, 119-143.
- **Haggett, Peter; Cliff, Andrew D.; Frey, Allan**, 1977. *Locational Analysis in Human Geography*. New York: Wiley.
- **Jones, Ch.**, 1997. *Geographical Information Systems and Computer Cartography*. Longman Singapore Publishers (pte) Ltd..
- **Sedgewick, Robert**, 1998. *Algorithmen*. Bonn: Addison-Wesley.
- **Solomon, Marius M.**, 1987. Algorithms for the Vehicle Routing and Scheduling Problems with Time Window Constraints.. *Operations Research*, 35(2), 254-265.
- **Spiekermann, Klaus**, 1999. *Visualisierung von Eisenbahnreisezeiten – Ein interaktives Computerprogramm*. Institut für Raumplanung, Universität Dortmund: Dortmund, Germany, 45.
  Download: http://www.raumplanung.uni-dortmund.de/irpud/pro/visual/ber45.pdf
- **Toth, Paolo**, 2001. *The Vehicle Routing Problem*. Society for Industrial and Applied Mathematics.
- **Transport for London** (2005). *Tube Maps of London* [online]. London: Mayor of London. Available from: https://tfl.gov.uk/maps/track/tube [Accessed 28.04.2016].
  Download: ../download/london_tube.pdf
- **Weibel, Robert; Brassel, Kurt**, 2002. *Grundlagen der geographischen Informationsverarbeitung*. Zürich: Geographisches Institut der Universität Zürich (unveröffentlichtes Manuskript).

1.8. Index