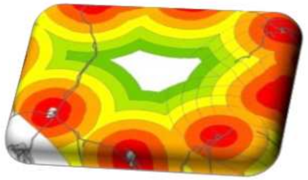


GIS Applications : Network Analysis

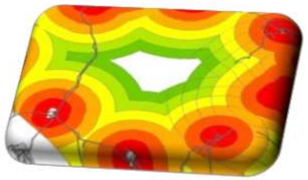
EGE 2421

Network and Location Analysis

Lecture No. 07

Felix Mutua, Ph. D

Wednesday, October 18, 2023

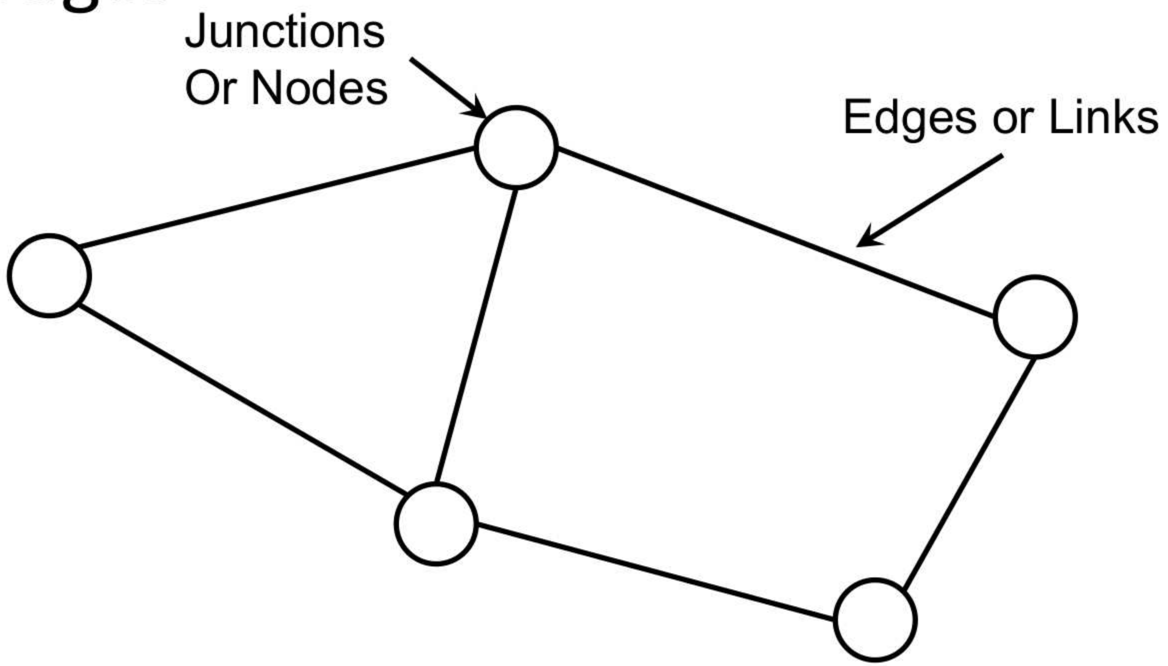


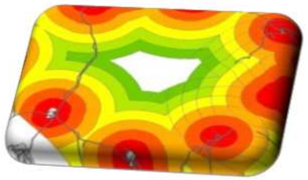
Lecture Plan

Week	Topic		Week	Topic	
1	Overview		8	Networks – I *(concepts, network problems)*	
2	Review of GIS analysis Techniques		9	Networks – II *(building networks, optimization)*	
3	GIS in Agriculture (concepts, application areas, Crop Suitability Analysis)		10	Networks – III *(routing, tracking)*	
4	Natural resource Management – I (concepts, application areas)		11	Utility Management *(concepts, viewsheds,lineof sight)*	
5	Natural resource Management – II (Groundwater, forestry)		12	Health and Disease control *(concepts in epidemiology)*	
6	GIS in Business *(store location, consumer profiling)*		13	Governance *(crime, districting, LIS, census)*	
7	CAT I		14	CAT II	

What are Networks?

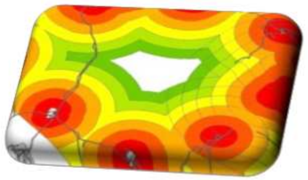
- Systems of connected lines
- Weights along edges





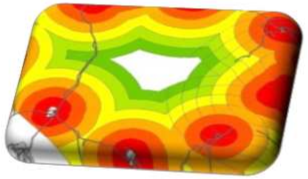
Spatial Examples

- Electrical Grid
  - Power lines and transfer stations
- Roadways (highways, freeways)
  - Roads and intersections
- Pipelines
- Canals, streams, rivers
- Shipping lanes
- Migration paths
- Social networks (non-spatial?)



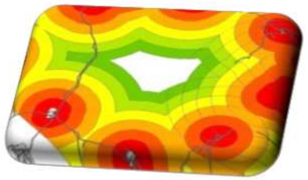
Network Analysis

- **Network Analysis** in GIS is based on the mathematical sub-disciplines of graph theory and topology. Any network consists of a set of connected vertices and edges.
- Graph theory describes, measures, and compares graphs or networks. Topological properties of networks are:
  - connectivity,
  - adjacency, and
  - incidence.
- These properties serve as a basis for analysis.



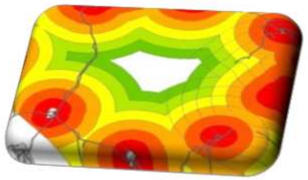
Network Analysis

- Networks consist of interconnected lines (known as edges) and intersections (known as junctions) that represent routes upon which people, goods, etc. can travel.
- The object traversing the network follows the edges, and junctions appear when at least two edges intersect.
- Junctions and edges can have certain attributes affixed to them that increase the cost of traveling in the network, known as *impedance*. For example, a road network can have speed limits attached to the edges, and a junction can prevent left turns.
- Networks are either directed, in which only one direction of travel is allowed within the network, or undirected, in which any direction of travel is allowed.



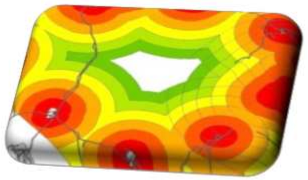
Network Analysis

- Networks include:
  - Utility networks: including water mains, sewage lines, and electrical circuits. These networks are generally directed.
  - Transportation networks: including roads, railroads, and flight paths. These networks are generally undirected.
  - Networks based on social connections.
- Typical problems to solve in network analysis include :
  - Shortest Path
  - Traveling Salesman
  - Network Partition



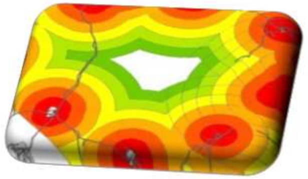
Concepts in Network analysis: Network Rules

- Edge-Junction:
  - Which edges can connect to a junction
  - Examples:
    - Three-phase 240 volt vs. high voltage
    - Superhighway to overpass
- Edge-Edge:
  - Freeway to Freeway: overpass with clover leaf
  - Freeway to Highway: overpass with lights
  - Highway to Highway: lights
  - Road to Road: stop signs



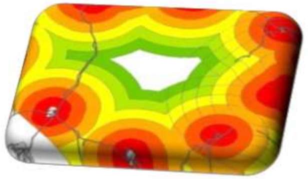
Concepts in Network analysis: Weights

- Edge
  - Type of surface, speed limit -> Travel time
- Junction
  - Type of intersection -> stop time
- Barrier
  - Stops travel, can be temporal
  - Examples:
    - Construction
    - Raising bridges
- Can be directional (i.e. fish move downstream easier than upstream)



Concepts in Network analysis: Weights

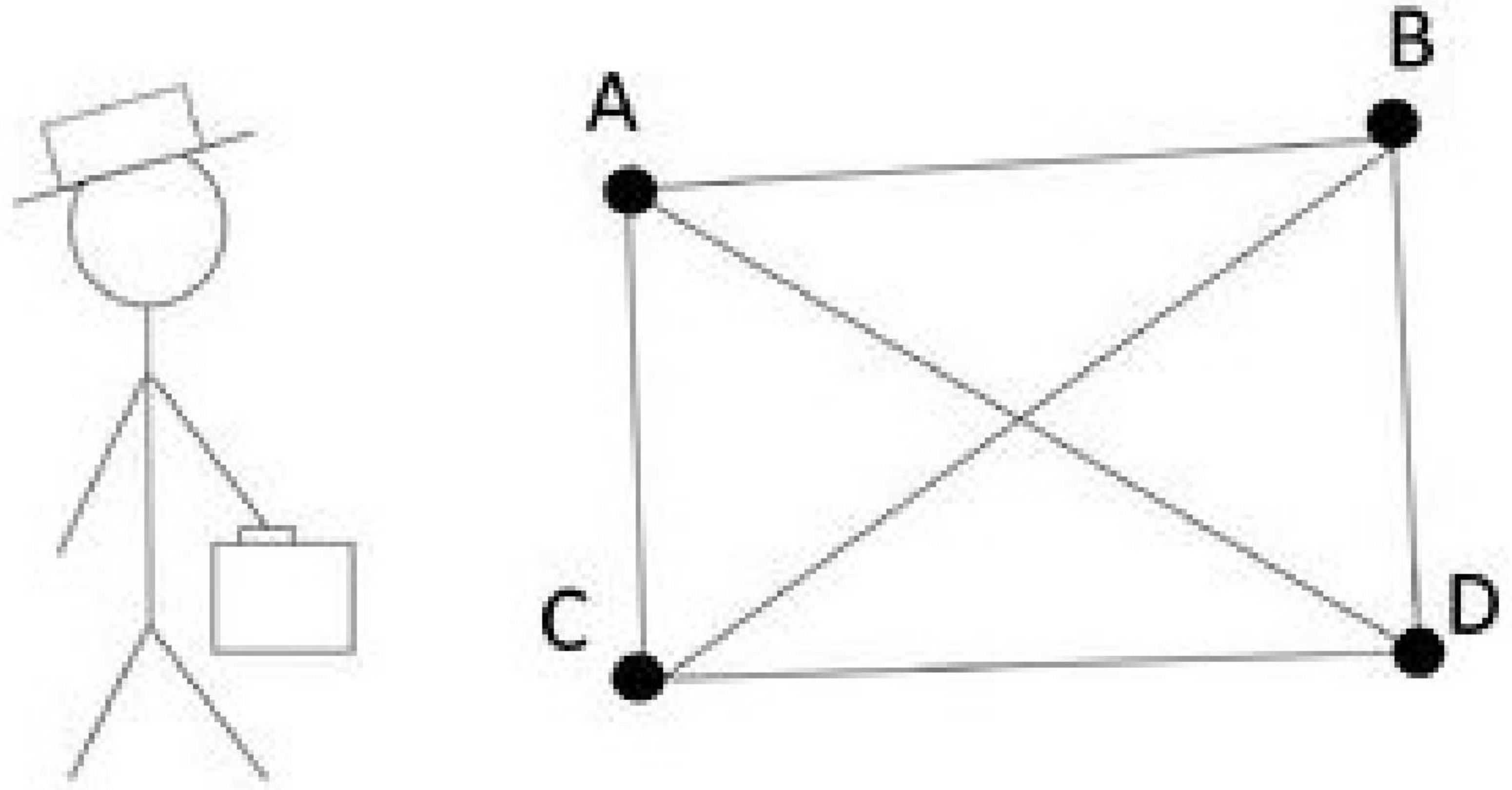
- Edge
  - Type of surface, speed limit -> Travel time
- Junction
  - Type of intersection -> stop time
- Barrier
  - Stops travel, can be temporal
  - Examples:
    - Construction
    - Raising bridges
- Can be directional (i.e. fish move downstream easier than upstream)



Concepts in Network analysis: Network Problems

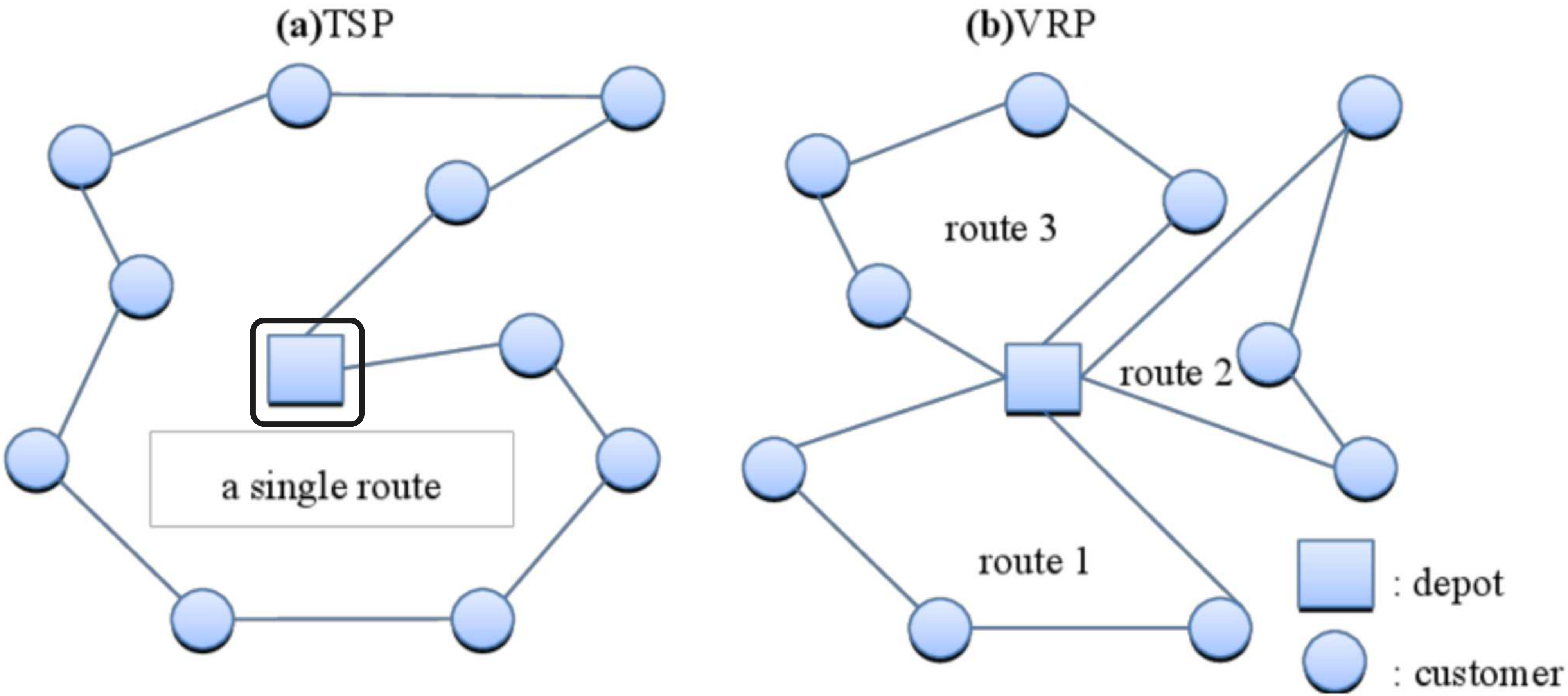
- Shortest path: Route
- Minimizing total distance traveled
- Minimizing the largest distance traveled by any customer
- Maximizing profit
- Minimizing a combination of travel distance and facility operating cost

Network Problems



The salesperson must find the shortest route that leads to each of the four points.

Network Problems



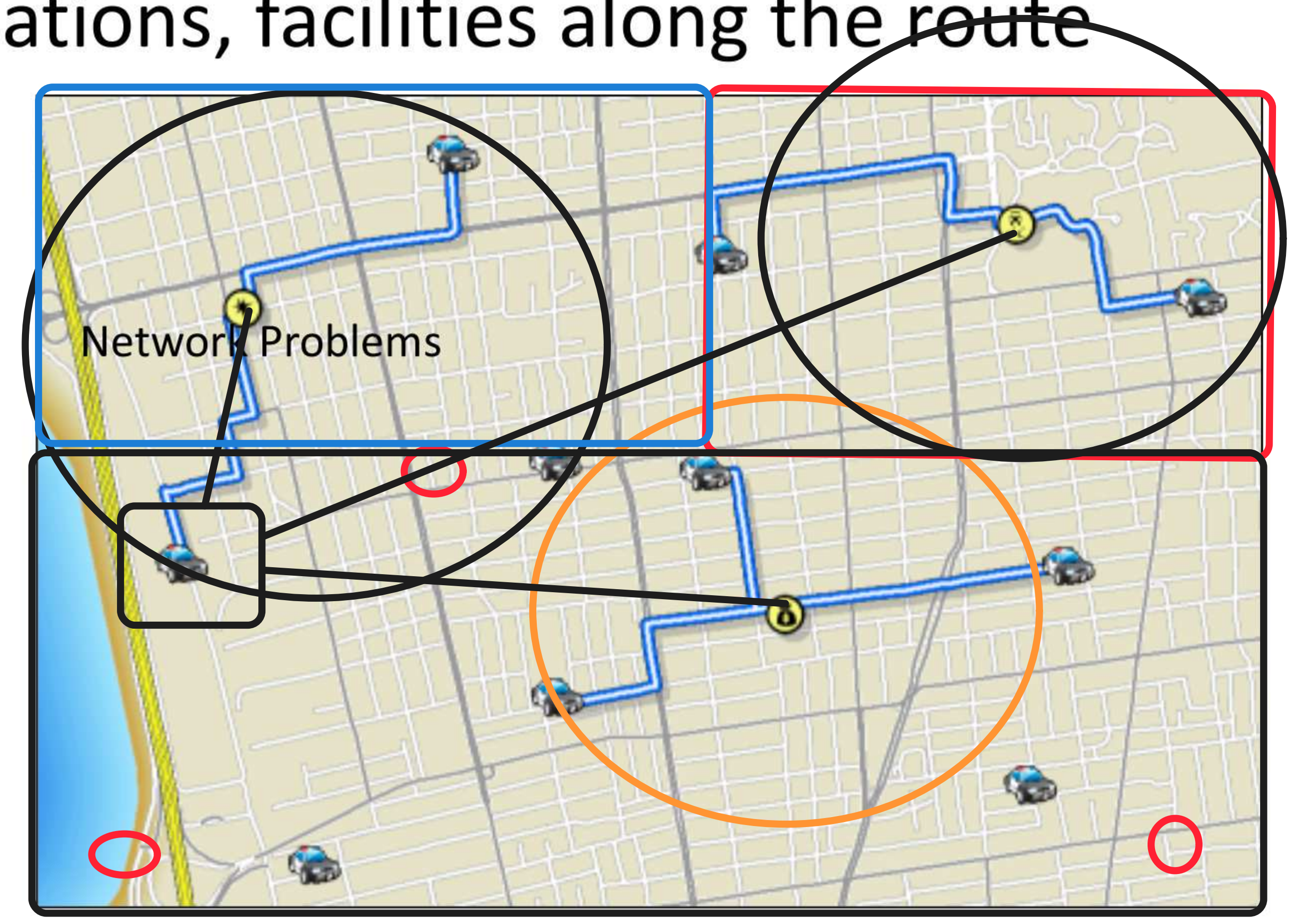
a) traveling salesman problem (TSP) and b) vehicle route problem (VRP) route patterns.

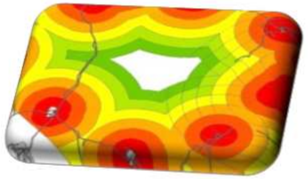
Shortest Path - Route

- Stops
- Time windows for deliveries
- Etc.

Closest facility analysis

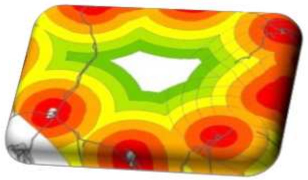
- Multiple origins, destinations, facilities along the route





Location-allocation Problems

- Design locations for services, and allocate demand to them, to achieve specified goals
- Goals might include:
  - minimizing total distance traveled
  - minimizing the largest distance traveled by any customer
  - maximizing profit
  - minimizing a combination of travel distance and facility operating cost

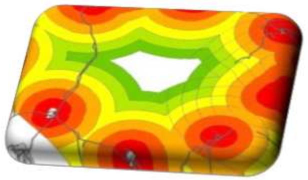


Optimizing Point Locations

- One service location and the goal of minimizing total distance traveled
- The operator of a chain of convenience stores or fire stations might want to solve for many locations at once
  - where are the best locations to add new services?
  - which existing services should be dropped?

Routing Problems

- Search for optimum routes among several destinations
- Draws on location-allocation
- The traveling salesman problem
  - find the shortest (cheapest) tour from an origin, through a set of destinations that *visits each destination only once*

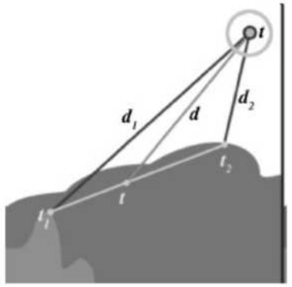


Optimum Paths

- Find the best path across a continuous surface
  - between defined origin and destination
  - to minimize total cost
  - cost may combine construction, environmental impact, land acquisition, and operating cost
  - used to locate highways, power lines, pipelines
  - requires a raster representation

EX: Optimization & Routing for Emergency/Disaster Response

- Kim et al. 2006 – PARs, Protective Action Recs



$$d = d_1 - \left[\left(\frac{d_1 - d_2}{t_2 - t_1} \right) \times (t - t_1) \right]$$

d= interpolated, shortest-distance of wildfire to community
d1 = shortest distance before PAR
d2 = shortest distance after PAR
t = time PAR was issued
t1 = time last known fire perimeter at d1
t2 = time last known fire perimeter at d2

Fire Origin to Communities: Estimate Avg. Speed of Fire Between Known Perimeters

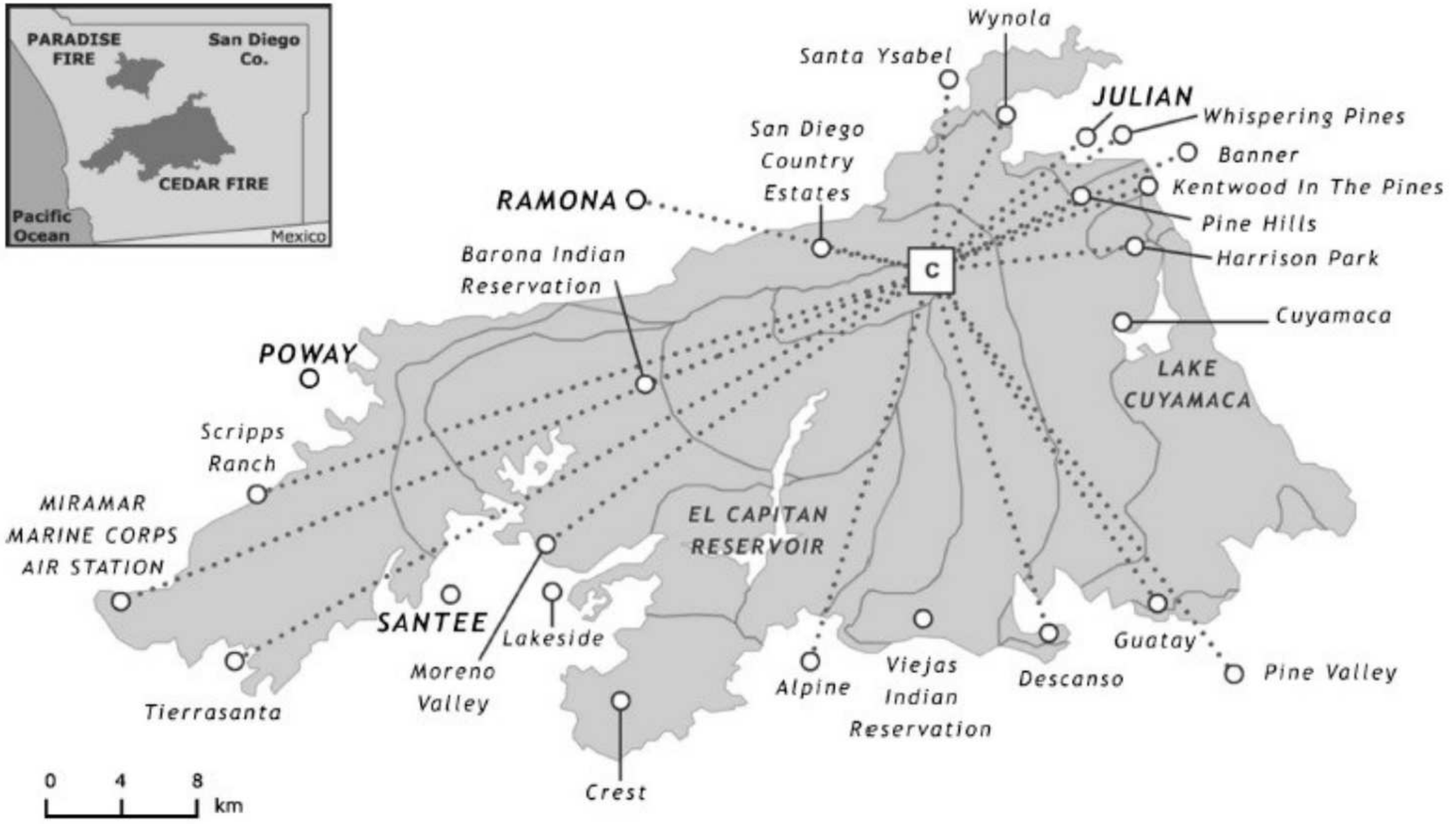
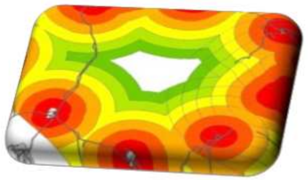


Fig. 11. Measuring the approximate fire spread distance from a fire to a community (Cedar Fire)

Kim et al. 2006

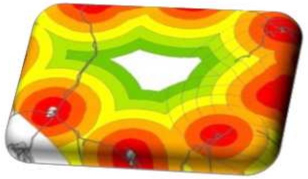


Network analysis terminology

Term	Description
Vertex	A point location considered as a network node.
Edge	A (directed or undirected) link between two vertices that are directly connected is called an edge
Degree (of a vertex)	In an undirected graph the number of edges meeting at a vertex.
Graph	A collection of vertices and edges constitutes a graph.
Path	A (network) path is a sequence of connected edges between vertices
Connected graph	If at least one path exists between every vertex and every other vertex in a graph it is described as connected
Connectivity	Network connectivity is the minimum number of nodes or links that must fail in order to partition the network (or sub-network) into two or more disjoint networks.
Planar graph	If a graph can be drawn in the plane (embedded) in such a way as to ensure edges only intersect at points that are vertices then the graph is described as planar
Network	A collection of vertices and edges together with associated attribute data that may be represented and analyzed using graph theoretic methods.
Diameter	The maximum number of links that must be traversed to reach any node along a shortest path
Cycle	A path from a given vertex to itself that traverses other vertices is a cycle. A graph that has no cycles is called *acyclic*
Tree	An *n*-vertex acyclic network or subnetwork in which every vertex is connected, for which the number of edges is *n*-1. A unique path exists between every pair of vertices in a tree

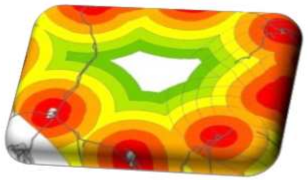
Network Data

- A set of line segments or polylines, such as a road or street map, does not typically constitute a network.
- In order to conduct many forms of network analysis it is necessary to ensure that a set of distinct nodes or vertices (V) exist and between each vertex one or more links to other vertices are defined.
- These links are referred to as edges (E).
- At the simplest level edges may be directed (e.g. one way) or undirected (two-way) and may have one or more attributes associated with them (e.g. name, length, travel time/cost, mode of transport)



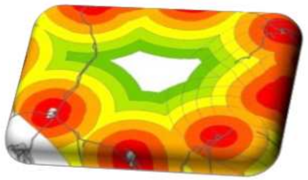
Algorithms and computational complexity theory

- Most of the key problems addressed in location and network analysis can be classed as “optimization problems”
- they are an important subset of a much broader set of problems
- In the case of networks, which typically consist of a set of n vertices, V, and m edges, E, the amount of time taken to compute an exact solution as a function of the number of these elements is vital



Key Problems in Network and Location Analysis

Problem	Description
Hamiltonian circuit (HC)	If a cycle exists from a given vertex that passes through every other vertex exactly once it is called a Hamiltonian circuit. Testing for the existence of Hamiltonian circuits in a graph is known as the Hamiltonian circuit problem (HCP).
Eulerian circuit(EC)	A circuit in a directed graph that visits every arc exactly once. A condition that a graph contains an Eulerian circuit is that the number of arcs arriving at every included vertex, *i*, must be the same as the number of arcs leaving vertex *i*
Shortest path (SP)	A path between two vertices that minimizes a pre-defined metric such as the total number of steps, total distance or time, is called a shortest path. Determination of shortest paths is often described as shortest path analysis (SPA).
Spanning tree (ST)	Given a fixed set of vertices, find a set of edges such that every vertex is connected and the network contains no cycles.
Minimal spanning tree (MST)	Find a (Euclidean) spanning tree of minimum total length.
Steiner MST, Steiner tree	As per the MST but with additional nodes permitted that are not co-located with the original vertex set
Traveling salesman problem (TSP)	Given a set of vertices and symmetric or asymmetric distance matrix for each pair of vertices, find a Hamiltonian circuit of minimal length (cost). Typically the start location (vertex) is pre-specified and the vertices are not necessarily assumed to lie on a pre-existing network. If certain nodes must be visited before others, the task is known as a sequential ordering problem (SOP).
Vehicle routing problem (VRP)	This class of problems relates to servicing customer demand (e.g. deliveries of fuel to retail garages) from a single depot, where each vehicle may have a known capacity (CVRP). If capacity is not restricted the problem in known simply as a vehicle routing problem (VRP).



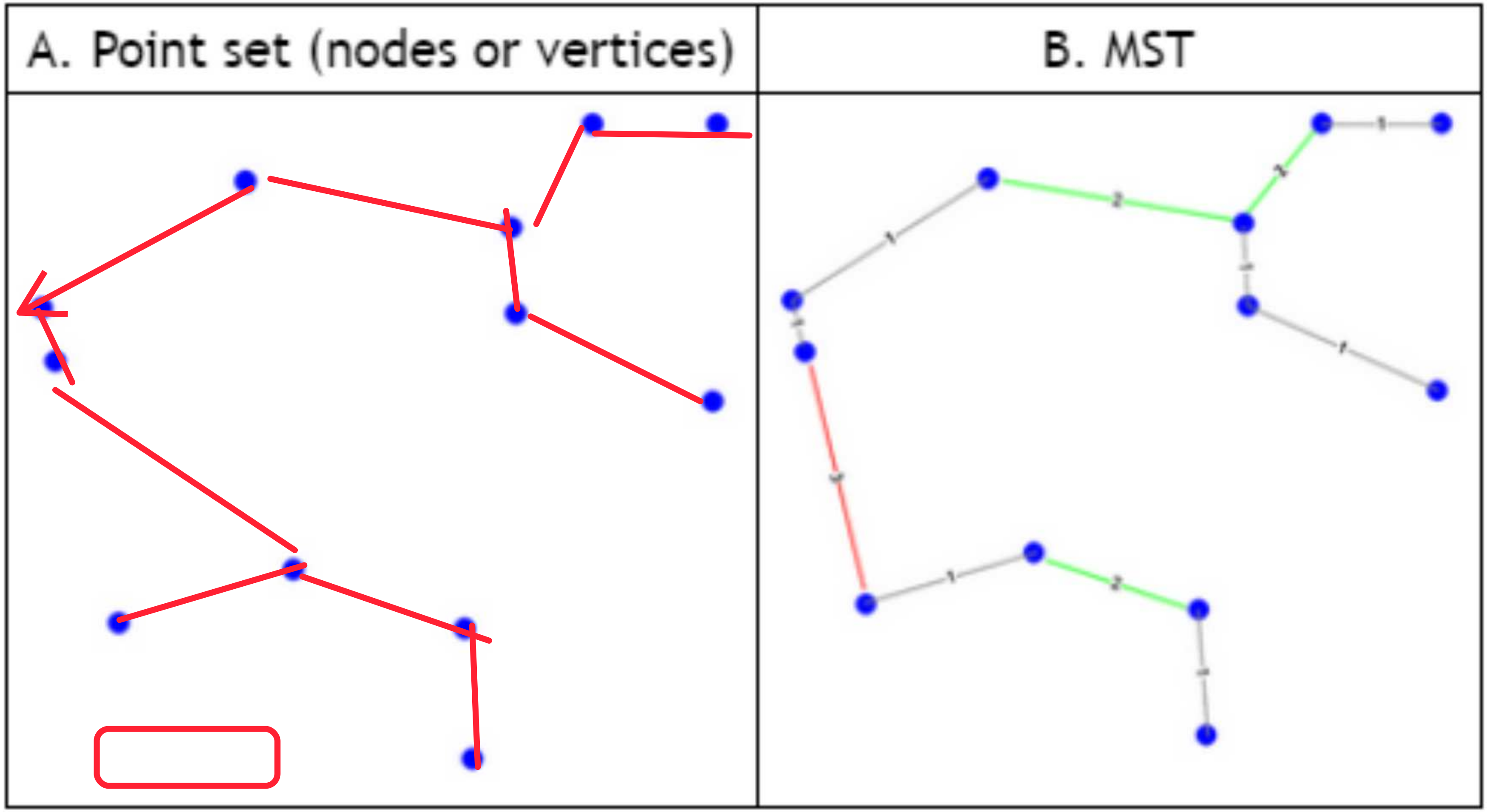
Key Problems in Network and Location Analysis

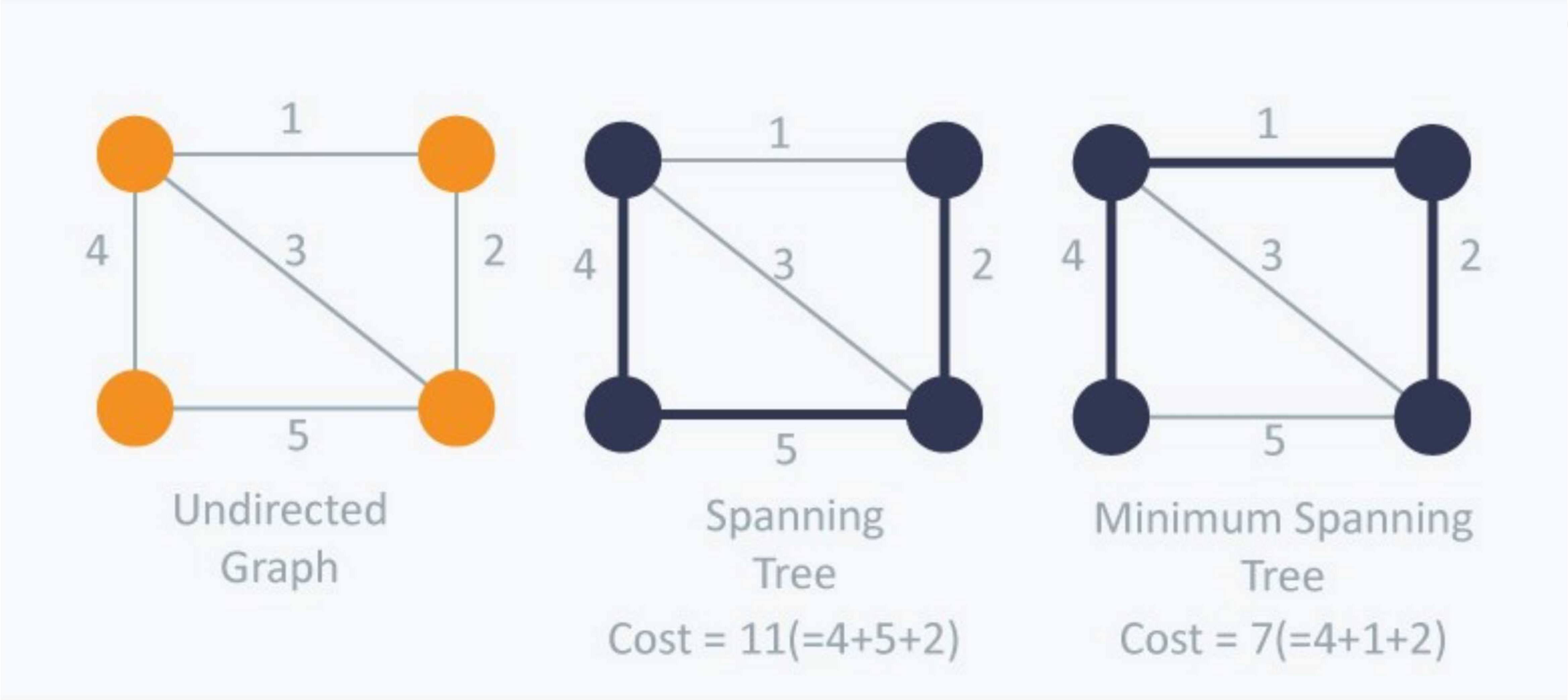
Transportation problem, Trans-shipment problem	The general problem of completely servicing a set of target locations with given demand levels from a set of source locations with given supply levels such that total costs are minimized is known as the transportation problem.
Arc routing problem (ARP)	Given a network (typically a street network or subset of a street network) find a route that completely traverses every edge, generally in both directions, that has the least cost (distance or time) subject to selected constraints (e.g. cost of turning). This problem applies to street cleaning, snow-plowing, postal deliveries, meter reading, garbage collection etc.
Facility location: *p*-median/*p*-center/ coverage	A collection of problems where the objective is to optimally locate one or more facilities within a network in order to satisfy customer requirements (demand, service level). The most commonly cited problem is • minimization of total (or average) travel cost/time to or from customers (the *p*-median problem). Minimization of maximum distance or time is known as a *p*-center problem. • A related set of problems seeks to ensure that all customers can be served within a fixed upper time or cost, or at least, as many as possible are served within a fixed time or cost. These are known as coverage problems.

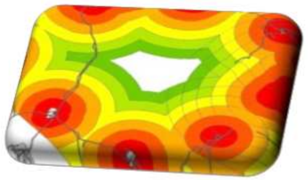
Network Construction, Optimal Routes and Optimal Tours

1. Minimum Spanning Tree

- Given a set of vertices (points, nodes) an enormous number of possible interconnections may be made to produce a network of direct or indirect connections between vertices.
- The set of connections that minimizes total edge length whilst ensuring every point is reachable from every other point is known as a minimal spanning tree (MST).
- The algorithm involves a construction or growth process as follows:
  - (i) connect every point to its nearest neighbor — typically this will result in a collection of unconnected sub-networks;
  - (ii) connect each sub-network to its nearest neighbor sub-network;
  - (iii) iterate step in (ii) until every sub-network is inter-connected.


A. Point set (nodes or vertices)
B. MST

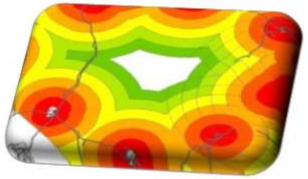

1
4
3
2
5
Undirected Graph
1
4
3
2
5
Spanning Tree
Cost = 11(=4+5+2)
1
4
3
2
5
Minimum Spanning Tree
Cost = 7(=4+1+2)



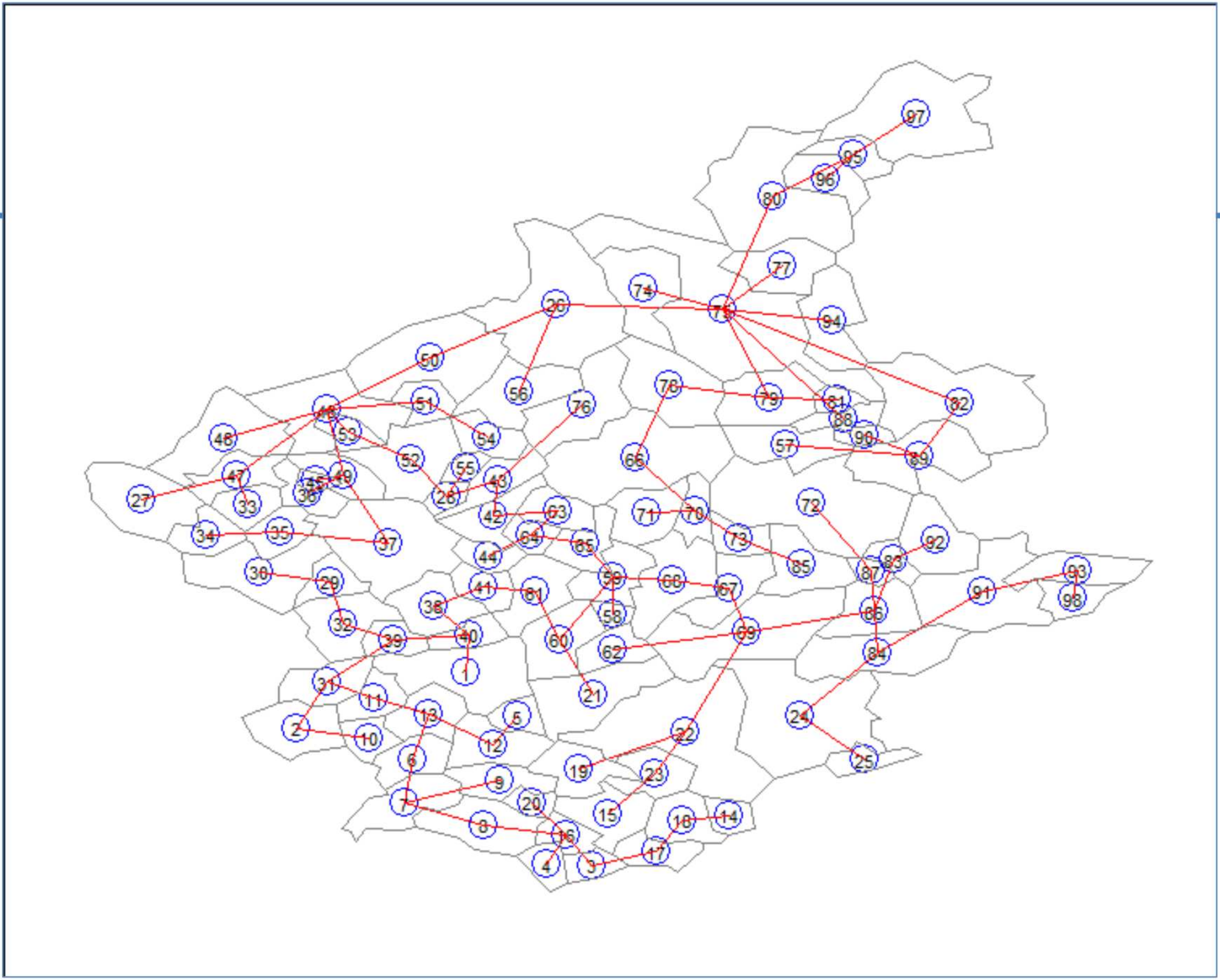
Example : MST

```
library(rgdal)
library(tmap)
 library(spdep)
 library(maptools)
 ### loading data
bh <- readShapePoly(("F://Dropbox//JKUAT//CourseWork//2020-
2021//Sem_II//EGE_2421//data//shapes//bhicv.shp" )[1])
#plot
Plot(bh)
### Scaling / Normalizing the column
dpad <- data.frame(scale(bh@data[,5:8]))
```

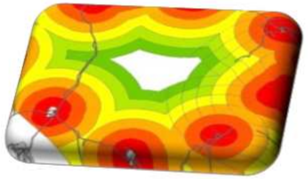
```
### neighbourhood list
#The function builds a neighbours list based on regions with contiguous boundaries, that is sharing one or more boundary point
bh.nb <- poly2nb(bh)

### calculating costs
#Compute Cost Of Edges. The cost of each edge is the distance between it nodes.
lcosts <- nbcosts(bh.nb, dpad)
```



```
### making listw
# Spatial Weights For Neighbours Lists. The nb2listw function supplements a neighbours list with spatial weights for the chosen coding scheme.
nb.w <- nb2listw(bh.nb, lcosts, style="B")

### find a minimum spanning tree
system.time(mst.bh <- mstree(nb.w,5))

### the mstree plot
par(mar=c(0,0,0,0))
plot(mst.bh, coordinates(bh), col=2,
    cex.lab=.7, cex.circles=0.035, fg="blue")
plot(bh, border=gray(.5), add=TRUE)
```

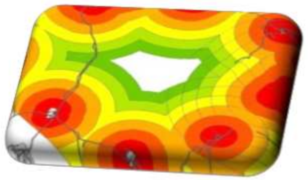


97
95
96
80
77
74
26
94
50
78
79
81
82
56
51
76
88
53
54
46
90
57
89
66
52
55
47
49
43
27
33
36
28
72
42
63
71
70
34
35
37
64
73
92
65
44
85
83
87
30
29
68
67
93
41
61
91
38
58
86
98
32
39
40
60
69
62
84
1
21
31
11
13
5
24
2
10
22
12
6
25
19
23
9
7
20
15
18
14
8
16
4
3
17



2. shortest path problem

- the general shortest path (SP) problem requires a predefined network.
- The basic problem is then to determine one or more shortest (or least cost) routes between a source vertex and a target vertex where a set of edges are given.
- The set of shortest paths generated from a single source is known as a shortest path tree (SPT).



Shortest Path Algorithm

- The determination of shortest paths can be specified as a linear programming problem, as follows.
- Let *s* be the source vertex, *t* be the target vertex and let $c_{ij}>0$ be the cost or distance associated with the link or edge (*i,j*). Then we seek to minimize *z*, where:

$$z = \sum_i \sum_j c_{ij} x_{ij},$$

subject to

$$\sum_j x_{ji} - \sum_k x_{ik} = m, \text{ where}$$

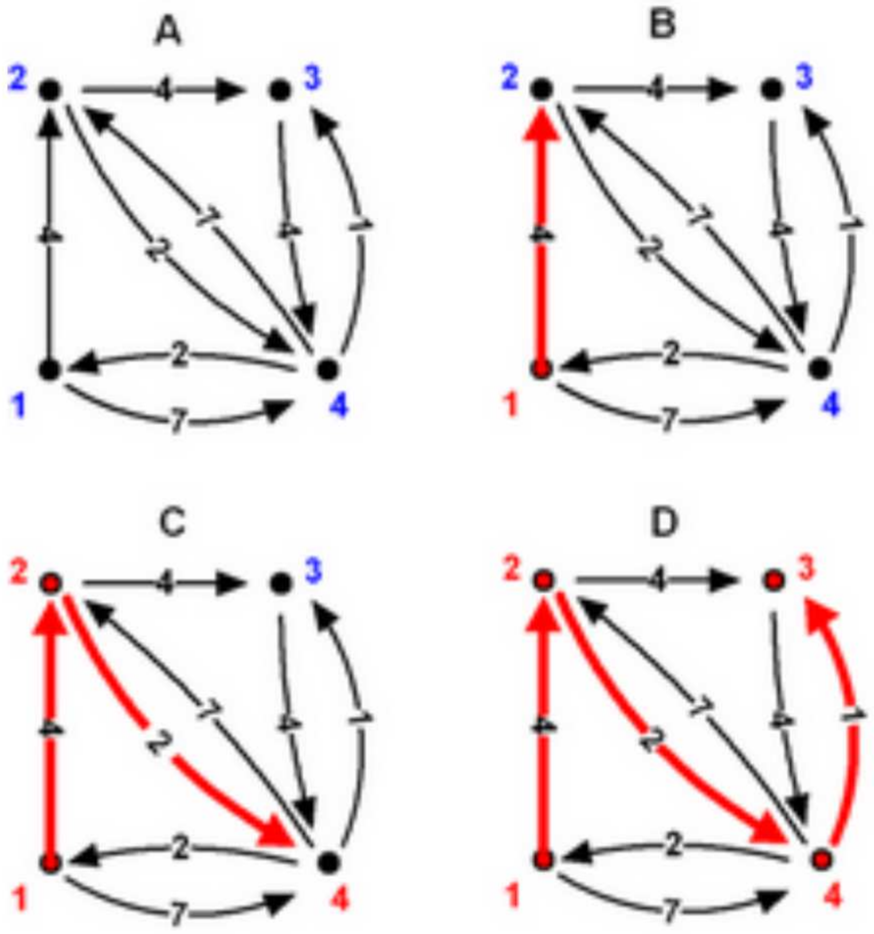
$$m = 0 \; for \; i \neq s,$$

$$m = 1 \; for \; i = t$$

$$m = -1 \; for \; i = s$$

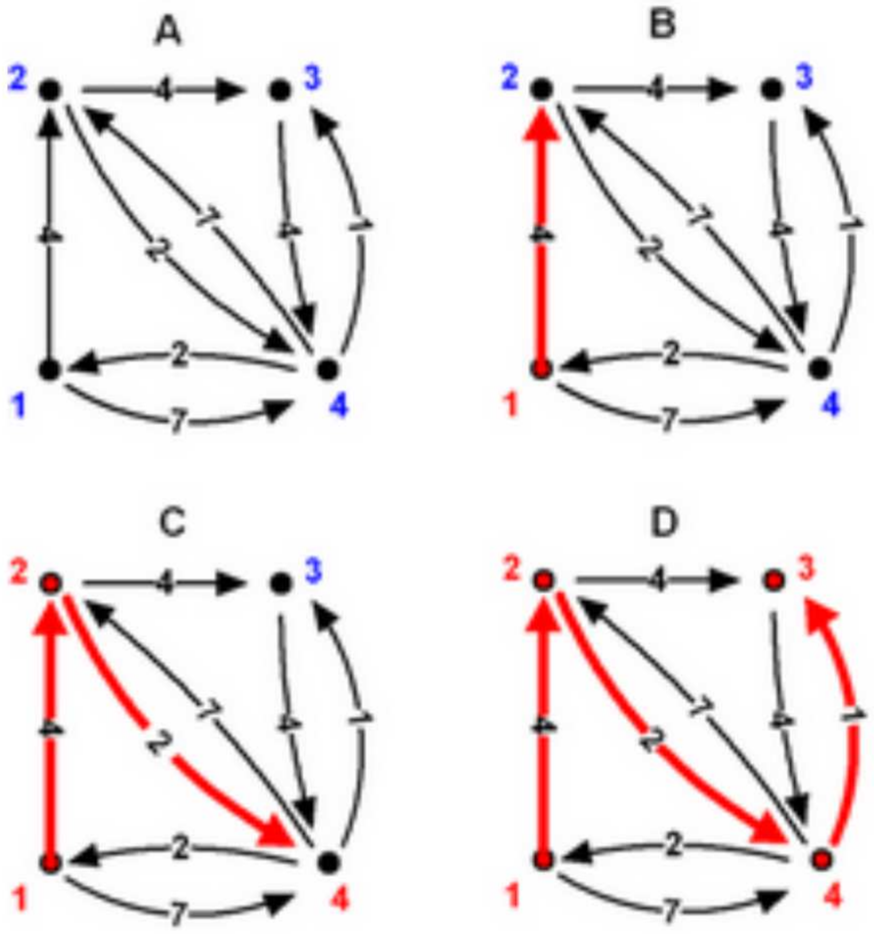
$$x_{ij} \in \{0,1\}$$

Dantzig algorithm

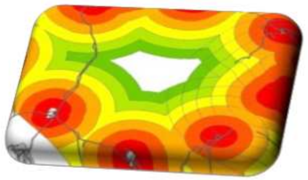


- A step-by-step algorithm for determining the shortest path from vertex 1, 2, 3 and 4 to all other vertices in a directed planar graph (a *digraph*) with positive edge weights.

Dantzig algorithm

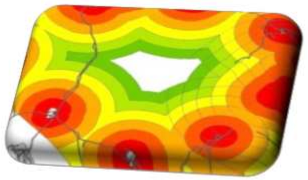


- The basic steps in this procedure are as follows:
  - 1: identify the shortest (least distance/cost/time) edge from vertex 1 — this is to vertex 2 (*cost*=4). Add vertex 2 and the edge or link from 1 to 2 to the tagged set. If a tie occurs, arbitrarily choose one of the edges
  - 2: identify the shortest (least cumulative cost/time) edge from vertex 1 or from vertex 2 plus edge (1,2) distance — this is to vertex 4 from 2 (*cost*=6). Add vertex 4 and edge 2 to 4 to the tagged set
  - 3: identify the shortest (least cumulative cost/time) edge from the tagged set — this is from vertex 1 to 2 to 4 to 3 (*cost=7*)
  - Stop — all vertices reached; repeat from vertex 2, 3 and 4 to compute all shortest paths from every vertex



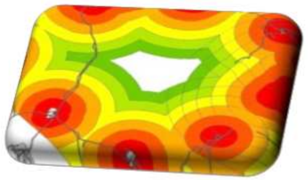
Dijkstra algorithm

- This algorithm works by storing the cost of the shortest path found so far between the start vertex, *s*, and each target vertex, *t*. We shall denote this distance $d(t)$.
- The basic steps in this classic algorithm are as follows:
  - 1: initialize all vertices such that $d(t)=\infty$ (infinity, or in practice, a very large value) and $d(s)=0$
  - 2: For each edge leading from *s*, add the edge length from *s* to the current value of the path length at *s*. If this new distance is less than the current value for $d(t)$ replace this with the lower value
  - 3: choose the lowest value in the set $d(t)$ and move the current (active) vertex to this location
  - 4: iterate steps 2 and 3 until the target vertex is reached or all vertices have been scanned
- unlike the Dantzig algorithm the shortest path(s) are not generated, only their lengths



Shortest path

```
## build a graph with 5 nodes
library("igraph")
## Nodes and cost (km):
df2 = rbind(c(234,235,21.6),
        c(234,326,11.0),
        c(235,241,14.5),
        c(326,241,8.2),
        c(241,245,15.3),
        c(234,245,38.46))
names(df2) = c("start_id","end_id","newcost")
```

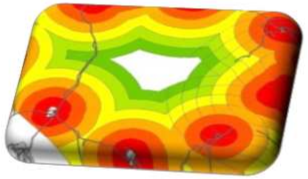


Shortest path

```
#Create Graph
g2 <- graph.data.frame(df2, directed=FALSE)

## calculate shortest path between vertex 234 and 245
(tmp2 = get.shortest.paths(g2, from='234', to='245',weights=E(g2)$newcost))

# compute the min distances from '234' to all other vertices
tmp3 <- shortest.paths(g2,v='234',weights=E(g2)$newcost)

# print min distance from '234' to '245'
tmp3[1, which(V(g2)$name == '245')]
```

Tutorial

https://www.r-spatial.org/r/2019/09/26/spatial-networks.html

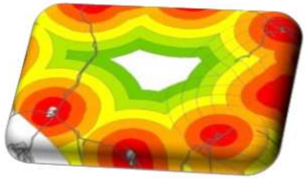


3. travelling salesman problem (TSP)

- The travelling salesman problem (TSP) asks the following question: Given a list of cities and the distances between each pair of cities, what is the shortest possible route that visits each city exactly once and returns to the origin city?
- With the basic definition you have a set of vertices (cities) and a set of edges (connection between cities).
- Each edge has an associated distance $d>0$. That distance could be travel time, distance in km or the monetary cost associated with traveling from one city to another.
- Restrictions on the distances lead to special cases of the problem.

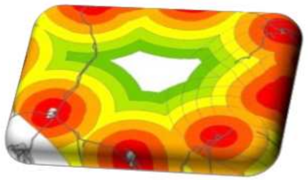
Tutorial

https://www.r-orms.org/mixed-integer-linear-programming/practicals/problem-tsp/



4. Location and Service Area Problems

- Optimally locating p facilities to serve customer demand at n>p locations is a simplified description of many locational analysis problems.
- If the existence of a network is ignored, and connectivity is assumed to be by direct connection in the plane from each customer to a single, closest, facility then a simple location-allocation problem is defined selecting locations and then allocating customers or demand to these locations according to some rule (e.g. nearest facility).
- Frequently, however, a network (e.g. a road network) exists together with a matrix of shortest path/least cost distances between vertices, and this provides a key input into the optimization process.

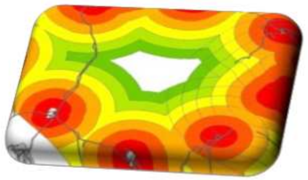


Service Area Problems

- An example of GIS-based service area functionality is provided in Figure 6.4 (next slide). This shows the locations of three ambulance stations situated in an urbanized region.
- In the figure all streets closest in network distance to each station are assigned as the primary service area for each station. As noted above, this form of simple network partitioning assumes pre-defined facility locations and takes no account of variations in demand (e.g. expected accident or illness rates) or in supply (e.g. number of vehicles available).

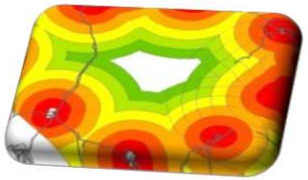
Service Area Problems

Figure 6.4: A. Ambulance Locations and B. Service areas (distance bands)



5. Vehicle Routing Problem

- the goal is to find optimal routes for multiple vehicles visiting a set of locations. (*When there's only one vehicle, it reduces to the Traveling Salesman Problem.)*
- But what do we mean by "optimal routes" for a VRP? One answer is the routes with the least total distance.
- However, if there are no other constraints, the optimal solution is to assign just one vehicle to visit all locations, and find the shortest route for that vehicle. This is essentially the same problem as the TSP.

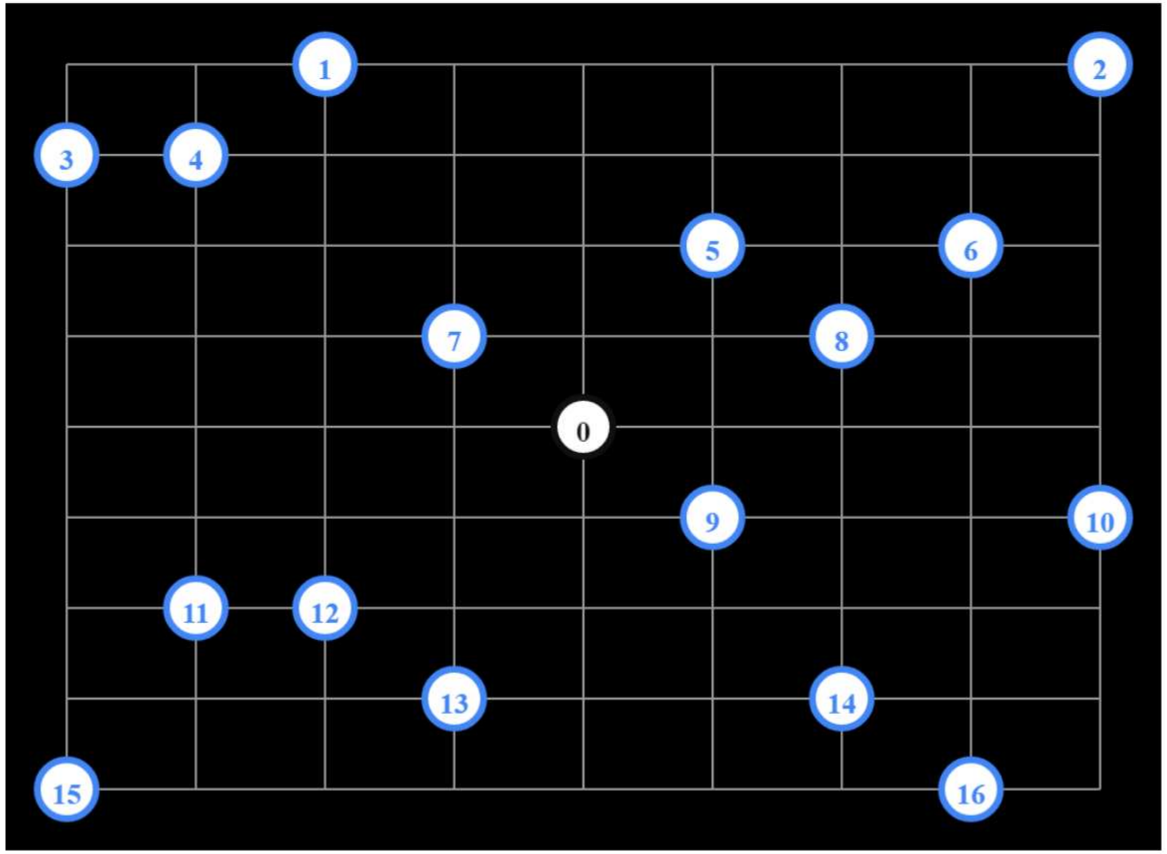


5. Vehicle Routing Problem

- A better way to define optimal routes is to minimize the length of the longest single route among all vehicles.
- This is the right definition if the goal is to complete all deliveries as soon as possible.
- *Constraints could include*
  - Capacity constraints: the vehicles need to pick up items at each location they visit, but have a maximum carrying capacity.
  - Time windows: each location must be visited within a specific time window.

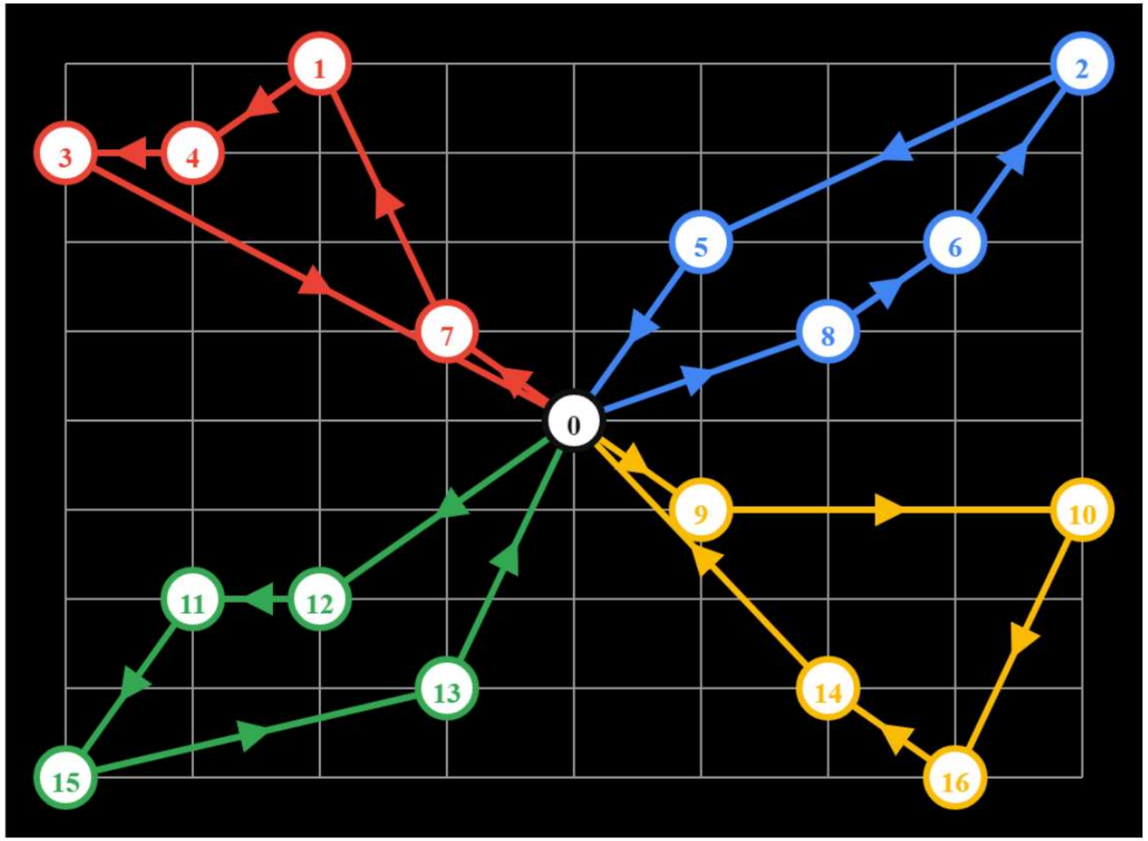
5. Vehicle Routing Problem

- Imagine a company that needs to visit its customers in a city made up of identical rectangular blocks.
- A diagram of the city is shown below, with the company location marked in black and the locations to visit in blue.



5. Vehicle Routing Problem

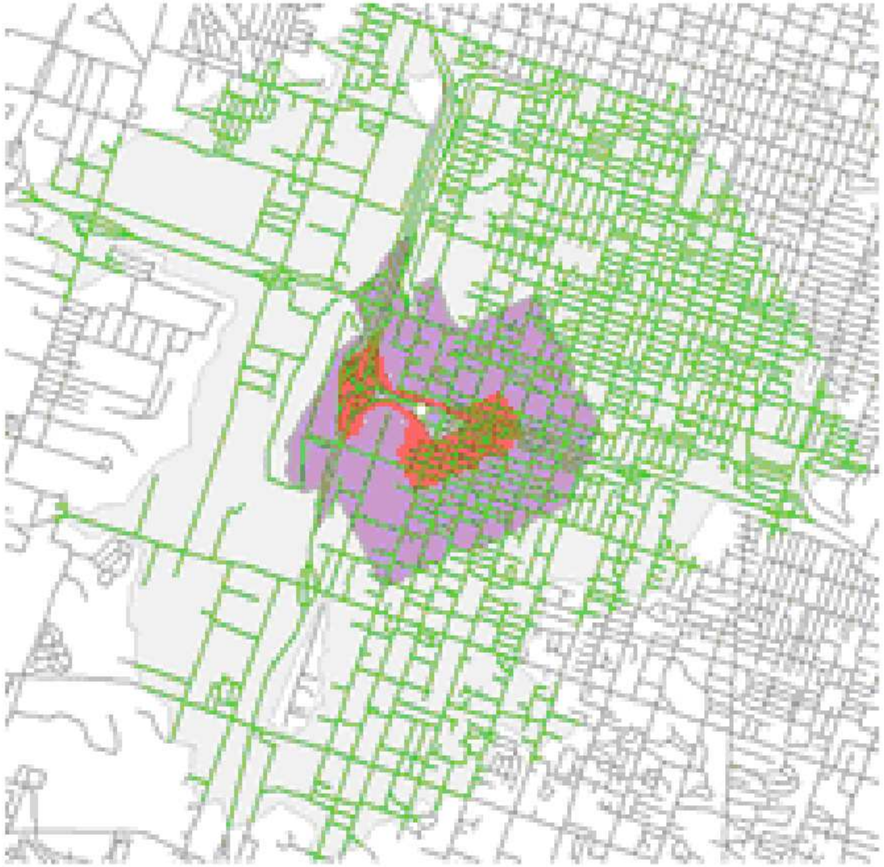
- The diagram below shows the assigned routes, in which the location indices have been converted to the corresponding *x*-*y* coordinates. Solved with VSP algorithms using Google OR-tools

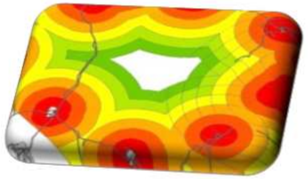


https://developers.google.com/optimization/install

Travel time zones

- A related facility to service area definition, also provided in many GIS packages, is that of identifying travel time zones.
- These are typically generated as a polygon layer, overlaid on the network, indicating bands of travel times or distances.





Network Analyst Resources

https://desktop.arcgis.com/en/arcmap/latest/extensions/network-analyst/about-the-network-analyst-tutorial-exercises.htm

Please Practice all the exercises listed on the Guidebook.

Assignment 03

Following the tutorial on https://www.r-orms.org/mixed-integer-linear-programming/practicals/problem-tsp/, each student has is required to pick a country. Assuming that you are a salesman for Coca Cola, you are required to compute the optimal route you will use to visit all towns in the selected country following the principles of TSP.

- The result will be a script that computes the TSP problem
- Produces a pdf map showing the Towns, and the optimal route overlaid with the country layer
- Deadline next week

Submission

EGE 2421: https://forms.gle/KXbninBFgCZTobtr6

EGS 2401 : https://forms.gle/eVjPVKRjqVSGPep47