CIE4801 Transportation and spatial modelling
(Uncongested) Assignment

Rob van Nes, Transport & Planning

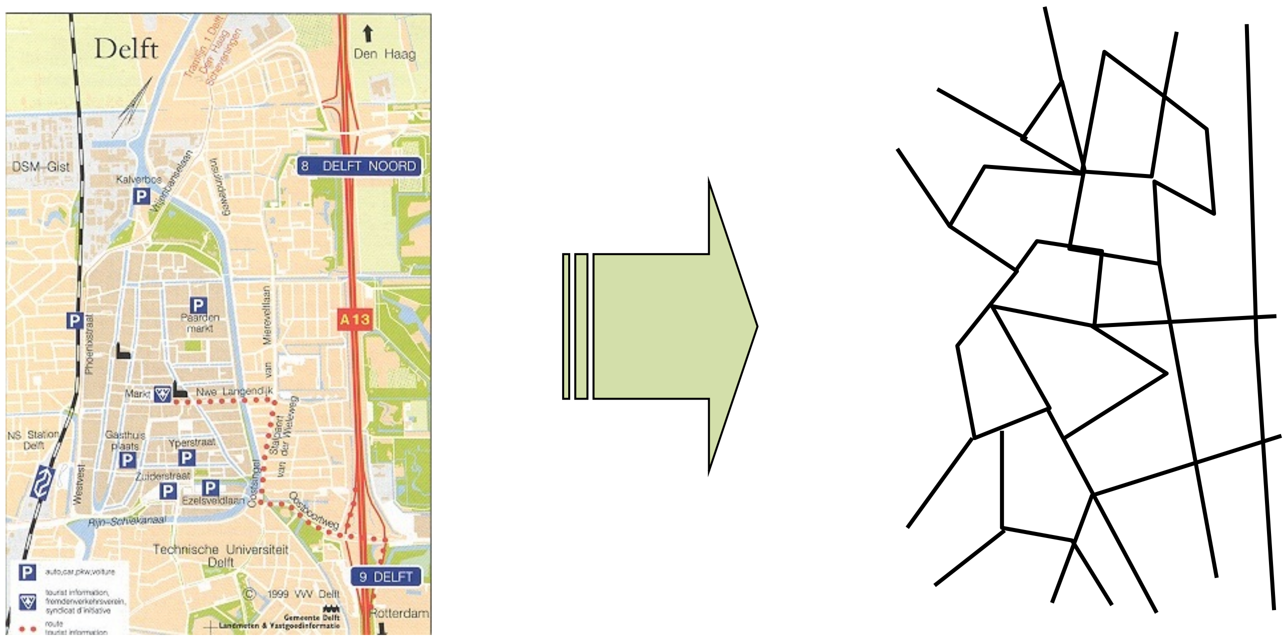
31-08-18

2.

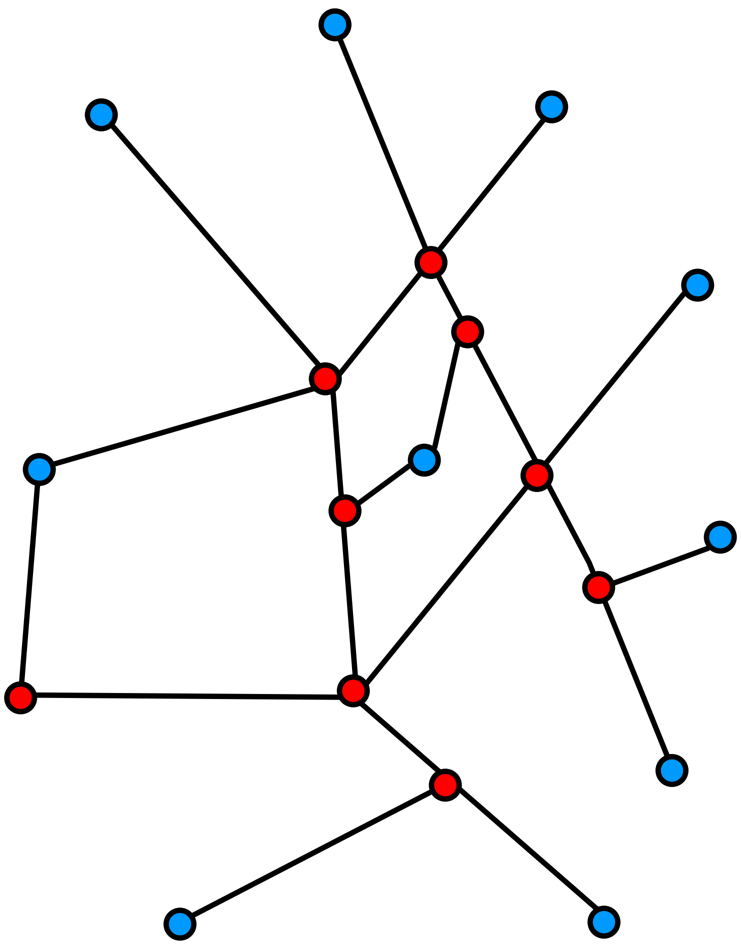
Networks

Constructing a transport network

Given a map of the study area,
how to represent the infrastructure and the travel demand in a model?



Network attributes

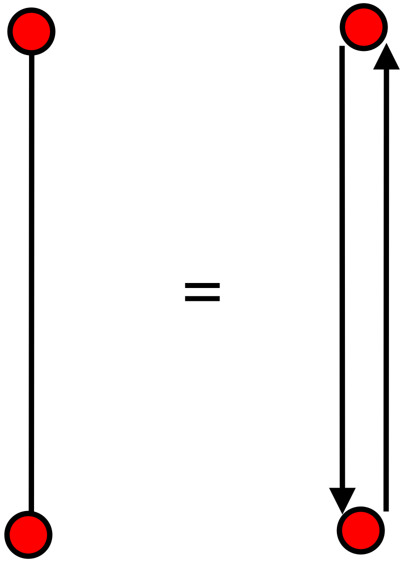


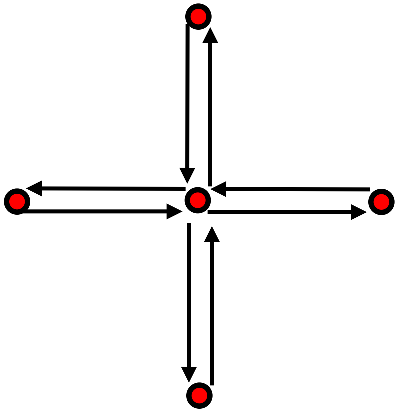
node
- x-coordinate
- y-coordinate

centroid node
- zonal data
- origin/destination

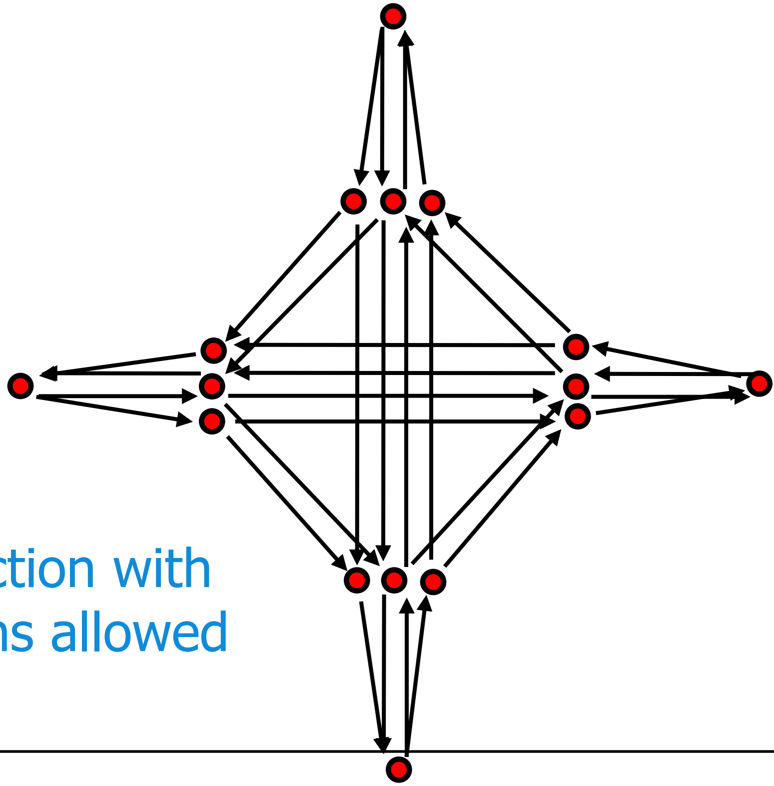
link
- node-from
- node-to
- length
- maximum speed
- number of lanes
- capacity

Links and junctions





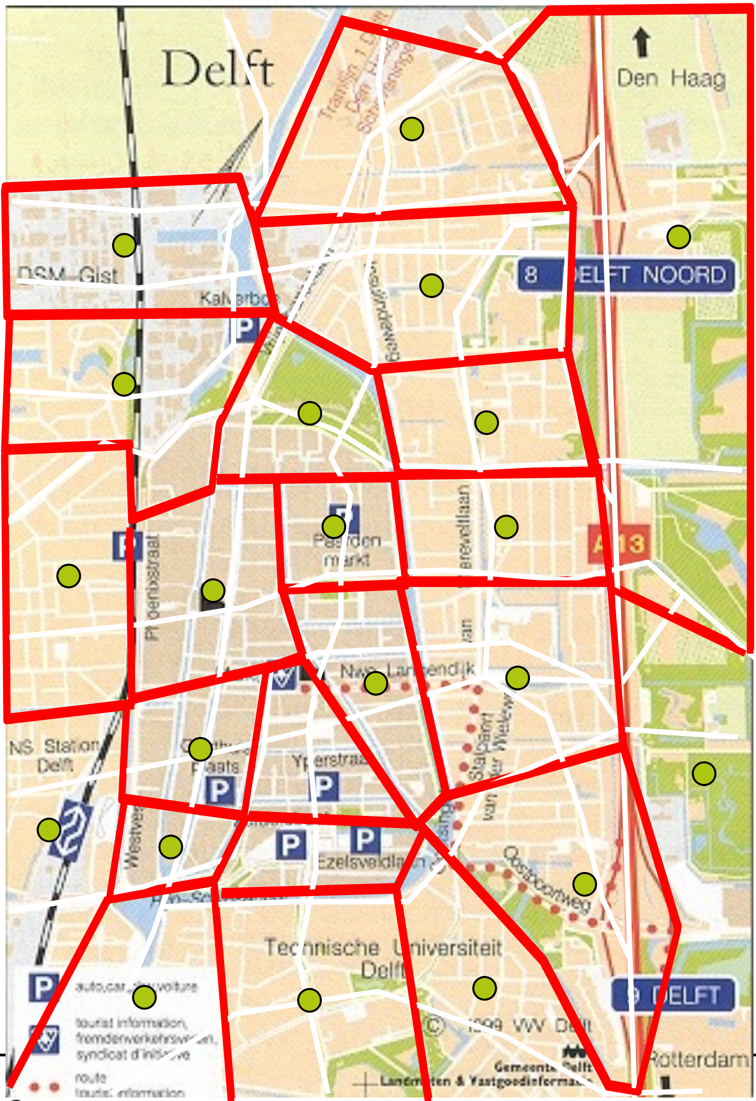
Junction with all turns allowed

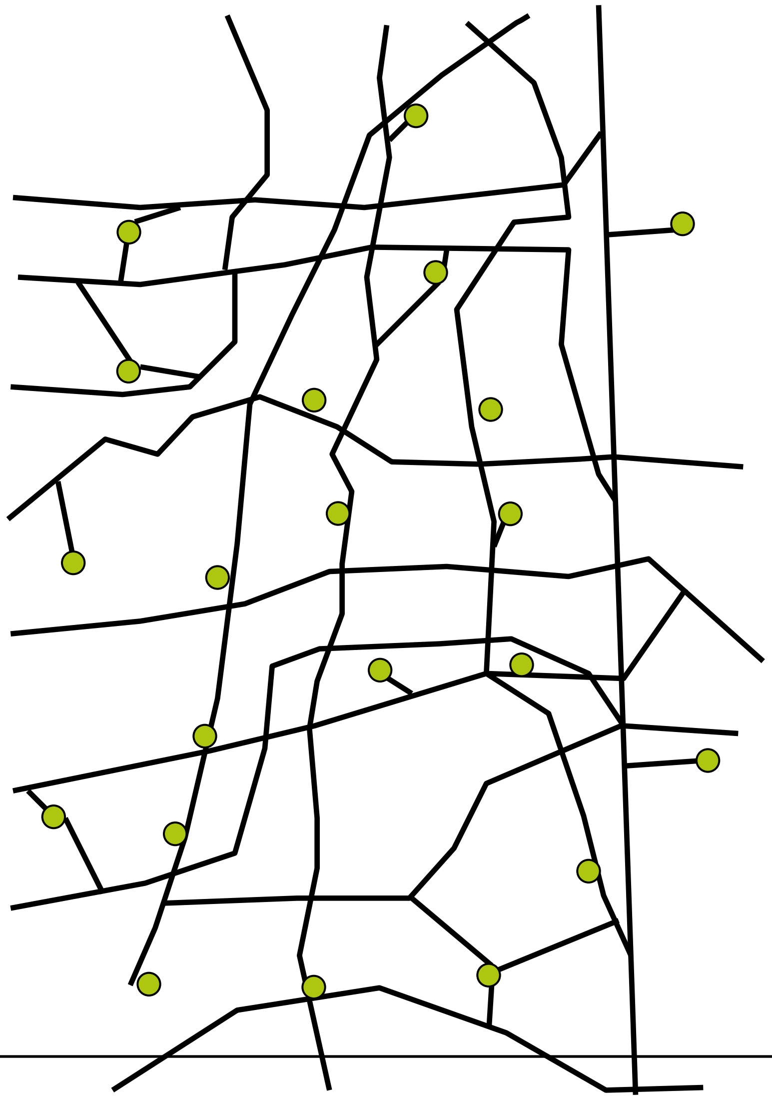


Junction with no left turns allowed

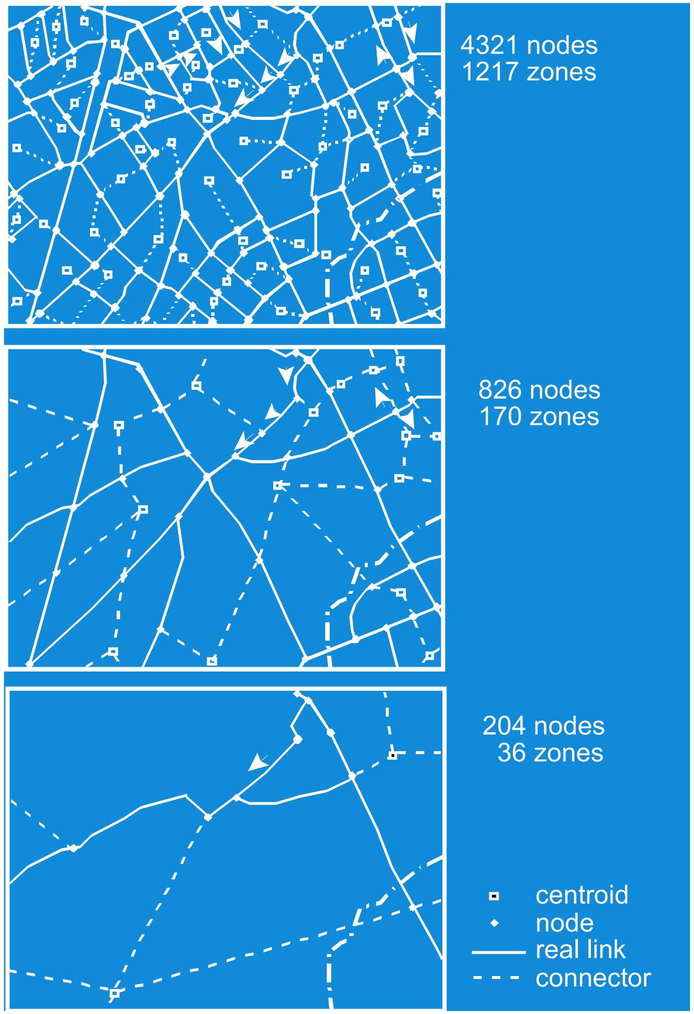
Using techniques like this you can model various types of intersections

Define zones and select roads





Select links



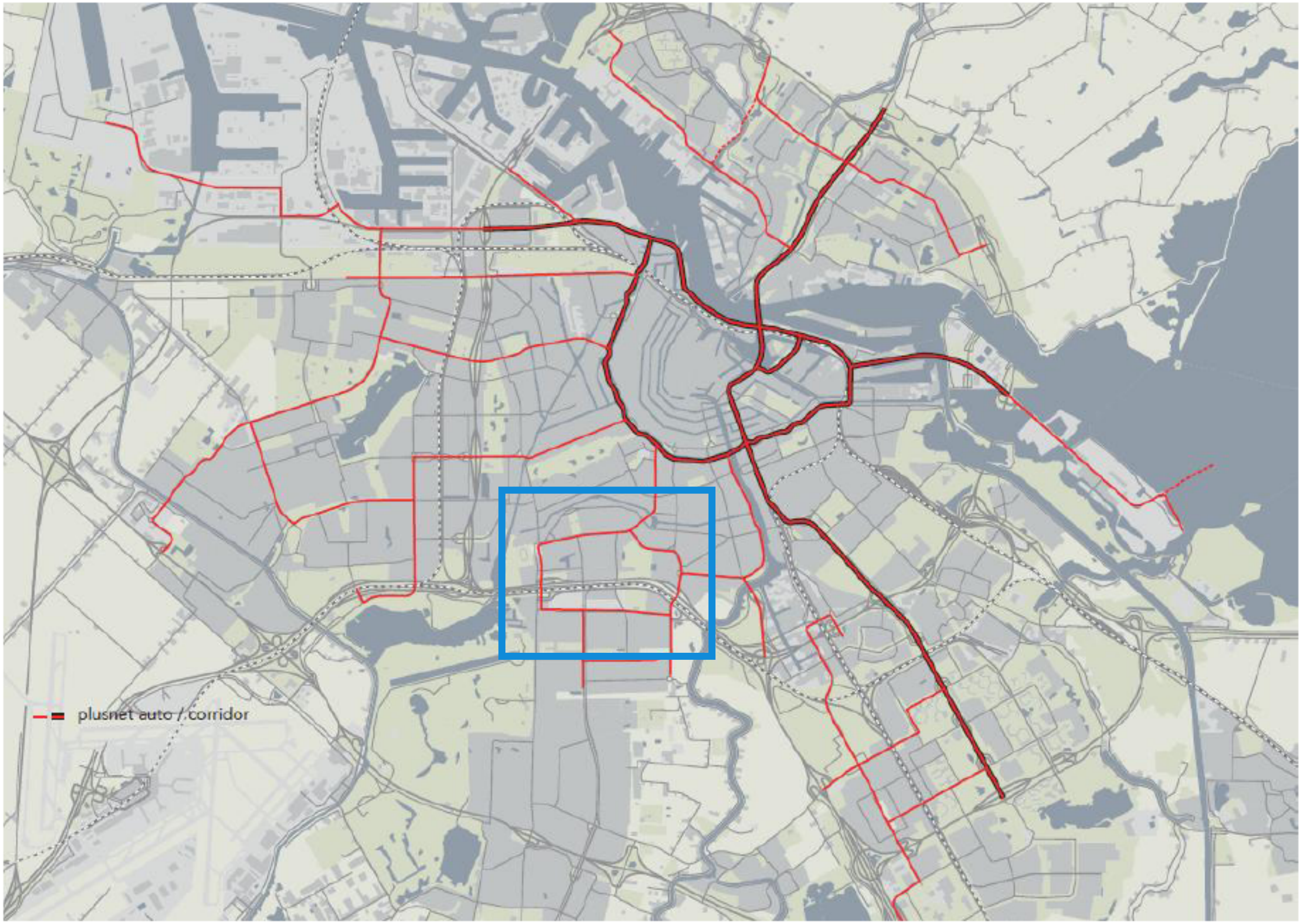
How many zones / nodes / links?

- depends on the application

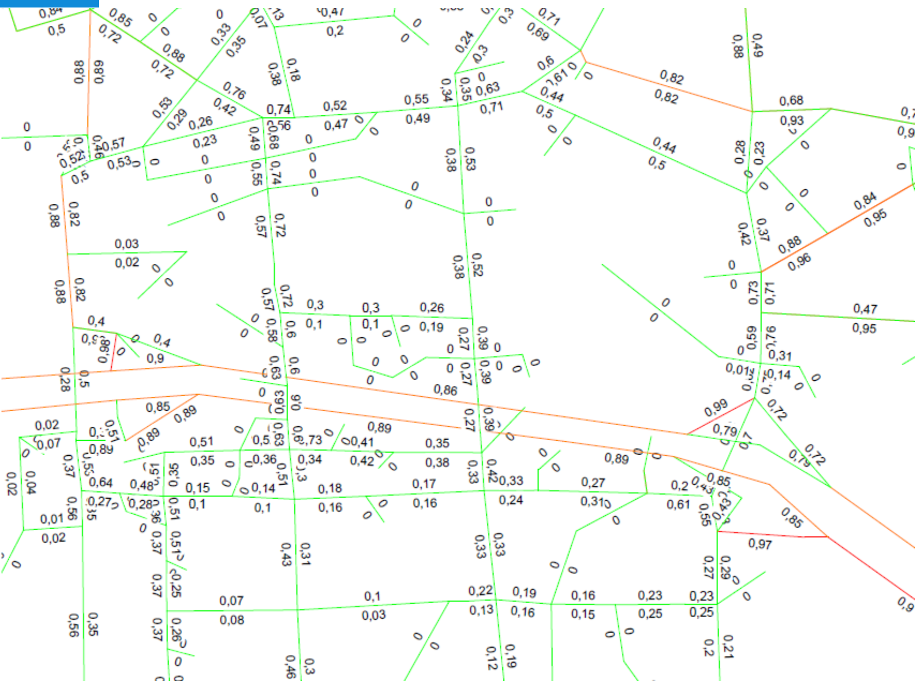
- include one network level lower than the network level you're interested in

modelling
=
the art of leaving things out

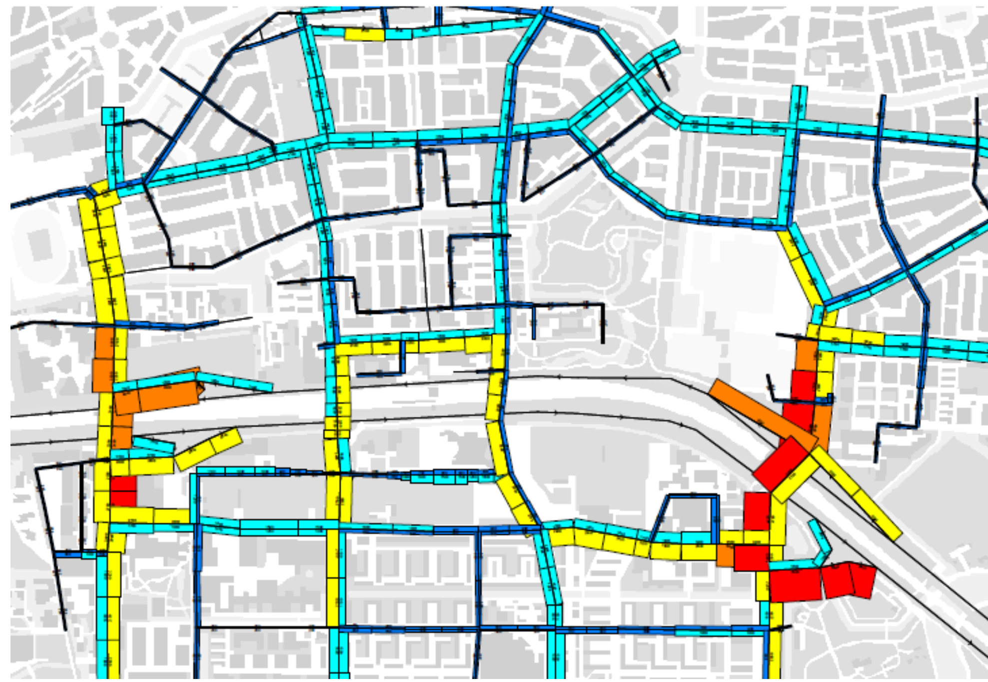
Which roads should be included?



Urban or regional model?

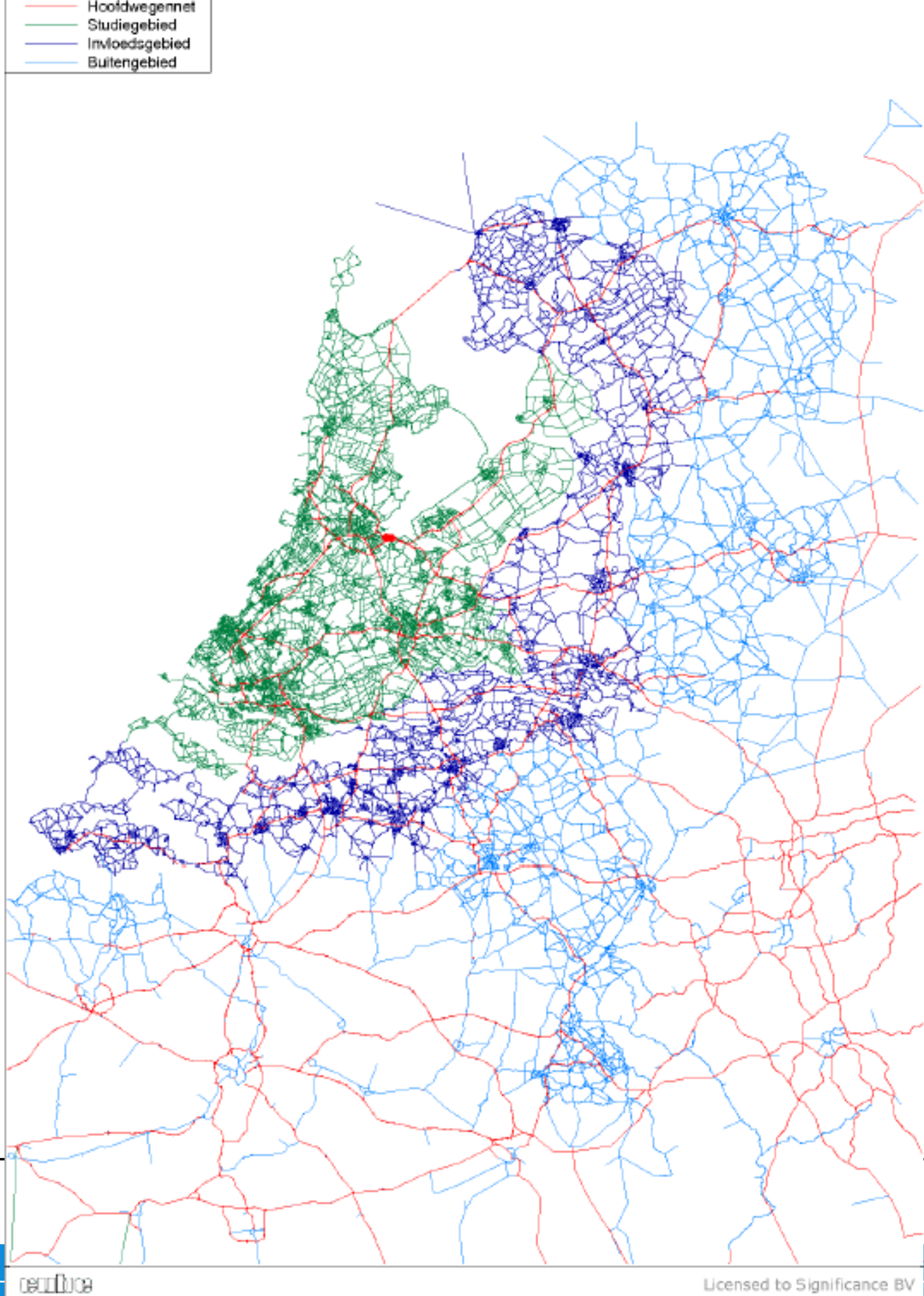


Regional



Urban

Example car network regional model



Important issue

- Connecting the zones to the network:
  - Single connector or multiple connectors?
  - Connecting to which type of node/link?
- Choices have major consequences for the assignment to the network!
- Preferably multiple connectors and connected to lower level network in your model

3.1

Main building block: shortest path

Shortest Path algorithms

- "Oldies"
  - Moore (1959)
  - Dijkstra (1959)
  - (Floyd-Warschall (1962))
- Still a topic for research

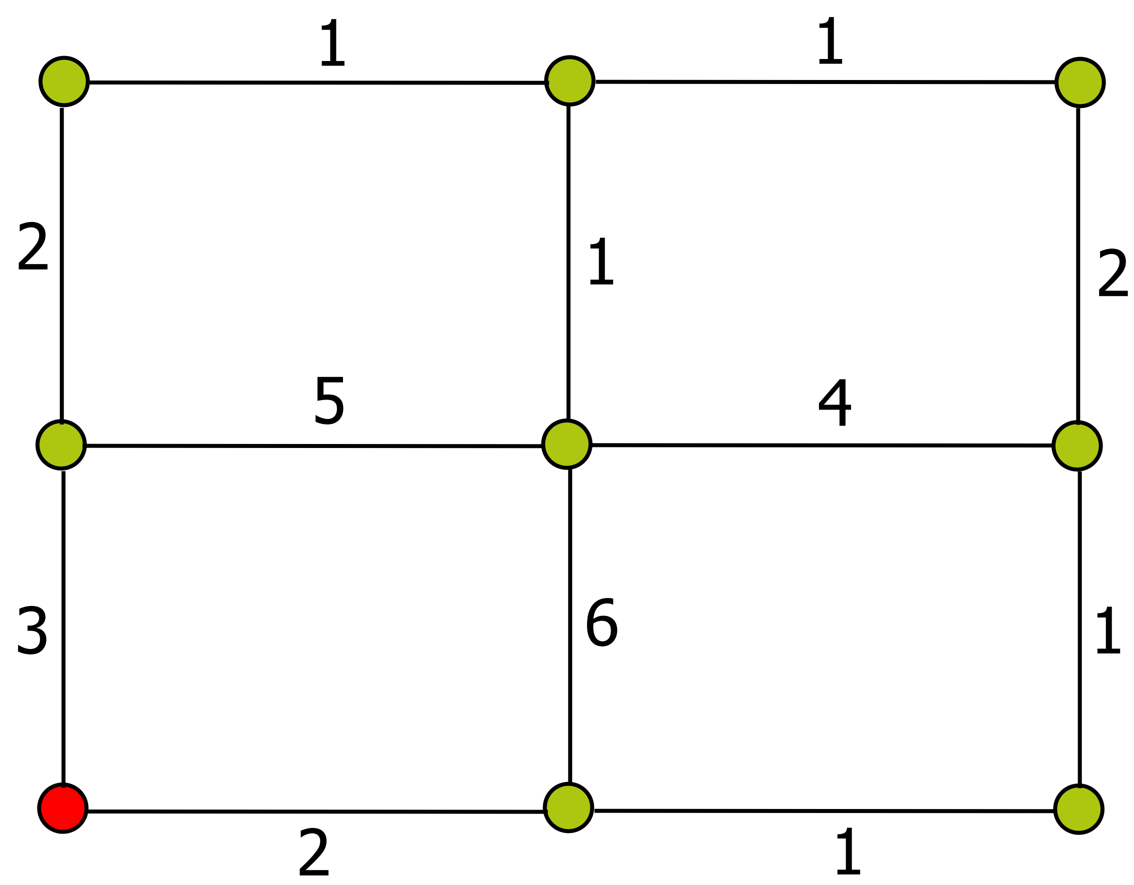
Main concept for 'tree algorithms'

(Moore and Dijkstra)

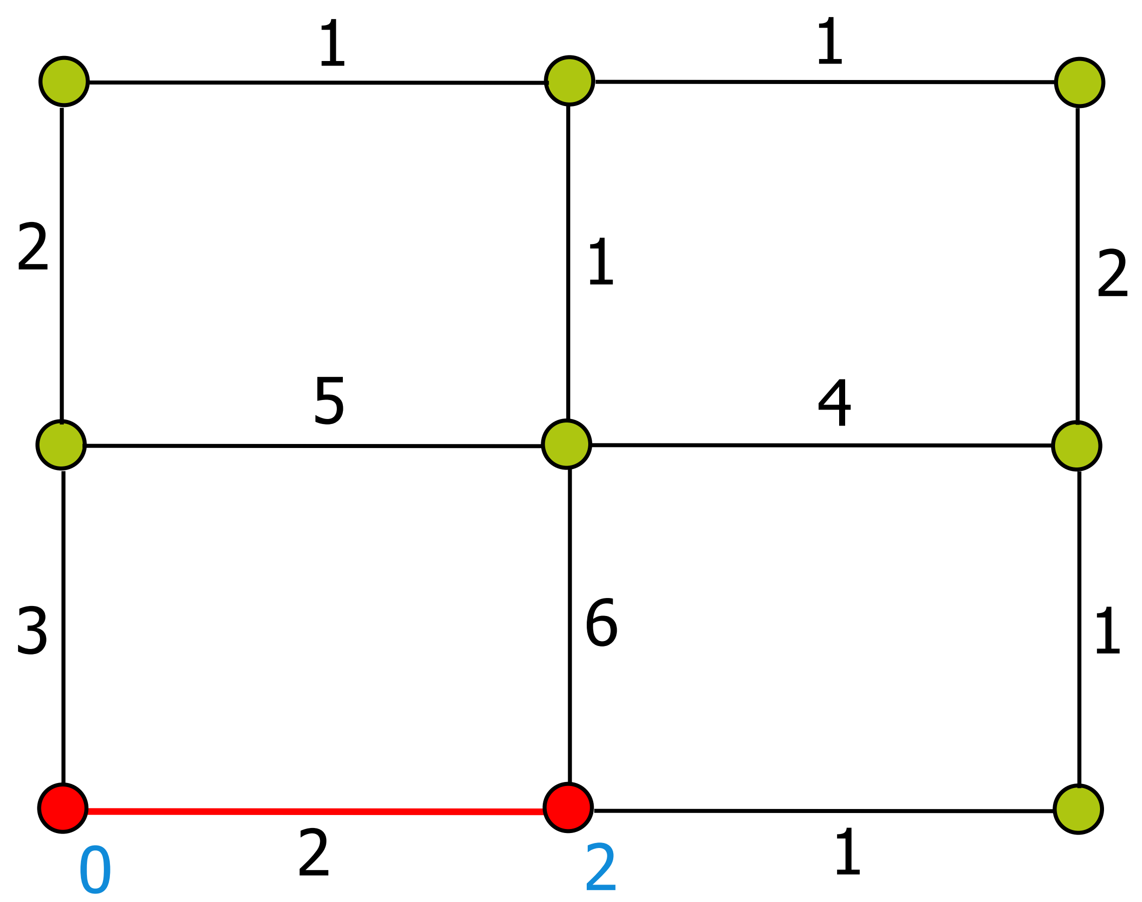
- For all nodes
  - Set travel time $tt(x)$ to ∞ and set the back node $bn(x)$ to *0*
- For the origin i set time to *0* and back node to *-1*
- Node i is the first active node a
- Select all links (a,j) and check travel times
  - If $tt(a)+time(a,j) < tt(j)$
    $tt(j)=tt(a)+time(a,j)$ and $bn(j)=a$ and node j becomes an active node
  - Node a is no longer active
- Select a new active node from the stack of active nodes and repeat previous step until there are no active nodes left

- For Dijkstra: select the link having the lowest travel time and select the active node having the lowest travel time

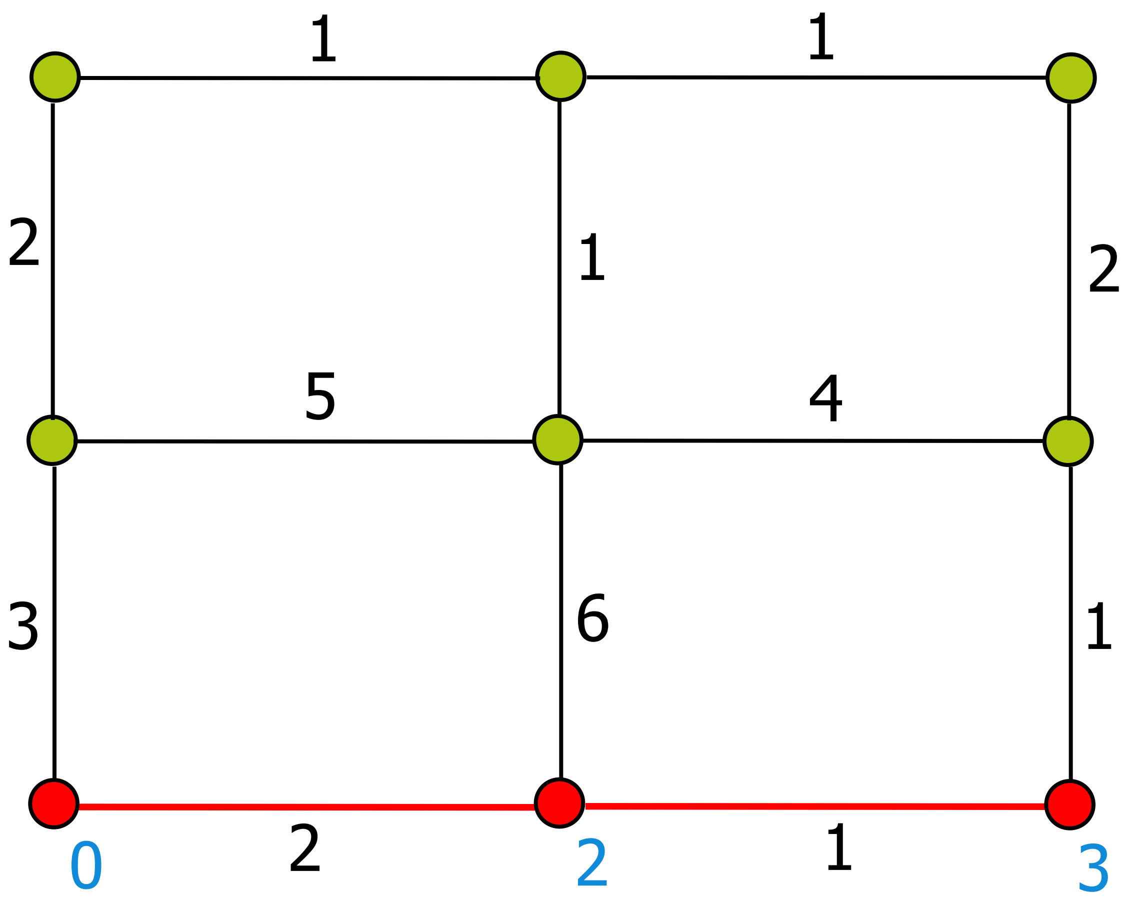
Shortest paths



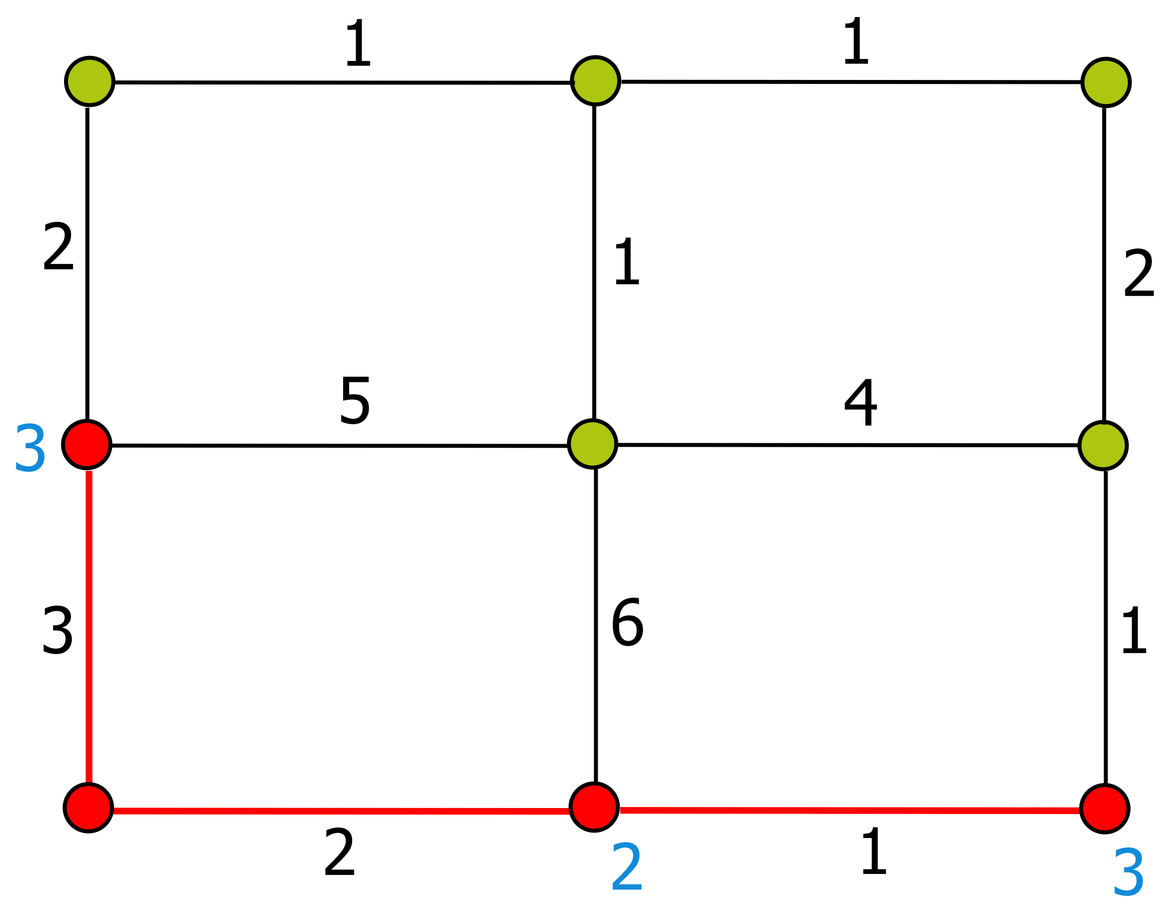
Shortest paths



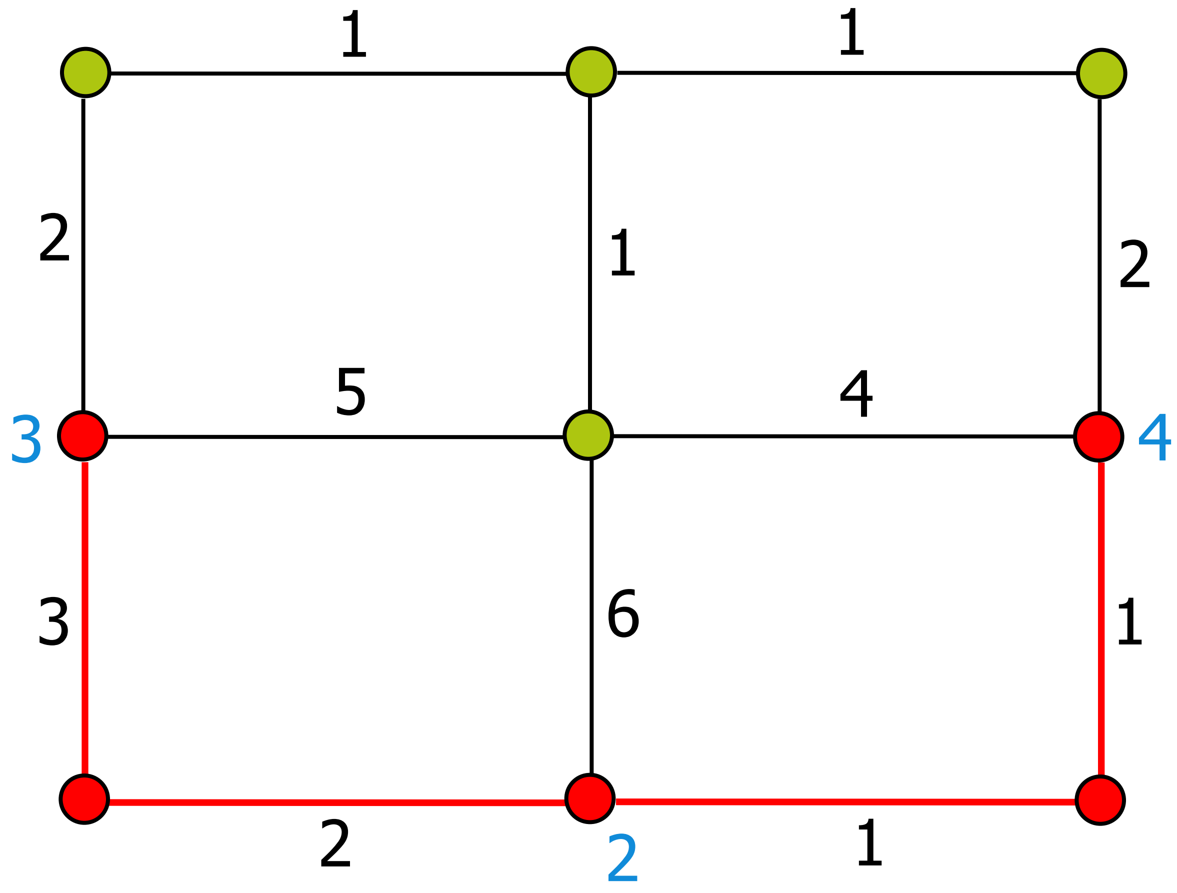
Shortest paths



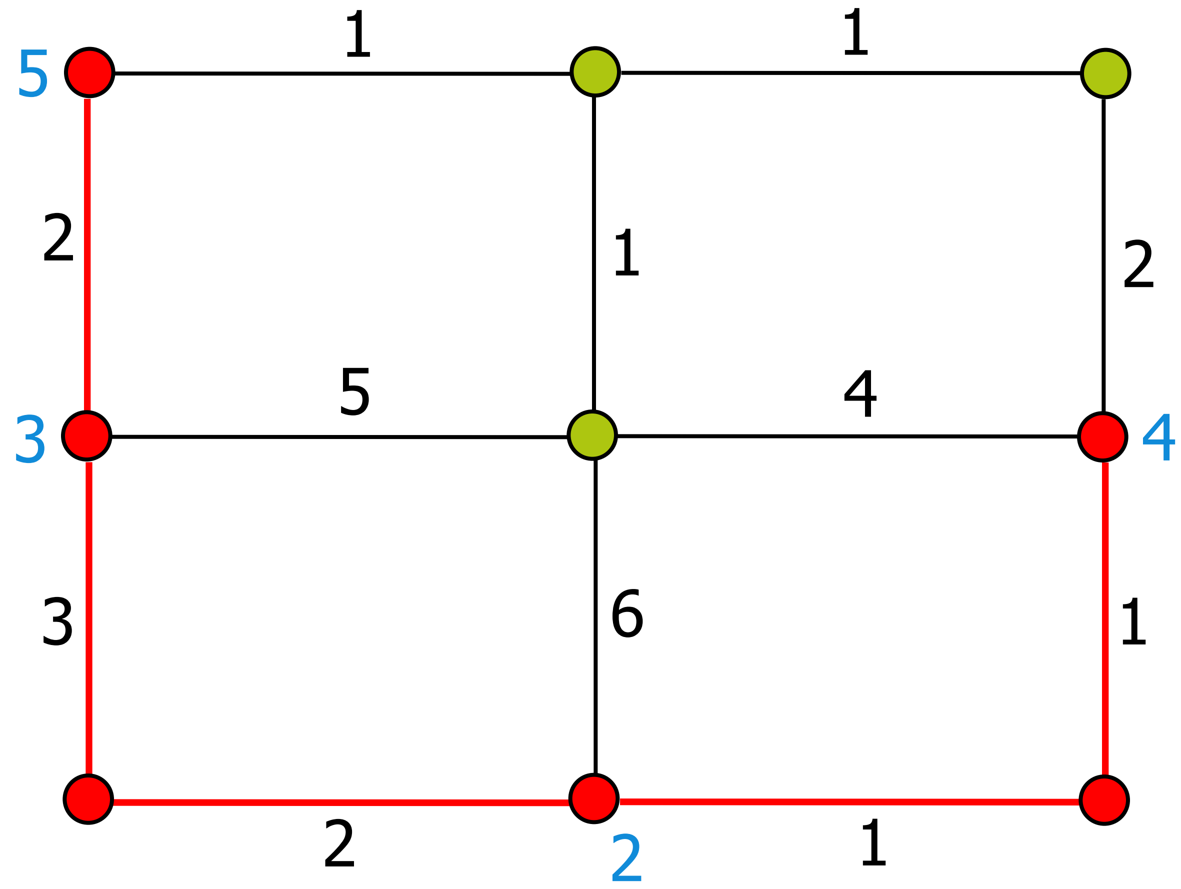
Shortest paths



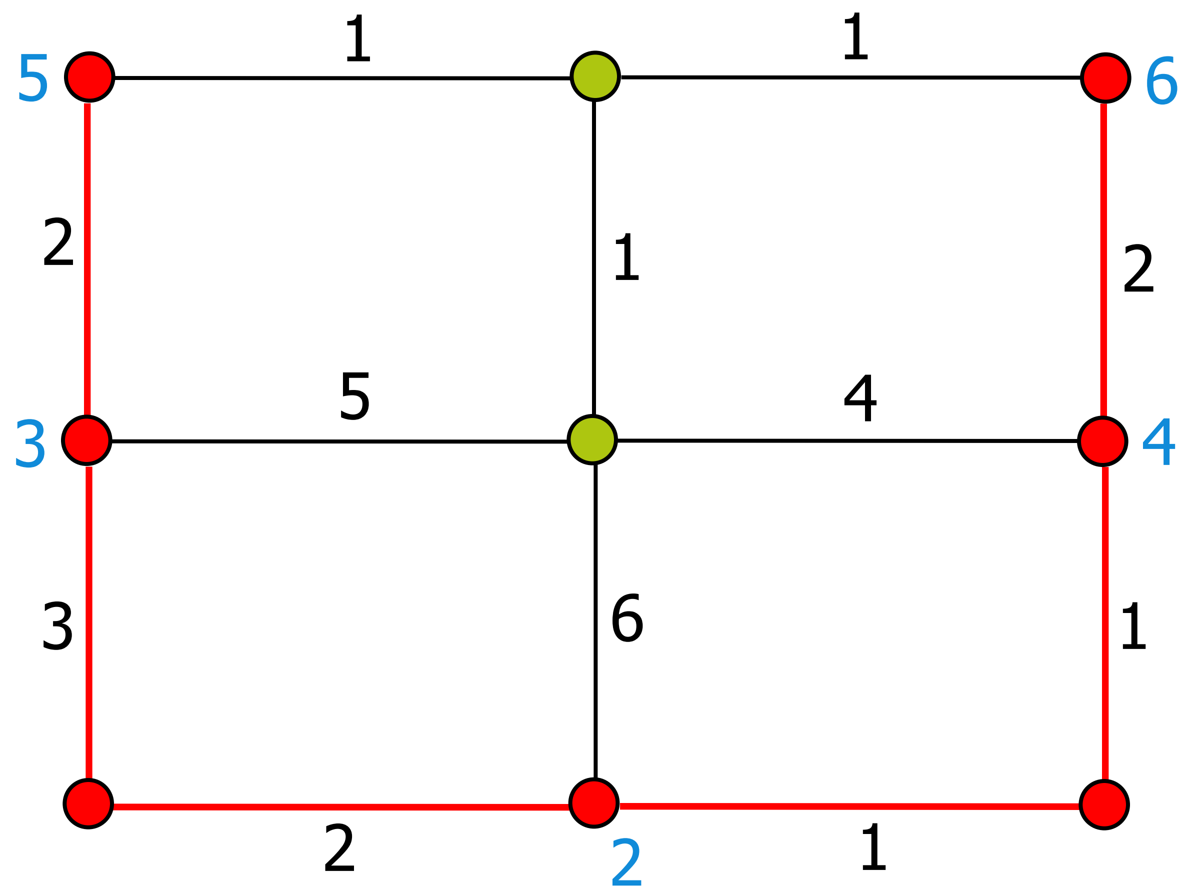
Shortest paths



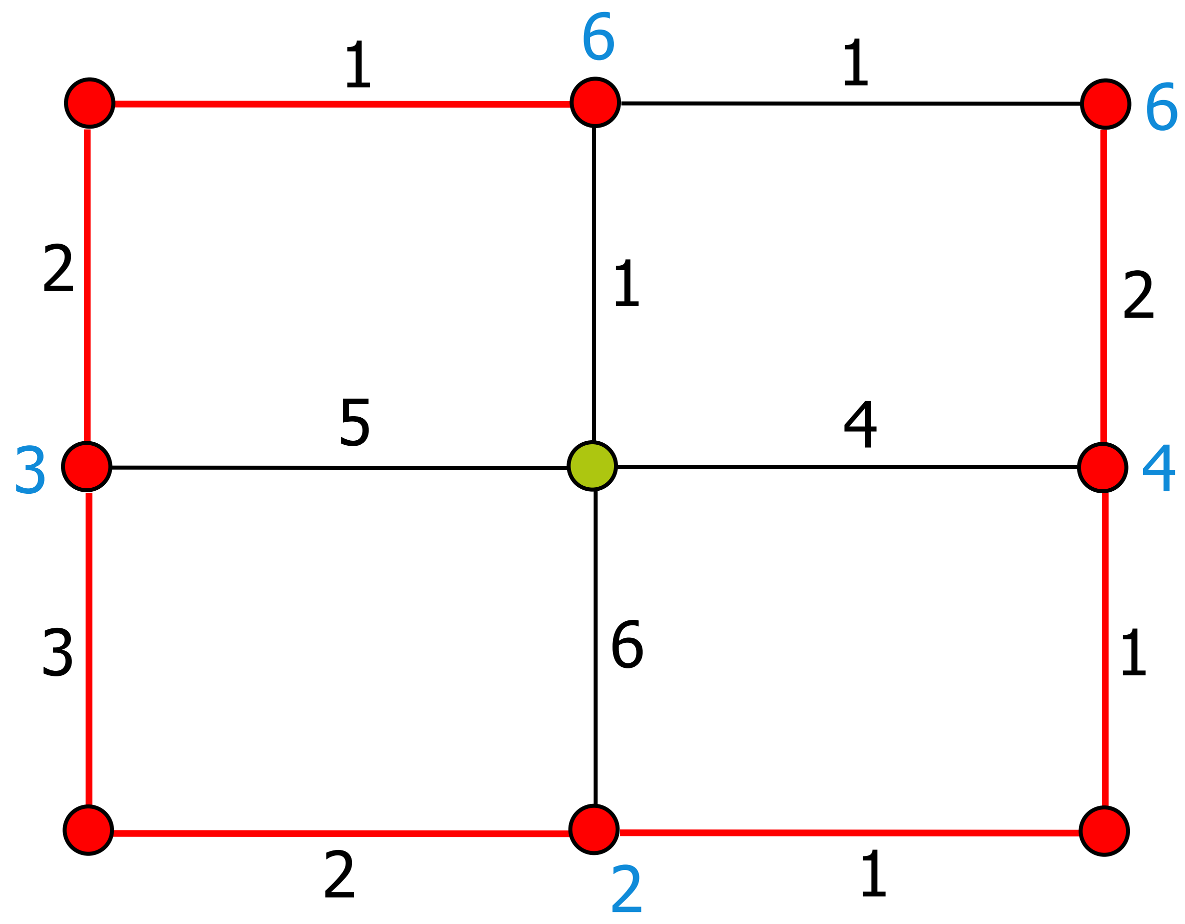
Shortest paths



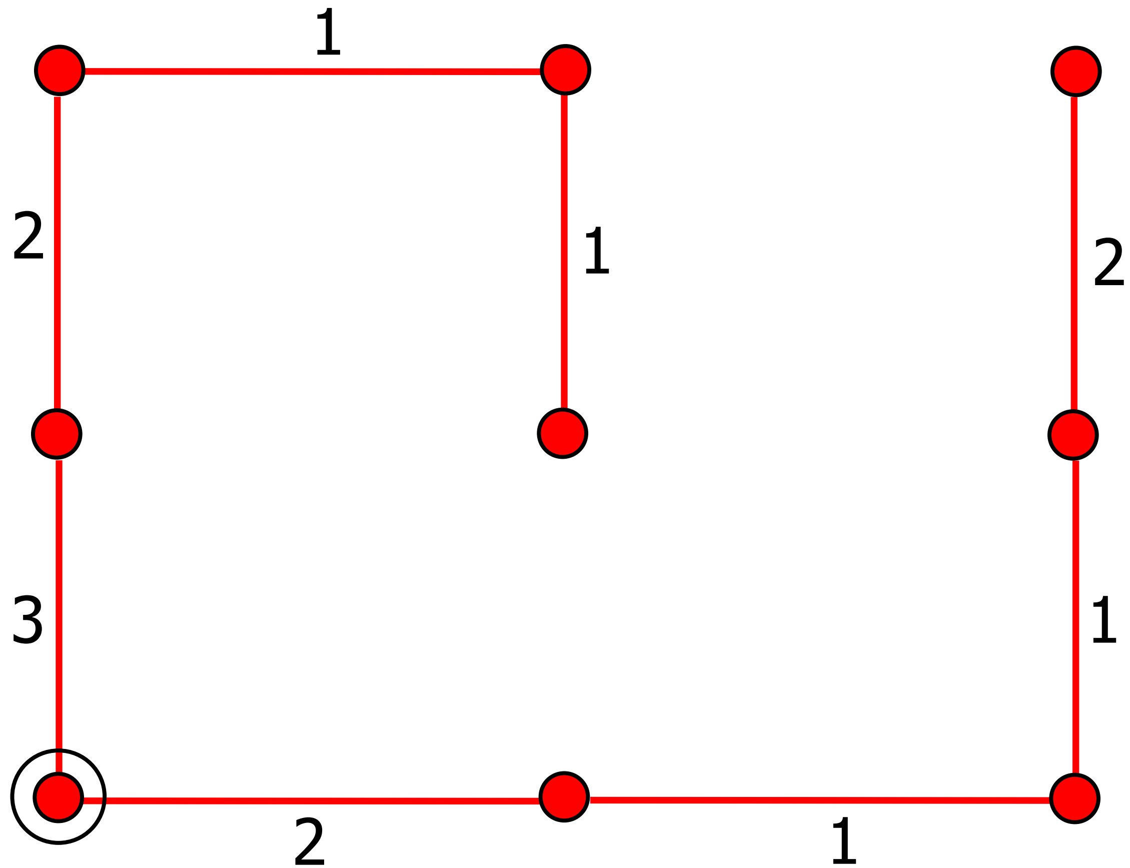
Shortest paths



Shortest paths



Shortest paths



Shortest path tree

Contains shortest paths to *all* nodes from *a certain origin*

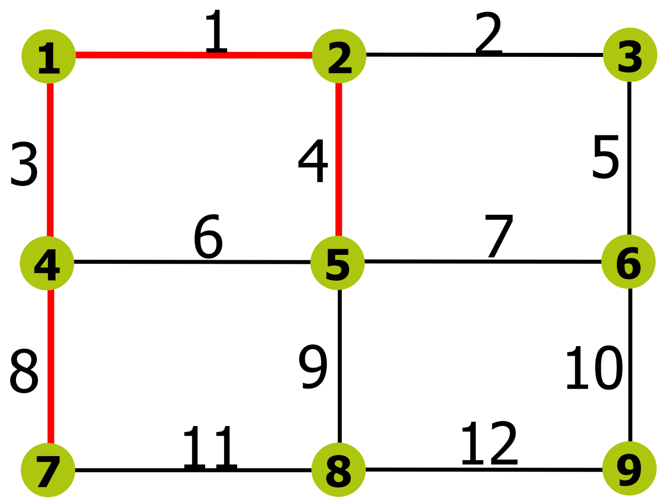
Main concept for matrix algorithm
(Floyd-Warschall)

- Create two matrices from all nodes to all nodes
  - $tt(i,j) = t(i,j)$ for all links (i,j) or $tt(i,j) = \infty$
  - $bn(i,j)=i$ for all links (i,j)
- For every node k check travel times
  - If $tt(i,k) + tt(k,j) < tt(i,j)$
    $tt(i,j) = tt(i,k) + tt(k,j)$ and $bn(i,j)=bn(k,j)$
- Repeat previous step until no changes are made
- Interesting option for public transport

3.2

Shortest path representation

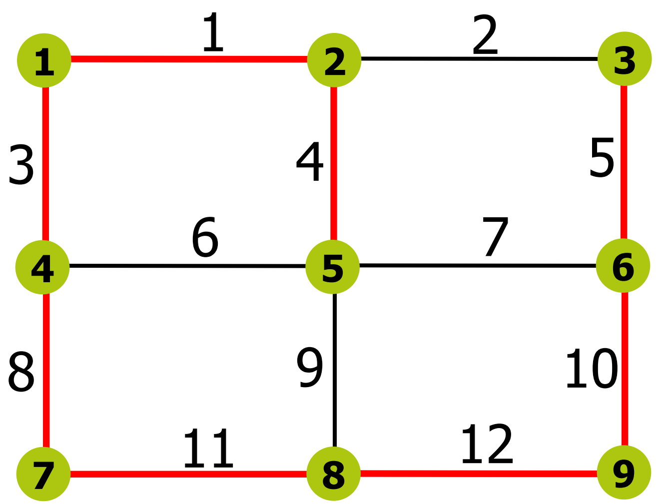
Representation of shortest paths



Shortest route from node 7 to node 5:

node 7 → node 4 → node 1 → node 2 → node 5
or
link 8 → link 3 → link 1 → link 4

Representation of shortest paths



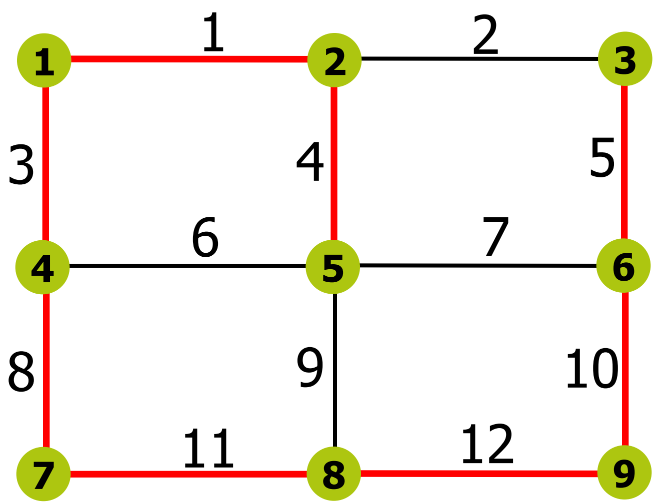
Very compact, less practical:
Back node representation

	node 1	node 2	node 3	node 4	node 5	node 6	node 7	node 8	node 9
route "7 → *x*"	4	1	6	7	2	9	-1	7	8

Number for elements for storing all shortest routes in the network:

$$N \times N = 9 \cdot 9 = 81$$

Representation of shortest paths



Very compact, more practical:
Back node representation
Plus link numbers

	node 1	node 2	node 3	node 4	node 5	node 6	node 7	node 8	node 9
route "7 → x"	4	1	6	7	2	9	-1	7	8
	3	1	5	8	4	10	-1	11	12

Number for elements for storing all shortest routes in the network:

$$2 \times N \times N = 2 \cdot 9 \cdot 9 = 162$$

Assignment algorithm using back nodes

- Set all link flows equal to 0
- For each origin
  - Determine shortest path tree
  - For each destination
    - Backtrack the route and add the OD-flow to the link flow of each link in the route

- Advantage: fast, efficient memory usage

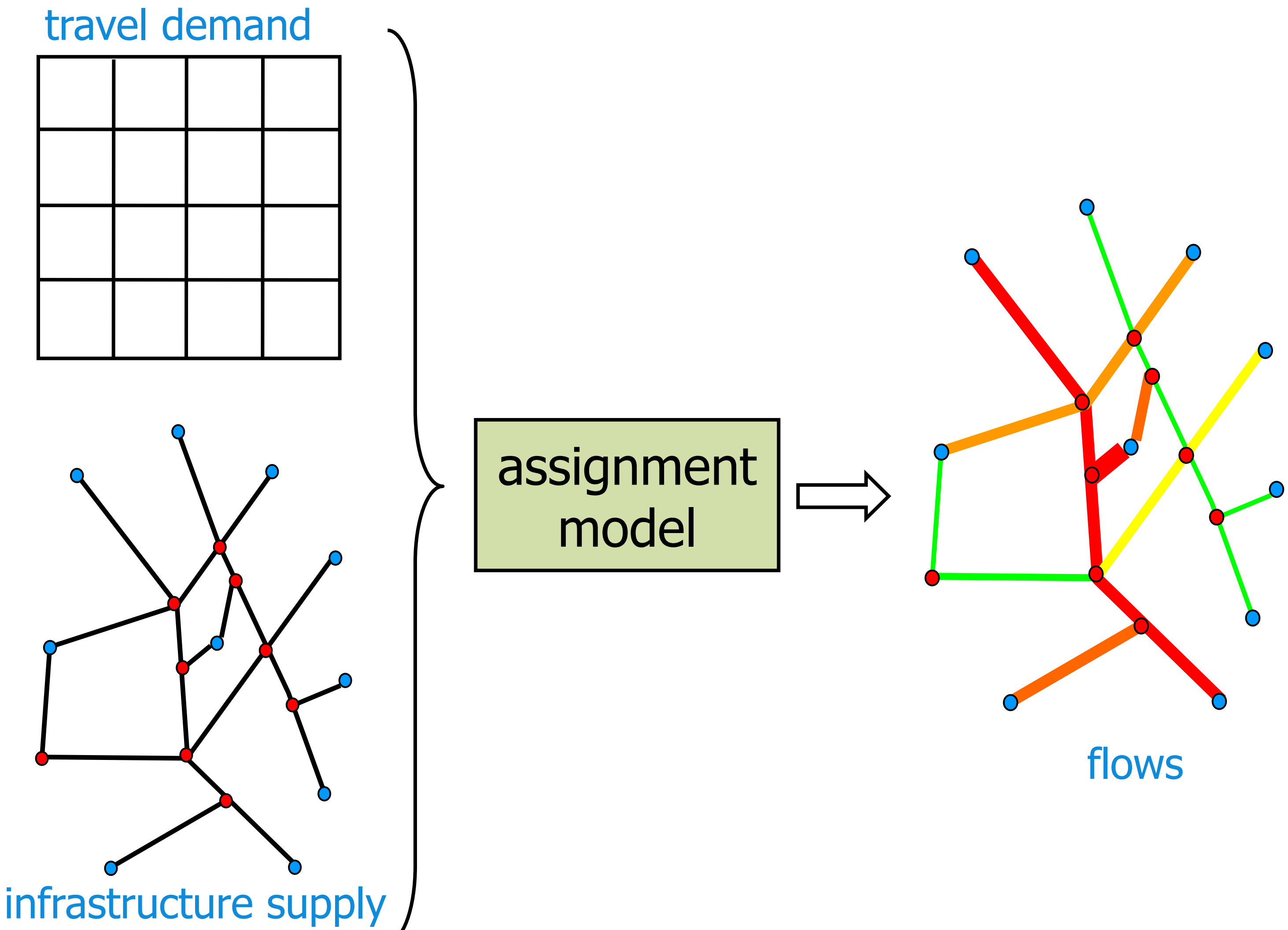
4.1

Introduction to assignment

What do we want to know?

- Often the main goal of a modelling study
  - Flows on links or lines
  - Bottlenecks
  - Travel times
- This is where demand really meets supply

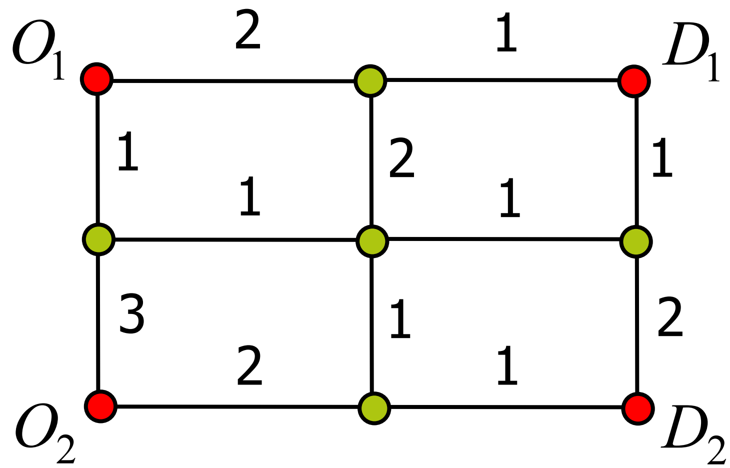
Introduction to assignment



Solving the assignment problem

	D_1	D_2
O_1	10	5
O_2	8	3

OD matrix



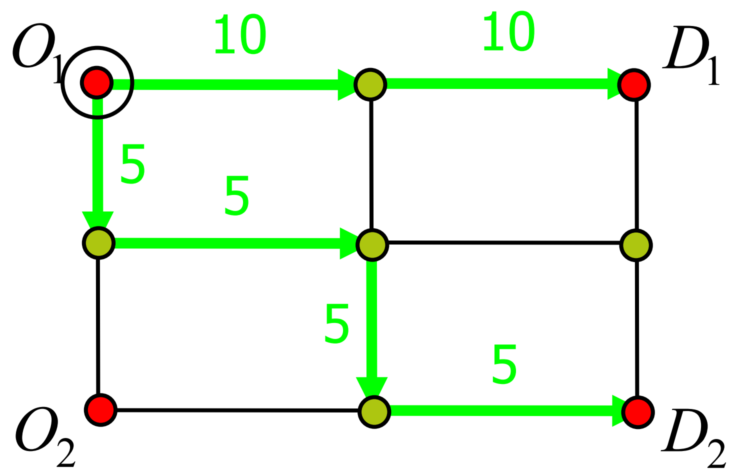
network

Given the travel demand and the infrastructure supply, determine assignment using the shortest paths.

Solving the assignment problem

	D_1	D_2
O_1	10	5
O_2	8	3

OD matrix



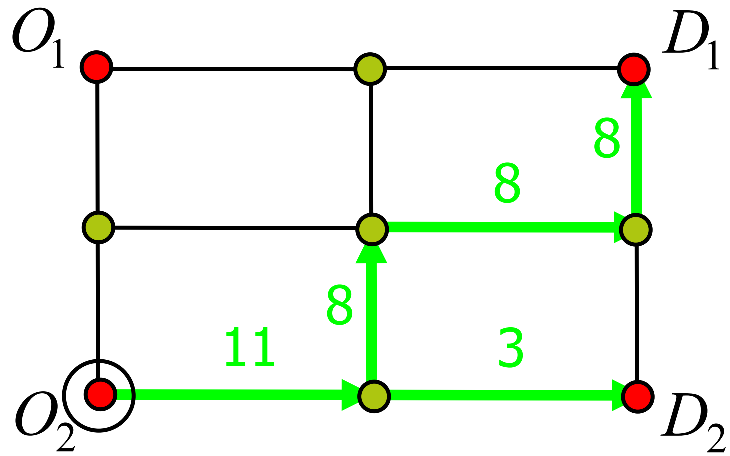
network

For the first origin, find all shortest paths to all destinations, and assign the corresponding trips to the shortest paths.

Solving the assignment problem

	D_1	D_2
O_1	10	5
O_2	8	3

OD matrix



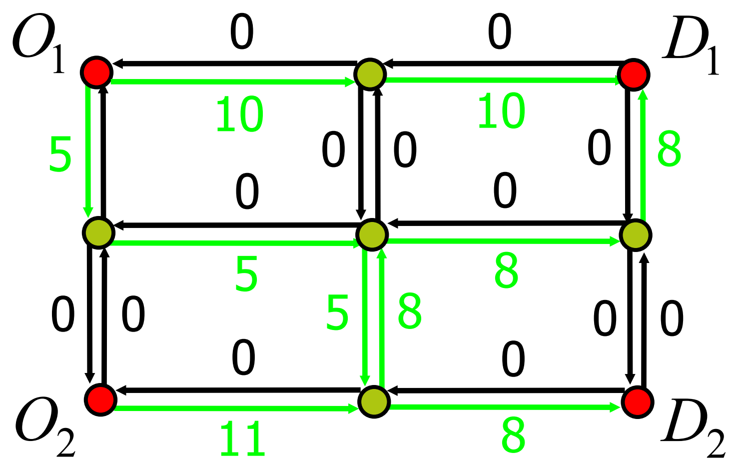
network

For the second origin, find all shortest paths to all destinations, and assign the corresponding trips to the shortest paths.

Solving the assignment problem

	D_1	D_2
O_1	10	5
O_2	8	3

OD matrix



network

Combine all assignments between all OD pairs, and compute the total link flows.

Assignment algorithm using back nodes

- Set all link flows equal to 0
- For each origin
  - Determine shortest path tree
  - For each destination
    - Backtrack the route and add the OD-flow to the link flow of each link in the route

- Advantage: fast, efficient memory usage

This was how you might do it

- Does it capture the relevant phenomena for the assignment?
- Differences in perception between users
  - Single route per OD or multiple routes
  - Multiple route concept implies route choice modelling
- Effect of congestion
  - Included or excluded

Main assignment types

		Congestion effect modelled?	
		No	Yes
Multiple routes modelled?	No	All-or-nothing assignment	Deterministic user-equilibrium assignment
	Yes	Stochastic assignment	Stochastic user-equilibrium assignment

4.2

All-or-Nothing assignment

All-or-Nothing (AON) assignment

Assumptions:

- All traffic between an OD pair is assigned to just one route (usually the 'shortest' route (in terms of cost (min, €, km)))
- There is no congestion

Topics

- Mathematical formulation
- Some comments

AON: Formulated as an optimisation problem (route based)

$T_{ijr} =$ number of OD trips from i to j taking route r

$t_{ijr} =$ travel time on route r from i to j

$T_{ij} =$ number of OD trips from i to j

$$\min_{T_{ijr}} Z = \sum_i \sum_j \sum_r t_{ijr} T_{ijr}$$

minimize the total travel time experienced by all travellers

s.t.

$$\sum_r T_{ijr} = T_{ij}$$

flow conservation

$$T_{ijr} \geq 0$$

non-negativities

Note that using this formulation you could have multiple routes having a similar travel time!

AON: Formulated as an optimisation problem (link based)

q_a = traffic flow on link a

q_a^s = traffic flow on link a with destination s

t_a = travel time on link a

T_{ms} = number of OD trips from node m to node s (m is an origin)

$$\min_{q_a} Z = \sum_a t_a q_a$$

minimize the total travel time experienced by all travellers

s.t.

$$\sum_s q_a^s = q_a; \; T_{ms} + \sum_{a \in M^-} q_a^s = \sum_{a \in M^+} q_a^s \quad \forall m \neq s$$

flow conservation

$$q_a^s \geq 0$$

M^-: incoming; M^+: outgoing

non-negativities

AON-assignment: some comments

- Very fast method
- Easy to interpret

- Misses essential concepts
- Very sensitive for small changes

- Acceptable if you have many zones?

- Note that AON of software package A might differ from AON of software package B

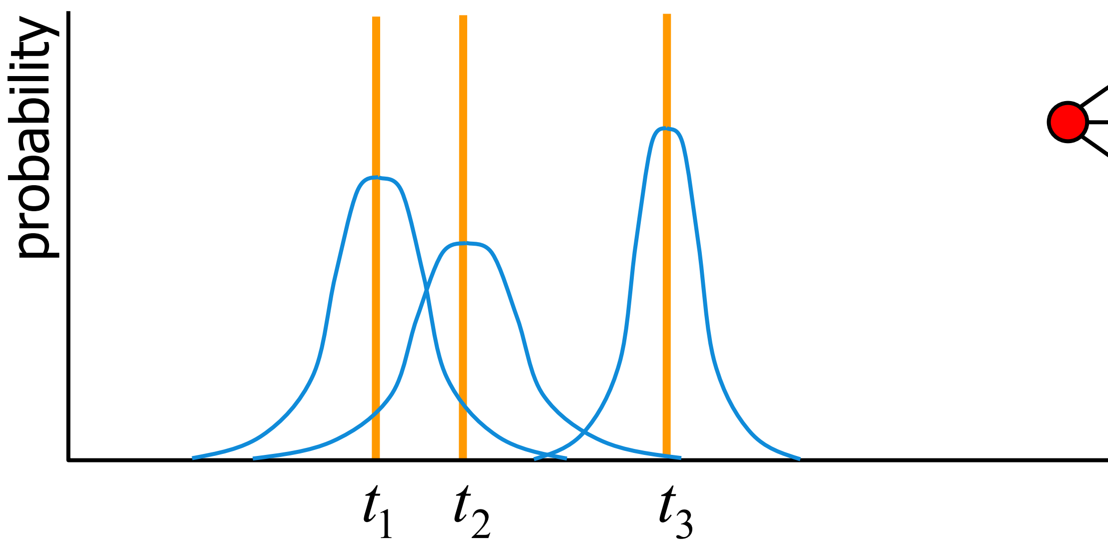
4.3

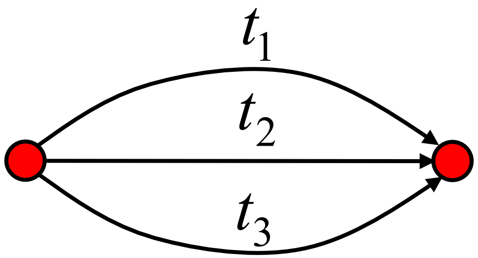
Multiple routes: route choice (stochastic assignment)

Stochastic assignment

Shortest path: All-or-nothing assignment:
For each OD pair, all travellers take the shortest route

Multiple routes: Stochastic assignment
For each OD pair, all travellers take their *perceived* shortest route





Discrete choice theory

$$p_i = P(U_i \geq U_j \text{ for all } j)$$

$$= P(V_i + \varepsilon_i \geq V_j + \varepsilon_j \text{ for all } j)$$

$$= P(V_i - V_j \geq \varepsilon_j - \varepsilon_i \text{ for all } j)$$

If ε_i's are all extreme value distributed (independent, with scale parameter β),

Logit-model

Easy to solve

$$p_i = \frac{e^{\beta V_i}}{\sum_j e^{\beta V_j}}$$

If ε_i's are all normally distributed (independent),

Probit-model

$$p_i = \ldots .$$

Can only be solved by simulation

Probit and logit in transport models

- Note that previous slide is independent on type of choice
- Modelling practice:
  - Demand modelling: dominated by logit
  - Assignment: Probit or logit
- In case of assignment the link is the basic element, thus ideally the error term should be defined at link level
- In this case normal distribution is the most likely assumption
  - Thus Probit

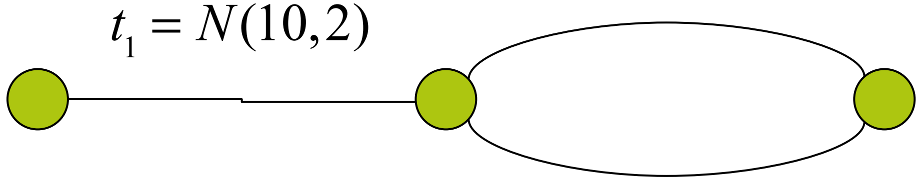
How is the probit assignment solved?

- Iteratively
- In each iteration a specific network state is considered i.e. a network having specific link times
- These link times are sampled from the distributions that are assumed at link level
- Note that overlapping routes in a specific network state are consistent

Example

$t_2 = N(5, 0.1)$

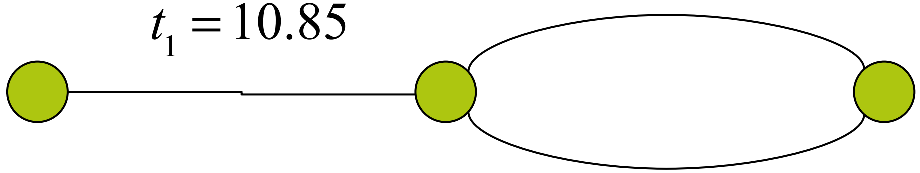
$t_1 = N(10, 2)$



$t_3 = N(5, 0.1)$

$t_2 = 4.98$

State 1 $t_1 = 10.85$

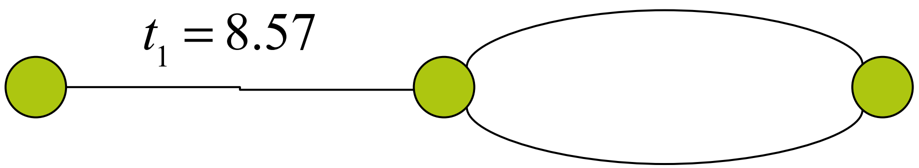


Route 1=15.83
Route 2=16.04

$t_3 = 5.19$

$t_2 = 4.89$

State 2 $t_1 = 8.57$



Route 1=13.46
Route 2=13.56

$t_3 = 4.99$

Scaling of the error term

The error term is related to the 'size' of link

$$\mathbf{t}_a = t_a + \mathbf{z} \cdot \sqrt{(\Theta L_a)}$$

$\Theta =$ Dispersion parameter

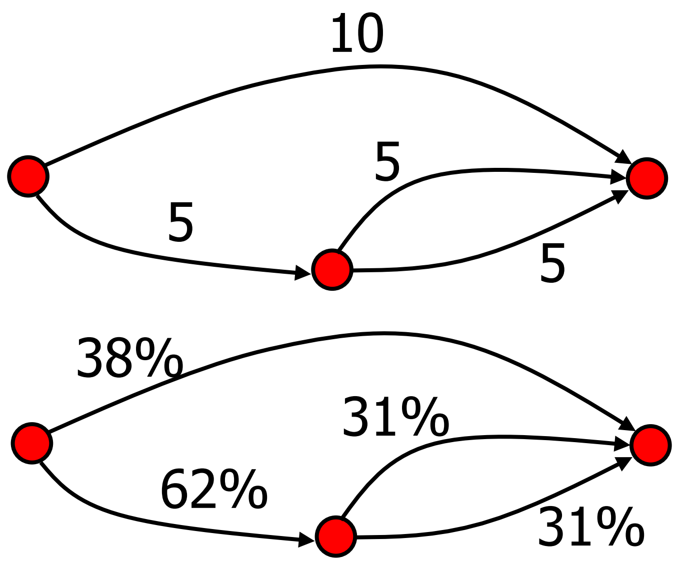
$L_a =$ Size of link a

$\mathbf{z} =$ Random variable e.g. $N(0,1)$

Size could be the length or free flow travel time of link

Probit assignment at link level

No explicit analytical expression, but can be computed by simulation.



For $\mathrm{n}=1$ to N do

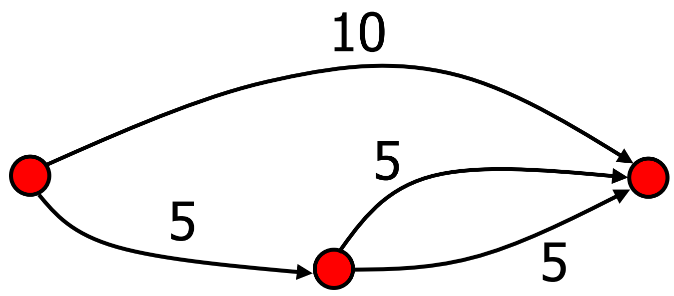
1. $t_a := t_a^0 + \sqrt{(\Theta t_a^0)}\mathbf{z}, \quad \mathbf{z} \sim N(0,1)$
2. Compute shortest route
3. If shortest route is alternative r

   then $\phi_{ijr} := \phi_{ijr} + 1/N$

Stochastic assignment

<u>Probit assignment:</u>

10 iterations:



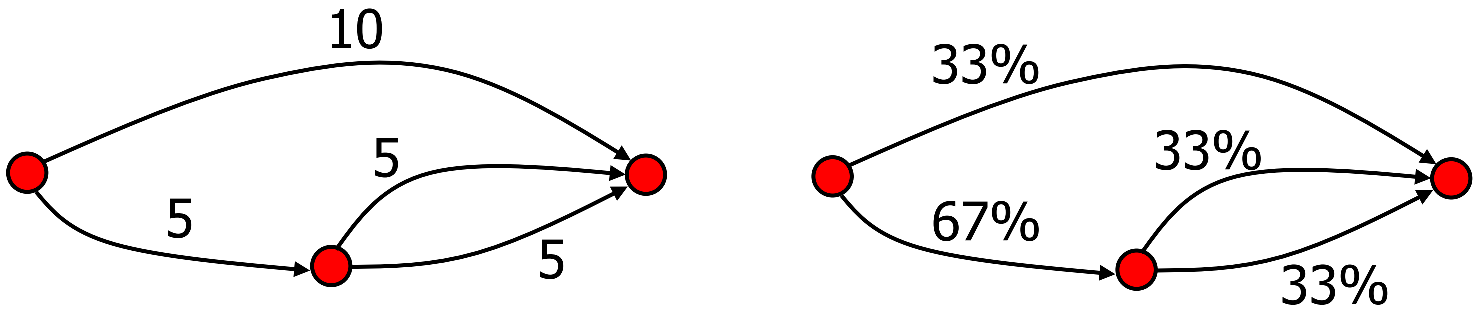
	route 1	route 2	route 3
	9.3600	9.5860	10.4766
	10.9212	7.9679	9.1340
	9.8664	8.5351	9.0683
	9.5078	11.0549	11.2326
	9.1196	10.3319	10.0110
	9.8607	10.5413	9.2257
	9.6406	10.8822	11.0726
	10.4757	9.9555	8.0073
	11.4155	9.5061	9.1937
	10.3215	9.8450	10.7971
	40%	30%	30%

Stochastic assignment at route level

Assume that **z** is Extreme value distributed:

$$\phi_{ijr} = \frac{\exp(-\beta t_{ijr})}{\sum_{p} \exp(-\beta t_{ijp})} \qquad \text{(Logit)}$$

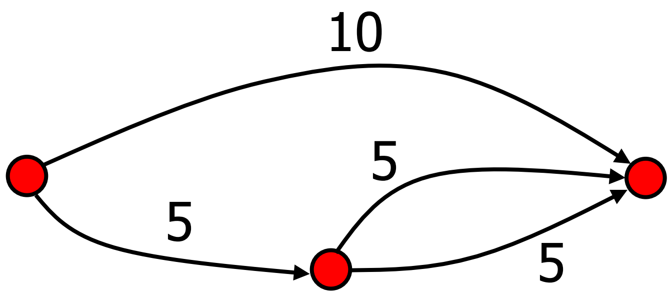
But ... logit assignment cannot deal properly with overlapping routes!

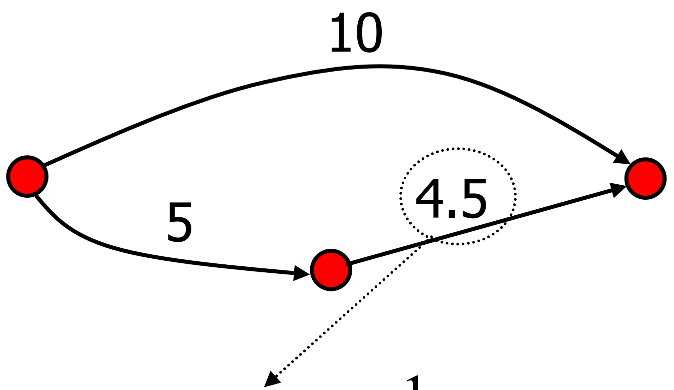


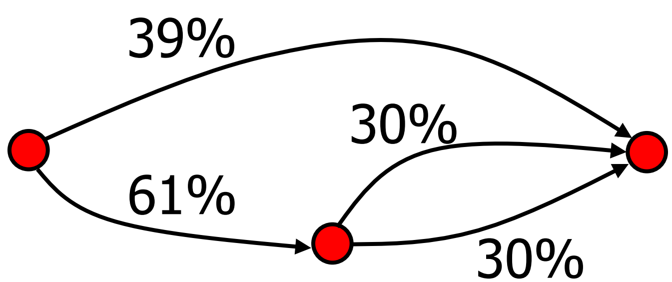
All 3 route alternatives: 10 min.

Stochastic assignment at route level

If there are simple overlapping routes, we can use the *nested* logit model







$$\bar{t} = -\frac{1}{\beta_2} \ln \sum_a (\exp(-\beta_2 t_a))$$

$$(\beta_2 = 1.5)$$

Stochastic assignment: comments

- In which cases would it be a realistic assignment?
    - Networks without congestion
        - Bicycle, 24-hour period (?)

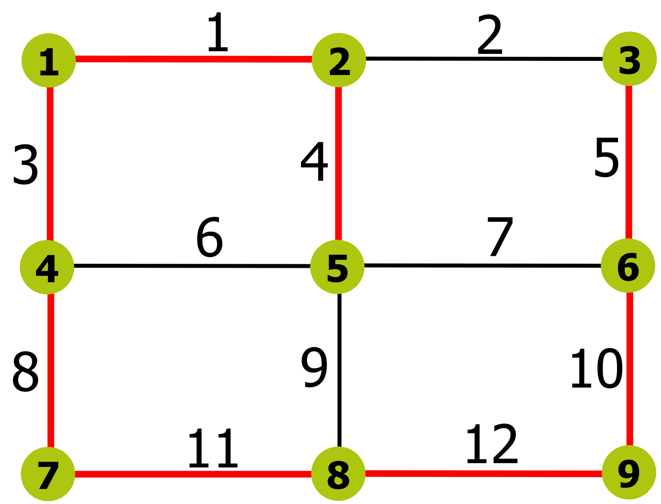
- How many iterations?
    - Old models: maximum of e.g. 6 (Burrell)
    - Theoretically: iterate until convergence

- Normal distribution?
    - Only positive travel times => e.g. Gamma distribution

5.

Assignment map and Path based approach

Alternative representation of paths



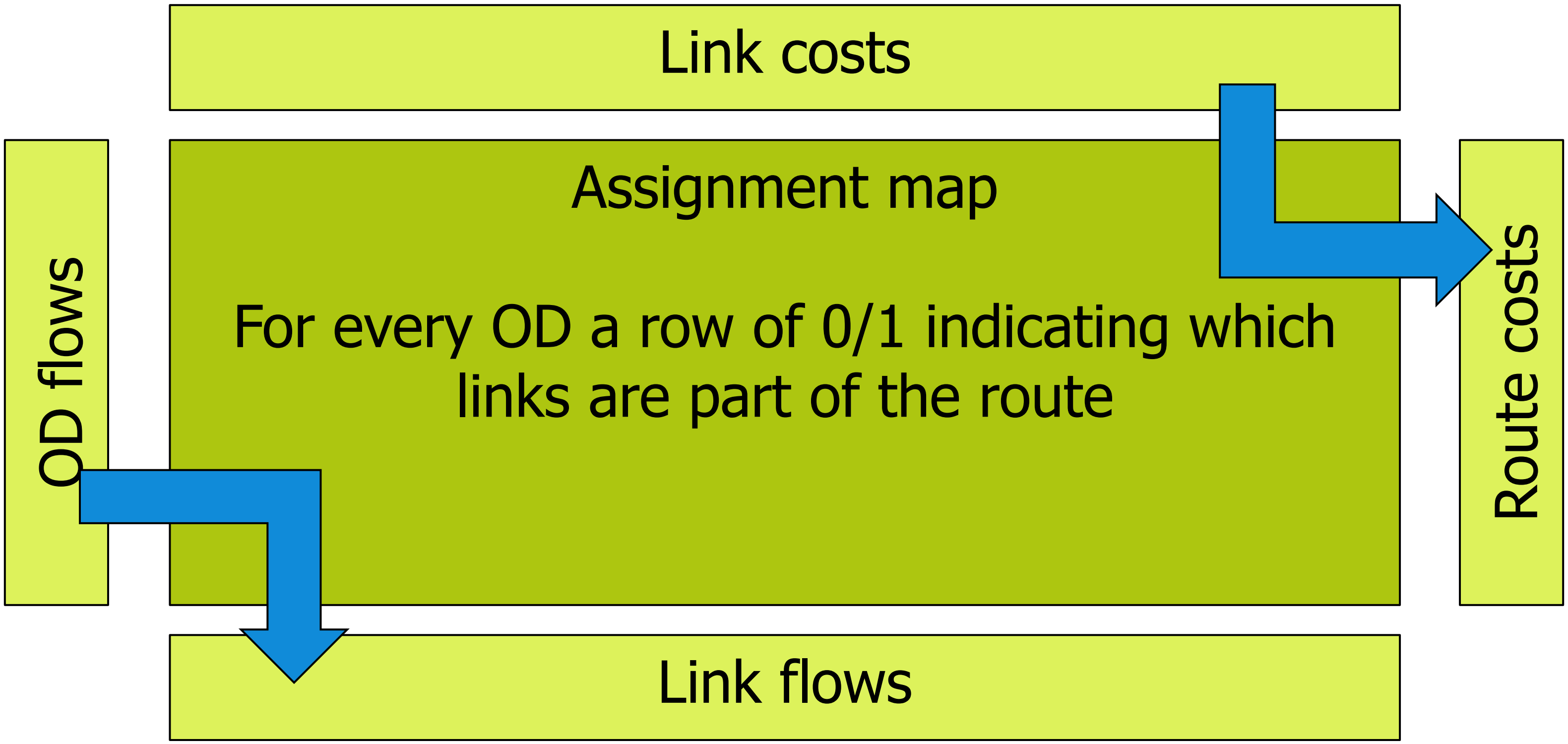
	route 7→1	route 7→2	route 7→3	route 7→4	route 7→5	route 7→6	route 7→8	route 7→9	
link 1	0	1	0	0	1	0	0	0	3
link 2	0	0	0	0	0	0	0	0	0
link 3	1	1	0	0	1	0	0	0	2
link 4	0	0	0	0	1	0	0	0	4
link 5	0	0	1	0	0	0	0	0	0
link 6	0	0	0	0	0	0	0	0	0
link 7	0	0	0	0	0	0	0	0	0
link 8	1	1	0	1	1	0	0	0	1
link 9	0	0	0	0	0	0	0	0	0
link 10	0	0	1	0	0	1	0	0	0
link 11	0	0	1	0	0	1	1	1	0
link 12	0	0	1	0	0	1	0	1	0

Less compact, very practical:
Assignment map

Number for elements for storing all shortest routes in the network:

$$A \times N \times (N-1) = 12 \cdot 9 \cdot 8 = 864$$

How to use an assignment map



All calculations can be performed using matrix algebra

Example

			1	2	3	4	5	6	7	8	9	10	11	12
			1	1	2	1	2	5	4	3	6	1	2	1
..	..		..											
7	1	5	0	0	1	0	0	0	0	1	0	0	0	0
7	2	5	1	0	1	0	0	0	0	1	0	0	0	0
7	3	5	0	0	0	0	0	1	0	0	0	1	1	1
7	4	2	0	0	0	0	0	0	0	1	0	0	0	0
7	5	2	1	0	1	1	0	0	0	1	0	0	0	0
7	6	2	0	0	0	0	0	0	0	0	0	1	1	1
7	8	3	0	0	0	0	0	0	0	0	0	0	1	0
7	9	3	0	0	0	0	0	0	0	0	0	0	1	1
..	..		..											

X

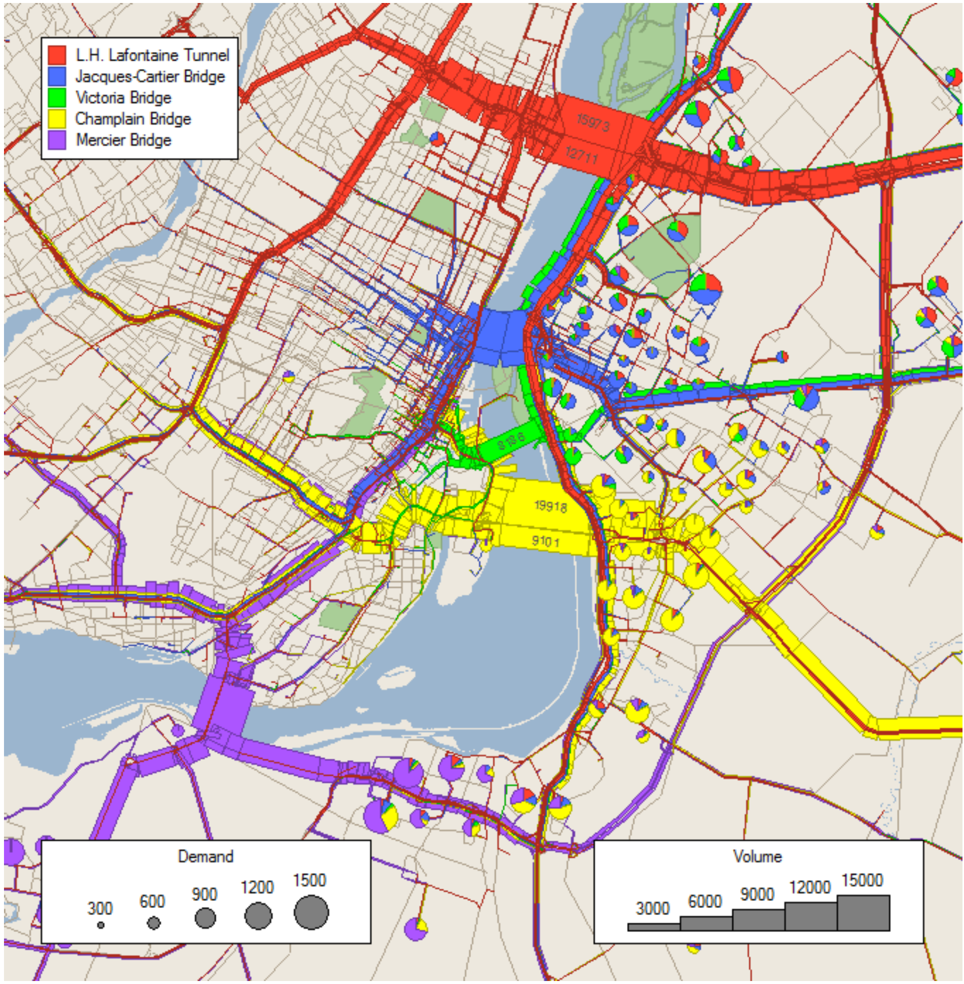
5+1+2+1 =9

X

5+5+2+2=14

Selected link analysis

- Assignment map tells which OD-pairs use a specific link
- So you can create an OD-matrix of OD-pairs using that link and assign that to the network
- Note that using a back node representation requires extra administration in the calculations to perform a selected link analysis



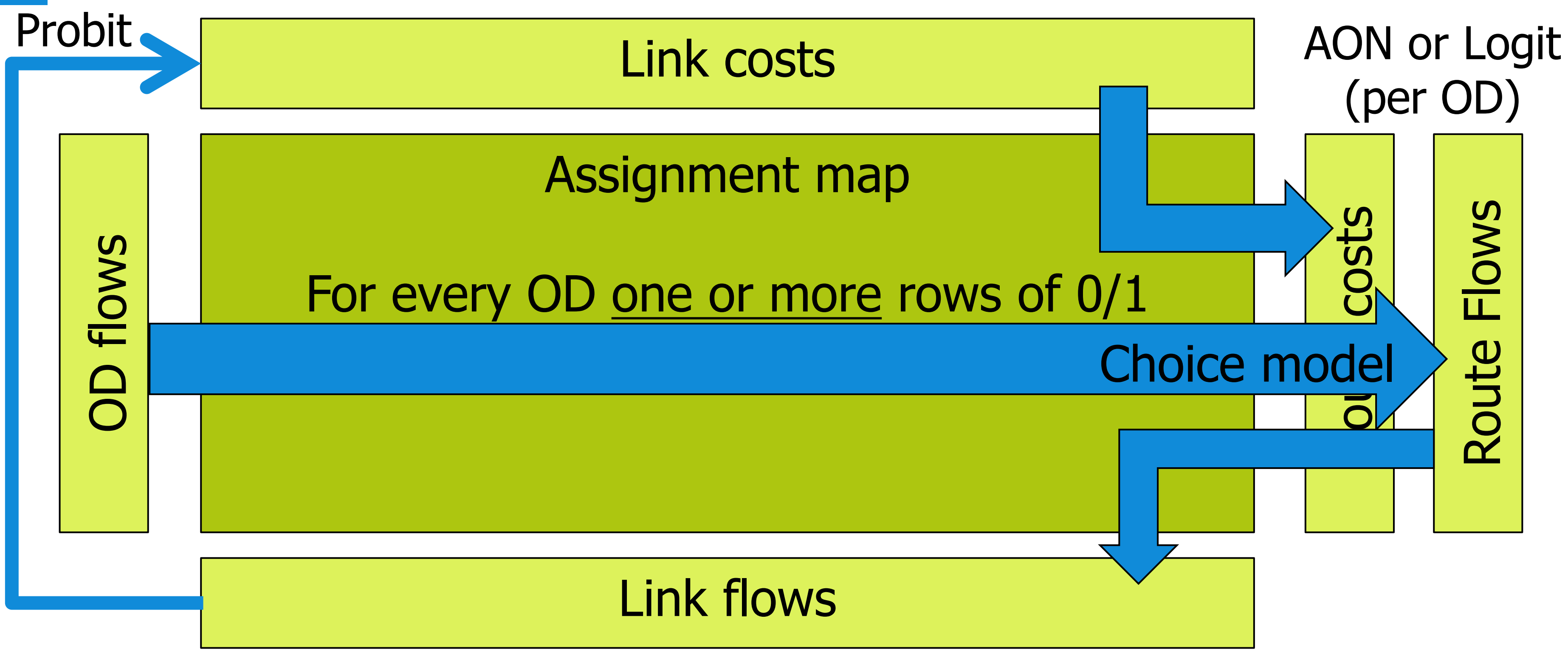
Stochastic assignment and path representation

- Iterative scheme as in Probit is tailor-made for tree search algorithms (or vice versa)

- Assignment map approach can be used as well, especially for Logit
  - Generate routes first,
    e.g. manually or using repetitive shortest path searches while systematically eliminating or penalizing links (k-shortest path algorithm)
  - Check whether routes are realistic (e.g. large or very short detours)

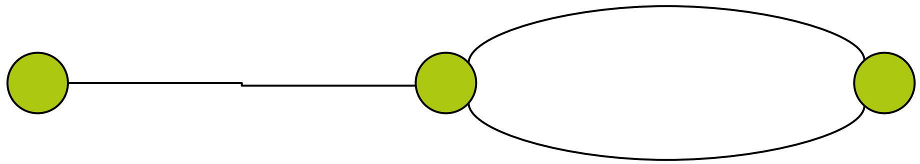
- Advantage of assignment map is that you can check for overlap

Assignment map and stochastic assignment



Switch from link level to route level

$t_{r1} = N(15,2)$

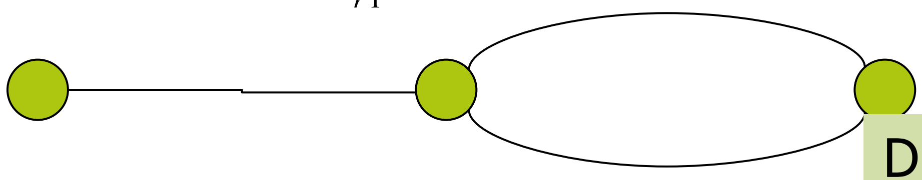


$t_{r2} = N(15,2)$

Thus sampling route travel times instead of link travel times

$t_{r1} = 15.85$

State 1

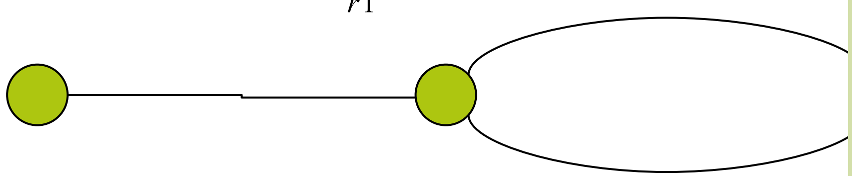


Route 1=15.83

$t_{r2} = 14.24$

$t_{r1} = 16.78$

State 2



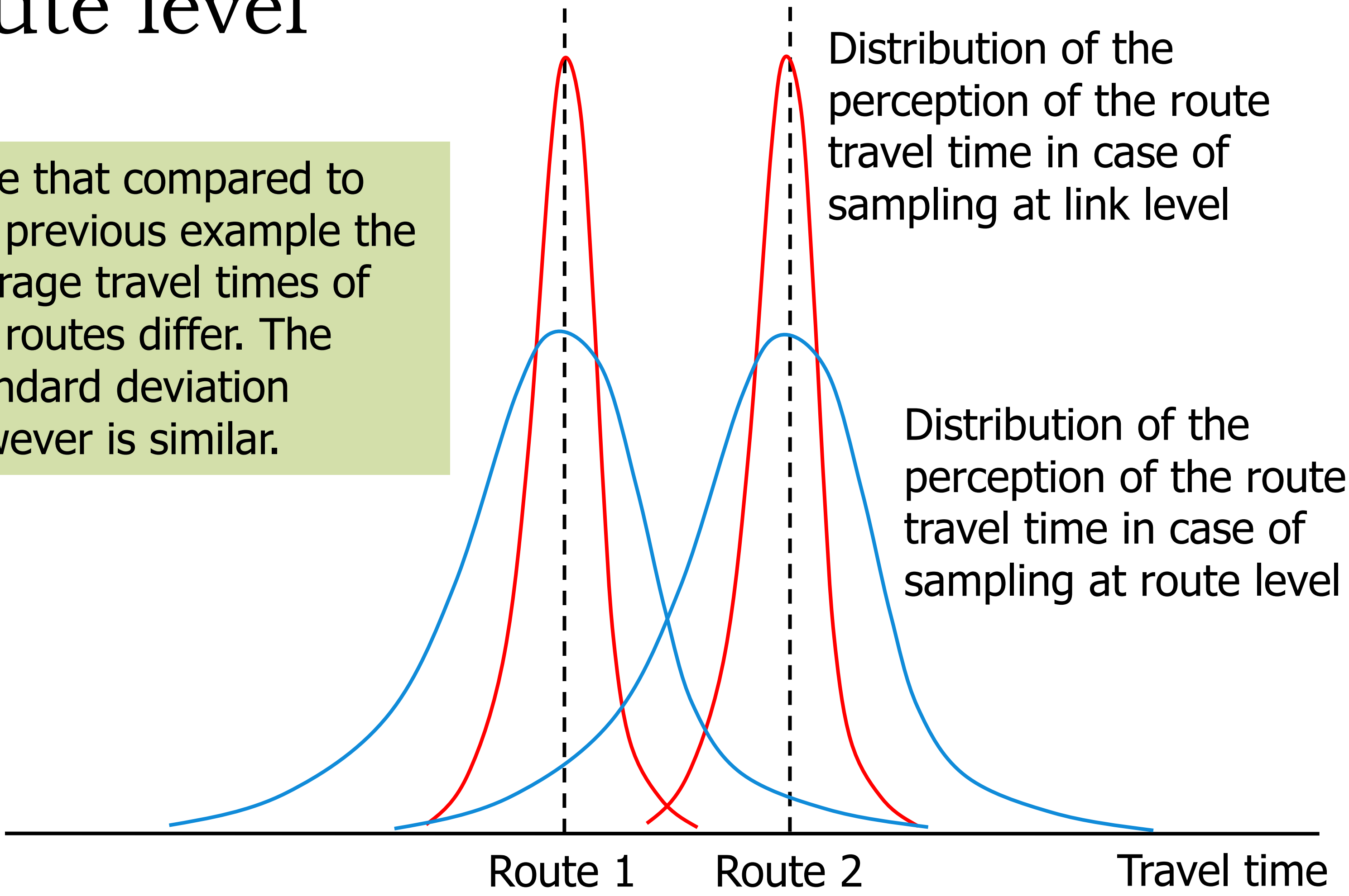
$t_{r2} = 13.57$

Differences between routes is much larger due to (implicit) different assumption for time of link 1 within a given state

This approach is therefore not correct

Effect of sampling at link level versus route level

Note that compared to the previous example the average travel times of the routes differ. The Standard deviation however is similar.



Probit and logit at route level

- For the error term at route level both extreme value distribution (=>logit) or normal distribution (=> probit) can be assumed
- In both cases the same error is made:
  in case of overlapping routes consistency within a network state is not guaranteed
- However, logit is much easier to compute…..
- Today, there are advanced logit models that can correct for overlap
  - For simple cases Nested Logit might be used

6.

Trucks

What's the problem (and possible solutions)

- Simply put, there are also trucks on the road……

- Add an OD-matrix of trucks to the OD-matrix of cars
  - Note that a truck is not a car
  - Use a PCU-value to correct for the size and impact:
    PCU for a truck is 1.5 or 2
  - Assign the resulting OD-matrix of vehicles

- Assign OD-matrix of trucks separately (e.g. using different speeds) and add the flows
  - Use a PCU-value

Topics Congested assignment

- What does this modelling component do? What's its output and what's its input? How does it fit in the framework?
- The main concepts
  - Wardrop's equilibrium, speed-flow curves
- The modelling methods
  - Iterative scheme: Method successive averages (MSA)
  - Convergence criteria: Relative or Duality Gap
  - Mathematical programming approach
  - Frank-Wolfe algorithm versus MSA
  - Stochastic equilibrium assignment (SUE)
- Practical issues
- Are these models appropriate?