

CIE4801 Transportation and spatial modelling (Uncongested) Assignment

Rob van Nes, Transport & Planning
17/4/13

Content

- What do we want to know?
- Shortest path Reprise
- All or Nothing assignment
- Multiple routes
- Trucks

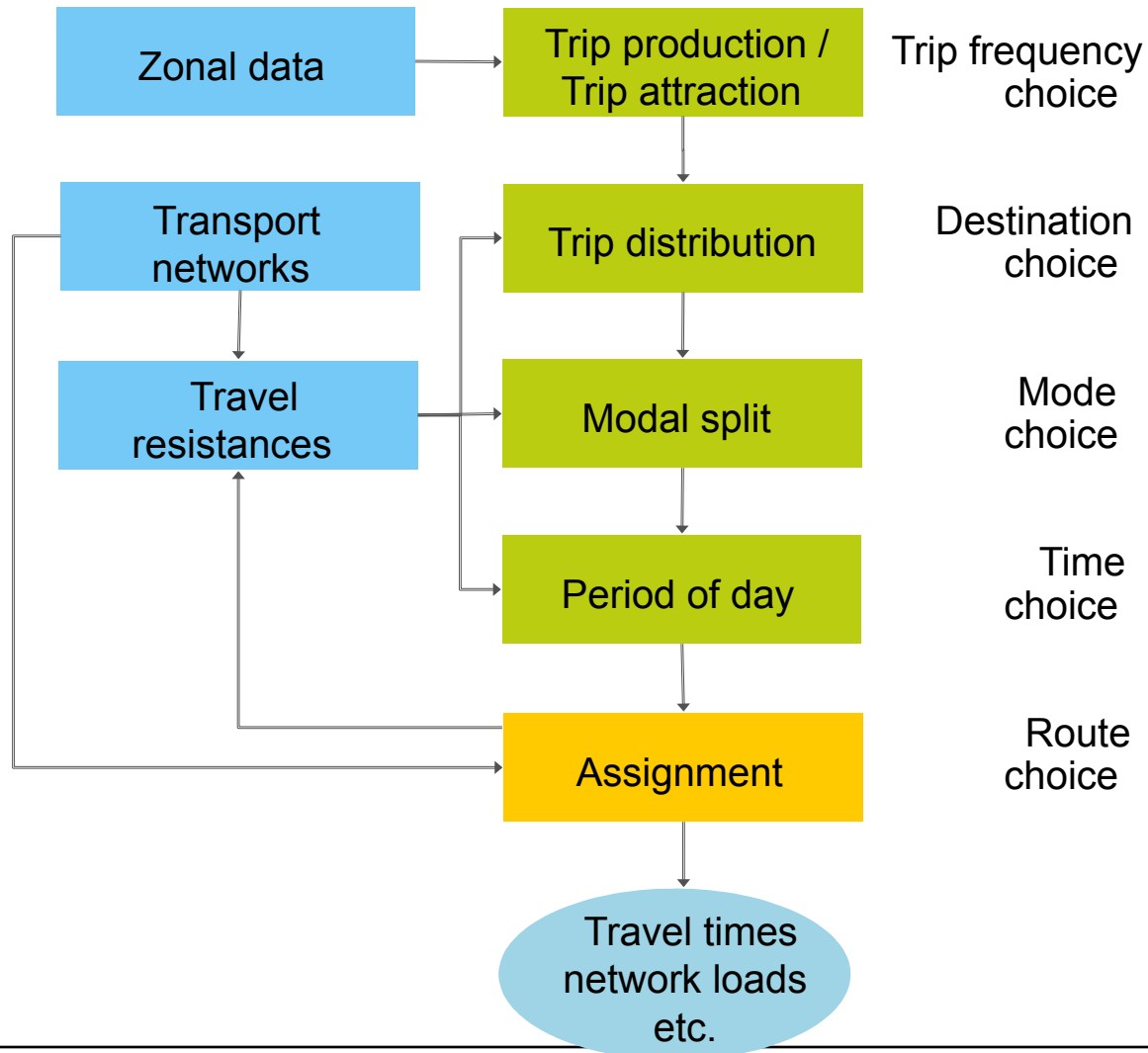
1.

What do we want to know

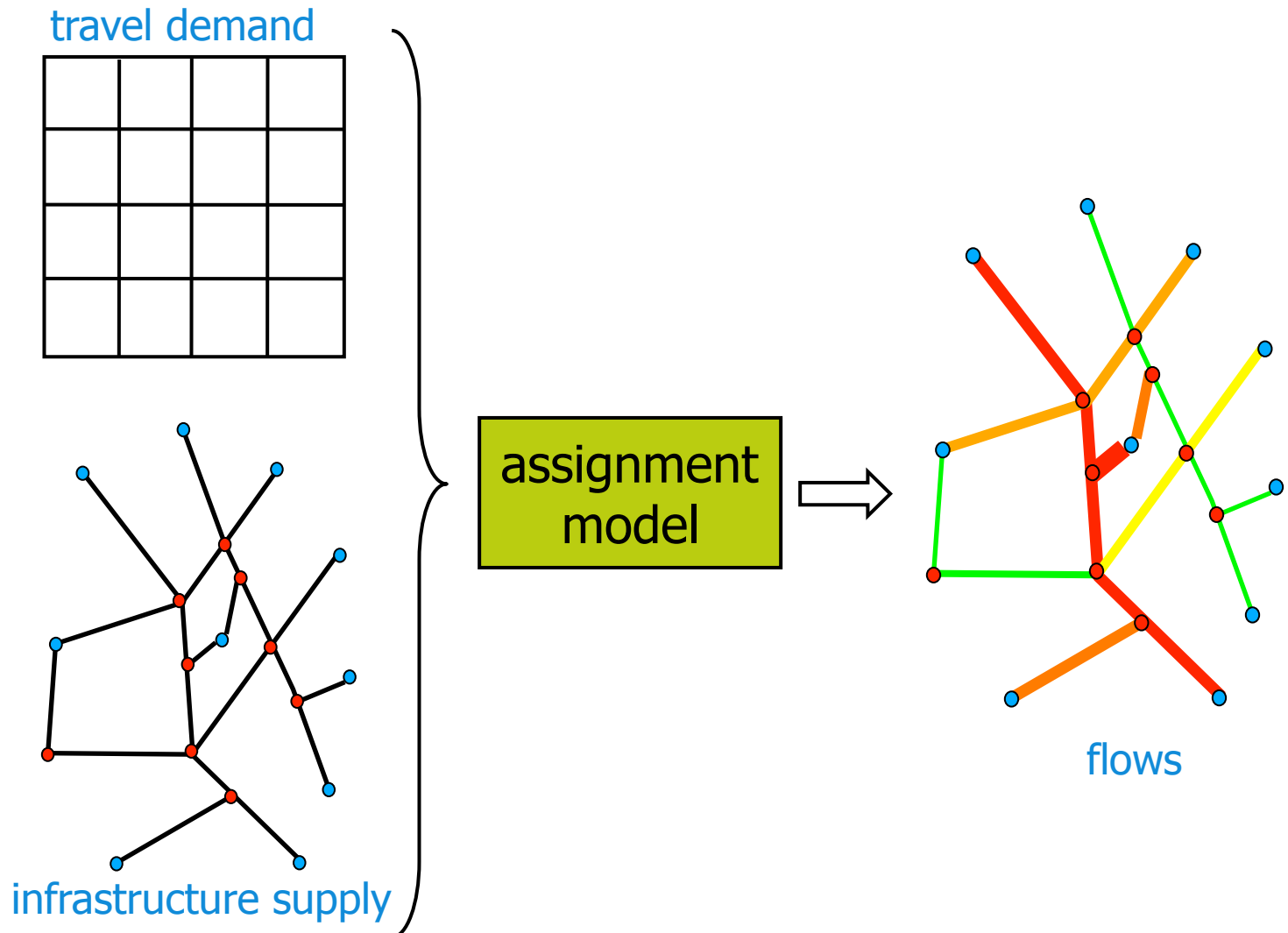
What do we want to know?

- Often the main goal of a modelling study
 - Flows on links or lines
 - Bottlenecks
 - Travel times

Introduction assignment



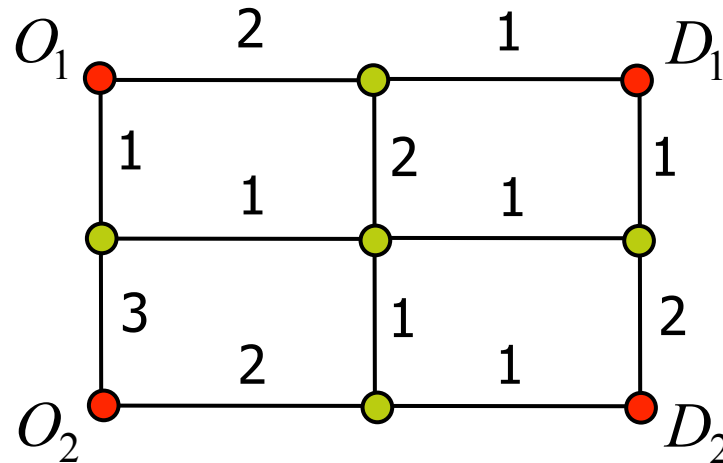
Introduction to assignment



Solving the assignment problem

	D_1	D_2
O_1	10	5
O_2	8	3

OD matrix



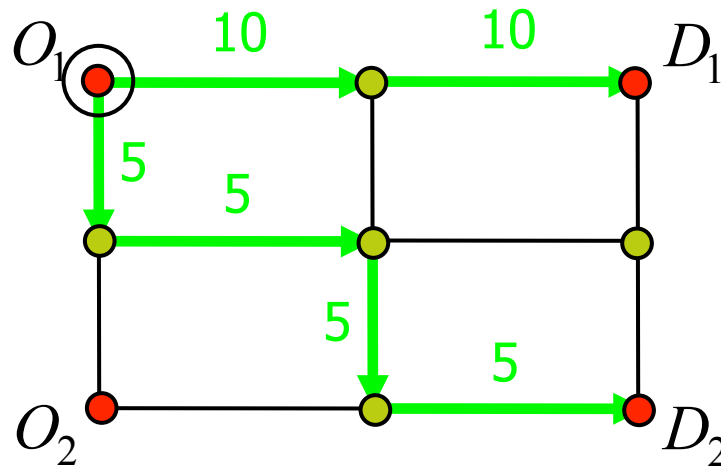
network

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



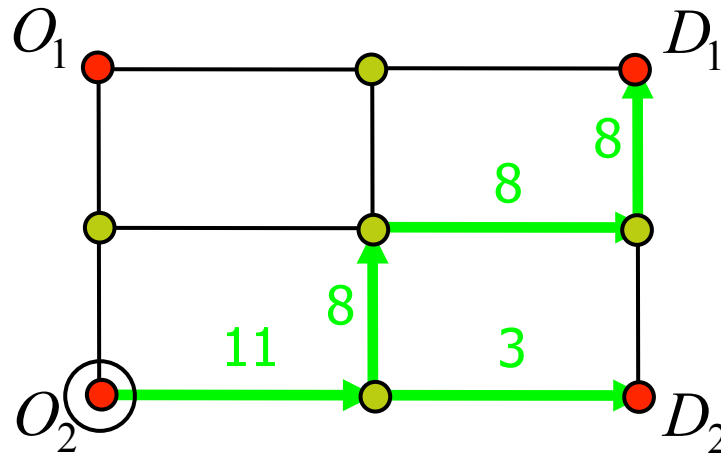
network

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



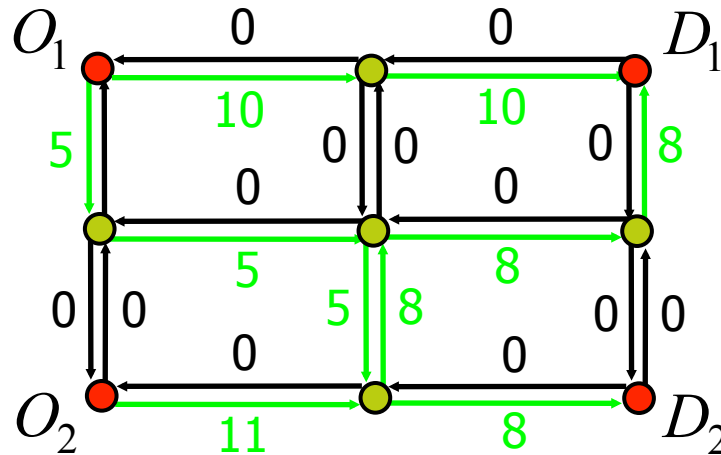
network

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.

2.

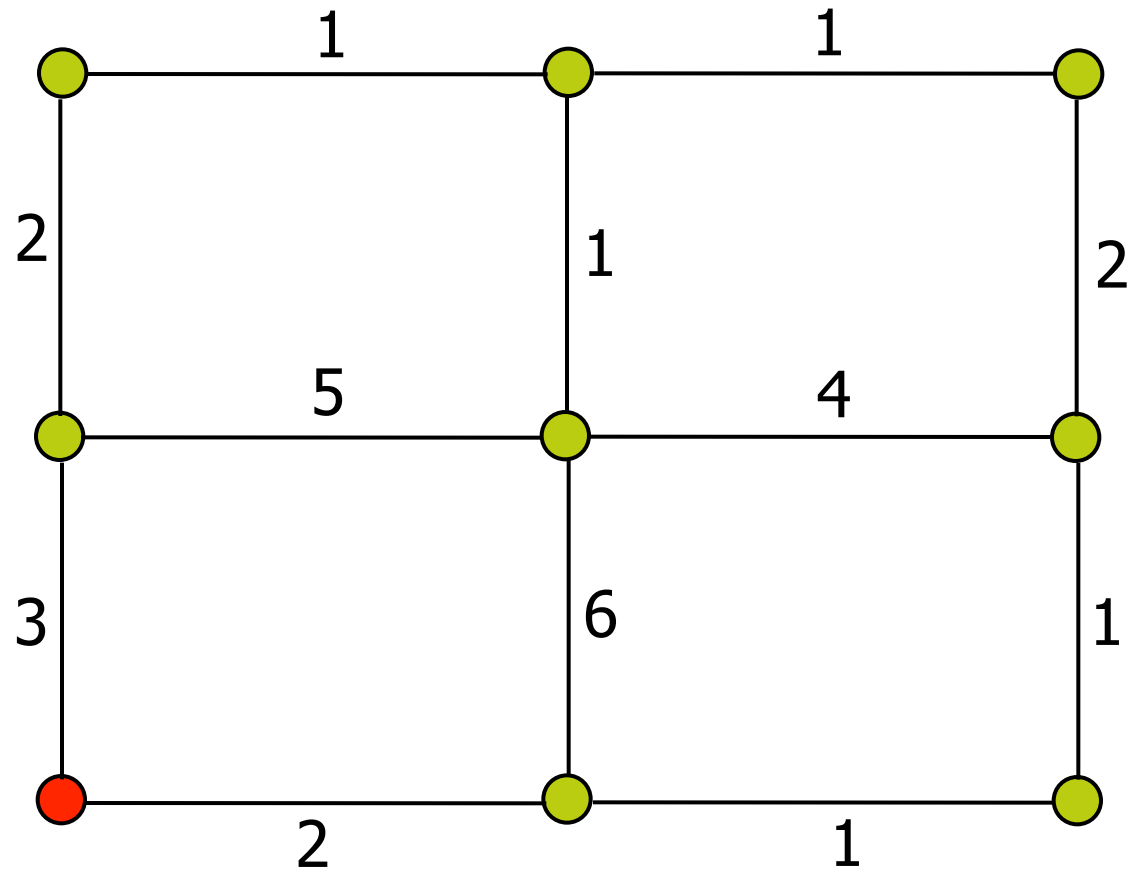
*Main building block: shortest path
Reprise*

Main concept for 'tree algorithms'

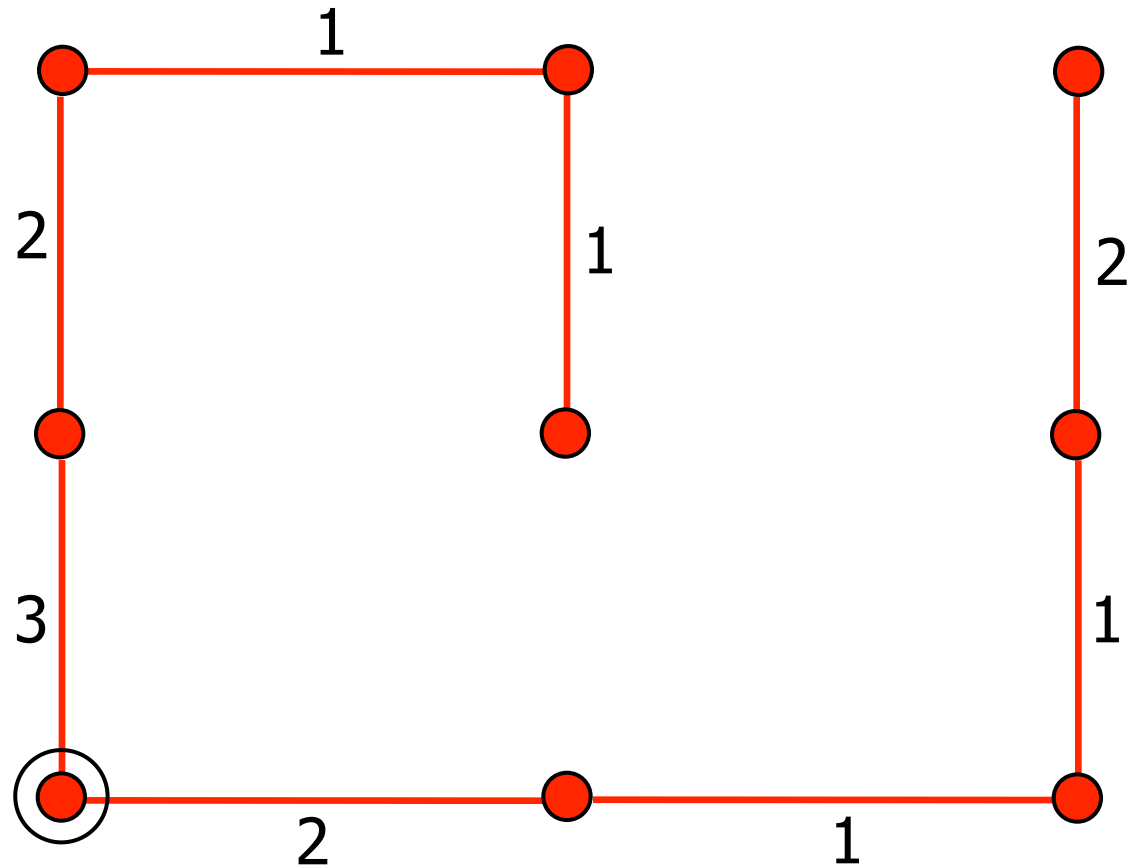
(Moore and Dijkstra)

- 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 $tt(a)+time(a,j) < tt(j)$
 $tt(j)=tt(a)+time(a,j)$, $bn(j)=a$ and node j becomes an active node
 - Node a is no longer active
- Select a new active node from the stack 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



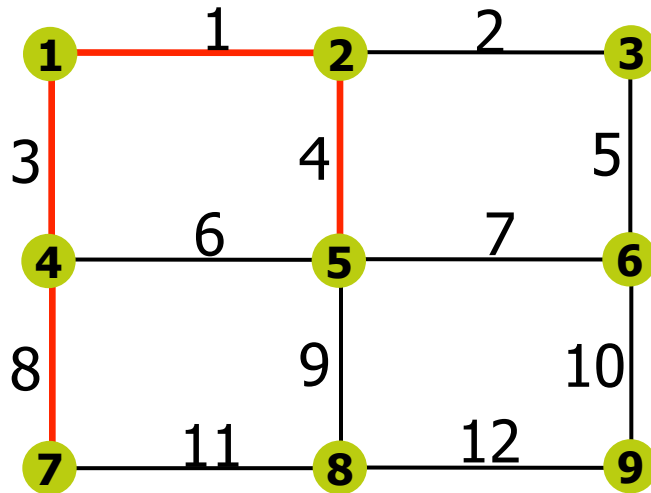
Shortest paths



Shortest path tree

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

Representation of shortest paths



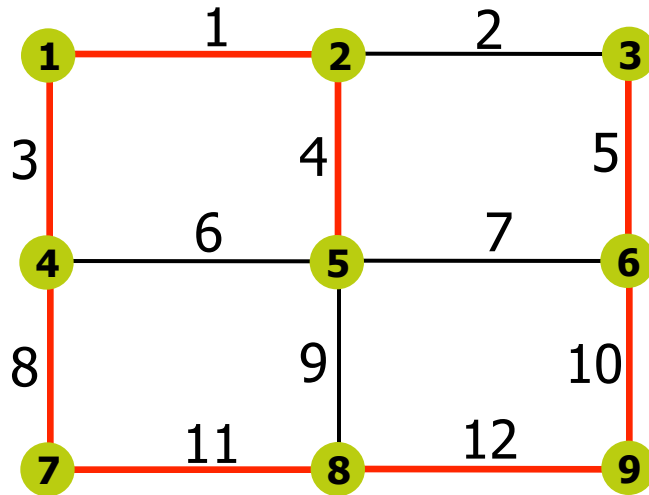
Shortest route from node 7 to node 5:

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:

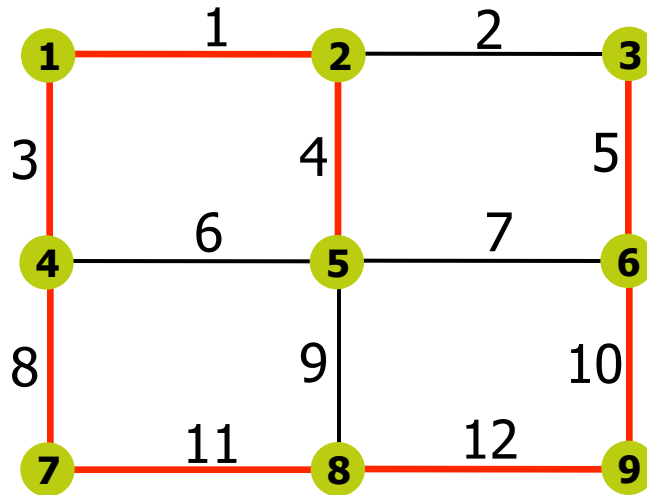
$$N \times N = 9 \cdot 9 = 81$$

17

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

Alternative representation



Less compact, very practical:
Assignment map

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

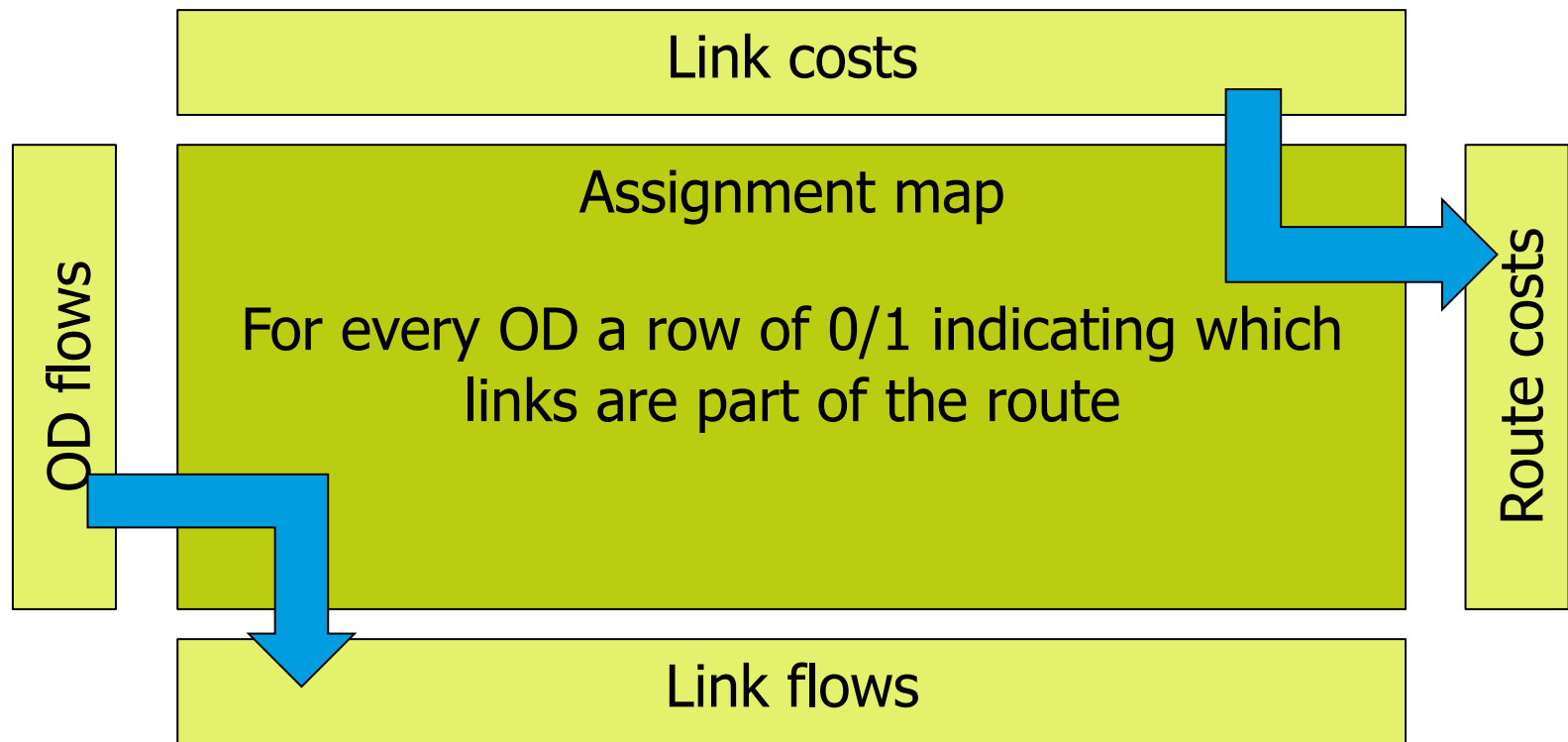
0	1	0	0	1	0	0	0	3
0	0	0	0	0	0	0	0	0
1	1	0	0	1	0	0	0	2
0	0	0	0	1	0	0	0	4
0	0	1	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
1	1	0	1	1	0	0	0	1
0	0	0	0	0	0	0	0	0
0	0	1	0	0	1	0	0	0
0	0	1	0	0	1	1	1	0
0	0	1	0	0	1	0	1	0

route 7 → 1
route 7 → 2
route 7 → 3
route 7 → 4
route 7 → 5
route 7 → 6
route 7 → 8
route 7 → 9

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

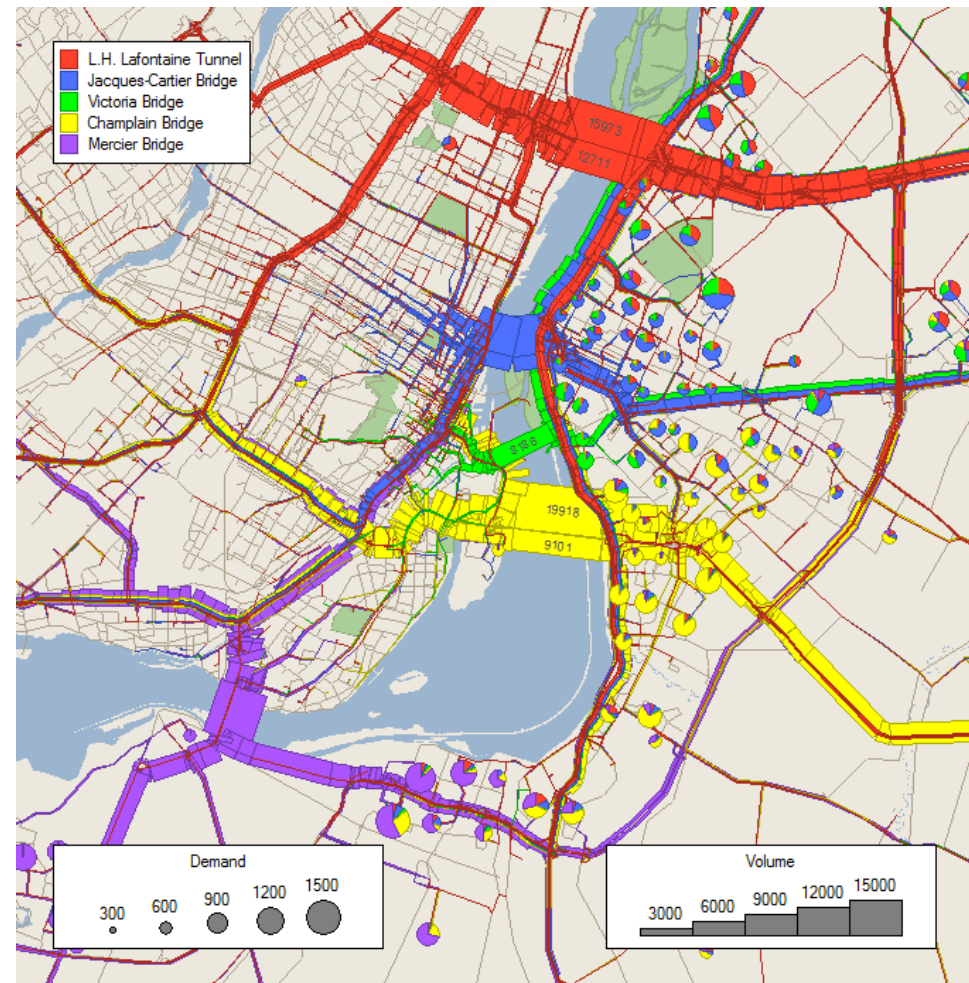
			1	2	3	4	5	6	7	8	9	10	11	12		
			1	1	2	1	2	5	4	3	6	1	2	1		
..	..		..													
7	1	5	0	0	1	0	0	0	0	1	0	0	0	0	X	
7	2	5	1	0	1	0	0	0	0	1	0	0	0	0		
7	3	5	0	0	0	0	0	1	0	0	0	1	1	1	$5+1+2+1=9$	
7	4	2	0	0	0	0	0	0	0	1	0	0	0	0		
7	5	2	1	0	1	1	0	0	0	1	0	0	0	0		
7	6	2	0	0	0	0	0	0	0	0	0	1	1	1		
7	8	3	0	0	0	0	0	0	0	0	0	0	1	0		
7	9	3	0	0	0	0	0	0	0	0	0	0	1	1		
..	..		..													

X

$5+5+2+2=14$

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



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

3.

All or Nothing assignment

All-or-nothing (AON) assignment

Reprise

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

$$T_{ijr} \geq 0$$

nonnegativities

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

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

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

$$q_a^s \geq 0$$

nonnegativities

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.

*Multiple routes: route choice
(stochastic assignment)*

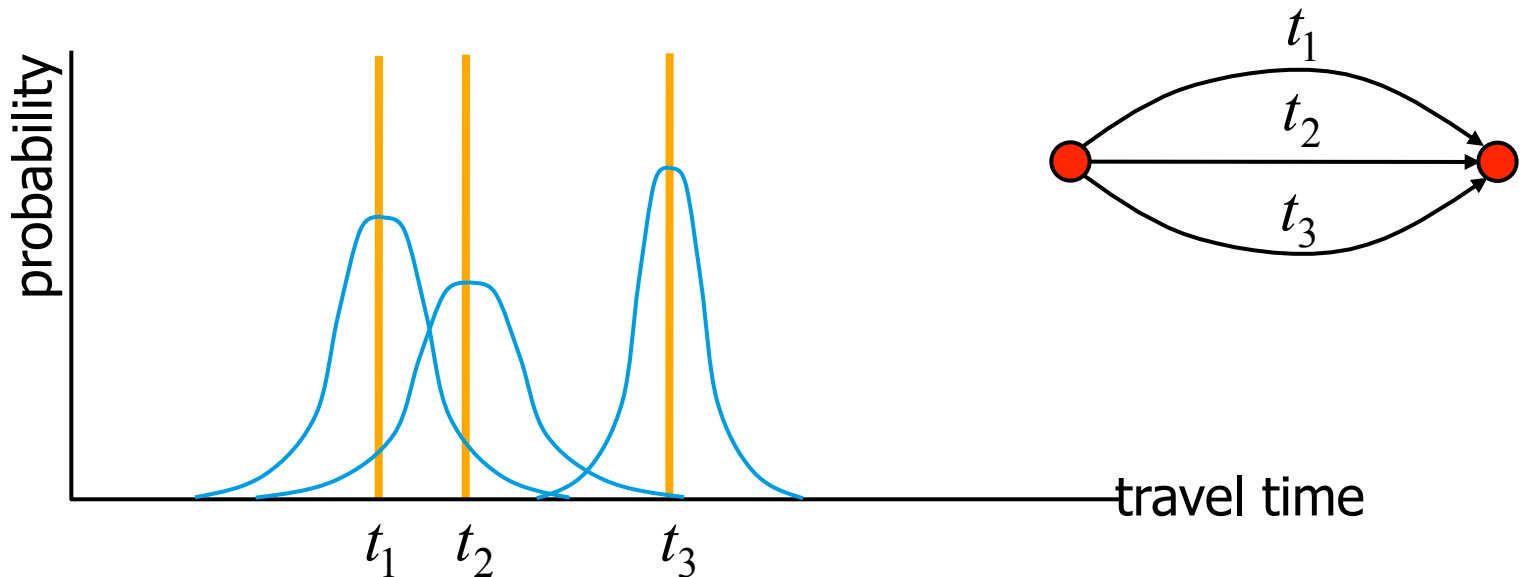
Stochastic assignment

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



Main concept

$$\mathbf{t}_{ijr} = t_{ijr} + \varepsilon_{ijr}$$

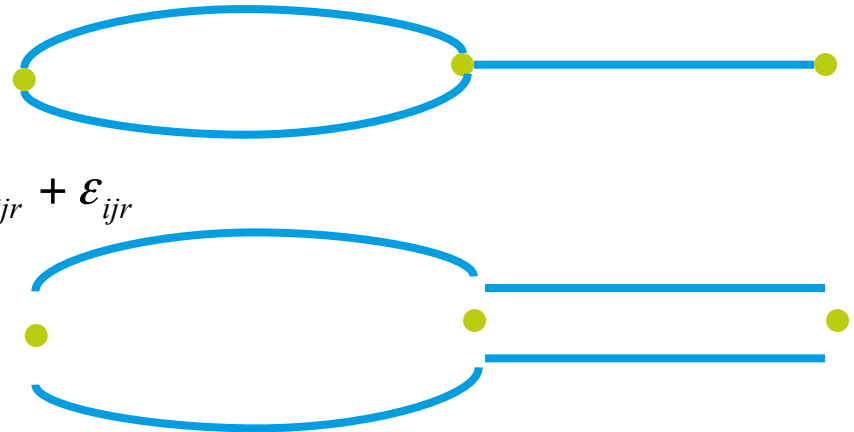
Usual assumption: Gumbel or Normal?

Gumbel leads to the Logit model

However is it about perception at route or link level?

$$t_{ijr} = \sum_{a \in r} t_a$$

$$\mathbf{t}_{ijr} = \sum_{a \in r} (t_a + \varepsilon_a) = \sum_{a \in r} t_a + \sum_{a \in r} \varepsilon_a = t_{ijr} + \varepsilon_{ijr}$$



Scaling or the error term

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

$$\mathbf{t}_{ijr} = t_{ijr} + \sqrt{(\Theta L_r)} \mathbf{z}$$

Θ = Dispersion parameter

L_r = Size of route r

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

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

Θ = Dispersion parameter

L_a = Size of link a

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

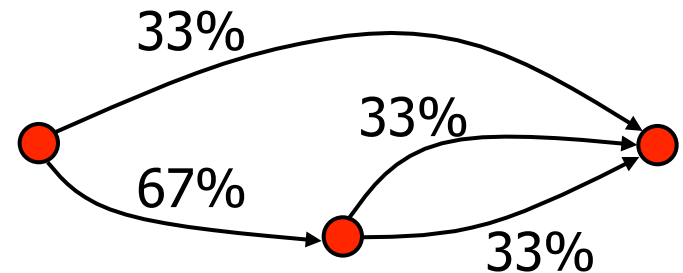
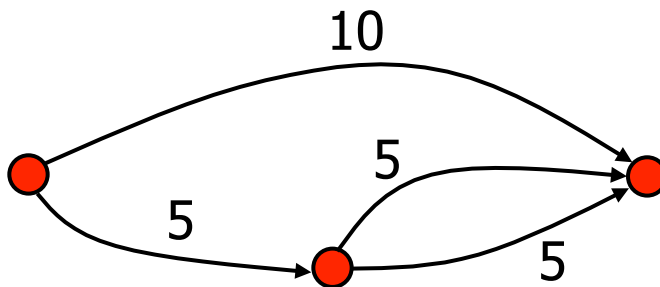
Size could be the length or free flow travel time of route or link

Stochastic assignment at route level

Assume that $\mathbf{z}$ is Gumbel distributed:

$$\beta_{ijr} = \frac{\exp(-\mu t_{ijr})}{\sum_p \exp(-\mu t_{ijp})} \quad (\text{Logit})$$

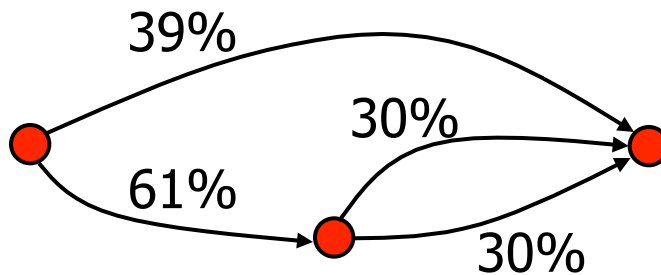
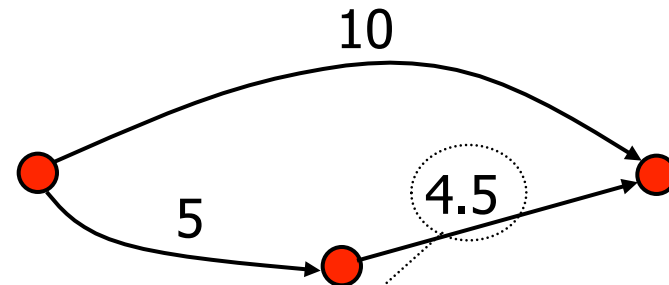
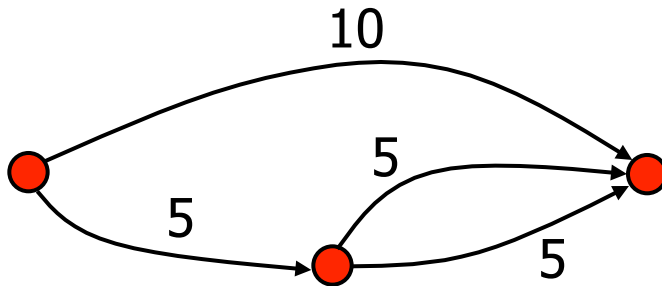
But ... logit assignment cannot deal properly with overlapping routes!



All 3 route alternatives: 10 min.

Stochastic assignment at route level

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



$$\bar{t} = -\frac{1}{\mu_2} \ln \sum_a (\exp(-\mu_2 t_a))$$

$(\mu_2 = 1.5)$

Stochastic assignment: route or link level?

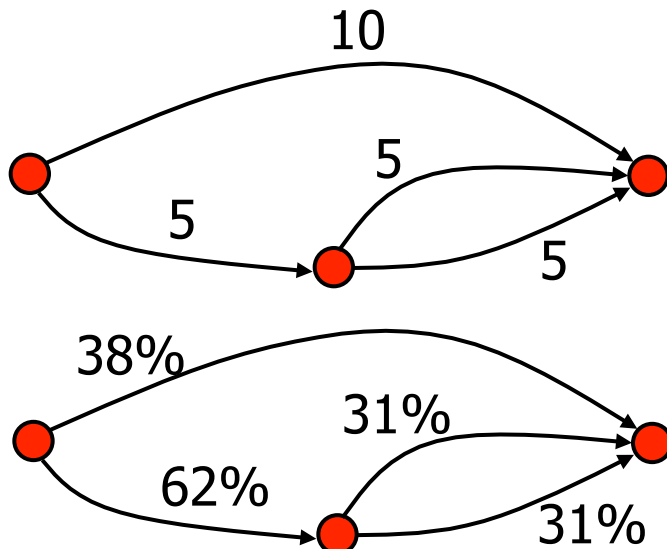
- Nested logit doesn't work in complex networks, e.g. Grid network in New York
- Alternatively we can switch from route to link level
 - Variation in perception at link level
 - While maintaining consistency at route level

Stochastic assignment at link level

Assume that $\mathbf{z}$ is Normally Distributed:

$$\beta_{ijr} = \dots \quad (\text{Probit})$$

No explicit analytical expression, but can be computed by simulation. Probit assignment has no problem with overlapping alternatives.



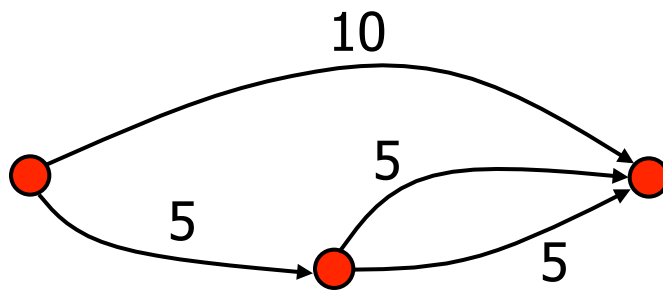
For $n=1$ to N do

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

Stochastic assignment

Probit assignment:

10 iterations:

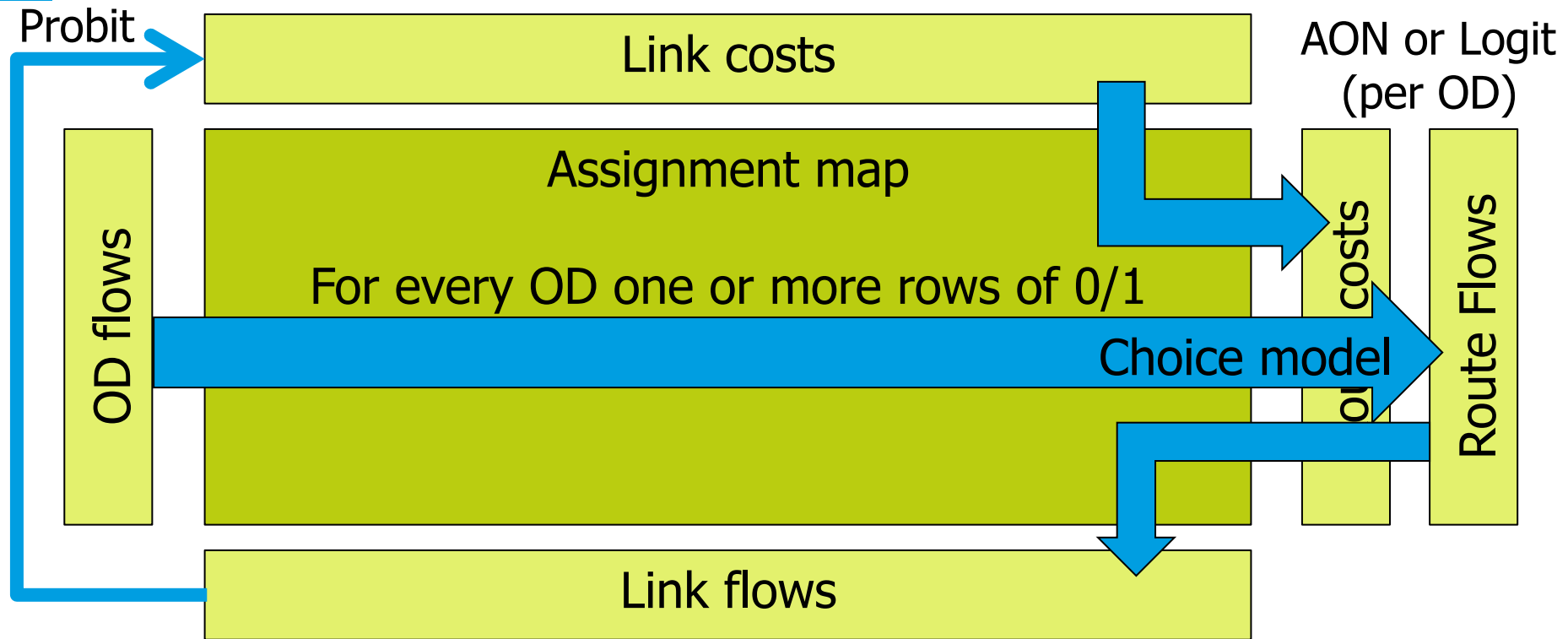


route 1	route 2	route 3
9.3600	9.5860	10.4766
10.9212	7.9679	9.1340
9.8664	8.5351	9.0683
9.5078	11.0549	11.2326
9.1196	10.3319	10.0110
9.8607	10.5413	9.2257
9.6406	10.8822	11.0726
10.4757	9.9555	8.0073
11.4155	9.5061	9.1937
10.3215	9.8450	10.7971
40%	30%	30%

Stochastic assignment and network representation

- Iterative scheme is tailor-made for tree search algorithms (or vice versa)
- Assignment map approach can be used as well
 - 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)

Assignment map and stochastic assignment



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.

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