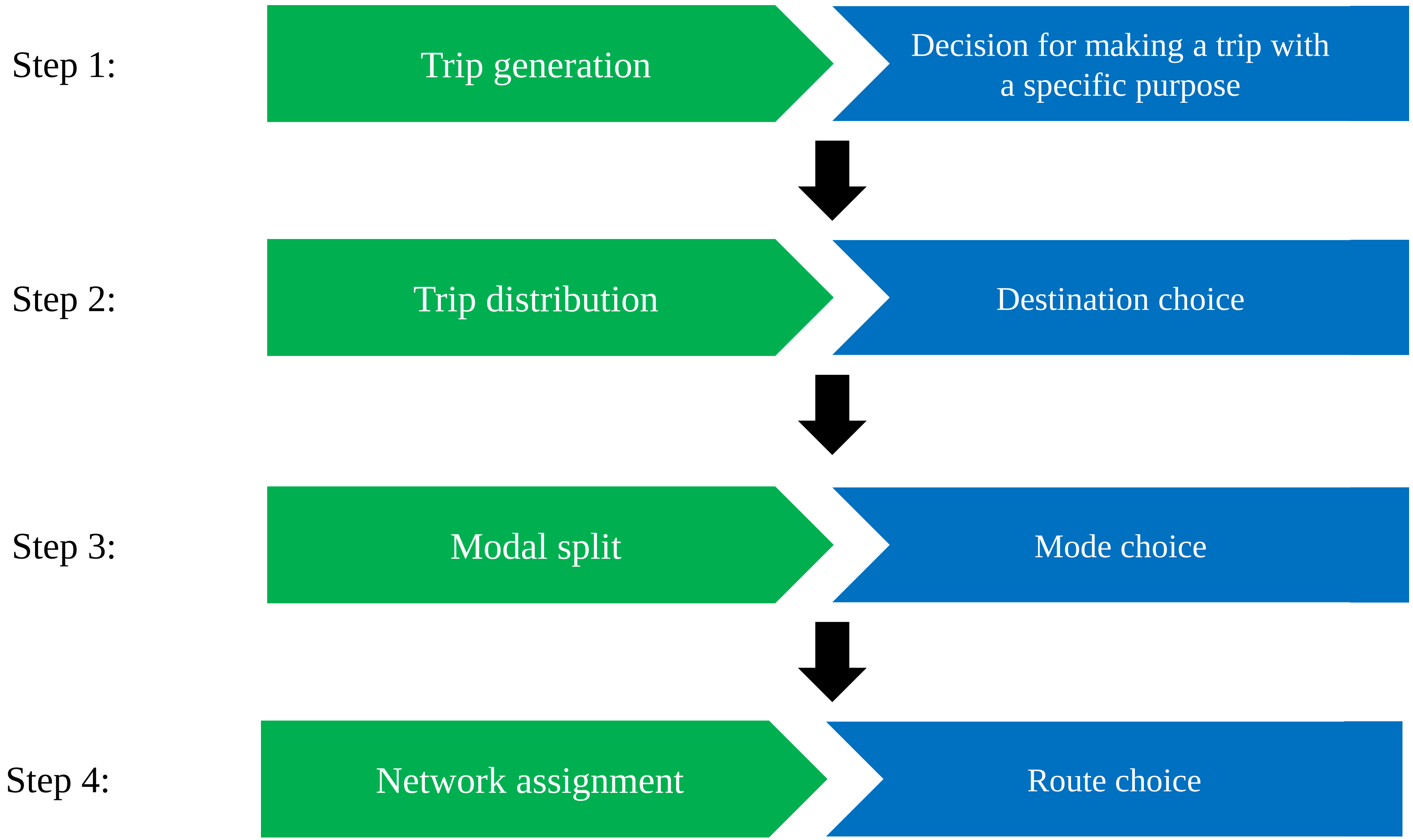
Decision-aid methodologies in transportation

CIVIL-557

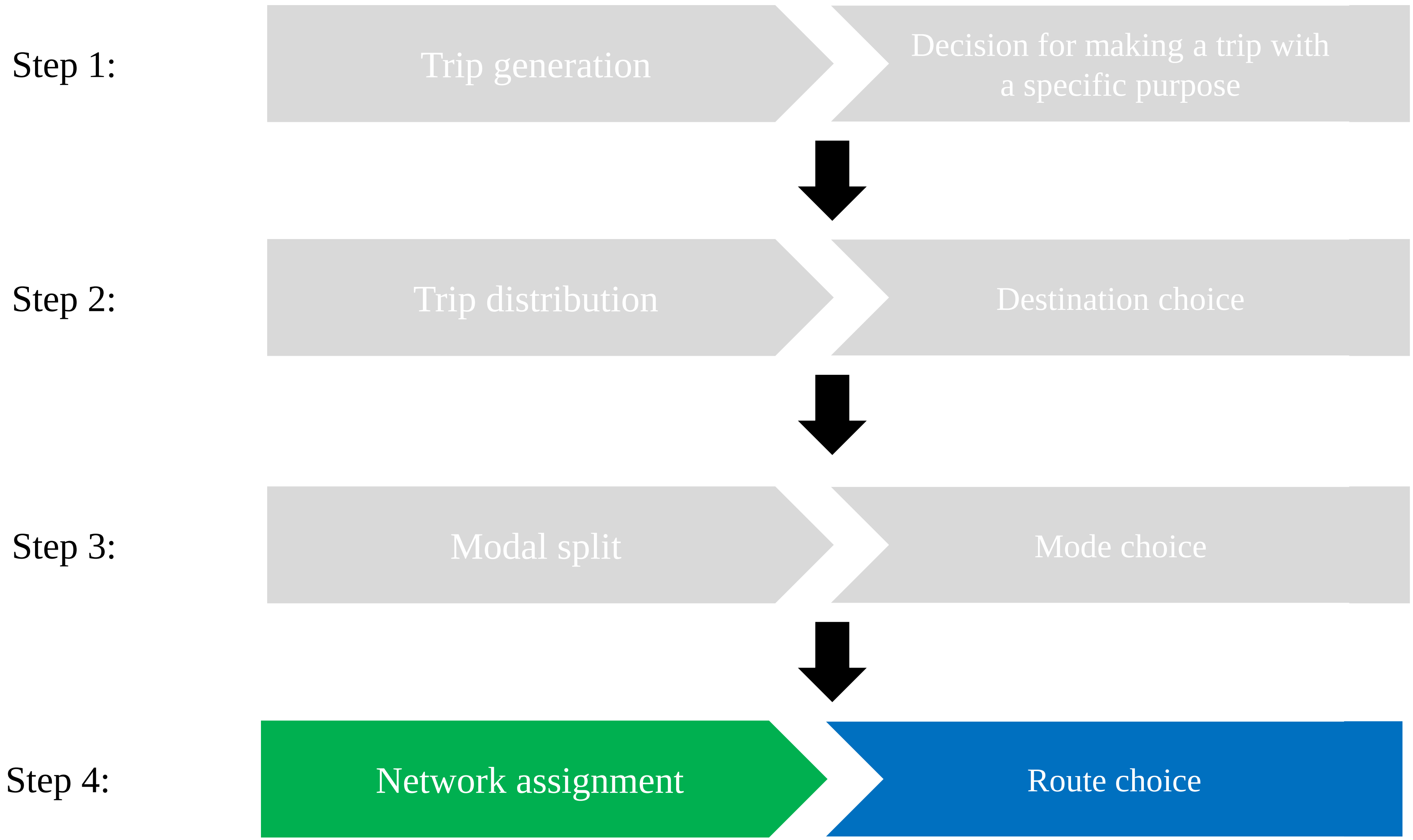
Modelling transportation systems
5. Traffic assignment

Evangelos Paschalidis

The 4-step model

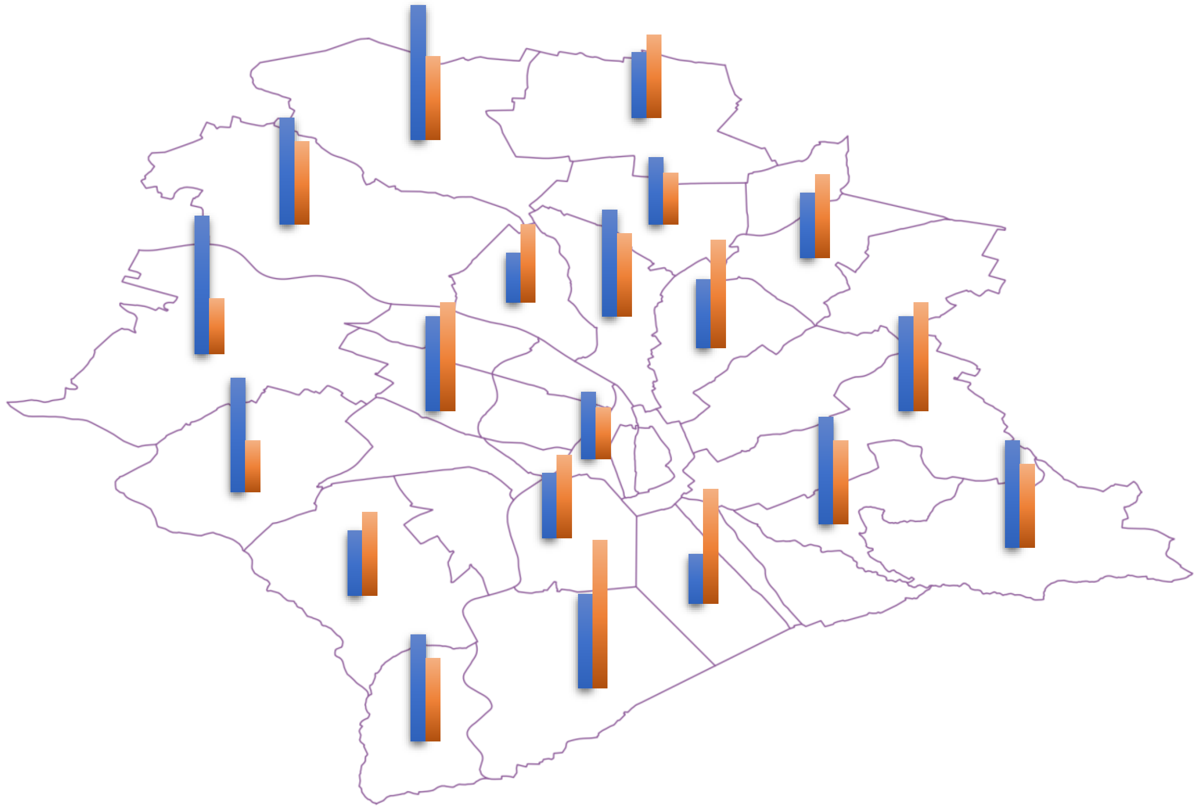


The 4-step model



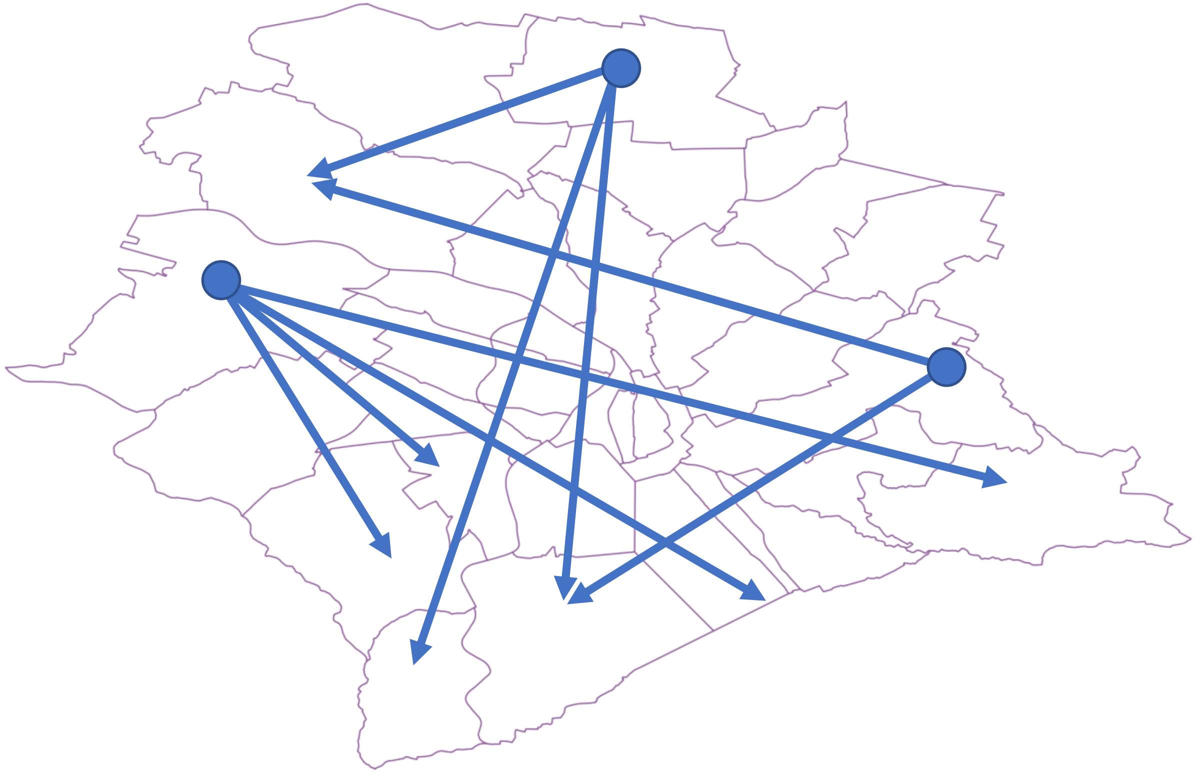
Step 1: Trip generation

- Trip generation step:
  - The number of trips generated in each zone
  - The number of trips attracted in each zone
- Generated trips are typically a function of socioeconomic characteristics and land use
- Attracted trips are typically a function of land use characteristics
- Output: The number of trips generated in and attracted to each zone



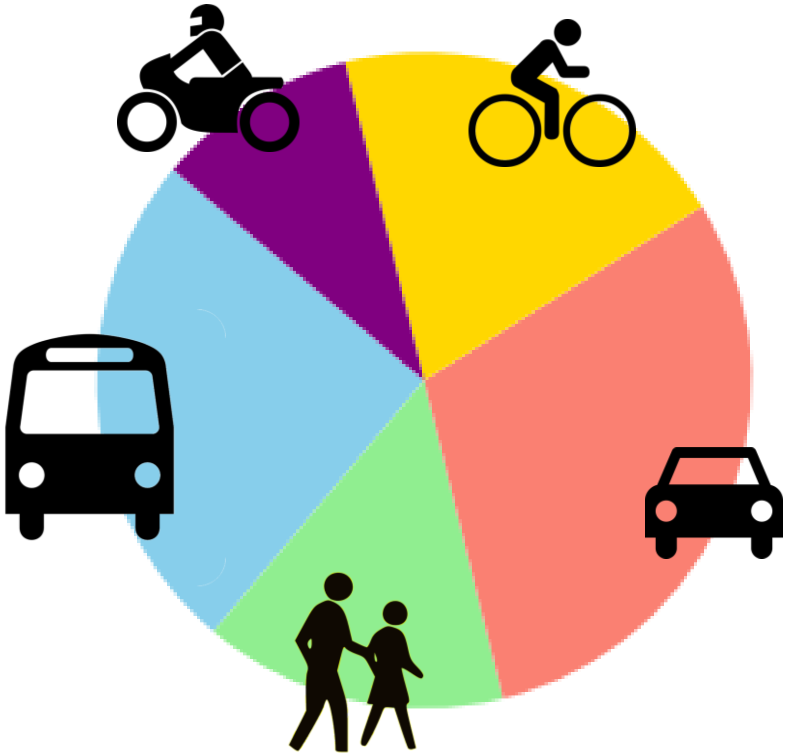
Step 2: Trip distribution

- Trip distribution step:
  - The number of trips between each origin-destination pair
  - The number of trips typically depends on the productivity of the origin zone and the attractiveness of the destination zone
  - Some typical factors that affect trip distribution are the size of a zone, the land use, and the trip cost between the origin and destination zones
  - Input: Trip production/attraction (from step 1), travel cost matrix
  - Output: Origin-destination matrix (typically by trip purpose)



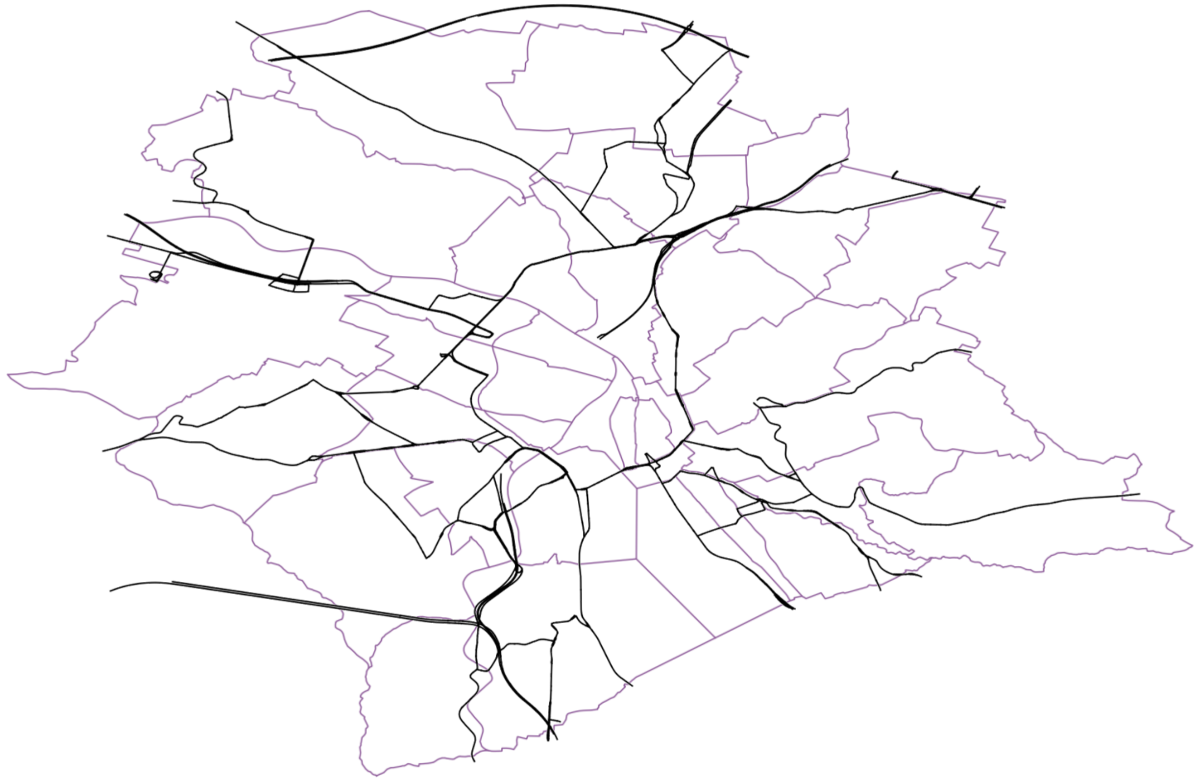
Step 3: Modal split (or modal share)

- Modal split step:
  - The number of trips per transport mode between each origin-destination pair
- Main factors are:
  - Mode attributes
  - Socioeconomic characteristics
  - Trip purpose
  - Availability of public transport
- Output: Proportion of each mode used by travellers
- Different OD matrices by mode



Step 4: Network assignment

- Data
  - Representation of the road network with links and nodes
  - Travel time functions per link
  - OD matrix (this is a merged matrix including the whole OD demand by trip purpose, household characteristics, travel mode etc.)



- Output
  - Traffic volume
  - Travel time per link

Traffic assignment

Assignment of vehicles to the road network

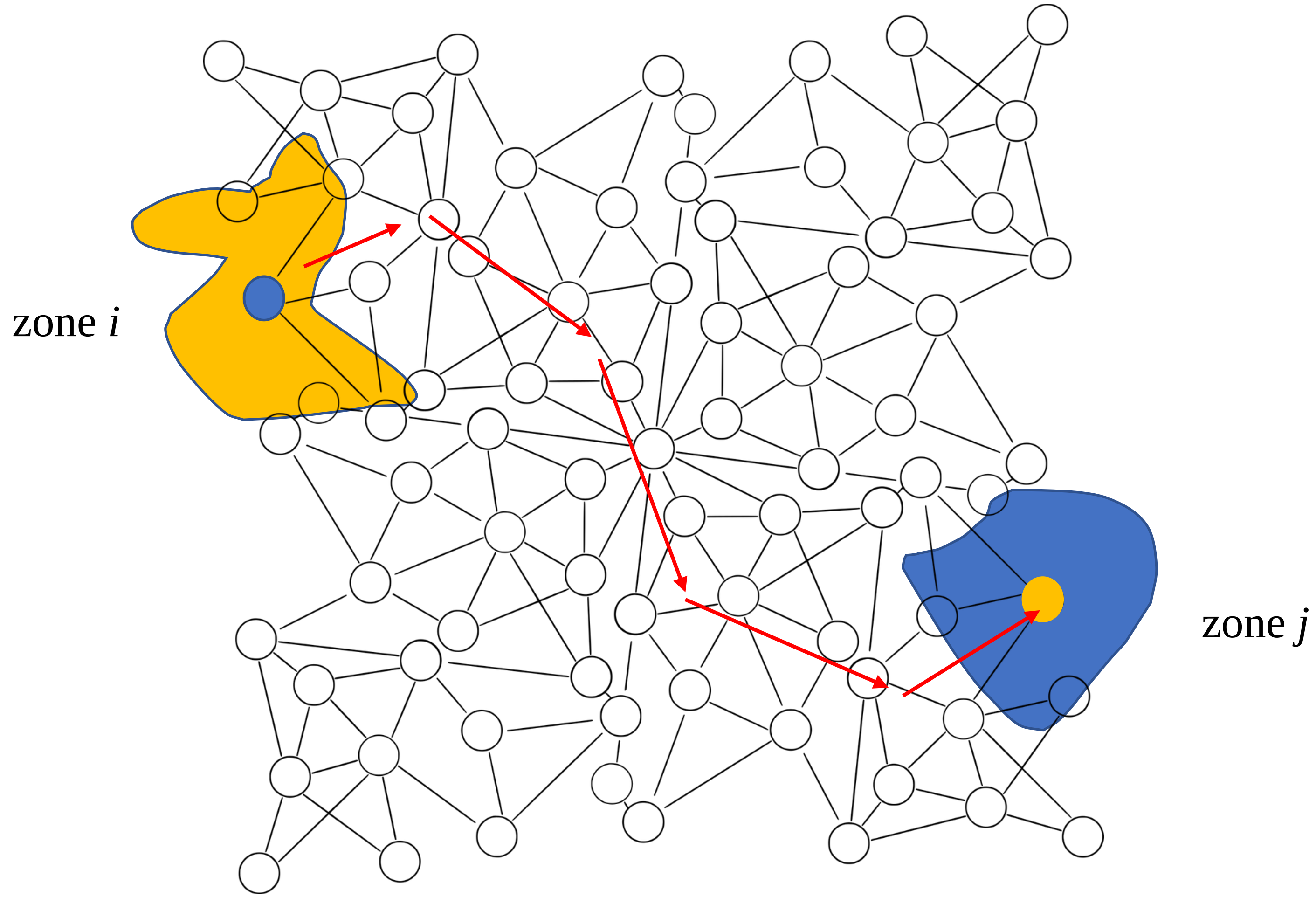
- Trip (OD) matrix of a given time period (e.g. morning peak)
- Trips are allocated simultaneously to the network so they all reach their destination within the time period under investigation
- Limitations
    - Trips may peak at different times in different parts of the network
    - In reality, queues may remain at the end of a particular time period

Network modelling

Moving from OD matrix representation to network (road) representation

- Assignment models (simpler output) ← We are doing this
- Detailed network – microsimulation & junction analysis

Traffic assignment



Given the segmentation by trip purpose, household type and travel mode:

- The route travellers will take from zone *i* to zone *j*
- What the cost (time) from *i* to *j* will be

The traffic network – supply

The traffic network is represented by:

- Links
- Nodes
- Costs – link cost function (a function of attributes associated with the links: distance, free–flow speed, capacity, the speed–flow relationship)

The traffic network – representation of traffic modes

- In a real network there are cars, taxis, motorcycles, bikes, e–scooters etc.
  - Not all modes represented – some are background traffic flows (case–specific)
  - E.g. in our example we will assume that only 80% – 90% of capacity is available for cars
- Network for public transport shared or separate depending on our project
- Bike network if available or exists in the software
- Modes per link: we explicitly define which modes are “allowed” to use a link

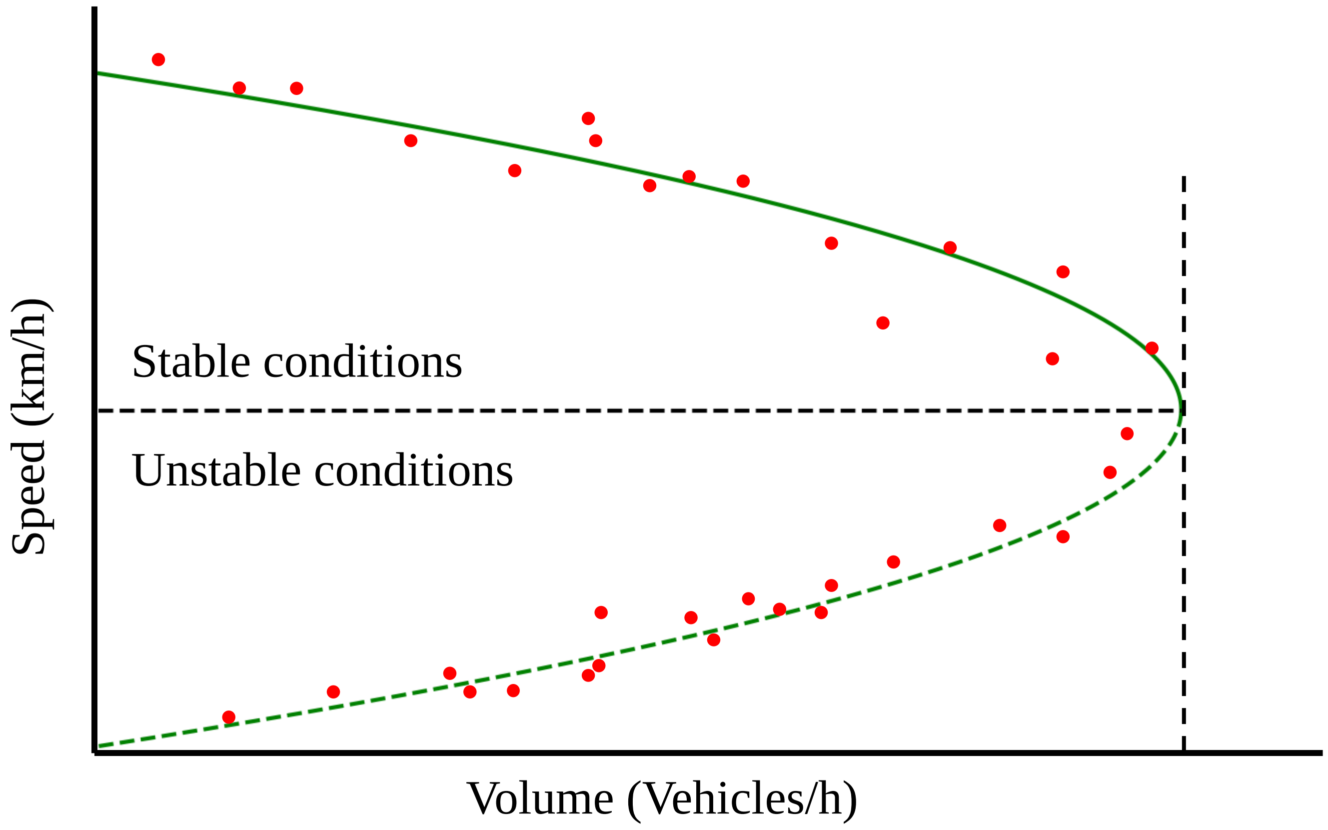
The traffic network – supply

Level–of–service (LOS): travel time, monetary costs (fares, fuel)

- If the offered level of service is lower than estimated, then a reduction in the demand and perhaps a shift to other destinations, modes and/or times of day is expected.
- The speed–flow (or generalised cost–flow) relationship is important as it relates the use of the network to the level of service it can offer.

Main attributes of flow

- Speed (S, km/h)
- Volume (V, Vehicles/h)
- Density (D, Vehicles /km)



Equilibrium

In economics:

- The actual exchanges of goods and services take place as a result of combining their demand with their supply
- The equilibrium point defines the goods' price and their quantities exchanged in the market.
- At equilibrium, the marginal cost of producing and selling the goods equals the marginal revenue

In transportation:

- Road network equilibrium: travellers from a fixed trip OD matrix seek routes to minimise their travel costs (times)
  - alternative routes, exploring new ones, and follow relatively stable pattern after trial and error
  - a pattern of path and link flows is in equilibrium when travellers can no longer find better routes to their destinations

Equilibrium

Travel Demand ⟷ Transport supply

- Demand on a road ↑
- Travel speed ↓
- Demand ↓
- Travel speed ↑

→ Equilibrium is reached when no travellers can find better routes

Equilibrium

In transportation:

- Multimode network equilibrium: when the flows of a mode affect the performance of other modes
  - E.g. congestion due to cars may increase the travel time of buses
  - Bus users may try to change their routes to avoid delays – additional capacity – new equilibrium points
- Network equilibrium: the resulting flows may affect mode choice, destination choice, and departure time
  - New flows – new equilibrium points – new levels of service
  - We may have to re–estimate the trip matrix

Equilibrium

Measure of disequilibrium

$$\delta = \frac{\sum_{ijr} T_{ijr}\left(C_{ijr} - C_{ijr}^{min}\right)}{\sum_{ij} T_{ij}\left(C_{ij}^{min}\right)}$$

- C_{ijr}: the travel cost from origin i to destination j via route r
- C_{ijr}^{min}: the cost from i to j via the least-cost route
- T_{ijr}: the number of trips from i to j via route r

If $\delta = 0$ then equilibrium conditions (no user can find a lower cost route to destination)

Model inputs

Travel demand:

- The OD matrix for a given period of time e.g. morning peak
- At this stage we have merge the various OD matrices segmented by trip purpose, mode, etc. into one OD matrix

Network supply:

- The traffic network e.g. links, nodes, zones
- The attributes of the links e.g. capacity, free–flow speed, cost function etc.

Selection of model:

- Stochasticity considered?
- Congestion effects considered?

Typical model outputs

- Summary statistics such as vehicle miles travelled on specific road types and total travel time
- Number of vehicles using each link
- Travel costs between zones based on a simulated demand level
- Link-level data including flows and associated travel times or costs
- Routes or specific links taken for each origin-destination (OD) pair
- Time of congestion per link
- Vehicle to capacity ratios

A note on the OD matrices (reminder)

- For urban areas where congestion is common, we use different OD matrices e.g. morning peak OD, afternoon peak OD, off-peak OD, weekend OD etc. Full day (24h) matrices are not particularly useful because they only provide a daily average illustration of the performance of the transportation system

- Maybe it is still fine to use 24h matrices if congestion is not a big issue in the study area.

- If we want to convert a 24h matrix to an 1-hour matrix, dividing the trips by 24 is not the solution
  - Full day matrices are fairly symmetrical (because people usually travel somewhere from home and back) while peak-hour matrices have a clear direction (e.g. from home to work)
  - We need to know what proportion of the daily trips takes place during the peak-hours and not simply assume a uniform distribution across the day (e.g. take information from travel diaries or historical trends)

- The OD matrices that have been estimated in steps 1–3 of the 4–step model reflect the movement of people not vehicles!
  - We should consider occupancy rates of vehicles e.g. one car moves 1.5 people or a bus moves 30 people etc. (numbers vary depending on the study area)

Link cost functions

Typically describe the travel time as a function of free–flow travel time, traffic volume, link capacity etc.

One of the most common (but not the only one!) link cost function is of Bureau of Public Roads – Federal Highway Administration (USA)

$$t(v) = t_0\left[1 + a\left(\frac{v}{c}\right)^{\beta}\right]$$

- $t(v)$ = travel time when volume is v
- t_0 = travel time in free flow conditions
- v = traffic volume (veh/h)
- c = capacity (veh/h)
- α, β = parameters (to be calibrated)

In standard practice we assume: $\alpha = 0.15$ και $\beta = 4$

Generating a path…

Typical sequence between and origin *i* and a destination *j* for cars:

origin zone centroid, zone connector, node, link, node, link, …, node, zone connector, centroid

In public transport we may use foot as a mode from connector to the first link of public transport (increase in travel time)

Network detail

- Simpler models capture the phenomena on main corridors but miss details on minor parts of the network
- Many different ways to code the same thing
- Should compare different coding approaches (we do not have time for this usually)

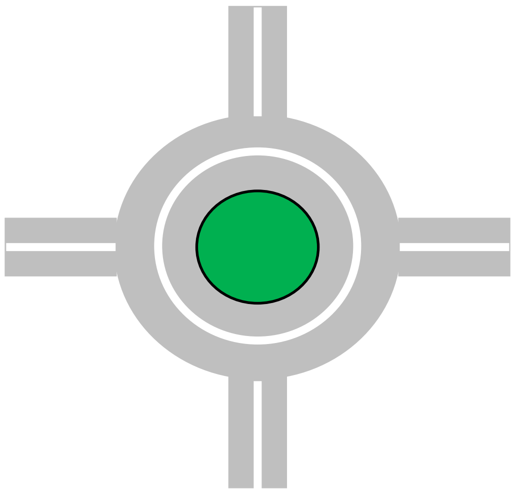
Detailed network	**Less detailed network**
More detail shows behaviour in all areas of the network e.g. what is the flow of a specific right turn	Less calculations, faster to run
Some problems may be captured only if there is sufficient detail	Can detect network errors faster
Higher computational cost	Easier and faster to examine and understand the outputs
We must ensure that all details are coded consistently in the network	It's "less of a problem" that we do not have information about the network of a future scenario
May be difficult to track errors and issues or not have enough time	We cannot answer some more detailed questions
We need to assume that our details will not be affected in the future scenario	Some information are necessary to capture some traffic phenomena

Network detail – different ways to draw the network

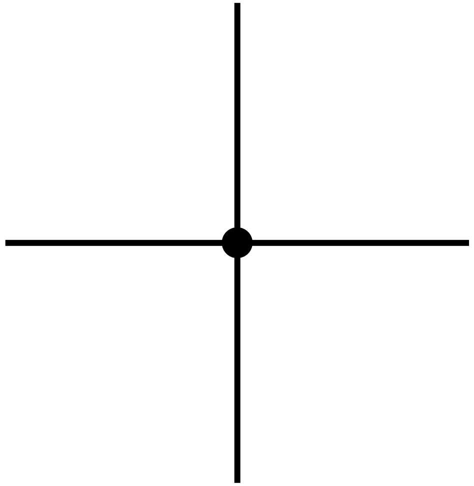
- Simpler models capture the phenomena on main corridors but miss details on minor parts of the network
- Many different ways to code the same thing
- Should compare different coding approaches (we do not have time for this usually)

Example:

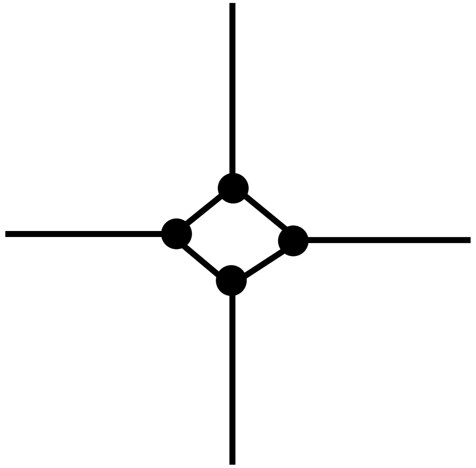
Real life



Simple representation



Detailed representation



Choosing a path…

Endless combinations from origin *i* to destination *j*

We either find try to enumerate all and remove the unreasonable or proceed only with the reasonable

What is reasonable when choosing a path?

- Shorter distance – longer to travel (user may or may not know) – less relevant nowadays when real time information is available
- Longer but better travel conditions (e.g. nicer trains)
- Faster but more expensive (e.g. tolls)
- Monetary cost e.g. fuel cost
- Shorter but steeper; maybe I do not like this if walking is involved
- Individual heterogeneity

The assignment problem

Each route has a generalised cost – routes with lower generalised cost are considered as more reasonable to be chosen

The assignment step is expected to replicate the way travellers choose the exact path from origin to destination

Complex problem:

- High number of OD
- Many possible routes for OD
- Similar routes for different OD, overlaps

Several models have been proposed to deal with network assignment

Types of assignment models

		Stochastic effects	
		No	Yes
Congestion effects	No	All-or-Nothing	Stochastic (Dial's method)
	Yes	Wardrop's user equilibrium (UE)	Stochastic user equilibrium (SUE)

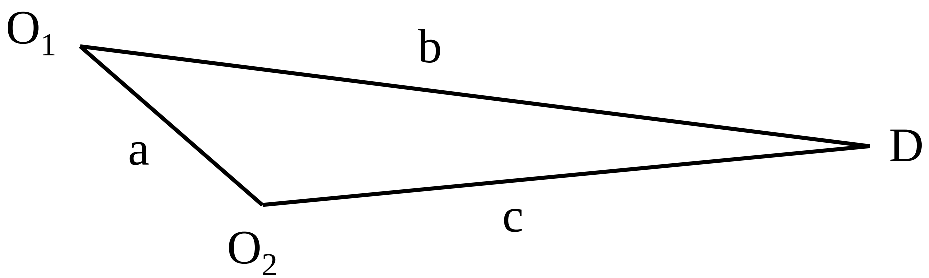
All-or-Nothing (AoN) assignment

- All trips between a given OD pair are assigned to the route with the lowest cost ignoring congestion effects
- All drivers are assumed to have perfect knowledge of the cost of each link (fixed cost with known value)
- Some links may be overused while others never considered.

Steps:

1. Definition of shortest path for each OD
2. Load the traffic to the links of a shortest path for every OD
3. Sum the flow per link after every OD assignment

All-or-Nothing (AoN) assignment – Example



Demand:

- O_1 to D: 4000 trips
- O_2 to D: 6000 trips

Cost:

- Link a: 2
- Link b: 10
- Link c: 5

Links a, b, c are two way links

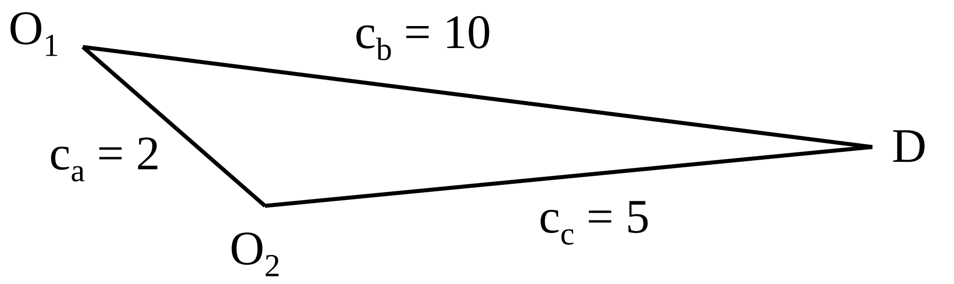
All-or-Nothing (AoN) assignment – Example

- We determine the cost of all possible paths:

  O_1 to D: $c_b = 10$ and, $c_a + c_c = 2 + 5 = 7$
  O_2 to D: $c_c = 5$ and, $c_a + c_b = 2 + 10 = 12$

- We assign the trips to the path with the lowest cost:
  - For O_1, all trips are assigned to links a and c ($f_a = 4000$, $f_b = 0$, $f_c = 4000$)
  - For O_2, all trips are assigned to link c ($f_a = 0$, $f_b = 0$, $f_c = 6000$)

- Total trips per link:
  - $f_a = 4000$
  - $f_b = 0$
  - $f_c = 10000$



All-or-Nothing (AoN) assignment – Example

- If we have a link cost function, we can compute new costs per link
  - For instance, let's assume the following link cost functions

    link a: 2 + flow/2000

    link b: 10 + flow/2000

    link a: 5 + flow/2000

Simplified example of a link cost function

$$t(v) = t_0 \left[1 + a\left(\frac{v}{c}\right)^{\beta}\right]$$

- The new link costs are:
  - $c_a = 2 + 4000/2000 = 4$
  - $c_b = 10$
  - $c_c = 10$

- The new path costs are:

  O_1 to D: $c_b = 10$ and, $c_a + c_c = 4 + 10 = 14 \Longrightarrow c_{O_1D}^{min} = 10$

  O_2 to D: $c_c = 10$ and, $c_a + c_b = 4 + 10 = 14 \Longrightarrow c_{O_2D}^{min} = 10$

All-or-Nothing (AoN) assignment – Example

Disequilibrium:

$$\delta = \frac{\sum_{ijr} T_{ijr}\left(C_{ijr} - C_{ijr}^{min}\right)}{\sum_{ijr} T_{ij} C_{ij}^{min}} = \frac{T_{O_1D}^{ac}\left(c_{O_1D}^{ac} - c_{O_1D}^{min}\right) + T_{O_1D}^{b}\left(c_{O_1D}^{b} - c_{O_1D}^{min}\right) + T_{O_2D}^{ab}\left(c_{O_2D}^{ab} - c_{O_2D}^{min}\right) + T_{O_2D}^{c}\left(c_{O_2D}^{c} - c_{O_2D}^{min}\right)}{T_{O_1D} c_{O_1D}^{min} + T_{O_1D} c_{O_2D}^{min}}$$

$$= \frac{4000 \times (14 - 10) + 0 \times (10 - 10) + 0 \times (10 - 10) + 6000 \times (14 - 10)}{4000 \times 10 + 6000 \times 10} = 0.4$$

The shortest path problem

The AoN requires the shortest path (or lowest cost) between the origin and the destination

- How can we determine the shortest path?

Input:

- The road network composed by links and nodes.
- Definition of an origin and a destination node
- Each link is associated with some cost (travel time, generalised cost etc.)

Objective:

- To define the shortest path from the origin node to the destination node
  - The definition of *shortest* variable depends on the analyst and it can be travel time, pure monetary cost, generalised cost etc.

The shortest path problem – the Dijkstra algorithm

- Determines the shortest path between two nodes
- Assumption: links have strictly positive cost
- Iterative process:
  - For each node i, we define an indicator l_i which is the minimum cost from the origin node to node i, for the current iteration
  - For each node i, we define an indicator p_i which represents the node before the node node i, in the current shortest path
- Indicators/nodes: permanent if we have found the shortest solution, otherwise temporary
- We store the results on a list that we update after each iteration
- The list includes the temporary nodes that should be examined in the following iterations

The Dijkstra algorithm – steps

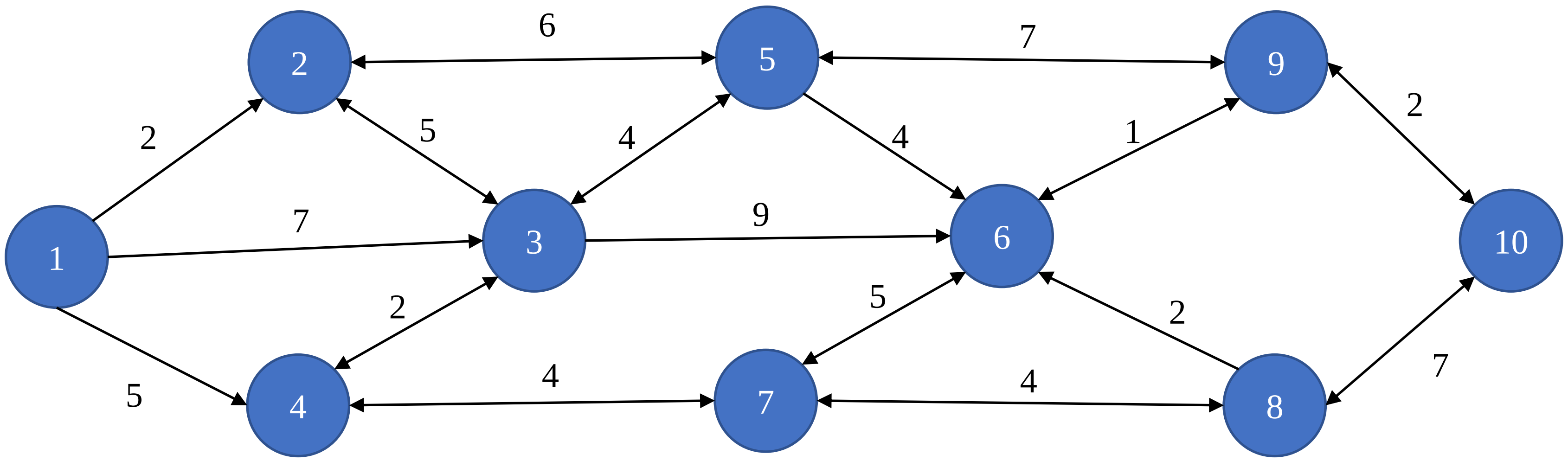
- Step 0: Initiation step
  - We define the origin node
  - $l_i = 0$ for the origin node, $l_i = \infty$ for the other nodes
  - We add the origin node to the list: L={o}
- Step 1: Optimisation step (stopping criterion)
  - If the list is empty, the algorithm is terminated. The indicators at the current loop represent the minimum cost from the origin node to the destination node
  - If the list is non-empty, we move to step 2
- Step 2: Selection step
  - We chose the node with the lowest l_i from the list and make it permanent
  - We consider this node as the node to investigate and we remove it from the list

The Dijkstra algorithm – steps

- Step 3: Update step
  - We find all nodes that are directly linked to the node under investigation and add them to the list
  - We update the cost of each node *j* as: $l_j = l_i + c_{ij}$
    - We only update if the new value is smaller than the current c_{ij} (from a previous iteration)
  - We set $p_j = i$ (we only update if we also updated c_{ij})
  - We add the nodes *j* in the list
  - We return to step 1

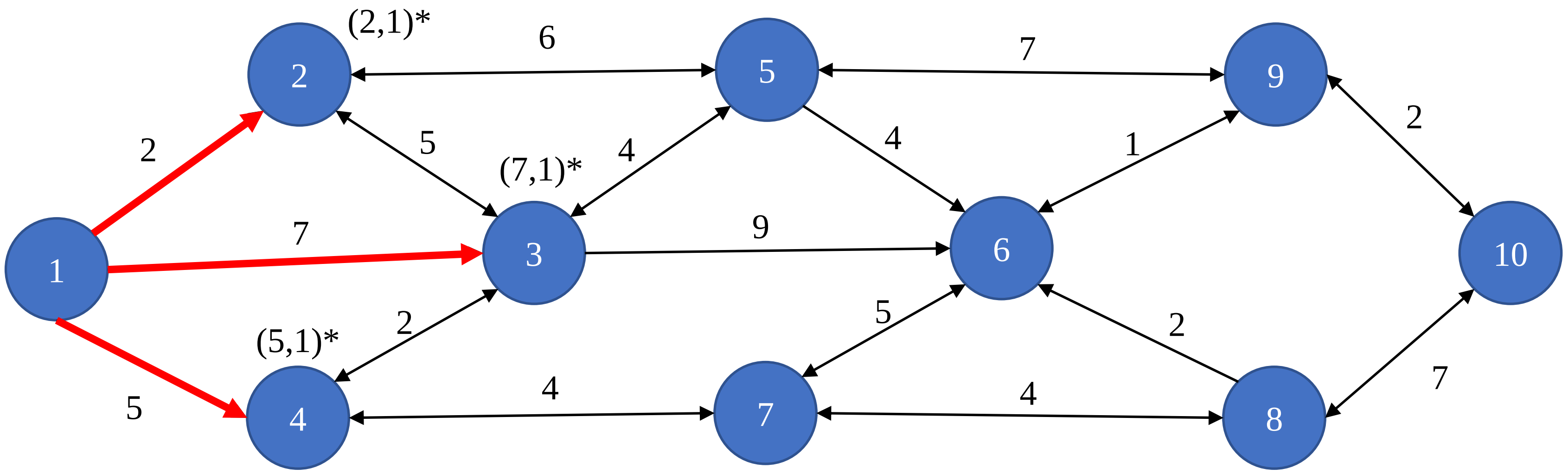
The Dijkstra algorithm – example

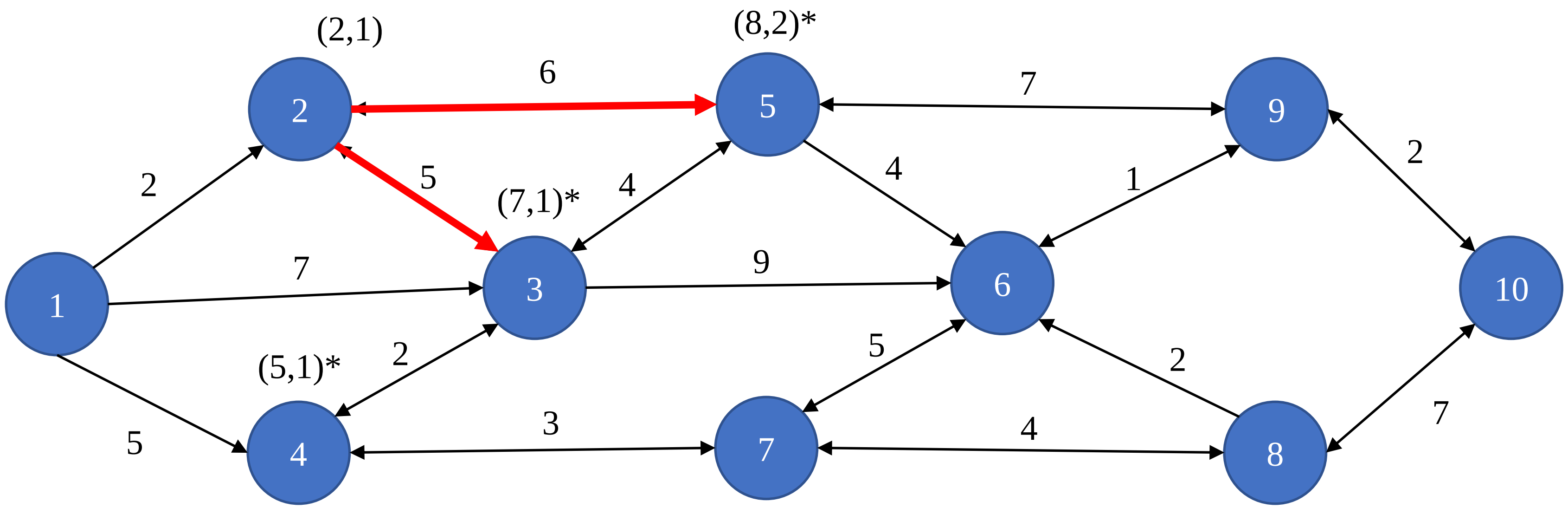
Implement the Dijkstra algorithm from node 1 to node 10



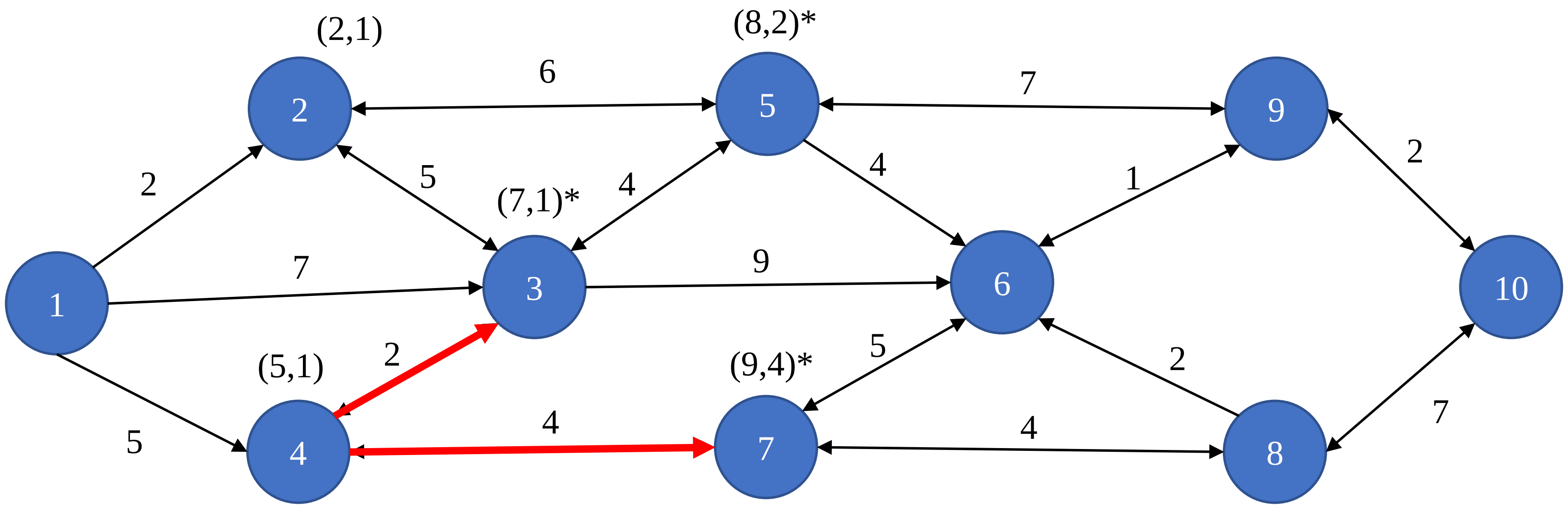
- In the solution we will use a bracket format $(l_i, p_i)^*$ to update the total cost of a node
- The asterisk denotes a temporary node
- Brackets without an asterisk suggest a permanent node

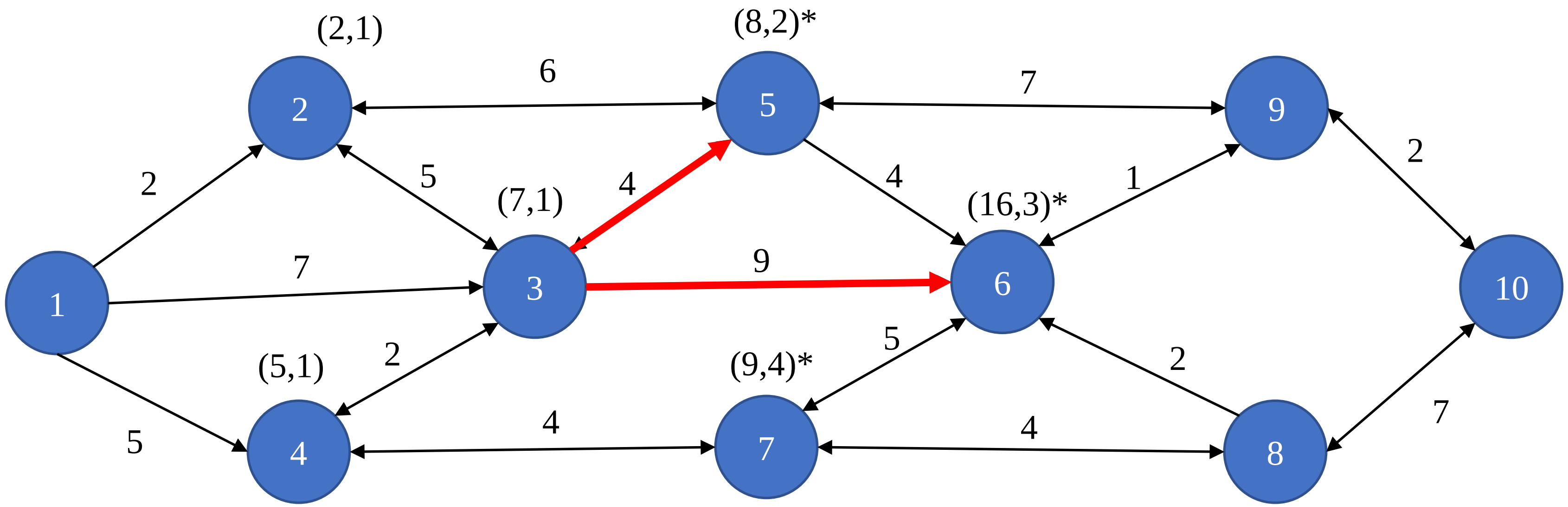
The Dijkstra algorithm – example



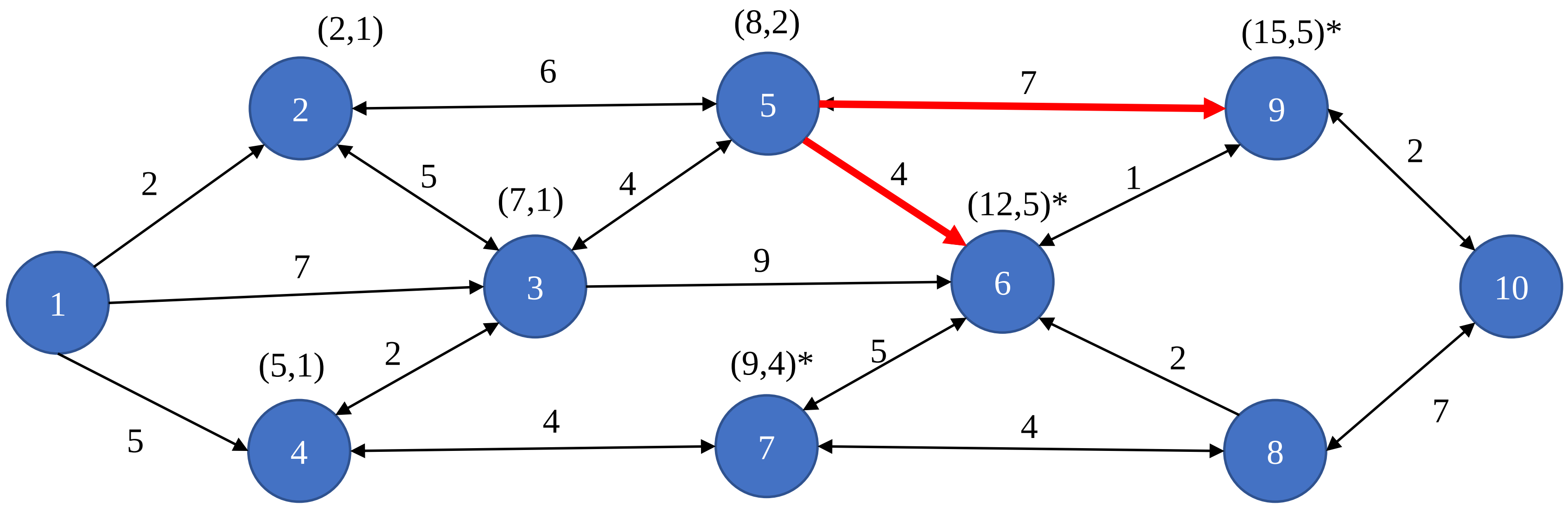


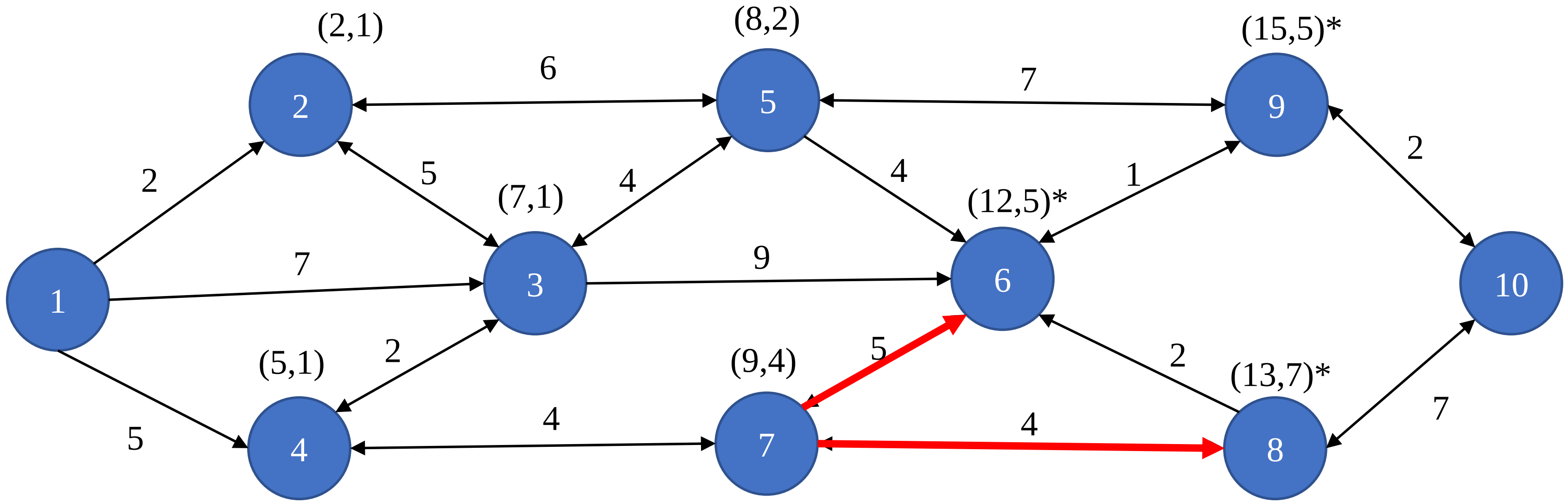
The Dijkstra algorithm – example



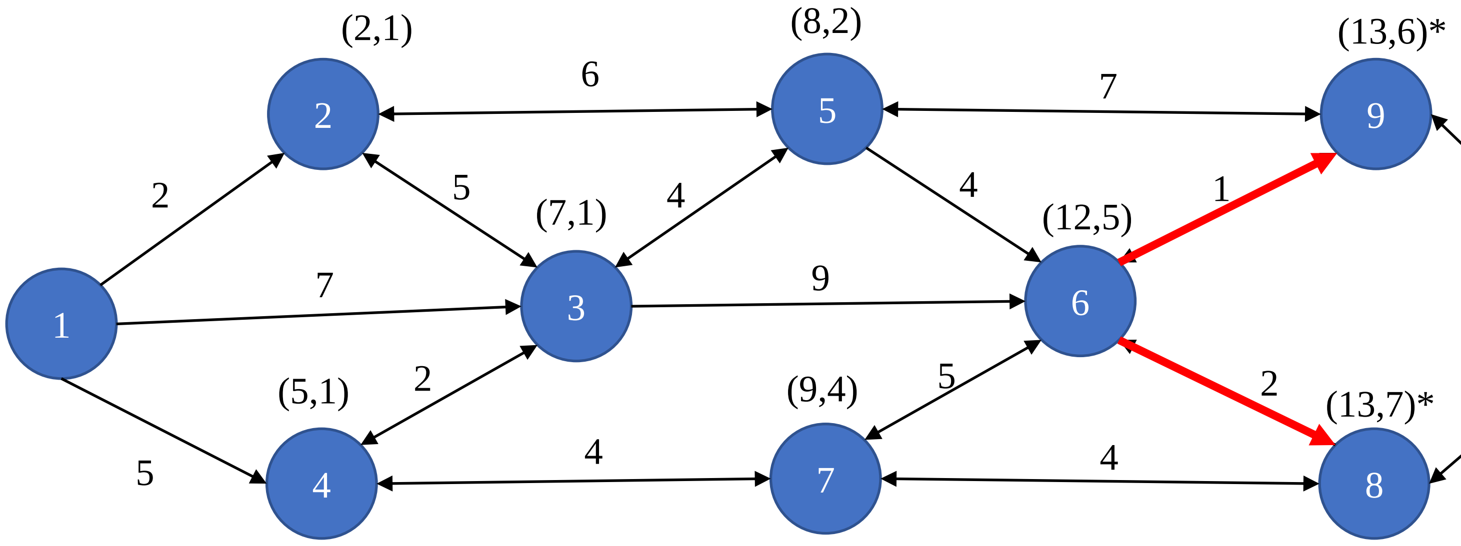


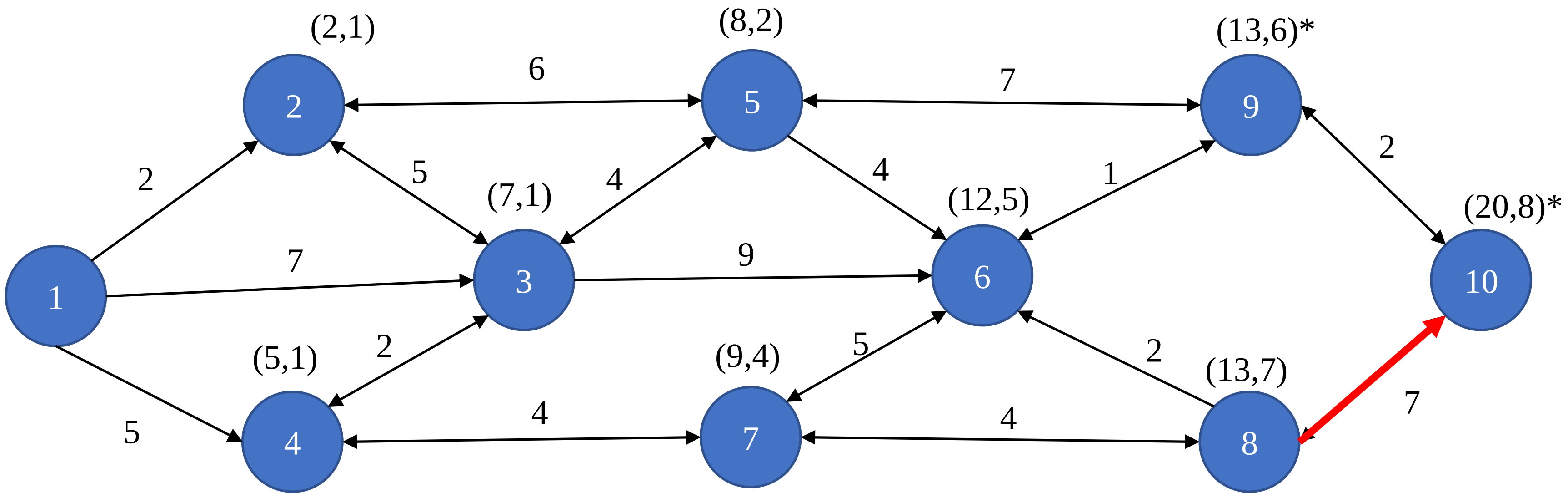
The Dijkstra algorithm – example



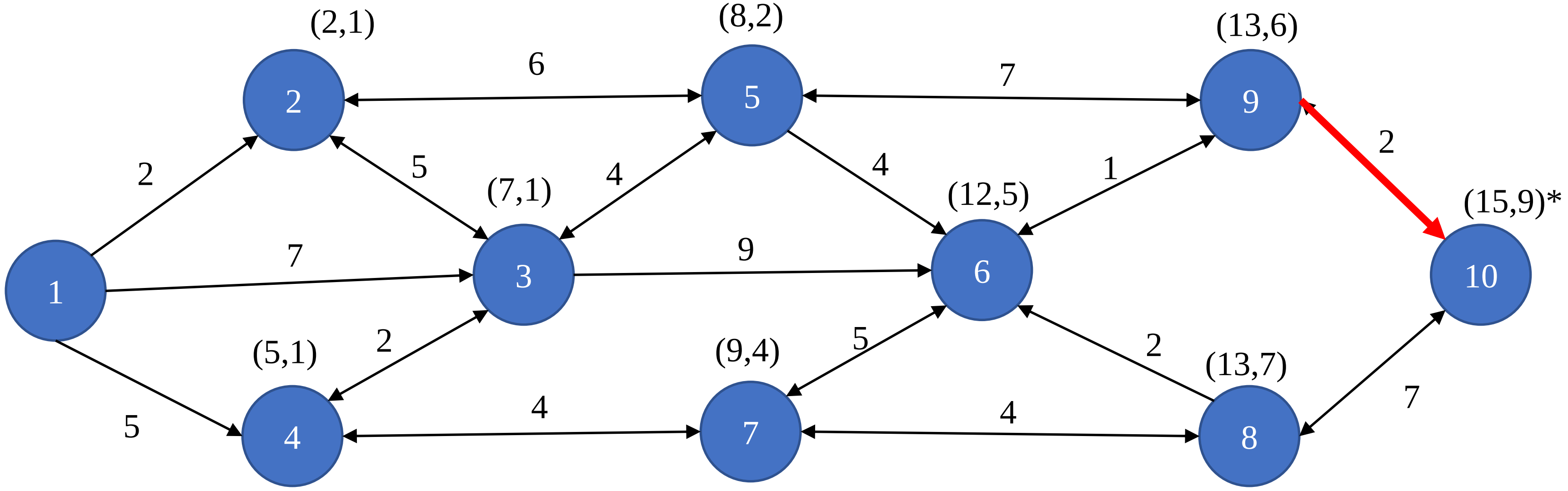


The Dijkstra algorithm – example

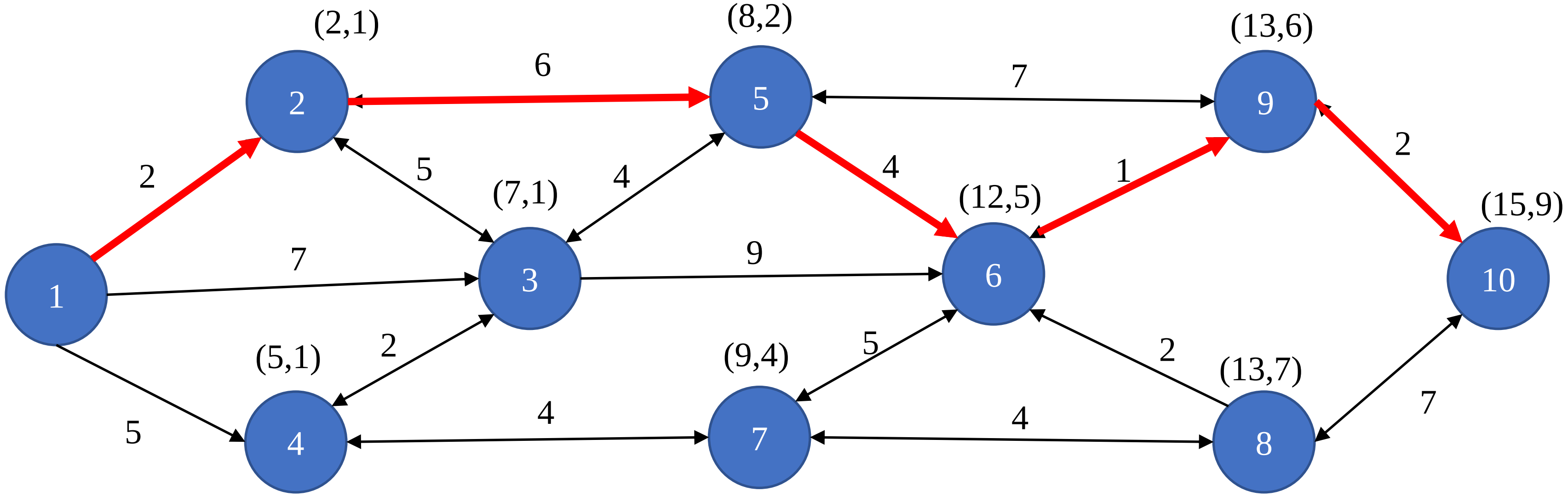




The Dijkstra algorithm – example



Shortest path



Stochastic models models

		Stochastic effects	
		No	Yes
Congestion effects	No	All-or-Nothing	Stochastic (Dial's method)
	Yes	Wardrop's user equilibrium (UE)	Stochastic user equilibrium (SUE)

Dial’s method

- It’s a proportion-based method
- Splitting trips of each OD pair among different paths according to the costs of the paths
- A logit model specification is implemented (similar approach to mode choice)

$$P_{irj} = \frac{exp(-\beta C_{ijr})}{\sum_{k \in R_{ij}} exp(-\beta C_{ijr})}$$

- Where

P_{ijr} = proportion of flow from *i* to *j* using route *r*

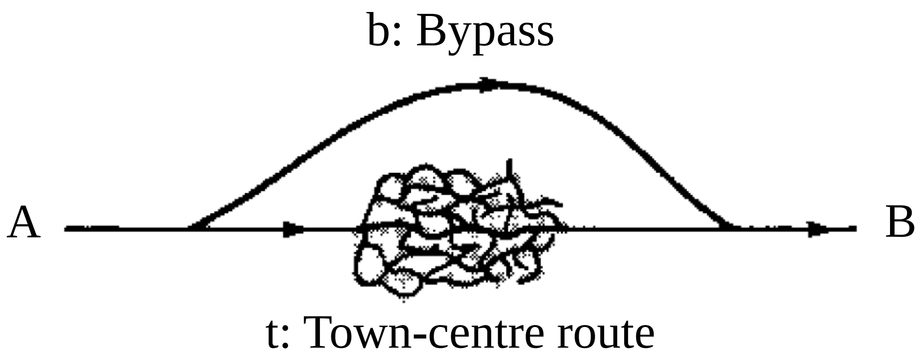
C_{ijr} = travel cost from *i* to *j* using route *r*

R_{ijr} = set of all routes from *i* to *j*

β = constant coefficient

Dial's method – Example

Network:



Two routes:
b: bypass
t: town–centre route

Demand: $T = T_b + T_t = 400$ trips

Cost functions

route	Cost function	Free flow cost
b	$12 + 0.003T_b$	15
t	$10 + 0.01T_t$	12

Exercise: Implement Dial's method with $\beta = 0.5$

Dial's method – Example

$$P_b = \frac{exp(-0.5 \times 15)}{exp(-0.5 \times 15) + exp(-0.5 \times 12)} = \frac{0.000553}{0.000553 + 0.002479} = 0.182$$

$$T_b = P_b \times T = 0.182 \times 400 = 73$$

$$T_t = 400 - 73 = 327$$

$$C_b = 12 + 0.003 \times 73 = 12.22$$

$$C_t = 10 + 0.01 \times 327 = 13.27$$

Dial's method – Example

Minimum cost route:

$$C_{min} = C_b = 12.22$$

Total network cost:

$$C = T_t C_t + T_b C_b = 5231.35$$

Measure of convergence:

$$\delta = \frac{\sum_{ijr} T_{ijr}\left(C_{ijr} - C_{ijr}^{min}\right)}{\sum_{ijr} T_{ij} C_{ij}^{min}} = \frac{T_t\left(C_t - C^{min}\right) + T_b\left(C_b - C^{min}\right)}{T C^{min}}$$

$$= \frac{332 \times (13.27 - 12.22) + 73 \times (12.22 - 12.22)}{400 \times 12.22} = 0.071$$

Dial’s method – Summary

- Greater spread of trips in the network (e.g. compared to AoN)
- Routes with lower costs are assigned more trips but not all trips
- It is still a logit model - Independence of irrelevant alternatives (IIA) assumption
  - Ignores the correlation between similar alternatives
  - Allocate more traffic to dense sections of the network with shorter links, than sparse parts of the network with longer links

Considering congestion in assignment

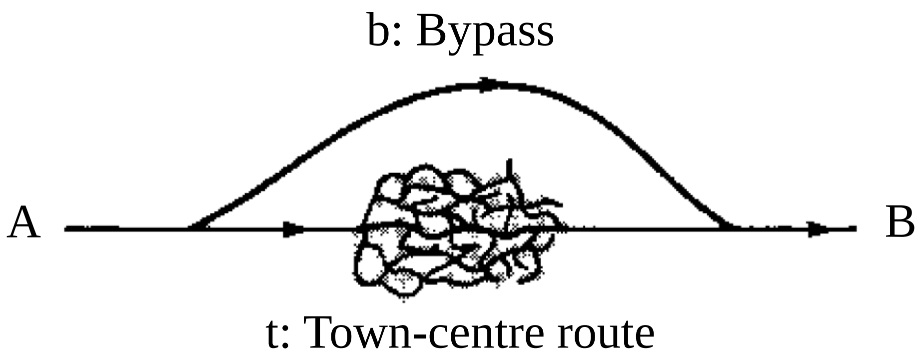
		Stochastic effects	
		No	Yes
Congestion effects	No	All-or-Nothing	Stochastic (Dial's method)
	Yes	Wardrop's user equilibrium (UE)	Stochastic user equilibrium (SUE)

Congested assignment – Equilibrium

- User Equilibrium (UE) Principle (also called Wardrop's first principle**)**
  - "*No individual trip maker can reduce their path costs by switching routes*", i.e. travel costs of the routes are equal.
  - Under User Equilibrium (UE) conditions, traffic in a congested network distributes such that all routes used between an origin-destination (OD) pair have equal and minimal travel costs, while any unused routes have travel costs that are equal to or greater than those of the used routes.
- System Optimal (SO) Principle (social equilibrium/Wardrop's 2nd principle**)**
  - Minimise the total cost of travel in the network.
  - Under System Optimal (SO) conditions, traffic in a congested network should be allocated in a manner that minimizes the total (or average) travel cost across the entire system.

UE & SO – Example

Network:



Two routes:
b: bypass
t: town–centre route

Demand: $T = T_b + T_t = 400$ trips

Cost functions

route	Cost function
b	$12 + 0.003T_b$
t	$10 + 0.01T_t$

Exercise: Implement UE assignment and SO assignment

Example – UE assignment

For user equilibrium we need:

$C_b = C_t \Longrightarrow 12 + 0.003T_b = 10 + 0.01T_t$

$T = T_b + T_t$

$\Longrightarrow T_b = 0.77T - 153.84$

<u>Two different cases</u>

- $T \leq 200$ (153.84/0.77) then $T_b = 0.77T - 153.84 \leq 0$
  - $C_b = 12$
  - $C_t \leq 12$
  - All traffic can use route *t*
- $T > 200$ then $T_b = 0.77T - 153.84 > 0$, both routes are used
- $T = 400$ then $T_b = 154$, $T_t = 246$, $C_b = 12.46$, $C_t = 12.46$
  - $\delta = \frac{\sum_{ijr} T_{ijr}\left(C_{ijr} - C_{ijr}^{min}\right)}{\sum_{ijr} T_{ij} C_{ij}^{min}} = \frac{T_t\left(C_t - C^{min}\right) + T_b\left(C_b - C^{min}\right)}{TC^{min}} = \frac{246 \times (12.46 - 12.46) + 154 \times (12.46 - 12.46)}{400 \times 12.46} = 0$

Example – SO assignment

- For system optimal we should minimise the total travel cost $C = T_b\, C_b + T_t\, T_t$:

$$C = T_b\, C_b + T_t\, T_t = T_b\,(12 + 0.003T_b) + (T- T_b)\,(10 + 0.01\,(T- T_b)) = 10T + 0.01T^2 + 2T_b + 0.013T_b^2 - 0.02\mathrm{T}T_b$$

- To get the minimum value of C:

$$\frac{dC}{dT_b} = 0.026T_b + 2 - 0.02T = 0 \Rightarrow T_b = \frac{0.02T - 2}{0.026}$$

- For T = 400 we have $T_b = 231$, $T_t = 169$, $C_b = 12.693$, $C_t = 11.69$
- $\delta = \frac{\sum_{ijr} T_{ijr}\left(C_{ijr} - C_{ijr}^{min}\right)}{\sum_{ijr} T_{ij} C_{ij}^{min}} = \frac{T_t(C_t - C^{min}) + T_b(C_b - C^{min})}{TC^{min}} = \frac{169\times(11.69-11.69)+231\times(12.693-11.69)}{400\times11.69} = 0.05$

Total costs:

- SO: C = 231×12.693 + 169×11.69 = 4907.693
- UE: C = 154×12.460 + 246×12.46 = 4984.000

Considerations on network equilibrium

For large-scale projects, OE or UE equilibrium is difficult to be obtained:

- Too many paths and OD pairs
- More complicated link cost functions
- Some paths may not be used
- Assignment is approached via algorithms
  - Method of successive average (MSA)
  - Frank–Wolfe algorithm

Method of successive average (MSA)

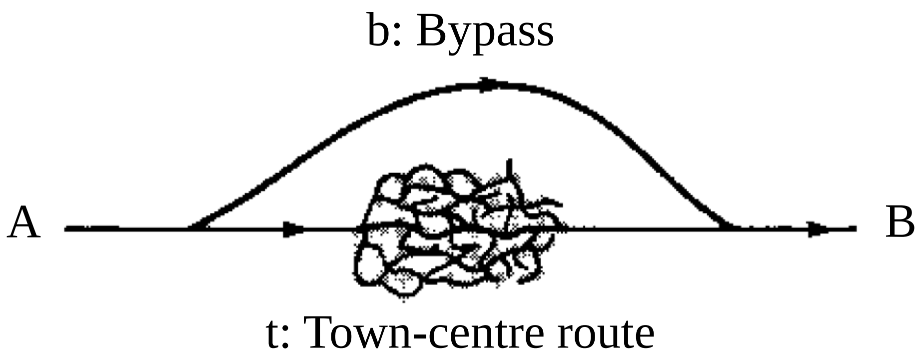
Method of successive average (MSA) – eventually converges to UE

Process:

1. Set $k = 0$, and the current flows to $T^{(k)} = 0$. We initialise the link costs using the free flow costs
2. $k = k + 1$. Then, we assign all demand with AoN assignment using the current iteration costs and we compute auxiliary flows F.
3. We compute the new flows as $T^{(k)} = (1 - \alpha)\, T^{(k-1)} + \alpha\, F$, where $\alpha = 1/k$. Then, we calculate the current costs in the network according to $T^{(k)}$. Finally, we calculate the measure of convergence δ.
4. If δ close to zero STOP and output $T^{(k)}$, else go to step 2.

Method of successive average (MSA) – Example

Network:



Two routes:
b: bypass
t: town–centre route

Demand: $T = T_b + T_t = 400$ trips

Cost functions

route	Cost function	Free flow cost
b	$12 + 0.003T_b$	15
t	$10 + 0.01T_t$	11

Exercise: Implement the Method of successive average (MSA)

Method of successive average (MSA) – Example

Iteration 0	$T_b^{(0)} = 0$, $T_t^{(0)} = 0$, k = 0
Iteration 1	α = 1, 1 – α = 0, k = 1
	$C_b = 15$, $C_t = 11$, T = 400
	$F_b = 0$, $F_t = 400$ $T_b^{(1)} = (1 - \alpha)\, T_b^{(0)} + \alpha\, F_b = 0$, $T_t^{(1)} = 400$ $C_b = 12 + 0.003T_b = 12$, $C_t = 10 + 0.01T_t = 14$
Iteration 2	α = 1/2, 1 – α = 1/2, k = 2
	$C_b = 12$, $C_t = 14$, T = 400
	$F_b = 400$, $F_t = 0$ $T_b^{(2)} = (1 - \alpha)\, T_b^{(1)} + \alpha\, F_b = 200$, $T_t^{(2)} = 200$ $C_b = 12 + 0.003T_b = 12$, $C_t = 10 + 0.01T_t = 14$

Method of successive average (MSA) – Example

Iteration 3	$\alpha = 1/3$, $1 - \alpha = 2/3$, $k = 3$
	$C_b = 12.6$, $C_t = 12$, $T = 400$
	$F_b = 0$, $F_t = 400$ $T_b^{(3)} = (1 - \alpha)\, T_b^{(2)} + \alpha\, F_b = 133$, $T_t^{(3)} = 267$ $C_b = 12 + 0.003T_b = 12.399$, $C_t = 10 + 0.01T_t = 12.67$
Iteration 4	$\alpha = 1/4$, $1 - \alpha = 3/4$, $k = 4$
	$C_b = 12.399$, $C_t = 12.67$, $T = 400$
	$F_b = 400$, $F_t = 0$ $T_b^{(4)} = (1 - \alpha)\, T_b^{(3)} + \alpha\, F_b = 200$, $T_t^{(4)} = 200$ $C_b = 12 + 0.003T_b = 12.6$, $C_t = 10 + 0.01T_t = 12$
Iteration 5	$\alpha = 1/5$, $1 - \alpha = 4/5$, $k = 5$
	$C_b = 12.6$, $C_t = 12$, $T = 400$
	$F_b = 0$, $F_t = 400$ $T_b^{(5)} = (1 - \alpha)\, T_b^{(4)} + \alpha\, F_b = 200$, $T_t^{(5)} = 200$ $C_b = 12 + 0.003T_b = 12.6$, $C_t = 10 + 0.01T_t = 12$

Method of successive average (MSA) – Example

If we stop at iteration 5 then:

- $T_b = 160$, $T_t = 240$, $C_b = 12.48$, $C_t = 12.4$
- $\delta = \frac{\sum_{ijr} T_{ijr}\left(C_{ijr} - C_{ijr}^{min}\right)}{\sum_{ijr} T_{ij} C_{ij}^{min}} = \frac{T_t\left(C_t - C^{min}\right) + T_b\left(C_b - C^{min}\right)}{TC^{min}} = \frac{160 \times (12.4 - 12.4) + 240 \times (12.48 - 12.4)}{400 \times 12.4} \approx 0$
- Eventually (in this simple example) with enough iterations we will get $\delta = 0$
- Results from the previous UE example: $T = 400$ then $T_b = 154$, $T_t = 246$, $C_b = 12.46$, $C_t = 12.46$
- The MSA can take a while to reach convergence
    - Alternative: the Frank–Wolfe algorithm

The Frank–Wolfe algorithm

Steps (generalised version of MSA)

1. Set $k = 0$, and the current flows to $T^{(k)} = 0$. We initialise the link costs using the free flow costs
2. We compute the minimum cost based on the current iteration costs; make $k = k + 1$. Then, we assign all demand with AoN assignment using the current iteration costs and we compute auxiliary flows F.
3. We compute the new flows as $T^{(k)} = (1 - \alpha)\, T^{(k\text{-}1)} + \alpha\, F$, choosing α such that the value of the objective function Z is minimised.
    - $\min Z = \sum_a \int_{x=0}^{T_a} t_a(x)\, dx$, where n are the paths and $t_a(x)$ is the cost function
4. Calculate a new set of current link costs based on the flows $T^{(k)}$; we check if our convergence indicator (e.g. δ <0.0001) to decide whether to stop or to proceed to step 2.

The Frank–Wolfe algorithm – bisection method

- It is a standard practice to use a bisection method instead of minimising the Z function.
- The algorithm changes then as follows (Instead of steps 3 and 4 in the previous slide):

1. We need to calculate α, for $0 \leq \alpha \leq 1$
2. We set $\alpha_{\text{min}} = 0$, $\alpha_{\text{max}} = 1$
3. We set $\alpha = (\alpha_{\text{min}} + \alpha_{\text{max}}) / 2$
4. We compute: $\frac{dZ}{d\alpha} = \sum_{all\ links}(\mathrm{F} - T^{(k-1)}) \times c_{link}\left(T^{(k-1)} - a(\mathrm{F} - T^{(k-1)})\right)$
5. If $\frac{df}{d\alpha} < 0$ then $\alpha_{\text{min}} = \alpha$, else if $\frac{df}{d\alpha} > 0$ $\alpha_{\text{max}} = \alpha$
6. If $\alpha_{\text{max}} - \alpha_{\text{min}} <$ threshold then $\alpha = (\alpha_{\text{min}} + \alpha_{\text{max}}) / 2$ and we move to step 7, else go to step 3
7. Update flows with $T^{(k)} = (1 - \alpha)\, T^{(k\text{-}1)} + \alpha\, F$
8. Repeat until δ <0.0001 or some other small value

Traffic assignment – Summary

- The assignment process – model input / output
- Equilibrium
- Factors affecting route choice
- Assignment models
- All–or–nothing
  - Shortest path – the Dijkstra algorithm
- Wardrop equilibrium
  - Method of successive averages (MSA)
  - The Frank–Wolfe algorithm

Decision-aid methodologies in transportation

CIVIL-557

Optional slides

Evangelos Paschalidis

TRANSP-OR EPFL

Travel cost by car

- Difference between ***actual cost*** and ***perceived cost***
- Cost, fuel cost, maintenance cost but the latter may be not perceived. Also we do not refuel in every trip
- Analytical way, average total cost per trip (nobody does that!)
- Cost varies based on total amount of trips, if others use the car, if others contribute to the car costs
- Typically, price per km, covers all costs when defining the generalised cost
- Road user charging, tolls, can be difficult to incorporate

Route choice public transport – challenges

- Generalised cost, we add for number of transfers, walking time, waiting time
- Travel time and interaction with the rest of the traffic
- Bus routes (add a node per bus stop not always a good idea)
- Coding timetables and replace with average waiting time
- Crowding and how to represent
- Fare not always easy to capture

Network detail

Depending on the project needs and available resources, the level of network detail varies:

- Minor-junctions, side streets
- Setting of traffic lights
- The correct split of lanes
- Vehicle restrictions
- Shape of roundabouts
- Locations of bus stops
- Bus lanes
- Lane width

Microsimulation

- Illustrates road user behaviour individually e.g. behaviour updated every second
- Can be implemented in the whole network or at a specific junction or road segment
- Typically a 2D visualisation of the road network (3D is also possible)
- A collection of sub-models
  - Car-following
  - Lane-changing
  - Gap-acceptance
- Limitations from macroscopic models remain: impact of adverse weather conditions, interaction with pedestrians, traffic incidents, aspects of driving behaviour that are difficult to be modelled

Types of assignment models

All-or-nothing

For every OD, all the demand is assigned to the route with the lowest generalised cost

- Cons: Simplistic, may assume everyone takes the same path, main corridors only, ignoring the actual route choice
- Prons: The route choice is spread across all OD so tips will be spread on several routes anyway

All-or-nothing by trip purpose

For every OD, all the demand is assigned to the route with the lowest generalised cost

- Cons: Still simplistic
- Prons: Better approximation since commuting travellers have a different generalised cost function that leisure/shopping travellers

Types of assignment models

User equilibrium (one of the most common approaches)

Demand for each OD is distributed between different routes (based on principles of network equilibrium). No user could chose another route to reduce the generalised cost any further

- Cons: Theoretical idea, does not reflect the complexity of real life behaviour
- Prons: It splits the demand in routes that more or less have the same generalised cost. This works relatively well usually.

User equilibrium with more variation

Same idea but some routes with higher generalised cost can still be used

- Cons: More complex mathematically and the observed choices are difficult to interpret sometimes
- Prons: More realistic e.g. may capture people who choose a higher generalised cost route because the do not know the network that well

Types of assignment models

Dynamic assignment

Run several assignments for smaller time periods e.g. run the 8.00-9.00 AM morning peak in 15-min intervals and in every new interval use input from the previous

- Higher computational challenges for calibration
- More realistic, captures how traffic develops and people choose different routes across time