1.225J (ESD 225) Transportation Flow Systems

Lecture 4

Introduction to Network Models and Shortest Paths

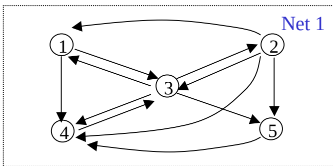
Profs. Ismail Chabini and Amedeo Odoni

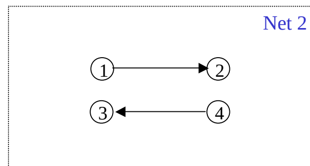
Lecture 4 Outline

- Conceptual Networks: Definitions
- Representation of an Urban Road Network (Supply)
- Shortest Paths (Reading: pp. 359-367, 6.2.3 and 6.2.4 of R6)
  - Introduction
  - Dijkstra's algorithm: example
  - Dijkstra's algorithm: statement
  - Observations
- Extensions to Classical Shortest Path Problems
- All-or-nothing traffic assignment
- Zoning and Analysis Periods (Demand)
- Motivation for more advanced traffic assignment models
- Summary

Conceptual Networks: Definitions

- A ***network*** is:
  - a set of ***nodes*** *N* and a set of ***links*** *A*
  - *nodes* are also called ***vertices*** or ***points***
  - *links* are also called ***arcs*** or ***edges***
- Examples:



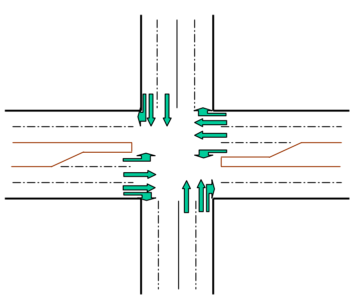


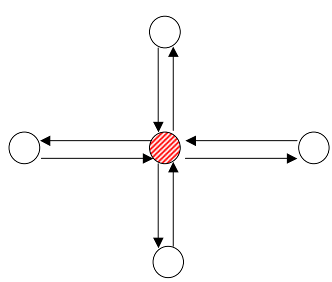
- ***Directed networks***: all links are directed
- ***Path***: a sequence of links from one node to another node (i.e., (5,4)-(4,3)-(3,2))
- A network is ***connected*** if there is at least one path from one node to another node (Net1 is connected whereas Net2 is not)

Representation of an Urban Road Network

- Physical
  - Intersections
  - Streets
  - Zones
- Conceptual
  - Nodes
  - Links
  - Centroids

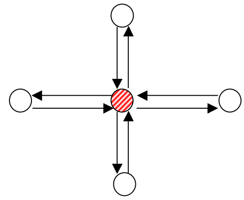
- Simple node representation



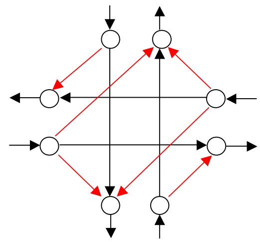


Intersection Representations

- Simple node representation:
  - no direction differenciation
  - no conflicting movement



- Subnetwork representation:
  - explicit direction representation
  - conflicting turns in an intersection are captured by internal links and their impedances

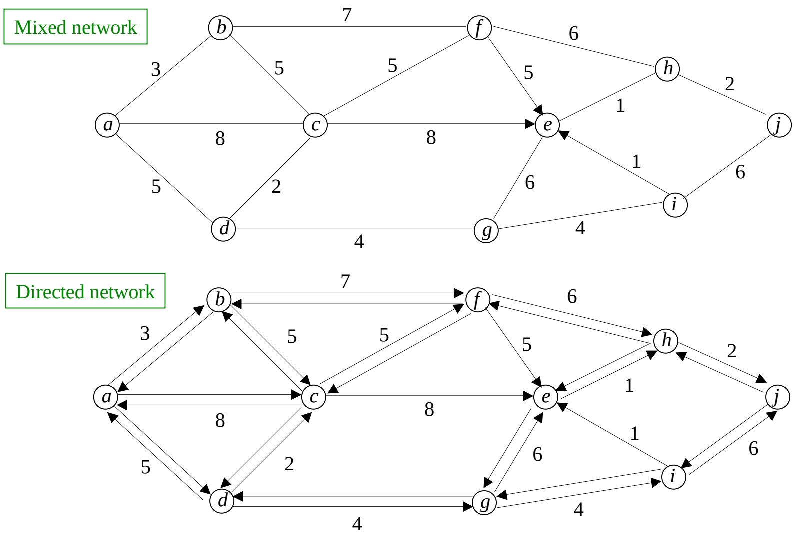


- Conceptual representation is not unique and depends on:
  - type of analysis
  - data availability to build, validate, and apply model
  - accuracy vs. computation time trade-off

Shortest Path Problems

- Basic problem: find a shortest path and the shortest distance between two nodes
- Basic problem is called the one-to-one shortest path problem
- Types of shortest path problems:
  - One-to-one
  - One-to-all: find shortest paths from one node to all nodes
  - All-to-one: find shortest paths from all nodes to one node
  - Many-to-many: find shortest paths from many nodes to many other nodes
  - All-to-all: find shortest paths from all node pairs
- "Shortest" also denotes minimum general cost
- There are hundreds of shortest path algorithms, but they are similar
- Some algorithms work for non-negative costs only

Dijkstra's Shortest Paths Algorithm: Example

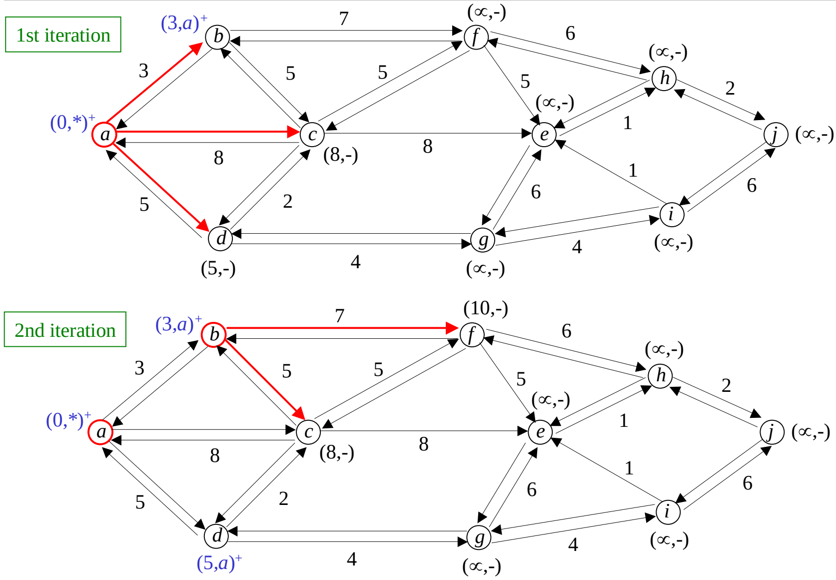


First Shortest Path Algorithm (Dijkstra's Algorithm)

- Notation:
  - *s*: source node
  - *d*(*j*): length of shortest path from *s* to *j* discovered so far
  - *p*(*j*): immediate predecessor to node *j* on shortest path from *s* to *j* discovered so far
  - *k*: last node selected by algorithm

- Step 1: Initialization
  - $d(s) = 0$, $p(s) = *$
  - $d(j) = \propto$, $p(j) = -$, for all other nodes $j \neq s$
  - $k = s$

Dijkstra's Shortest Paths Algorithm: Example



First Shortest Path Algorithm (Dijkstra's Algorithm)

- Step 2: Update labels of neighbors in open state
  - For all (k, j), if j is open do:

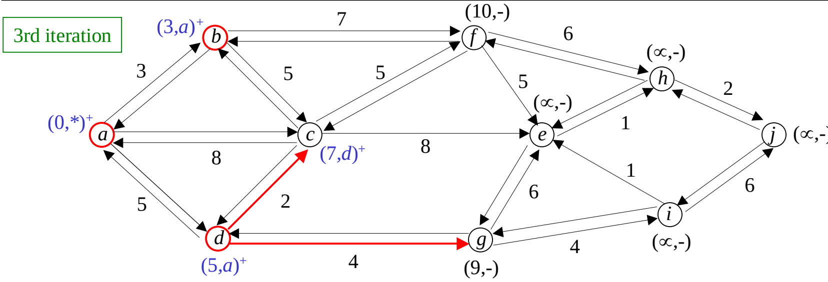
    If $d(j) < d(k) + l(k, j)$ then

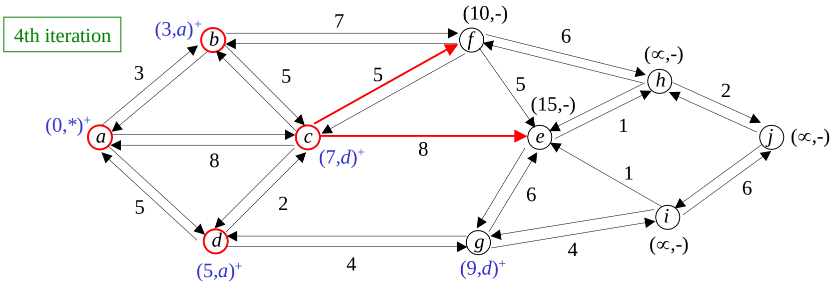
    $d(j) = d(k) + l(k, j)$

    $p(j) = k$
- Step 3
  - Find a open state node i such that $d(i) = \min\{d(j), j \text{ is an open node}\}$
- Step 4
  - Find a closed state node j^* such that $d(i) = d(j^*) + l(j^*, i)$
- Step 5
  - Node i is closed. If no node in open state , STOP.

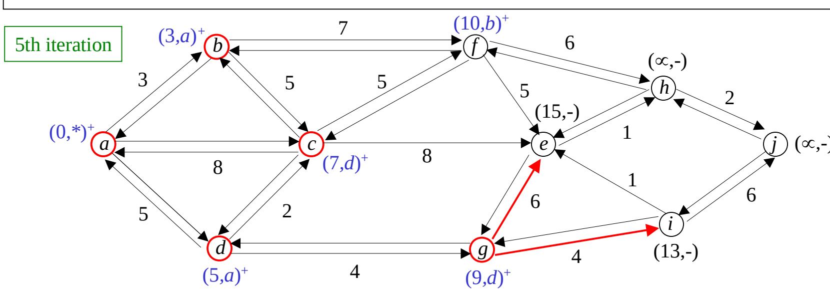
    Otherwise $k = i$, return to Step 2

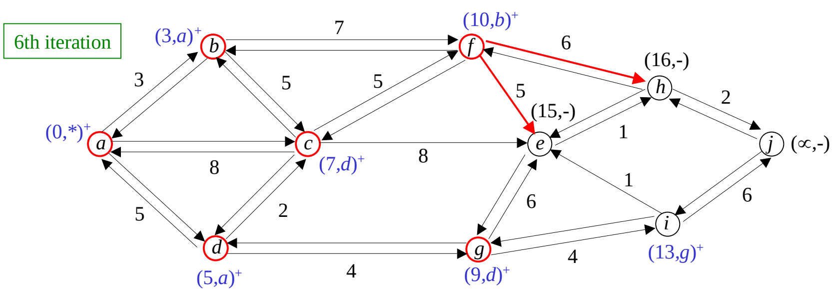
Shortest Paths Algorithm: Example



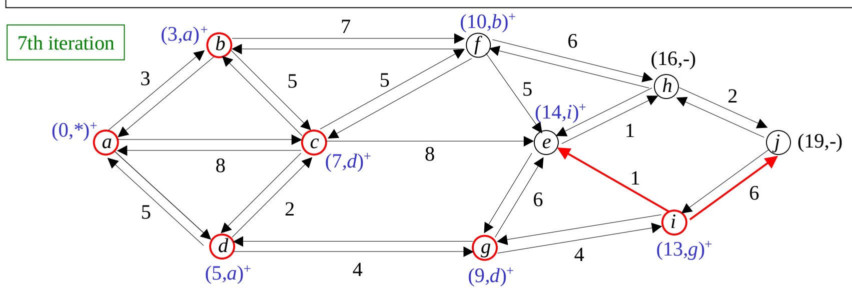


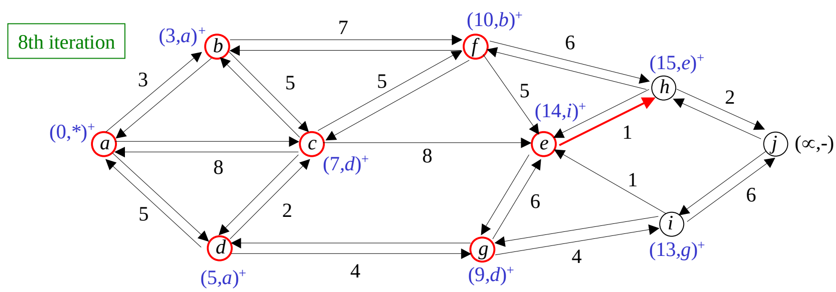
Shortest Paths Algorithm: Example



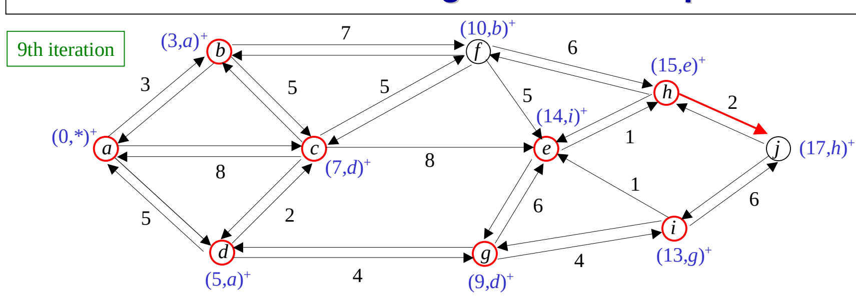


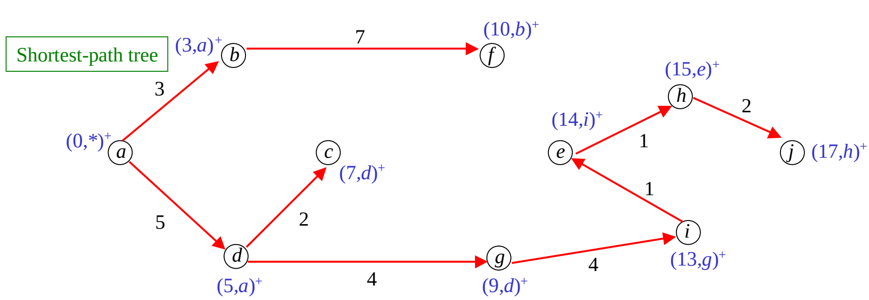
Shortest Paths Algorithm: Example





Shortest Paths Algorithm: Example





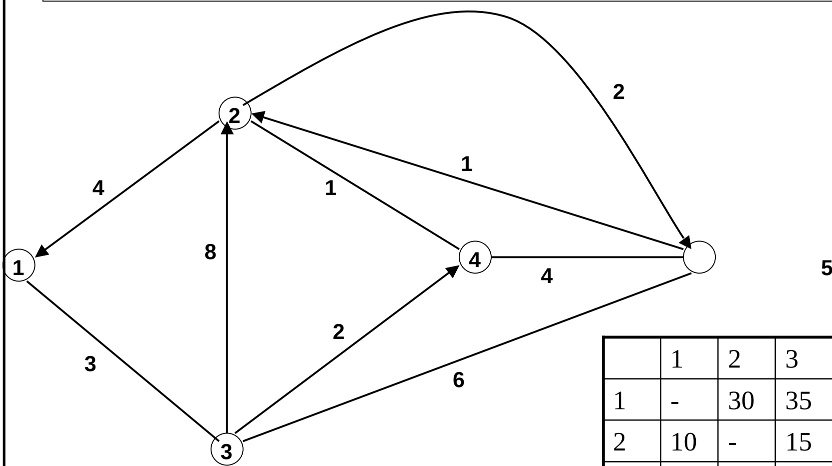
Observations about Dijkstra's Algorithm

- Dijkstra's algorithm is in general not valid if some $l(i, j) < 0$
- Shortest paths form a tree
- The algorithm can also solve the all-to-one problem
- If you solve for a one-to-many problem, stop the algorithm when all destination nodes are closed
- Shortest path problem is an LP problem, but it is more efficient and intuitive to look at it as a network problem as we did in class

Extensions of Shortest Path Problem

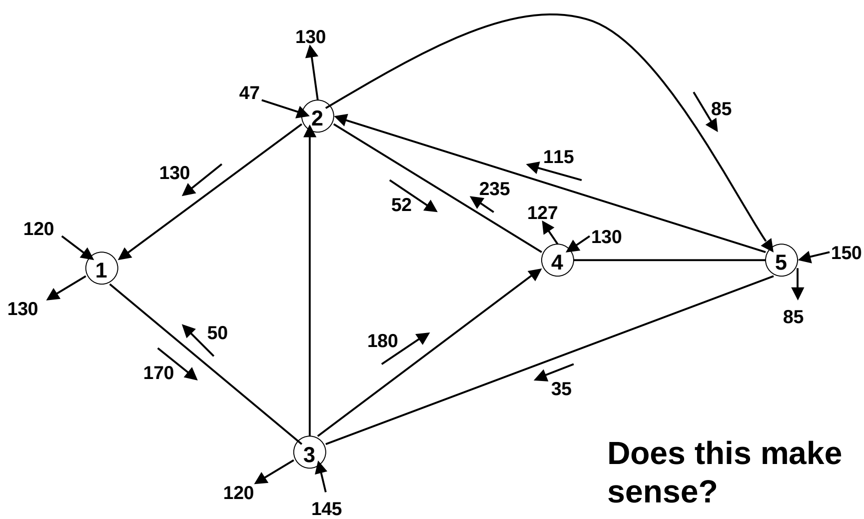
- There is a huge number of potential extensions of the classical shortest path problem
- Problems on dynamic networks (link lengths change over time)
- Problems on probabilistic networks (link lengths are random variables assuming discrete values or a continuous range of values)
- Combinations thereof
- Solutions to these problems depend on the assumptions regarding the state of knowledge and on the relative magnitude of the parameters involved
- The meaning of "shortest path" is also an issue in some cases

A Traffic Assignment Problem



	1	2	3	4	5
1	-	30	35	40	15
2	10	-	15	12	10
3	50	40	-	35	20
4	25	30	35	-	40
5	45	30	35	40	-

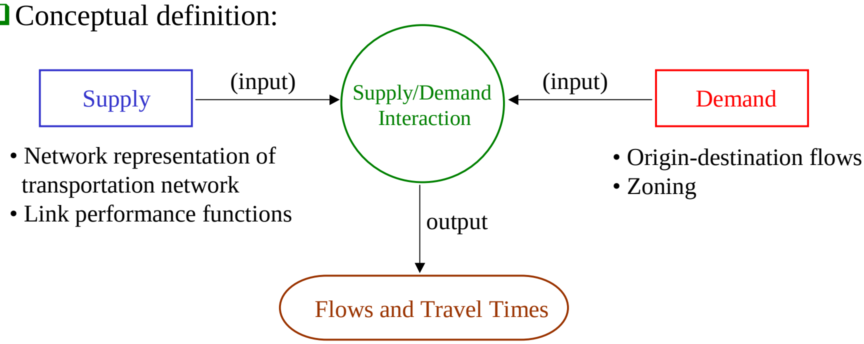
"All-or-nothing" Traffic Assignment



Does this make sense?

Traffic Assignment Models

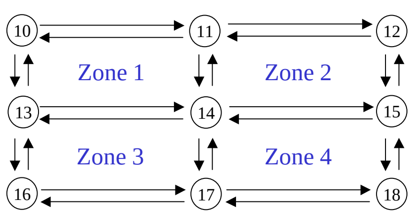
- Conceptual definition:



- Principles of assignment to represent the interaction
  - ***User Optimal (U.O.)***: O-D flows are assigned to paths with minimum travel time
  - ***System Optimal (S.O.)***: O-D flows are assigned such that total travel time on the network is minimum

Zoning

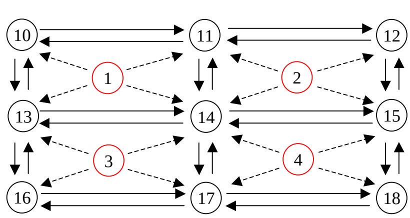
- Physical zones



- Zone-to-zone Flows

	Zone 1	Zone 2	Zone 3	Zone 4
Zone 1	0	90	120	80
Zone 2	100	0	60	130
Zone 3	120	180	0	50
Zone 4	40	70	150	0

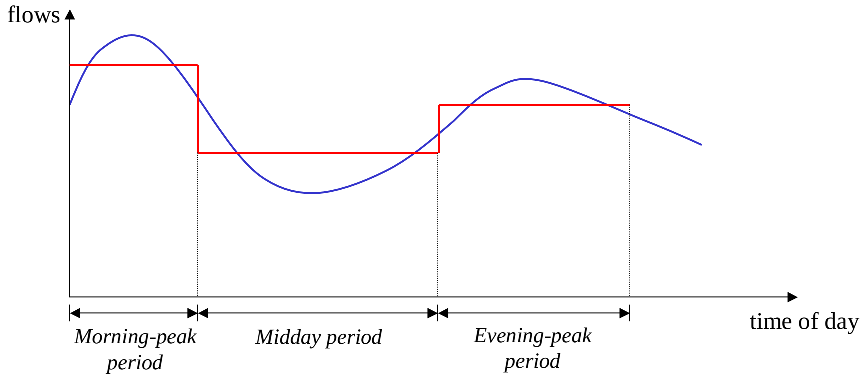
- Centroid nodes and Connectors



- O/D Flows

	1	2	3	4
1	0	90	120	80
2	100	0	60	130
3	120	180	0	50
4	40	70	150	0

Analysis Periods



- Over an analysis period, flows are assumed constant in order for steady-state analysis to apply
- The duration of a period is longer than a trip
- Typical analysis periods: morning-peak, midday, evening-peak

Lecture 4 Summary

- Conceptual Networks: Definitions
- Representation of an Urban Road Network (Supply)
- Shortest Paths (Reading: pp. 359-367, 6.2.3 and 6.2.4 of R6)
  - Introduction
  - Dijkstra’s algorithm: example
  - Dijkstra’s algorithm: statement
  - Observations
- Extensions to Classical Shortest Path Problems
- All-or-nothing traffic assignment
- Zoning and Analysis Periods (Demand)
- Motivation for more advanced traffic assignment models