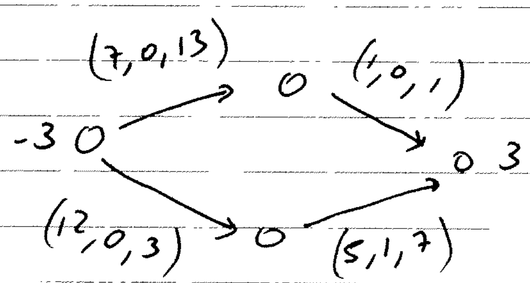
Notes 6

Minimum-cost flow:

- we are given supplys (negative) and demands (positive) at each node
- we are given upper/lower bounds on the carrying capacity of each edge
- we are given the cost of carrying a unit of goods across each edge.

Goal: To find the minimum cost plan to "clear" the market, i.e. satisfy all demands with the supplys we have.

e.g. · For each edge we have (cost, lower, upper)
· For each node we have a supply/demand.



- where is the supply?
- the demand?
- what is our shipping plan & its cost?

Lets formulate this problem as an LP.

y_{ij} - # of trucks using arc (i,j)

$$\min_{y} \sum_{(i,j)\in E} c_{ij}\, y_{ij}$$

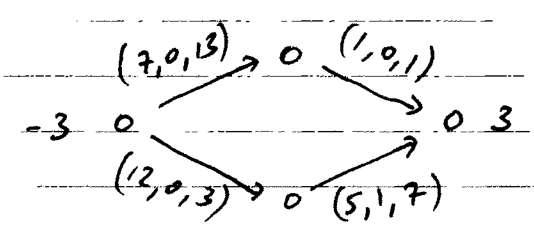
$$\text{s.t.} \quad \sum_{(i,a)\in E} y_{ia} - \sum_{(a,i)\in E} y_{ai} = b(a) \quad \forall a \in V$$

$$\ell_{ij} \le y_{ij} \le u_{ij} \quad \forall (i,j)\in E$$

$$0 \le y_{ij} \quad \forall (i,j)\in E$$

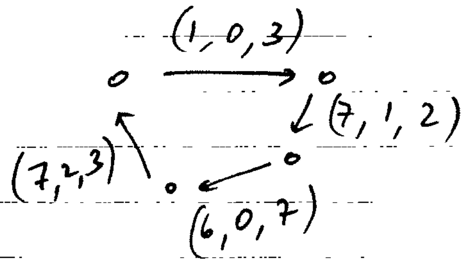
- can think of $b(a)$ as the imbalance between in-flow & out-flow at node a.
    - this is also the supply / demand at node a.

Let's look at some intuitive examples:

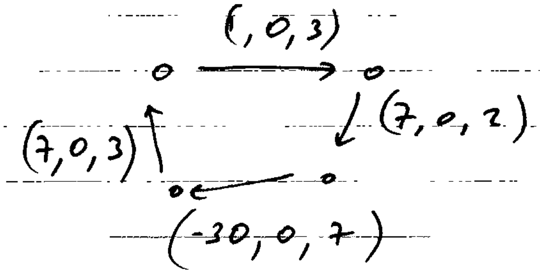


(cost, lower, upper)

intuitive solution?



intuitive solution?



intuitive solution?

$$\circ \xrightarrow[(-30,0,7)]{} \circ \xrightarrow[(7,0,2)]{} \circ$$

intuitive solution?

$$\circ \xrightarrow[(7,1,2)]{} \circ \xrightarrow[(6,0,7)]{} \circ$$

feasible?

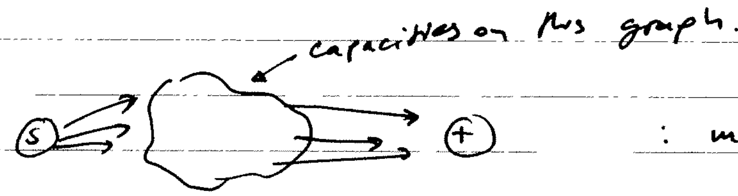
- A number of things can drive the flow on the graph
  - supply / demand we have to clear
  - negative costs we want to take advantage of
  - lower bounds we have to satisfy

- Is shortest path a special case of min-cost flow?

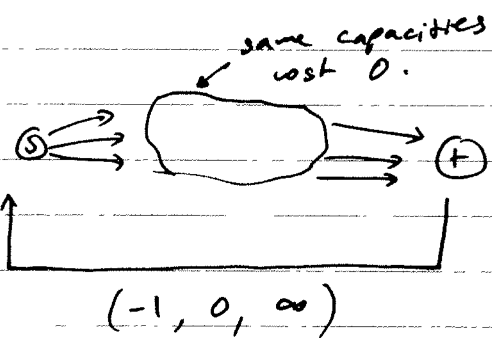
Yes, just make the $b(a)$ -1 for s
1 for t and 0 everywhere else.

- Is max-flow a special case of min-cost flow?

Yes, it turns out, but maybe it is not as easy to see.



: max-flow problem.



: equivalent min-cost flow

- to take advantage of the -1 cost on the arc (t,s) we have to route flow through the original graph.
  - we want to put as much flow on the (t,s) arc as possible.

- we can also see that max-flow is a special case of min-cost flow by looking at the LPs.

Let's do the -1 trick here

$$\max \quad v$$

s.t.

$$\sum_{is} y_{is} - \sum_{si} y_{si} = -v$$ Lets move these to the other side.

$$\sum_{it} y_{it} - \sum_{ti} y_{ti} = v$$

$$\sum_{ia} y_{ia} - \sum_{ai} y_{ai} = 0$$

$$y_{ij} \le u_{ij}$$

$$0 \le y_{ij}$$

v has a cost of -1.

$$\parallel$$

$$-\min_y \quad -v$$

s.t. $$\left(\sum_{is} y_{is} + v\right) - \sum y_{si} = 0$$

v appears where a y_{ts} variable would appear but it has no capacity.

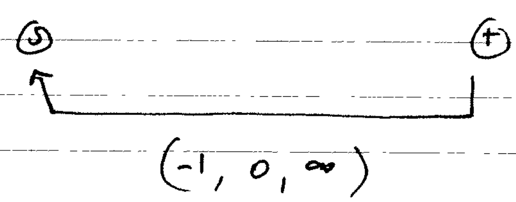
$$\sum_{it} y_{it} - \left(\sum_{ti} y_{ti} + v\right) = 0$$

$$\sum y_{ia} - \sum y_{ai} = 0$$

$$y_{ij} \le u_{ij}$$

$$0 \le y_{ij}$$

So, these manipulations tell us exactly about this arc.

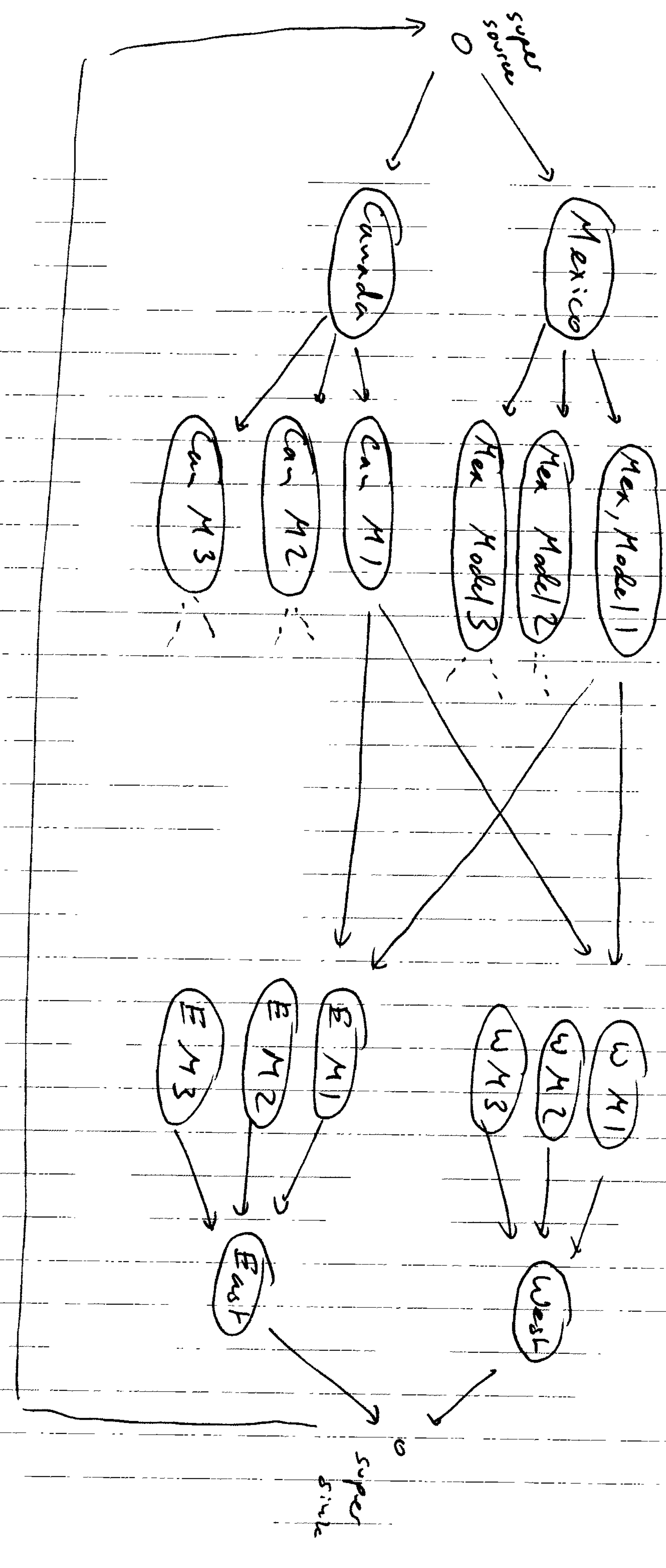


Applications of Min-Cost Flow

Production & Transportation Planning (AMO 9.1)

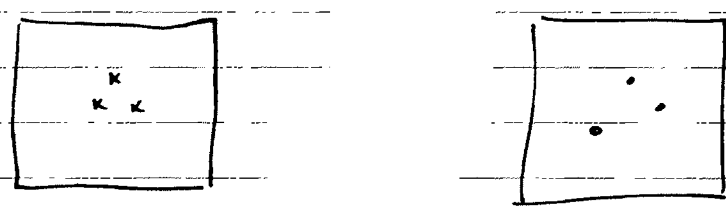
- Suppose we are operating a car manufacturing company
- we own 2 production plants, one in Mexico the other in Canada
- we make three car models, lets suppose each plant can make every model
- we have two retail centers, one on the east coast and one on the west coast
- each retail center has demands based on what the local customers want

Q: How many cars should each plant make of each model and how should we transport them to our retail centers?


super source
Mexico
Canada
Mex, Model 1
Mex Model 2
Mex Model 3
Can M1
Can M2
Can M3
W M1
W M2
W M3
E M1
E M2
E M3
West
East
super sink

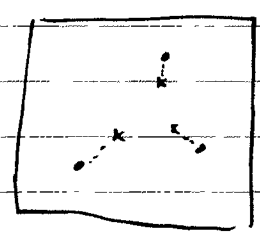
Tracking Multiple Moving Objects

- Suppose we'd like to have a computer track multiple moving objects using a video camera
- We have two consecutive frames and we want to figure out where objects moved.

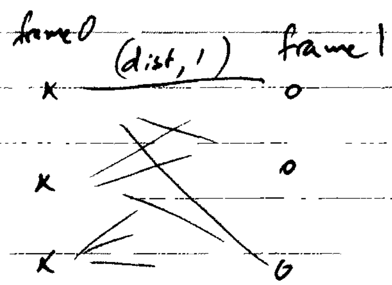


Frame 0 Frame 1

overlay:



← correct matchings



- $b(i)$'s ?
- three different networks ?
- Min-cost flow

finds the minimum distance matching.

Maximizing Water Taxi Revenue (AMO 9.4)

- we run a water taxi that has the following route every day:
  - port (1)
  - port (2)
  - port (3)
  - port (4)

- the boat can hold at most p passengers

- there are b_{ij} passengers that want to be picked up at port (i) and dropped off at port (j)

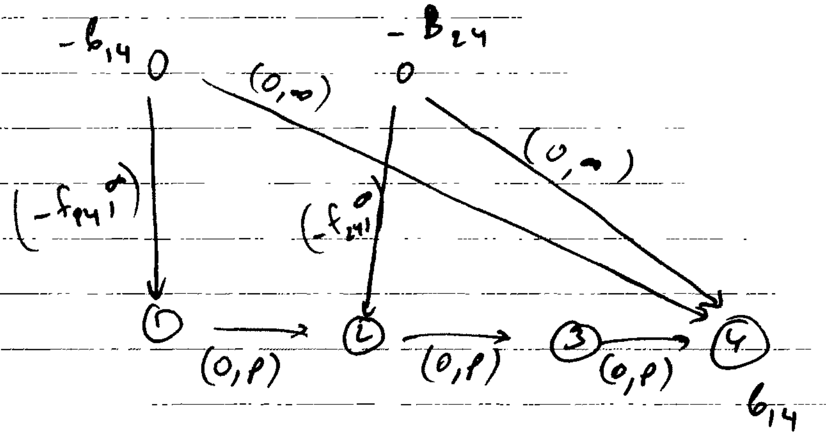
- passengers from (i) to (j) are willing to pay f_{ij}

- who should we pick up to maximize our revenue?

- you might think... let's pick up as many of the highest fare as possible, then fill in the remaining seats with others.

  - why doesn't this work?

- What does work:



- every port is a node. edges of capacity p and cost 0 between consecutive ports

- every (start port, dest port) pair is a node. with supply equal to the # of passengers that want that transit. so, (i,j) is a node with supply b_{ij}.

- add demand b_{ij} at j.

- add arcs from node $[(i,j)]$ to i with cost $-f_{ij}$, representing payment of those passengers.

- add arcs from node $[(i,j)]$ to j with cost 0.. makes things feasible & represents "alternate" travel for passengers we don't pick up.