HW 6 Problem 2:

A transportation problem:

Plant ↓	DC 1	2	3	4	Supply ↓
1	800	1300	400	700	12
2	1100	1400	600	1000	17
3	600	1200	800	900	11
Demand →	10	10	10	10	

Here, supply = demand, so a solution exists (see pyomo code example)

Now: suppose we can increase the supply in one of the plants by 1 unit, and thus the demand in one of the DCs by one unit. Which ones should we choose?

Look at shadow prices: $y_{plant\,1} = -100$, $y_{plant\,2} = 0$, $y_{plant\,3} = -100$,
$y_{DC1} = 500$, $y_{DC2} = 800$, $y_{DC3} = 400$, $y_{DC4} = 550$

$\Rightarrow$ Increase in plants 1 or 3 and ship to DC 3. Here, the cost increase will be the lowest, namely $-100 + 400 = 300$

Now: Demand in Center 1 goes up to 15 units, but production cannot be increased, so some centers will be undersupplied.

Problem: We need total supply = total demand for solution to exist.

Solution: Introduce "dummy source" with 0 associated shipping cost.

Plant ↓	DC 1	2	3	4	Supply ↓
1	800	1300	400	700	12
2	1100	1400	600	1000	17
3	600	1200	800	900	11
4	0	0	0	0	5
Demand →	15	10	10	10	

New solution: see pyomo code (DC2 will be undersupplied).

We continue our discussion of network optimization problems

- <u>Minimum Spanning Tree problem:</u>

Find a path that connects each pair of nodes, with minimal cost (say, cost is proportional to distance here).

no cycle

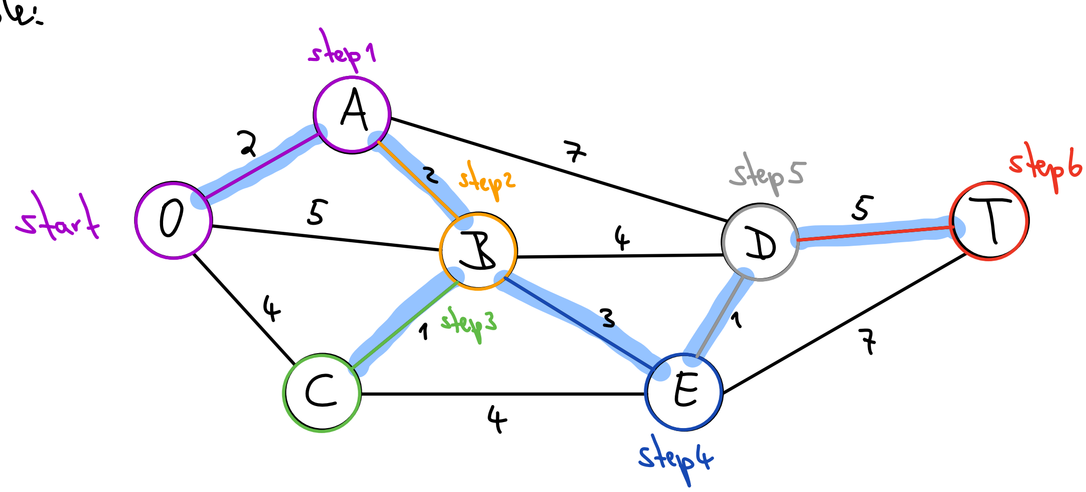
Note: for n nodes we need to find $n-1$ links, i.e., a tree connecting all nodes

otherwise we could always delete a link to make cost smaller

Algorithm: • start with any node, add link to nearest node

• connect linked notes to next nearest node; repeat

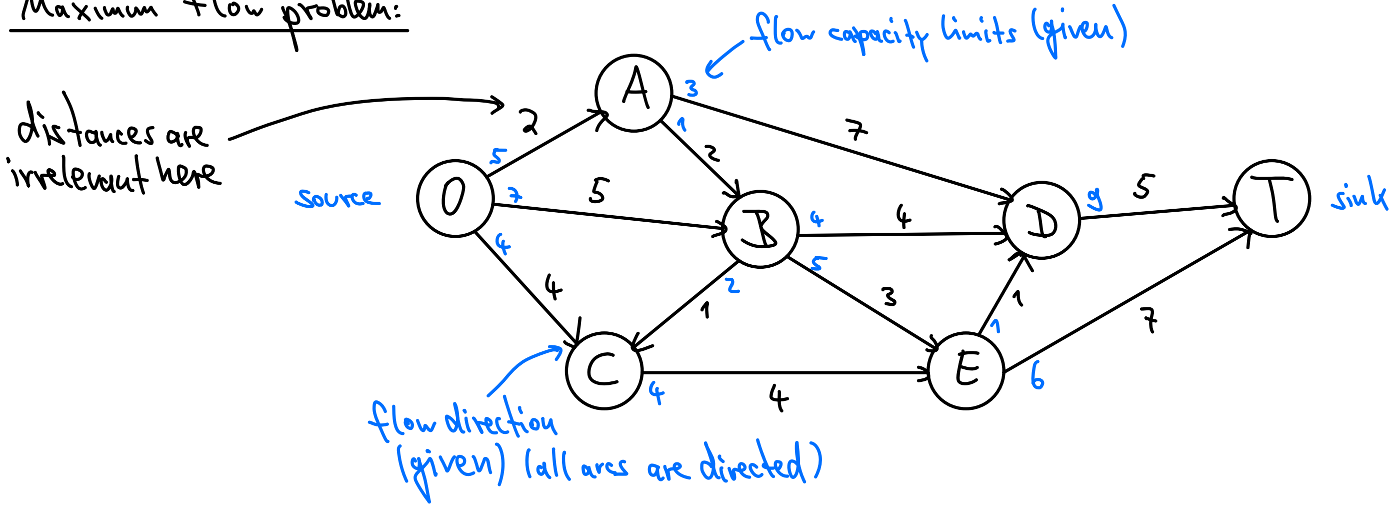
In our example:



$\Rightarrow$ Total cost $= 2+2+1+3+1+5 = 14$

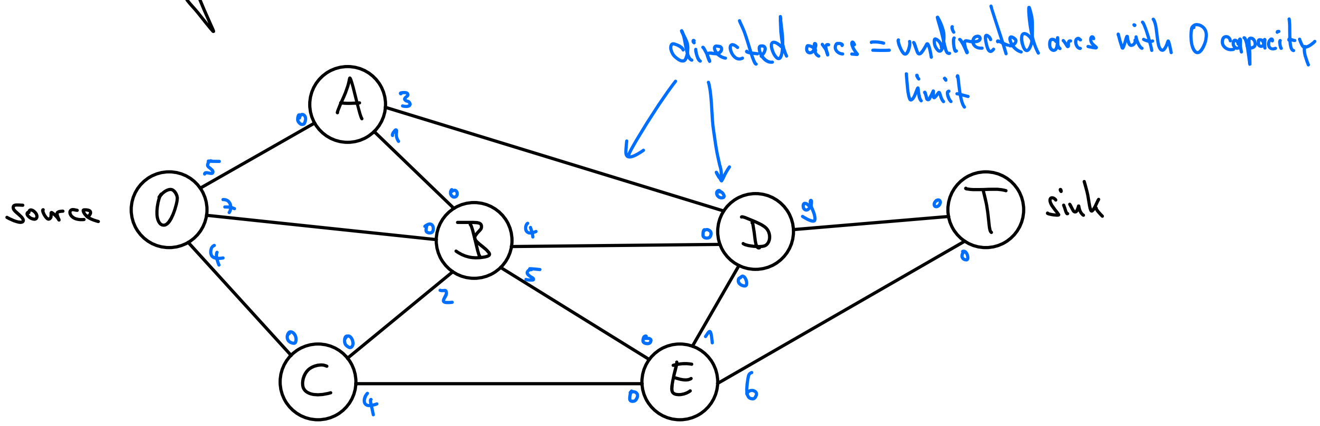
Another problem type is the following:

- <u>Maximum Flow problem:</u>



Objective: maximize flow from source (0) to sink (T), obeying capacity limits.

Augmented Path algorithm: Draw network as



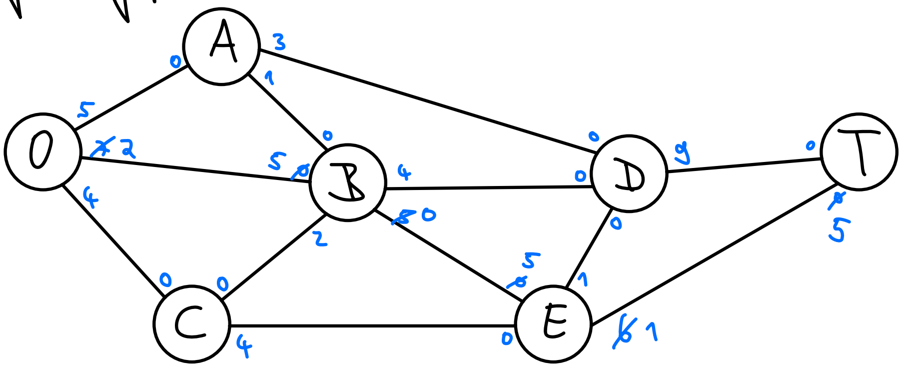
Augmenting path: directed path from source to sink s.t. every arc has strictly positive capacity

in picture: smallest possible number at beginning of arcs

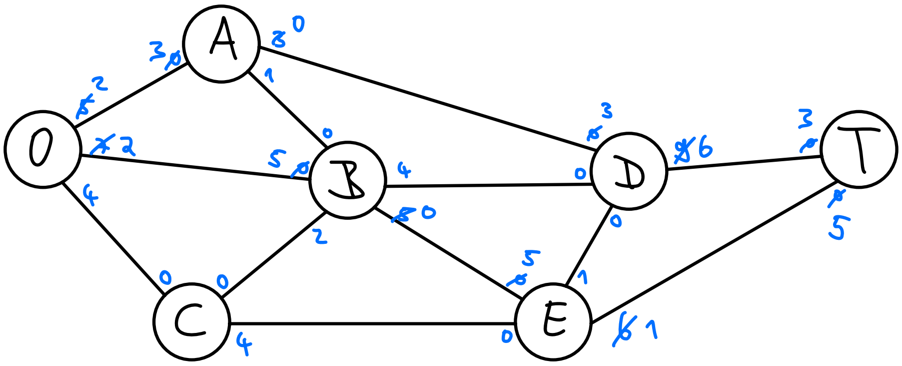
Now: • choose an augmenting path
- increase flow by residual capacity = minimum of all capacities along path
- change capacity limits accordingly
- repeat until no augmenting path can be chosen anymore

For our example:

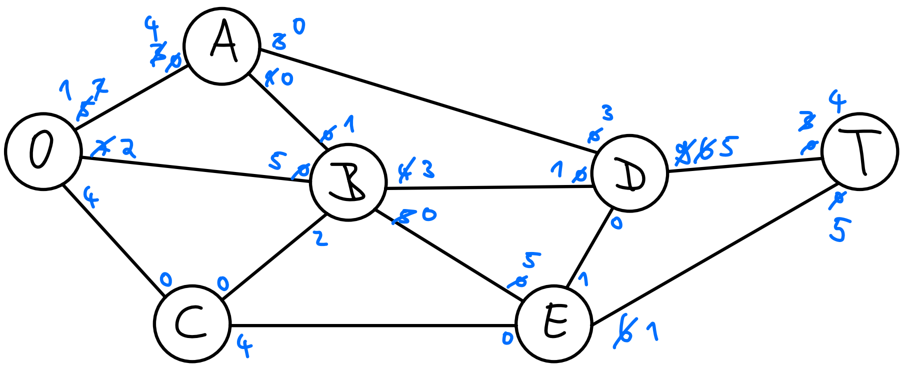
A possible augmenting path is 0-B-E-T: residual capacity: 5 (B-E)



arbitrary next choice: 0-A-D-T: res. cap.: 3 (A-D)



0-A-B-D-T: res. cap.: 1 (A-B)

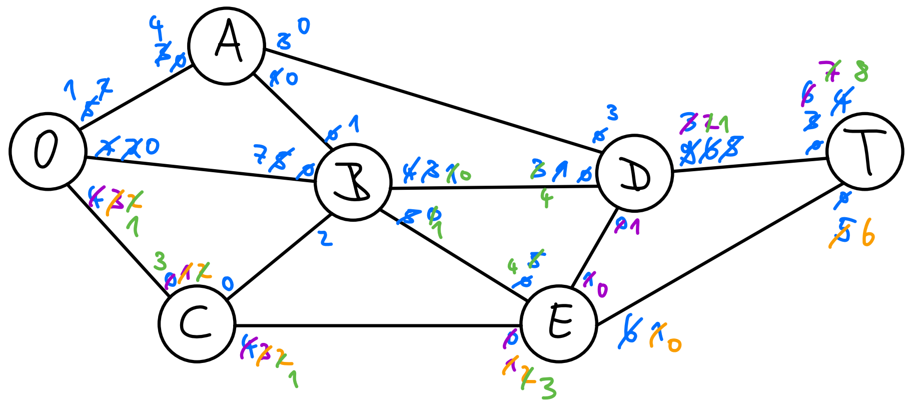


0-B-D-T: res. cap.: 2 (0-B)

0-C-E-D-T: res. cap.: 1 (E-D)

0-C-E-T: res. cap.: 1 (E-T)

0-C-E-B-D-T: res. cap.: 1 (B-D)



$\Longrightarrow$ No more augmenting paths, we have found an optimal solution: $8+6=14$ trips can be made from O to T (more details can be read off from final picture).

Next time: Minimum Cost Flow problems

↳ Def. of model

↳ Write it as LP problem

↳ Formulate all 3 previous problem types as min. cost flow problems