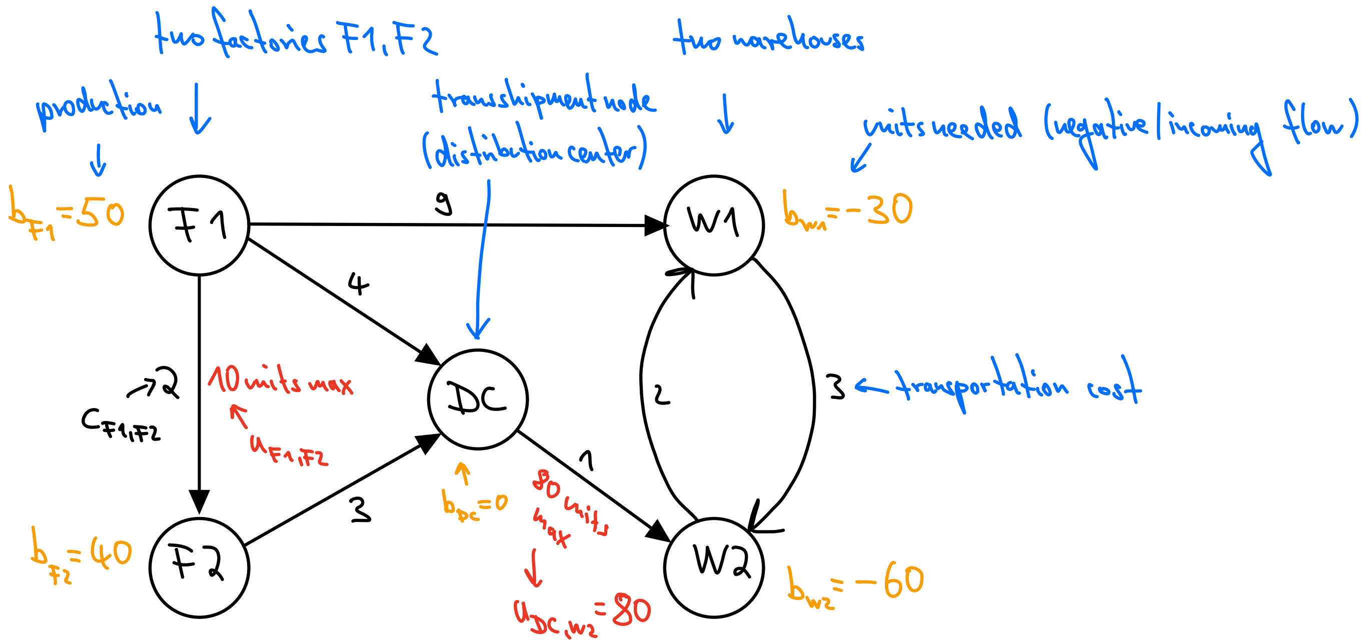
Last time we started discussing $\boxed{\text{minimum cost flow problems}}$.

Example (Hillier, Lieberman Chapters 3.4 and 9.6):



Set of nodes $N=\{F1,F2,DC,W1,W2\}$

Set of arcs $A=\{(F1,F2),(F1,DC),(F1,W1),(F2,DC),(DC,W2),(W1,W2),(W2,W1)\}$

General formulation:
- nodes $i\in N$
- directed arcs $(i,j)\in A$
- c_{ij}: unit cost of transportation on arc (i,j)
- u_{ij}: max. capacity on arc (i,j)
- node constraints
  - $b_i>0$ for supply / source nodes
  - $b_i<0$ for demand / sink nodes
  - $b_i=0$ for transshipment nodes
- x_{ij}: flow from i to j (decision variables)

LP formulation: • Minimize cost $Z = \sum_{(i,j) \in A} c_{ij} x_{ij}$

• Constraints: $\underbrace{\sum_j x_{ij}}_{\text{outgoing flow at node } i} - \underbrace{\sum_j x_{ji}}_{\text{incoming flow at node } i} = b_i$ for all nodes $i \in N$

and $0 \leq x_{ij} \leq u_{ij}$ for all arcs $(i,j) \in A$.

Note: Similarly as discussed before:

- One can show that a necessary condition for feasible solutions is $\sum_i b_i = 0$ (supply = demand). This can always be achieved by introducing dummy nodes (similarly as we discussed before).
- All basic variables in all basic feasible solutions are integer, if all b_i and u_{ij} are integer.
- A faster network simplex method is available.

How can our previous cases be formulated as min. cost flow problems?

- Transportation problem: – only supply and demand nodes (no transshipment nodes), all supply nodes connected to all demand nodes
  – all $u_{ij} = \infty$ since no upper bound constraints

- Shortest Path problem: – origin = supply node with $b_o = 1$
  – destination = demand node with $b_T = -1$
  – other nodes are transshipment, i.e., $b_i = 0$.
  – draw all arcs in both directions (except source/sink)
  – all $u_{ij} = \infty$
  – c_{ij} = distances as given

- Max Flow problem: – all $c_{ij} = 0$ larger than a good guess for the max. flow given the u_{ij}
  - source $b_0 = F$ large, sink $b_T = -F$, all other nodes $b_i = 0$
  - u_{ij} as given
  - extra arc from source to sink with $c_{0T} = M$ very large (and $u_{0T} = \infty$)
    ↳ then $c_{ij} = 0$ arcs are preferred, rest is sent through c_{0T} arc at high cost

Another example: Project Management (extra material)

Problem type 1:

- Given: set of activities taking time T_i to complete, and their dependencies (e.g., building construction)
- Goal: find minimal time to completion, and the corresponding order of activities (= critical path through network)
- Set up: – decision variables t_i = starting time of activity i
  - minimize t_{finish}
  - constraints: $t_j \geq t_i + T_i$ if j depends on i

    $t_{start} = 0$, $t_i \geq 0$

Problem type 2:

- Suppose a completion time is prescribed, but it is shorter than the critical path from above. Assume we can reduce the times of certain activities at a cost (this is called "crashing" an activity).
- Introduce x_i = units of time saved on activity i (decision variables)

  T_i = regular time for completion

  R_i = maximal time that can be saved

  c_i = cost of saving one unit of time

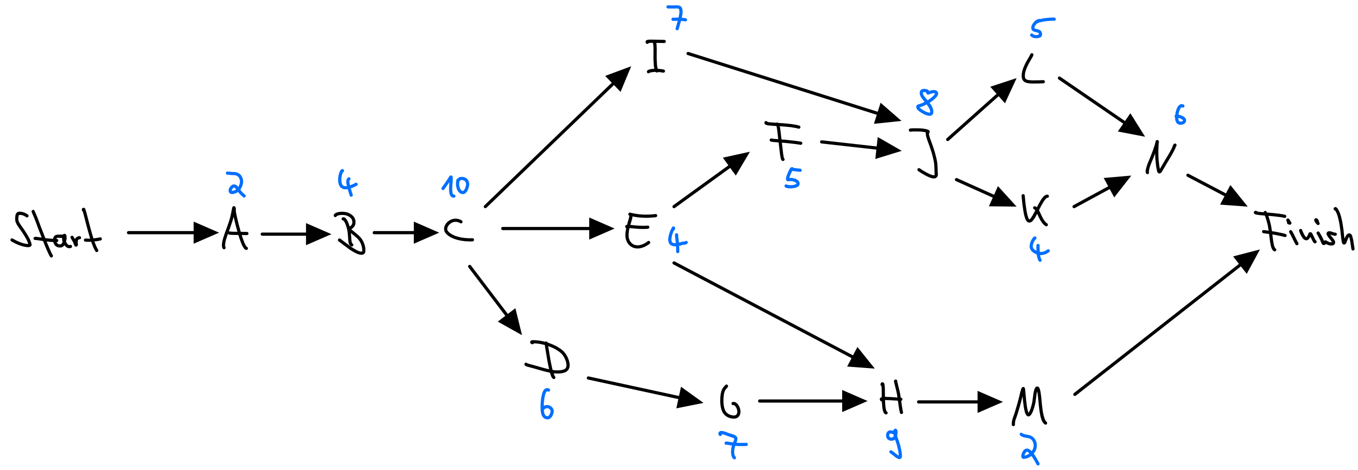
- LP problem: minimize cost $\sum_i c_i x_i$

  subject to $x_i \le R_i$ for all activities i

  $t_j \ge t_i + (T_i - x_i)$

  $t_i \ge 0, \; x_i \ge 0$

Example (Hillier, Liebermann Chapter 9.8 (9th edition)): Reliable Construction Company



Critical path = longest path = A-B-C-E-F-J-L-N = 44 weeks

so all activities can be finished

Suppose project needs to be completed in 40 weeks, i.e., we need to crash some activities $\longrightarrow$ see promo code discussion.

Some possible exam topics/questions:

- Formulate a given "text problem" as LP
- Solve LP problem graphically (also: shape of feasible region, number of solutions)
- Write LP problem in standard form
- Gaussian elimination and basic solutions
- Use simplex method to solve LP problem (what if feasible region is unbounded?)
- Shadow prices and their meaning
- Dual LP problems, weak and strong duality
- Transportation problems and their LP formulation
- Integer solution property, dummy variables
- Solve shortest path, minimum spanning tree, maximum flow problems
- Minimum cost flow problem and LP
- Pyomo: explain code; explain output; extract LP problem in mathematical notation from code; what happens if something is changed in the code

Good practice midterm: Fall 2020 (see website)