CS/ECE/ISYE524: Introduction to Optimization – Linear Optimization Models

Jeff Linderoth

Department of Industrial and Systems Engineering
University of Wisconsin-Madison

February 11, 2024

Example: Sailco

Sailco manufactures sailboats. During the next 4 months the company must meet the following demands for their sailboats:

Month	1	2	3	4
Number of boats	40	60	70	25

At the beginning of Month 1, Sailco has 10 boats in inventory. Each month it must determine how many boats to produce. During any month, Sailco can produce up to 40 boats with regular labor and an unlimited number of boats with overtime labor. Boats produced with regular labor cost $400 each to produce, while boats produced with overtime labor cost $450 each. It costs $20 to hold a boat in inventory from one month to the next. Find the production and inventory schedule that minimizes the cost of meeting the next 4 months' demands.

Example: Sailco

Summary of problem data:

- Regular labor: \$400/boat (at most 40 boats/month).
- Overtime labor: \$450/boat (no monthly limit).
- Holding a boat in inventory costs \$20/month.
- Inventory initially has 10 boats.
- Demand for next 4 months is:

Month	1	2	3	4
Number of boats	40	60	70	25

What are the decision variables?

Example: Sailco

Remember: Decision variables aren't always things that you decide directly!

For this problem, the decision variables are:

- x_1, x_2, x_3, x_4: boats produced each month with regular labor.
- y_1, y_2, y_3, y_4: boats produced each month with overtime.
- h_1, h_2, h_3, h_4, h_5: boats in inventory at start of each month.

Parameters:

- d_1, d_2, d_3, d_4: demand at each month

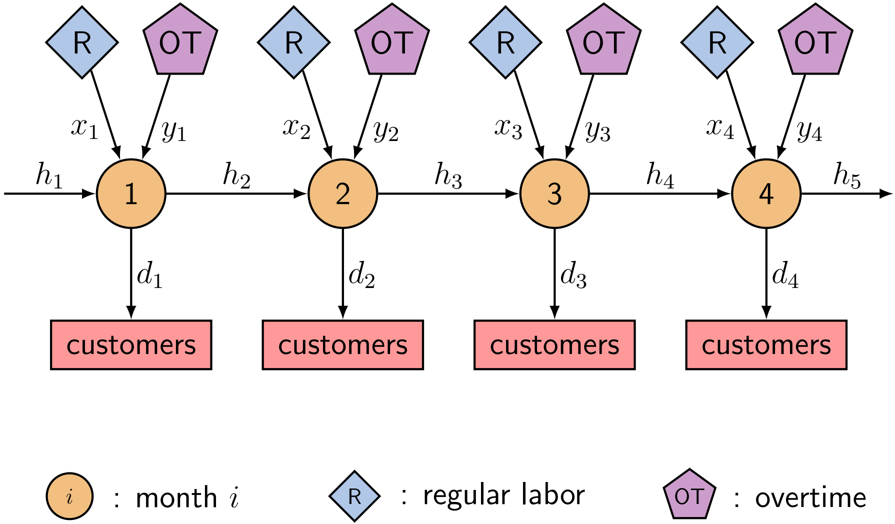
Example: Sailco

The constraints are:

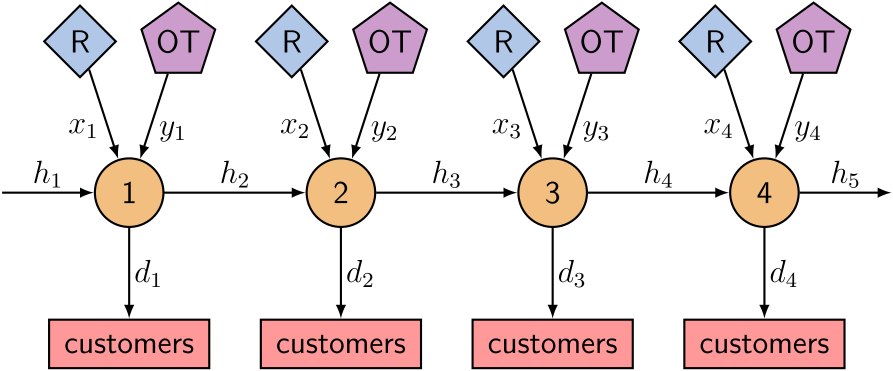
- $0 \leq x_i \leq 40$ (monthly limit of regular production)
- $y_i \geq 0$ (unlimited overtime production)
- Conservation of boats:
    - $h_i + x_i + y_i = d_i + h_{i+1}$ (for $i = 1, 2, 3, 4$)
    - $h_1 = 10$ (initial inventory)

Solution: Sailco.ipynb

Example: Sailco



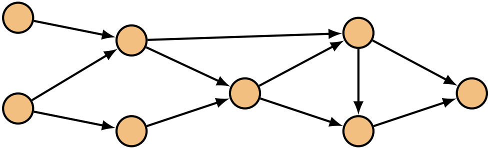
Example: Sailco



- Arrows indicate flow of boats
- conservation at nodes: $h_1 + x_1 + y_1 = d_1 + h_2$, etc.

Minimum-cost flow problems

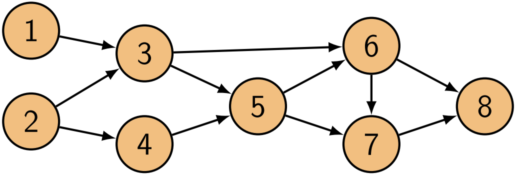
- Many optimization problems can be interpreted as **network flow problems** on a directed graph.



- Decision variables: **flow on each edge**.
- Edges have flow costs and capacity constraints
- Each node can:
  - produce/supply flow (source)
  - consume/demand flow (sink)
  - conserve flow (relay)

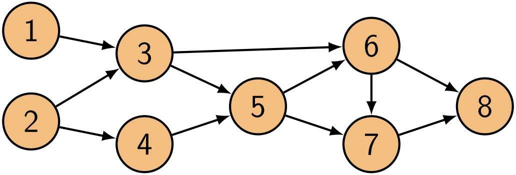
What is the minimum-cost feasible flow?

Minimum-cost flow problems



- The set of nodes: $\mathcal{N} = \{1, \ldots, 8\}$.
- The set of directed edges: $\mathcal{E} = \{(1,3), (2,3), (2,4), \ldots\}$.
- Each node $i \in \mathcal{N}$ supplies a flow b_i. Node i is called a *source* if $b_i > 0$, a *relay* if $b_i = 0$, and a *sink* if $b_i < 0$.
- **Decision variables**: x_{ij} is the flow on edge $(i,j) \in \mathcal{E}$.
- **Flow cost**: c_{ij} is cost per unit of flow on edge $(i,j) \in \mathcal{E}$.

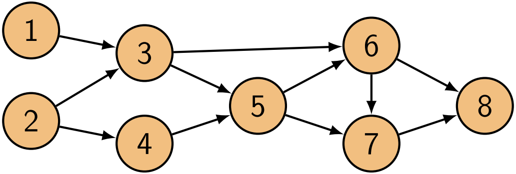
Minimum-cost flow problems



- **Decision variables**: x_{ij} is the flow on edge $(i,j) \in \mathcal{E}$.
- **Capacity constraints**: $p_{ij} \le x_{ij} \le q_{ij} \qquad \forall (i,j) \in \mathcal{E}$.
- **Conservation**: $\sum_{j \in \mathcal{N}} x_{kj} - \sum_{i \in \mathcal{N}} x_{ik} = b_k \qquad \forall k \in \mathcal{N}$.
- **Total cost**: $\sum_{(i,j) \in \mathcal{E}} c_{ij} x_{ij}$.

Note: $b_k, c_{ij}, p_{ij}, q_{ij}$ are *parameters*.

Minimum-cost flow problems



$$\underset{x_{ij} \in \mathbb{R}}{\text{minimize}} \quad \sum_{(i,j)\in\mathcal{E}} c_{ij} x_{ij}$$

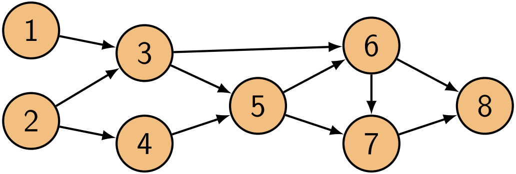
$$\text{subject to:} \quad \sum_{j\in\mathcal{N}:(k,j)\in\mathcal{E}} x_{kj} - \sum_{i\in\mathcal{N}:(i,k)\in\mathcal{E}} x_{ik} = b_k \quad \forall k \in \mathcal{N}$$

$$p_{ij} \le x_{ij} \le q_{ij}$$

"Flow Balance" Mantra

"Out" - "In" = "Net Supply"

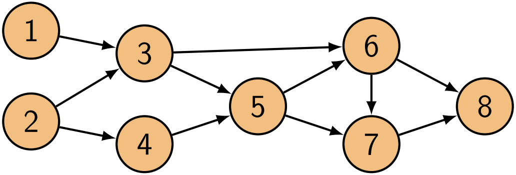
Minimum-cost flow problems



Expanded conservation constraint:

$$\underbrace{\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & -1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & -1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 & -1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & -1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & -1 \end{bmatrix}}_{A \,=\, \text{incidence matrix}} \begin{bmatrix} x_{13} \\ x_{23} \\ x_{24} \\ x_{35} \\ x_{36} \\ x_{45} \\ x_{56} \\ x_{57} \\ x_{67} \\ x_{68} \\ x_{78} \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \\ b_4 \\ b_5 \\ b_6 \\ b_7 \\ b_8 \end{bmatrix}$$

Minimum-cost flow problems



The entire model (compact form):

$$
\begin{aligned}
\underset{x \in \mathbb{R}^{|\mathcal{E}|}}{\text{minimize}} \quad & c^{\mathsf{T}} x \\
\text{subject to:} \quad & Ax = b \\
& p \leq x \leq q
\end{aligned}
$$

Note: The incidence matrix A is a property of the graph.
It does not depend on which nodes are sources/sinks/relays.

Balanced problems

The incidence matrix has the property that all columns sum to zero: $1^\mathsf{T} A = 0$.

This is because each column corresponds to a singleedge, which has one origin node (-1) and one destination $(+1)$.

Since $Ax = b$ is a constraint, we must therefore have: $1^\mathsf{T} Ax = 1^\mathsf{T} b = 0$. Therefore:

$$\sum_{i \in \mathcal{N}} b_i = 0 \quad \text{(total supply = total demand)}$$

- If $\sum_{i \in \mathcal{N}} b_i = 0$, the model is called **balanced**.
- Unbalanced models are *always* infeasible.
- Note: balanced models may still be infeasible.

Balanced problems

Unbalanced models still make sense in practice, e.g. we may have excess supply or allow excess demand. These cases can be handled by making small modifications to the problem, such as changing "$=$" to "$\leq$".

Minimum-cost flow problems

Many problem types are actually min-cost flow models:

- transportation problems
- assignment problems
- transshipment problems
- shortest path problems
- max-flow problems

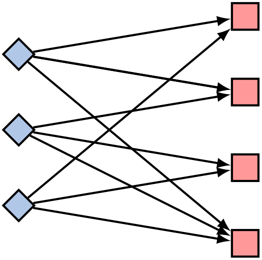
Let's look at these in more detail...

Legend: $\Diamond$: source $\bigcirc$: relay $\square$: sink

Transportation problems

The objective is to transport a particular commodity from several possible sources to several possible destinations while minimizing the total cost.

- Sources have known supply limits
- Destinations each have demands
- Edges may have capacity limits
- Each link has an associated cost



Transportation example

Millco has three wood mills and is planning three new logging sites. Each mill has a maximum capacity and each logging site can harvest a certain number of truckloads of lumber per day. The cost of a haul is \$2/mile of distance. If distances from logging sites to mills are given below, how should the hauls be routed to minimize hauling costs while meeting all demands?

Logging site	Distance to mill (miles) Mill A	Mill B	Mill C	Maximum truckloads/day per logging site
1	8	15	50	20
2	10	17	20	30
3	30	26	15	45
Mill demand (truckloads/day)	30	35	30	

Note: problem is balanced!

Source: J. Reeb and S. Leavengood, 2002

Transportation example

- Arrange nodes as: $\begin{bmatrix} 1 & 2 & 3 & A & B & C \end{bmatrix}$ (sources, sinks).
- Graph is fully connected. Incidence matrix:

$$A = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ -1 & 0 & 0 & -1 & 0 & 0 & -1 & 0 & 0 \\ 0 & -1 & 0 & 0 & -1 & 0 & 0 & -1 & 0 \\ 0 & 0 & -1 & 0 & 0 & -1 & 0 & 0 & -1 \end{bmatrix}$$

Julia code: Millco.ipynb

Transportation example

Logging site	Distance to mill (miles) Mill A	Mill B	Mill C	Maximum truckloads/day per logging site
1	8	15	50	20
2	10	17	20	30
3	30	26	15	45
Mill demand (truckloads/day)	30	35	30	

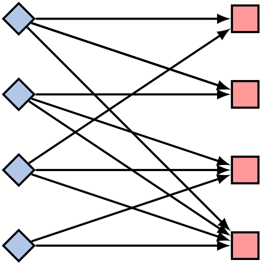
Solution is:

	A	**B**	**C**	
1	20	0	0	20
2	10	20	0	30
3	0	15	30	45
	30	35	30	

Assignment problems

We have n people and n tasks. The goal is to assign each person to a task. Each person has different preferences (costs) associated with performing each of the tasks. The goal is to find an assignment that minimizes the total cost.

- It's just a transportation problem!
- Each source has supply $= 1$
- Each sink has demand $= 1$
- Edges are unconstrained



What about the integer constraint? More about this later...

Assignment example

The coach of a swim team needs to assign swimmers to a 200-yard medley relay team to compete in a tournament. The problem is that his best swimmers are good in more than one stroke, so it's not clear which swimmer to assign to which stroke. Here are the best times for each swimmer:

Stroke	Carl	Chris	David	Tony	Ken
Backstroke	37.7	32.9	33.8	37.0	35.4
Breaststroke	43.4	33.1	42.2	34.7	41.8
Butterfly	33.3	28.5	38.9	30.4	33.6
Freestyle	29.2	26.4	29.6	28.5	31.1

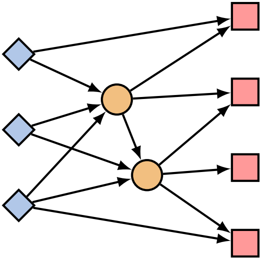
Julia code: Swim Relay.ipynb

Source: B. Van Roy and K. Mason

Transshipment problems

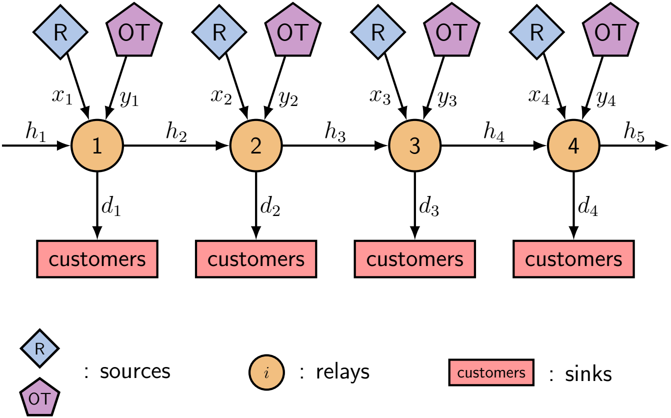
The same as a transportation problem, but in addition to sources and destinations, we also have warehouses that can store goods. The warehouses are **relay nodes**.

- Sources have known supply limits
- Destinations each have demands
- Links may have capacity limits
- Each link has an associated cost
- For warehouses, inflow = outflow.



Sailco problem is a transshipment problem!

Transshipment example: Sailco

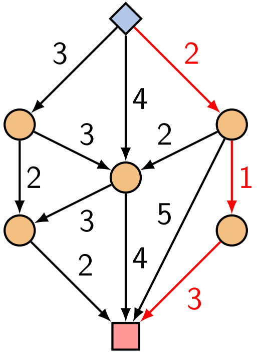


- The "warehouses" are the different months.
- Storing in inventory = shipping to the future.

Shortest/longest path problems

We have a directed graph and edge lengths. The goal is to find the shortest or longest path between two given nodes.

- Again, a transportation problem!
- Edge cost $=$ length of path.
- The source has supply $= 1$
- The sink has demand $= 1$
- To find longest path, just change the min to a max!
- Only works for max if graph does not have a loop/cycle

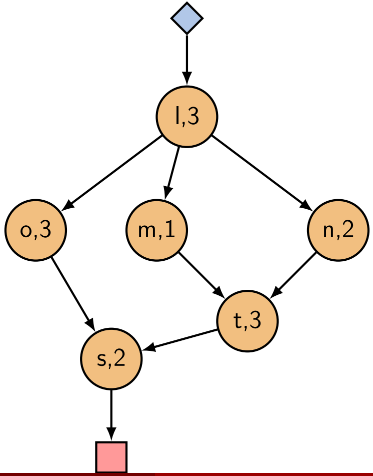


Again we need integer constraints on the edges...

Longest path example

The house building example is a longest path problem!

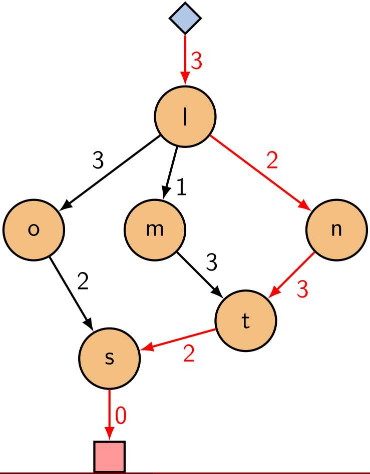
- Add source and sink nodes
- Move times out of nodes and onto preceding edges



Longest path example

The house building example is a longest path problem!

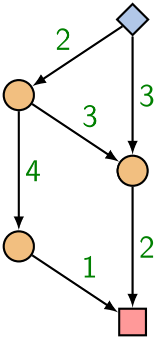
- Add source and sink nodes
- Move times out of nodes and onto preceding edges. The cost on the edge now means "it takes this long to finish the task at the destination node."
- Each path says "it takes at least this long" Longest path gives the shortest time we have to wait.



Max-flow problems

We are given a directed graph and **edge capacities**. Find the maximum flow that we can push from source to sink.

- Edges have max capacities
- Flow can split!
- notions of supply and demand don't make sense...
- add a feedback path and make every node a relay!

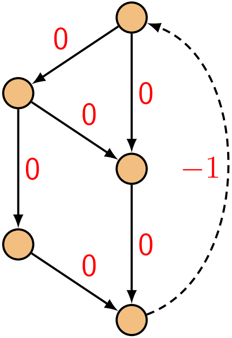


(edge capacity)

Max-flow problems

We are given a directed graph and **edge capacities**. Find the maximum flow that we can push from source to sink.

- Edges have max capacities
- Flow can split!
- notions of supply and demand don't make sense...
- add a feedback path and make every node a relay!



(cost)

Solve minimum-cost flow where feedback path has cost (-1) and all other paths have zero cost.

Max Flow

Maximum Flow Problem

Given a capacitated network $G = (N, A)$, with capacities $u \in \mathbb{R}_+^{|A|}$, a source node $s \in N$, and a sink node $t \in V$, what is the maximum flow that can be sent from s to t.

- Model “what goes in = what come out”.

$$\sum_{j:(i,j)\in A} x_{ij} - \sum_{j:(j,i)\in A} x_{ji} = 0$$

- Be sure to add an arc from $t \to s$ in A

Max Flow Problem (as MCNF)

$$-\min -x_{ts}$$

$$\begin{aligned}
\text{s.t.} \sum_{j:(i,j)\in A} x_{ij} - \sum_{j:(j,i)\in A} x_{ji} &= 0 \qquad \forall i \in N \\
x_{ij} &\leq u_{ij} \qquad \forall (i,j) \in A \\
x_{ij} &\geq 0 \qquad \forall (i,j) \in A
\end{aligned}$$

- Note that due to flow balance, objective is the same as

$$\sum_{j:(j,t)\in A} x_{jt}$$

Let's Have a Picnic!

- The Hatfields, Montagues, McCoys and Capulets are going on their annual family picnic.
- Four cars are available to transport the families to the picnic.
- The cars can carry the following numbers of people: car 1, 4; car 2, 3; car 3, 3; car 4, 4.
- There are four people in each family, and no car can carry more than two people from any one family.
- Determine the maximum number of people that can be transported to the picnic.

Max Flow Problem

- Let's try and model this is a max flow problem

The $0.0001 Question

- Who Can Make This A Max Flow Problem?

Picnic.ipynb

Integer solutions

Some minimum-cost flow problems require integer solutions (assignment problems and shortest path problems). Is there a way of guaranteeing integer solutions? **yes!**

Definition: A matrix A is *totally unimodular* (TU) if every square submatrix of A has determinant 0, 1, or -1.

- The definition includes 1×1 submatrices, so every entry of A must be 0, 1, or -1.
- ex. $\begin{bmatrix} 1 & 1 & 0 \\ -1 & 0 & 1 \end{bmatrix}$ is TU but $\begin{bmatrix} 1 & 1 & 1 \\ -1 & 0 & 1 \end{bmatrix}$ is not.

Integer solutions

Theorem: If A is TU and b is an integer vector, then the vertices of $\{x \mid Ax \leq b\}$ have integer coordinates.

Theorem: Every incidence matrix is TU.

What does this mean? (**a lot!**)

- If a minimum-cost flow problem has integer supplies, integer demands, and integer edge capacities, then there is a minimum-cost **integer** flow.
- every assignment problem is an LP.
- every shortest path problem is an LP.

Next class...

- Duality theory
- Shadow price interpretation
- Sensitivity analysis