Recap, and outline of Lecture 21

Previously
- Linear programming sensitivity analysis
- Decomposition methods for large-scale LPs
  - Column generation methods
  - Constraint generation (cutting plane) methods
- Graphs
  - Undirected and directed graphs: basic definitions

Today
- Networks
- NFP definition and AMPL modeling tools
- Basic solutions and feasible directions in NFP

Networks

A network consists of
- a directed graph \( G = (\mathcal{N}, \mathcal{A}) \)
- \( b_i \in \mathbb{R}, \ i \in \mathcal{N} \) — supplies at nodes
  - if \( b_i < 0 \), \( i \) is a sink and \(|b_i|\) is the demand,
  - if \( b_i > 0 \), \( i \) is a source
- \( u_{ij} \in \mathbb{R}_+ \cup \infty, \ (i,j) \in \mathcal{A} \) — capacities of arcs
  - If all \( u_{ij} = \infty \), \( G \) is uncapacitated
  - Also, possibly, \( l_{ij} \geq 0 \) \((i,j) \in \mathcal{A}\) — minimum flow requirements
- \( c_{ij} \in \mathbb{R}, \ (i,j) \in \mathcal{A} \) — cost per unit of flow on arcs

Motivation behind a network flow problem
- We are interested in finding the cheapest way of shipping the product from the suppliers to the customers, subject to arc capacity limits, if any.
- Nodes represent locations (suppliers of a product, customers, or simply transshipment locations)
- Arcs (with directions) represent available shipping routes
Formulation of the network flow problem (NFP)

- **Variables:** \( f_{ij} \) — amount of “flow” on arc \((i,j)\)
- **Constraints:**
  - Nonnegativity:
    \[ f_{ij} \geq 0, \forall (i,j) \in \mathcal{A} \]
  - Capacities, if any:
    \[ l_{ij} \leq f_{ij} \leq u_{ij}, \forall (i,j) \in \mathcal{A} \]
  - Flow conservation:
    \[ b_i + \sum_{j \in I(i)} f_{ji} = \sum_{j \in O(i)} f_{ij}, \forall i \in \mathcal{N} \]
- Note that \( \sum_{i \in \mathcal{N}} b_i = 0 \) is necessary for feasibility.
- **Objective:**
  \[ \min \sum_{(i,j) \in \mathcal{A}} c_{ij} f_{ij} \]
  over all feasible flows.

**Assumption 7.1**
\[ \sum_{i \in \mathcal{N}} b_i = 0, \text{ and the graph } G \text{ is connected.} \]

A matrix-vector formulation

- Let \( n = |\mathcal{N}| \) and \( m = |\mathcal{A}| \)
  - Note that \( m \) and \( n \) play different (in fact, opposite) roles than what we are used to.
- **Node-arc incidence matrix** \( \mathbf{A} \in \mathbb{R}^{n \times m} \)
  \[
a_{ik} = \begin{cases} 
1 & \text{if } i \text{ is the start node of arc } k, \\
-1 & \text{if } i \text{ is the end node of arc } k, \\
0 & \text{otherwise.}
\end{cases}
\]
- For \( i \in \mathcal{N}, a_i^t \mathbf{f} = \sum_{j \in O(i)} f_{ij} - \sum_{j \in I(i)} f_{ji} \).
- The (uncapacitated) NFP is an LP in standard form:
  \[ \min \mathbf{c}^t \mathbf{f} \text{ s.t. } \mathbf{A} \mathbf{f} = \mathbf{b}, \mathbf{f} \geq \mathbf{0} \]
- Capacitated NFP:
  \[ \min \mathbf{c}^t \mathbf{f} \text{ s.t. } \mathbf{A} \mathbf{f} = \mathbf{b}, \mathbf{l} \leq \mathbf{f} \leq \mathbf{u} \]
Network flow problems: solution approaches

- The (uncapacitated) NFP:

\[ \min \ c'f \ \text{s.t.} \ A f = b, \ f \geq 0 \]

— an LP in standard form

- Capacitated NFP:

\[ \min \ c'f \ \text{s.t.} \ A f = b, \ l \leq f \leq u \]

— an LP in standard form with upper bound constraints (recall Exercises 2.3 and 3.25)

Need to examine:
- full row rank assumption
  - Needs work!
- basic solutions
  - Trees
- feasible directions and basic directions
  - Circulations

Uncapacitated NFP — the full row rank assumption

\[(\text{NFP}) \ \min \ c'f \ \text{s.t.} \ A f = b, \ f \geq 0,\]

where

\[ a_{ik} = \begin{cases} 
1 & \text{if } i \text{ is the start node of arc } k, \\
-1 & \text{if } i \text{ is the end node of arc } k, \\
0 & \text{otherwise.} 
\end{cases} \]

Assumption 7.1

(a) \( \sum_{i \in N} b_i = 0 \); (b) \( G \) is connected.

- Notice: the rows of the incidence matrix \( A \) sum to 0’
- Let the truncated node-arc incidence matrix \( \tilde{A} \) consist of the first \( n - 1 \) rows of \( A \).
  - \( \tilde{b} \) consists of the first \( n - 1 \) elements of \( b \)
- Under Assumption 7.1,
  - \( \tilde{A} f = \tilde{b}, \ f \geq 0 \) has a solution iff \( A f = b, \ f \geq 0 \) does
  - \( \tilde{A} \) has full row rank — will show
Trees (on undirected graphs)
Towards understanding of basic solutions of NFP

**Definition: Tree**

\[ G = (\mathcal{N}, \mathcal{E}) \] is called a tree if it is connected and has no cycles.

- If a node of a tree has degree 1, it is called a leaf.

**Theorem 7.1**

(a) Every tree with \(|\mathcal{N}| > 1\) has at least one leaf

(b) An undirected graph is a tree \(\iff\) it is connected and has \(|\mathcal{N}| - 1\) arcs.

(c) In a tree, there exists a unique path between each pair of distinct nodes.

(d) Adding one arc to a tree creates exactly one cycle. (We do not distinguish between cycles involving the same set of nodes.)

**Proof outline of Theorem 7.1**

(a) Every tree with \(|\mathcal{N}| > 1\) has at least one leaf

- If there are no leaves, then
  - Every node has degree at least 2
  - For each node, can follow one arc in, and another arc out
  - Repeating this process, we find a cycle — contradiction.
Proof outline of Theorem 7.1

(b) Tree ⇔ connected, has $|\mathcal{N}| - 1$ arcs

- Every tree is connected by definition
- Every tree has $|\mathcal{N}| - 1$ arcs:
  - Trivial if $|\mathcal{N}| = 1$
  - If $|\mathcal{N}| > 1$, find and remove a leaf node with its arc. What remains is still at tree; repeat until one node is left.
- Given a connected graph with $|\mathcal{N}| - 1$ arcs,
  - If there is a cycle, can remove one arc and the graph remains connected
  - Repeat this process until no cycles remain — the remaining graph is a tree
  - But a tree must have $|\mathcal{N}| - 1$ arcs — there could not have been any cycles

Proof outline of Theorem 7.1

(c) In a tree, there exists a unique path between each pair of distinct nodes and (d) Adding one arc to a tree creates exactly one cycle

(c) If there are two paths between some two nodes in a tree
  - Join the two paths; delete the arcs that are common to both
  - We are left with one or more cycles — contradiction

(d) Let’s add an arc $\{i,j\}$ to a tree
  - Resulting graph has $|\mathcal{N}|$ arcs — not a tree, so it must have a cycle
  - The created cycle consists of $\{i,j\}$ and the unique path connecting $i$ and $j$ in the tree, so its a unique cycle.