Introduction to Mathematical Programming
IE406

Lecture 16

Dr. Ted Ralphs

Reading for This Lecture

- Bertsimas 7.1-7.3

Network Flow Problems

- *Networks* are used to model systems in which a commodity or commodities must be transported from one or more supply points to one or more demand points along defined pathways.
- These models occur naturally in many contexts.
  - Transportation
  - Logistics
  - Telecommunications
- Network flow problems are defined on *graphs* that define the structure of the pathways in the network.

Undirected Graphs

- An undirected graph $G = (N, E)$ consists of
  - A finite set of *nodes* N representing the supply and demand points.
  - A set E of unordered pairs of nodes called *edges* representing the pathways joining pairs of nodes.
- We say that the edge $\{i, j\}$ is *incident to* nodes i and j and i and j are its *endpoints*.
- The *degree* of a node is the number of edges incident to it.
- The degree of a graph is the maximum of the degrees of its nodes.

Basic Definitions (Undirected)

- A *walk* is a finite sequence of nodes $i_1, \ldots, i_t$ such that $\{i_k, i_{k+1}\} \in E \; \forall k = 1, 2, \ldots, t-1$.
- A walk is called a *path* if it has no repeated nodes.
- A *cycle* is a path with $i_1 = i_t, t > 2$.
- An undirected graph is said to be *connected* if for every pair of nodes i and j, there is a path from i to j.
- For undirected graphs, our convention will be to denote $|N| = n$ and $|E| = m$.

Directed Graphs

- A directed graph $G = (N, A)$ consists of a set of N nodes and a set A of ordered pairs of nodes called *arcs*.
- For any arc (i, j), we say that j is the *head* and i is the *tail*.
- The arc (i, j) is said to be *outgoing* from node i and *incoming* to node j and incident to both i and j.
- We define $I(i)$ and $O(i)$ as

$$I(i) = \{j \in N | (j, i) \in A\}$$

and

$$O(i) = \{j \in N | (i, j) \in A\}$$

Basic Definitions (Directed)

- Corresponding to every directed graph is the *underlying undirected graph* obtained by ignoring the direction of the arcs.
- A walk in a directed graph is a sequence of nodes $i_1, \ldots, i_t$ plus a corresponding sequence of arcs such that $a_1, \ldots, a_{t-1}$ such that for every k, either
  - $a_k = (i_k, i_{k+1})$, in which case a_k is a *forward arc*, or
  - $a_k = (i_{k+1}, i_k)$, in which case a_k is a *backward arc*.
- Again, a walk is a *path* if its nodes are distinct and a cycle is a path in which $i_1 = i_t$.
- A walk, path, or cycle is *directed* if it contains only forward arcs.
- A directed graph is *connected* if the underlying undirected graph is connected.

Trees

- An undirected graph is a tree if it is connected and *acyclic* (has no cycles).
- In a tree, nodes of degree one are called *leaves*.
- Properties of trees
  - Every tree with more than one node has at least one leaf.
  - An undirected graph is a tree if and only if it is connected and has $|N| - 1$ edges.
  - For any two distinct nodes i and j in a tree, there exists a unique path from i to j.
  - If we add an edge to a tree, the resulting graph contains exactly one cycle.

Subgraphs and Spanning Trees

- Given a connected, undirected graph $G = (N, E)$, a graph $G' = (N', E')$ is a *subgraph* of G if $N' \subseteq N$ and $E' \subseteq E$.
- A subgraph $T = (N, E_1)$ that is also a tree is called a *spanning tree* of G.
- Any subset of edges of an undirected, connected graph that does not form any cycles can be extended to form a spanning tree.

Network Flow Problems

- A *network* is a directed graph together with the following additional information:
  - A *supply* b_i associated with each node i (negative supply is interpreted as demand).
  - A nonnegative *capacity* u_{ij} associated with each arc (i, j).
  - A cost c_{ij} for transporting a unit of the commodity from i to j.
- We visualize a network by envisioning the *flow* of some commodity through the network.
- A node i such that $b_i > 0$ is called a *source*.
- A node i such that $b_i < 0$ is called a *sink*.
- A node i such that $b_i = 0$ is called a *transshipment node*.
- We denote the flow of the commodity from i to j by the variable f_{ij}.

Formulating the Network Flow Problem

- A *feasible flow* satisfies the following conditions on the flow variables:

$$\sum_{j \in O(i)} f_{ij} - \sum_{j \in I(i)} f_{ji} = b_i \quad \forall i \in N$$
$$0 \leq f_{ij} \leq u_{ij} \, \forall (i,j) \in A$$

- The first set of constraints are called the *flow balance constraints* and can be read as "flow out - flow in = supply".
- Any setting of the variables satisfying the flow balance constraints is called a *flow*.
- The second set of constraints are called the *capacity constraints*.
- In order for a feasible flow to exist, we must have $\sum_{i \in N} b_i = 0$.

Types of Network Flow Problems

- The *minimum cost network flow problem* is to find a feasible flow minimizing the objective function

$$\sum_{(i,j)\in A} c_{ij} f_{ij}$$

- Special cases
  - Shortest path problem: Determine the length of a shortest path between a specified pair of nodes.
  - Maximum flow problem: Determine the the maximum amount of flow that can be sent through the network from a given source to a given sink without exceeding arc capacities.
  - Transportation problem: Minimize the cost of shipping a good from a specified set of suppliers to a specified set of customers.
  - Assignment problem: A special case of the transportation problem used to model the minimum cost of assigning workers to jobs.

Variants of the Network Flow Problem

- Every network flow problem can be reduced to one with exactly one source and one sink.
- Every network flow problem can be reduced to one without any sources or sinks.
  - This is called a *circulation*.
  - A *simple circulation* is one in which the only nonzero flows are on the arcs of a cycle.
- We can also easily model problems in which the nodes have capacities (how?).
- We may also want to have lower bounds on the arcs.

The Node-arc Incidence Matrix

- The *node-arc incidence matrix* A is defined as follows:

$$a_{ik} = \begin{cases} 1, & \text{if } i \text{ is the tail of the of the } kth \text{ arc,} \\ -1, & \text{if } i \text{ is the head of the } kth \text{ arc, and} \\ 0, & \text{otherwise} \end{cases}$$

- With this matrix, we can now rewrite the flow balance constraints more compactly as $Af = b$.
- Note that the rows of A sum to zero and are hence linearly dependent.
- Under the assumption that
  - $\sum_{i \in N} b_i = 0$, and
  - G is connected,

  if we delete the last row to obtain $\tilde{A}$, then $\tilde{A}$ has full rank.