15.093 Optimization Methods

Lecture 10: Network Optimization
The Network Simplex Algorithm

Network Optimization

Why do we care?

- Networks and associated optimization problems constitute reoccurring structures in many real-world applications.
- The network structure often leads to additional insight and improved understanding.
- It also enable us to design more efficient algorithms.

Outline

Today's Lecture

- The Simplex Algorithm: A Reminder
- The Network Simplex: A Combinatorial View
- The Network Simplex: An Animated View
- The Network Simplex: An Algebraic View

The Simplex Algorithm

A Reminder

The Problem...

\[ \min c^T x \]
\[ \text{s.t. } Ax = b \]
\[ x \geq 0 \]

The Simplex Algorithm

The Algorithm

1. Start with basis \( B = [A_{B(1)}, \ldots, A_{B(m)}] \) and BFS \( x \).
2. Compute \( \bar{z}_j = c_j - c_B^T B^{-1} A_j \)
   - If \( \bar{z}_j \geq 0 \): optimal; stop.
   - Select \( j \) such that \( \bar{z}_j < 0 \).
3. Compute \( u = B^{-1} A_j \)
   - \( \theta^* = \min_{i: \text{nonzero } x_i} \frac{x_B(i)}{u_i} = \frac{x_B(i)}{u} \)
4. Form a new basis by replacing \( A_{B(j)} \) with \( A_j \).
5. \( y_j = \theta^*; x_B(j) = x_B(i) \cdot \theta^* u_i \)

The Network Simplex Algorithm

Combinatorially...

Determine a least cost shipment of a commodity through a network in order to satisfy demands at certain nodes from available supplies at other nodes. Arcs have costs associated with them.
What is the flow in arc (4,3)?

What is the flow in arc (5,3)?
What is the flow in arc (3,2)?

What is the flow in arc (2,6)?

What is the flow in arc (7,1)?

What is the flow in arc (1,2)?

Note: there are two different ways of calculating the flow on (1,2), and both ways give a flow of 4. Is this a coincidence?

- Every tree flow has a corresponding tree (and perhaps more than one).
- Given a tree, we obtain a unique tree flow associated with it.
Theorem 1: If the objective function is bounded from below, a min-cost flow problem always has an optimal tree solution.

Proof:
- \( \min \sum_{ij \in T} c_{ij} x_{ij} \) is equivalent to \( \min \sum_{ij \in T} c_{ij} x_{ij} \).
- \( \min \sum_{ij \in T} c_{ij} x_{ij} \) is equivalent to \( \min \sum_{ij \in A \setminus T} c_{ij} x_{ij} \).
- For any solution \( x \), \( x_{ij} \geq x_{ij} \) for all \( (i,j) \in A \setminus T \).
Here is a spanning tree with arc costs. How can one choose node potentials so that reduced costs of tree arcs are 0?

There is a redundant constraint in the minimum cost flow problem. One can set $p_1$ arbitrarily. We will let $p_1 = 0$.

What is the node potential for 2?

What is the node potential for 7?

What is the potential for node 3?

What is the potential for node 6?

What is the potential for node 4?
What is the potential for node 5?

These are the node potentials associated with this tree. They do not depend on arc flows, nor on costs of non-tree arcs.

Node potentials
Original costs

Flow on arcs
Reduced costs

Flow on arcs
The Network Simplex Algorithm

Tree Solutions

Updating the Tree...

The Network Simplex Algorithm

Overview of the Algorithm

1. Determine an initial feasible tree $T$. Compute flow $x$ and node potentials $y$ associated with $T$.
2. Calculate $e_{ij} = c_{ij} - p_i + p_j$ for $(i, j) \notin T$.
   - If $e_{ij} \geq 0$, $x$ optimal; stop.
   - Select $(i, j)$ with $e_{ij} < 0$.
3. Add $(i, j)$ to $T$ creating a unique cycle $C$. Send a maximum flow around $C$ while maintaining feasibility. Suppose the exiting arc is $(k, i)$.
4. $T := (T \setminus (k, i)) \cup (i, j)$

The Network Simplex Algorithm

An Animation

Min-Cost Flow

Integrality

Our reasoning has two important and far-reaching implications:

- There always exists an integer optimal flow (if node balances $b_n$ are integer).
- There always exist optimal integer node potentials (if arc costs $c_{ij}$ are integer).

The Network Simplex Algorithm

The Algebraic View

- Bases and trees.
- Dual variables and node potentials.
- Changing bases and updating trees.
- Optimality testing.

The Network Simplex Algorithm

The Algebraic View

Bases vs. Trees...

The constraint matrix $A$ of the min-cost flow problem is the node-arc incidence matrix of the underlying network.

$$
\begin{array}{cccccccccc}
(1, 2) & (2, 3) & (3, 4) & (4, 5) & (5, 6) & (6, 7) & (7, 3) \\
1 & +1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
2 & -1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
3 & 0 & 0 & +1 & 0 & 0 & 0 & 0 & 0 & 0 \\
4 & 0 & 0 & 0 & +1 & +1 & 0 & 0 & 0 & 0 \\
5 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
6 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & -1 & +1 \\
7 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 \\
\end{array}
$$

The rows of $A$ are linearly dependent.
The Network Simplex Algorithm

Let $B$ be the submatrix corresponding to the tree.

The Algebraic View

...Bases vs. Trees...

Corollary 1

(a) The matrix $A$ has rank $n - 1$.

(b) Every tree solution is a basic solution.

Theorem 3

Every tree defines a basis and, conversely, every basis defines a tree.
The Network Simplex Algorithm

Optimality Testing...

The Algebraic View

Remember, the simplex algorithm computes the reduced costs \( \tilde{c}_j \) as 
\[
\tilde{c}_j = c_j - \bar{p}^T \bar{A}_j.
\]

\[
\begin{array}{cccccccc}
1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\
1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\
1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\
1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\
1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\
1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\
1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\
1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\
\end{array}
\]

Therefore, \( \tilde{c}_j = c_j - p_1 + p_2 \).

The Network Simplex Algorithm

Summary

- The network simplex algorithm is extremely fast in practice.
- Relying on network data structures, rather than matrix algebra, causes the speedups. It leads to simple rules for selecting the entering and exiting variables.
- Running time per pivot:
  - arcs scanned to identify an entering arc,
  - arcs scanned of the basic cycle,
  - nodes of the subtree.
- A good pivot rule can dramatically reduce running time in practice.