Graphs and Network Flows
IE411

Lecture 19

Dr. Ted Ralphs

Network Simplex Algorithm

Input: A network $G = (N, A)$, a vector of capacities $u \in \mathbb{Z}^A$, a vector of costs $c \in \mathbb{Z}^A$, and a vector of supplies $b \in \mathbb{Z}^N$

Output: x represents a minimum cost network flow

Determine an initial feasible tree structure (T, L, U)
Let x be flow and π be node potentials associated with (T, L, U)
while Some non-tree arc violates the optimality conditions **do**
  Select an entering arc (k, l) violating optimality conditions
  Add arc (k, l) to tree and determine leaving arc (p, q)
  Perform a tree update and update solutions x and π
end while

Initial Spanning Tree Structure

- Idea 1: Recall our assumption that network G contains an uncapacitated directed path between every pair of nodes.
  - How can we satisfy this assumption if it is not true?
  - How can we use this to create an initial structure?
- Idea 2: Suppose we do not want to make this assumption. What can we do?

Checking Optimality

- Suppose we have a feasible (T, L, U) and a corresponding set of π.
- What can we conclude if

$$c_{ij}^{\pi} \geq 0 \ \forall (i,j) \in L \text{ and } c_{ij}^{\pi} \leq 0 \ \forall (i,j) \in U?$$

- What arcs are *eligible* to enter and what is an arc's *violation*?

Pivot Rules

- Dantzig's Pivot Rule: Arc with maximum reduced cost (expensive)
- First Eligible Pivot Rule: First arc with negative reduced cost (cheap)
- Candidate List Pivot Rule: A compromise

Leaving Arc

- What happens when we add an entering arc (k, l) to the current spanning tree?
- The cycle W created is called *a pivot cycle*.
- If $(k, l) \in L$, define orientation in direction of (k, l); if $(k, l) \in U$, define orientation in reverse.
- Let $\bar{W}$ denote the set of forward arcs and let $\underline{W}$ denote the set of reverse arcs.
- What happens to cost of solution if we send additional flow around W in direction of orientation?
- How much flow can we augment?

Augmenting Flow in Pivot Cycle

- What happens when we augment flow along W?
- The maximum flow change δ_{ij} on arc $(i, j) \in W$ satisfies:

$$\delta_{ij} = \begin{cases} u_{ij} - x_{ij} & \text{if } (i, j) \in \bar{W} \\ x_{ij} & \text{if } (i, j) \in \underline{W} \end{cases}$$

- Therefore, to maintain feasibility, $\delta = ?$
- Algorithmically, how do we find the pivot cycle and determine δ?

Leaving Arc

- What is a *blocking arc*?
- What is a *non-degenerate* iteration?
- What is a *degenerate* iteration?
- If two arcs tie in determining δ, what can we say about the next spanning tree?
- What is the difficult part of this step?

Updating the Tree

At this point, we have identified the leaving arc (p,q) associated with the entering arc (k,l).

- Can $(p,q) = (k,l)$?
- What do we need to update?
  - Update flow
    - Find apex and determine δ.
    - Augment flow and update data structure.
  - Update node potentials.
    - Deletion of (p,q) creates T_1 (containing root) and T_2.
    - Depending on orientation of (p,q), all node potentials in T_2 change by $\pm c^{\pi}_{kl}$.

Termination

- If each pivot operation is non-degenerate, it is easy to show that the algorithm terminates finitely.

- If one or more pivot operations is degenerate, $\cdots$