Graphs and Network Flows
IE411

Lecture 18

Dr. Ted Ralphs

Network Simplex Algorithm

- Simplex method for LPs "perhaps most powerful algorithm ever devised for solving constrained optimization problems" (Ahuja et al., p.402)
  - Pervasiveness of its applications
  - Extraordinary efficiency
  - Sensitivity analysis and duality theory
- Can simplex method compete with other combinatorial methods for solving MCFPs?

Spanning Tree Solutions

- A graph $G' = (N', A')$ is a *spanning subgraph* of $G = (N, A)$ if $N' = N$ and $A' \subseteq A$.
- A *spanning tree* is a tree T that is a spanning subgraph of G.
- For any feasible solution x:
  - an arc (i, j) is a *free arc* if $0 < x_{ij} < u_{ij}$
  - an arc (i, j) is a *restricted arc* if $x_{ij} = 0$ or $x_{ij} = u_{ij}$
- We refer to a solution x as
  - *a cycle free solution* if the network contains no cycle composed only of free arcs, and
  - *a spanning tree solution* (with respect to a given spanning tree T) if every non-tree arc is a restricted arc.

Fundamental Result 1

Theorem 1. [11.1] *If the objective function of a MCFP is bounded from below over the feasible region, the problem always has a cycle free solution.*

Proof Idea:

Suppose we want increase flow by θ units along a cycle.

- What happens to flow on arcs?
- What happens to objective function?
- What is the "optimal" value of θ?

Fundamental Result 2

Theorem 2. [11.2] *If the objective function of a MCFP is bounded from below over the feasible region, the problem always has an optimal spanning tree solution.*

Creating a Spanning Tree Solution

Consider the collection of free arcs.

- What is this structure?
- How do we get a spanning tree?
- Is the spanning tree solution unique?

Spanning Tree Solution

A spanning tree solution partitions the arc set A into 3 subsets.

1. $T =$ arcs in the spanning tree
2. $L =$ non-tree arcs whose flow is restricted to value 0
3. $U =$ non-tree arcs whose flow if restricted to arc capacity

(T, L, U) form *a spanning tree structure*.

- From a spanning tree solution, we can get (T, L, U).
- From (T, L, U), we can obtain a unique spanning tree solution. How?

Spanning Tree Structure Definitions

A spanning tree structure (T, L, U) is

- *feasible* if its associated spanning tree solution satisfies all flow bounds
- *non-degenerate* if every arc in spanning tree solution is a free arc; otherwise it is *degenerate*
- *optimal* if its associated spanning tree solution is optimal for the MCFP

Optimality Conditions

Theorem 3. [11.3] *A spanning tree structure* (T, L, U) *is an optimal spanning tree structure of the MCFP if it is feasible and for some choice of node potentials* π*, the arc reduced costs* $c_{ij}^{\pi} = c_{ij} - \pi(i) + \pi(j)$ *satisfy*

1. $c_{ij}^{\pi} = 0 \ \forall (i,j) \in T$
2. $c_{ij}^{\pi} \geq 0 \ \forall (i,j) \in L$
3. $c_{ij}^{\pi} \leq 0 \ \forall (i,j) \in U$

Proof:

Maintaining a Spanning Tree Structure

Consider a tree hanging from its root node, which we call node 1.

- *predecessor index*: each node i has a unique path connecting it to the root. pred(i) stores the node after i on the path from i to node 1.
- *depth index*: number of arcs in path from i to root
- *thread index*: discovery time of the node in a depth-first search

How can we easily find all descendants?

Spanning Tree Example

Computing Node Potentials

- The network simplex algorithm maintains a spanning tree solution at all times.
- In each iteration, it moves to a new spanning tree, maintaining the invariant that the reduced cost of every arc (i, j) in current spanning tree is zero.
- Given a (T, L, U), how can we determine π that satisfies $c_{ij}^{\pi} = 0$ $\forall (i, j) \in T$?
  - True or False? Adding a constant k to each $\pi(i)$ does not alter the reduced cost of any arc.
  - Let's assume that $\pi(1) = 0$. How can we compute $\pi(i)$?
  - Does this suggest an algorithm?

Computing Node Potentials

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 $b \in \mathbb{Z}^N$, and a tree solution (T, L, U) along with corresponding thread and pred indices.

Output: A set of node potential π such that the reduced cost of all arcs in T is zero.

$\pi(1) \leftarrow 0$
$j \leftarrow$ thread(1)
while $j \neq 1$ **do**
  $i =$ pred(j)
  if $(i, j) \in A$ **then**
    $\pi(j) = \pi(i) - c_{ij}$
  end if
  if $(j, i) \in A$ **then**
    $\pi(j) = \pi(i) + c_{ji}$
  end if
  $j =$ thread(j)
end while

Computing Arc Flows

Consider first the uncapacitated version. Then, all non-tree arcs carry zero flow (why?).

1. If we delete a tree arc (i, j), what happens?
2. Since we have a spanning tree solution, $\sum_{k \in T_1} b(k) = ?$

Procedure *Compute Flows*

- Start with $x_{ij} = 0$ for all $(i, j) \in L$
- Select a leaf node j other than node 1 and let $i = \text{pred}(j)$
- If $(i, j) \in T$ then $x_{ij} = -b(j)$; else $x_{ji} = b(j)$
- Add $b(j)$ to $b(i)$, delete node j and arc (i, j)
- How do we adjust for the capacitated version?

Primal Integrality Property

Theorem 4. *If capacities of all the arcs and supplies/demands of all the nodes are integer, the MCFP always has an integer optimal flow.*

Proof: