In the name of God

Network Flows

3. Shortest Path Problems
3.1 Introduction

Fall 2010

Instructor: Dr. Masoud Yaghini

Shortest Path Problems

- ***Shortest path problems*** are attractive to both researchers and to practitioners for several reasons:
  - (1) they arise frequently in practice
  - (2) they are easy to solve efficiently
  - (3) they provide both a benchmark and a point of departure for studying more complex network models
  - (4) they arise frequently as subproblems when solving many combinatorial and network optimization problems

Notation and Assumptions

Notation and Assumptions

- ***Notation and Assumptions***
  - $G = (N, A)$: a directed network with an arc length (or arc cost) c_{ij} associated with each arc $(i, j) \in A$.
  - s : the source.
  - $A(i)$: represent the arc adjacency list of node I
  - $C = \max\{c_{ij}: (i, j) \in A\}$.
  - ***the length of a directed path*** : the sum of the lengths of arcs in the path.

Notation and Assumptions

- The ***shortest path problem***
  - is to determine for every non source node $i \in N$ a shortest length directed path from node s to node i.
- Alternatively,
  - the problem as sending 1 unit of flow as cheaply as possible (with arc flow costs as c_{ij}) from node s to each of the nodes in N - $\{s\}$ in an **uncapacitated network**.
-

Notation and Assumptions

- ***The linear programming formulation of the shortest path problem:***

$$\text{Minimize} \sum_{(i,j)\in A} c_{ij} x_{ij}$$

subject to

$$\sum_{\{j:(i,j)\in A\}} x_{ij} - \sum_{\{j:(j,i)\in A\}} x_{ji} = \begin{cases} n-1 & \text{for } i = s \\ -1 & \text{for all } i \in N - \{s\} \end{cases}$$

$$x_{ij} \geq 0 \qquad \text{for all } (i,j) \in A.$$

Notation and Assumptions

- ***Assumption 1. All arc lengths are integers***
  - The integrality assumption imposed on arc lengths is necessary for some algorithms and unnecessary for others.
  - We can always transform rational arc capacities to integer arc capacities by multiplying them by a suitably large number.
  - Therefore, the integrality assumption is really not a restrictive assumption in practice.

Notation and Assumptions

- ***Assumption 2. The network contains a directed path from node s to every other node in the network.***
  - We can always satisfy this assumption by adding a ***fictitious*** arc (s, i) of suitably large cost for each node i that is not connected to node s by a directed path.

Notation and Assumptions

- ***Assumption 3. The network does not contain a negative cycle (i.e., a directed cycle of negative length).***
  - For any network containing a negative cycle W, the linear programming formulation has an unbounded solution because we can send an infinite amount of flow along W.
  - The shortest path problem with a negative cycle is an **NP-complete** problem, no polynomial-time algorithm for this problem is likely to exist

Notation and Assumptions

- ***Assumption 4.4. The network is directed.***
  - If the network were undirected and all arc lengths were nonnegative, we could transform this shortest path problem to one on a directed network.

Various Types of Shortest Path Problems

- ***Various Types of Shortest Path Problems***
  - ***the single-source shortest path problem***
    - Finding shortest paths from one node to all other nodes
  - ***the all-pairs shortest path problem***
    - Finding shortest paths from every node to every other node

Algorithmic approaches for solving shortest path problems

Label-Setting and Label-Correcting Algorithms

- Algorithmic approaches for solving shortest path problems:
  - ***label setting algorithms***
  - ***label correcting algorithms***
- ***Characteristics:***
  - Both approaches are iterative.
  - They assign tentative ***distance labels*** to nodes at each step
  - The distance labels are estimates of (i.e., upper bounds on) the shortest path distances.
  - The approaches vary in how they update the distance labels from step to step and how they ***converge*** toward the shortest path distances.

Label-Setting and Label-Correcting Algorithms

- ***Updating the distance labels***
  - Label-setting algorithms designate one label as permanent (optimal) at each iteration.
  - Label-correcting algorithms consider all labels as temporary until the final step, when they all become permanent.

Label-Setting and Label-Correcting Algorithms

- ***The class of problems that they solve***
  - Label-setting algorithms are applicable only to
    - (1) shortest path problems defined on acyclic networks with arbitrary arc lengths, and to
    - (2) shortest path problems with nonnegative arc lengths.
  - The label-correcting algorithms
    - are more general and apply to all classes of problems, including those with negative arc lengths.

Label-Setting and Label-Correcting Algorithms

- ***Efficiency of the approaches***
  - The label-setting algorithms are much more efficient, that is, have much better worst-case complexity bounds;
  - The label-correcting algorithms not only apply to more general classes of problems, but as we will see, they also offer more algorithmic flexibility.
- In fact, we can view the label-setting algorithms as special cases of the label-correcting algorithms.

Shortest Path Tree

- In the shortest path problem, we wish to determine a shortest path from the source node to all other (n - 1) nodes.
- How much storage would we need to store these paths?
  - One naive answer would be an upper bound of $(n – 1)^2$ since each path could contain at most (n - 1) arcs.
- We need not use this much storage: (n - 1) storage locations are sufficient to represent all these paths.

Shortest Path Tree

- ***Shortest path tree***
  - We can always find a ***directed out-tree rooted*** from the source with the property that the unique path from the source to any node is a shortest path to that node.
  - We refer to such a tree as a ***shortest path tree***.

- The shortest path tree relies on the following property.
  - **If the path s = i_1 - i_2 - … - i_h = k is a shortest path from node s to node k, then for every q = 2, 3, ... , h - 1, the subpath s = i_1 - i_2 - … - i_q is a shortest path from the source node to node i_q.**

The End