15.053 Thursday, March 14

- Introduction to Network Flows
- Handouts: Lecture Notes

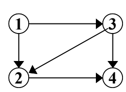
Network Models

- Linear Programming models that exhibit a very special structure
- Can use this structure to dramatically reduce computational complexity
- First widespread application of LP to problems of industrial logistics
- Addresses huge number of diverse applications

Notation and Terminology

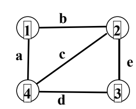
Note: Network terminology is not (and never will be) standardized. The same concept may be denoted in many different ways.

Called:
- NETWORK
- directed graph
- digraph
- graph

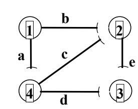


Class Handouts (Ahuja, Magnanti, Orlin)	Also Seen
Network G = (N,A)	*Graph* G = (V,E)
Node set N = {1,2,3,4}	*Vertex set* V = {1,2,3,4}
Arc Set {(1,2),(1,3),(3,2),(3,4),(2,4)}	*Edge set:* A={1-2,1-3,3-2,3-4,2-4}

Directed and Undirected Networks



An Undirected Graph



A Directed Graph

- Networks are used to transport commodities
  - physical goods (products, liquids)
  - communication
  - electricity, etc.
- The field of Network Optimization concerns optimization problems on networks

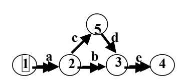
An Overview of Some Applications of Network Optimization

Applications	Physical analog of nodes	Physical analog of arcs	Flow
Communication systems	phone exchanges, computers, transmission facilities, satellites	Cables, fiber optic links, microwave relay links	Voice messages, Data, Video transmissions
Hydraulic systems	Pumping stations Reservoirs, Lakes	Pipelines	Water, Gas, Oil, Hydraulic fluids
Integrated computer circuits	Gates, registers, processors	Wires	Electrical current
Mechanical systems	Joints	Rods, Beams, Springs	Heat, Energy
Transportation systems	Intersections, Airports, Rail yards	Highways, Airline routes Railbeds	Passengers, freight, vehicles, operators

Examples of terms.

Path: **Example: 5, 2, 3, 4.**
(or 5, c, 2, b, 3, e, 4)
Note that directions are ignored.

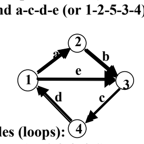
Directed Path **. Example: 1, 2, 3, 4**
(or 1, a, 2, b, 3, e)
Directions are important.



Two paths a-b-e (or 1-2-3-4) and a-c-d-e (or 1-2-5-3-4)

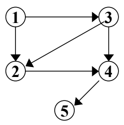
Cycle or circuit (or loop)
1, 2, 3, 1. (or 1, a, 2, b, 3, e)
Note that directions are ignored.

Directed Cycle: (1, 2, 3, 4, 1) or 1, a, 2, b, 3, c, 4, d, 1
Directions are important.



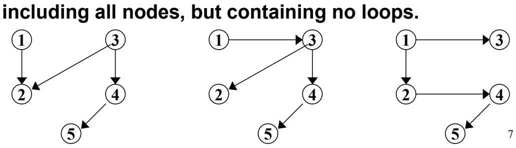
Cycles (loops):
a-b-c-d (or 1-2-3-4-1)
b-a-d-c (or 3-2-1-4-3)
e-b-a (or 1-3-2-1)
c-d-e (or 3-4-1-3)

More Definitions



A network is ***connected*** if every node can be reached from every other node by following a sequence of arcs in which direction is ignored.

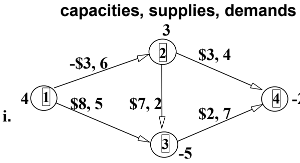
A *spanning tree* is a connected subset of a network including all nodes, but containing no loops.



The Minimum Cost Flow Problem

Network $G = (N, A)$
- Node set N, arc set A;
- Capacities u_{ij} on arc (i,j)
- Cost c_{ij} on arc (i,j)
- Supply/demand b_i for node i. (Positive indicates supply)

A network with costs, capacities, supplies, demands



Minimize the cost of sending flow
s.t. Flow out of i - Flow into i = b_i
Flow on arc (i,j) $\le u_{ij}$

The Minimum Cost Flow Problem

Let x_{ij} be the flow on arc (i,j).

Minimize the cost of sending flow
s.t. Flow out of i - Flow into i = b_i
$0 \le x_{ij} \le u_{ij}$

Minimize $\sum_{i=1}^{m}\sum_{j=1}^{n} c_{ij}x_{ij}$

s.t. $\sum_{j=1}^{n} x_{ij} - \sum_{k=1}^{n} x_{ki} = b_i$ for all i

$0 \le x_{ij} \le u_{ij}$ for all i-j

Example Formulation

Min $-3x_{12} + 8x_{13} + 7x_{23} + 3x_{24} + 2x_{34}$

s.t $x_{12} + x_{13} = 4$

$x_{23} + x_{24} - x_{12} = 3$

$x_{34} - x_{13} - x_{23} = -5$

$- x_{24} - x_{34} = -2$

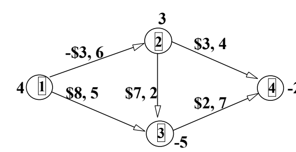
$0 \le x_{12} \le 6$

$0 \le x_{13} \le 5$

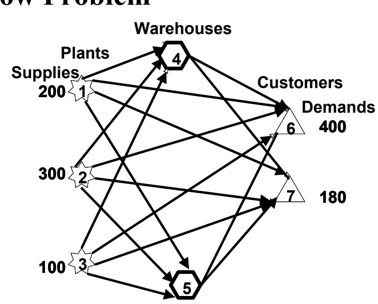
$0 \le x_{23} \le 2$

$0 \le x_{24} \le 4$

$0 \le x_{34} \le 7$



An Application of the Minimum Cost Flow Problem



Ship from suppliers to customers, possibly through warehouses, at minimum cost to meet demands.

Useful Facts About The Minimum Cost Flow Problem

- **Suppose the following properties of the constraint matrix, A (ignoring simple upper and lower variable bounds, such as $x \le 7$) hold:**
  - **(1) all entries of A are 0 or 1 or -1**
  - **(2) there is at most one 1 in any column and at most one -1.**
- **Then this is a minimum cost flow problem.**

Useful Facts (cont'd)

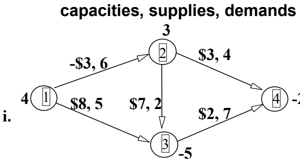
Theorem. If one carries out the simplex algorithm on the minimum cost flow problem with integer valued capacities and RHS, then at every iteration of the simplex algorithm, each coefficient in the tableau (except for costs and RHS) is either 0 or -1 or 1.

Corollary. The optimal LP solution is integer valued.

The Minimum Cost Flow Problem

Network G = (N, A)
- **Node set N, arc set A;**
- **Capacities u_{ij} on arc (i,j)**
- **Cost c_{ij} on arc (i,j)**
- **Supply/demand b_i for node i. (Positive indicates supply)**

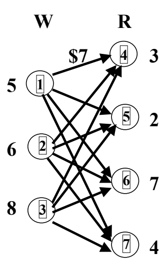
A network with costs, capacities, supplies, demands



Minimize the cost of sending flow
s.t. Flow out of i - Flow into i = b_i
Flow on arc (i,j) ≤ u_{ij}

The Transportation Problem

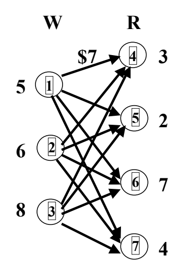
Suppose that one wants to ship from warehouses to retailers



In this example:
3 warehouses
4 retailers
a_i is the supply at warehouse i.
b_j is the demand at retailer j.
c_{ij} is the cost of shipping from i to j.
There are no capacities on the arcs.

Let x_{ij} be the amount of flow shipped from warehouse i to retailer j.
How do we formulate an LP?

The Transportation Problem is a Min Cost Flow Problem



Minimize the cost of sending flow
s.t. Flow out of i - Flow into i = b_i
$0 \leq x_{ij} \leq u_{ij}$

Flow out occurs at the supply nodes.
Flow in occurs at demand nodes.
Capacities are infinite: $u_{ij} = \infty$

The Transportation Problem

In general the LP formulation is given as

Minimize $$\sum_{i=1}^{m}\sum_{j=1}^{n} c_{ij} x_{ij}$$

$$\sum_{j=1}^{n} x_{ij} = a_i, \forall i = 1,\ldots,m$$ S: Supply nodes

$$\sum_{i=1}^{m} x_{ij} = b_j, \forall j = 1,\ldots,n$$ D: Demand nodes

$$x_{ij} \geq 0, \forall ij$$

All arcs are from a node in S to a node in D, and uncapacitated.

Useful Facts About Transportation Problem

Suppose that
- **(1) the constraint matrix can be partitioned into $A_1x = b_1$ and $A_2x = b_2$.**
- **(2) all entries of A_1 and A_2 are 0 or 1**
- **(3) there is at most one 1 in any column of A_1 or A_2**

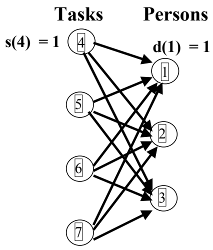
Then this is a transportation problem.

Theorem. If one carries out the simplex algorithm on the transportation problem, then at every iteration of the simplex algorithm, each coefficient in the tableau (except for costs and RHS) is either 0 or -1 or 1. The costs and RHS are both integer valued.

Corollary. The optimal solution to the LP is integer valued.

The Assignment Problem

Suppose that one wants to assign tasks to persons



In this example:
- 4 tasks
- 3 persons

No two tasks to the same person
Each person gets a task
c_{ij} is the “cost” of assigning task i to person j.

Let $x_{ij} = 1$ if task i is assigned to j.
Let $x_{ij} = 0$ otherwise.
How do we formulate an LP?

The Assignment Problem

In general the LP formulation is given as

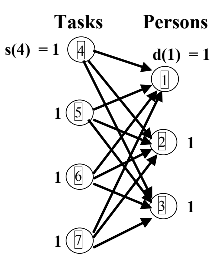
Minimize $\sum_{i=1}^{n}\sum_{j=1}^{n} c_{ij} x_{ij}$

$$\sum_{j=1}^{n} x_{ij} = 1, \forall i = 1,\ldots,n \qquad \text{Each supply is 1}$$

$$\sum_{i=1}^{n} x_{ij} = 1, \forall j = 1,\ldots,n \qquad \text{Each demand is 1}$$

$$x_{ij} = 0 \text{ or } 1, \forall ij$$

More on the Assignment Problem

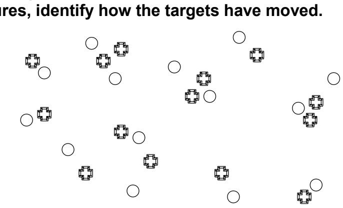


The assignment problem is a special case of the transportation problem.

The simplex algorithm can solve the LP relaxation, and it will give integer answers, that is, it will solve the assignment problem.

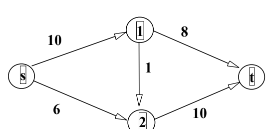
An Application of the Assignment Problem

Suppose that there are moving targets in space. You can identify each target as a pixel on a radar screen. Given two successive pictures, identify how the targets have moved.



The Maximum Flow Problem

Network G = (N, A).
- Source s and sink t
- Capacities u_{ij} on arc (i,j)
- Variable: Flow x_{ij} on arc (i,j)



Graph with capacities

Maximize the flow leaving s
s.t. Flow out of i - Flow into i = 0 for $i \neq s, t$
$0 \leq x_{ij} \leq u_{ij}$

The Max Flow Problem

In general the LP formulation is given as

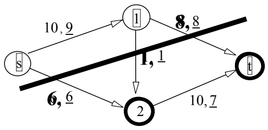
Maximize v

$$\sum_{j=1}^{n} x_{ij} - \sum_{k=1}^{n} x_{ki} = \begin{cases} v, & i = s \\ -v, & i = t \\ 0, & otherwise \end{cases} \quad \forall i = 1,\ldots,n$$

$$0 \leq x_{ij} \leq u_{ij}, \ \forall ij$$

**This is not formulated as a special case of a minimum cost flow formulation.
Can we reformulate it in this way?**

More on the maximum flow problem



Graph with capacities and flows (underlined)

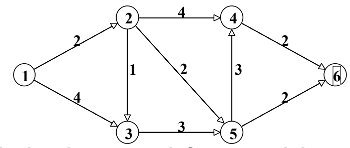
Is the current flow optimal?

An s-t cut is a separation of the nodes into two parts S and T, with s in S and T in T.

The capacity of the cut is the sum of the capacities from S to T.

The max flow from s to t is at most the capacity of any s-t cut

The Shortest Path Problem



What is the shortest path from an origin or source node (often denoted as s) to a destination or sink node, (often denoted as t)? What is the shortest path from node 1 to node 6?

Assumptions for now:

1. There is a path from node s to all other nodes.
2. All arc lengths are non-negative

Direct Applications

- What is the path with the shortest driving time from 77 Massachusetts Avenue to Boston City Hall?
- What is the path from Building 7 to Building E40 that minimizes the time spent outside?
- What is the communication path from i to j that is the fastest (taking into account congestion at nodes)?

Formulation as a linear program

In general the LP formulation is given as

Minimize $\sum_{i=1}^{n}\sum_{j=1}^{n} c_{ij}x_{ij}$

$$\sum_{j=1}^{n} x_{ij} - \sum_{k=1}^{n} x_{ki} = \begin{cases} 1, & i = s \\ -1, & i = t \\ 0, & otherwise \end{cases} \quad \forall i = 1,\ldots,n$$

$$x \geq 0, \ \forall ij$$

The Shortest Path Problem

- Fact: The Shortest path problem is a special case of the minimum cost flow problem
- Lots of interesting applications (coming up)
- Very fast algorithm (coming up)
- Connection to dynamic programming (several lectures from now)

Conclusions.

- Advantages of the transportation problem and the minimum cost flow problem
  - Integer solutions
  - Very fast solution methods
  - Extremely common in modeling
- Today we saw the following:
  - The minimum cost flow problem
  - The transportation problem
  - The assignment problem
  - The maximum flow problem
  - The shortest path problem