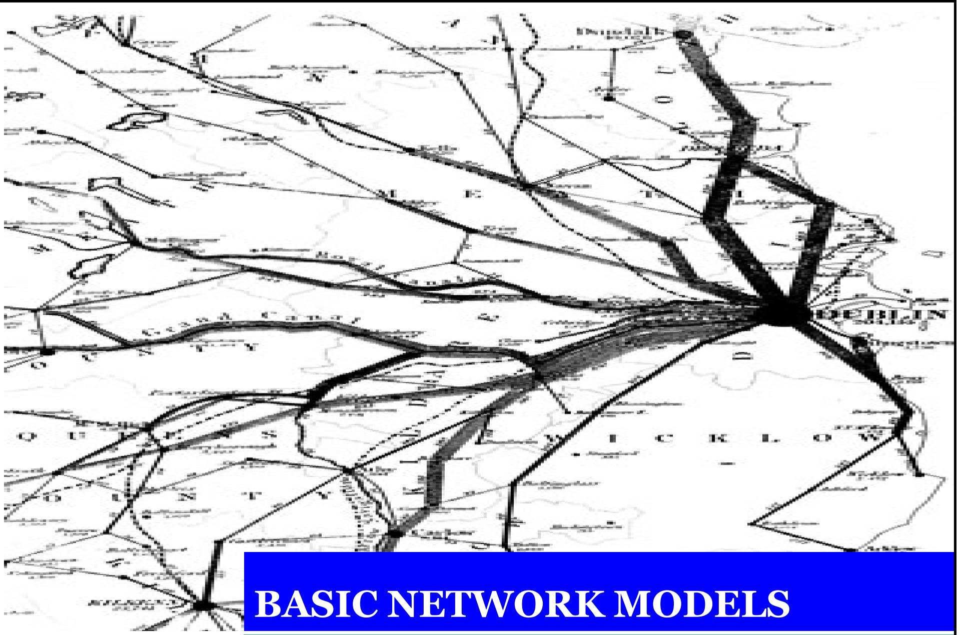


BASIC NETWORK MODELS

Course: Transportation Network Design

Network Optimization

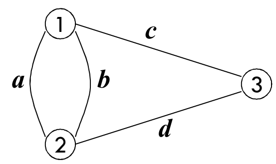
- Network flow programming (NFP) is a special case of linear programming
- Important to identify problems that can be modeled as networks because:

  (1) Network representations make optimization models easier to visualize and explain

  (2) Very efficient algorithms are available

Terminology: Graph, oriented graph, network

- A graph G =(V, E) is specified by a non empty set of nodes V and a set of edges E such that each edge a is identified by a pair of nodes (u, v).

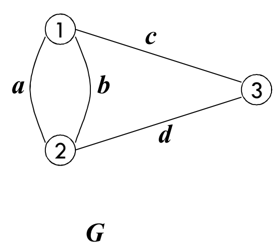


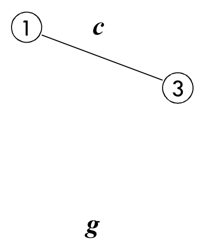
V = {1, 2, 3} E = {a, b, c, d}

a = b = (1, 2) ; c = (1, 3) ; d =(2, 3)

Terminology: Graph, oriented graph, network

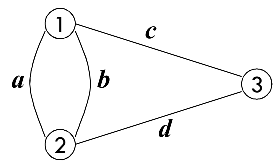
- A graph g is a sub-graph of a graph G if all the nodes and all the edges of g are nodes and edges of G.



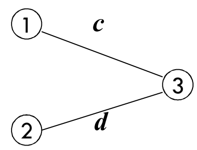


Terminology: Graph, oriented graph, network,…

- **A sub graph of a graph *G* including all the nodes of *G* is a partial graph of *G*.**



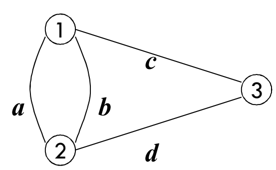
G



g

Terminology: Graph, oriented graph, network,…

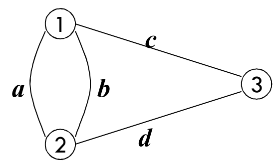
- **A chain in a graph *G* is a sequence of distinct edges $a_1, a_2, \ldots, a_p$ such that there exist $(p+1)$ nodes $u_1, u_2, \ldots, u_{p+1}$ where $a_i = (u_i, u_{i+1})$.**



The sequence *a*, *c* is a chain.

Terminology: Graph, oriented graph, network,…

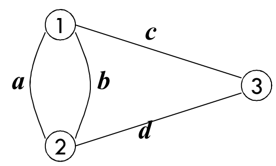
- A cycle in a graph G is a chain such that $u_1 = u_{p+1}$



The sequence c, b, d is a cycle.

Terminology: Graph, oriented graph, network,…

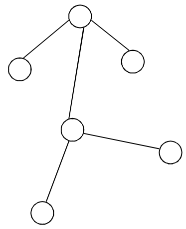
- A graph G is connected if for all pair of nodes, there exists a chain linking them.



This graph is connected.

Terminology: Graph, oriented graph, network,…

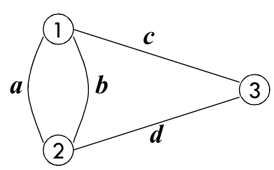
- **A tree is a connected graph with no cycle**

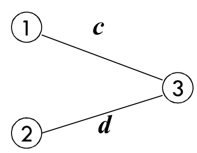


Property : A tree with n nodes includes exactly $(n-1)$ edges

Terminology: Graph, oriented graph, network,…

- A **partial tree** (**spanning tree**) of a connected graph G is a partial graph of G being a tree

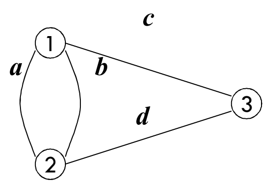




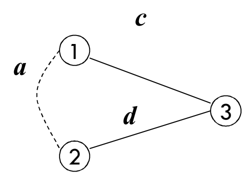
G **partial tree**

Terminology: Graph, oriented graph, network,…

- A **fondamental cycle** with respect to a partial tree is a cycle including an edge of the graph not included in the partial tree and edges of the tree.



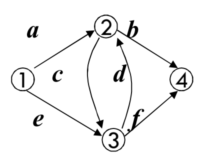
G



partial tree

Terminology: Graph, oriented graph, network,…

- **An oriented (directed) graph *G* = (*V*, *E*) is specified by a non empty set of nodes *V* and a set of arcs *E* such that each arc *a* is identified by an ordered pair of nodes (*u*, *v*).**



V = {1, 2, 3, 4} E = {a, b, c, d, e, f}

a = (1, 2), b = (2, 4), c = (2, 3), d =(3, 2), e = (1, 3), f = (3, 4)

Terminology: Graph, oriented graph, network,…

- A non oriented graph obtained from an oriented graph $\boldsymbol{G}$ by eliminating the direction of the arc is denoted the **corresponding graph**.

- The notions of chain, cycle, connectedness, tree, spanning tree, and fondamental cycle for an oriented graph refer to the corresponding graph.

- A **path** in an oriented graph is a sequence of distinct arcs $\boldsymbol{a_1, a_2, \ldots, a_p}$ being a chain where all the arcs are oriented in the same direction.

- A directed graph is **simple** if the nodes identifying each arc are distinct, and if there is no pairs of arcs specified by the same pair of nodes.

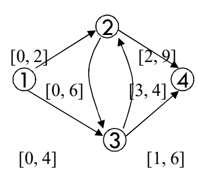
Terminology: Graph, oriented graph, network,…

- A **network** is a connected oriented graph in which a flow can move over the arcs. Each arc $\boldsymbol{(i, j)}$ is characterized as follows

  a **capacity** $\boldsymbol{u_{ij}}$ corresponding to an upper bound on the flow in the arc

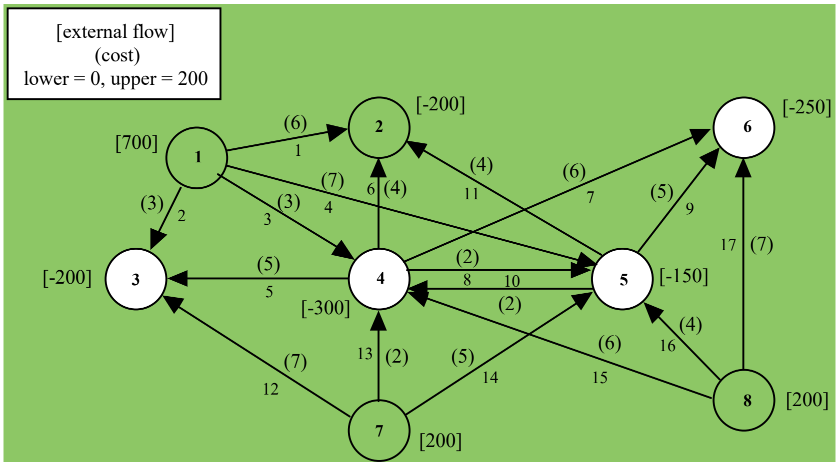
  a **lower bound** $\boldsymbol{l_{ij}}$ on the flow in the arc

Moreover, $0 \le l_{ij} \le u_{ij}$



At each arc (i, j) is associated a pair $[l_{ij}, u_{ij}]$.

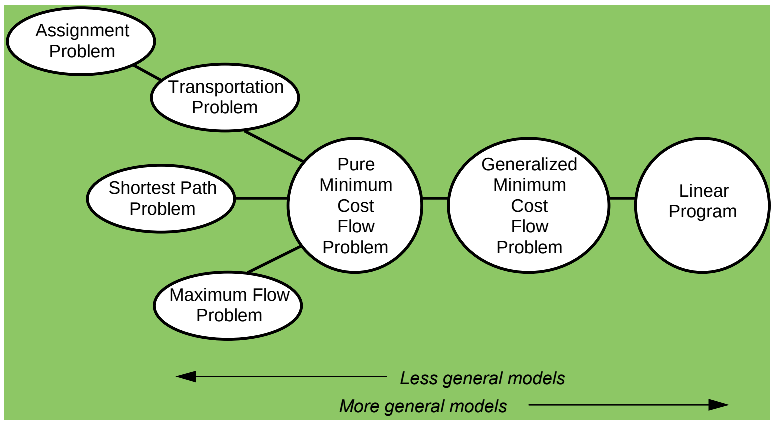
Example of (Distribution) Network



Terminology

- Nodes and arcs
- Arc flow (variables)
- Upper and lower bounds
- Cost
- Gains (and losses)
- External flow (supply an demand)
- Optimal flow

Network Flow Problems



Transportation Problem

We wish to ship goods (a single commodity) from m warehouses to n destinations at minimum cost.

Warehouse i has s_i units available $i = 1,\ldots,m$ and destination j has a demand of d_j, $j = 1,\ldots,n$.

Goal: Ship the goods from warehouses to destinations at minimum cost.

Example:

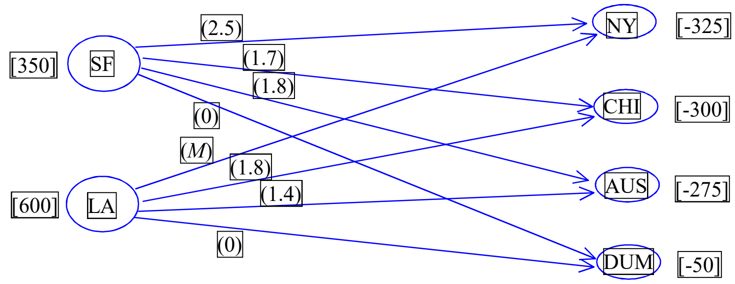
Warehouse	Supply	Markets	Demand
San Francisco	350	New York	325
Los Angeles	600	Chicago	300
		Austin	275

Unit Shipping Costs

From/To	NY	Chicago	Austin
SF	2.5	1.7	1.8
LA	--	1.8	1.4

Transportation Problem-Cont.

- The min-cost flow network for this transportation problem is given by



- Total supply = 950, total demand = 900
- Transportation problem is defined on a bipartite network
- Arcs only go from supply nodes to destination nodes; to handle excess supply we can create a dummy destination with a demand of 50 and 0 shipment cost

Transportation Problem: Modeling Issues

- Costs on arcs to dummy destination = 0
  (In some settings it would be necessary to include a nonzero warehousing cost.)

- The objective coefficient on the LA → NY arc is *M*.
  This denotes a large value and effectively prohibits use of this arc (could eliminate arc).

- We are assured of integer solutions because technological matrix *A* is totally unimodular.
  (Very important in some applications)

- Decision variables: x_{ij} = amount shipped from warehouse i to destination j

Transportation Problem: LP Formulation

The LP formulation of the transportation problem with m sources and n destinations is given by:

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

$$\text{s.t.} \quad \sum_{j=1}^{n} x_{ij} \leq s_i, \quad i = 1,\ldots,m \quad \text{(no dummy node)}$$

$$\sum_{i=1}^{m} x_{ij} = d_j, \quad j = 1,\ldots,n$$

$$0 \leq x_{ij} \leq u_{ij}, \quad i = 1,\ldots,m, \quad j = 1,\ldots,n$$

Assignment Problem

✓ Special case of transportation problem:
- same number of sources and destinations
- all supplies and demands = 1

Example

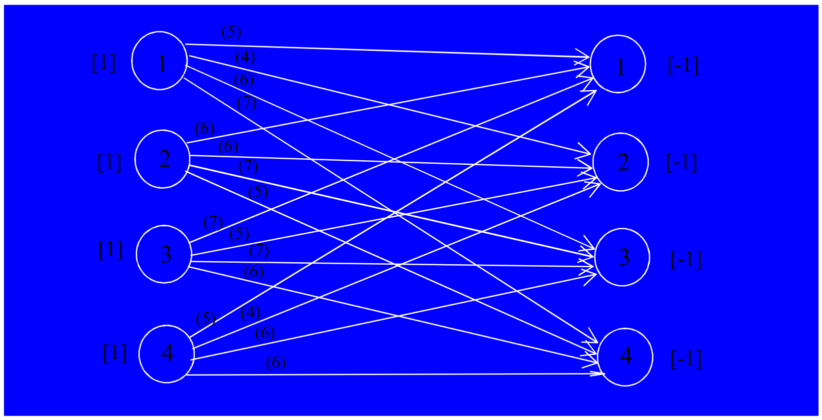
4 ships to transport **4** loads from single port to **4** separate ports;

Each ship will carry exactly **1** load;

Associated shipping costs as shown.

		Port/load			
		1	2	3	4
Ship	1	5	4	6	7
	2	6	6	7	5
	3	7	5	7	6
	4	5	4	6	6

Problem: Find a one-to-one matching between ships and ports in such a way as to minimize the total shipping cost.



Decision variables are $x_{ij} = \begin{cases} 1, & \text{if ship } i \text{ goes to port } j \\ 0, & \text{otherwise} \end{cases}$

Characteristics of Assignment Problem

- Note that from a feasibility perspective it could be possible to have $x_{11} = x_{12} = x_{13} = x_{14} = ¼$. But we know that a pure network flow problem guarantees that the simplex method will yield an integer solution. In this case we know that each x_{ij} will either take on 0 or 1.

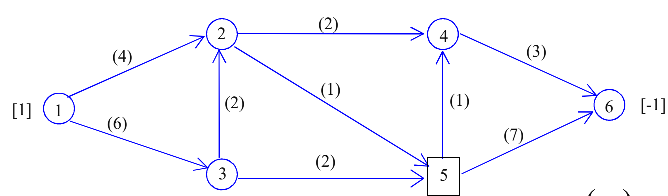
- If a particular ship cannot carry a particular load then we can use M as in the transportation problem.

- Other types of assignments:
  a. workers to jobs
  b. tasks to machines
  c. swimmers to events (in a relay)
  d. students to internships

Shortest Path Problem

- Given a network with “distances” on the arcs, our goal is to find the shortest path from the origin to the destination.
- These distances might be length, time, cost, etc., and the values can be positive or negative. (A negative c_{ij} can arise if we earn revenue by traversing an arc.)
- The shortest path problem may be formulated as a special case of the pure min-cost flow problem.

Shortest Path Problem: Example



(c_{ij}) = cost/length

- We wish to find the shortest path from node 1 to node 6.
- To do so we place one unit of supply at node 1 and push it through the network to node 6 where there is one unit of demand.
- All other nodes in the network have external flows of zero.

← SP Tree Solution

Shortest Path Problem Solution

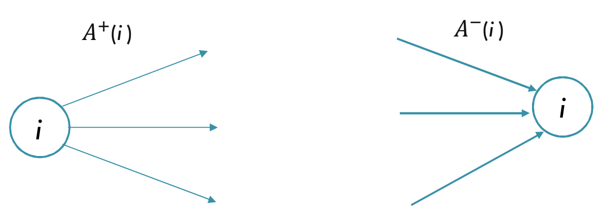
	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
1	**Network Model**				Name:	SP_9x6				Solver:	Jensen	Network		Ph. 1 Iter.	7
2					Type:	Net				Type:	Linear			Total Iter.	7
3			**Change**		Goal:	Min				Sens.:	Yes			Comp. Time	00:00
4					Objective:	9								Status	Optimal
5			**Solve**												
6															
7			**Arc Data and Flows**							**Node Data and Balance Constraints**					
8			Num.	Name	Flow	Origin	Term.	Cost		Num.	Name	Fixed	Balance		
9			1	Arc1	1	1	2	4		1	Node1	1	0		
10			2	Arc2	0	1	3	6		2	Node2	0	0		
11			3	Arc3	0	3	2	2		3	Node3	0	0		
12			4	Arc4	1	2	4	2		4	Node4	0	0		
13			5	Arc5	0	2	5	1		5	Node5	0	0		
14			6	Arc6	0	3	5	2		6	Node6	-1	0		
15			7	Arc7	0	5	4	1							
16			8	Arc8	1	4	6	3							
17			9	Arc9	0	5	6	7							
18															

Network Notation

A = set of Arcs, N = set of nodes

Forward Star for node i : $A^{+}(i) = \{ (i,j) : (i,j) \in A \}$

Reverse Star for node i : $A^{-}(i) = \{ (j,i) : (j,i) \in A \}$



Shortest Path Model

In general, if node *s* is the source node and node *t* is the termination node then the shortest path problem may be written as follows.

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

$$\text{s.t.} \quad \sum_{(i,j)\in A^+(i)} x_{ij} - \sum_{(j,i)\in A^-(i)} x_{ji} = \begin{cases} 1, & i = s \\ -1, & i = t \\ 0, & i \in N \setminus \{s,t\} \end{cases}$$

$$x_{ij} \geq 0, \quad \forall (i,j) \in A$$

General Solution to Shortest Path Problem

- In general, $x^*_{ij} = \begin{cases} 1, & \text{if } (i,j) \text{ is on the shortest path} \\ 0, & \text{otherwise} \end{cases}$
- As in the assignment problem, the integer nature of the solution is key to this shortest path formulation.
- Examples of shortest path problems:
  a. airline scheduling
  b. equipment replacement
  c. routing in telecommunications networks
  d. reliability problems
  e. traffic routing

Shortest Path Tree Problem

- It is sometimes useful to find the shortest path from node s to all other $m-1$ nodes in the network.
- We could do this by solving a collection of shortest path problems, but it is simpler to use a single min-cost flow formulation:

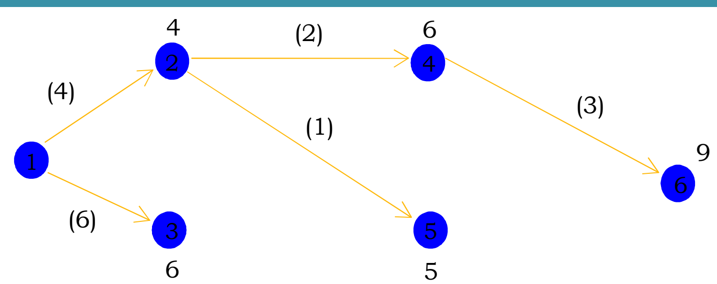
$$\text{Min} \sum_{(i,j)\in A} c_{ij} x_{ij}$$

$$\text{s.t.} \quad \sum_{(i,j)\in A^{+}(i)} x_{ij} - \sum_{(j,i)\in A^{-}(i)} x_{ji} = \begin{cases} m-1, & i = s \\ -1, & i \in N \setminus \{s\} \end{cases}$$

$$x_{ij} \geq 0, \quad \forall\ (i,j) \in A$$

where $m = |N|$ = number of nodes

In our example, the shortest path tree is



Each node is labeled with its shortest-path distance to node 1.

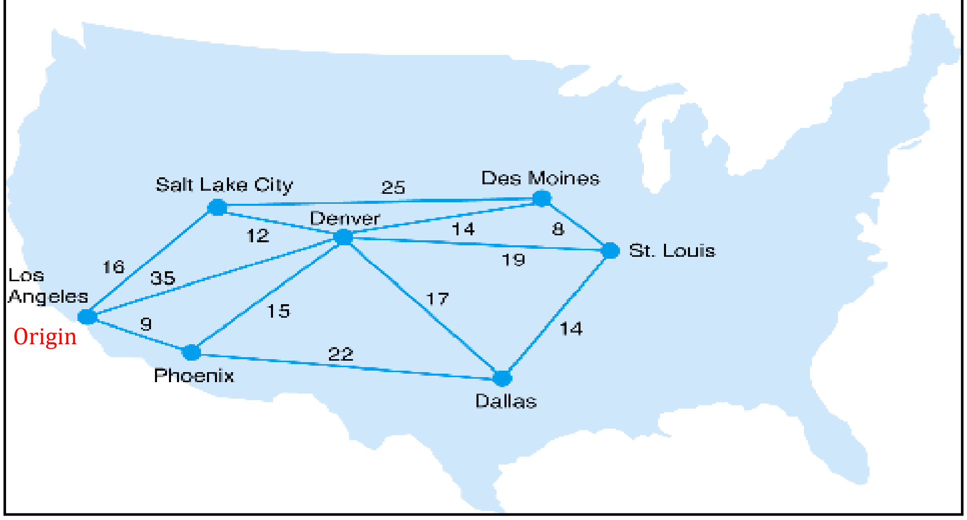
Shortest Path between all Pairs of Nodes

How to Solve Such a Repeated Problem?!

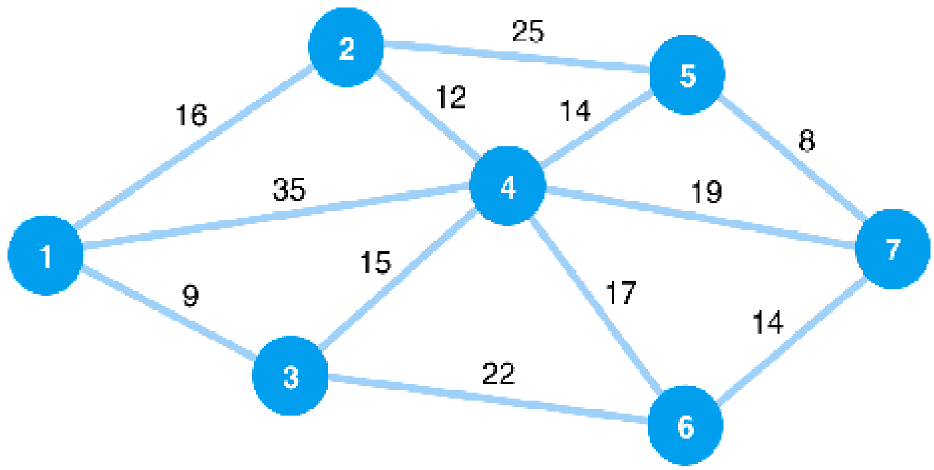
Assignment 1:
Try to find a good and efficient algorithm to solve it

The Shortest Path Tree

Problem: Determine the shortest routes from the origin to all destinations.

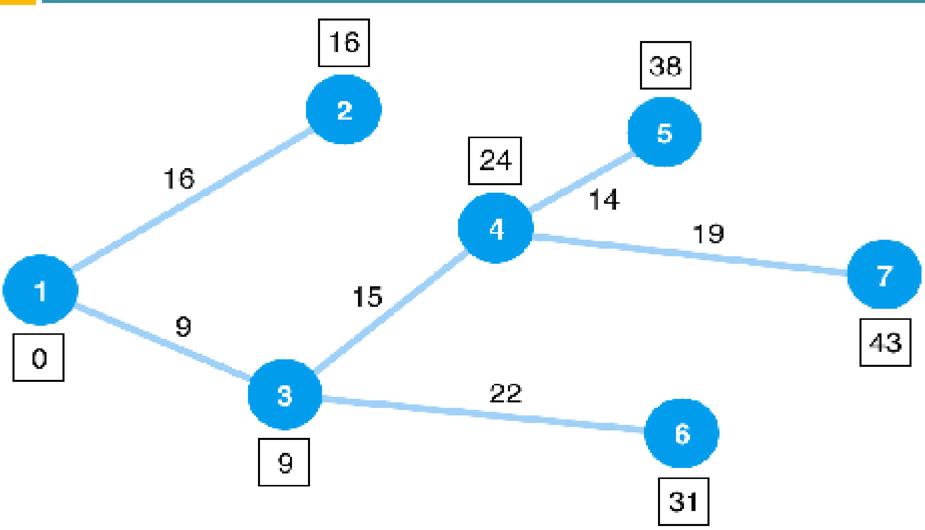


The Shortest Path Tree



Network Representation

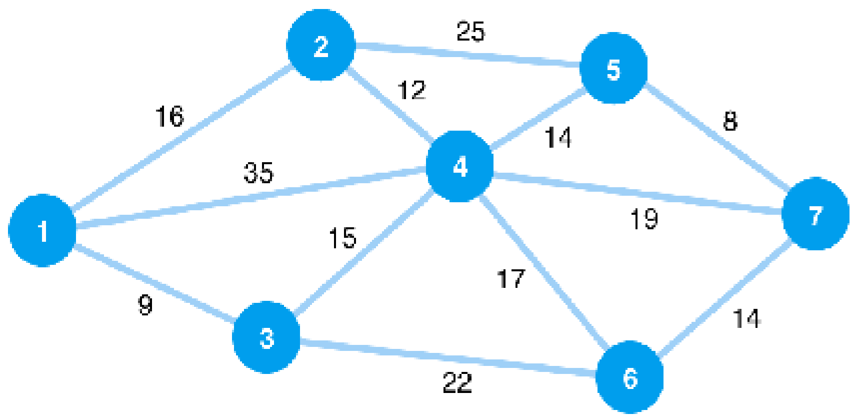
The Shortest Path Tree



Network with Optimal Routes

The Minimal Spanning Tree (MST) Problem

Problem: Connect all nodes in a network so that the total of the branch lengths are minimized.



MST: Mathematical Model

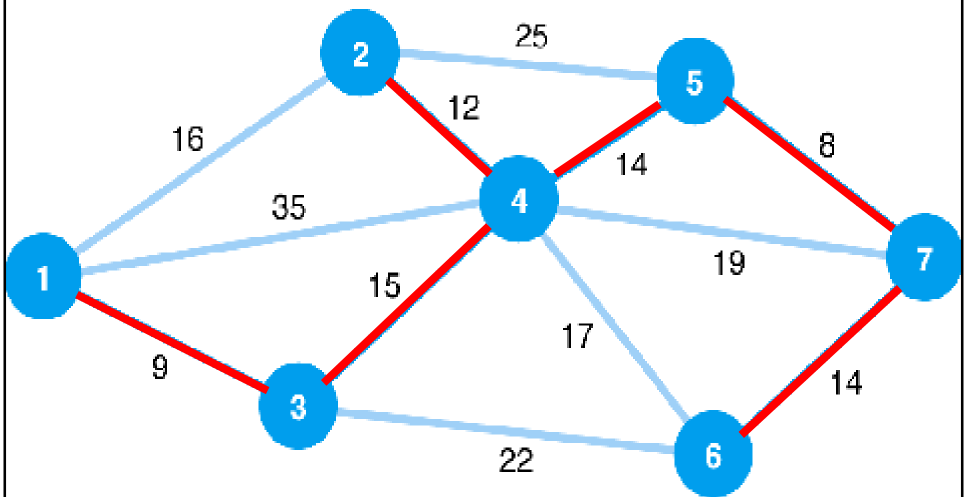
$$\text{Min} \quad \sum_{(i,j)\in A} c_{ij}x_{ij}$$

$$\text{s.t.} \quad \sum_{(i,j)\in A} x_{ij} = n-1$$

$$\sum_{(i,j)\in A_S} x_{ij} \le |S| - 1 \quad \text{for any set } S \text{ of nodes}$$

$$x_{ij} \in \{0,1\}, (i,j) \in A$$

The Minimal Spanning Tree Problem

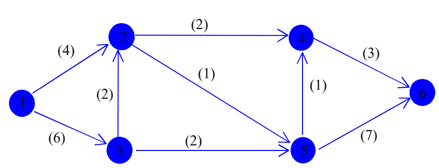


Optimal Solution to MST

Maximum Flow Problem

- In the maximum flow problem our goal is to send the largest amount of flow possible from a specified origin node to a specified destination node subject to arc capacities.
- This is a pure network flow problem (i.e., $g_{ij} = 1$) in which all the (real) arc costs are zero ($c_{ij} = 0$) and at least some of the arc capacities are finite.

Example

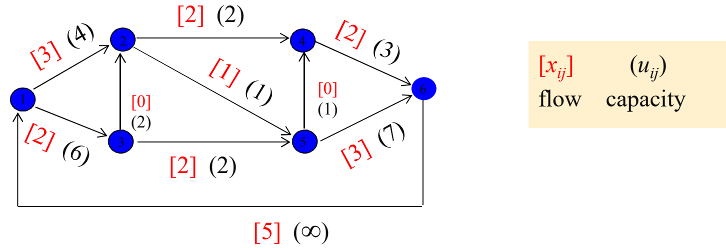


(u_{ij}) = arc capacity

Max Flow Example

Our goal is to send as much flow as possible from node 1 to node 6. (This is the same network we used in the shortest path discussion but now the arc labels represent capacities not costs.)

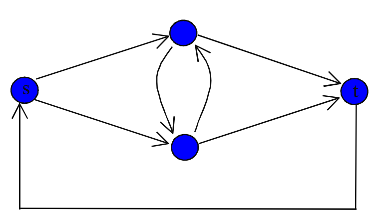
Solution



Maximum flow = 5

Max Flow Problem Formulation

- There are several different linear programming formulations.
- The one we will use is based on the idea of a "circulation."
- We suppose an artificial return arc from the destination to the origin with u_{ts} = +∞ and c_{ts} = 1.
- External flows (supplies and demands) are zero at all nodes.



Max Flow LP Model

Max x_{ts}

s.t. $\sum_{(i,j)\in A^+(i)} x_{ij} - \sum_{(j,i)\in A^-(i)} x_{ji} = 0, \quad \forall\, i \in N$

$0 \le x_{ij} \le u_{ij} \qquad \forall\,(i,j) \in A$

where x_{ts} is the flow on the circulation arc (t,s).

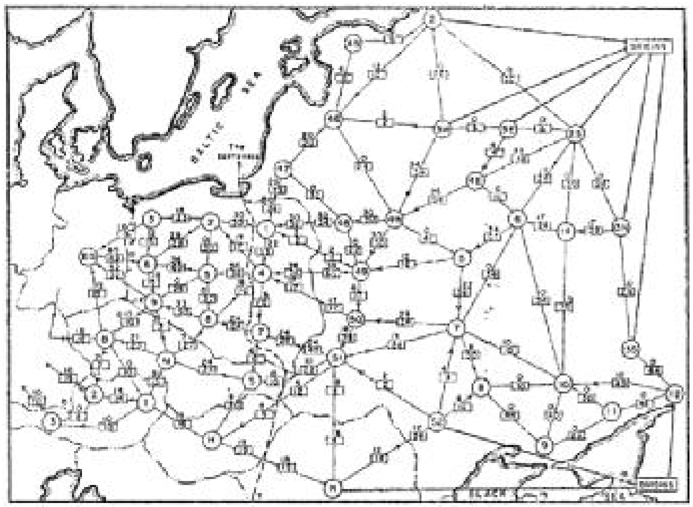
Max Flow-Another LP Model!

$$\max\ z = v$$

$$\text{s. t.}\ \sum_{j=1}^{n} x_{ij} - \sum_{k=1}^{n} x_{ki} = \begin{cases} v & (i=1) \\ 0 & (i=2,3,\cdots,n-1) \\ -v & (i=n) \end{cases}$$

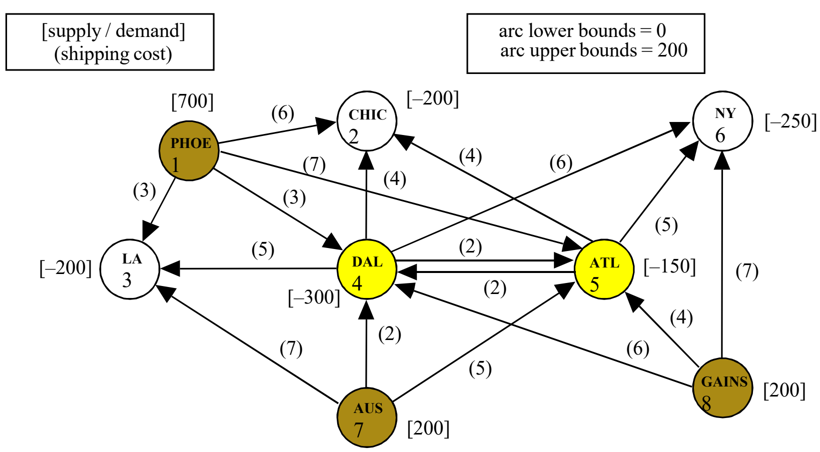
$$0 \le x_{ij} \le u_{ij} \quad (i,j=1,2,\cdots,n)$$

Max Flow: War Application!



Reference: *On the history of the transportation and maximum flow problems.* Alexander Schrijver in Math Programming, 91: 3, 2002.

Distribution Problem!



Min-Cost Flow Problem

Example: Distribution problem

- Warehouses store a particular commodity in Phoenix, Austin and Gainesville.
- Customers - Chicago, LA, Dallas, Atlanta, & New York

  Supply [s_i] at each warehouse i

  Demand [$-d_j$] of each customer j
- Shipping links depicted by arcs, flow on each arc is limited to 200 units.
- Dallas and Atlanta - transshipment hubs
- Per unit transportation cost (c_{ij}) for each arc

Problem: Determine optimal shipping plan that minimizes transportation costs

Notation for Min-Cost Flow Problem

In general: [supply/demand] on nodes
(shipping cost per unit) on arcs

In example: all arcs have an upper bound of 200
nodes labeled with a number 1,…,8

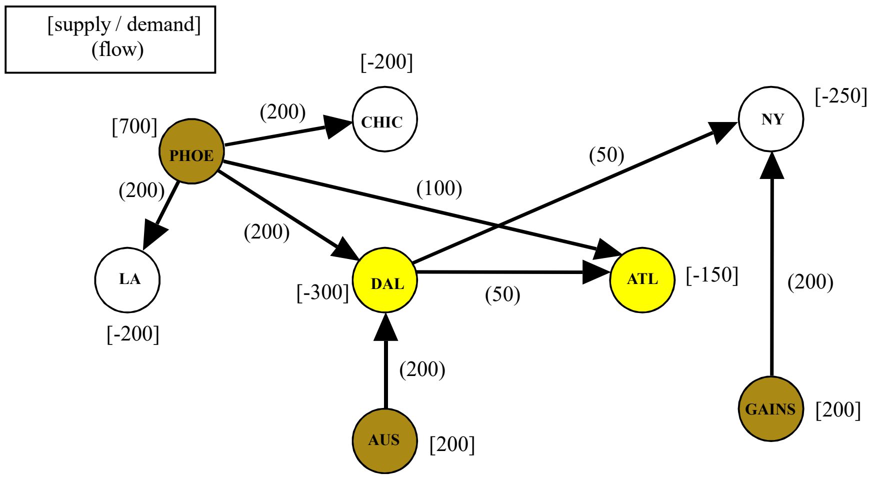
- Must indicate notation that is included in model:

  (c_{ij}) unit flow cost on arc (i, j)

  (u_{ij}) capacity (or simple upper bound) on arc (i, j)

  (g_{ij}) gain or loss on arc (i, j)
- All 3 could be included: (c_{ij}, u_{ij}, g_{ij})

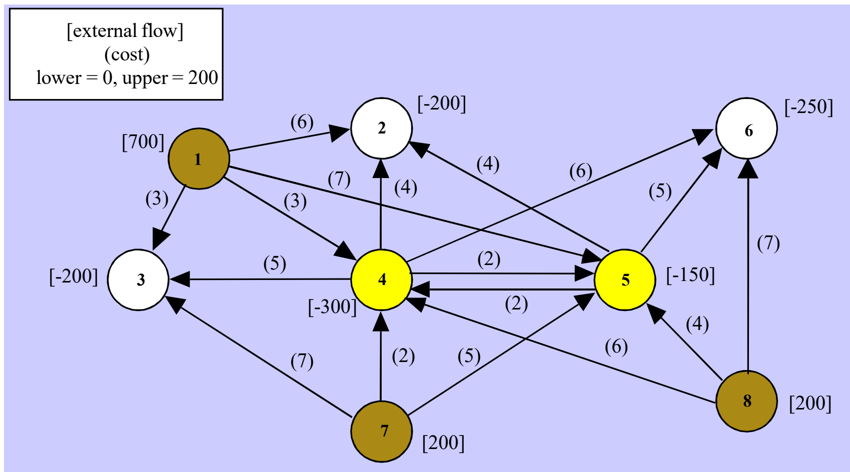
Solution to Distribution Problem



Characteristics of Network Flow Problems

- Conservation of flow at nodes. At each node
  flow in = flow out.
  At supply nodes there is an external inflow
  (positive)
  At demand nodes there is an external outflow
  (negative).
- Flows on arcs must obey the arc bounds; i.e.,
  lower bound & upper bound (capacity)
- Each arc has a per unit cost & the goal is to minimize
  total cost.

Distribution Network Used in Formulation



Pure Minimum Cost Flow Problem

$G = (N, A)$ → network with node set N and arc set A

Indices $i, j \in N$ denote nodes and $(i, j) \in A$ denote arcs

Originating set of arcs for node i (tails are i) is the forward star of i

$$A^+(i) = \{(i,j): (i,j) \in A\}$$

Terminating set of arcs for node i is the reverse star of i

$$A^-(i) = \{(j, i): (j, i) \in A\}.$$

In our example:

$A^+(1) = \{ (1,2), (1,3), (1,4), (1, 5) \}$

$A^-(1) = \emptyset$

$A^+(4) = \{ (4,2), (4,3), (4,5), (4,6) \}$

$A^-(4) = \{ (1,4), (5, 4), (7,4), (8,4) \}$

Flow balance equation for node *i* :

$$\sum_{(i,j)\in A^+} x_{ij} - \sum_{(j,i)\in A^-} x_{ji} = b_i$$

where b_i = positive for supply node *i*
= negative for demand node *i*
= 0 otherwise

Pure Min-Cost Flow Model

Indices/sets

$i, j \in N$	nodes
$(i, j) \in A$	arcs
$A^-(i)$	Arcs entering to *i*
$A^+(i)$	Arcs leaving *i*

Data

c_{ij}	unit cost of flow on (i,j)
l_{ij}	lower bound on flow (i,j)
u_{ij}	upper bound on flow (i,j)
b_i	external flow at node *i*

Total supply = total demand: $\sum_i bi = 0$

Pure Min-Cost Flow Model

Decision variables

x_{ij} = flow on arc (i,j)

Formulation for pure min-cost flow model

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

$s.t.$

$$\sum_{(i,j)\in A^+} x_{ij} - \sum_{(j,i)\in A^-} x_{ji} = b_i, \qquad \forall i \in N$$

$$l_{ij} \le x_{ij} \le u_{ij}, \qquad \forall\, (i,j) \in A$$

A Magic Property!!!

Decision variables are the flow variables x_{ij}

By examining the flow balance constraints we see that x_{ij} appears in exactly two of them:

$$\begin{array}{cl} x_{ij} & \\ 0 & \\ \vdots & \\ 0 & \\ +1 & \text{node } i \\ 0 & \\ \vdots & \\ 0 & \\ -1 & \text{node } j \\ 0 & \\ \vdots & \\ 0 & \end{array}$$

(or in the other order if $i > j$)

This structure is called **total unimodularity** and guarantees integer solutions

Observations from LP Model

- If we add the constraints we obtain zero on the left-hand side so the right-hand side must also be zero for feasibility.
- In particular, this means
  sum of supplies = sum of demands.
- Mathematically, we have one redundant constraint.
- Must be careful in interpreting shadow prices on the flow balance constraints.
- Cannot change only a supply or demand and have model make sense.

Generalized Minimum Cost Network Flow Model

- Only one modification to "pure" formulation
  - a possible gain (or loss) on each arc, denoted by g_{ij}
  - If g_{ij} = 0.95 then 100 units of flow leaves node *i* and 95 units arrive at node *j*

Generalized Formulation

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

$s.t.$

$$\sum_{(i,j)\in A^+} x_{ij} - \sum_{(j,i)\in A^-} g_{ji}x_{ji} = b_i, \qquad \forall i \in N$$

$$l_{ij} \leq x_{ij} \leq u_{ij}, \qquad \forall\, (i,j) \in A$$

Note that if $g_{ij} = 1 \;\forall\, (i,j) \in A$, then we obtain the "pure" model

The Multicommodity Flow Problem

On the Multicommodity Flow Problem O-D version

61

K origin-destination pairs of nodes

$(s_1, t_1), (s_2, t_2), \ldots, (s_K, t_K)$

Network $G = (N, A)$

d_k = amount of flow that must be sent from s_k to t_k.

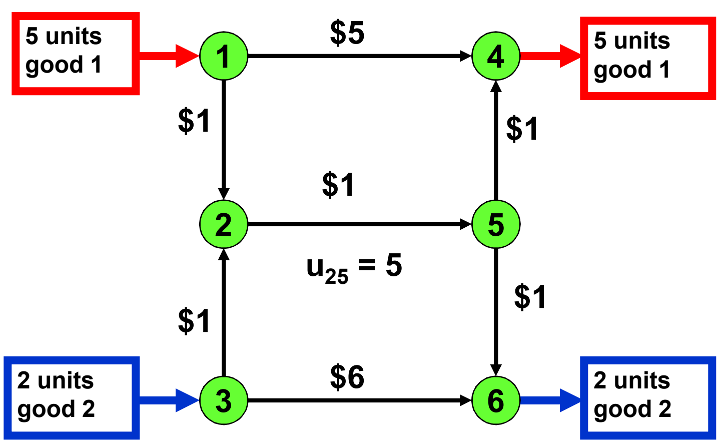
u_{ij} = capacity on (i, j) shared by all commodities

c_{ij}^k= cost of sending 1 unit of commodity k in (i, j)

x_{ij}^k= flow of commodity k in (i, j)

A Linear Multicommodity Flow Problem

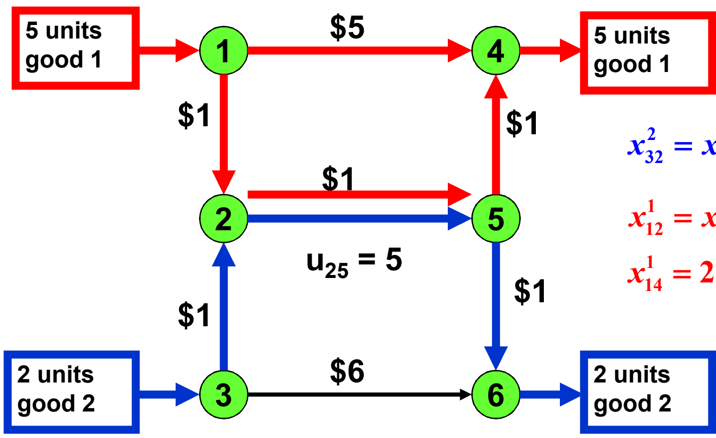
62



Quick exercise: determine the optimal multicommodity flow.

A Linear Multicommodity Flow Problem

63



$x^2_{32} = x^2_{25} = x^2_{56} = 2$

$x^1_{12} = x^1_{25} = x^1_{54} = 3$

$x^1_{14} = 2$

The Multicommodity Flow Formulation (LP)

64

Min $\sum_{(i,j)\in A} \sum_k c^k_{ij} x^k_{ij}$

$$\sum_j x^k_{ij} - \sum_j x^k_{ji} = \begin{cases} d_k & \text{if } i = s_k \\ -d_k & \text{if } i \in t_k \\ 0 & \textit{otherwise} \end{cases}$$

Supply/ demand constraints

$$\sum_k x^k_{ij} \le u_{ij} \quad \text{for all } (i,j) \in A$$

Capacity constraints

$$x^k_{ij} \ge 0 \quad \forall (i,j) \in A,\ k \in K$$

Assumptions (for now)

65

- Homogeneous goods. Each unit flow of commodity k on (i, j) uses up one unit of capacity on (i, j).
- No congestion. Cost is linear in the flow on (i, j) until capacity is totally used up.
- Fractional flows. Flows are permitted to be fractional.
- OD pairs. Usually a commodity has a single origin and single destination.

Application areas

66

Type of Network	Nodes	Arcs	Flow
Communic. Networks	O-D pairs for messages	Transmission lines	message routing
Computer Networks	storage dev. or computers	Transmission lines	data, messages
Railway Networks	yard and junction pts.	Tracks	Trains
Distribution Networks	plants warehouses,...	highways railway tracks etc.	trucks, trains, etc

On Fractional Flows

67

- In general, multicommodity flow problems have fractional flows, even if all data is integral.
- The integer multicommodity flow problem is difficult to solve to optimality.

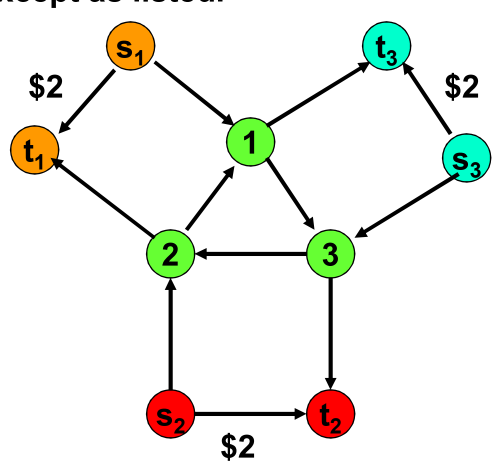
A fractional multicommodity flow

68

$u_{ij} = 1$ **for all arcs**

$c_{ij} = 0$ **except as listed.**

1 unit of flow must be sent from s_i to t_i for $i = 1, 2, 3$.



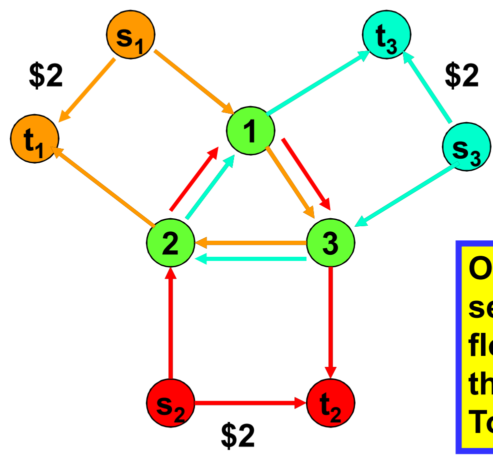
A fractional multicommodity flow

69

u_{ij} = 1 for all arcs

c_{ij} = 0 except as listed.

1 unit of flow must be sent from s_i to t_i for i = 1, 2, 3.



Optimal solution: send ½ unit of flow in each of these 15 arcs. Total cost = $3.

A formulation without OD pairs

70

$$\text{Minimize} \quad \sum_{1 \le k \le K} c^k x^k \qquad (17.1a)$$

$$\text{subject to} \quad \sum_{1 \le k \le K} x_{ij}^k \le u_{ij} \quad \text{for all } (i, j) \in A \qquad (17.1b)$$

$$Nx^k = b^k \quad \text{for } k = 1, 2, \ldots, k \qquad (17.1c)$$

$$0 \le x_{ij}^k \le u_{ij}^k \quad \begin{array}{l}\text{for all } (i, j) \in A \\ \text{for } k = 1, 2, \ldots, k\end{array} \qquad (17.1d)$$

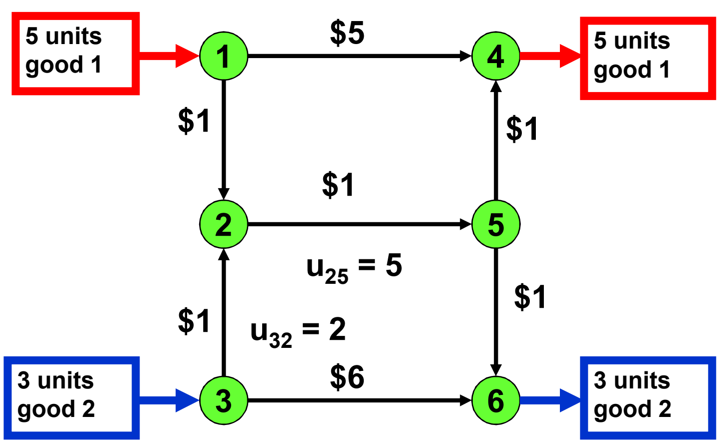
Another approach: path-based approach

71

- Represent flows from s_k to t_k as the sum of flows in paths.
- The resulting LP may have an exponential number of columns
- Use "column generation" to solve the LP.

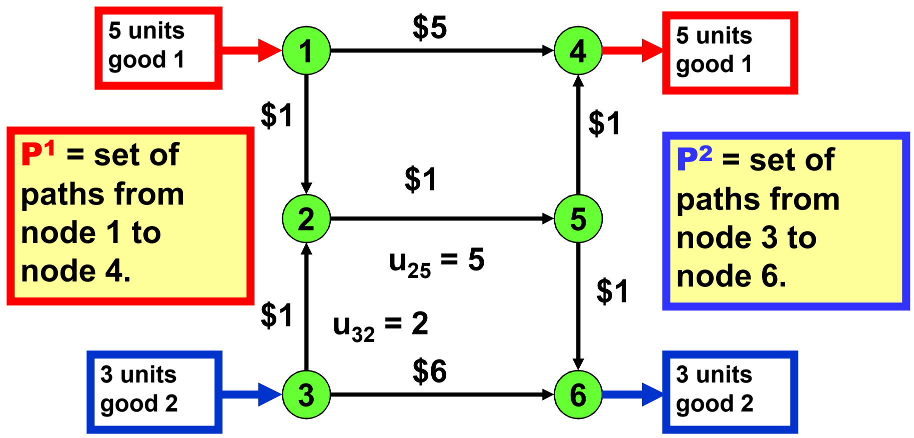
A Linear Multicommodity Flow Problem

72



A Linear Multicommodity Flow Problem

73



P^1 = {1-4, 1-2-5-4}

P^2 = {3-6, 3-2-5-6}

A path based formulation

74

f(P) = flow in path P

c(P) = cost of path P

c(1-4)	=	5
c(1-2-5-4)	=	3
c(3-6)	=	6
c(3-2-5-6)	=	3

Minimize 5 f(1-4) + 3 f(1-2-5-4) + 6 f(3-6) + 3 f(3-2-5-6)

subject to

f(1-4) + f(1-2-5-4) = 5

f(3-6) + f(3-2-5-6) = 3

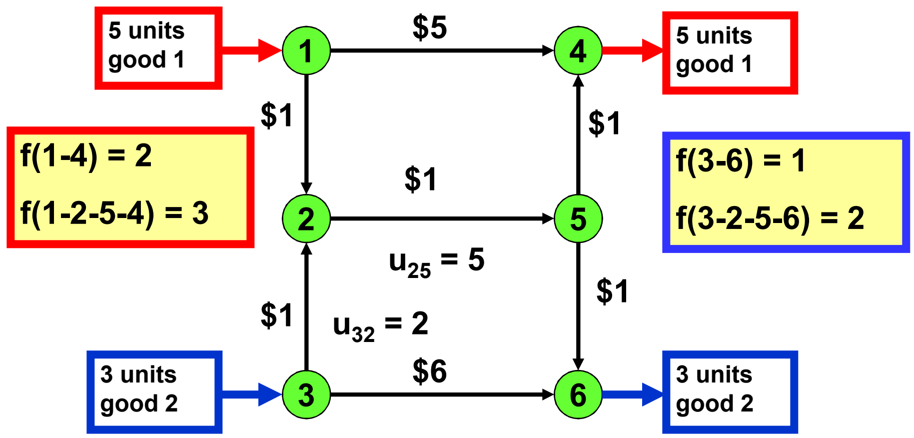
f(1-2-5-4) + f(3-2-5-6) ≤ u_{25} = 5

f(3-2-5-6) ≤ u_{32} = 2

f(P) ≥ 0 for all paths P

Optimal solution for the path based version

75



The path based LP can be solved using the simplex method.

General formulation for the path based version

76

Let $\mathbf{P}^k$ = set of directed paths from s_k to t_k

Let $c^k(P)$ = cost of path $P \in \mathbf{P}^k$.

Let $f(P)$ = flow on path P.

$$\text{Let } \delta_{ij}(P) = \begin{cases} 1 & \text{if } (i,j) \in P \\ 0 & \text{otherwise} \end{cases}$$

Master Problem

$$\text{Minimize} \quad \sum_k \sum_{P \in \mathbf{P}^k} c^k(P) f(P)$$

$$\sum_k \sum_{P \in \mathbf{P}^k} \delta_{ij}(P) f(P) \le u_{ij} \qquad \text{for all } (i,j) \in A$$

$$\sum_{P \in \mathbf{P}^k} f(P) = d^k \qquad \text{for } k = 1 \text{ to } K$$

$$f(P) \ge 0 \qquad \text{for } P \in \bigcup_{k=1}^{K} \mathbf{P}^k$$

77

Minimize $\sum_{k} \sum_{P \in \boldsymbol{P}^k} c^k(P) f(P)$

$$\sum_{k} \sum_{P \in \boldsymbol{P}^k} \delta_{ij}(P) f(P) \le u_{ij} \quad \text{for all } (i,j) \in A$$

$$\sum_{P \in \boldsymbol{P}^k} f(P) = d^k \quad \text{for } k = 1 \text{ to } K$$

$$f(P) \ge 0 \quad \text{for } P \in \bigcup_{k=1}^{K} \boldsymbol{P}^k$$

bundle constraints: one for each capacitated arc.

supply demand constraints: one for commodity.

variables: one for each path from origin to destination

Solution Approaches?
Let's postpone them!

Decomposition based approaches

79

Price directed decomposition.

Focus on prices or tolls on the arcs. Then solve the problem while ignoring the capacities on arcs.

Resource directive decomposition.

Allocate flow capacity among commodities and solve

Simplex based approaches

Try to speed up the simplex method by exploiting the structure of the MCF problem.

Optimality Conditions: Partial Dualization

80

Theorem. The multicommodity flow $x = (x^k)$ is an optimal multicommodity flow for (17) if there exists non-negative prices $w = (w_{ij})$ on the arcs so that the following is true

1. If $w_{ij} > 0$, then $\sum_k x_{ij}^k = u_{ij}$

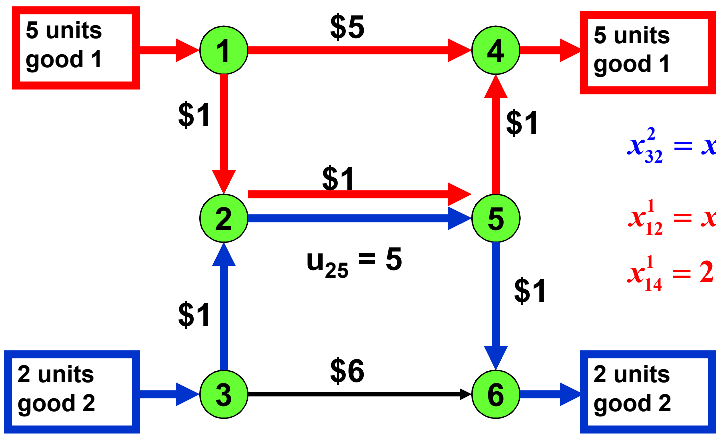
2. The flow x^k is optimal for the k-th commodity if c^k is replaced by $c^{w,k}$, where

$$c_{ij}^{w,k} = c_{ij}^k + w_{ij}$$

Recall: x^k is optimal for the k-th commodity if there is no negative cost cycle in the kth residual network.

A Linear Multicommodity Flow Problem

81



$x_{32}^{2} = x_{25}^{2} = x_{56}^{2} = 2$

$x_{12}^{1} = x_{25}^{1} = x_{54}^{1} = 3$

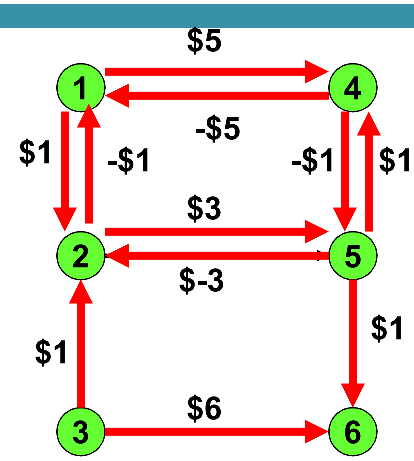
$x_{14}^{1} = 2$

Set $w_{2,5} = 2$

Create the residual networks:

The residual network for commodity 1

82

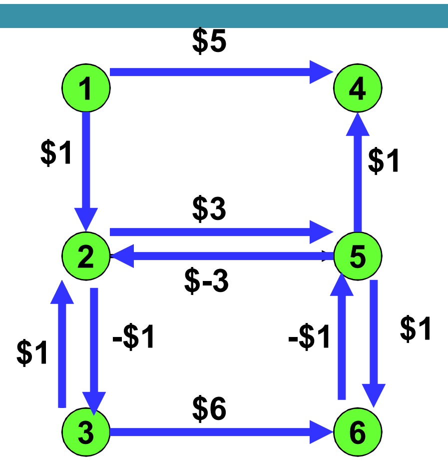


Set $w_{2,5}$ = \$2

There is no negative cost cycle.

The residual network for commodity 2

83



Set $w_{2,5}$ = $2

There is no negative cost cycle.

Optimality Conditions: full dualization

84

One can also define node potentials π so that the reduced cost

$$c_{ij}^{\pi,k} = c_{ij}^{k} + w_{ij} - \pi_i^k + \pi_j^k \geq 0$$

for all $(i, j) \in A$ and $k = 1, \ldots, K$

This combines optimality conditions for min cost flows with the partial dualization optimality conditions for multicommodity flows.

Lagrangean relaxation for multicommodity flows

85

Min $\sum_{(i,j)\in A} \sum_k c_{ij}^k x_{ij}^k$

$$\sum_j x_{ij}^k - \sum_j x_{ji}^k = \begin{cases} d_k & \text{if } i = s_k \\ -d_k & \text{if } i \in t_k \\ 0 & otherwise \end{cases}$$

Supply/ demand constraints

$$\sum_k x_{ij}^k \le u_{ij} \quad \text{for all } (i,j) \in A$$

Bundle constraints

$$x_{ij}^k \ge 0 \quad \forall (i,j) \in A,\ k \in K$$

Lagrangean relaxation for multicommodity flows

86

Min $\sum_{(i,j)\in A} \sum_k c_{ij}^k x_{ij}^k + \sum_{(i,j)\in A} w_{ij} (\sum_k x_{ij}^k - u_{ij})$

$$\sum_j x_{ij}^k - \sum_j x_{ji}^k = \begin{cases} d_k & \text{if } i = s_k \\ -d_k & \text{if } i \in t_k \\ 0 & otherwise \end{cases}$$

Supply/ demand constraints

$$x_{ij}^k \ge 0 \quad \forall (i,j) \in A,\ k \in K$$

Penalize the bundle constraints.

Relax the bundle constraints.

Lagrangean relaxation for multicommodity flows

87

$$L(w) = \text{Min} \sum_{(i,j)\in A} \sum_k (c_{ij}^k + w_{ij}) x_{ij}^k - \sum_{(i,j)\in A} w_{ij} u_{ij}$$

$$\sum_j x_{ij}^k - \sum_j x_{ji}^k = \begin{cases} d_k & \text{if } i = s_k \\ -d_k & \text{if } i \in t_k \\ 0 & \text{otherwise} \end{cases}$$

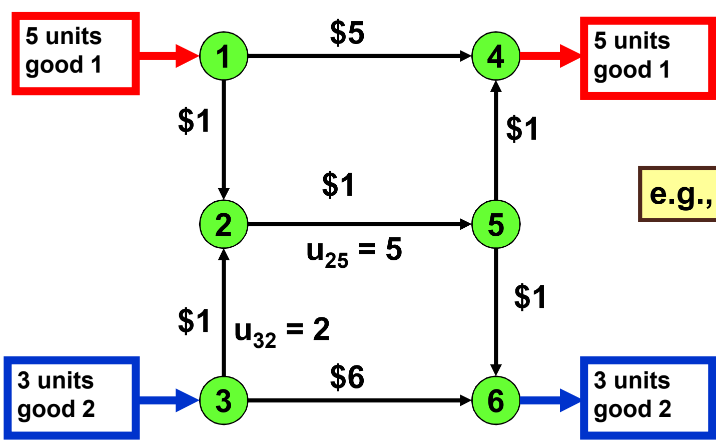
Supply/ demand constraints

$$x_{ij}^k \geq 0 \quad \forall (i,j) \in A, \; k \in K$$

Simplify the objective function.

Subgradient Optimization for solving the Lagrangean Multiplier Problem

88

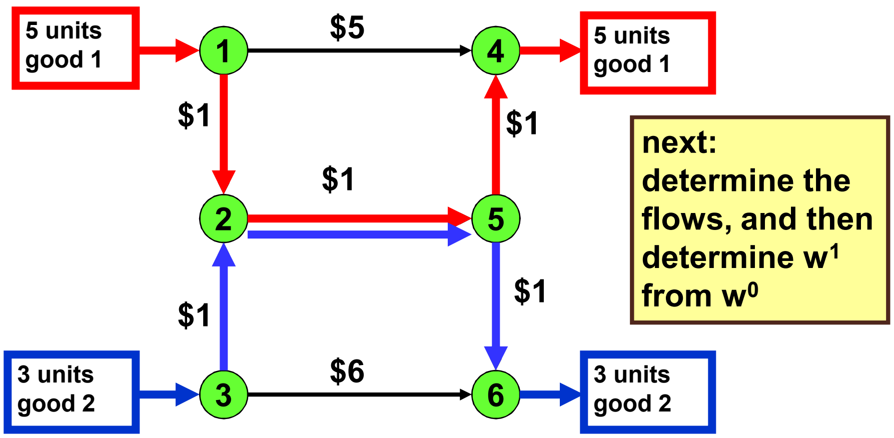


e.g., set $w^0 = 0$.

Choose an initial value w^0 of the "tolls" w, and find the optimal solution for L(w).

Subgradient Optimization for solving the Lagrangean Multiplier Problem

89



The flow on (2,5) = 8 > u_{25} = 5.

The flow on (3,2) = 3 > u_{32} = 2.

Choosing a search direction

90

$$r^+ = \max(0, r)$$

$$y_{ij} = \sum_k x_{ij}^k \quad \text{= flow in arc (i,j)}$$

$$w_{ij}^{q+1} = [w_{ij}^q + \theta_q(y_{ij} - u_{ij})]^+$$

$(y\text{-}u)^+$ is called the ***search direction***.

$$w_{25}^1 = [w_{25}^0 + \theta_0(8-5)]^+ = 3\theta_0$$

θ_q is called the ***step size***.

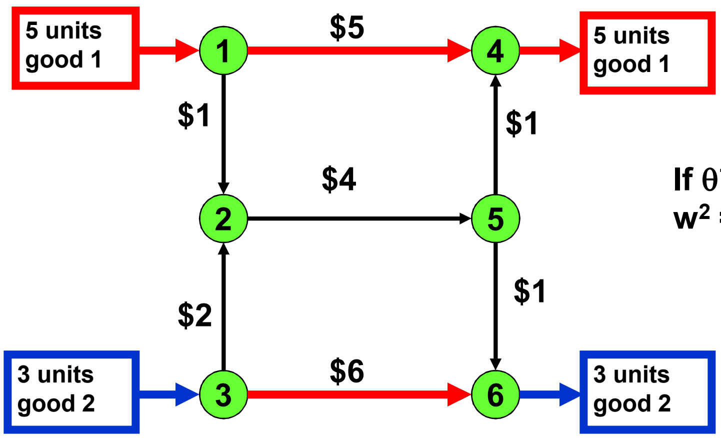
$$w_{32}^1 = [w_{32}^0 + \theta_0(3-2)]^+ = \theta_0$$

So, if we choose $\theta_0 = 1$, then $w_{25}^1 = 3$ and $w_{32}^1 = 1$

Then solve $L(w^1)$.

Solving L(w^1)

91



If $\theta^1 = 1$, then $w^2 = 0$.

$$w_{25}^2 = [w_{25}^1 + \theta_1(0-5)]^+ = [3 - 5\theta_1]^+$$

$$w_{32}^2 = [w_{32}^1 + \theta_1(0-2)]^+ = [1 - 2\theta_1]^+$$

Comments on the step size

92

- The search direction is a good search direction.
- But the step size must be chosen carefully.
- Too large a step size and the solution will oscillate and not converge
- Too small a step size and the solution will not converge to the optimum.

On choosing the step size

93

The step size θ_q should be chosen so that

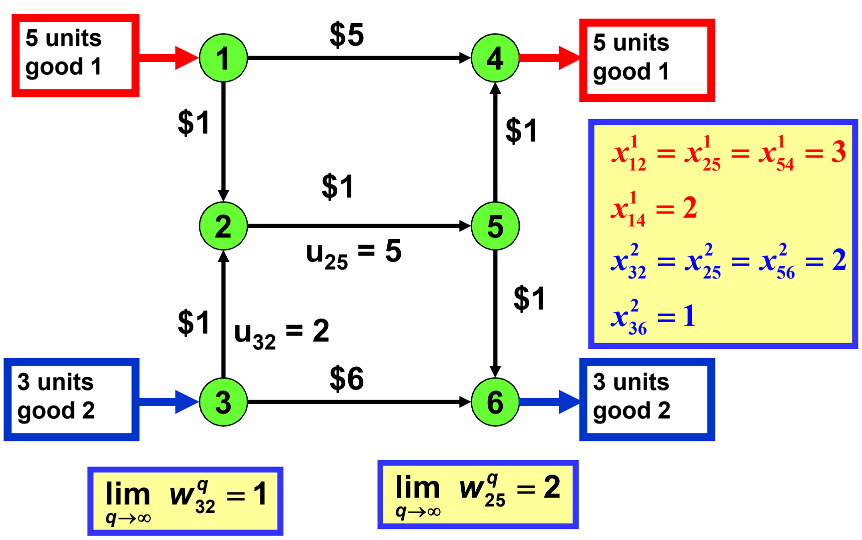
$$\lim_{q \to \infty} \theta_q = 0 \quad \text{and} \quad \sum_{q=1}^{\infty} \theta_q = \infty \qquad (1)$$

e.g., take $\theta_q = 1/q$.

Theorem. **If the step size is chosen as on the previous slides, and if (θ_q) satisfies (1), then the w^q converges to the optimum for the Lagrangean dual.**

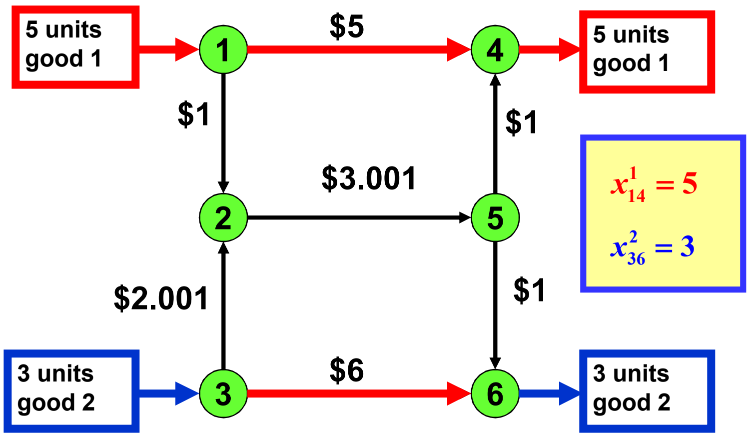
The optimal multipliers and flows.

94



Suppose that $w_{32} = 1.001$ and $w_{25} = 2.001$

95



Conclusion: Near Optimal Multipliers do not always lead to near optimal (or even feasible) flows.

Summary of MCF

96

- Applications
- Formulations
  - Link-Node formulation
  - Path-based formulation
- Optimality Conditions
- Lagrangean Relaxation
  - subgradient optimization