Optimization 2

(for FERM program)

Prof. DrSc. Nguyen Dinh

Department of mathematics
INTERNATIONAL UNIVERSITY, VNU-HCM

September 19, 2021

Part I

APLLIED LINEAR PROGRAMMING

Contents

A quick review on Linear Programming (LP)

∇ Linear programming is a class of optimization problems that minimizes or maximizes a linear objective function under some linear constraints.

∇ The problems concern many topics:

- Management Sciences
- Decision Making
- Computer science
- Sciences and Technology (designing problems, etc.)
- Physics

General LP problem

A Linear Program is the problem of maximizing (or minimizing) a linear function subject to a finite number of linear constraints:

$$
\begin{array}{lll}
\text{(LP)} & \text{Minimize} & \sum_{j=1}^{n} c_j x_j \\
 & \text{subject to} & \sum_{j=1}^{n} a_{ij} x_j \leq b_i, i = 1, 2, ..., m \\
 & & x_j \geq 0, j = 1, 2, ..., n.
\end{array}
$$

General LP problem

or in the matrix form:

$$\text{(LP)} \quad \text{Minimize} \quad c^T x$$

$$\text{subject to} \quad \begin{aligned} &Ax \leq b \\ &x \geq 0, \end{aligned}$$

where

$$A = \begin{pmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{pmatrix}, x = \begin{pmatrix} x_1 \\ x_2 \\ \dots \\ x_n \end{pmatrix}, c = \begin{pmatrix} c_1 \\ c_2 \\ \dots \\ c_n \end{pmatrix}, b = \begin{pmatrix} b_1 \\ b_2 \\ \dots \\ b_m \end{pmatrix},$$

and $x \geq 0$ means that $x_j \geq 0$ for all $j = 1, 2, \cdots, n$.

1. Network flow problems

Many linear programming problems can be viewed as network flow problems, or, in other words, network problems can be modeled as linear programming problems. Network flow problems are the ones of of minimizing "transportation cost" of moving materials through a network from several locations of given sources to meet demands for materials at several other locations. For these problems, the simplex method has a very simple description.

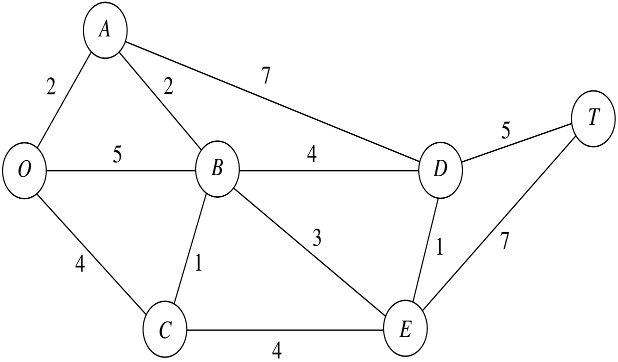
Network flow problems are also called transhipment problems or minimal cost network flow problems.

Introductory examples

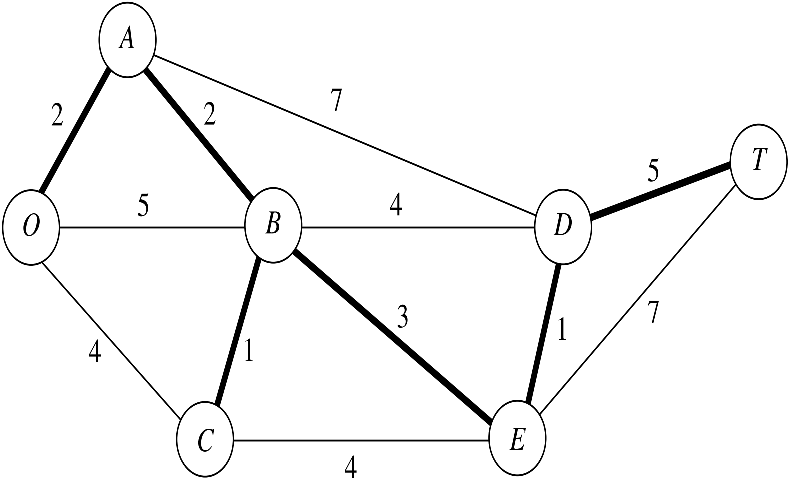
Introductory examples

The Seervada Park management (see Sec. 9.1) needs to determine under which roads telephone lines should be installed to connect all stations with a minimum total length of line. Using the data given in Fig. 9.1, we outline the step-by-step solution of this problem.

Nodes and distances for the problem are summarized below, where the thin lines now represent *potential* links.



Introductory examples



Networks. [also, directed networks, graphs, directed graphs]

A network $\mathcal{G}$ is a pair $(\mathcal{N}, \mathcal{A})$, denoted by $\mathcal{G} = (\mathcal{N}, \mathcal{A})$, where $\mathcal{N}$ is a set of nodes, $\mathcal{N} = \{1, 2, , \cdots, n\}$, and $\mathcal{A}$ is a set of arcs

$$\mathcal{A} = \{(i, j), (k, l), \cdots, (s, t)\}.$$

An arc (i, j) is an ordered pair joining the nodes i and j of $\mathcal{N}$, directed from i to j.

Assume that the network $\mathcal{G}$ (in consideration) has n nodes and m arcs.

Example 1. Figure 1 describes a network $\mathcal{G}$ with $\mathcal{N} = \{1, 2, 3, 4\}$ and

$$\mathcal{A} = \{(1,2), (2,4), (2,3), (3,4), (1,3), (3,2), (4,1)\}.$$

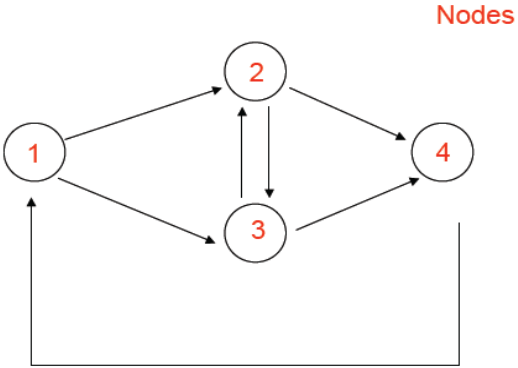


Figure 1

Example 1. Figure 1 describes a network $\mathcal{G}$ with $\mathcal{N} = \{1, 2, 3, 4\}$ and

$$\mathcal{A} = \{(1, 2), (2, 4), (2, 3), (3, 4), (1, 3), (3, 2), (4, 1)\}.$$

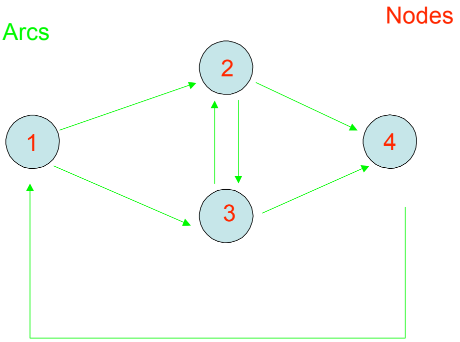
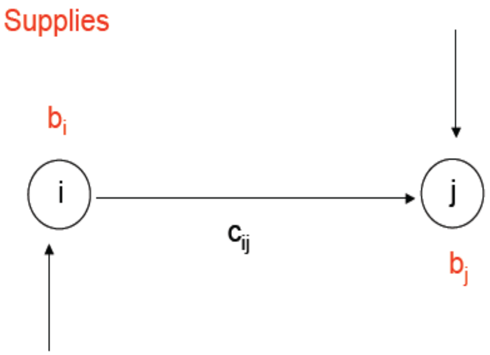


Figure 1

(α) **Supply and demand nodes.** At each node i of $\mathcal{G}$ we associate a number b_i which is the available supply of material (commodity) if $b_i > 0$ or, the demand for the material if $b_i < 0$.

- If $b_i > 0$, i is called a source or a supply node,
- If $b_i < 0$, i is called a sink or a demand node,
- If $b_i = 0$, i is called an intermediate node.

Networks



(β) **Flow and shipping cost.** For each arc (i, j), c_{ij} denotes the unit shipping cost along this arc and x_{ij} denotes the amount of flow on the arc (i, j). Of course, the flows should satisfy the conditions $x_{ij} \geq 0$.

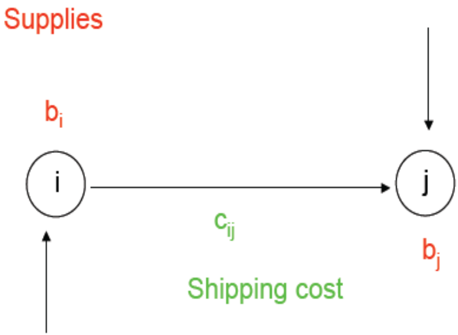
So the cost of the flow along the arc (i, j) is

$$c_{ij}x_{ij} \text{ for each } (i, j) \in \mathcal{A},$$

and hence, the total cost of the flows through the network is

$$\sum_{(i,j)\in\mathcal{A}} c_{ij}x_{ij}. \tag{1}$$

Networks



Assumption:

$$\sum_{i=1}^{n} b_i = 0 \qquad (2)$$

(the total supply equals the total demand).
The assumption seems unrealistic! The lack of realism in this assumption is obvious, for example, the total demand for oranges matches their supply precisely!

However, in the case where (2) does not hold, e.g., $\sum_{i=1}^{n} b_i > 0$, we can add to the network a dummy node $n+1$ with the demand $b_{n+1} = -\sum_{i=1}^{n} b_i$ and arcs with zero cost from each supply node to the new (dummy) node.

(γ) **Total flow out/into a node.** At each node $k \in \mathcal{N}$, the total flow *into* the node i is:

$$\sum_{(k,i)\in\mathcal{A}} x_{ki} = \sum_{k=1}^{n} x_{ki}.$$

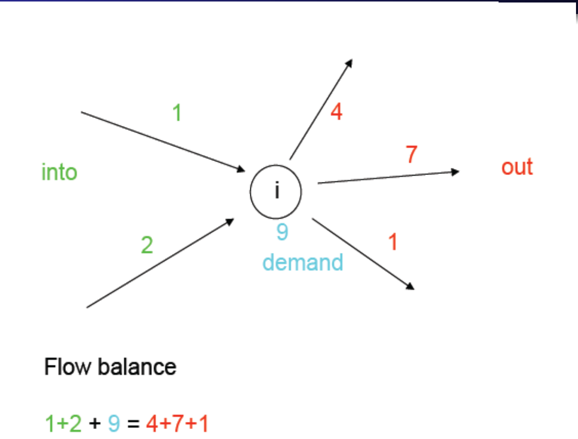
The total flow *out* from the node i is:

$$\sum_{(i,j)\in\mathcal{A}} x_{ij} = \sum_{j=1}^{n} x_{ij}.$$

The constraints on the decision variables x_{ij} are that they ensure the flow balance at each node. That is,

$$\sum_{k=1}^{n} x_{ki} + b_i = \sum_{j=1}^{n} x_{ij}, \ \forall i = 1, 2, \cdots, n. \tag{3}$$

Networks



The difference between the total flow into and the total flow out from a node must equal to the demand of this node. The constraints (3) are called flow conservation (after the transshipment, the material left at node i is zero).

(δ) **The (minimal cost) network flow problem:** may be stated as follows:

"Ship the available supply through the network to satisfy demand at minimum cost."

Mathematically, the problem becomes (summations are taken over all existing arcs)

$$\begin{array}{ll} \text{Minimize} & \sum_{i=1}^{n}\sum_{j=1}^{n} c_{ij}x_{ij} \\ \text{subject to:} & \sum_{k=1}^{n} x_{ki} + b_i = \sum_{j=1}^{n} x_{ij},\ \forall i = 1, 2, \cdots, n, \\ & x_{ij} \geq 0,\ \ i, j = 1, 2, \cdots, n. \end{array}$$

Example 2. (Example 1 revisited, see Figure 1b)
Nodes: 1,2, 3, 4.
Arcs: $(1,2),(1,3),(2,3),(2,4),(3,2),(3,4),(4,1)$.

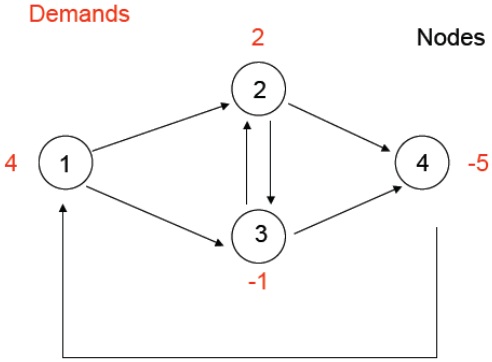


Figure 1b

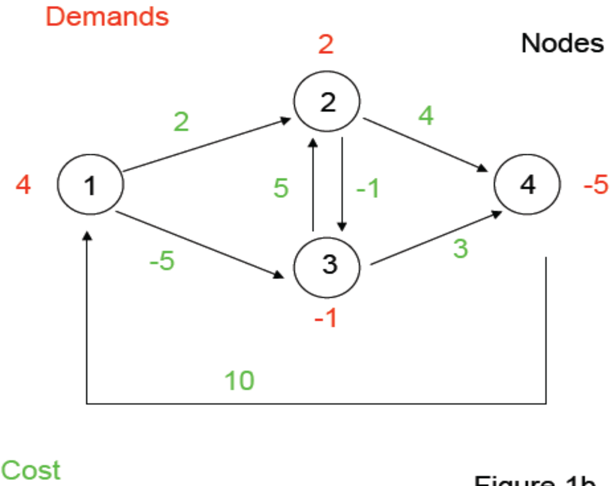


Figure 1b

$$\begin{aligned} c^T &= (c_{12}, c_{13}, c_{23}, c_{24}, c_{32}, c_{34}, c_{41}) \\ &= (2, -5, -1, 4, 5, 3, 10), \\ x^T &= (x_{12}, x_{13}, x_{23}, x_{24}, x_{32}, x_{34}, x_{41}). \end{aligned}$$

The cost functional (objective function) to be minimized is

$$\begin{aligned} Z = \; & 2x_{12} - 5x_{13} - 1x_{23} \\ & +4x_{24} + 5x_{32} + 3x_{34} + 10x_{41}. \end{aligned}$$

The constraints:

$$\begin{aligned} -x_{12} - x_{13} + x_{14} &= -4 \\ x_{12} - x_{23} - x_{24} + x_{32} &= -2 \\ x_{13} + x_{23} - x_{32} - x_{34} &= 1 \\ x_{24} + x_{34} - x_{41} &= 5. \end{aligned}$$

The vector

$$b^T := (-b_1, -b_2, \cdots, -b_n)$$

is called the demand vector (of the network).

The network flow problem can be written in the matrix form as follows:

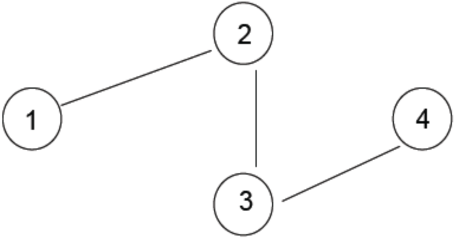
$$\begin{array}{ll} \text{Minimize} & c^T x \\ \text{subject to} & Ax = b, \\ & x \geq 0, \end{array}$$

where c and x are the vectors defined as above while A is the matrix of coefficients of the system of equality constraints given in the previous page.

So, the **network flow problem** is a **linear programming problem**.

Let $\mathcal{G} = (\mathcal{N}, \mathcal{A})$ be a network.

• A *path* connecting nodes u and v is a network with nodes $w_1, w_2, \cdots, w_k$ and arcs $e_1, e_2, \cdots, e_{k-1}$ such that $w_1 = u$, $w_k = v$, and two end points of each e_i are w_i and w_{i+1} (see Figure 1c). An arc (w_i, w_{i+1}) is called a *forward arc*. Otherwise, it is called a *backward arc*.



- *A cycle* (Figure 1d) is a path which connects a node to itself. A cycle may have two different nodes. [How to orient a cycle?]

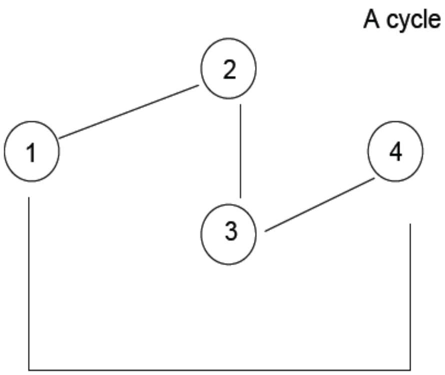


Figure 1d

A cycle

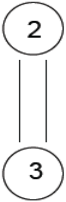


Figure 1d

• A network is called *connected* if for every two of its nodes there is a path connecting them. It is called *acycle* if it contains no cycle.

• A *tree* (of a network) is a connected and acycle network that may involve a subset of nodes of a network.
A *spanning tree* is a tree that links all nodes of a network.

• A *feasible tree solution* is a feasible solution x associated with a spanning tree T in such a way that $x_{ij} = 0$ if $(i,j) \notin T$ and $x_{ij} \geq 0$ if $(i,j) \in T$. For example, the tree T in Figure 2b determines a feasible tree solution.

2. Terminologies of networks

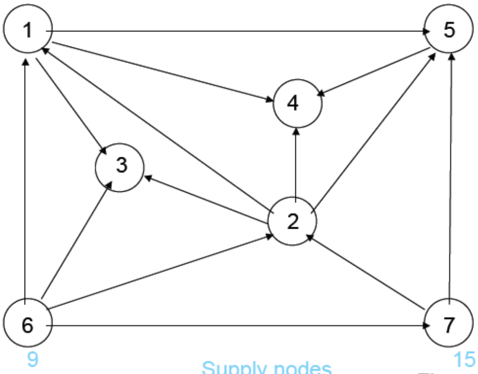


Figure 2a

2. Terminologies of networks

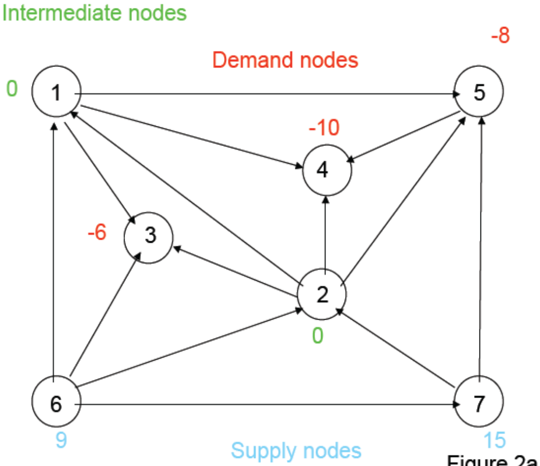
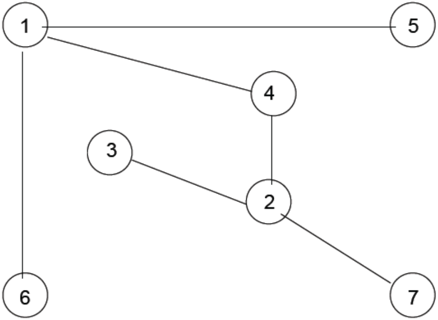


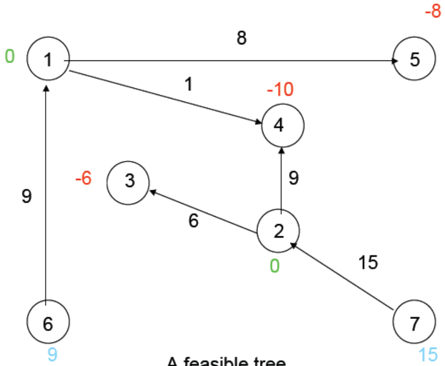
Figure 2a

2. Terminologies of networks



A tree

Figure 2b



A feasible tree

Figure 2b

- *Degree* of a node is the number of arc(s) connected to it. Each node of degree one is called a *leaf*. In Figure 2b, nodes $5, 7, 3, 6$ are leaves of the tree T.

• *Degree* of a node is the number of arc(s) connected to it. Each node of degree one is called a *leaf*. In Figure 2b, nodes $5, 7, 3, 6$ are leaves of the tree T.

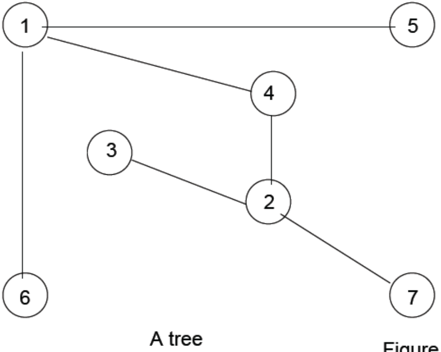


Figure 2b

Remark.

(i) For a feasible tree solution x, the variables x_{ij} with $(i,j) \in T$ are called *basic variables*. *Nonbasic variables* are those x_{ij} with $(i,j) \notin T$.

(ii) For a tree (of a network), the following properties hold:

(ii) For a tree (of a network), the following properties hold:

(a) For any different nodes i, j of T, there exists a unique path connecting these nodes.

(b) When adding a new arc e whose ends are nodes of T to T, the new network has exactly one cycle. We will denote this unique cycle by $C(T, e)$. By convention, we orientate the cycle in such a way that e is a forward arc.

• *Network simplex method*, as the name suggests, is an interpretation of the LP simplex method applied to the (minimum cost) network flow problems. The link can be considered as the relation between a feasible solution x and a feasible tree solution (associated to x).

• Given a feasible tree solution, the idea of the method is to improve it to get a better one (in the sense that the cost corresponding to the latter is smaller than that of the original one). We begin with some facts related to entering and leaving variables.

3.1. Leaving and entering variables

Let (P) be the network flow problem described in Section 1 and let $\mathcal{G} = (\mathcal{N}, \mathcal{A})$ be the corresponding network. Assume that $\mathcal{G}$ is connected.

Let T be a feasible tree solution.

Network simplex method will move from one feasible tree solution to another feasible tree solution which improves the objective function, using only cycles formed by adding an arc to T.

Let (i, j) be a nonbasic arc, that is $(i, j) \notin T$ (x_{ij} is a nonbasic variable).

- The arc (i, j), together with other arcs in T, forms a unique cycle, denoted by $C(T, ij)$. The direction on $C(T, ij)$ is such that (i, j) is a forward arc. Note that $x_{ij} = 0$ (see Figure 2c).

- The arc (i,j), together with other arcs in T, forms a unique cycle, denoted by $C(T,ij)$. The direction on $C(T,ij)$ is such that (i,j) is a forward arc. Note that $x_{ij} = 0$ (see Figure 2c).

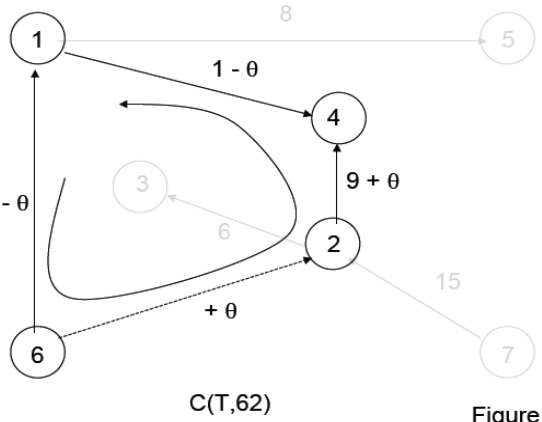


Figure 2c

• Let B be the set of all backward arcs in the cycle $C(T, ij)$ and let F be the set of all forward arcs on $C(T, ij)$.

• Let us suppose that the flow in each arc of this cycle is only one unit. The cost for the transshipment around the cycle is

$$\bar{c}_{ij} := \sum_{(k,l)\in F} c_{kl} - \sum_{(k,l)\in B} c_{kl}. \tag{4}$$

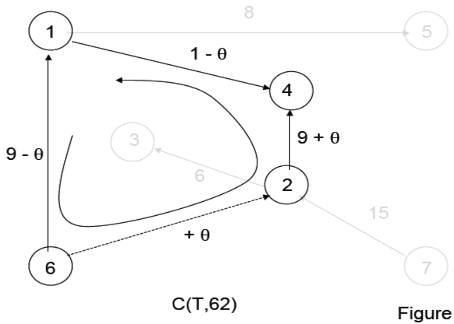


Figure 2c

This number, $\bar{c}_{ij}$, is called the cost of the cycle $C(T, ij)$. It is also called the reduced cost corresponding to the variable x_{ij}.

• Suppose that $\bar{c}_{ij} < 0$. This means that in this case, if we ship around the cycle (following the direction of forward arcs) one unit of commodity the cost will reduces $\bar{c}_{ij}$ unit.

So if we ship θ units of commodity around the cycle (while keeping all the other flows of other arcs on T the same as before and $x_{kl} = 0$ for all $(k, l) \notin T$) then the cost will reduce an amount:

$$\theta \bar{c}_{ij} = \theta\Big(\sum_{(k,l)\in F} c_{kl} - \sum_{(k,l)\in B} c_{kl} \Big).$$

The process (of doing this) means that we increase x_{ij} from 0 to $\theta > 0$.

• It is clear that we can not increase $\theta > 0$ too much. As in the case of LP problem, we have to keep all other flows in the arcs of the cycle $C(T, ij)$ nonnegative. Note that when we increase $x_{ij} = \theta > 0$, the old basic variables on the cycle will change following the formula (by the flow conservation):

$$\bar{x}_{kl} = \begin{cases} x_{kl} + \theta & \text{if } (k,l) \in F, \\ x_{kl} - \theta & \text{if } (k,l) \in B, \\ x_{kl} & \text{otherwise} \end{cases} \tag{5}$$

(and so, $x_{ij} = x_{ij} + \theta = 0 + \theta = \theta > 0$).

• The maximum increment of x_{ij} (i.e., θ)

- It is clear that we can not increase $\theta > 0$ too much. As in the case of LP problem, we have to keep all other flows in the arcs of the cycle $C(T, ij)$ nonnegative. Note that when we increase $x_{ij} = \theta > 0$, the old basic variables on the cycle will change following the formula (by the flow conservation):

$$\bar{x}_{kl} = \begin{cases} x_{kl} + \theta & \text{if } (k,l) \in F, \\ x_{kl} - \theta & \text{if } (k,l) \in B, \\ x_{kl} & \text{otherwise} \end{cases} \tag{5}$$

(and so, $x_{ij} = x_{ij} + \theta = 0 + \theta = \theta > 0$).

- The maximum increment of x_{ij} (i.e., θ) should be chosen as

$$\theta = \theta^* = \min_{(k,l) \in B} x_{kl}. \tag{6}$$

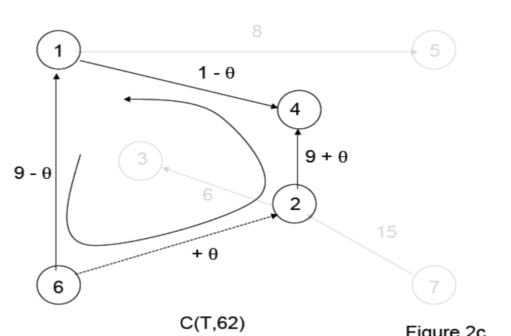


Figure 2c

- Let $(u, v) \in B$ be such that

$$x_{uv} = \theta^* = \min_{(k,l)\in B} x_{kl}. \tag{7}$$

Choose x_{uv} as the leaving variable. Note that after shipping θ^* more units through the cycle, the flow in (u, v) is zero, that is, $x_{uv} = 0$.

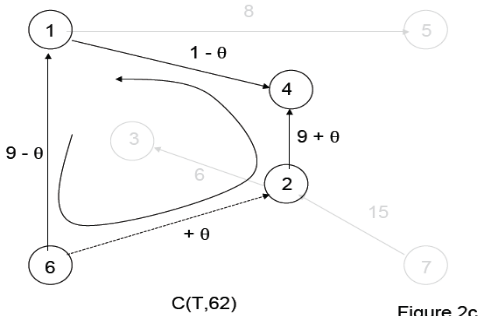


Figure 2c

- Let $(u, v) \in B$ be such that

$$x_{uv} = \theta^* = \min_{(k,l)\in B} x_{kl}. \tag{7}$$

Choose x_{uv} as the leaving variable. Note that after shipping θ^* more units through the cycle, the flow in (u, v) is zero, that is, $x_{uv} = 0$.

- The entering variable in this case is x_{ij} (note that we are in the case where $\bar{c}_{ij} < 0$).

3.2. Reduced cost $\bar{c}_{ij}$

The process in the previous subsection gives us an idea to improve a given feasible tree solution. The question that remains unsolved is how to calculate the reduced cost $\bar{c}_{ij}$ corresponding to an arc $(i,j) \notin T$.

$$\bar{c}_{ij} := \sum_{(k,l)\in F} c_{kl} - \sum_{(k,l)\in B} c_{kl}. \tag{8}$$

[**Q:** What's this? for all $(i,j) \notin T$?]

One way of computing this reduced cost is to use the definition of $\bar{c}_{ij}$ given by (4) [new (8)]. However, the following suggests another way which will be much simpler.

• Let $y = (y_1, y_2, \cdots, y_n)$ be a dual vector (see the dual problem of LP problem). The reduced cost $\bar{c}_{ij}$ for each $(i, j) \notin T$ can be calculated by the formula:

$$\bar{c}_{ij} = c_{ij} + y_i - y_j \tag{9}$$

while $y = (y_1, y_2, \cdots, y_n)$ can be found by

• Let $y = (y_1, y_2, \cdots, y_n)$ be a dual vector (see the dual problem of LP problem). The reduced cost $\bar{c}_{ij}$ for each $(i, j) \notin T$ can be calculated by the formula:

$$\bar{c}_{ij} = c_{ij} + y_i - y_j \tag{9}$$

while $y = (y_1, y_2, \cdots, y_n)$ can be found by

$$c_{ij} = y_j - y_i, \ \forall (i, j) \in T \tag{10}$$

and $y_n = 0$.

• We can compute y_i from node n (called the root of the tree T) to the leaves of T. Then $\bar{c}_{ij}$ follows from (9).

Remark. (i) y_i can be thought of as the unit price at the node i. If the company ship a kind of commodity along the arc (i,j), then it is fair to expect the unit price at the node j is:

$$y_j = y_i + c_{ij}.$$

This explains the economical meaning of (10).

Remark. (i) y_i can be thought of as the unit price at the node i.
If the company ship a kind of commodity along the arc (i, j), then it is fair to expect the unit price at the node j is:

$$y_j = y_i + c_{ij}.$$

This explains the economical meaning of (10).
(ii) If for some $(i, j) \notin T$,

$$y_i + c_{ij} < y_j,$$

which means that $\bar{c}_{ij} = c_{ij} + y_i - y_j < 0$, the total expense $y_i + c_{ij}$ compares favorably with the selling price at node j (means that the company will get more profit when shipping commodity along the arc (i, j)). This explains why we should choose (i, j) with $\bar{c}_{ij} < 0$ as an entering variable.

3.3. Network simplex method. Consider the network flow problem (NFP)

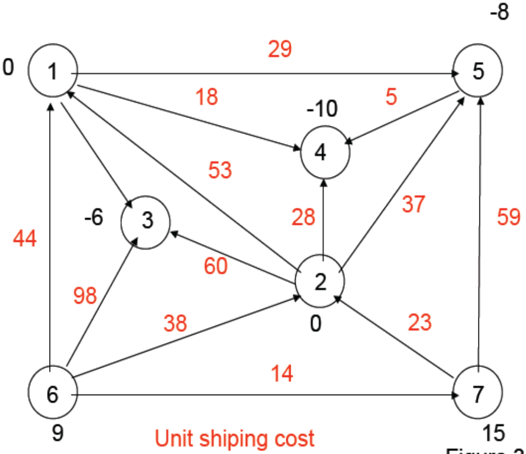


Figure 3

Let T be a feasible tree solution associated with the feasible flow $\mathbf{x}$.

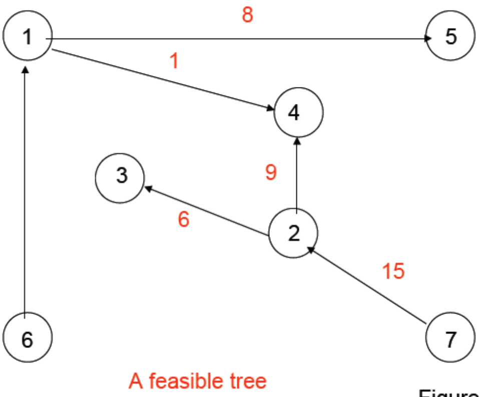


Figure 3

Algorithm

Step 1. Compute the dual vector $y = \big(y_1, y_2, ..., y_n\big)$ from (3.4). Go to Step 2.

Step 2. Compute reduced costs $\bar{c}_{ij} = c_{ij} + p_i - p_j$ for all $(i,j) \notin T$.

• If for all $(i,j) \notin T$, $\bar{c}_{ij} \geq 0$ then the feasible tree solution is optimal. STOP.

• Otherwise, Choose an arc $(i,j) \notin T$ with $\bar{c}_{ij} < 0$. Choose x_{ij} as entering variable. Go to Step 3.

Step 3. Consider the cycle $C(T, ij)$.

• If all arcs of $C(T, ij)$ have the same direction as (i, j), the minimal cost is $-\infty$. STOP.

• Otherwise, let

$$\theta^* := \min\{x_{kl} \ : \ (k, l) \in B\}$$

and choose (u, v) with $x_{uv} = \theta^*$ to be the leaving variable. Go to Step 4.

Step 4. Update the feasible tree solution.

• The new flows will be determined by (see $(*)$): $\bar{x} = (\bar{x}_{kl})_{(k,l)\in\mathcal{A}}$ with

$$\bar{x}_{kl} = \begin{cases} x_{kl} + \theta^* & \text{if } (k,l) \in F, \\ x_{kl} - \theta^* & \text{if } (k,l) \in B, \\ x_{kl} & \text{otherwise.} \end{cases}$$

Step 4. Update the feasible tree solution.

• The new flows will be determined by (see $(*)$): $\bar{x} = (\bar{x}_{kl})_{(k,l)\in\mathcal{A}}$ with

$$\bar{x}_{kl} = \begin{cases} x_{kl} + \theta^* & \text{if } (k,l) \in F, \\ x_{kl} - \theta^* & \text{if } (k,l) \in B, \\ x_{kl} & \text{otherwise.} \end{cases}$$

• The new feasible tree solution:

$$\bar{T} = (T \setminus \{(u,v)\}) \bigcup \{(i,j)\}.$$

Go to Step 1.

Example 1. Consider the network flow problem described in Figure 3 with the feasible tree solution given in Figure 3b.

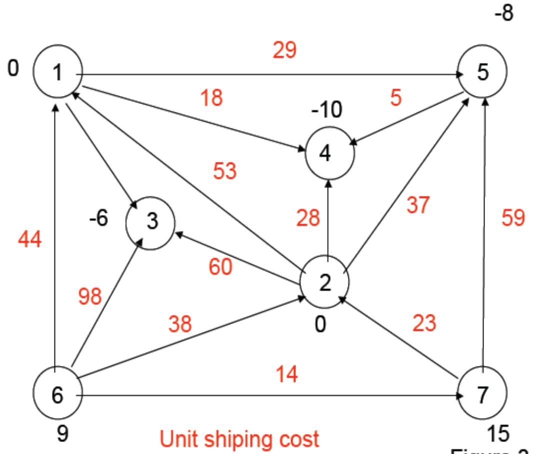
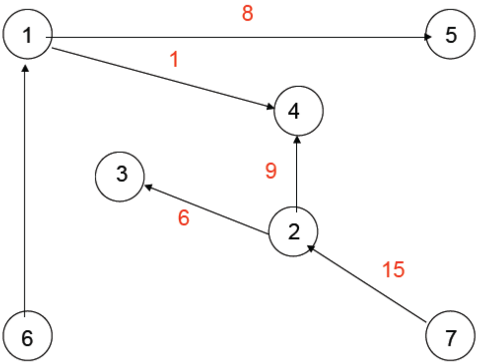


Figure 3

3.3. Network simplex method



A feasible tree

Figure 3

Iteration 1

Step 1. Compute $y_1, y_2, \ldots, y_n$ with

$$c_{ij} = y_j - y_i, \quad \forall (i,j) \in T \quad \text{and} \quad y_n = 0.$$

$$\begin{array}{llll}
y_2 - y_7 & = 23, & \text{node} & (7,2) \\
-y_2 + y_3 & = 60, & \text{node} & (2,3) \\
-y_2 + y_4 & = 28, & \text{node} & (2,4) \\
-y_1 + y_4 & = 18, & \text{node} & (1,4) \\
-y_1 + y_5 & = 29, & \text{node} & (1,5) \\
+y_1 - y_6 & = 44. & \text{node} & (6,1)
\end{array}$$

Iteration 1

Step 1. Compute $y_1, y_2, \ldots, y_n$ with

$$c_{ij} = y_j - y_i, \quad \forall (i,j) \in T \quad \text{and} \quad y_n = 0.$$

$$\begin{array}{lll} y_2 \quad - y_7 & = 23, & \text{node} \quad (7,2) \\ -y_2 + y_3 & = 60, & \text{node} \quad (2,3) \\ -y_2 + y_4 & = 28, & \text{node} \quad (2,4) \\ -y_1 + y_4 & = 18, & \text{node} \quad (1,4) \\ -y_1 + y_5 & = 29, & \text{node} \quad (1,5) \\ +y_1 - y_6 & = 44. & \text{node} \quad (6,1) \end{array}$$

This system gives:

$$y_2 = 23, \;\; y_3 = 83, \;\; y_4 = 51, \;\; y_1 = 33,$$

$$y_5 = 62, \;\; y_6 = -11, y_7 = 0.$$

Step 2. Compute $\bar{c}_{ij} = c_{ij} + y_i - y_j$ for all $(i,j) \notin T$.

$$\begin{aligned}
\bar{c}_{13} &= c_{13} + y_1 - y_3 &&= 53 + 33 - 83 &&= 3 > 0,\\
\bar{c}_{21} &= c_{21} + y_2 - y_1 &&= 8 + 23 - 33 &&= -2 < 0,\\
\bar{c}_{63} &= c_{63} + y_6 - y_3 &&= &&= 4 > 0,\\
\bar{c}_{54} &= c_{54} + y_5 - y_4 &&= &&= 16 > 0,\\
\bar{c}_{25} &= c_{25} + y_2 - y_5 &&= &&= -2 < 0,\\
\bar{c}_{62} &= c_{62} + y_6 - y_2 &&= &&= 5 > 0,\\
\bar{c}_{75} &= &&= &&= -3 < 0,\\
\bar{c}_{67} &= &&= &&= 3 > 0.
\end{aligned}$$

Step 2. Compute $\bar{c}_{ij} = c_{ij} + y_i - y_j$ for all $(i,j) \notin T$.

$$
\begin{aligned}
\bar{c}_{13} &= c_{13} + y_1 - y_3 &&= 53 + 33 - 83 &&= 3 > 0,\\
\bar{c}_{21} &= c_{21} + y_2 - y_1 &&= 8 + 23 - 33 &&= -2 < 0,\\
\bar{c}_{63} &= c_{63} + y_6 - y_3 &&= &&= 4 > 0,\\
\bar{c}_{54} &= c_{54} + y_5 - y_4 &&= &&= 16 > 0,\\
\bar{c}_{25} &= c_{25} + y_2 - y_5 &&= &&= -2 < 0,\\
\bar{c}_{62} &= c_{62} + y_6 - y_2 &&= &&= 5 > 0,\\
\bar{c}_{75} &= &&= &&= -3 < 0,\\
\bar{c}_{67} &= &&= &&= 3 > 0.
\end{aligned}
$$

Note that $\bar{c}_{21} = \bar{c}_{25} = -2 < 0$ and $\bar{c}_{75} = -3 < 0$.

Step 2. Compute $\bar{c}_{ij} = c_{ij} + y_i - y_j$ for all $(i,j) \notin T$.

$$
\begin{aligned}
\bar{c}_{13} &= c_{13} + y_1 - y_3 &&= 53 + 33 - 83 &&= 3 > 0,\\
\bar{c}_{21} &= c_{21} + y_2 - y_1 &&= 8 + 23 - 33 &&= -2 < 0,\\
\bar{c}_{63} &= c_{63} + y_6 - y_3 &&= &&= 4 > 0,\\
\bar{c}_{54} &= c_{54} + y_5 - y_4 &&= &&= 16 > 0,\\
\bar{c}_{25} &= c_{25} + y_2 - y_5 &&= &&= -2 < 0,\\
\bar{c}_{62} &= c_{62} + y_6 - y_2 &&= &&= 5 > 0,\\
\bar{c}_{75} &= &&= &&= -3 < 0,\\
\bar{c}_{67} &= &&= &&= 3 > 0.
\end{aligned}
$$

Note that $\bar{c}_{21} = \bar{c}_{25} = -2 < 0$ and $\bar{c}_{75} = -3 < 0$.

Choose $(7,5)$ as entering arc.

Step 3. Determine the cycle $C(T, 75)$.

$$C(T, 75) = \{(7,5), (1,5), (1,4), (2,4), (7,2)\},$$

oriented following the direction of the arc (7, 5) as in Figure 3b.

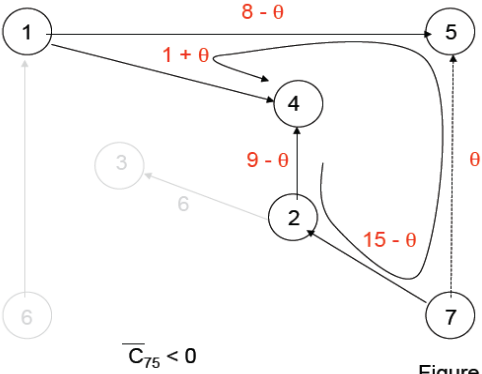


Figure 3b

$$\Theta^* := \min\{x_{15}, x_{24}, x_{72}\} = x_{15} = 8.$$

Leaving arc: $(1,5)$ (or, leaving variable x_{15}).

Step 4. Update the feasible tree solution T. The new feasible tree solution reads as (see Figure 5).

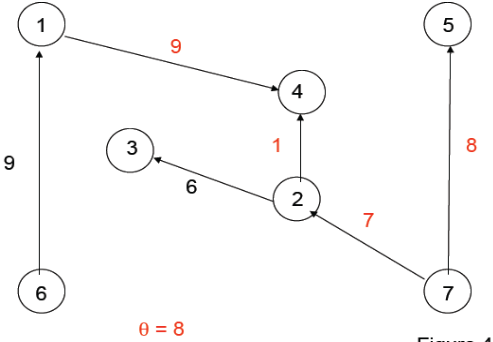


Figure 4/5

Iteration 2

Step 1. Compute $y_1, y_2, ..., y_n$:

$$y_7 = 0, y_5 = 59, y_2 = 23, y_3 = 83,$$

$$y_4 = 51, y_1 = 33, y_6 = -11.$$

Step 2. Compute $\bar{c}_{ij}$ with $(i, j) \notin T$:

$$\bar{c}_{21} = c_{21} + y_2 - y_1 = 8 + 23 - 33 = -2 < 0.$$

Iteration 2

Step 1. Compute $y_1, y_2, ..., y_n$:

$$y_7 = 0, y_5 = 59, y_2 = 23, y_3 = 83,$$

$$y_4 = 51, y_1 = 33, y_6 = -11.$$

Step 2. Compute $\bar{c}_{ij}$ with $(i, j) \notin T$:

$$\bar{c}_{21} = c_{21} + y_2 - y_1 = 8 + 23 - 33 = -2 < 0.$$

Entering arc: $(2, 1)$.

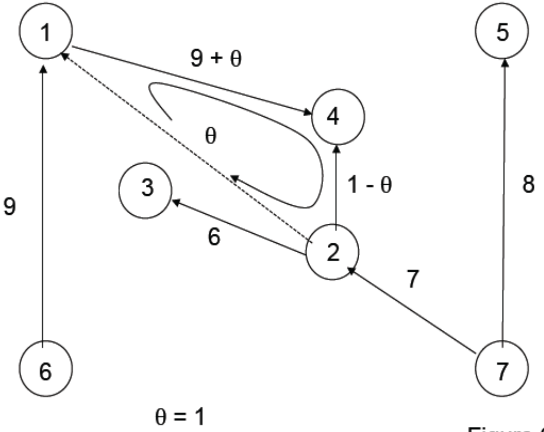


Figure 6

Step 3. Determine the cycle $C(T, 21)$.

$$C(T, 21) = \{(2,1), (1,4), (2,4)\}$$

with the direction as shown in Figure 6

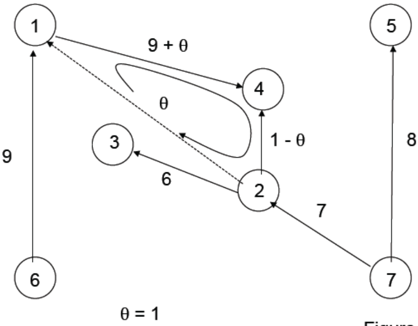


Figure 6

$\theta^* = x_{24} = 1$. Leaving arc: $(2,4)$.

Step 4. Update the feasible tree solution T. The new feasible tree solution reads as (see Figure 7)

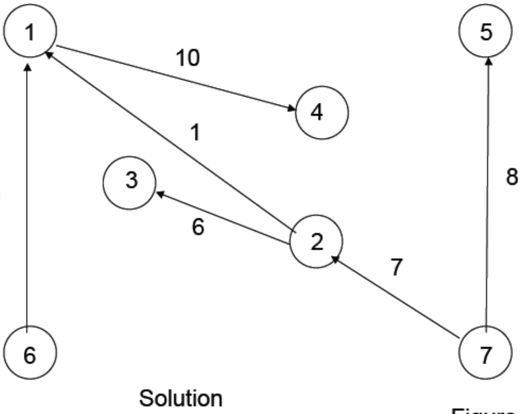


Figure 7

Iteration 3

Step 1. Compute $y_1, y_2, ..., y_n$.

$$y_7 = 0, y_5 = 59, y_2 = 23, y_3 = 83, y_4 = 49,$$

$$y_1 = 31, y_6 = -13.$$

Iteration 3

Step 1. Compute $y_1, y_2, ..., y_n$.

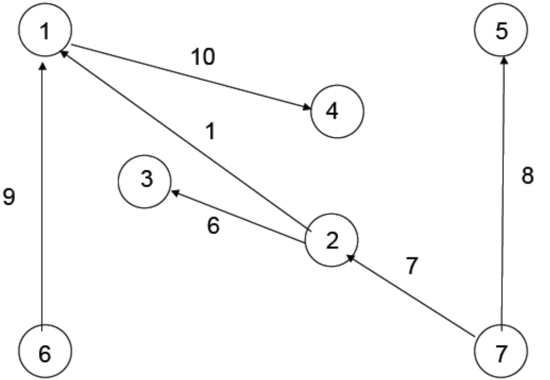
$$y_7 = 0, y_5 = 59, y_2 = 23, y_3 = 83, y_4 = 49,$$

$$y_1 = 31, y_6 = -13.$$

Step 2. Now $\bar{c}_{ij} = c_{ij} + y_i - y_j \geq 0$ for all $(i, j) \notin T$.

The current feasible tree solution is optimal.

Solution tree



Solution

Figure 7

The optimal flows are:

$$x_{14} = 10, x_{21} = 1, x_{23} = 6,$$

$$x_{61} = 9, x_{75} = 8, x_{72} = 7$$

and for other (i, j), $x_{ij} = 0$.

The optimal cost of the problem is:

$$\begin{aligned} Z^* &= 10 \times 18 + 1 \times 8 + 6 \times 60 + 9 \times 44 \\ &\quad +8 \times 59 + 7 \times 23 \\ &= 1577(\text{units}). \end{aligned}$$

5.1. Maximum flow problems (MFP)

Given a network $G = (\mathcal{N}, \mathcal{A})$. Two distinguished nodes are considered:

- Source: 1 or S
- Sink: n or T

(other nodes are called intermediate nodes).

- No demand, i.e., $b_i = 0$ for all i
- No cost, i.e., $c_{ij} = 0$ for all (i, j)
- Assume that NO arc enters S and NO arc leaves the sink T.

5. Maximum flow problems

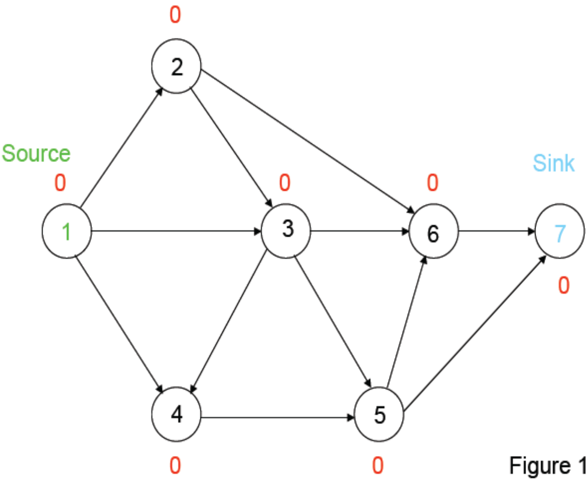


Figure 1

- Flow: A flow $\mathbf{x}$ through the network is an assignment of numbers x_{ij} to various arcs (i,j) such that
$x = (x_{ij})_{(i,j)\in\mathcal{A}}$ satisfying at each node intermediate node j (flow conservation law):

$$\sum_i x_{ij} = \sum_k x_{jk}. \tag{11}$$

- For each flow $\mathbf{x} = (x_{ij})_{(i,j)\in\mathcal{A}}$, its Volume is defined by:

$$V := \sum_j x_{Sj} = \sum_k x_{kT} \ . \tag{12}$$

- A flow $\mathbf{x}$ is feasible if

$$0 \leq x_{ij} \leq u_{ij}, \quad \text{for each} \quad (i,j) \in \mathcal{A},$$

where u_{ij} is the maximum flow capacity (upper bound of flows) through the arc (i,j).

Statement of the maximum flow problems (MFP)

The maximum flow problem is the problem of finding a flow of largest possible volume through a prescribed network from the source to the sink (see Figure 1)

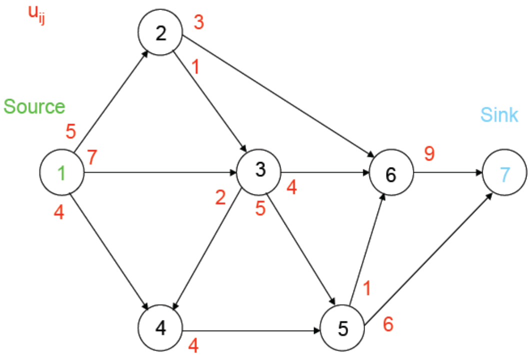


Figure 1

Statement of the maximum flow problems (MFP)

The maximum flow problem is the problem of finding a flow of largest possible volume through a prescribed network from the source to the sink (see Figure 1).

Mathematical model:

$$
\begin{array}{lll}
\text{(P)} & \text{Maximize} & b_S \\
 & \text{subject to} & Ax = b, \\
 & & b_T = -b_S, \\
 & & b_i = 0, \ \forall i \neq s, t, \\
 & & 0 \leq x_{ij} \leq u_{ij}, \ \forall (i,j) \in \mathcal{A}.
\end{array}
$$

This is a Linear Programming problem.
Question: The equation $Ax = b$ indicates what condition in this (NFP)?

5.2. Residual networks and augmenting paths

Residual network

After some flows have been assigned to the arcs, the residual network shows the remaining arc capacities (called residual capacities) for assigning additional flows (See Figure 2)

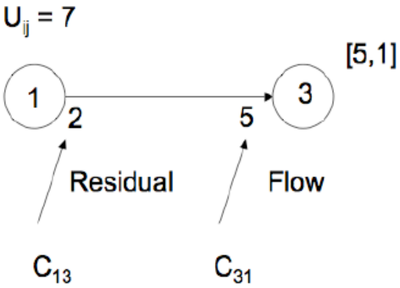


Figure 2

Augmenting paths

Augmenting path is a path from the source to the sink in the residual network such that every arc on this path has strictly positive residual capacity (see Figure 3).

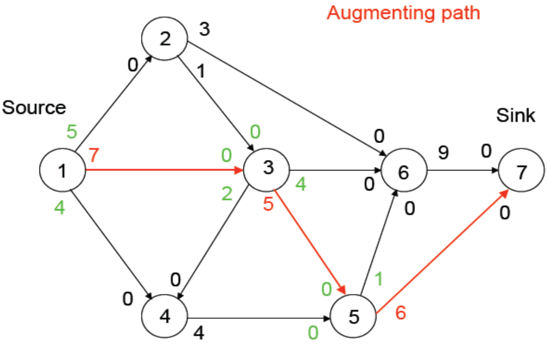


Figure 3

- The minimum of these residual capacities of arcs on the augmenting path is called the residual capacity of the augmenting path.

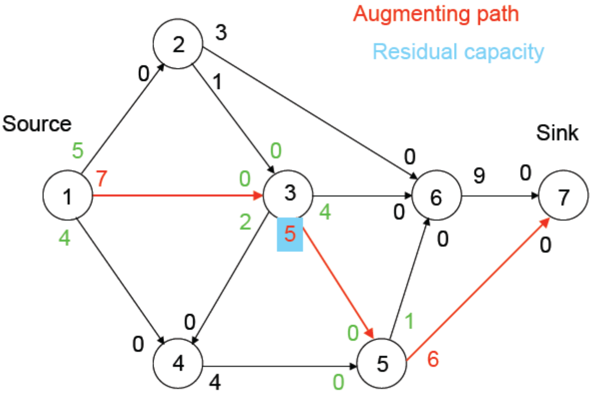


Figure 4

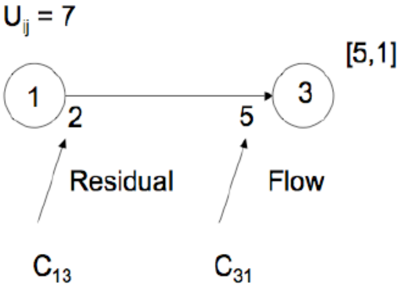
5. Maximum flow problems

- Residual capacity of an augmenting path represents the amount of flow that can feasibly be added to the entire path.

- Thus, each augmenting path provides an opportunity to further augment the flow through the network.

5.3. Algorithm for Maximum flow problems

- Initial capacity of (i, j): $(\bar{C}_{ij}, \bar{C}_{ji})$
- For a node j that receives a flow from node i, we define a label $[a_i, i]$, where a_i is the flow from node i to node i.



Algorithm

Set $(c_{ij}, c_{ji}) = (\bar{C}_{ij}, \bar{C}_{ji})$ for all $(i, j) \in \mathcal{A}$. Suppose that $\mathcal{N} = \{1, 2, ..., n\}$ and Source: 1; Sink: n.

Step 1. Set $a_1 = +\infty$ and label node 1 with $[a_1, -]$. Set $i := 1$. Go to Step 2.

Algorithm

Set $(c_{ij}, c_{ji}) = (\bar{C}_{ij}, \bar{C}_{ji})$ for all $(i, j) \in \mathcal{A}$. Suppose that $\mathcal{N} = \{1, 2, ..., n\}$ and Source: 1; Sink: n.

Step 1. Set $a_1 = +\infty$ and label node 1 with $[a_1, -]$. Set $i := 1$. Go to Step 2.
Step 2. Determine S_i as the set of nodes that can be reached directly from i with an arc of positive residual.

If $S_i \neq \emptyset$, go to Step 3,

Otherwise, go to Step 4.

Step 3. Determine $k \in S_i$ such that:

$$c_{ik} = \max_{j \in S_i} \{c_{ij}\}.$$

Set $a_k = c_{ik}$ and label node k with $[a_i, i]$.

If $k = n$, the sink n has been reached and an augmenting path is found. Go to Step 5,

Otherwise, set $i = k$ and go to Step 2.

Step 3. Determine $k \in S_i$ such that:

$$c_{ik} = \max_{j \in S_i}\{c_{ij}\}.$$

Set $a_k = c_{ik}$ and label node k with $[a_i, i]$.

If $k = n$, the sink n has been reached and an augmenting path is found. Go to Step 5,

Otherwise, set $i = k$ and go to Step 2.

Step 4. (backtracking)

If $i = 1$, no further augmenting paths are possible. Go to Step 6.

Otherwise, let r be the node that has been labeled immediately before the current node i and remove i from the set of nodes that are adjacent to r (i.e., the set S_r). Set $i = r$ and go to Step 2.

Step 5. (update the residual network)

Suppose that we have just constructed the p^{th} augmenting path from the source 1 to the sink n. Let $N_p = (1, k_1, k_2, ..., n)$ be the nodes of this path.

The maximum flow (feasibly ship) along the path is computed as:

$$f_p = \min\{a_1, a_{k_1}, a_{k_2}, ..., a_n\}$$

The new residual capacity of each arc (i, j) along the augmenting path is defined as:

$$(c_{ij}, c_{ji})[\text{new}] = (c_{ij} - f_p, c_{ji} + f_p)$$

(see Figure 5).

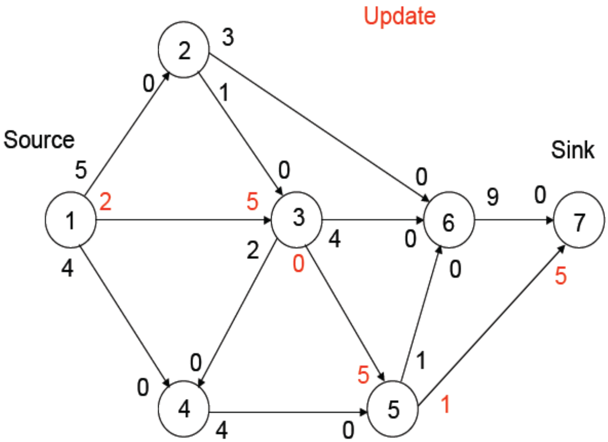


Figure 5

Note: (c_{ij}, c_{ji}) remains the same if the arc (i, j) does not involve in this augmenting path.

Update the residual network. Set $i = 1$ and return to Step 2 to attempt a new augmenting path.

Step 6. (Solution)

Suppose that m augmenting paths have been determined (and no more possible).

- The maximum flow through the network is:

$$F = f_1 + f_2 + ... + f_m.$$

• Determine the optimal flow in each arc (i, j):
If (c_{ij}, c_{ji}) is the final residual of the arc (i, j), the optimal flow in (i, j) is computed as follows:

Let

$$\begin{aligned}(\alpha, \beta) &= (\bar{C}_{ij}, \bar{C}_{ji}) - (c_{ij}, c_{ji}) \\ &= (\bar{C}_{ij} - c_{ij}, \bar{C}_{ji} - c_{ji}).\end{aligned}$$

If $\alpha > 0$ then the optimal flow from i to j is α.

If $\beta > 0$ then the optimal flow from j to i is β.

[Note: It is impossible to have both α and β positive ! prove this.]

Example.

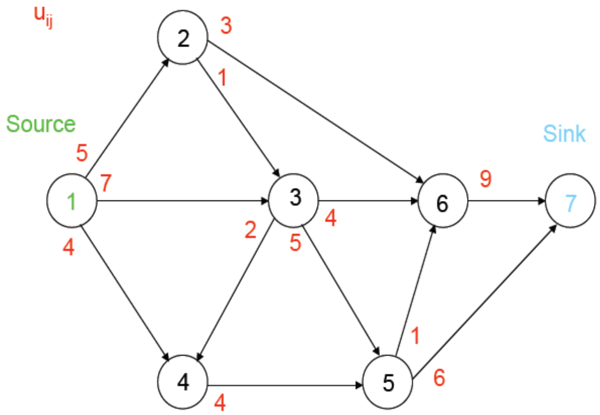


Figure 1

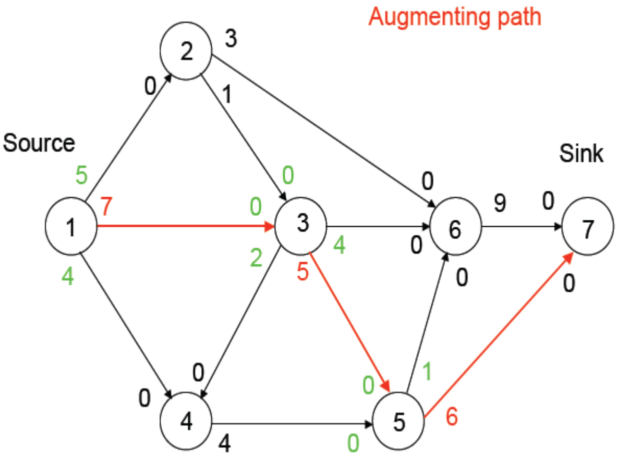

Augmenting path
Source
Sink
1
2
3
4
5
6
7
3
0
1
5
7
0
0
0
4
0
2
5
4
0
9
0
0
0
0
0
1
0
4
0
6

Figure 3

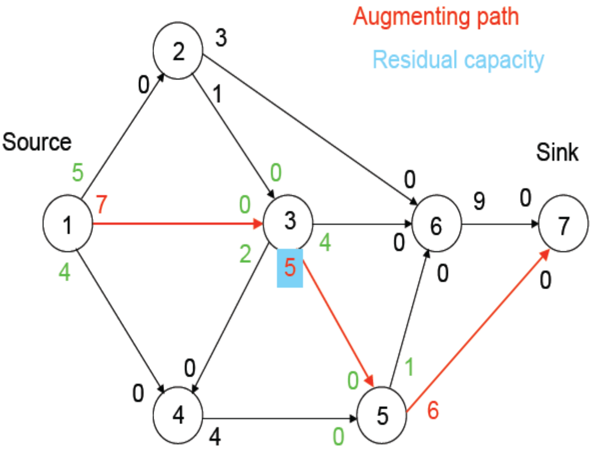

Augmenting path
Residual capacity
Source
Sink
1
2
3
4
5
6
7
3
0
1
0
5
7
0
0
0
9
0
4
2
5
4
0
0
0
0
0
1
0
4
0
6

Figure 4

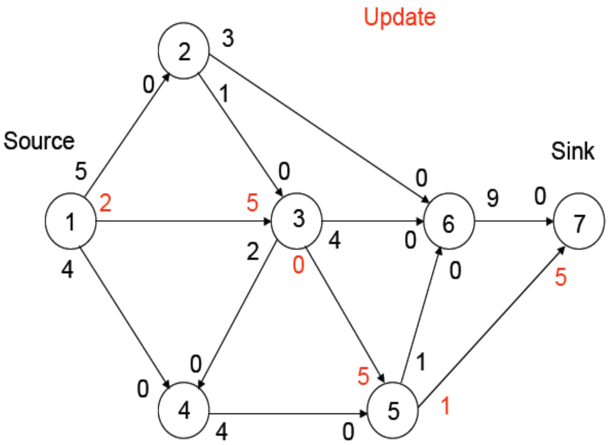


Figure 5

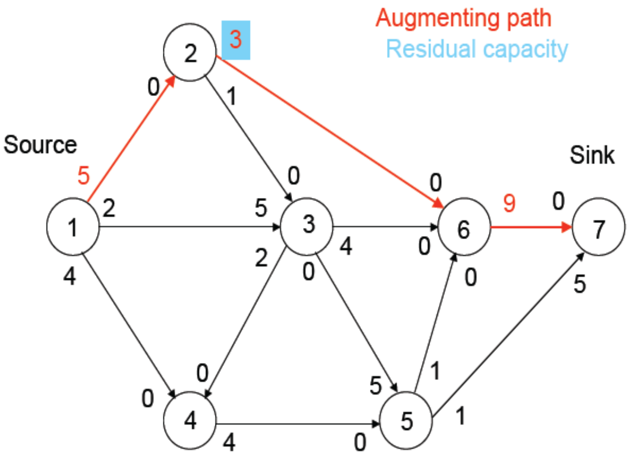

Augmenting path
Residual capacity
Source
Sink
1
2
3
4
5
6
7
3
0
1
5
2
5
4
0
0
2
0
4
0
9
0
0
5
0
0
5
1
4
0
1

Figure 6

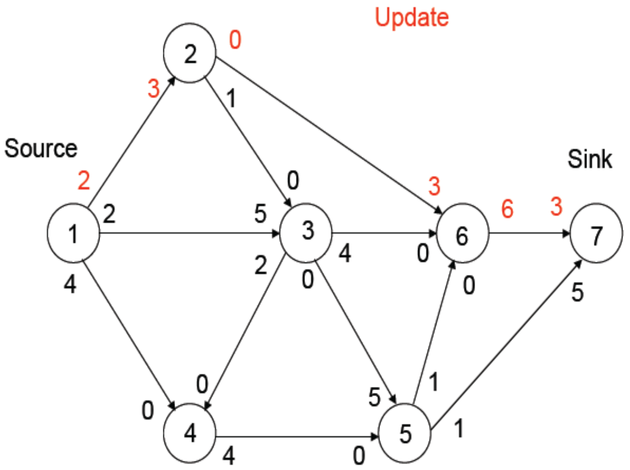


Figure 7

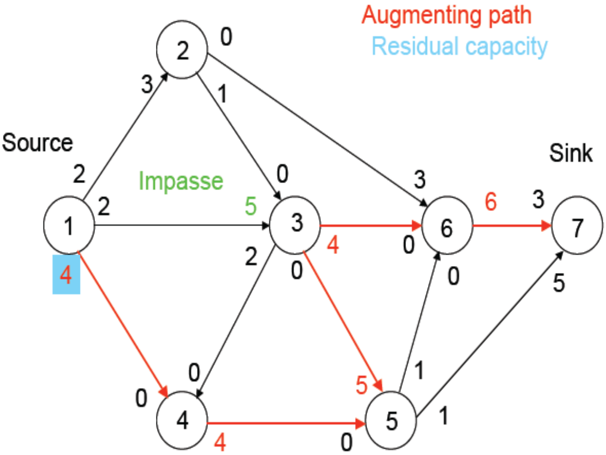

Augmenting path
Residual capacity
Source
Sink
Impasse
1
2
3
4
5
6
7
3
2
2
4
0
1
0
5
0
2
0
3
6
3
4
0
0
5
0
5
4
0
1
1

Figure 8

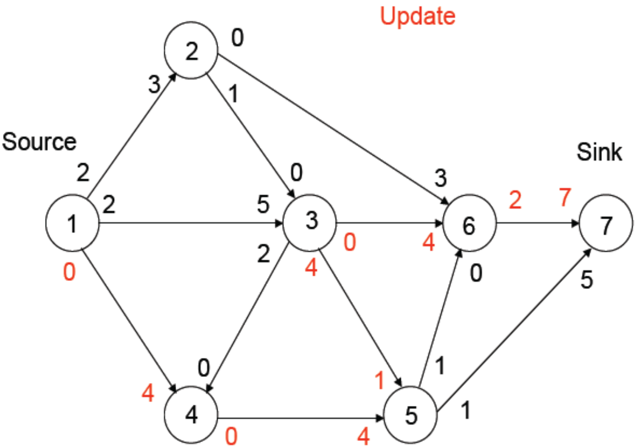


Figure 9

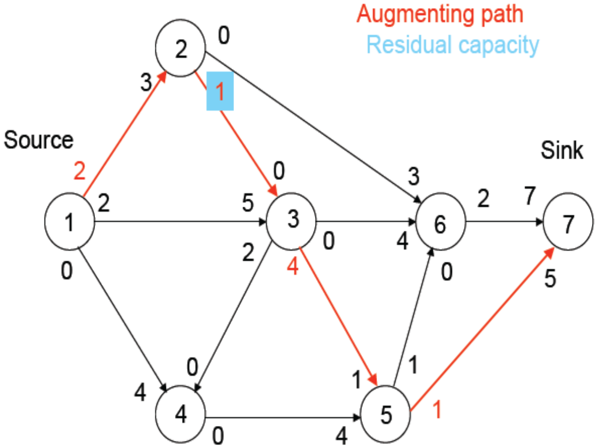

Augmenting path
Residual capacity
Source
Sink
1
2
3
4
5
6
7
2
2
0
3
1
0
0
5
2
0
4
3
4
2
7
0
5
0
4
1
1
0
4
1

Figure 10

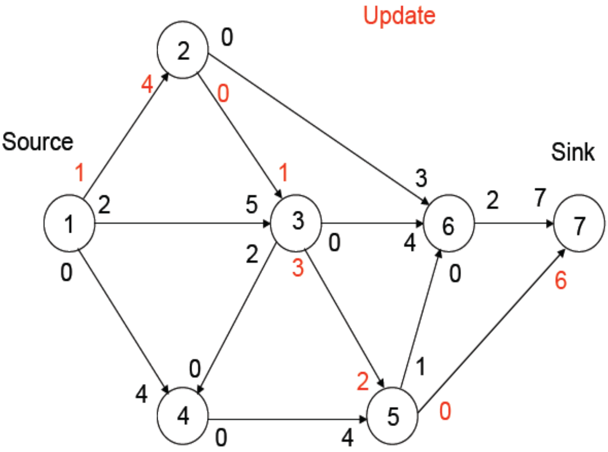


Figure 11

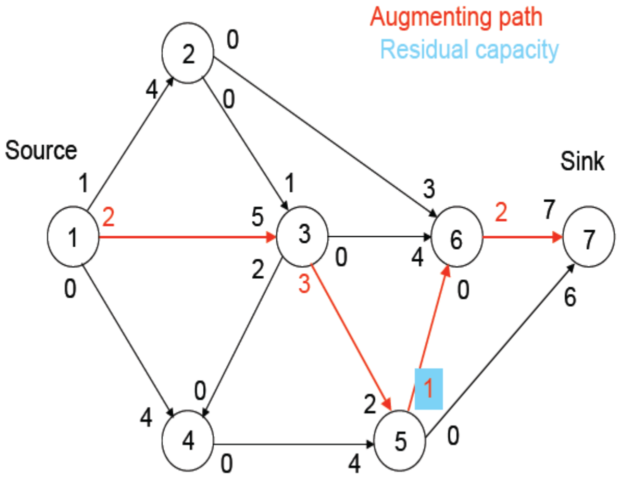

Augmenting path
Residual capacity
Source
Sink
1
2
3
4
5
6
7
0
4
0
1
2
5
1
3
2
7
0
4
2
3
0
6
0
4
0
2
1
4
0
0

Figure 13

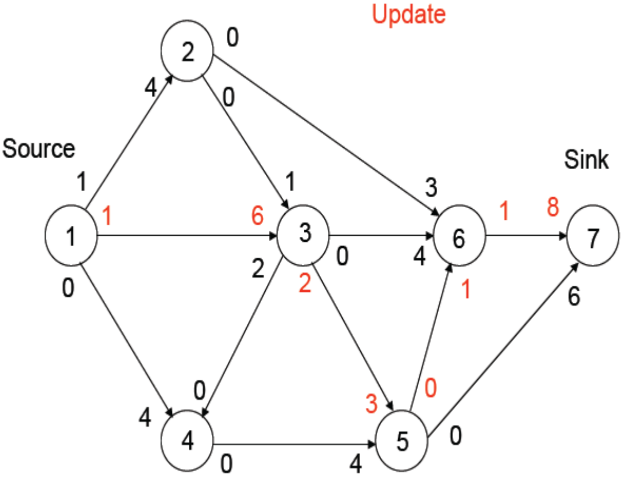


Figure 13