Generalized Network Flow Programming

This chapter adapts the bounded variable primal simplex method to the generalized minimum cost flow problem. Generalized networks are far more useful than pure networks because the arc gain parameter allows significantly richer models. The implementation of the simplex method for this problem is similar to the simplex method for the pure problem; however, the generalized algorithm is more complex in several ways. An important difference is that the solutions obtained no longer have the integrality property. We begin by describing the basis and basic solutions of the generalized problem. For the pure problem the basis always defines a subnetwork that is a tree. For the generalized problem, the basis again describes a subnetwork, but now the subnetwork may have several components. One component will be a tree rooted at the slack node, while each of the others includes a single cycle.

C.1 Problem Statement

The optimization problem is to determine the minimum cost flow in a network with given sets of arcs and nodes with the parameters of upper bound, cost and gain defined for each arc and fixed external flow for each node.

The Example Problem

As an example for this discussion we will use the network of Fig. 1 with the number of nodes, *m*, equal to 5 and the number of arcs, *n*, equal to 5. Node *m* is called the slack node and can supply or absorb any amount of flow. Conservation of flow is required at the other nodes. Arcs representing slack external flows and artificial arcs originate or terminate at the slack node. In the example, no arcs are incident to the slack node. The generalized model differs from the pure network model because the gain parameters on the arcs may have values other than 1.

The matrices defining the linear programming model for the example are

Flow Variables: $\mathbf{x} = [\ x_1, x_2, x_3, x_4, x_5\ ]^T$

Upper Bound: $\mathbf{u} = [\ 3, 4, 3/2, 1, 1\ ]^T$

Cost: $\mathbf{c} = [\ 2, 20\ , 1, 12, 2\ ]$

Gain: $\mathbf{g} = [1/3, 1/2, /2, 1/4, 1/4\ ]$

$$\begin{array}{cc} & \begin{array}{ccccc} \mathbf{a}_1 & \mathbf{a}_2 & \mathbf{a}_3 & \mathbf{a}_4 & \mathbf{a}_5 \end{array} \\ \mathbf{A} = & \begin{bmatrix} 1 & 1 & 0 & 0 & 0 \\ -1/3 & 0 & 1 & 1 & 0 \\ 0 & -1/2 & -1/2 & 0 & 1 \\ 0 & 0 & 0 & -1/4 & -1/4 \end{bmatrix} \end{array}, \ \mathbf{b} = \begin{bmatrix} 4 \\ 0 \\ 0 \\ -3/8 \end{bmatrix}$$

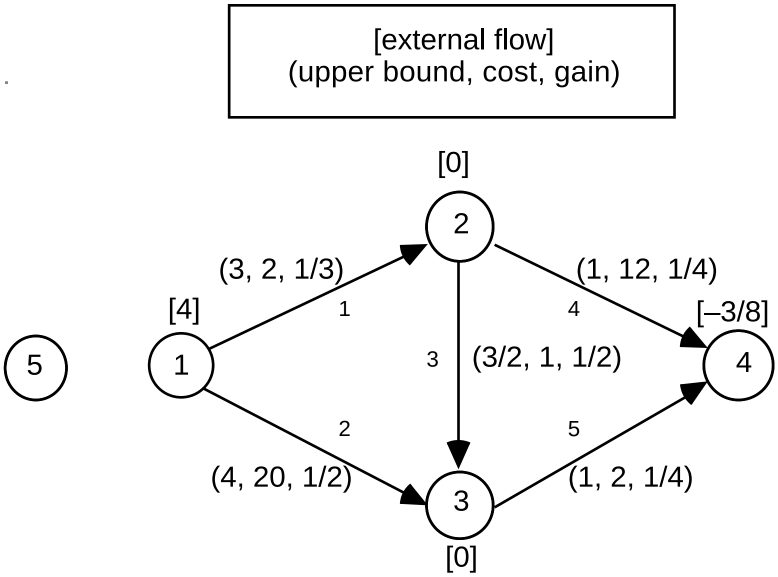


Figure 1. Example of generalized network

Each column of the **A** matrix represents an arc and has at most two nonzero entries. For the general arc k, the quantity +1 appears at the row representing the origin node of the arc and $-g_k$ appears at the row representing the terminal node of the arc. Columns for arcs originating or terminating at the slack node have only one nonzero entry (±1). Because the constraint matrix does not have all unity coefficients, the matrix does not have the property of total unimodularity, and basic solutions to the generalized problem are not usually integer. This is illustrated in Fig. 2 by a basic feasible solution for the example problem that happens also to be optimal.

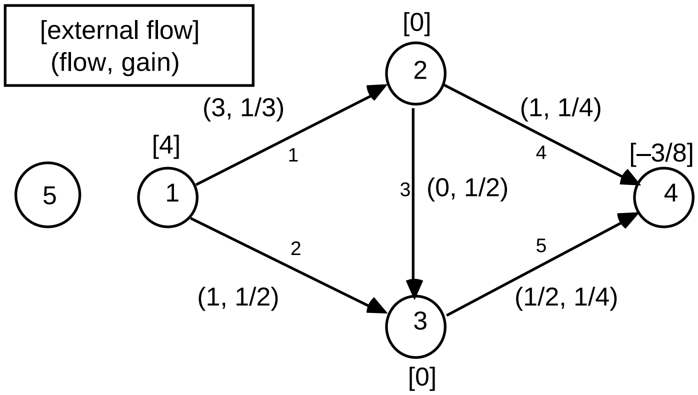


Figure 2. Optimal solution to the example problem

Conditions for Optimality

Our problem is to find a solution to the primal problem, **x**, and a solution to the dual problem π to satisfy the conditions for optimality. The solutions **x** and π are optimal if the following conditions are satisfied. For simplicity define for the arcs:

$$d_k = \pi_i - g_k\pi_j + c_k \text{ for all } k(i, j) \varepsilon \mathrm{K}.$$

The solutions $\mathbf{x}$ and π are optimal if the following conditions are satisfied.

1. *Primal feasibility*

   a. $\mathbf{x}$ provides conservation of flow at all nodes except the slack node.

   b. $0 \leq x_k \leq u_k$ for all arcs.

2. *Complementary Slackness*

   For each arc:

   a. if $0 < x_k < u_k$, then $d_k = 0$,

   b. if $x_k = 0$ then $d_k \geq 0$,

   c. if $x_k = u_k$ then $d_k \leq 0$.

When we specify the values of dual variables associated with each node as

$$\pi = (\pi_1, \pi_2, \pi_3, \pi_4, \pi_5) = (24, 78, 88, 360, 0),$$

the solution in Fig. 2 satisfies these conditions. The solution is optimal.

C.2 Basic Solutions

The generalized minimum cost network flow problem is a special case of the linear programming problem. We are assured, therefore, that there is at least one optimum basic solution.

The Basis

As for the pure problem, we identify a basic solution by specifying the set of basic arcs. An acceptable basis matrix has *m* – 1 independent columns selected from **A**. As for the pure problem, any spanning tree rooted at the slack node will result in a basis. For the generalized problem, some additional structures involving cycles also yield bases. For example consider the subnetwork consisting of arcs 1, 2, 4, and 5 shown in Fig. 3. Because this network forms a cycle, it would not be acceptable for the pure problem, but it will be for the generalized problem.

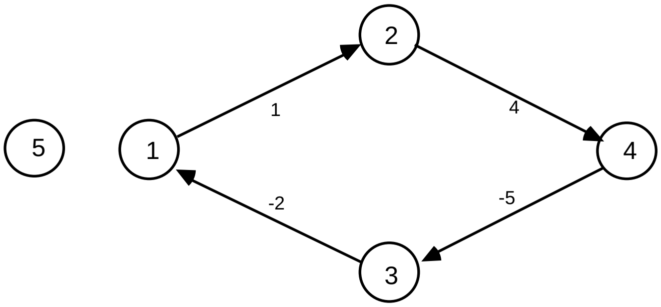


Figure 3. Basis subnetwork with a cycle

For the example of Fig. 3, with basis {1, 2, 4, 5} we have

$$\begin{array}{cc} \text{Column} & \begin{array}{cccc} \mathbf{a}_1 & \mathbf{a}_2 & \mathbf{a}_4 & \mathbf{a}_5 \end{array} \\ \mathbf{B} = & \begin{bmatrix} 1 & 1 & 0 & 0 \\ -1/3 & 0 & 1 & 0 \\ 0 & -1/2 & 0 & 1 \\ 0 & 0 & -1/4 & -1/4 \end{bmatrix} \end{array}$$

The fact that the determinant of this matrix is not zero implies that the columns are independent and that the selected arcs do form a basis.

It can be shown that the subnetwork defining a basis for the generalized network problem will consist of a tree rooted at the slack node and any number of components, each containing a single cycle. The components containing cycles are called 1-trees because they each contain one more arc than a tree. The total number of arcs in the collection of 1-trees and the single tree rooted at the slack node will always be *m* – 1. Fig. 4 shows several basis subnetworks for the example network. For the illustration we have added an artificial arc (6) from the slack node to node 1.

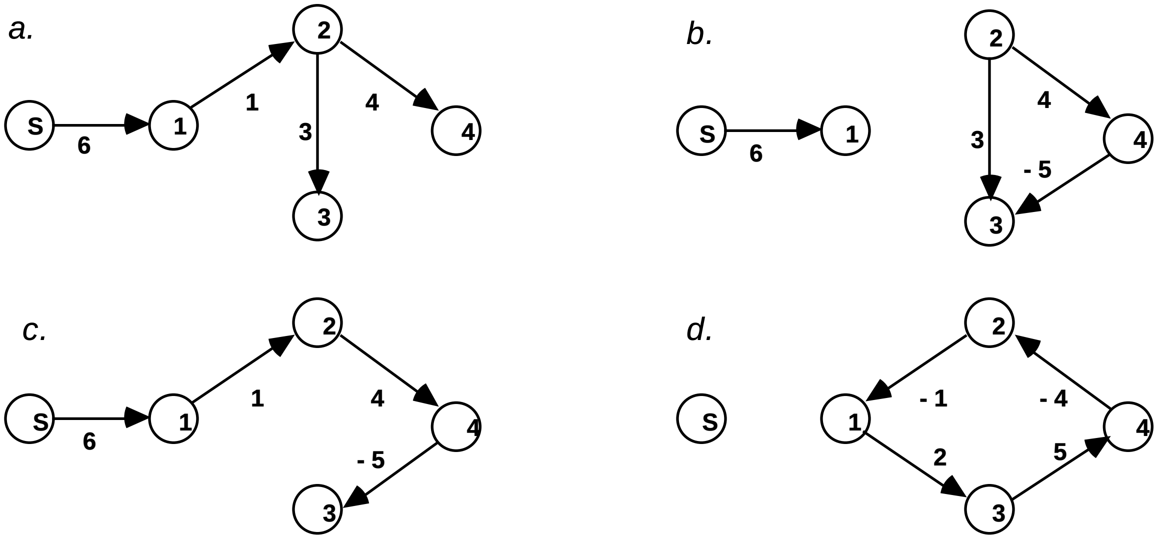


Figure 4. Subnetworks defining bases for the example problem

We show the basis as a subnetwork consisting of both forward and mirror arcs as in Fig. 3 and 4. For the tree containing the slack node, the arcs are oriented so that the tree is a directed tree rooted at the slack node. For subnetwork components containing a cycle, the cycle arcs are oriented to form a directed cycle (there are always two acceptable orientations). The arcs not on the cycle are oriented to form a directed spanning tree with the cycle representing the root node. For many purposes it is useful to consider a mirror arc (-*k*) as an arc oriented in the opposite direction to arc *k* with gain and cost respectively:

$$g_{-k} = 1/g_k,\ c_{-k} = -c_k/g_k.$$

This is equivalent to making the transformation $x_{-k} = -x_k\, g_k$ in the original model. The transformed column in **A** for arc $-k$ is computed as $\mathbf{a}_{-k} = -\mathbf{a}_k/g_k$. In the following discussion we make whatever transformation necessary to represent the basis with directed trees and cycles, realizing the original flows can be recovered by taking the inverse transformation:

$$x_k = -x_{-k}/g_k.$$

In terms of the transformed variables, the basis for the example of Fig. 3 is $\mathbf{n}_B = [-2, 1, -5, 4]$. As for other linear programming problems we use the $\mathbf{n}_B$ vector to describe the basis. The element n_{Bi} is the predecessor arc for node *i* in the basis subnetwork. Now the basis matrix becomes

$$\begin{array}{lcl}
\text{Arc:} & \begin{array}{cccc} \mathbf{a}_{-2} & \mathbf{a}_{1} & \mathbf{a}_{-5} & \mathbf{a}_{4} \end{array} & \text{Node} \\
\mathbf{B} = & \left[\begin{array}{cccc} -2 & 1 & 0 & 0 \\ 0 & -1/3 & 0 & 1 \\ 1 & 0 & -4 & 0 \\ 0 & 0 & 1 & -1/4 \end{array}\right] & \begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \end{array}
\end{array}$$

It is useful to determine descendant lists of the nodes and arcs in the basis subnetwork. This is complicated by the fact that all the nodes on a cycle are in fact both successors and predecessors of each other. We arbitrarily select some node on the cycle, identify it as the root node, and for the purpose of creating the list, neglect the predecessor arc to that node. The remaining nodes are ordered so that a node will appear before all its descendants. Choosing node 2 as the root for the example, the descendant node list is {2, 4, 3, 1}. The descendant arc list is created by listing the predecessor arcs for each node in the order in which they appear in the descendant node list. For the example, the descendant arc list is {1, 4, –5, –2}.

It is interesting to arrange the rows and columns of the basis matrix according to the descendant node and arc lists. Then we have.

$$\begin{array}{rcc} \text{Arc:} & \begin{array}{cccc}\mathbf{a}_1 & \mathbf{a}_4 & \mathbf{a}_{-5} & \mathbf{a}_{-2}\end{array} & \text{Node} \\ \mathbf{B} = & \begin{bmatrix} -1/3 & 1 & 0 & 0 \\ 0 & -1/4 & 1 & 0 \\ 0 & 0 & -4 & 1 \\ 1 & 0 & 0 & -2 \end{bmatrix} & \begin{array}{c} 2 \\ 4 \\ 3 \\ 1 \end{array} \end{array}$$

This matrix has all the arc gains on the diagonal, and all the entries below the diagonal are zero except one. This form is typical when the basis is a 1-tree.

To present the results in somewhat greater generality we consider the two types of basis components, shown in Fig. 5, that might be present in a generalized problem. Fig. 5a shows a component rooted at the slack node, and Fig. 5b shows a component rooted at a cycle. For simplicity we have indexed the nodes and arcs so that arc *i* is the predecessor arc of node *i*.

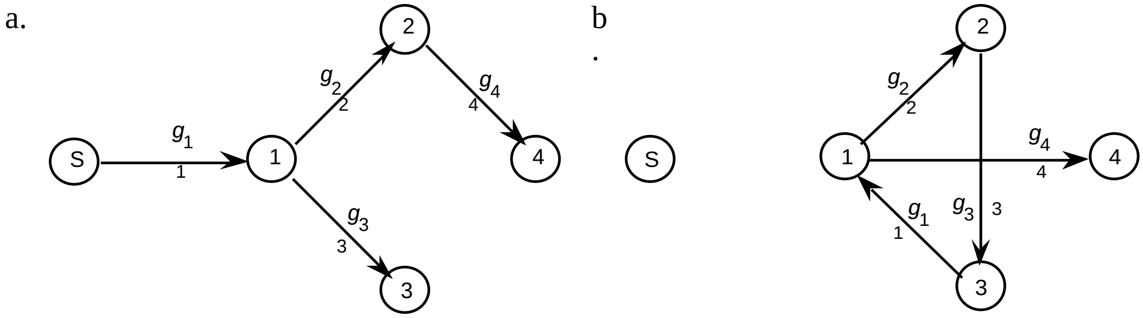


Figure 5. a. A basis with a tree rooted at the slack node

b. A basis with a component rooted at a 1-tree

We show the basis matrices for the two cases with the subscripts *a* and *b* providing identification.

$$\begin{array}{rcl} \text{Arc:} & \begin{array}{cccc} \mathbf{a}_1 & \mathbf{a}_2 & \mathbf{a}_3 & \mathbf{a}_4 \end{array} & \text{Node} \\ \mathbf{B}_\mathrm{a} = & \left[\begin{array}{cccc} -g_1 & 1 & 1 & 0 \\ 0 & -g_2 & 0 & 1 \\ 0 & 0 & -g_3 & 0 \\ 0 & 0 & 0 & -g_4 \end{array}\right] & \begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \end{array} \end{array}$$

$$\begin{array}{rcl} \text{Arc:} & \begin{array}{cccc} \mathbf{a}_1 & \mathbf{a}_2 & \mathbf{a}_3 & \mathbf{a}_4 \end{array} & \text{Node} \\ \mathbf{B}_\mathrm{b} = & \left[\begin{array}{cccc} -g_1 & 1 & 0 & 1 \\ 0 & -g_2 & 1 & 0 \\ 1 & 0 & -g_3 & 0 \\ 0 & 0 & 0 & -g_4 \end{array}\right] & \begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \end{array} \end{array}$$

The basis of type *a* is an upper diagonal matrix, with the gain factors appearing on the diagonal and all entries below the diagonal equal to zero. The determinant of this matrix is easily seen to be the product of the arc gains.

$$|\mathbf{B}_\mathrm{a}| = g_1 g_2 g_3 g_4$$

In general for a basis component describing a tree rooted at the slack node, the determinant of the basis matrix is equal in magnitude to the product of the gains of the basic arcs. Let ***B*** be the set of arcs in the basis component.

$$|\mathbf{B}| = \pm \prod_{k \in \boldsymbol{B}} g_k$$

As long as the gains are nonzero the determinant cannot be zero, so any tree defines a nonsingular basis.

To evaluate the determinant of $\mathbf{B}_\mathrm{b}$, we note that the matrix has two parts, the cycle part consisting of the intersection of rows and columns 1, 2, and 3, and the tree part consisting in this case of row and column 4. Expanding the determinant about row 4, we have

$$|\mathbf{B}_\mathrm{b}| = \left|\begin{array}{cccc} -g_1 & 1 & 0 & 1 \\ 0 & -g_2 & 1 & 0 \\ 1 & 0 & -g_3 & 0 \\ 0 & 0 & 0 & -g_4 \end{array}\right| = -g_4 \left|\begin{array}{ccc} -g_1 & 1 & 0 \\ 0 & -g_2 & 1 \\ 1 & 0 & -g_3 \end{array}\right|$$

Now expanding the determinant about the first column, we find that

$$|\mathbf{B}_\mathrm{b}| = -g_4(-g_1 g_2 g_3 + 1).$$

We identify the factor $g_1 g_2 g_3$ as the product of the gains on the cycle arcs and give it the general designation β, called the cycle gain. Thus the determinant is

$$|\mathbf{B}_b| = g_4(\beta - 1).$$

In general for a 1-tree, the determinant of the basis is equal in magnitude to the product of the gains for the arcs not on the cycle multiplied by the cycle gain less 1. Let ***T*** be the set of arcs not on the cycle. Then the determinant is

$$|\mathbf{B}| = \pm\ (\beta - 1) \prod_{k \in \boldsymbol{T}} g_k$$

where the cycle gain is

$$\beta = \prod_{k \in \boldsymbol{B}-\boldsymbol{T}} g_k$$

This determinant is nonzero if none of the arc gains are zero and the cycle gain does not equal 1. The latter condition is a requirement for nonsingular basis matrices. The cycle gain plays an important role in the simplex algorithm for generalized networks.

Basis Inverse

As for the pure problem, we do not use the basis inverse explicitly in the computational methods, rather, we compute the information we need about the basis inverse whenever it is required. To gain some understanding of the procedures, however, and to provide a link to general linear programming, we describe in this section a method for constructing the basis inverse matrix.

The basis inverse is a square matrix with a column for each node and a row for each basic variable. Let ρ_{ki} represent the entry in the basis inverse for basic arc k and node i. Note that i will range from 1 to m–1, however, k will take on the designations of arcs in the basis network. The index k will be negative for mirror arcs. It can be shown that ρ_{ki} is the negative of the flow required in arc k if one unit of flow is withdrawn at node i. In the following we determine this flow in order to determine the value of ρ_{ki}.

Let the path terminating at arc i in the basis tree be a subnetwork with t arcs, $\boldsymbol{K}_p = \{k_1, k_2, k_3, \ldots, k_t\}$, and $t + 1$ nodes, $\boldsymbol{I}_p = \{i_0, i_1, i_2, \ldots, i_t\}$. The node set and arc set are ordered so that the indices show the sequence of arcs in the path. The final node is $i_t = i$. The path may originate at the slack node as in Fig. 6, in which $i_0 = m$, or originate with a cycle as in Fig. 7. When the path originates with a cycle, some node j is listed twice in the node list. We will call this the junction node and note that $i_0 = j$.

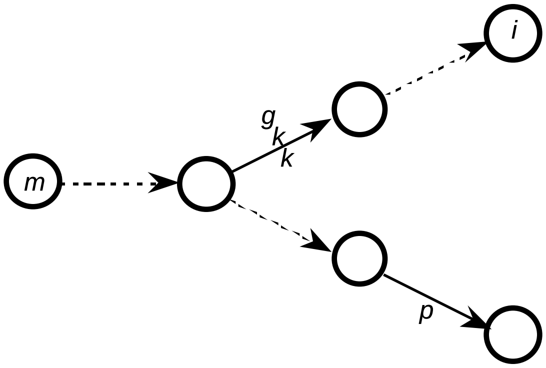


Figure 6. Tree component of basis network

The effect on the flow of basic arc *k* of drawing one unit of flow out of node *i* depends on whether the arc is on a cycle. First consider the case when arc *k* is not on a cycle. Withdrawing one unit of flow from node *i* only affects the flows on basic arcs that are on the path to node *i*, so assume *k* is on the path. Let γ_{ki} be the product of the arc gains from the origin node of arc *k* to node *i* (note that γ_{ki} includes the factor g_k). The amount of flow required on arc *k* is simply $1/\gamma_{ki}$. Then the entry in the basis inverse for the row associated with arc *k* and the column for node *i* is:

$$\rho_{ki} = \begin{cases} 0 & \text{if arc } k \text{ is not on the directed path to node } i \\ -1/\gamma_{ki} & \text{if arc } k \text{ is on the directed path to node } i \\ & \text{but not on a cycle.} \end{cases} \qquad (1)$$

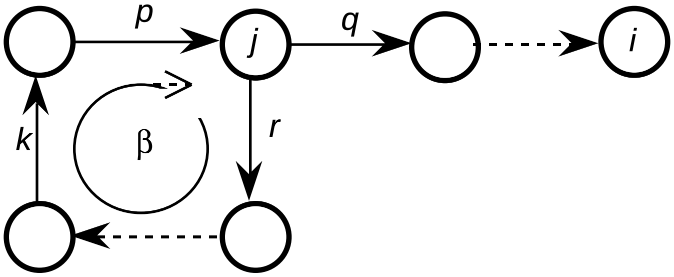


Figure 7. 1-Tree component of basis network

To determine the flow required on basic arc *k* when it is on the cycle of a 1-tree as in Fig.7, consider first the effect of the cycle in a generalized network. Say node *j* is a node on the cycle that originates arc *q*, where arc *q* is on the directed path to node *i* but not on the cycle. Let arc *p* be the cycle arc terminating at node *j* and let *r* be the cycle arc originating at node *j*. x_p and x_r are related by the cycle gain: $g_p x_p = \beta x_r$. Conservation of flow at node *j* requires that the flow entering node *j* on arc *p* must equal the flow leaving node *j* on arcs *q* and *r*.

$g_p x_p = x_q + x_r$ or

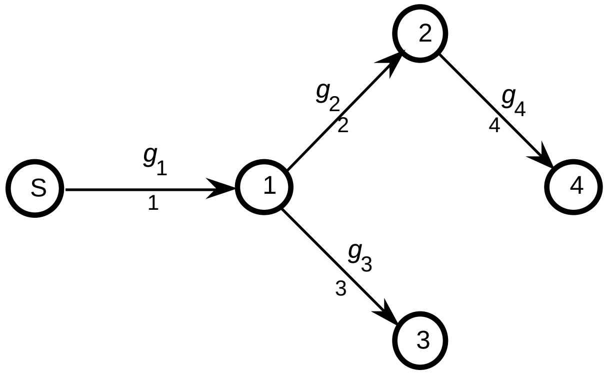
$$x_p = \left(\frac{\beta}{\beta - 1}\right)\left(\frac{x_q}{g_p}\right)$$

Note that the flow on adjacent arcs *p* and *q* are again related by the arc gain of *p*, but the fact that arc *p* is on the cycle requires that the flow x_q be multiplied by the adjustment factor β/(β – 1). To simplify the following expressions let η = β/(β – 1). The term in the basis inverse corresponding to basic arc *k* and node *i* is:

$$\rho_{ki} = \begin{cases} 0 & \text{if arc } k \text{ is not on the directed path to node } i \\ -1/\gamma_{ki} & \text{if arc } k \text{ is on the directed path to node } i \\ & \text{but not on a cycle} \\ -\eta/\gamma_{ki} & \text{if arc } k \text{ is on the directed path to node } i \\ & \text{but on a cycle.} \end{cases} \tag{2}$$

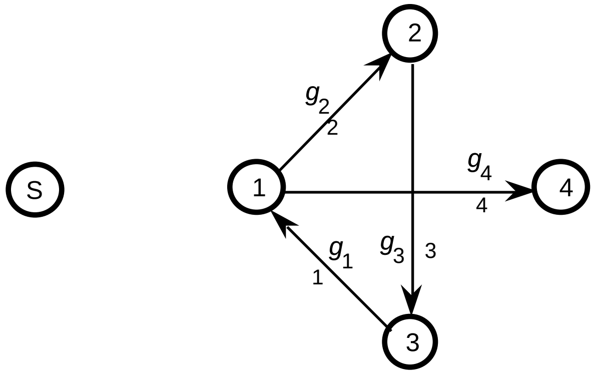
Equation (2) is general in that it also covers the tree case of Eq. (1), so we will use Eq. (2) in the following.

To illustrate the application of this expression, consider again Fig. 5. In Fig. 5a the basis has a tree with no cycles. Repeating the figure with the arc gains shown, one can easily apply Eq. 2 to obtain the basis inverse below. As the basis is an upper diagonal matrix, so is the basis inverse.



$$\begin{array}{lccccc} \text{Node:} & 1 & 2 & 3 & 4 & \text{Arc} \\ & 1/g_1 & 1/g_1g_2 & 1/g_1g_3 & 1/g_1g_2g_4 & 1 \\ & 0 & 1/g_2 & 0 & 1/g_2g_4 & 2 \\ \mathbf{B}_a^{-1} = - & 0 & 0 & 1/g_3 & 0 & 3 \\ & 0 & 0 & 0 & 1/g_4 & 4 \end{array}$$

Applying Eq. 2 to the 1-tree of Fig. 5b (repeated below) we obtain the basis inverse for this case.



Here $\beta = g_1g_2g_3$, and $\eta = \beta/(\beta - 1)$. The basis inverse determined according the rules of Eq. (2) is:

$$\begin{array}{r|cccc|l} \text{Node:} & 1 & 2 & 3 & 4 & \text{Arc} \\ \hline & \eta/g_1 & \eta/g_1g_2 & \eta/g_1g_2g_3 & \eta/g_1g_4 & 1 \\ \mathbf{B}_b^{-1} = - & \eta/g_1g_2g_3 & \eta/g_2 & \eta/_{g2g3} & \eta/g_1g_2g_3g_4 & 2 \\ & \eta/g_1g_3 & \eta/g_1g_2g_3 & \eta/g_3 & \eta/g_1g_3g_4 & 3 \\ & 0 & 0 & 0 & 1/g_4 & 4 \end{array}$$

In Fig. 8, we show again the basis 1-tree for the example of Fig. 3. The arc gains associated with the basic arcs are shown on the figure to help with the computation of the basis inverse.

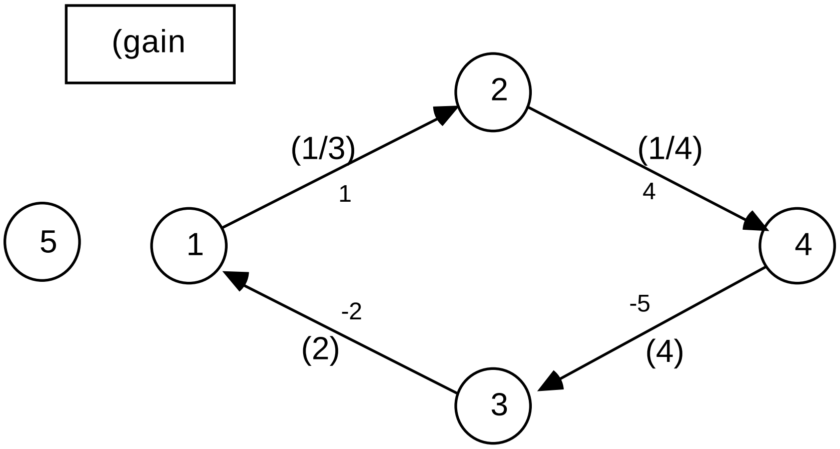


Figure 8. Example basis with gains

To compute the basis inverse for the example it is best to list the arcs and nodes in descendant order and then construct a table for computing the product of arc gains, γ_{ki}, as illustrated below. The elements of the table are computed one column at a time. The column for node *i* is constructed by tracing the path entering node *i* backwards using the predecessor arc defined for each node. The γ_{ki} quantities can then be easily computed in a cumulative fashion. When the path to a node includes a cycle, the cycle gain, β, and corresponding multiplier, η, are shown at the bottom of the column. In the following table we use the values of γ_{ki} and Eq. (2) to compute the basis inverse.

Computation of γ_{ki}

		Nodes			
Arc	Arc gain	2	4	3	1
1	1/3	1/3	1/12	1/3	2/3
4	1/4	2/3	1/4	1	2
–5	4	8/3	2/3	4	8
–2	2	2/3	1/6	2/3	2
	β	2/3	2/3	2/3	2/3
	η	–2	–2	–2	–2

Computation of the basis inverse

	Nodes			
Arcs	2	4	3	1
1	6	24	6	3
4	3	8	2	1
–5	3/4	3	1/2	1/4
–2	3	12	3	1

Although we have shown how to compute the basis inverse for the generalized network problem, computer programs implementing the simplex algorithm will not actually maintain this inverse for computational processes, rather, each column will be computed when needed directly from the tree representation.

Nonbasic Variables

The arcs not selected as basic arcs are the nonbasic arcs. In a basic solution, each nonbasic arc, *k*, will have its flow at zero, the lower bound, or u_k, the upper bound. Let the set $\boldsymbol{n}_0$ be the set of nonbasic arcs with zero flow, and let $\boldsymbol{n}_1$ be the set of arcs with upper bound flow. To represent a specific case graphically we will show the members of $\boldsymbol{n}_1$ as dotted lines.

Primal Basic Solution

Given a selection of basic arcs and an assignment of nonbasic arcs to either $\boldsymbol{n}_0$ or $\boldsymbol{n}_1$, there is a unique assignment of flows to the basic arcs. Let $\mathbf{x}_\mathrm{B}$ be the vector of flows on the basic arcs.

To solve for the basic arc flows, we must first adjust the external flows for the flows in the nonbasic arcs at their upper bounds. Eq. 3 defines an adjusted external flow for node i, b_i', that is the original external flow reduced by the flow of the upper bound arcs leaving the node and increased by the flow on the upper bound arcs entering the node.

$$b_i' = b_i - \sum_{j \; (\boldsymbol{K}_{\mathrm{O}(i)} \cap \boldsymbol{n}_1)} u_j + \sum_{j \; (\boldsymbol{K}_{\mathrm{T}(i)} \cap \boldsymbol{n}_1)} g_k u_j \qquad (3)$$

Let ρ_i be column i of the basis inverse. Then the basic solution is

$$\mathbf{x}_\mathrm{B} = \sum_{i=1}^{m-1} \rho_i b_i' \qquad (4)$$

For the basis given for the example problem the basic variables are

Node 1 4

$$\mathbf{x}_\mathrm{B} = \begin{bmatrix} x_1 \\ x_4 \\ x_{-5} \\ x_{-2} \end{bmatrix} = 4 \begin{bmatrix} 3 \\ 1 \\ 1/4 \\ 1 \end{bmatrix} + (-3/8) \begin{bmatrix} 24 \\ 8 \\ 3 \\ 12 \end{bmatrix} = \begin{bmatrix} 3 \\ 1 \\ -0.125 \\ -0.5 \end{bmatrix}$$

Transforming the mirror arc flows to obtain the original flows using the equation $x_k = -x_{-k} / g_k$, we obtain the basic flows shown in Fig. 9.

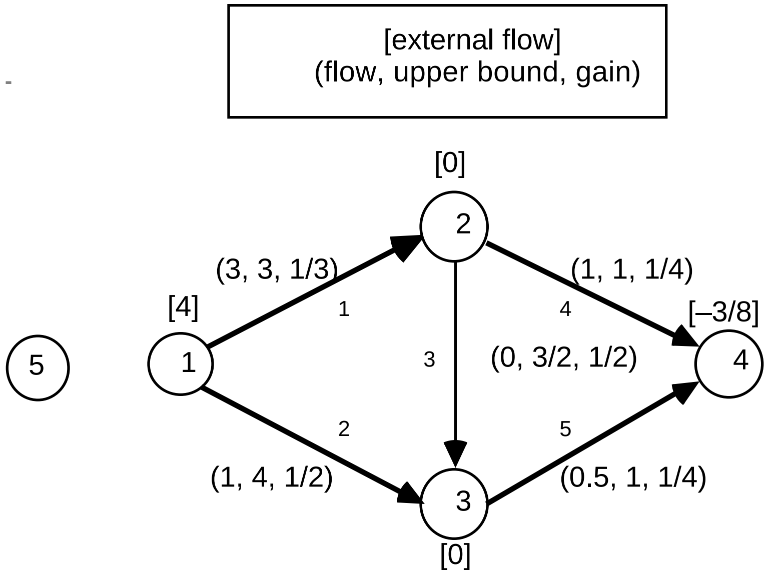


Figure 9. Primal network for example basis

For this computation, it is only necessary to compute the columns of the basis inverse that are associated with nonzero node external flows.

Computing the Dual Variables

The dual variables are identified with the nodes of the network. Thus we call π_i , the dual variable, or alternatively the node cost, for node *i*. The complementary slackness condition of linear programming requires that the dual constraints associated with the basic variables of the primal must be tight. Thus for each basic arc, *k*(*i*, *j*), the following constraint must be satisfied.

$$\pi_i - g_k \pi_j = c_k$$

For that part of the basis comprised of the tree rooted at the slack node, the dual variables are easily determined by first assigning the slack node the value zero.

$$\pi_m = 0$$

Then for each basic arc in the tree, *k*(*i*, *j*), where $i = o_k$ and $j = t_k$, assign dual values so that

$$\pi_j = (\pi_i + c_k)/g_k. \tag{5}$$

When the dual variables are assigned in order of the descendant vector of the slack node, this expression is used sequentially to assign dual variables to all the nodes in the subtree of the slack node. For the example case of Fig. 5a.

$$\boldsymbol{I}_d = \{s, 1, 2, 3, 4\}.$$

Following the order prescribed by this vector

$$\pi_s = 0$$

$$\pi_1 = (\pi_s + c_1)/g_1$$

$$\pi_2 = (\pi_1 + c_2)/g_2$$

$$\pi_3 = (\pi_1 + c_3)/g_3$$

$$\pi_4 = (\pi_2 + c_4)/g_4$$

For components rooted at a cycle, the process is a little more complicated. For the example of Fig. 5b, the following conditions must hold relating the nodes on the cycle.

$$\pi_1 = (\pi_3 + c_1)/g_1$$
$$\pi_2 = (\pi_1 + c_2)/g_2$$
$$\pi_3 = (\pi_2 + c_3)/g_3$$

Solving for π_1 we discover that

$$\pi_1 = (c_2 + g_2c_3 + g_2g_3c_1)/(\beta - 1)$$

Once one of the dual variables on the cycle is determined, all others can be determined using Eq (4). In general, for any node on a cycle the dual variable is

$$\pi_i = \frac{\text{cost of routing one unit around the cycle starting at node } i}{\beta - 1}$$

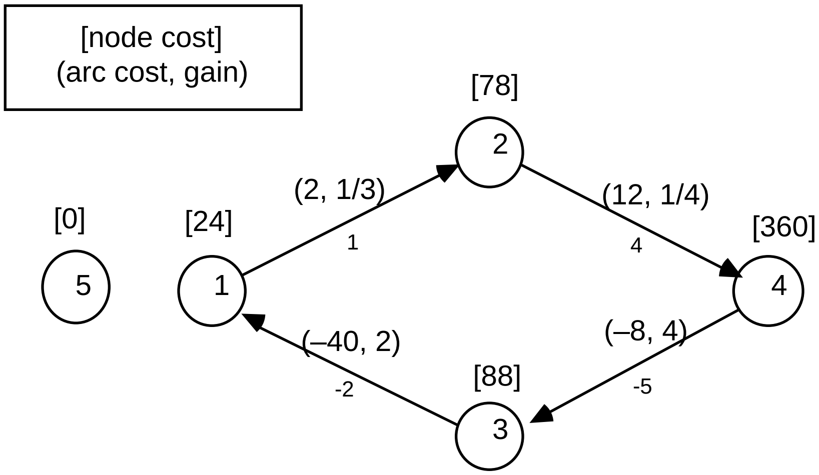


Figure 10. Dual network for the example

For the example shown in Fig. 10, all nodes are on a single cycle. Mirror arc costs are calculated from original arcs as: $c_{-k} = -c_k/g_k$. We have β as 2/3. First determining π_1, we compute the cost of routing one unit of flow around the cycle starting at node 1.

$$\begin{aligned} &c_1 + g_1c_4 + g_1g_4c_{-5} + g_1g_4g_{-5}c_{-2} \\ &\qquad = 2 + (1/3)(12) + (1/3)(1/4)(-8) + (1/3)(1/4)(4)(-40) \\ &\qquad = -8 \end{aligned}$$

Dividing this number by β –1 we obtain

$$\pi_1 = 24.$$

The other dual variables are computed with Eq. (5).

$$\pi_2 = (24 + 2)/(1/3) = 78,$$

$$\pi_4 = (78 + 12)/(1/4) = 360,$$

$$\pi_3 = (360 - 8)/4 = 88.$$

The vector of dual variables is then

$$\pi = (\pi_1, \pi_2, \pi_3, \pi_4, \pi_5) = (24, 78, 88, 360, 0),$$

Arc Marginal Cost and Conditions for Optimality

For the generalized network, the marginal cost for the general nonbasic arc $k(i, j)$ is

$$d_k = c_k + \pi_i - g_k \pi_j . \tag{5}$$

If the solution is optimal, for each nonbasic arc $k(i, j)$ one of the following conditions must hold:

$$\text{If } x_k = 0, \text{ then } d_k \geq 0. \tag{6a}$$

$$\text{If } x_k = u_k , \text{ then } d_k \leq 0. \tag{6b}$$

When some nonbasic arcs do not satisfy the condition, the solution is not optimal, and an arc, which violates the optimality condition, must be selected to enter the basis. For the bounded variable simplex method a variable can enter the basis by going up from zero or by going down from its upper bound. Actually any nonbasic variable violating an optimality condition may be selected, however, for simplicity we choose the variable which violates the optimality condition the most. Thus we will choose the entering variable k^* according to the rule

$$d_k{}^* = \max \begin{cases} \max\{-d_k : d_k < 0 \text{ and } k \quad \boldsymbol{n}_0\} \\ \max\{d_k : d_k > 0 \text{ and } k \quad \boldsymbol{n}_1\} \end{cases} \tag{7}$$

For the example problem the only nonbasic arc is arc 3(2, 3). The marginal cost for this arc is

$$\begin{aligned} d_3 &= \pi_2 + c_3 - g_3\pi_3 \\ &= 78 + 1 - (0.5)(88) \\ &= 35. \end{aligned}$$

Since the marginal cost is positive, and the flow in arc 3 is zero, this arc satisfies the optimality condition. Since this is the only nonbasic arc, the solution given must be optimal.

C.3 Primal Simplex

There are a number of aspects of the primal simplex algorithm that are the same for the pure and generalized problem. After a review of the similarities, we consider the primary difference, determining the arc to leave the basis. We then present an example of a complete generalized network solution.

The Primal Simplex Algorithm

The overall algorithm implementing the primal simplex remains the same as for the pure problem. The algorithm begins with a basic feasible solution. After the initial primal and dual solutions and the initial basis subnetworks are constructed, the algorithm proceeds iteratively as follows:

- Check the solution for optimality.
- If not optimum, select an arc to enter the basis.
- Determine the arc that must leave the basis.
- Change the basis subnetwork by deleting the leaving arc and adding the entering arc.
- Compute the new primal and dual solutions.

The Initial Solution

As for the pure problem, one can form an initial basic solution by adding an artificial arc for each node. If some node *i* has negative or zero external flow, the artificial arc is directed from the slack node to node *i*. If node *i* has positive external flow, the artificial arc is directed from node *i* to the slack node. The gains of the artificial arcs are set to one. Initially the artificial arcs carry the external flows. The initial basis subnetwork consists only of the artificial arcs, so the initial basis forms a tree. The simplex algorithm is implemented in two phases, with the first driving out the artificial arcs and the second obtaining the optimal solution.

The Entering Arc

The process of selecting the entering arc uses the marginal cost associated with each nonbasic arc. The marginal cost for arc $k(i, j)$ is as follows.

$$d_k = \pi_i + c_k - g_k\pi_j$$

Although, the computation of the marginal cost is slightly different than for the pure problem since it involves the gain factor, the conditions for optimality remain the same as for the pure problem. Any arc that does not satisfy the conditions may enter the basis. The strategies for the selecting the entering arc remain an optional part of the algorithm and can follow the approaches for the pure problem or for more general linear programs.

The Leaving Arc

Structure

Given an arc to enter the basis, some basic arc must leave. For the pure problem, the entering arc always forms a cycle when inserted in the basis subnetwork. Because a flow change on the entering arc only affects basic arcs on the cycle, the leaving arc must be one of these arcs.

For the generalized problem, the basis subnetwork consists of a tree rooted at the slack node and perhaps other components in the form of 1-trees. Because of the nonunity arc gains, a flow change in the entering arc will cause unequal flow changes on basic arcs, and the leaving arc may be in a variety of locations. Fig. 11 illustrates the possible situations that can arise.

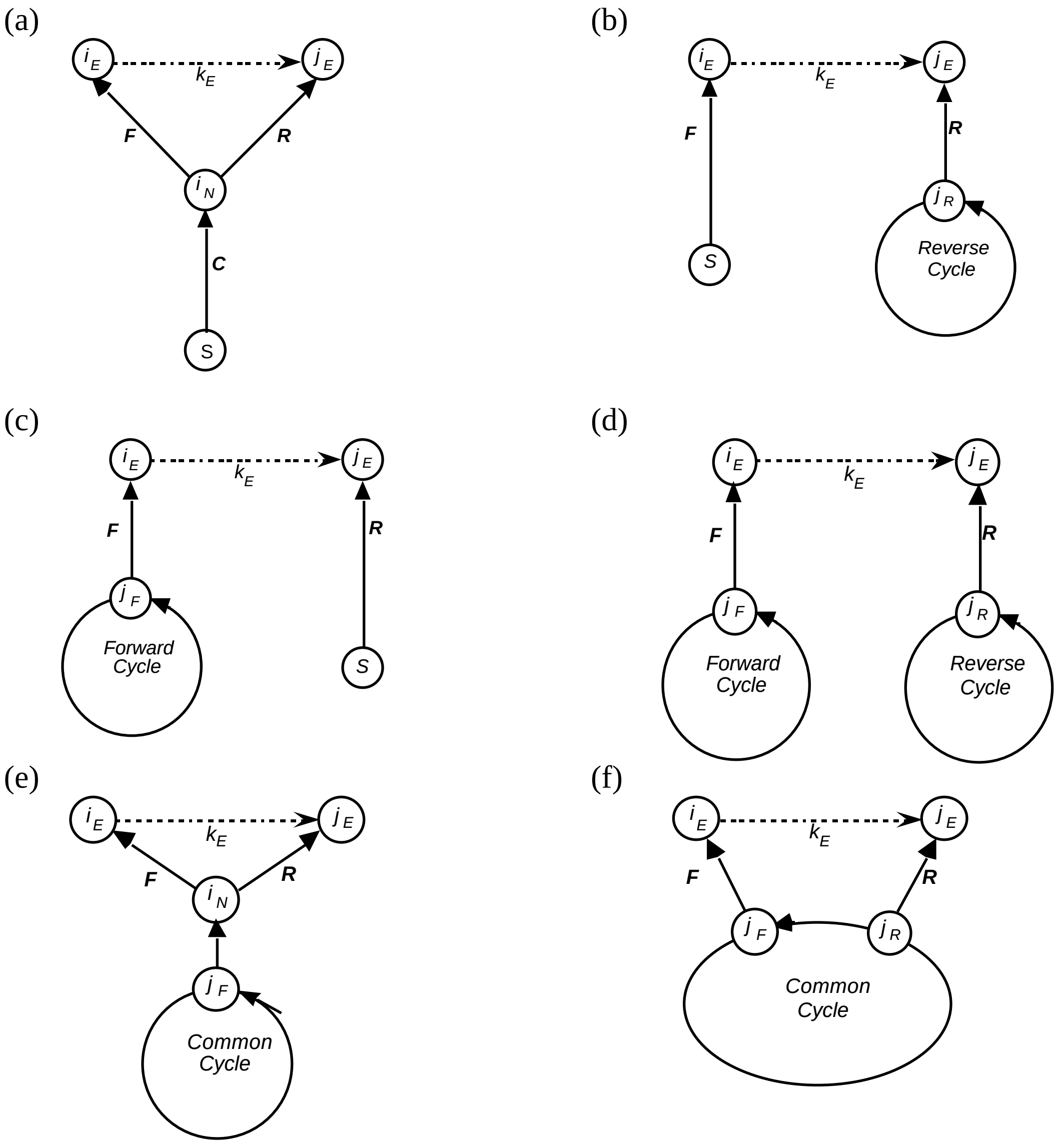


Figure 11. Possible basis configurations with the entering arc

In the figure we assume that flow is increasing in the entering arc. This causes flow to increase in the forward paths, labeled ***F***, and decrease in the reverse paths, labeled ***R***. The paths labeled ***C*** are common to the forward and reverse paths. The direction and amounts of flow change in ***C*** depend on the relative gains of the forward and reverse paths.

Fig. 11a illustrates the only case possible for the pure problem. Here the entering arc connects two nodes on the tree forming the cycle indicated by ***F*** and ***R***. The leaving arc must be in one of these two paths. For the generalized network when the gains of the paths are not the same, flow will also change in path ***C***, and the leaving arc may be in this segment. If the leaving arc does happen to be in path ***C***, the new basis subnetwork will have a new 1-tree with the cycle formed by ***F***, ***R***, and k_E**.**

In Fig. 11b, c, and d, the entering arc connects two components of the basis subnetwork. In case b, increasing flow in the entering arc draws additional flow from the slack node, and flow is decreased in the reverse path. The flow change must be *absorbed* in the cycle. In the other cases flow is either absorbed or generated by the cycles, represented as a decrease or increase of flow on the cycle arcs. Flow generation or absorption in cycles is possible only because the cycle gain is not unity. The leaving arc may be in any path shown on these figures. After the basis change, the basis subnetwork may take a variety of forms. When a cycle arc leaves, a 1-tree is destroyed and two components become one.

In Fig. 11e and f, the forward and reverse paths are in the same component rooted at a cycle. When the leaving arc is on the cycle, the entering arc will form a new cycle and the component will remain as a 1-tree.

Computing the Marginal Flow Change

After an arc has been selected to enter the basis, we move to an adjacent feasible basic solution by increasing or decreasing the flow in the entering arc. While changing the flow on the entering arc, the flows in some of the basic arcs must also change in order to maintain conservation of flow at the nodes. The flow changes in the entering arc by an amount that will just drive the flow on one of the basic arcs to zero or to its upper bound, or drive the flow on the entering arc to zero or to its upper bound.

Although algorithms that implement this step construct the list of arcs and their marginal flow changes from the basis tree itself, we resort to using the columns of the basis inverse as discussed in the last section. Let ρ_i be column *i* of the basis inverse. An element ρ_{hi} is the marginal decrease in the flow in basic arc *h* per unit increase in the flow withdrawn from node *i.* As discussed in the last section, the components of the inverse are computed directly from the basis subnetwork whenever required.

Say we have identified an arc $k(i$, j) with gain g_k to enter the basis by increasing its flow from zero by an amount Δ_f. This causes an increase in the flow withdrawn from node i of Δ_f, and a decrease in the flow withdrawn from node j of $g_k\Delta_f$. Assume that the current basic flows are described by the vector $\mathbf{x}'_B$. The new flows caused by a change of Δ_f will be

$$\mathbf{x}'_B = \mathbf{x}_B\Delta_f\,(\rho_i - g_k\rho_j)$$

We identify the terms in the parenthesis as the marginal change vector $\mathbf{y}_k$. When the flow in the entering arc k is to increase from zero, the marginal change vector is given by Eq. (8).

$$\text{If } x_k \text{ is increasing: } \mathbf{y}_k = \rho_i - g_k\rho_j. \tag{8}$$

Note that this corresponds to a similar definition of $\mathbf{y}_k$ from general linear programming theory. In general, $\mathbf{y}_k = \mathbf{B}^{-1}\mathbf{a}_k$. Since the for the generalized network problem, the vector $\mathbf{a}_k$ for arc $k(i, j)$ has a +1 coefficient for node i, and a $-g_k$ coefficient in row j, the result for a network problem is as noted in Eq. (8).

The arc entering the basis may be decreasing from its upper bound. In this case we will still take the flow change, Δ_f, as a positive quantity, but we will define the marginal change to be the negative of Eq. (8).

$$\text{If } x_k \text{ is decreasing: } \mathbf{y}_k = -\rho_i + g_k\rho_j \tag{9}$$

Identifying the arc to leave the basis

In order to determine the arc to leave the basis, we must write explicitly the basic flows as a function of the flow change. We have chosen to always show flow in the forward arc direction, regardless of the orientation of the arc in the basis subnetwork. A basic arc, h, may appear as a forward arc ($h > 0$) or as a mirror arc ($h < 0$), but we will always refer to the flow in the arc in the forward direction $x_{|h|}$.

The vector $\mathbf{y}_k$ has component, y_{hk}, that indicates the marginal change in arc h as arc k enters the basis. Let $\underline{x}_{|h|}$ be the current flow in arc $|h|$. With the flow change, the new flow in the arc depends on whether it appears as a forward or mirror arc in the basis.

$$x_{|h|} = \begin{cases} \underline{x}_{|h|} - \Delta_f y_{hk} & \text{if } h > 0 \\ \underline{x}_{|h|} + \Delta_f g_h y_{hk} & \text{if } h < 0 \end{cases} \tag{10}$$

For feasibility the new flow must between zero and the upper bound:

$$0 \le x_{|h|} \le u_{|h|}$$

Considering the signs of the various factors this leads to the following conditions that must be satisfied by the flow change.

Forward Arc (*h* > 0)	$y_{hk} > 0$ $\Delta_f \le \underline{x}_h/y_{hk}$	$y_{hk} < 0$ $\Delta_f \le -(u_h - \underline{x}_h)/y_{hk}$
Mirror Arc (*h* < 0)	$g_h y_{hk} > 0$ $\Delta_f \le (u_{\lvert h\rvert} - x_{\lvert h\rvert})/g_h y_{hk}$	$g_h y_{hk} < 0$ $\Delta_f \le -(x_{\lvert h\rvert})/g_h y_{hk}$

Thus we have a quite complicated ratio test that depends on the orientation of the arcs in the basis and the signs of the marginal change and gain factors.

$$\Delta_f = \mathrm{Min}\left\{\begin{array}{l} \min\{\underline{x}_h/y_{hk} : y_{hk} > 0 \text{ and } h > 0\} \\ \min\{-(u_h - \underline{x}_h)/y_{hk} : y_{hk} < 0 \text{ and } h > 0\} \\ \min\{(u_{|h|} - x_{|h|})/g_h y_{hk} : g_h y_{hk} > 0 \text{ and } h < 0\} \\ \min\{-(x_{|h|})/g_h y_{hk} : g_h y_{hk} < 0 \text{ and } h < 0\} \\ u_k \end{array}\right\}\begin{array}{l} 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{array} \qquad (11)$$

The arc leaving the basis is the arc *h* for which the minimum is obtained. In cases 1 and 4, arc *h* leaves the basis by going to zero. In cases 2 and 3 the arc leaves the basis by going to its upper bound. Case 5 indicates that no arc leaves the basis as the flow goes to its opposite bound.

Illustration

Although we have an optimal solution for the example, we will allow arc 3 to enter the basis to illustrate this procedure. The basis inverse obtained for the example in the last section is shown again below. The columns are identified by node number and the rows are identified with the basic arcs.

Basis inverse

	Nodes			
Arcs	2	4	3	1
1	6	24	6	3
4	3	8	2	1
–5	3/4	3	1/2	1/4
–2	3	12	3	1

Since arc 3(2,3) is to enter the basis, the marginal change vector is computed from Eq. (1) as

$$\mathbf{y}_3 = \rho_2 - g_3\rho_3.$$

$$\mathbf{y}_3 = \begin{pmatrix} 6 \\ 3 \\ 3/4 \\ 3 \end{pmatrix} - (1/2)\begin{pmatrix} 6 \\ 2 \\ 1/2 \\ 3 \end{pmatrix} = \begin{pmatrix} 3 \\ 2 \\ 1/2 \\ 3/2 \end{pmatrix}.$$

The table below helps to perform the ratio test. The case column refers to the line of Eq. (11) used to compute the ratio.

Basic arc (h)	gain (g_h)	y_{hk}	Case	$x_{\|h\|}$	Ratio
1	1/3	3	1	3	1
4	1/4	2	1	1	1/2
–5	4	1/2	3	1/2	1/4
-4	2	3/2	3	1	1

The minimum ratio is 1/4, obtained for arc –5. Because 1/4 < 3/2, the upper bound for arc 3, arc 5 will leave the basis $\Delta_f = 1/4$. Computing the new flows, we have

for forward arcs: $x_h = x_h' - \Delta_f y_{hk}$

$x_1 = 3 - 3(1/4) = 9/4$

$x_4 = 1 - 2(1/4) = 1/2$

for mirror arcs: $x_{-h} = x_{-h}' + \Delta_f y_{hk} g_h$

$x_5 = 1/2 + (1/2)(4)(1/4) = 1$

$x_2 = 1 + (3/2)(2)(1/4) = 7/4$

for the entering arc: $x_k = \Delta_f$, $x_3 = 1/4$.

These flows are shown in Fig. 12.

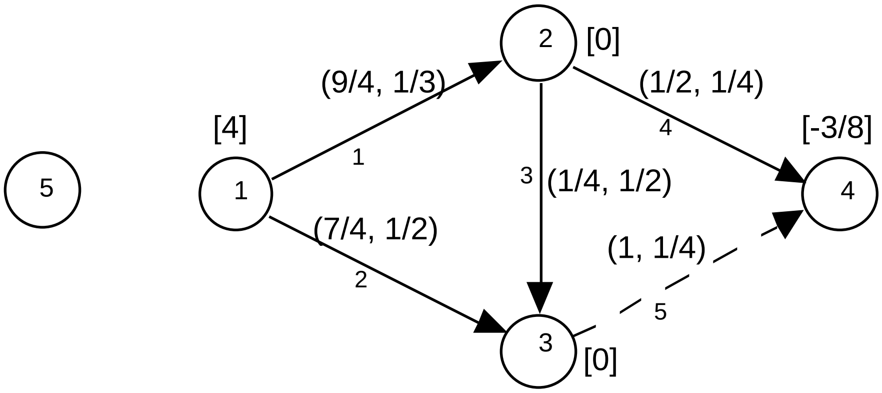


Figure 12. Arc flows for the basis when arc 3 is added and arc 5 is removed

Changing the Basis Subnetwork

Although Fig. 11 illustrates the complexity that can arise concerning the entering and leaving arc, the actual process of forming the new basis subnetwork is the same for the generalized network as for the pure network. Given the identity of the entering and leaving arc, the basis subnetwork is changed by:

- Delete the leaving arc.
- Add the entering arc.
- Reverse the orientation of all arcs between the terminal node of the entering arc and the terminal node of the entering arc.

The last step of this procedure assures that each component of the basis subnetwork is either a directed tree rooted at the slack node, or a directed 1-tree rooted at a cycle. Fig.13 shows the new basis when arc 3 enters and arc –5 leaves. We will see a number of other examples of this in the problem that appears at the end of this section.

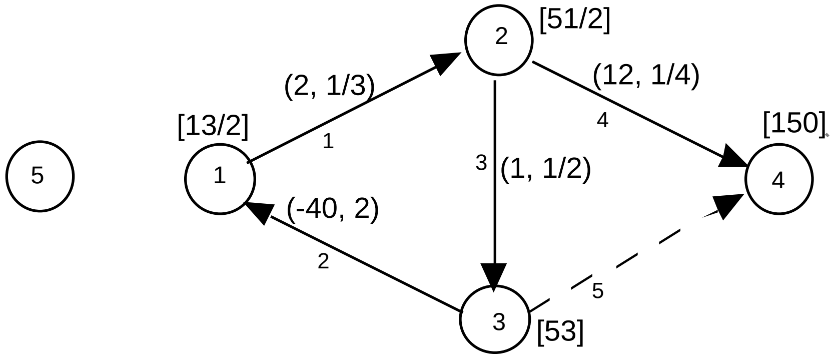


Figure 13. New basis subnetwork formed by adding arc 3 and removing arc 5

Compute the New Primal and Dual Solution

The new primal solution is obtained using the marginal equations used earlier in the development of the ratio test.
For the basis arcs

$$x_{|h|} = \begin{cases} \underline{x}_{|h|} - \Delta_f y_{hk} & \text{if } h > 0 \\ \underline{x}_{|h|} + \Delta_f g_h y_{hk} & \text{if } h < 0 \end{cases} \qquad (12)$$

For the entering arc

If x_k is increasing from zero: $x_k = \Delta_f$.

If x_k is decreasing from its bound: $x_k = u_k - \Delta_f$.

The new dual solution need be computed for only the portion of the basis tree that has changed. Let $k(i, j)$ be entering the basis. Define a node r as the root node. If k appears in the tree in the forward direction, the component of the tree that is rooted at node *j* will have revised dual values. In this case let $r = j$. If *k* appears in the mirror direction in the new subnetwork (–*k* is in the subnetwork), the component of the tree rooted at node *i* will have changed dual values, so let $r = i$.

List the successor nodes to node *r* in successor order, and use the defining equations to compute the new dual values for the tree rooted at *r*.

Complete Algorithm

To illustrate the simplex algorithm applied to the generalized network, we show the sequence of basic subnetworks obtained for the example network given in Fig. 14 starting from a completely artificial basis.

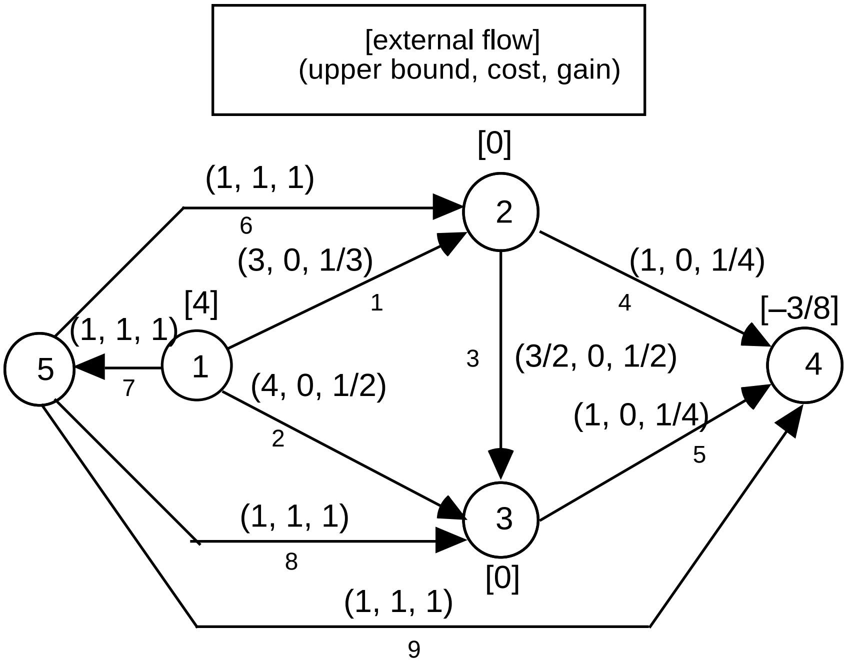
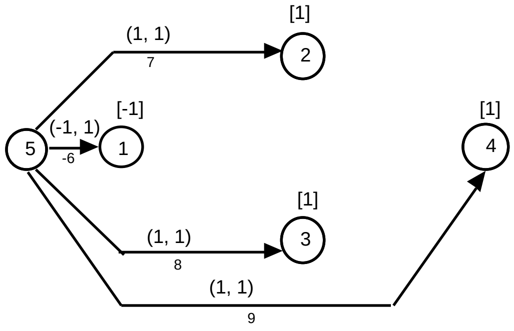
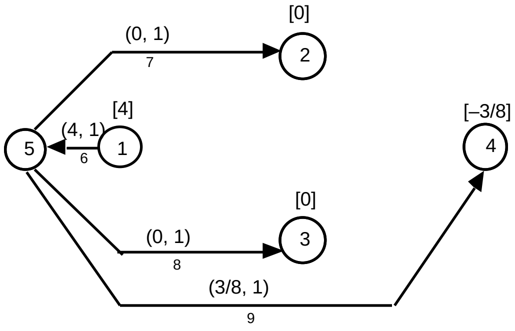
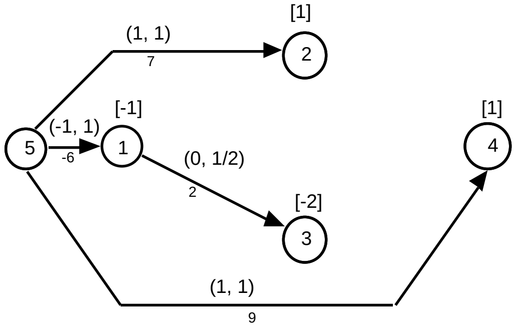
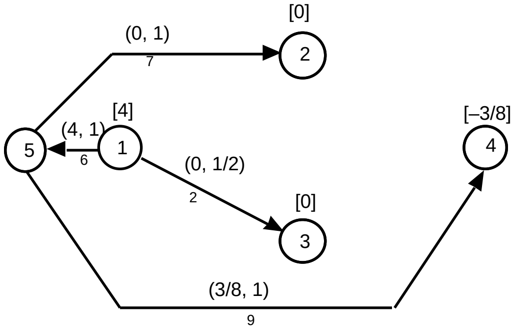
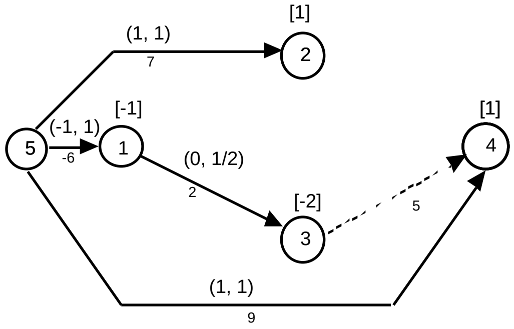
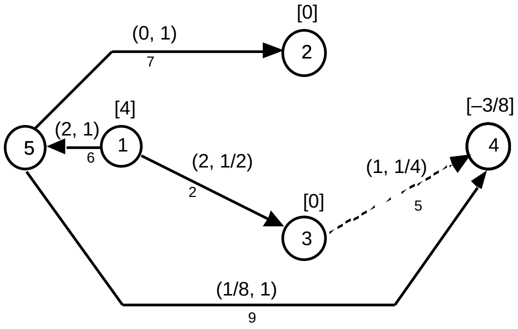
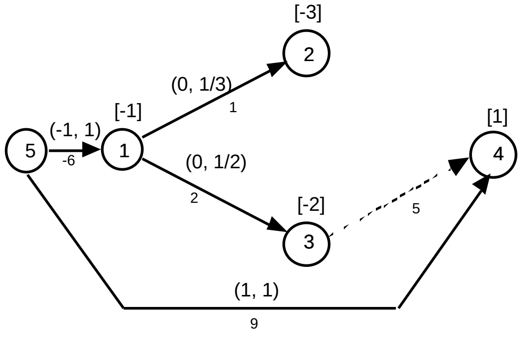
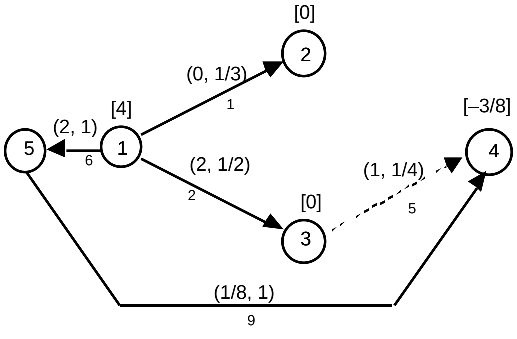
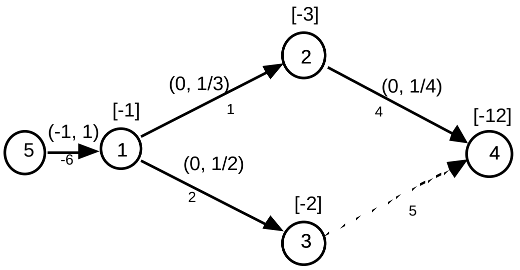
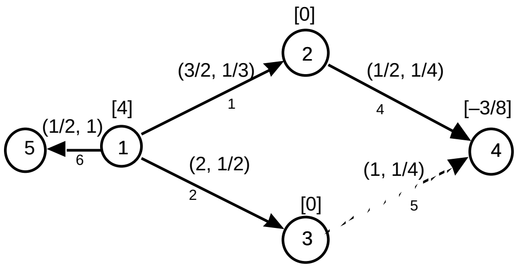


Figure 14. Example network with artificial arcs

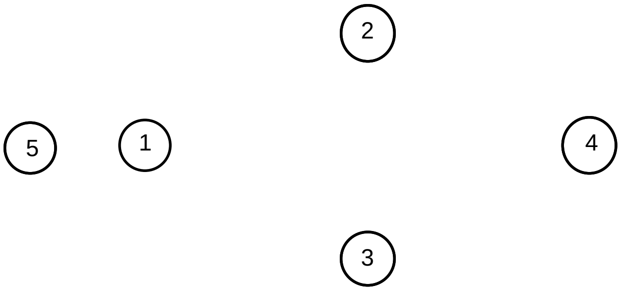
Dual Network [Node Cost] (Arc Cost, Arc Gain)	**Primal Network** [Fixed External Flow] (Arc Flow, Arc Gain)
Initial Artificial Basis  [1] (1, 1) 7 2 [-1] (-1, 1) 5 -6 1 [1] 4 [1] (1, 1) 8 3 (1, 1) 9	 [0] (0, 1) 7 2 [4] (4, 1) 5 6 1 [–3/8] 4 [0] (0, 1) 8 3 (3/8, 1) 9
Arc 2 enters the basis and arc 8 leaves  [1] (1, 1) 7 2 [-1] (-1, 1) 5 -6 1 [1] 4 (0, 1/2) 2 [-2] 3 (1, 1) 9	 [0] (0, 1) 7 2 [4] (4, 1) 5 6 1 [–3/8] 4 (0, 1/2) 2 [0] 3 (3/8, 1) 9
Arc 5 enters the basis and arc 5 leaves  [1] (1, 1) 7 2 [-1] (-1, 1) 5 -6 1 [1] 4 (0, 1/2) 2 [-2] 3 5 (1, 1) 9	 [0] (0, 1) 7 2 [4] (2, 1) 5 6 1 [–3/8] 4 (2, 1/2) 2 (1, 1/4) [0] 3 5 (1/8, 1) 9
Arc 1 enters the basis and arc 7 leaves  [-3] 2 (0, 1/3) 1 [-1] (-1, 1) 5 -6 1 [1] 4 (0, 1/2) 2 [-2] 3 5 (1, 1) 9	 [0] 2 (0, 1/3) 1 [4] (2, 1) 5 6 1 [–3/8] 4 (2, 1/2) 2 (1, 1/4) [0] 3 5 (1/8, 1) 9

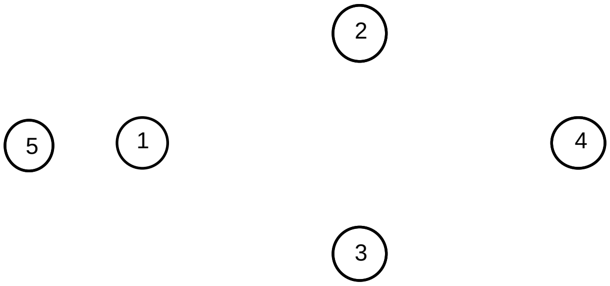
Arc 4 enters the basis and arc 9 leaves



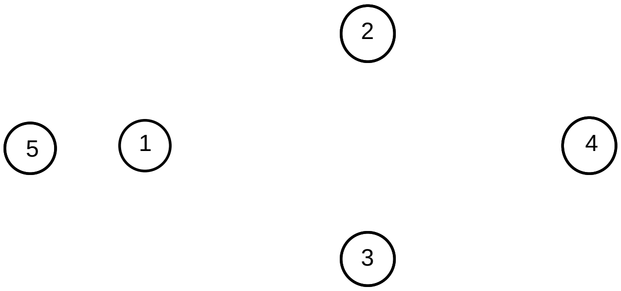


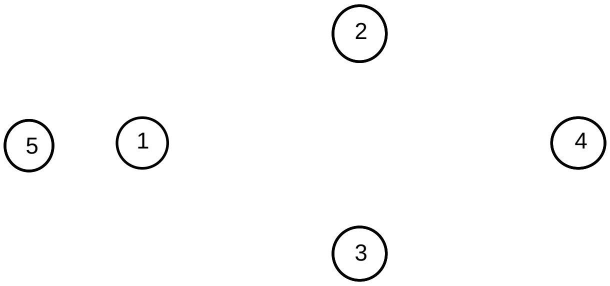
Arc 3 enters the basis and arc 1 leaves



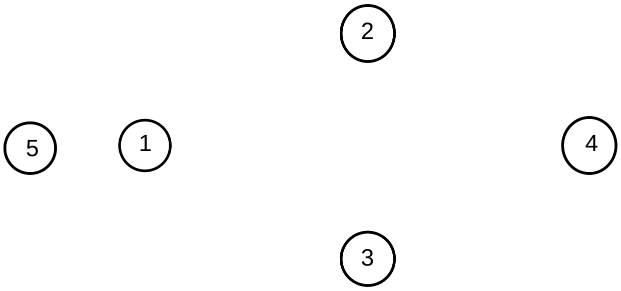


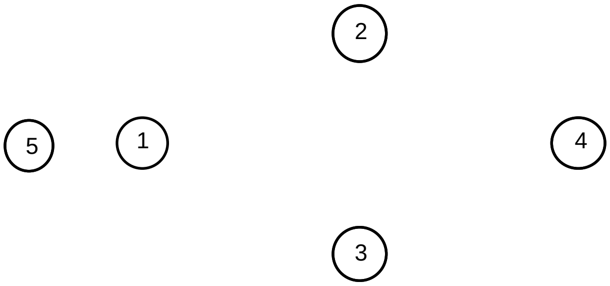
End phase 1 and begin phase 2





Arc -4 enters the basis and arc 3 leaves





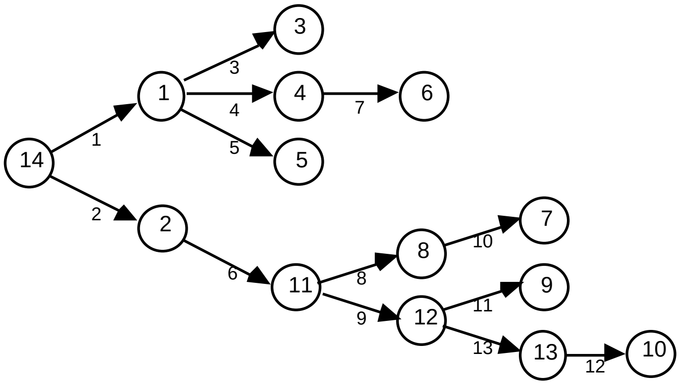
C.4 Exercises

1. For the example of Fig. 1, show the basis and basis inverse for each of the basis subnetworks shown in Fig. 4.

2. The tree shown below represents a basic solution to some network problem. Arc (13,9) is chosen to enter the basis. List the set of arcs from which the leaving arc must be chosen for each of the problem classes listed.

   a. Generalized minimum cost flow:

   b. Pure minimum cost flow:

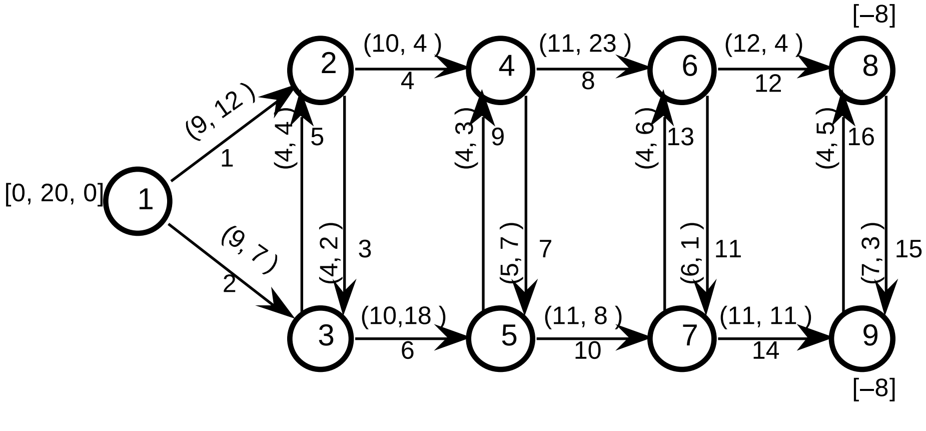


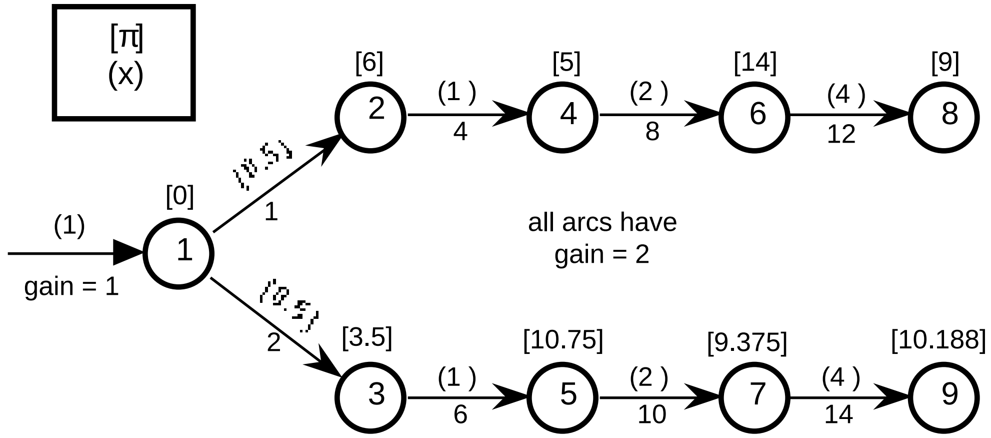
3. Consider the network below with the added information that all arc gains are 2. For each basis show the basis tree, primal variables, and dual variables. All nonbasic variables have 0 flow. Select the arc to enter the basis for the next iteration.

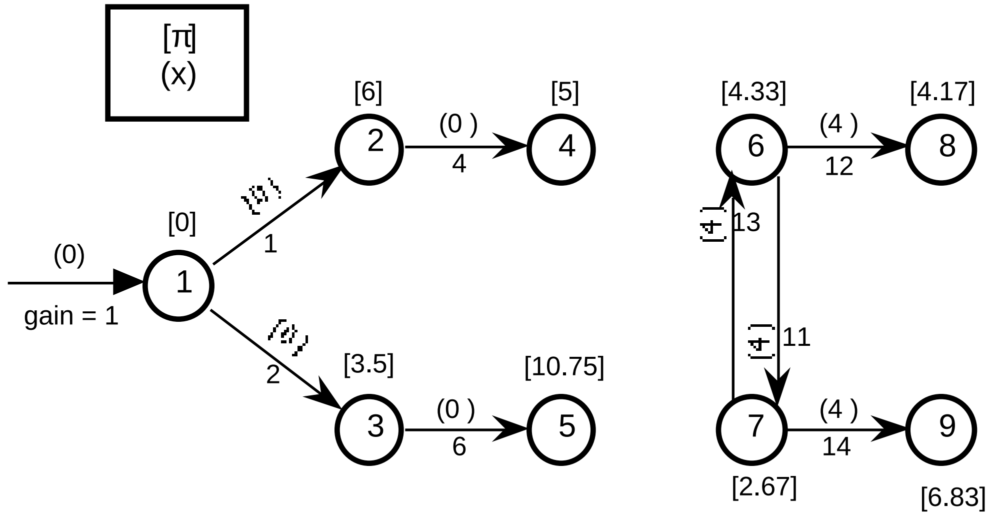
   a. $\mathbf{n}_B$ = { 1, 2, 4, 6, 8, 10, 12, 14}

   b. $\mathbf{n}_B$ = { 1, 2, 4, 6, 11, 12, 13, 14}

[Fixed flow, slack flow, slack cost]
(upper bound, cost)

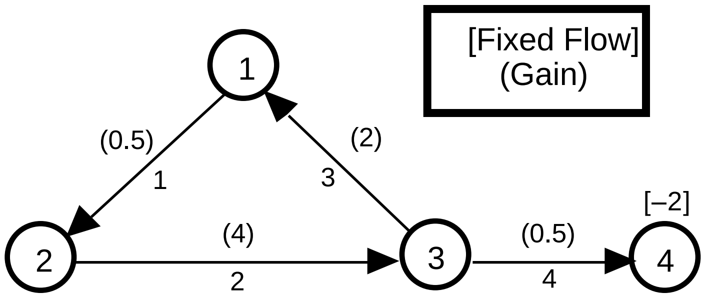




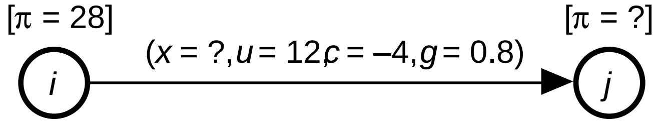


4. The LP formulation of a network problem includes the set of linear conservation of flow equations: **Ax** = **b**, where **A** is the incidence matrix defined by the arcs of the network, **b** is a vector of external flows and **x** is the flow vector. When a basis is defined and all nonbasic flows are zero, the set of equations: $\mathbf{B}\mathbf{x}_B = \mathbf{b}$ is appropriate to determine the basic flows $\mathbf{x}_B$.

The network below shows part of the basis for a generalized network problem. The arc gains and external flows are as shown. Write the set of equations $\mathbf{B}\mathbf{x}_B = \mathbf{b}$ and solve them for the basic flows.



5. The figure provides some information about a general arc *k* that originates at node *i* and terminates at node *j*. Each part of the problem gives some additional information. Given the conditions stated in each part, say as much as you can about the unknowns x_k and π_j. The parts are independent and the conditions given in one part do not hold in the other parts.



   a. Arc *k* is basic.

   b. Arc *k* is nonbasic with x_k = 12. The solution is optimal.

   c. Arc *k* is nonbasic, the solution is optimal and π_j = 27.

   d Arc *k* is chosen to enter the basis with π_j = 23.

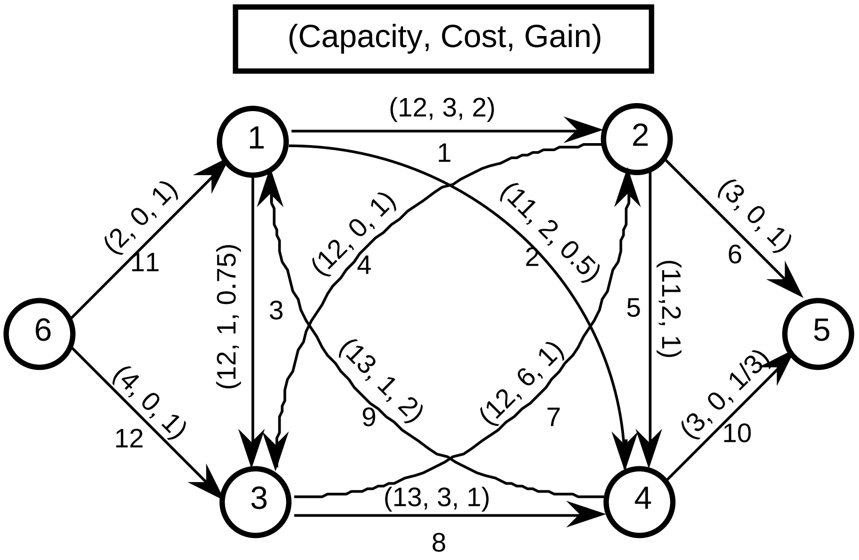
6. The figure below shows a generalized network. Node 6 is the slack node. All flows are zero. Let the basis consist of arcs 3, 6, 8, 9, and 10.

   a. Compute the node potentials (π).

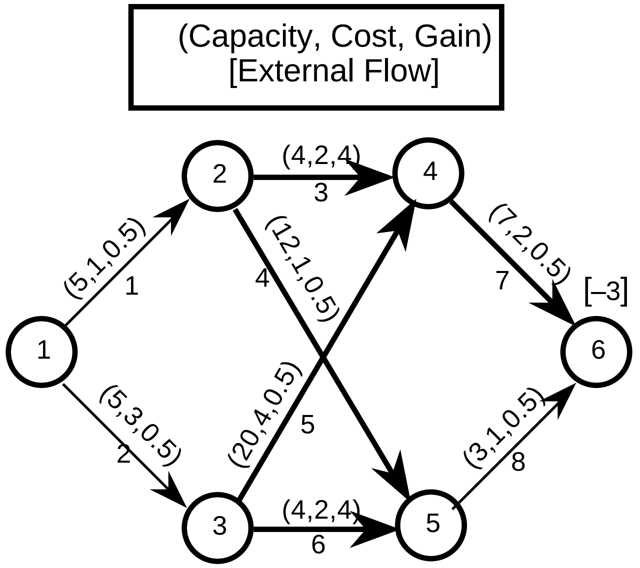
   b. Compute the flows on the basic arcs for an external flow of 1 at node 5. All nonbasic flows are zero.

   c. List the tree arcs and tree nodes in predecessor order. Also show the depth of each node.

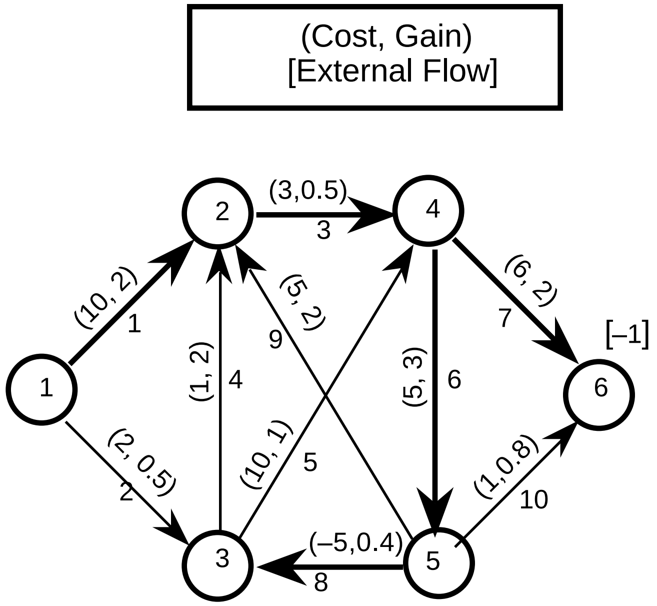
   d. Let arc 4 enter the basis and arc 3 leave. Sketch the new basis network, and list the nodes and arcs in predecessor order.



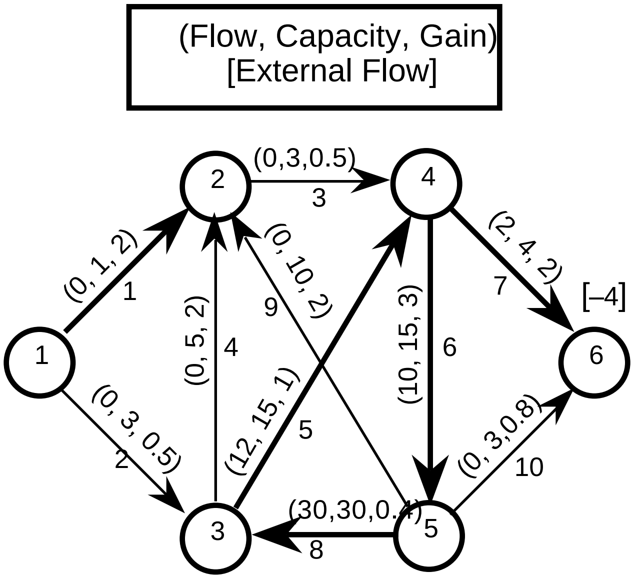
7. The figure shows a generalized minimum cost flow problem. Using the basis shown by the heavier lines, compute the flows on the basic arcs and the node costs. Assume all nonbasic arcs have zero flow. Check the solution for optimality. If it is not optimal identify an arc to enter the basis. Select the arc to leave the basis, and construct the new basis tree. Node 1 is the slack node.



8. The figure shows a generalized network with infinite capacity on all arcs. The heavy lines show the initial basis. The arc to enter the basis using the "most negative" rule. Change the basis. Show the new basis and new node costs. Node 1 is the slack node.



9. The figure shows a generalized network with flows capacities and gains. The heavier lines identify the current basis. Let arc 3 enter the basis. Find the arc to leave the basis. Find the flows associated with the new basic solution. Draw the new basis network. Node 1 is the slack node.


(Flow, Capacity, Gain)
[External Flow]
1
2
3
4
5
6
(0,3,0.5)
3
(0, 1, 2)
1
(0, 5, 2)
4
(0, 10, 2)
9
(10, 15, 3)
6
(2, 4, 2)
7
[–4]
(0, 3, 0.5)
2
(12, 15, 1)
5
(0, 3,0.8)
10
(30,30,0.4)
8