Linear Programming and Network Flow, Fourth Edition
by Mokhtar S. Bazaraa, John J. Jarvis and Hanif D.Sherali

ELEVEN: THE OUT–OF–KILTER ALGORITHM

In Chapter 9, we presented a network simplex method for solving minimal–cost network flow problems. In this chapter, we present another method for solving minimal–cost network flow problems, called the out–of–kilter algorithm. This algorithm is similar to the primal–dual algorithm in that it begins with dual feasibility, but not necessarily primal feasibility, and iterates between primal and dual problems until optimality is achieved. However, it differs from the primal–dual algorithm (as strictly interpreted) in that the out–of–kilter algorithm does not always maintain complementary slackness. In fact, the principal thrust is to attain complementary slackness. A version of the algorithm can be designed in which we maintain primal and dual feasibility (not necessarily basic solutions) and strive to achieve complementary slackness. Computationally, the state–of–the–art primal simplex codes run two–three times faster than traditional out–of–kilter codes. However, advances in primal–dual methods that use the basic ingredients of the out–of–kilter algorithm, although in a manner different from this algorithm, have resulted in algorithms that run two–four times faster than the best primal simplex code with the speed–up factor increasing by an order of magnitude for large problems. (We provide some related comments on such procedures in the concluding section of this chapter, and refer the reader to the Notes and References section for further reading on this subject.) From this viewpoint, the concepts of the various algorithmic steps presented in this chapter are important.

11.1 THE OUT–OF–KILTER FORMULATION OF A MINIMAL–COST NETWORK FLOW PROBLEM

For convenience of presentation, the form of the minimal–cost flow problem that we shall work with is:

$$\begin{array}{lll} \text{Minimize} & \sum_{i=1}^{m}\sum_{j=1}^{m} c_{ij}x_{ij} & \\ \text{subject to} & \sum_{j=1}^{m} x_{ij} - \sum_{k=1}^{m} x_{ki} = 0, & i = 1,\ldots,m \\ & x_{ij} \geq \ell_{ij}, & i,j = 1,\ldots,m \\ & x_{ij} \leq u_{ij}, & i,j = 1,\ldots,m, \end{array} \tag{11.1}$$

where it is understood that the sums and bounding inequalities are taken over existing arcs only. We call any flow (choice of the x_{ij}–variables) that satisfy the equality constraints in Problem (11.1) a *conserving flow*. A conserving flow that satisfies the remaining constraints $\ell_{ij} \leq x_{ij} \leq u_{ij}$ is a *feasible flow* (solution).

We shall assume that c_{ij}, ℓ_{ij}, and u_{ij} are integers and that $-\infty < \ell_{ij} \le u_{ij} < \infty$. Since all right–hand–side values of the flow conservation equations in Problem (11.1) are zero, we conclude that the flow in the network does not have a starting point or an ending point, but *circulates* continuously throughout the network. Thus a conserving flow in the network will involve flows along circuits (directed cycles). For this reason, the representation in Problem (11.1) is known as a *circulatory network flow problem.*

The foregoing formulation is completely equivalent to the formulation of the minimal–cost network flow problem presented in Chapter 9. This is readily seen by noting that a minimum cost network flow problem can be first of all put in the form (11.1) with a general (integer) right–hand–side vector **b** instead of a zero vector in the flow conservation constraints, and in particular, with $0 \le \ell_{ij} < u_{ij} < \infty$ for all (i, j). Here, we assume that the variables x_{ij} that are unbounded from above are artificially bounded by using $u_{ij} = M$, where M is sufficiently large. By replacing each right–hand–side magnitude $|b_i|$ with a variable y_i, where y_i is bound restricted as $|b_i| \le y_i \le |b_i|$, it is readily seen how we can transform the given problem into the form (11.1) (see Exercise 11.1). Hence, in particular, note that we may possibly have some $\ell_{ij} = u_{ij}$ in Problem (11.1).

We emphasize here that the homogeneous form in Problem (11.1) is only for the sake of convenience. The same algorithm discussed in the sequel is applicable with a general right–hand–side vector and with $-\infty < \ell_{ij} \le u_{ij} < \infty$, given a starting solution that is flow–conserving (perhaps to an artificial network). More specifically, consider a bounded variables network flow problem in the following general form as discussed in Chapter 9.

$$\text{Minimize } \{\mathbf{cy} : \mathbf{Ay} = \mathbf{b}, \bar{\boldsymbol{\ell}} \le \mathbf{y} \le \bar{\mathbf{u}}\},$$

where **A** is a node–arc incidence matrix, and where $-\infty < \bar{\ell}_{ij} \le \bar{u}_{ij} < \infty$, for all arcs (i, j). Now let $\bar{\mathbf{y}}$ be any arbitrary flow–conserving solution that satisfies $\mathbf{A}\bar{\mathbf{y}} = \mathbf{b}$, and consider the transformation

$$\mathbf{x} = \mathbf{y} - \bar{\mathbf{y}}, \qquad \text{i.e.,} \qquad \mathbf{y} = \bar{\mathbf{y}} + \mathbf{x}.$$

Under this transformation, the preceding problem becomes:

$$\text{Minimize } \{\mathbf{c}(\bar{\mathbf{y}} + \mathbf{x}) : \mathbf{A}\bar{\mathbf{y}} + \mathbf{Ax} = \mathbf{b}, \bar{\boldsymbol{\ell}} - \bar{\mathbf{y}} \le \mathbf{x} \le \bar{\mathbf{u}} - \bar{\mathbf{y}}\}.$$

Dropping the constant term $\mathbf{c}\bar{\mathbf{y}}$ from the objective function, using the fact that $\mathbf{A}\bar{\mathbf{y}} = \mathbf{b}$, and defining $\boldsymbol{\ell} \equiv \bar{\boldsymbol{\ell}} - \bar{\mathbf{y}}$ and $\mathbf{u} \equiv \bar{\mathbf{u}} - \bar{\mathbf{y}}$, this problem is equivalent to:

$$\text{Minimize } \{\mathbf{cx} : \mathbf{Ax} = \mathbf{0}, \boldsymbol{\ell} \le \mathbf{x} \le \mathbf{u}\},$$

where $-\infty < \ell_{ij} \le u_{ij} < \infty$. This problem is now in the form (11.1). Note that in this form, **x** can be interpreted as the modification in the given flow $\bar{\mathbf{y}}$ that is required to achieve an optimal solution (if one exists). Hence, Problem (11.1) can essentially be viewed as a *flow augmentation formulation* of a minimal–cost network flow program. We ask the reader to explore this concept further in Exercises 11.2 and 11.3.

The Dual of the Circulatory Network Flow Problem and Its Properties

If we associate a dual variable w_i with each node's flow conservation equation in Problem (11.1), a dual variable h_{ij} with the constraint $x_{ij} \leq u_{ij}$ (which is treated as $-x_{ij} \geq -u_{ij}$ for the purpose of taking the dual), and a dual variable v_{ij} with the constraint $x_{ij} \geq \ell_{ij}$, the dual of the out–of–kilter formulation for the minimal–cost network flow problem is given by:

$$\begin{array}{lll} \text{Maximize} & \sum_{i=1}^{m}\sum_{j=1}^{m} \ell_{ij}v_{ij} - \sum_{i=1}^{m}\sum_{j=1}^{m} u_{ij}h_{ij} & \\ \text{subject to} & w_i - w_j + v_{ij} - h_{ij} = c_{ij}, & i,j = 1,\ldots,m \\ & h_{ij}, v_{ij} \geq 0, & i,j = 1,\ldots,m \\ & w_i \quad \text{unrestricted}, & i = 1,\ldots,m, \end{array}$$

where the summations and the constraints are taken over existing arcs. The dual problem has a very interesting structure. Suppose that we select any set of w_i–values (we shall assume throughout the development that the w_i–values are integers). Then the dual constraint for arc (i, j) becomes

$$v_{ij} - h_{ij} = c_{ij} - w_i + w_j, \qquad h_{ij} \geq 0, \qquad v_{ij} \geq 0,$$

and is satisfied by letting

$$v_{ij} = \text{maximum } \{0, c_{ij} - w_i + w_j\}$$

$$h_{ij} = \text{maximum } \{0, -(c_{ij} - w_i + w_j)\}.$$

Thus, the dual problem always possesses a feasible solution given any set of w_i–values. In fact, the choices of v_{ij} and h_{ij} just given yield optimal values of v_{ij} and h_{ij} for a fixed set of w_i–values (why?).

The Complementary Slackness Conditions

The complementary slackness conditions for optimality of the out–of–kilter formulation are (review the Karush–Kuhn–Tucker optimality conditions) the following:

$$(x_{ij} - \ell_{ij})v_{ij} = 0, \qquad i,j = 1,\ldots, m \tag{11.2}$$

$$(u_{ij} - x_{ij})h_{ij} = 0, \qquad i,j = 1,\ldots, m. \tag{11.3}$$

Define $z_{ij} - c_{ij} \equiv w_i - w_j - c_{ij}$. Then by the definition of v_{ij} and h_{ij} we get

$$v_{ij} = \text{maximum } \{0, -(z_{ij} - c_{ij})\} \tag{11.4}$$

$$h_{ij} = \text{maximum } \{0,\ z_{ij} - c_{ij}\}. \tag{11.5}$$

Note that $z_{ij} - c_{ij}$ would be the familiar coefficient of x_{ij} in the objective function row of the lower–upper bounded simplex tableau if we had a basic solution to the primal problem. However, we need not have a basic solution here, and no such implication of a basis is being made here by this notation.

Given a set of w_i–values, we can compute $z_{ij} - c_{ij} = w_i - w_j - c_{ij}$. Noting Equations (11.4) and (11.5), the complementary slackness conditions (11.2) and (11.3) hold only if

$$z_{ij} - c_{ij} < 0 \Rightarrow v_{ij} > 0 \Rightarrow x_{ij} = \ell_{ij}, \qquad i, j = 1, \dots, m$$

$$z_{ij} - c_{ij} > 0 \Rightarrow h_{ij} > 0 \Rightarrow x_{ij} = u_{ij}, \qquad i, j = 1, \dots, m.$$

Conversely, if $x_{ij} = \ell_{ij}$ for all (i, j) such that $z_{ij} - c_{ij} < 0$, and if $x_{ij} = u_{ij}$ for all (i, j) such that $z_{ij} - c_{ij} > 0$, then defining v_{ij} and h_{ij} as in Equations (11.4) and (11.5), respectively, we have complementary slackness holding true. Hence, we obtain the following key result that embodies the optimality conditions of Problem (11.1), as well as the principal thrust of the out–of–kilter algorithm.

***Theorem* 11.1**

Let $\mathbf{x}$ be any conserving flow, and let $\mathbf{w} = (w_1, \dots, w_m)$ be any integer vector. Then $\mathbf{x}$ and $\mathbf{w}$ are, respectively, primal and dual optimal solutions to Problem (11.1) if and only if for all (i, j)

$z_{ij} - c_{ij} < 0$ implies $x_{ij} = \ell_{ij}$,

$z_{ij} - c_{ij} > 0$ implies $x_{ij} = u_{ij}$,

$z_{ij} - c_{ij} = 0$ implies $\ell_{ij} \le x_{ij} \le u_{ij}$,

where $(z_{ij} - c_{ij}) \equiv w_i - w_j - c_{ij}$ for all (i, j).

The problem then is to search over values of the w_i–variables and flow conserving x_{ij}–variables until the three conditions of Theorem 11.1 are satisfied.

Consider Figure 11.1a. Selecting a set of starting w_i–values, say, each $w_i = 0$, and a conserving flow, say, each $x_{ij} = 0$, we can check for optimality. Figure 11.1b displays the values for $z_{ij} - c_{ij}$, x_{ij}, and w_i for the network of Figure 11.1a. In Figure 11.1b we see that $z_{12} - c_{12} = -2$ and $x_{12} = 0$ $(= \ell_{12})$, and thus arc (1, 2) is said to be *in–kilter*, that is, *well.* On the other hand, $z_{23} - c_{23} = 3$ and $x_{23} = 0$ $(< u_{23})$, and thus arc (2, 3) is said to be *out–of–kilter*, that is, *unwell* or in *improper condition*. Hence, the name *out–of–kilter.*

To bring arc (2,3) into kilter we must either increase x_{23} or decrease $z_{23} - c_{23}$ by changing the w_i–values. This is exactly what the out–of–kilter algorithm attempts to do. During the primal phase of the out–of–kilter algorithm we shall be changing the x_{ij}–values in an attempt to bring arcs into kilter. During the dual phase we change the w_i–values in an attempt to reach an in–kilter state.

The Kilter States and Kilter Numbers for an Arc

The in–kilter and out–of–kilter states for each arc in a network are given in Figure 11.2. Note that an arc is in kilter if $\ell_{ij} \le x_{ij} \le u_{ij}$ and the conditions (of Theorem 11.1) hold true. As we change the flow on arc (i, j), the arc moves up and down a particular column in Figure 11.2 depending on whether x_{ij} is increased or is decreased. As we change the w_i–values the arc moves back and forth along a row. Figure 11.2b gives a graphical depiction of the kilter states of an arc. Each of the cells in the matrix in Figure 11.2a corresponds to a particular subregion in Figure 11.2b.

In order to assure that the algorithm will converge, we need some measure of the "distance" from optimality. If we can construct an algorithm that periodically (at finite intervals) reduces the distance from optimality by an integer, then the algorithm will eventually converge.

There are many different measures of distance for the out–of–kilter method. We present in Figure 11.3 one measure of distance that we call the *kilter number* K_{ij} for an arc (i, j). The kilter number is defined here to be the *minimal change of flow on the arc that is needed to bring it into kilter*. The kilter number of an arc is illustrated graphically in Figure 11.3b. Notice that since all terms involve absolute values, the kilter number for an arc is nonnegative. Also, notice that if the arc is in–kilter, the associated kilter number

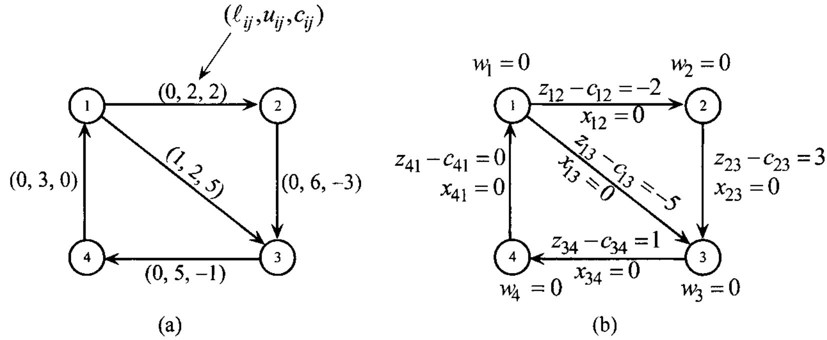
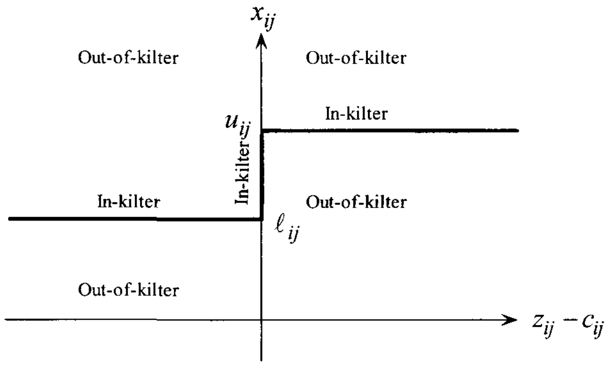


Figure 11.1. An example network: (a) The network. (b) Values of w_i, $z_{ij} - c_{ij}$, x_{ij}.

	$z_{ij}-c_{ij}<0$	$z_{ij}-c_{ij}=0$	$z_{ij}-c_{ij}>0$
$x_{ij}>u_{ij}$	Out-of-kilter	Out-of-kilter	Out-of-kilter
$x_{ij}=u_{ij}$	Out-of-kilter	In-kilter	In-kilter
$\ell_{ij}<x_{ij}<u_{ij}$	Out-of-kilter	In-kilter	Out-of-kilter
$x_{ij}=\ell_{ij}$	In-kilter	In-kilter	Out-of-kilter
$x_{ij}<\ell_{ij}$	Out-of-kilter	Out-of-kilter	Out-of-kilter

(a)



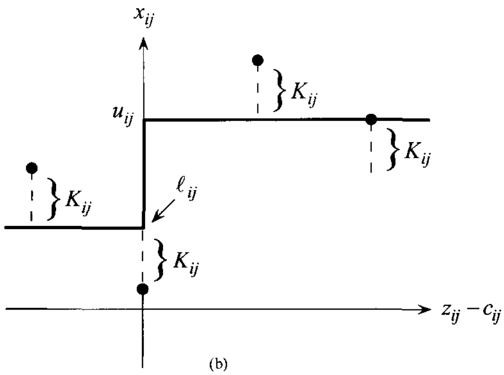
(b)

Figure 11.2. The possible kilter states for an arc.

is zero, and if the arc is out–of–kilter, the associated kilter number is strictly positive. Note that if $z_{ij}-c_{ij}<0$, then arc (i, j) is in–kilter only if the flow is equal to ℓ_{ij}, and hence the kilter number $|x_{ij}-\ell_{ij}|$ indicates how far the current flow x_{ij} is from the required value ℓ_{ij}. Similarly, if $z_{ij}-c_{ij}>0$, then the kilter number $|x_{ij}-u_{ij}|$ gives the distance from the required flow of u_{ij}. Finally, if $z_{ij}-c_{ij}=0$, then the arc is in–kilter if $\ell_{ij}\le x_{ij}\le u_{ij}$. In particular, if $x_{ij}>u_{ij}$, then the arc is brought in–kilter if the flow decreases by $|x_{ij}-u_{ij}|$, and if $x_{ij}<\ell_{ij}$, then the arc is brought in–kilter if the flow increases by $|x_{ij}-\ell_{ij}|$, and hence we obtain the entries in Figure 11.3, as shown under the column $z_{ij}-c_{ij}=0$.

	$z_{ij}-c_{ij}<0$	$z_{ij}-c_{ij}=0$	$z_{ij}-c_{ij}>0$
$x_{ij}>u_{ij}$	$\lvert x_{ij}-\ell_{ij}\rvert$	$\lvert x_{ij}-u_{ij}\rvert$	$\lvert x_{ij}-u_{ij}\rvert$
$x_{ij}=u_{ij}$	$\lvert x_{ij}-\ell_{ij}\rvert$	0	0
$\ell_{ij}<x_{ij}<u_{ij}$	$\lvert x_{ij}-\ell_{ij}\rvert$	0	$\lvert x_{ij}-u_{ij}\rvert$
$x_{ij}=\ell_{ij}$	0	0	$\lvert x_{ij}-u_{ij}\rvert$
$x_{ij}<\ell_{ij}$	$\lvert x_{ij}-\ell_{ij}\rvert$	$\lvert x_{ij}-\ell_{ij}\rvert$	$\lvert x_{ij}-u_{ij}\rvert$

(a)



(b)

Figure 11.3. The kilter numbers K_{ij}.

One method of assuring finite convergence of the out–of–kilter algorithm is to guarantee the following:

1. The kilter number of any arc never increases.
2. At finite intervals, the kilter number of some arc is reduced (by an integer).

This is exactly what we shall be able to achieve.

11.2 STRATEGY OF THE OUT–OF–KILTER ALGORITHM

As indicated before, the out–of–kilter algorithm may be generally viewed as a primal–dual type of algorithm. In this respect the generic steps of the algorithm are as follows:

1. Begin with a conserving flow, such as each $x_{ij} = 0$, and a feasible solution to the dual with **w** arbitrarily chosen, such as each $w_i = 0$, and with h_{ij} and v_{ij} as defined in Equations (11.4) and (11.5) for all arcs (i, j). Identify the kilter states and compute the kilter numbers.

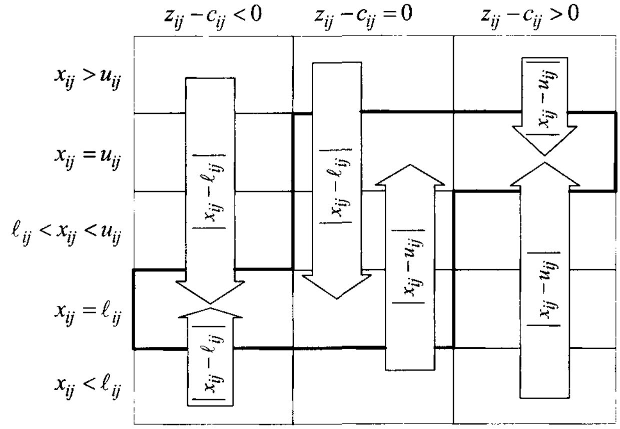
2. If the network has an out–of–kilter arc, conduct a primal phase of the algorithm. During this phase an out–of–kilter arc is selected and an attempt is made to construct a new conserving flow in such a way that the kilter number of no arc is worsened and that of the selected arc is improved.
3. When no such improving flow can be constructed during the primal phase, the algorithm constructs a new dual solution in such a way that no kilter number is worsened and Step 2 is repeated.
4. Iterating between Steps 2 and 3, the algorithm eventually constructs an optimal solution or determines that no feasible solution exists.

The Primal Phase: Flow Change

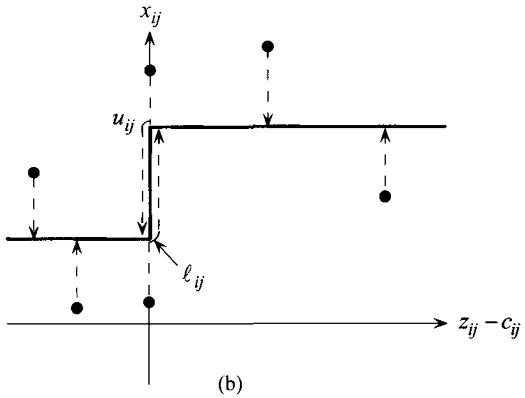
During the primal phase, the out–of–kilter algorithm attempts to decrease the kilter number of an out–of–kilter arc by changing the conserving flows in such a way that the kilter number on any other arc is not worsened. Examining Figure 11.3, we see that the flows must be changed in such a way that the corresponding kilter states move closer to the in–kilter states. For example, for the out–of–kilter state $x_{ij} > u_{ij}$ and $z_{ij} - c_{ij} < 0$, we can decrease x_{ij} by as much as $|x_{ij} - \ell_{ij}|$ before the arc comes into kilter. If we decrease x_{ij} beyond this, the arc will pass the in–kilter state (we do not want this to happen). Also, we do not permit any increase in this x_{ij} value. A similar analysis of the other kilter states produces the results in Figure 11.4a.

Several cells in Figure 11.4a deserve special attention. The out–of–kilter state $x_{ij} > u_{ij}$ and $z_{ij} - c_{ij} = 0$ indicates that the flow can be decreased by as much as $|x_{ij} - \ell_{ij}|$. Referring to Figure 11.3, we see that we really only need to decrease the particular x_{ij} by $|x_{ij} - u_{ij}|$, a smaller amount, to reach an in–kilter state. However, as can be seen in Figure 11.3, we may continue to decrease x_{ij} by an amount up to $|x_{ij} - \ell_{ij}|$ from its original value and the arc will still remain in–kilter. It might be indeed desirable to do this in order to aid other arcs in reaching in–kilter states. Also, an arc in the in–kilter state for which $\ell_{ij} < x_{ij} < u_{ij}$ and $z_{ij} - c_{ij} = 0$ may have its flow appropriately either increased or decreased, while still maintaining its in–kilter status. Figure 11.4b illustrates the permitted flow changes graphically.

Now that we have ascertained how much an individual flow on an arc may change, we must still determine what combination of flows we can change in order to maintain a *conserving flow*. If $\bar{\mathbf{x}}$ is the vector of (current) conserving flows, then the conservation of flow equality constraints in Problem (11.1) can be rewritten as $\mathbf{A}\bar{\mathbf{x}} = \mathbf{0}$, where $\mathbf{A}$ is the node–arc incidence matrix. If Δ is a vector of flow changes, then we must have



(a)



(b)

Figure 11.4. Permitted flow change directions and amounts.

$$\mathbf{A}(\bar{\mathbf{x}} + \Delta) = \mathbf{0} \qquad \text{or} \qquad \mathbf{A}\Delta = \mathbf{0}.$$

If $\mathbf{A}\Delta = \mathbf{0}$ for a nonzero Δ, then the columns of $\mathbf{A}$ corresponding to the nonzero components of Δ must be linearly dependent. Since $\mathbf{A}$ is a node–arc incidence matrix, then each column of $\mathbf{A}$ has exactly one $+1$ and one -1, and the nonzero components of Δ must correspond to a (not necessarily directed) cycle or a set of cycles (why?). Hence, flows must be changed along a cycle or a set of cycles in order to continue satisfying the conservation of flow equations.

Given an out–of–kilter arc, we need to construct a cycle containing that arc. This cycle must have the property that when assigned an orientation and when flow is added, no arc has its kilter number worsened. A convenient method for doing this is to construct a new network G' from the original network according to the information in Figure 11.4. First, every node of the original network is in the new network. Next, if an arc (i, j) is in the original network and the flow may be increased, then arc (i, j) becomes part of the new network with the appropriate permitted flow change being as indicated in Figure

11.4. Finally, if an arc (i, j) is in the original network and the flow can be decreased, then arc (j, i) becomes part of the new network with the permitted flow change being as indicated for arc (i, j) in Figure 11.4. Arcs in the original network having $\ell_{ij} < x_{ij} < u_{ij}$ and $z_{ij} - c_{ij} = 0$ will produce two arcs, (i, j) and (j, i), each with an appropriate permitted flow change in the new network. Arcs not permitted to change in flow are omitted entirely from G'.

Given the example indicated in Figure 11.1, a new network G' is constructed by the foregoing rules and is presented in Figure 11.5. To illustrate, consider arc (1, 3) in Figure 11.1. Note that $x_{13} < \ell_{13}$ and $z_{13} - c_{13} < 0$. From Figure 11.4 the flow on (1, 3) can increase to $\ell_{13} = 1$. This results in arc (1, 3) in Figure 11.5, with a permitted flow change of 1.

Once the new network G' is constructed and an out–of–kilter arc (p, q) in G' is selected, we look for a circuit (directed cycle) containing that arc in G'. This circuit in G' corresponds to a cycle in G. The flow in the cycle in G is changed according to the orientation provided by the circuit in G'. The amount of change is specified by the smallest permitted flow change of any arc that is a member of the circuit in G'. If no circuit containing the selected out–of–kilter arc exists in G', then we proceed to the dual phase of the algorithm.

We remark here that the construction of G' presented is only for pedagogical purposes. In essence, given an actual out–of–kilter arc (p, q) in G', we wish to determine if there exists a circuit in G' that involves this arc. Note that this arc may be (p, q) or (q, p) in G, depending on whether an increase or a decrease in the flow on this arc is required in G. Hence, in G', we need to find a (directed) path from q to p. This may be done by constructing a tree T that begins with the node q, and at each stage scans for a pair of nodes i and j with $i \in T$ and $j \notin T$ such that there is an arc (i, j) in G that can permit a flow increase or such that there is an arc (j, i) in G that can permit a flow decrease. Arcs of this type are called *label eligible arcs*. If such an arc exists, it is used to include node j in the tree T and the process repeats. If node p gets included in T at any step, then we will have found a circuit in G' oriented along (p, q) that permits a positive flow change. This situation is called a *breakthrough*. If no

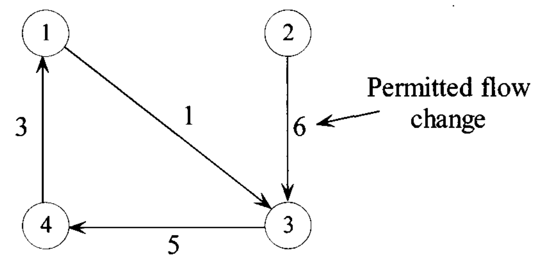


Figure 11.5. The modified network G' for Figure 11. 1.

additional nodes can be added to T at some step and $p \notin T$, then there does not exist any circuit in G' involving (p, q) (why?). This situation is called a *nonbreakthrough*. In this event, we proceed to the dual phase with the nodes in T forming a set X and the remaining nodes in $\overline{X}$. A labeling scheme that constructs such a tree T is described in Section 11.5. For now, let us continue to work with the network G'.

As an illustration of the primal phase, consider the modified network G' of Figure 11.5. We select an out–of–kilter arc, say, (1, 3). From Figure 11.5, we see, that a circuit exists in G' containing arc (1, 3), namely, $C = \{(1, 3), (3, 4), (4, 1)\}$. Hence, we can change the flow around the associated cycle in G, increasing flows on arcs having the orientation of the circuit in G' and decreasing flows on arcs against the orientation of the circuit in G', and obtain an improved (in the kilter number sense) solution. The amount of permitted change in flow is Δ = minimum $\{1, 5, 3) = 1$. The new solution and associated modified network is given in Figure 11.6a. Arcs (2, 3) and (3, 4) are still out–of–kilter in G. Selecting one of the associated arcs in G' (see Figure 11.6b), say, (2, 3), we attempt to find a circuit in G' containing the selected arc. Because no such circuit exists, we must pass to the dual phase of the out–of–kilter algorithm. The tree T in this case consists of nodes 3, 4, and 1, and arcs (3, 4) and (4, 1) from G'.

It is convenient (but not necessary—see Section 11.6) for the various proofs of convergence to work on the same out–of–kilter arc (p, q) until it comes in–kilter. We shall assume throughout our discussion of the algorithm that this is done.

The Dual Phase: Dual Variable Change

When it is no longer possible to construct a circuit in G' containing a specific out–of–kilter arc, then we must change the $(z_{ij} - c_{ij})$–values so that no kilter number is worsened and either the out–of–kilter arc is brought into kilter or some new arcs are introduced into G' that would eventually allow us to find a circuit containing the out–of–kilter arc under consideration.

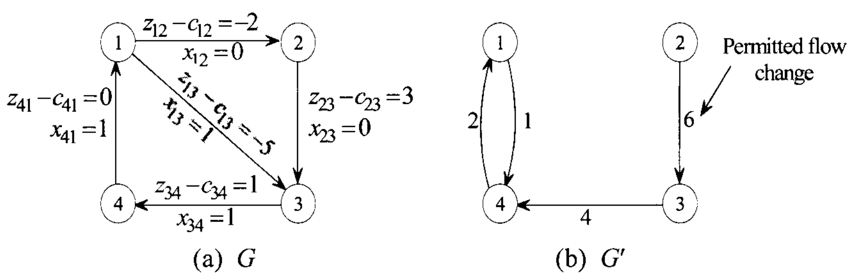


Figure 11.6. The new solution for the network of Figure 11. 1.

Since $z_{ij} - c_{ij} = w_i - w_j - c_{ij}$, we must change the w_i–values in order to change the $(z_{ij} - c_{ij})$–values. Let (p, q) be an out–of–kilter arc in G', and let X be the set of nodes in G' that can be reached from node q along some paths in

G'. Hence, X is the set of nodes in the tree T constructed earlier. Let $\overline{X} \equiv \mathcal{N} - X$. where $\mathcal{N} = \{1,\ldots,m\}$. Note that neither X nor $\overline{X}$ is empty when we pass to the dual phase, since $q \in X$ and $p \in \overline{X}$. For $(p, q) = (2, 3)$ in Figure 11.6 we have $X = \{3, 4, 1\}$ and $\overline{X} = \{2\}$. In Figure 11.7, we illustrate the sets X and $\overline{X}$.

We would like to change the w_i–values so that no kilter number is worsened and the set X gets larger periodically. If at least one node comes into X at finite intervals, then eventually p will come into X and a circuit will be created in G'. We have implicitly assumed that X will not get smaller. To ensure this, we should change the w_i–values so that all arcs having both ends in X are retained in the modified graph.

Consider $z_{ij} - c_{ij} = w_i - w_j - c_{ij}$. If w_i and w_j are changed by the same amount, then $z_{ij} - c_{ij}$ remains unchanged. Thus, we can ensure that the set X will contain at least all of the same nodes after a dual variable change if we change all of the w_i–values in X by the same amount θ. Suppose that we leave the w_i–values in $\overline{X}$ unchanged. Then the only arcs that will be affected will be arcs going from X to $\overline{X}$ and from $\overline{X}$ to X. Specifically, if $\theta > 0$ and we change the w_i–values according to

$$w_i' = \begin{cases} w_i + \theta, & i \in X \\ w_i, & i \in \overline{X} \end{cases}$$

then the revised ($z_{ij} - c_{ij}$)–values are given by

$$(z_{ij} - c_{ij})' = z_{ij} - c_{ij} \quad \text{if} \quad i \in X, j \in X \quad \text{or} \quad i \in \overline{X}, j \in \overline{X}.$$

However, if $i \in X$ and $j \in \overline{X}$, we get

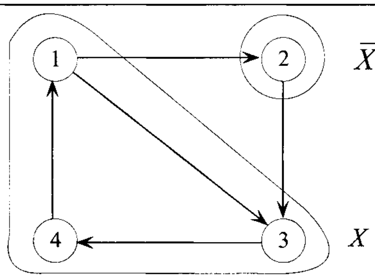


Figure 11.7. X and $\overline{X}$ in G for $(p, q) = (2, 3)$ in Figure 11.6.

$$\begin{aligned}(z_{ij} - c_{ij})' &= (w_i + \theta) - w_j - c_{ij} \\ &= (z_{ij} - c_{ij}) + \theta.\end{aligned}$$

Also, for $i \in \overline{X}$ and $j \in X$ we get

$$\begin{aligned}(z_{ij}-c_{ij})' &= w_i-(w_j+\theta)-c_{ij}\\ &= (z_{ij}-c_{ij})-\theta.\end{aligned}$$

Thus, arcs from X to $\bar{X}$ will have their $(z_{ij}-c_{ij})$–values increased by θ, and those from $\bar{X}$ to X will have their $(z_{ij}-c_{ij})$–values decreased by θ. We must determine θ so that the kilter number of no arc is worsened and the kilter state of some arc is changed. First, we must identify the arcs that can be in the set $(X, \bar{X})$ and in the set $(\bar{X}, X)$. (The notation (X, Y) represents the set $S = \{(x, y)\colon x \in X,\ y \in Y\}$. The set of arcs in G going between X and $\bar{X}$, in either direction, are said to constitute the *cut* with respect to X and $\bar{X}$, and will be denoted by $[X, \bar{X}]$. Hence, $[X, \bar{X}] = (X, \bar{X}) \cup (\bar{X}, X)$.)

Examining Figure 11.4, we see that the set $(X, \bar{X})$ cannot contain an arc associated with the kilter state $x_{ij} < \ell_{ij}$ and $z_{ij}-c_{ij} < 0$, since such an arc (i, j) in G would become an arc in G' with the result that if i can be reached (along a path) from q, then j can be reached from q, and thus $j \in X$ (a contradiction). Examining the remaining kilter states, we find that the only candidates for membership in $(X, \bar{X})$ are those identified in Figure 11.8. Recall that arcs from X to $\bar{X}$ in G have their $(z_{ij}-c_{ij})$–values increased. Thus, these arcs change kilter states in a left–to–right fashion as indicated in Figure 11.8a. Examining an arc from X to $\bar{X}$ in G that has $x_{ij} > u_{ij}$ and $z_{ij}-c_{ij} < 0$, we see from Figure 11.3 that as θ increases, K_{ij} decreases from $K_{ij} = |x_{ij}-\ell_{ij}|$ to $K_{ij} = |x_{ij}-u_{ij}|$ and thereafter remains constant. Thus, for such an arc, we can increase θ as much as we like and the arc's kilter number will never increase. Hence, such an arc gives rise to an upper limit on θ of ∞, as indicated in Figure 11.8a. Any arc from X to $\bar{X}$ in G that has $x_{ij} = u_{ij}$ and $z_{ij}-c_{ij} < 0$ will have its kilter number first decrease and then remain unchanged as θ increases (why?). Thus, again ∞ is an upper limit on the permitted change in θ for such an arc to ensure that no kilter number will worsen. However, examining an arc from X to $\bar{X}$ in G that has $\ell_{ij} < x_{ij} < u_{ij}$ and $z_{ij}-c_{ij} < 0$, we see that the associated kilter number K_{ij} first decreases (to zero), then starts to increase. In order to eliminate the potential increase in K_{ij} for the arc we must place a limit of $|z_{ij}-c_{ij}|$ on θ. Similarly, we must place a limit of $|z_{ij}-c_{ij}|$ on θ for arcs having $x_{ij} = \ell_{ij}$ and $z_{ij}-c_{ij} < 0$. This analysis justifies the entries in Figure 11.8. Each of the possible cases for arcs in $(X, \bar{X})$ is graphically portrayed in Figure 11.8b.

A similar analysis of arcs from $\bar{X}$ to X in G gives rise to the information in Figure 11.9.

Insofar as worsening of kilter numbers is concerned, Figures 11.8 and 11.9 indicate that we need only compute θ based on arcs from X to $\bar{X}$ having $x_{ij} < u_{ij}$ and arcs from $\bar{X}$ to X with $x_{ij} > \ell_{ij}$. However, if we proceed to define a method of computing θ based only on these considerations, difficulties would arise in interpreting the meaning of the value $\theta = \infty$. Matters are greatly simplified if, instead of strict inequalities on flow (that is, $x_{ij} < u_{ij}$ and $x_{ij} > \ell_{ij}$), we admit weak inequalities on flow (that is, $x_{ij} \le u_{ij}$ and $x_{ij} \ge \ell_{ij}$). The reason for this deviation from intuition will become apparent when we proceed to establish convergence of the algorithm.

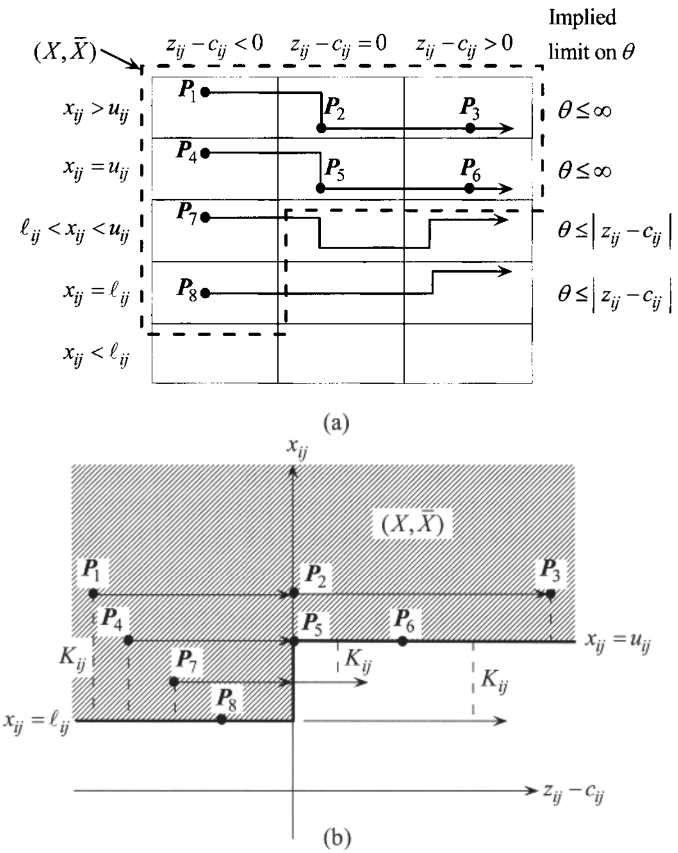


Figure 11.8. Possible kilter states for arcs from X to $\bar{X}$ in G and limits on θ.

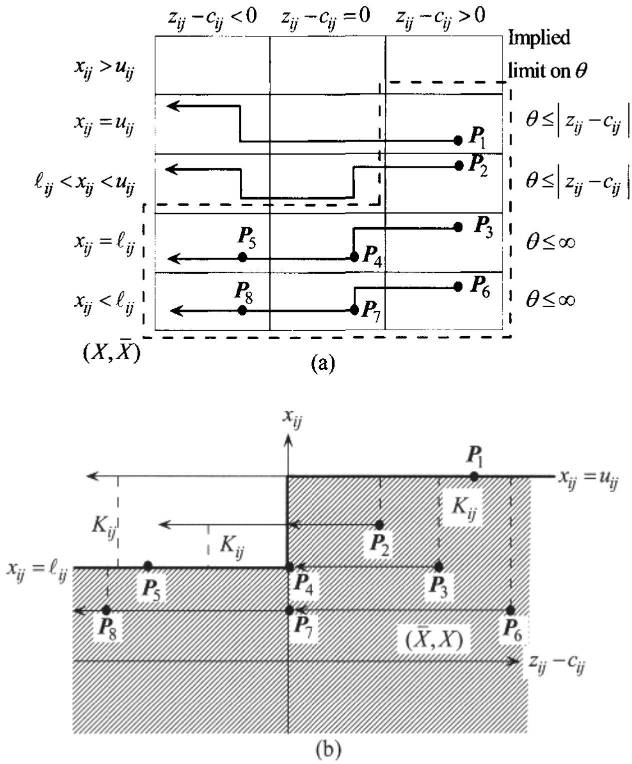


Figure 11.9. Possible cases for arcs from $\bar{X}$ to X in G and limits on θ.

The previous discussion concerning limits on θ based on kilter number considerations and on (yet to be established) convergence properties leads to the following formal procedure for computing θ.

In G define S_1 and S_2 by

$$S_1 \equiv \{(i,j): i \in X, j \in \bar{X}, z_{ij}-c_{ij}<0, x_{ij} \le u_{ij}\}$$

and

$$S_2 \equiv \{(i,j): i \in \bar{X}, j \in X, z_{ij}-c_{ij}>0, x_{ij} \ge \ell_{ij}\}.$$

Let

$$\theta_1 = \underset{(i,j)\in S_1}{\text{minimum}} \{|z_{ij}-c_{ij}|\}$$

$$\theta_2 = \underset{(i,j)\in S_2}{\text{minimum}} \{|z_{ij}-c_{ij}|\}$$

$$\theta = \text{minimum } \{\theta_1, \theta_2\},$$

where $\theta_i \equiv \infty$ if S_i is empty. Thus, θ is either a positive integer or ∞. These two possibilities are briefly discussed.

Case 1: $0 < \theta < \infty$.

In this case, we make the appropriate changes in w_i (that is, $w_i' = w_i + \theta$ if $i \in X$ and $w_i' = w_i$ if $i \in \overline{X}$) and pass to the primal phase of the algorithm.

Case 2: $\theta = \infty$.

In this case, the primal problem has no feasible solution. (We shall show this shortly.)

This completes the specification of the dual phase of the out–of–kilter algorithm and provides the foundation of the overall out–of–kilter algorithm.

As an illustration, consider the example of Figure 11.1 with the current solution specified by Figures 11.6 and 11.7. Here,

$$S_1 = \{(1,2)\}, \quad \theta_1 = |-2| = 2$$
$$S_2 = \{(2,3)\}, \quad \theta_2 = |3| = 3$$
$$\theta = \text{minimum } \{2, 3\} = 2.$$

This gives rise to the following change in dual variables:

$$\begin{aligned} w_1' &= w_1 + \theta = 2 \\ w_2' &= w_2 = 0 \\ w_3' &= w_3 + \theta = 2 \\ w_4' &= w_4 + \theta = 2\,. \end{aligned}$$

The x_{ij}–values and the new $(z_{ij} - c_{ij})$–values are given in Figure 11.10a. Passing to the primal phase of the out–of–kilter algorithm, we see that G' in Figure 11.10b contains a circuit involving arc (2, 3), and so we may change the flows. The remaining iterations are not shown.

There is really no need to work directly with the dual variables themselves since we may transform the $(z_{ij} - c_{ij})$–values directly as follows:

$$(z_{ij} - c_{ij})' = \begin{cases} (z_{ij} - c_{ij}) & \text{if} \quad i \in X, j \in X \text{ or } i \in \overline{X}, j \in \overline{X} \\ (z_{ij} - c_{ij}) + \theta & \text{if} \quad i \in X, j \in \overline{X} \\ (z_{ij} - c_{ij}) - \theta & \text{if} \quad i \in \overline{X}, j \in X. \end{cases}$$

In Exercise 11.18 we ask the reader to show how the dual variables can be recovered from these $(z_{ij} - c_{ij})$–values anytime we need them. Note that the $(z_{ij} - c_{ij})$–values are integral (why?).

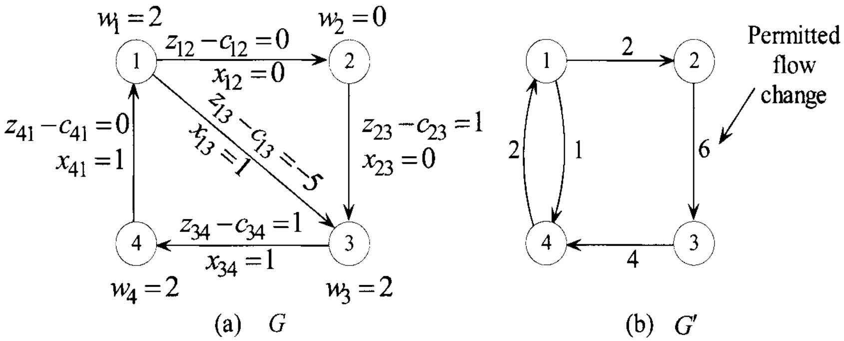


Figure 11.10. The new solution obtained from Figure 11.6 after the first dual variable change.

As an example of infeasibility, consider the example network of Figure 11.11a. Selecting a set of x_{ij}– and w_i–values, we find in Figure 11.11b that arc (2, 1) is out–of–kilter. Setting up G' in Figure 11.11c, we find no circuit containing the arc (2, 1). In this case, $X = \{1\}$ and $\bar{X} = \{2\}$. Here, $S_1 = \varnothing$ (the empty set) and $S_2 = \varnothing$, and thus $\theta = \infty$. It is clear by examining u_{12} and ℓ_{21} that no feasible solution exists.

Infeasibility of the Problem When $\theta = \infty$

Suppose that during some application of the dual phase of the out–of–kilter algorithm, we reach the case where $\theta = \infty$. When this occurs, we must have $S_1 = S_2 = \varnothing$. Since $S_1 = \varnothing$, then by reviewing the definition of S_1, we conclude that $i \in X$ and $j \in \bar{X}$ imply one of the following cases:

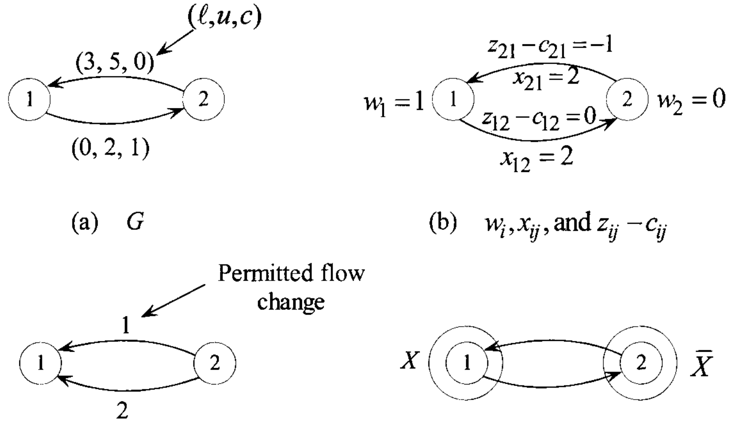
1. $z_{ij} - c_{ij} < 0$ and $x_{ij} > u_{ij}$;
2. $z_{ij} - c_{ij} = 0$;
3. $z_{ij} - c_{ij} > 0$.

From Figure 11.8 and since $i \in X$ and $j \in \bar{X}$, possibility (2) or (3) can hold only if $x_{ij} \geq u_{ij}$. Hence, $S_1 = \varnothing$ holds true only if $x_{ij} \geq u_{ij}$ for $i \in X$ and $j \in \bar{X}$. Similarly, $S_2 = \varnothing$ holds true only if $i \in \bar{X}$ and $j \in X$ implies that $x_{ij} \leq \ell_{ij}$. Hence, $S_1 = S_2 = \varnothing$ implies

$$x_{ij} \geq u_{ij} \quad \text{if} \quad i \in X, j \in \bar{X} \tag{11.6}$$

and

$$x_{ij} \leq \ell_{ij} \quad \text{if} \quad i \in \bar{X}, j \in X. \tag{11.7}$$



(a) G (b) w_i, x_{ij}, and $z_{ij}-c_{ij}$

(c) G' (d) X and $\bar{X}$

Figure 11.11. An example of an infeasible network.

In particular, consider the out–of–kilter arc (p, q) in G'. If (p, q) is in G, then by Equation (11.7) we have $x_{pq} \le \ell_{pq}$. Suppose that $x_{pq} = \ell_{pq}$. Since (p, q) is out–of–kilter, then $z_{pq} - c_{pq} > 0$, violating the assumption that $S_2 = \varnothing$. Thus, $x_{pq} < \ell_{pq}$. If, on the other hand, (q, p) is in G, then by a similar argument, we may show that $x_{qp} > u_{qp}$. Thus, at least one of the inequalities (11.6) or (11.7) is strict. Summing these two inequalities, we get

$$\sum_{\substack{i\in X\\ j\in \bar{X}}} x_{ij} - \sum_{\substack{i\in \bar{X}\\ j\in X}} x_{ij} > \sum_{\substack{i\in X\\ j\in \bar{X}}} u_{ij} - \sum_{\substack{i\in \bar{X}\\ j\in X}} \ell_{ij}. \tag{11.8}$$

Since the current flow given by the x_{ij}–values is conserving, then the equality constraints in Problem (11.1) hold true. Noting that the node set consists of $X \cup \bar{X}$ and that $X \cap \bar{X} = \varnothing$, these conservation of flow constraints can be written as

$$\sum_{j\in X} x_{ij} + \sum_{j\in \bar{X}} x_{ij} - \sum_{j\in X} x_{ji} - \sum_{j\in \bar{X}} x_{ji} = 0, \qquad i = 1,\ldots,m.$$

Summing these equations over $i \in X$, we get

$$\sum_{\substack{i\in X\\ j\in X}} x_{ij} + \sum_{\substack{i\in X\\ j\in \bar{X}}} x_{ij} - \sum_{\substack{j\in X\\ i\in X}} x_{ji} - \sum_{\substack{i\in X\\ j\in \bar{X}}} x_{ji} = 0.$$

Noting that

$$\sum_{\substack{i\in X\\ j\in X}} x_{ij} = \sum_{\substack{j\in X\\ i\in X}} x_{ji},$$

and that

$$\sum_{\substack{i\in X\\ j\in \bar{X}}} x_{ji} = \sum_{\substack{i\in \bar{X}\\ j\in X}} x_{ij},$$

the foregoing equation reduces to

$$\sum_{\substack{i\in X\\ j\in \bar{X}}} x_{ij} - \sum_{\substack{i\in \bar{X}\\ j\in X}} x_{ij} = 0. \tag{11.9}$$

Substituting in Equation (11.8), we get

$$0 > \sum_{\substack{i\in X\\ j\in \bar{X}}} u_{ij} - \sum_{\substack{i\in \bar{X}\\ j\in X}} \ell_{ij}. \tag{11.10}$$

Suppose by contradiction that there is a feasible flow represented by $\hat{x}_{ij}$ for $i, j = 1,\ldots,m$. Therefore, $u_{ij} \geq \hat{x}_{ij}$ and $-\ell_{ij} \geq -\hat{x}_{ij}$, and so, Equation (11.10) gives

$$0 > \sum_{\substack{i\in X\\ j\in \bar{X}}} u_{ij} - \sum_{\substack{i\in \bar{X}\\ j\in X}} \ell_{ij} \geq \sum_{\substack{i\in X\\ j\in \bar{X}}} \hat{x}_{ij} - \sum_{\substack{i\in \bar{X}\\ j\in X}} \hat{x}_{ij}. \tag{11.11}$$

But since the $\hat{x}_{ij}$–values represent a feasible flow, they must be conserving. In a fashion similar to Equation (11.9), it is clear that the right–hand–side of the inequality in (11.11) is equal to zero. Therefore, Equation (11.11) implies that 0 > 0, which is impossible. This contradiction shows that if $\theta = \infty$, there could be no feasible flow.

Note that if we had defined S_1 and S_2 by strict inequalities on x_{ij} (namely, $x_{ij} < u_{ij}$ and $x_{ij} > \ell_{ij}$, respectively), we could not have produced the strict inequality needed in Equation (11.8).

Convergence of the Out–of–Kilter Algorithm

For the purpose of the following finite convergence argument we make the assumption that the vectors $\boldsymbol{\ell}$, $\mathbf{u}$, and $\mathbf{c}$ are integer–valued.

In developing a finite convergence argument for the out–of–kilter algorithm, there are several properties of the algorithm that should be noted. First, every time a circuit is constructed in G' containing an out–of–kilter arc, the kilter number of that arc and of the total network is reduced by an integer (why?). We can construct only a finite number of circuits containing out–of–kilter arcs before an optimal solution is obtained (why?). Second, after each dual variable change, the kilter state of each arc in G that has both ends in X remains unchanged. Hence, if (p, q) is not in kilter, then after a dual variable change, each node in X before the change is in X after the change. Two possibilities exist. One possibility is that a new node k may be brought into X by virtue of an arc being added in G' from some node in X to node k. Each time this occurs the set X grows by at least one node. This can occur at most a finite number of times before node p becomes a member of X and a circuit is created containing (p, q). Thus, if the algorithm is not finite, it must be the case that an infinite number of dual variable changes take place without the set X increasing or θ equalling ∞. We shall show that this cannot occur.

Suppose that after a dual variable change, no new node becomes a member of X; that is, X does not increase. Then, upon passing to the next dual phase, we have the same sets X and $\bar{X}$ and the same x_{ij}–values. In addition, each arc from X to $\bar{X}$ has had its $z_{ij} - c_{ij}$ increased and each arc from $\bar{X}$ to X has had its $z_{ij} - c_{ij}$ decreased. Thus, after the dual variable change, the new sets S_1' and S_2' satisfy

$$S_1' \subseteq S_1 \quad \text{and} \quad S_2' \subseteq S_2$$

(why?). Furthermore, by the choice of the (finite) value of θ, at least one arc has been dropped from either S_1 or S_2. Thus, at least one of the foregoing inclusions is proper. Now, S_1 and S_2 may decrease at most a finite number of times before $S_1 \cup S_2 = \varnothing$ and $\theta = \infty$ occurs, in which case the algorithm stops.

This completes a finiteness argument for the out–of–kilter algorithm. We now summarize the algorithm and present an example.

11.3 SUMMARY OF THE OUT–OF–KILTER ALGORITHM

The complete algorithm consists of three phases: the initialization phase, the primal phase, and the dual phase.

Initialization Phase

Begin with a conserving (integer) flow, say, each $x_{ij} = 0$, and an initial set of (integral) dual variables, say, each $w_i = 0$. Compute $z_{ij} - c_{ij} = w_i - w_j - c_{ij}$.

Primal Phase

Determine the kilter state and the kilter number for each arc. If all arcs are in kilter, stop; an optimal solution has been obtained. Otherwise, select or continue with a previously selected out–of–kilter arc. From the network G construct a new network G' according to Figure 11.4. For each arc (i, j) in G that is in a kilter state that permits a flow increase, place an arc (i, j) in G' with a permitted flow increase, as indicated in Figure 11.4. For any arc (i, j) in G that is in a kilter state that permits a flow decrease, place an arc (j, i) in G' with the permitted flow as indicated in Figure 11.4. For those arcs in G that are members of states that permit no flow change, place no arc in G'. In G', attempt to construct a circuit containing the selected out–of–kilter arc (p, q). If such a circuit is available, we have a breakthrough. Determine a flow change Δ equal to the minimum of the permitted flow changes on arcs of the circuit. Change the flow on each arc of the associated cycle in G by the amount Δ using the orientation specified by the circuit as the direction of increase. In particular, let $x_{ij}' = x_{ij} + \Delta$ if (i, j) was a member of the circuit in G'; let $x_{ij}' = x_{ij} - \Delta$ if (j, i) was a member of the circuit in G'; let $x_{ij}' = x_{ij}$ otherwise. Repeat the primal phase. If no circuit containing arc (p, q) is available in G', we have a nonbreakthrough, and we pass to the dual phase. (Note that the construction of the tree T described in Section 11.2 can be used to detect breakthroughs or a nonbreakthrough, in lieu of actually constructing the graph G'.)

Dual Phase

Determine the set of nodes X that can be reached from node q along a path in G'. (This is available via the tree T constructed as in Section 11.2.) Let $\bar{X} = \mathcal{N} - X$. In G, define S_1 and S_2 by

$$S_1 = \{(i, j) : i \in X, j \in \bar{X}, z_{ij} - c_{ij} < 0, x_{ij} \le u_{ij}\}$$

$$S_2 = \{(i, j) : i \in \bar{X}, j \in X, z_{ij} - c_{ij} > 0, x_{ij} \ge \ell_{ij}\}.$$

Let

$$\theta = \underset{(i,j)\in S_1 \cup S_2}{\text{minimum}} \{\left|z_{ij} - c_{ij}\right|, \infty\}.$$

If $\theta = \infty$, stop; no feasible solution exists. Otherwise, change the w_i–values and the corresponding $(z_{ij} - c_{ij})$–values according to:

$$w_i' = \begin{cases} w_i + \theta & \text{if} \quad i \in X \\ w_i & \text{if} \quad i \in \bar{X} \end{cases}$$

$$(z_{ij} - c_{ij})' = \begin{cases} (z_{ij} - c_{ij}) & \text{if} \quad (i, j) \in (X, X) \cup (\bar{X}, \bar{X}) \\ (z_{ij} - c_{ij}) + \theta & \text{if} \quad (i, j) \in (X, \bar{X}) \\ (z_{ij} - c_{ij}) - \theta & \text{if} \quad (i, j) \in (\bar{X}, X) \end{cases}$$

and pass to the primal phase.

11.4 AN EXAMPLE OF THE OUT–OF–KILTER ALGORITHM

Consider the network given in Figure 11.12. Initializing the out–of–kilter algorithm with each $x_{ij} = 0$ and each $w_i = 0$, we get the sequence of primal and dual phases given in Figure 11.13.

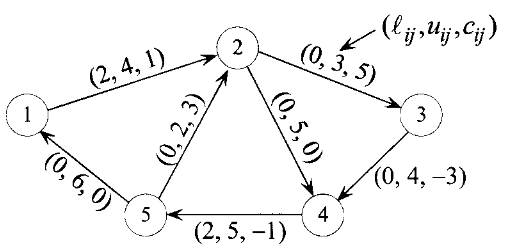
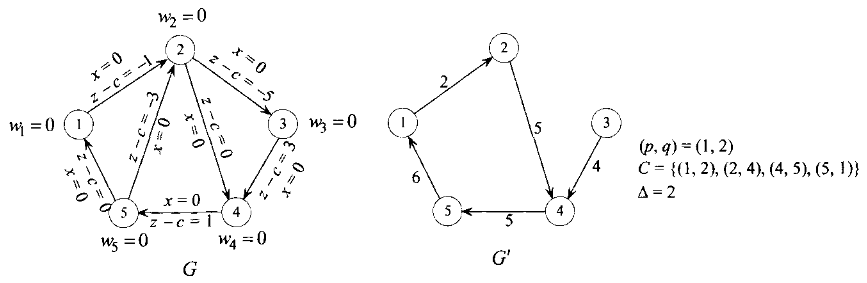
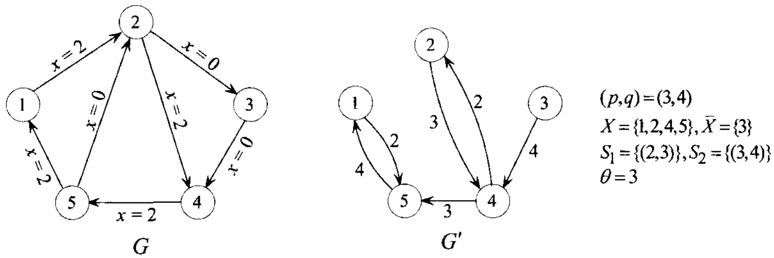


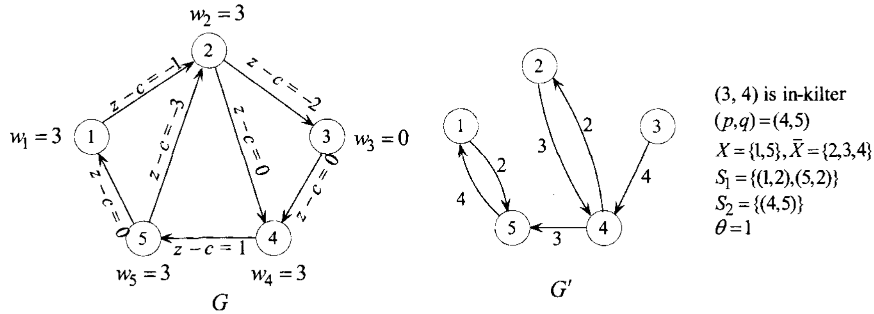
Figure 11.12. An example network.



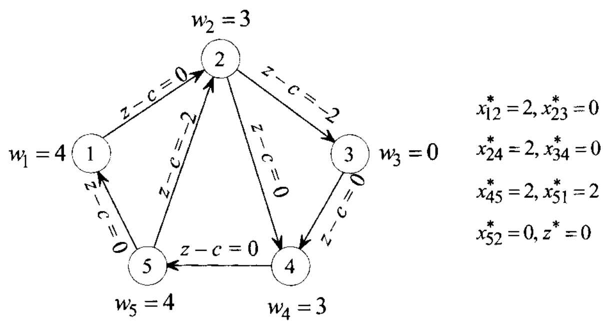
(a) Breakthrough in the First Primal Phase



(b) Nonbreakthrough and the First Dual Phase



(c) Nonbreakthrough and the Second Dual Phase



(d) An Optimal Solution

Figure 11.13. The out–of–kilter method solution for Figure 11.12.

11.5 A LABELING PROCEDURE FOR THE OUT–OF–KILTER ALGORITHM

Either for hand or computer calculations there are simple and convenient ways to maintain the information required to solve a minimal–cost flow problem by the out–of–kilter algorithm. Suppose that we associate with each node j a label $L(j) = (\pm i,\ \Delta_j)$. A label $(i,\ \Delta_j)$ indicates that the flow on arc (i, j) could be increased by an amount Δ_j without worsening the kilter number of any arc. A label $(-i,\ \Delta_j)$ indicates that the flow on arc $(j,\ i)$ could be decreased by an amount Δ_j without worsening the kilter number of any arc. Note that Δ_j represents the current estimate of the amount of flow change that can take place along some cycle containing an out–of–kilter arc and either arc (i, j) or $(j,\ i)$ in such a way that the kilter number of no arc is increased. The labeling algorithm becomes as follows.

INITIALIZATION STEP

Select a conserving flow, for example, each $x_{ij} = 0$, and a set of dual variables, such as each $w_i = 0$.

Main Step

1. If all arcs are in kilter according to Figure 11.2, stop; an optimal solution is obtained. Otherwise, select (or continue with a previously selected) out–of–kilter arc, say $(s,\ t)$. Erase all labels. If $(s,\ t)$ is in one of the states where a flow increase, Δ_{st}, is required according to Figure 11.4, then set $q = t, p = s$, and $L(q) = (+p, \Delta_{st})$. Otherwise, if $(s,\ t)$ is in one of the states where a flow decrease, Δ_{st} , is required according to Figure 11.4, then set $q = s$, $p = t$, and $L(q) = (-p, \Delta_{st})$.
2. If node i has a label, node j has no label, and flow may be increased by an amount Δ_{ij} along arc (i, j) according to Figure 11.4, then assign node j the label $L(j) = (+\ i, \Delta_j)$ where Δ_j = minimum $\{\Delta_i, \Delta_{ij}\}$. If node i has a label, node j has no label, and flow may be decreased by an amount Δ_{ji} along arc $(j,\ i)$ according to Figure 11.4, then give node j the label $L(j) = (-i, \Delta_j)$ where Δ_j = minimum (Δ_i, Δ_{ji}). Repeat Step 2 until either node p is labeled or until no more nodes can be labeled. If node p is labeled, go to Step 3 (a *breakthrough* has occurred); otherwise, go to Step 4 (a *nonbreakthrough* has occurred and the labeled nodes are in the tree T).

3. Let $\Delta = \Delta_p$. Change flow along the identified cycle as follows. Begin at node p. If the first entry in $L(p)$ is $+k$, then add Δ to x_{kp}. Otherwise, if the first entry in $L(p)$ is $-k$, then subtract Δ from x_{pk}. Backtrack to node k and repeat the process until node p is reached again in the backtracking process. Go to Step 1.
4. Let X be the set of labeled nodes and let $\overline{X} = \mathcal{N} - X$. Define $S_1 = \{(i,j)$: $i \in X,\ j \in \overline{X},\ z_{ij} - c_{ij} < 0,\ x_{ij} \le u_{ij}\}$ and $S_2 = \{(i,j)$: $i \in \overline{X},\ j \in X,\ z_{ij} - c_{ij} > 0,\ x_{ij} \ge \ell_{ij}\}$. Let $\theta = \text{minimum}\{|z_{ij} - c_{ij}|,\ \infty:\ (i,j) \in S_1 \cup S_2\}$. If $\theta = \infty$, stop; no feasible solution exists. Otherwise, let

$$w_i' = \begin{cases} w_i + \theta & \text{if} \quad i \in X \\ w_i & \text{if} \quad i \in \overline{X} \end{cases}$$

and

$$(z_{ij} - c_{ij})' = \begin{cases} (z_{ij} - c_{ij}) & \text{if} \quad (i,j) \in (X,X) \cup (\overline{X},\overline{X}) \\ (z_{ij} - c_{ij}) + \theta & \text{if} \quad (i,j) \in (X,\overline{X}) \\ (z_{ij} - c_{ij}) - \theta & \text{if} \quad (i,j) \in (\overline{X},X) \end{cases}$$

and return to Step 1.

An Example of the Labeling Algorithm

We shall illustrate the labeling method for the out–of–kilter algorithm by performing the first two iterations represented in Figure 11.13a and b. From Figure 11.13a we find that arc (1, 2) is an out–of–kilter arc whose flow must be increased.

The sequence of operations of the labeling algorithm are as follows:

1. $(s, t) = (1, 2), q = 2, p = 1, L(2) = (+\,1, 2)$.
2. $L(4) = (+2, 2)$.
3. $L(5) = (+4, 2)$.
4. $L(1) = (+5, 2)$.
5. Breakthrough: $\Delta = 2$.
6. $L_1(1) = +5 \Rightarrow x_{51}' = x_{51} + \Delta = 2$.
7. $L_1(5) = +4 \Rightarrow x_{45}' = x_{45} + \Delta = 2$.
8. $L_1(4) = +2 \Rightarrow x_{24}' = x_{24} + \Delta = 2$.
9. $L_1(2) = +1 \Rightarrow x_{12}' = x_{12} + \Delta = 2$.
10. Erase all labels, $(s, t) = (3, 4), q = 4, p = 3, L(4) = (+\,3, 4)$.
11. $L(5) = (+4, 3)$.
12. $L(1) = (+5, 3)$.
13. $L(2) = (-4, 2)$.
14. Nonbreakthrough: $X = \{1, 2, 4, 5\}$, $\overline{X} = \{3\}$, $\theta = 3$.
15. $w_1 = w_2 = w_4 = w_5 = 3, w_3 = 0$.

Since arc (3, 4) is now in–kilter, we select another out–of–kilter arc, erase all labels, and continue.

11.6 INSIGHT INTO CHANGES IN PRIMAL AND DUAL FUNCTION VALUES

It is instructive to see how the primal (penalty) and dual objective function values change in the primal and dual phases of the out–of–kilter algorithm. Not only does this provide an alternative convergence argument in which we need not necessarily work on the same out–of–kilter arc until it comes in kilter, but it also provides an insight into other acceptable ways of modifying primal and dual solutions.

First, consider the primal phase. Define for each arc (i, j) in G a function $P_{ij}(x_{ij})$ that measures the infeasibility of a conserving flow $\mathbf{x}$ as follows:

$$P_{ij}(x_{ij}) = \text{maximum}\{0, \ell_{ij} - x_{ij}\} + \text{maximum}\{0, x_{ij} - u_{ij}\}.$$

Note that $P_{ij}(x_{ij}) = 0$ if and only if $\ell_{ij} \le x_{ij} \le u_{ij}$, and is positive otherwise. Construct a primal ***penalty function*** $f(\mathbf{x})$ defined for any conserving flow $\mathbf{x}$ as follows:

$$f(\mathbf{x}) = \mathbf{cx} + M \sum_{(i,j)} P_{ij}(x_{ij}) \tag{11.12a}$$

where M is sufficiently large (for example, $M > 2 \sum_{(i,j)} |c_{ij}| \max\{u_{ij} - \ell_{ij}, |u_{ij}|, |\ell_{ij}|\}$, assuming that $-\infty < \ell_{ij} \le u_{ij} < \infty$ and that the algorithm is initialized with $x_{ij} = 0$ for all (i, j). Observe that the function $f(\mathbf{x})$ composes the original objective function with a penalty term and is similar to the big–M objective function. For $\mathbf{x}$ restricted to be flow–conserving, we have $\mathbf{Ax} = \mathbf{0}$ and so, $\mathbf{wAx} = 0$. Subtracting $\mathbf{wAx} = 0$ from the expression in Equation (11.12a) and noting that $(\mathbf{c} - \mathbf{wA})$ has components $(c_{ij} - z_{ij})$, we can equivalently rewrite this function as

$$f(\mathbf{x}) = \sum_{(i,j)} (c_{ij} - z_{ij})x_{ij} + M \sum_{(i,j)} P_{ij}(x_{ij}) \tag{11.12b}$$

for any conserving flow $\mathbf{x}$. Whenever we have a breakthrough in the primal phase, we modify the flow in a cycle in G such that no kilter number worsens, and the kilter number of at least one arc strictly reduces. In particular, no $P_{ij}(x_{ij})$ term increases (why?). If any such term decreases (by an integer), then since M is large enough, so does the value of $f(\cdot)$. On the other hand, if no $P_{ij}(x_{ij})$ term decreases in the cycle, then we must have $\ell_{ij} \le x_{ij} \le u_{ij}$ for all arcs (i, j) in this cycle (why?). Thus, the penalty term in Equation (11.12) remains constant during the flow change. However, from Figure 11.4 and the foregoing fact, if any arc (i, j) in the cycle satisfies $z_{ij} - c_{ij} < 0$, then we must have $\ell_{ij} < x_{ij} \le u_{ij}$ and x_{ij} must be decreasing in the breakthrough. Similarly, if $z_{ij} - c_{ij} > 0$ for any arc (i, j) in the cycle, then we must have $\ell_{ij} \le x_{ij} < u_{ij}$ and x_{ij} must be increasing in the breakthrough. In either event, the first term in Equation (11.12b) strictly falls. Because we cannot have $z_{ij} - c_{ij} = 0$ for all arcs (i, j) in the cycle in this case (why?), we again obtain a strict decrease in $f(\cdot)$. Therefore, $f(\cdot)$falls by an integer at every primal breakthrough.

Next, consider the dual phase. Note from Equations (11.4) and (11.5) that $h_{ij} - v_{ij} - (z_{ij} - c_{ij}) = 0$ for all (i, j). Using the current primal flow $\mathbf{x}$, we can equivalently rewrite the dual objective function as follows:

$$\sum_{(i,j)} x_{ij}(c_{ij} - z_{ij}) + \sum_{(i,j)} (\ell_{ij} - x_{ij})v_{ij} + \sum_{(i,j)} (x_{ij} - u_{ij})h_{ij}. \qquad (11.13)$$

During the dual phase, assuming that $\theta < \infty$, we note that $z_{ij} - c_{ij}$ increases by θ for all $(i, j) \in (X, \bar{X})$, and decreases by θ for all $(i, j) \in (\bar{X}, X)$. Hence, from Equation (11.9), the first term in Equation (11.13) remains constant (why?). Next, consider an arc $(i, j) \in (X, \bar{X})$. Referring to Equations (11.4) and (11.5) and Figure 11.8, suppose that $\bar{c}_{ij} \equiv z_{ij} - c_{ij} < 0$. Since $\bar{c}_{ij}$ increases by θ, we have that v_{ij} decreases by θ and h_{ij} remains zero in case $\ell_{ij} \le x_{ij} \le u_{ij}$, and v_{ij} decreases by min $\{\theta, |\bar{c}_{ij}|\}$ and h_{ij} increases by max $\{0, \theta - |\bar{c}_{ij}|\}$ in case $x_{ij} > u_{ij}$. Therefore, the corresponding term $(\ell_{ij} - x_{ij})v_{ij} + (x_{ij} - u_{ij})h_{ij}$ in Equation (11.13) remains unchanged if and only if $x_{ij} = \ell_{ij}$, that is, arc (i, j) is in–kilter, and increases by a positive integer otherwise. Furthermore, if $z_{ij} - c_{ij} \ge 0$, then h_{ij} increases by θ, v_{ij} remains zero, and we have $x_{ij} \ge u_{ij}$. Consequently, the corresponding term in Equation (11.13) remains unchanged if and only if $x_{ij} = u_{ij}$, that is, arc (i, j) is in–kilter, and increases by a positive integer otherwise. Similarly, consider $(i, j) \in (\bar{X}, X)$. If $\bar{c}_{ij} \equiv z_{ij} - c_{ij} > 0$, then h_{ij} falls by θ and v_{ij} remains zero in case $\ell_{ij} \le x_{ij} \le u_{ij}$, and h_{ij} falls by min $\{\theta, \bar{c}_{ij}\}$ and v_{ij} increases by max $\{0, \theta - \bar{c}_{ij}\}$ in case $x_{ij} < \ell_{ij}$. On the other hand, if $z_{ij} - c_{ij} \le 0$, then v_{ij} increases by θ and h_{ij} remains zero, while we have $x_{ij} \le \ell_{ij}$. Again, the corresponding term in Equation (11.13) remains unchanged if and only if (i, j) is in kilter, and increases by a positive integer otherwise. Because the out–of–kilter arc (p, q) is either in $(X, \bar{X})$ or $(\bar{X}, X)$ in G, we obtain a strict increase (by an integer) in the dual objective value during any iteration of the dual phase.

Because the primal penalty function $f(\cdot)$ in Equation (11.12) falls by an integer in a primal breakthrough, and the dual objective function (11.13) increases by an integer in every dual phase iteration, the difference given by the primal minus the dual function values (call it the ***duality gap***) falls by an integer every iteration. Hence, if $S_1 = \varnothing$ and $S_2 = \varnothing$ is not realized, the duality gap must become nonpositive finitely. In this event, if $P_{ij}(x_{ij}) = 0$ for all (i, j), that is, we have feasibility, then the current solution $(\mathbf{x}, \mathbf{w})$ is primal–dual optimal. On the other hand, if some $P_{ij}(x_{ij})$ is positive, then since it is an integer, we have from Equation (11.12a) that the dual value is $\geq \mathbf{cx} + M > \sum_{(i,j)} |c_{ij}| \max\{|u_{ij}|, |\ell_{ij}|\}$. This exceeds any possible feasible value realizable in Problem (11.1). Hence, the dual is unbounded and the primal is infeasible, and so we may terminate.

For the illustrative example of Section 11.4, we compute $M > 2(4 + 6 + 5 + 15 + 12) = 84$, say, $M = 85$. For the starting solution $(\overline{\mathbf{x}}, \overline{\mathbf{w}}) = (\mathbf{0}, \mathbf{0})$, we have $f(\mathbf{0}) = M(2 + 2) = 4M = 340$ and the dual objective value is $2 - 5 - 12 = -15$ from Equations (11.12) and (11.13), respectively. After the first breakthrough, the primal penalty function value becomes $2(1) + 2(-1) + M(0) = 0$. Note that feasibility is achieved and will be maintained by the algorithm henceforth (why?). After the first dual phase, the dual objective value becomes $1(2 - 5) = -3$, which is an increase of 12 units from -15. After the second (consecutive) dual phase, the dual objective value also becomes zero. Thus, we achieve optimality.

Observe that by keeping track of the duality gap, we need not work with the same out–of–kilter arc until it comes in–kilter in order to guarantee finite convergence. Furthermore, any algorithmic scheme that ensures that the duality gap decreases from its previous lowest value (not necessarily monotonically) by an integer at finite intervals, is guaranteed to converge finitely.

11.7 RELAXATION ALGORITHMS

We close our discussion in this chapter by providing some elements of another highly competitive class of primal–dual methods, known as *relaxation algorithms*, for solving minimal–cost network flow programming problems. (Actually, this technique can be extended to solve general linear programming problems as well.) The approach adopted by this algorithm is to maintain dual feasibility, along with complementary slackness with respect to a *pseudoflow*, where the latter satisfies the flow–bounding constraints but not necessarily the flow conservation equations. The method then strives to attain primal feasibility via suitable flow augmentations, and in the event of a specific type of primal nonbreakthrough following certain flow adjustments, it modifies the dual variables in order to obtain a dual ascent.

To elucidate somewhat further, consider a minimal–cost network flow problem that is cast in the following form:

$$\text{Minimize } \{\mathbf{cx} : \mathbf{Ax} = \mathbf{b}, \mathbf{0} \leq \mathbf{x} \leq \mathbf{u}\} \tag{11.14}$$

where $\mathbf{A}$ is an $m \times n$ node–arc incidence matrix of a connected digraph, $\sum_{i=1}^{m} b_i = 0$, and where $0 < u_{ij} < \infty$, $\forall (i, j)$. Examining (7.16) and (7.17), we can write the Lagrangian dual to Problem (11.14) as follows:

$$\text{Maximize } \{\theta(\mathbf{w}) : \mathbf{w} \text{ unrestricted}\} \tag{11.15a}$$

where

$$\theta(\mathbf{w}) = \mathbf{wb} + \text{minimum } \{(\mathbf{c} - \mathbf{wA})\mathbf{x} : \mathbf{0} \le \mathbf{x} \le \mathbf{u}\} \qquad (11.15\text{b})$$
$$= \mathbf{wb} - \text{maximum } \{(\mathbf{wA} - \mathbf{c})\mathbf{x} : \mathbf{0} \le \mathbf{x} \le \mathbf{u}\}.$$

Note that if we define $\bar{c}_{ij} \equiv z_{ij} - c_{ij} \equiv w_i - w_j - c_{ij}, \forall(i,j)$, as before, we can simplify the computation of $\theta(\mathbf{w})$ in Equation (11.15b) as:

$$\theta(\mathbf{w}) = \mathbf{wb} - \sum_{(i,j)}\sum u_{ij} \max\{0, \bar{c}_{ij}\}. \qquad (11.15\text{c})$$

Now, suppose that we have a current dual solution $\bar{\mathbf{w}}$. (To begin with, we can use $\bar{\mathbf{w}} = \mathbf{0}$, or estimate some advanced–start solution.) Let $\bar{\mathbf{x}}$ evaluate $\theta(\bar{\mathbf{w}})$ via Equation (11.15b). If this pseudoflow $\bar{\mathbf{x}}$ is a "flow," i.e., it also satisfies $\mathbf{A}\bar{\mathbf{x}} = \mathbf{b}$, then $\bar{\mathbf{x}}$ is a primal feasible solution for which $\theta(\bar{\mathbf{w}}) = \bar{\mathbf{w}}\mathbf{b} + (\mathbf{c} - \bar{\mathbf{w}}\mathbf{A})\bar{\mathbf{x}} = \mathbf{c}\bar{\mathbf{x}} + \bar{\mathbf{w}}(\mathbf{b} - \mathbf{A}\bar{\mathbf{x}}) = \mathbf{c}\bar{\mathbf{x}}$. Hence, the dual and primal objective values match, and so, $\bar{\mathbf{x}}$ and $\bar{\mathbf{w}}$ are respectively optimal to the primal and dual problems. Otherwise, we compute the net *excess function* $e(i) \equiv b_i - \mathbf{A}_i\mathbf{x}$ at each node i = 1,..., m, where $\mathbf{A}_i$ is the ith row of $\mathbf{A}$. Note that since $\sum_{i=1}^{m} b_i = 0$ and $\sum_{i=1}^{m} \mathbf{A}_i = \mathbf{0}$ (why?), we have that $\sum_{i=1}^{m} e(i) = 0$. However, since $\mathbf{A}\bar{\mathbf{x}} \neq \mathbf{b}$ at this point, some nodes have a net excess resident supply $(e(i) > 0)$, and some other nodes have a net unsatisfied demand requirement $(e(i) < 0)$. Observe also that in the solution of the subproblem (11.15b), if $\bar{c}_{ij} > 0$, then we must have $\bar{x}_{ij} = u_{ij}$, and if $\bar{c}_{ij} < 0$, then we must have $\bar{x}_{ij} = 0$. However, if $\bar{c}_{ij} = 0$, then we have the flexibility of selecting any value for $\bar{x}_{ij} \in [0, u_{ij}]$. The relaxation algorithm (so–called because of this Lagrangian relaxation framework employed) consequently attempts to begin with some node t having $e(t) > 0$, and examines if this excess net supply can be dissipated from node t along an incident arc (t, j) that has $\bar{c}_{tj}$ = 0, or in the reverse direction along an incident arc (j, t) that has $\bar{c}_{jt}$ = 0. If *saturating* these arcs (i.e., making the maximum permissible flow augmentations along these arcs) yet leaves an excess supply, then a *single–node nonbreakthrough* results. In this event, the dual variable value $\bar{w}_t$ of this node can be increased to obtain a dual ascent, thereby indicating that $\bar{\mathbf{w}}$ was indeed a non–optimal dual solution (see Exercise 11.35; in fact, it is the frequent occurrence and the efficiency of such single–node dual ascent steps that are largely responsible for the computational effectiveness of this method.) Naturally, if this dual ascent leads to an indication of unboundedness of the dual problem, then we can declare the primal problem to be infeasible and terminate the algorithm. Otherwise, in case all the excess $e(t)$ can possibly be dissipated, the procedure dissipates this excess only in a fashion such that the receiving node j has $e(j) < 0$. In this manner, the resultant *primal breakthrough* improves the overall feasibility of the current primal solution $\bar{\mathbf{x}}$ (while maintaining it optimal for Problem (11.15b)), without worsening any excess function values (i.e., without driving any of these values further away from zero). If $e(t)$ gets reduced to zero, then we select another node that has a positive excess (if it exists; else we have achieved optimality), and we repeat this process. On the other hand, if $e(t)$ is still positive, then we begin growing a tree T comprised of nodes having nonnegative excess values and connecting arcs that have zero $\overline{\mathbf{c}}$–values and that have positive *residual capacities* (i.e., would permit a flow change leading away from node t). In this process, treating the nodes in T as a supernode of the type t in the foregoing single–node case, and treating the cut $[T, \overline{T}]$ as the arcs incident at this supernode, we again seek a similar type of dual ascent, or a primal breakthrough, or else, grow the tree further. Therefore, while the dual solution is not yet optimal, because the excess function imbalances do not worsen and the total imbalance $\sum_{i=1}^{m} |e(i)|$ strictly falls finitely often while the dual solution remains unchanged, we finitely obtain a dual ascent by at least a unit amount. Hence, either a dual ascent leads to dual unboundedness (thereby establishing primal infeasibility), or we attain dual optimality finitely. Thereafter, we must obtain a sequence of primal breakthroughs, resulting in a finite convergence to a pair of primal and dual optimal solutions. We refer the reader to the Notes and References section for further details on this algorithm. Also, see Exercise 11.43 for some related dual ascent concepts.

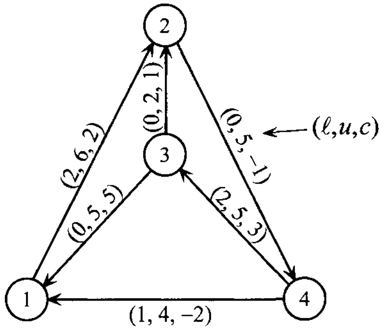
EXERCISES

[11.1] Show by manipulating the constraint equations mathematically that any minimal–cost network flow problem of the type discussed in Chapter 9 can be transformed into the out–of–kilter form (11.1) with, in particular, $0 \leq \ell_{ij} \leq u_{ij} < \infty$, $\forall (i, j)$, by adding an additional node and at most m additional arcs. Explain how you would compute, and use in this context, upper bounds on the values that the variables can attain at extreme point solutions.

[11.2] Consider the minimal–cost network flow problem: Minimize $\mathbf{cx}$ subject to $\mathbf{Ax} = \mathbf{b}$, $\boldsymbol{\ell} \leq \mathbf{x} \leq \mathbf{u}$, where $\mathbf{A}$ is a node–arc incidence matrix. Define a *conserving flow* to be any $\mathbf{x}$ satisfying $\mathbf{Ax} = \mathbf{b}$ (the conservation equations). Show that without transforming the network to the out–of–kilter form, the out–of–kilter algorithm can be applied directly on the original network with a starting conserving flow to solve the problem.

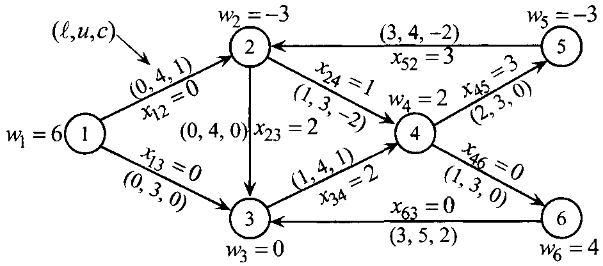
[11.3] Referring to Exercise 11.2, suppose further that we initialize the algorithm with a (possibly artificial) conserving flow that is also feasible to the bounds. Develop in detail the primal and dual phases of the algorithm. State the possible cases that may arise in this specialization. Also, show directly that the primal and dual objective values strictly improve during a (primal) breakthrough and during a dual phase, respectively.

[11.4] Solve the following problem by the out–of–kilter algorithm:



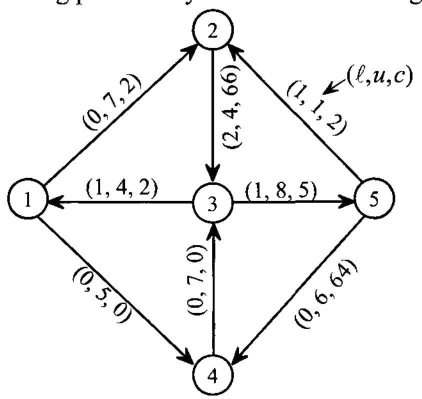
[11.5] Solve the problem of Exercise 11.4 after replacing $(\ell_{31}, u_{31}, c_{31})$ for arc (3, 1) by (–1, 5, 5).

[11.6] Consider the following network flow problem:

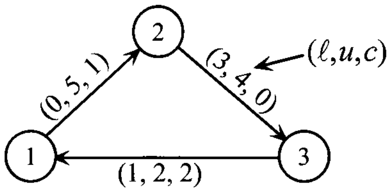


a. Give the kilter state of each arc.
b. Solve the problem by the out–of–kilter algorithm.

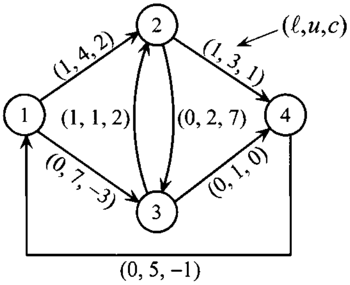
[11.7] Solve the following problem by the out–of–kilter algorithm:



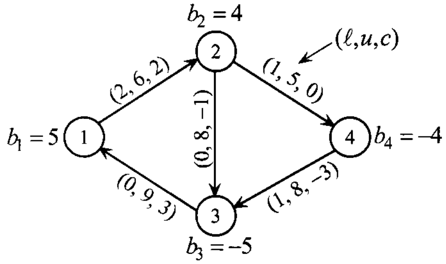
[11.8] Show how the out–of–kilter algorithm detects infeasibility in the following problem:



[11.9] Solve the following problem by the out–of–kilter algorithm:

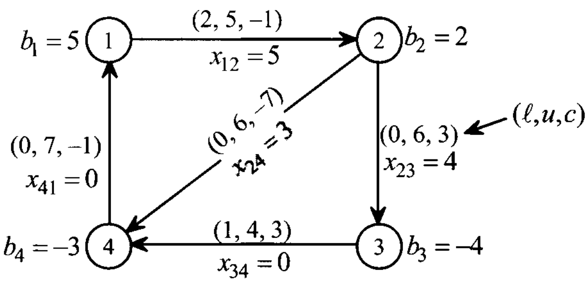


[11.10] Consider the following network flow problem:



a. Solve the problem by the network simplex method of Chapter 9.
b. Transform the problem into a circulation form and solve it by the out–of–kilter algorithm.

[11.11] Consider the following network flow problem having flows as indicated:



a. Ignoring the fact that the b_i–values are not zero, apply the out–of–kilter algorithm directly to the foregoing network with the starting x_{ij}–values as given.
b. Solve by the network simplex method of Chapter 9.
c. Are the solutions of Parts (a) and (b) the same? Discuss!

[11.12] Show that after each dual phase we can replace each new w_i by w_i − w_k, where k is some arbitrary node, and the out–of–kilter algorithm remains unaffected. (In a computer implementation, we might do this to force one dual variable, such as w_k, to remain zero and keep all of the dual variables from getting too large.)

[11.13] How can alternative optimal solutions be detected in the out–of–kilter algorithm?

[11.14] Explain in detail the data and list structures you would use in order to efficiently implement the out–of–kilter algorithm. Compare your storage requirement with that for a primal (network) simplex implementation.

[11.15] In the primal phase of the out–of–kilter algorithm, suppose that we need to find a (directed) path from q to p (in G'), given an out–of–kilter arc (p, q). Consider the construction of the usual flow change tree T described in Sections 11.2 and 11.5. Recall that starting with $T = \{q\}$, each node that is included in T receives a label equal to the (positive) flow it can receive from q. Suppose that at each step we add to T that node, which among all candidates that can be connected to nodes already in T, can receive the largest flow change label. Show that if $p \in T$, then this procedure will have found the maximum flow change path from q to p in G'.

[11.16] Is there any difficulty with the out–of–kilter algorithm when $\ell_{ij} = u_{ij}$ for some (i, j)? Carefully work through the development of the out–of–kilter algorithm for this case!

[11.17] Suppose that we have a feasible solution to Problem (11.1). Assuming that the selected arc remains out–of–kilter, is it possible for no new node to come into X after a dual variable change? Discuss!

[11.18] Suppose that we work only with the $(z_{ij} - c_{ij})$–values after the initial dual solution and never bother to change the w_i–values. Show how the w_i–values can be recovered anytime we want them. (*Hint*: The w_i–values are not unique. Set any one $w_i = 0$.)

[11.19] Interpret the dual of the out–of–kilter formulation of Problem (11.1) from a marginal–cost viewpoint.

[11.20] Show that the dual solution given in Section 11.1 is optimal for a fixed set of w_i–values.

[11.21] Demonstrate directly that the dual objective function value is unbounded as $\theta \to \infty$ when $S_1 = \varnothing$ and $S_2 = \varnothing$ during the dual phase of the out–of–kilter algorithm.

[11.22] Considering the out–of–kilter problem, show that a feasible solution exists if and only if for every choice of X and $\overline{X} = \mathscr{N} - X$ we have

$$\sum_{\substack{i \in X \\ j \in \bar{X}}} \ell_{ij} \le \sum_{\substack{i \in \bar{X} \\ j \in X}} u_{ij}$$

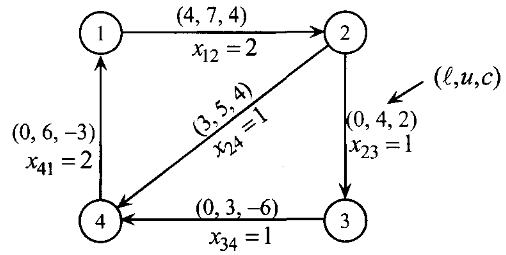
(*Hint*: Review the section on the case where $\theta = \infty$.)

[11.23] Is there any problem with degeneracy in the out–of–kilter algorithm?

[11.24] If during the primal phase we permit some kilter numbers to increase as long as the sum of all the kilter numbers decreases, will the out–of–kilter algorithm work? How could this be made operational?

[11.25] Extend the out–of–kilter algorithm to handle rational values of c_{ij}, ℓ_{ij}, and u_{ij} directly.

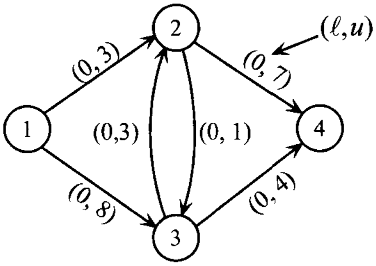
[11.26] In the out–of–kilter algorithm, show that if no cycle exists in the subset of arcs in G with $x_{ij} \neq \ell_{ij}$ and $x_{ij} \neq u_{ij}$, then the current solution corresponds to a basic solution of the associated linear program. Indicate how the out–of–kilter algorithm can be initiated with a basic solution if one is not readily available. Illustrate using the network given below with the indicated conserving flow. (*Hint*: Start with a conserving flow. If a cycle exists among arcs where $x_{ij} \neq \ell_{ij}$ and $x_{ij} \neq u_{ij}$, consider modifying the flow around the cycle in a direction that improves the penalty function value in Equation (11.12).)



[11.27] Using the results of Exercise 11.26, if the out–of–kilter algorithm is initialized with a basic solution to the linear program, show how a basic solution can be maintained thereafter. (*Hint*: Let $E = \{(i, j)\colon\ x_{ij} \neq \ell_{ij}$ and $x_{ij} \neq u_{ij}\}$. Start with only appropriate arcs associated with E as members of G'. Whenever a circuit exists, change flows. Eliminate any residual cycles in E, as in Exercise 11.26. Otherwise, after developing X, add an appropriate arc to G' that is not a member of E and that enlarges X; then work with E as much as possible again. If no circuit still exists in G', add another arc that is not in E but does not enlarge X. Continue as often as necessary. If no such arc that does not belong to E exists that enlarges X, then pass to the dual phase. This is an example of *block pivoting*.)

[11.28] Suppose that we are given a network having m nodes and n arcs, all lower bounds equal to zero, positive upper bounds, and no costs involved. Show how the out–of–kilter algorithm can be used to find the maximum amount of flow from node 1 to node m. (*Hint*: Consider adding an arc from node m to node 1 with $\ell_{m1} = 0$, $u_{m1} = \infty$, $c_{m1} = -1$ with all other c_{ij}-values set at zero.)

[11.29] Find the maximum flow in the following network from node 1 to node 4 using the out–of–kilter algorithm. (*Hint*: Refer to Exercise 11.28.)



[11.30] Suppose that we are given a network having m nodes and n arcs with a cost c_{ij} for each arc. Assume that there are no negative total cost (directed) circuits. Show how the out–of–kilter algorithm can be used to find the shortest (least) cost path from node 1 to node m via the following construction. Add an arc from node m to node 1 having $\ell_{m1} = u_{m1} = 1$ and $c_{m1} = 0$. Set the lower and upper bounds of all other arcs at 0 and 1, respectively. Can this scheme be used to find the shortest simple path from node 1 to node m in the presence of negative cost circuits? Explain.

[11.31] Let c_{ij} be the length associated with arc (i, j) in a given network having no (directed) circuits. It is desired to find a path with the shortest distance and that with the maximum distance between any two given nodes. Formulate the two problems so that the out–of–kilter algorithm can be used. Make all possible simplifications in the application of the out–of–kilter algorithm for these two problems. What is the significance of assuming that the network has no circuits? (*Hint*: See Exercise 11.30.)

[11.32] An assembly consists of three parts A, B, and C. These parts go through the following operations in order: forging, drilling, grinding, painting, and assembling. The duration of these operations in days is summarized below:

	Duration of Operation			
Part	Forging	Drilling	Grinding	Painting
A	1.2	0.8	1.0	0.7
B	2.3	0.5	0.6	0.5
C	3.2	1.0	—	0.6

Upon painting, parts A and B are assembled in two days and then A, B, and C are assembled in one day. It is desired to find the least time required for the assembly (this problem is called the *critical path problem*).

a. Formulate the problem as a network problem.
b. Solve the problem by any method you wish.
c. Solve the problem by the out–of–kilter algorithm.

d. Solve the problem by the simplified procedure you obtained in Exercise 11.31.

e. Because of the shortage of forging machines, suppose that at most two parts can go through forging at any particular time. What is the effect of this restriction on the total processing duration?

[11.33] Provide an interpretation of the primal network simplex algorithm for bounded variables in terms of an appropriately restricted execution of the primal and dual phases of the out–of–kilter algorithm. In particular, specify the flow change $\Delta \geq 0$ along with the appropriate circuit in the "primal phase," the sets X, $\overline{X}$, and the value θ for the "dual phase."

[11.34] For the problem in Equation (11.1), develop an expression for the rate in change in dual objective value as the value of the dual variables of nodes in some subset X of $\mathcal{N}$ are (marginally) increased. If this rate is positive, how would you determine the best possible value by which to increase the dual variables of nodes in X? How could you use this information algorithmically? (Also, see Exercise 11.43.)

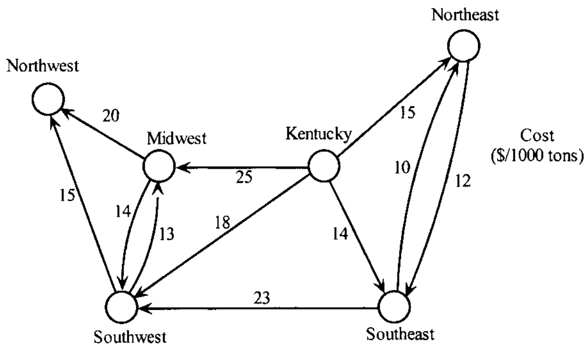
[11.35] Consider the event of the single–node nonbreakthrough described for the relaxation algorithm of Section 11.7. Derive an expression for the amount by which the corresponding dual variable value $\overline{w}_t$ can be increased, and show that this results in an increase in the dual objective function value, where the dual is specified in Equation (11.15).

[11.36] Suppose that the air freight charge per ton between locations is given by the following table (except where no direct air freight service is available):

Location	1	2	3	4	5	6	7
1	—	13	28	—	47	35	18
2	13	—	12	25	32	—	22
3	28	12	—	28	50	28	10
4	—	25	28	—	18	20	35
5	47	32	50	18	—	26	37
6	36	—	28	20	26	—	22
7	18	22	10	35	37	22	—

A certain corporation must ship a certain perishable commodity from locations 1, 2, and 3 to locations 4, 5, 6, and 7. A total of 40, 70, and 50 tons of this commodity are to be sent from locations 1, 2, and 3, respectively. A total of 25, 50, 40, and 45 tons are to be sent to locations 4, 5, 6, and 7, respectively. Shipments can be sent through intermediate locations at a cost equal to the sum of the costs for each of the legs of the journey. The problem is to determine the shipping plan that minimizes the total freight cost. Formulate the problem and solve it by the out–of–kilter algorithm.

[11.37] Coal is being hauled out of Kentucky bound for locations in the Southeast, Southwest, Midwest, Northwest, and Northeast. The network of routes is given below.

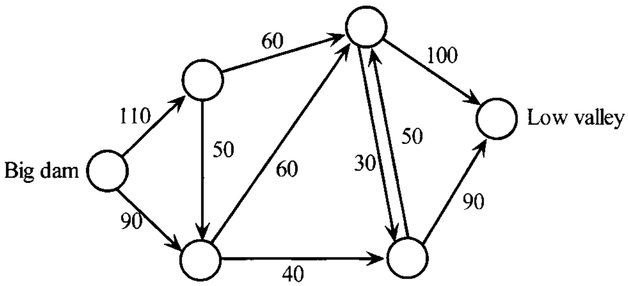


The demands are given by the following chart:

LOCATION	DEMAND (1000s OF TONS)
Southeast	5
Southwest	3
Northwest	10
Midwest	8
Northeast	20

Kentucky has a supply of 65,000 tons per week. In addition to the nonnegativity restrictions, there is an upper limit on the flow of 17,000 tons on each arc. Ignoring the return route for coal cars, use the out–of–kilter algorithm to find the least cost distribution system for coal.

[11.38] Water is to be transported through a network of pipelines from the big dam to the low valley for irrigation. A network is shown where arcs represent pipelines and the number on each arc represents the maximum permitted rate of water flow in kilo–tons per hour. It is desired to determine the maximum rate of flow from the big dam to the low valley.



a. Formulate the problem so that it can be solved by the out–of–kilter algorithm.
b. Solve the problem by the out–of–kilter algorithm.
c. Through the use of a more powerful pumping system the maximum rate of flow on any arc can be increased by a maximum of 15 kilo–tons of water per hour. If the rate is to be increased on only one pipeline, which one would you recommend and why.)

[11.39] The "Plenty of Water Company" wishes to deliver water for irrigation to three oases: the sin oasis, the devil's oasis, and the pleasure oasis. The company has two stations A and B in the vicinity of these oases. Because of other commitments, at most 700 kilo–tons and 300 kilo–tons can be delivered by the two stations to the oases. Station A is connected with the sin oasis by a 13 kilometer pipeline system and with the devil's oasis by a 17 kilometer pipeline system. Similarly, Station B is connected with the pleasure oasis by a 21 kilometer pipeline system and with the devil's oasis by a 7 kilometer pipeline system. Furthermore, the pleasure oasis and the devil's oasis are connected by a road allowing the transportation of water by trucks. Suppose that the sin oasis, the devil's oasis, and the pleasure oasis require 250, 380, and 175 kilo–tons of water. Furthermore, suppose that the transportation cost from station A is \$0.06 per kilo–ton per kilometer, and the transportation cost from station B is \$0.075 per kilo–ton per kilometer. Finally, suppose that the transportation cost between the pleasure oasis and the devil's oasis is \$0.25 per kilo–ton.

a. Formulate the problem so that the out–of–kilter algorithm can be used.
b. Solve the problem by the out–of–kilter algorithm.
c. Suppose that a road is built joining the sin oasis and the devil's oasis with a shipping cost of \$0.15 per kilo–ton. Would this affect your previous optimal solution? If so, find a new optimal solution.

[11.40] A manufacturer must produce a certain product in sufficient quantity to meet contracted sales in the next four months. The production facilities available for this product are limited, but by different amounts in the respective months. The unit cost of production also varies according to the facilities and personnel available. The product can be produced in one month and then held for sale in a later month, but at an estimated storage cost of \$2 per unit per month. No storage cost is incurred for goods sold in the same month in which they are produced. There is currently no inventory of this product, and none is desired at the end of the four months. Pertinent data are given below.

Month	Contracted Sales	Maximum Production	Unit Cost of Production
1	20	40	15
2	30	50	17
3	50	30	16
4	40	50	19

Formulate the production problem as a network problem and solve it by the out–of–kilter algorithm.

[11.41] Show how a transportation problem and an assignment problem can be solved by the out–of–kilter algorithm.

[11.42] Consider a general linear program of the form: Minimize $\mathbf{cx}$ subject to $\mathbf{Ax} = \mathbf{b}$, $\boldsymbol{\ell} \leq \mathbf{x} \leq \mathbf{u}$. Suppose that we begin with a solution $\mathbf{x}$ that satisfies $\mathbf{Ax} = \mathbf{b}$. Develop primal and dual phases of a linear programming algorithm, based on the out–of–kilter algorithm, for solving this general linear program.

[11.43] Consider the network flow problem stated in Equation (11.14) and its dual problem given by Equations (11.15a) and (11.15c), and assume that an optimum exists. Let $\bar{\mathbf{x}}$ be a primal feasible solution and let $\bar{\mathbf{w}}$ be an integral m–vector.

(a) Show that $\theta(\mathbf{w})$ in (11.15c) can be written as

$$\theta(\mathbf{w}) = \mathbf{c}\bar{\mathbf{x}} + \sum_{(i,j)}\sum \bar{x}_{ij}(w_i - w_j - c_{ij}) - \sum_{(i,j)}\sum u_{ij}\max\{0, w_i - w_j - c_{ij}\}. \quad (11.16)$$

(*Hint*: Rewrite $\mathbf{wb}$ as $\mathbf{wb} = (\mathbf{wA} - \mathbf{c})\bar{\mathbf{x}} + \mathbf{c}\bar{\mathbf{x}}$.)

(b) Consider an out–of–kilter arc (p, q) in G' and suppose that the primal phase results in a nonbreakthrough, yielding the cut $[X, \bar{X}]$ with $q \in X$ and $p \in \bar{X}$, where X is the set of nodes in the tree T described in Section 11.2. Consider augmenting the dual variables $\mathbf{w}$ according to:

$$w_i = \bar{w}_i + \theta, \forall i \in X, \text{ and } w_i = \bar{w}_i, \forall i \in \bar{X}, \text{ where } \theta \geq 0. \quad (11.17)$$

Let $h(\theta)$ be the function of θ obtained by substituting (11.17) into (11.16). Compute the *right–hand derivative* of $h(\theta)$ with respect to θ at the value $\theta = 0$ (i.e., the rate of change of h with respect to an *increase* in θ from the value $\theta = 0$), and show that this is given as follows, where $\bar{c}_{ij} \equiv \bar{w}_i - \bar{w}_j - c_{ij}, \forall (i,j)$ in this equation:

$$\left.\frac{\partial^+ h(\theta)}{\partial\theta}\right|_{\theta=0} = \left[\sum_{\substack{(i,j)\in(X,\bar{X})\\ :\bar{c}_{ij}<0}}\sum \bar{x}_{ij} - \sum_{\substack{(i,j)\in(X,\bar{X})\\ :\bar{c}_{ij}\geq 0}}\sum (u_{ij} - \bar{x}_{ij})\right] + \left[\sum_{\substack{(i,j)\in(\bar{X},X)\\ :\bar{c}_{ij}>0}}\sum (u_{ij} - \bar{x}_{ij}) - \sum_{\substack{(i,j)\in(\bar{X},X)\\ :\bar{c}_{ij}\leq 0}}\sum \bar{x}_{ij}\right]. \quad (11.18)$$

(c) Show that the second and fourth terms in (11.18) are zero. What is the interpretation of the first and third terms in (11.18) in terms of the kilter numbers of the arcs in the cut $[X, \bar{X}]$? Hence show that the expression in (11.18) has a value of at least one, thereby yielding a dual ascent of at least one unit as θ is increased. By how much can θ be increased while maintaining the same rate of ascent as in (11.18)?

(d) Discuss if it is possible to get a further ascent in the dual objective value $h(\theta)$ if θ is increased beyond the value determined in Part c. Accordingly, explain how you might solve the *line search problem* to maximize $\{h(\theta) : \theta \geq 0\}$ using the expression (11.18). Show that while this might worsen some kilter numbers, the out–of–kilter algorithm can still achieve finite convergence when the dual phase is implemented with such a linear search strategy. (*Hint*: Examine the effect on the duality gap.)

(e) If the cut $[X, \bar{X}]$ is not necessarily determined as a consequence of a nonbreakthrough in the primal phase, but is arbitrarily selected with both X and $\bar{X}$ being nonempty, verify that (11.18) still holds true. Hence, discuss how you might perform *coordinate ascent steps* or *single–node ascent steps* (see Section 11.17) in the present context when either X or $\bar{X}$ is a singleton. How would you incorporate such ascent steps within the overall framework of the out–of–kilter algorithm, and what is the potential advantage of employing such a strategy?

NOTES AND REFERENCES

1. Fulkerson [1961a] developed the out–of–kilter algorithm for network flow problems. For a slightly different development of the out–of–kilter algorithm, see Ford and Fulkerson [1962], and for a specialization, see Kennington and Helgason [1980].
2. The presentation of the out–of–kilter algorithm in this chapter follows that of Clasen [1968], especially the division of states according to values of flows x_{ij} and reduced costs $z_{ij} - c_{ij}$.
3. The spirit of the out–of–kilter algorithm can be extended to a procedure for general linear programs. This has been done by Jewell [1967]. The corresponding steps in the general case require the solution to linear programming subproblems instead of finding cycles or changing dual variables in a simple way.
4. Barr et al. [1974] provide a streamlined implementation scheme for the out–of–kilter algorithm along with computational results. Another implementation scheme is described in Singh [1986]. Computational experience and comparisons are also provided by Glover and Klingman [1978] and Hatch [1975].
5. Bertsekas and Tseng [1988b] describe a primal–dual network flow relaxation algorithm that operates two–three times faster on the NETGEN benchmark test problems compared with RNET, an effective primal simplex code developed at Rutger's University by Professor Grigoriadis and Professor Hsu. This algorithm is discussed briefly in Section 11.7. Its version RELAX–IV, and extension to solve generalized networks are respectively described in Bertsekas and Tseng [1988b, 1994]. The principal computational advantage comes from the quick dual ascent steps using cuts based on single nodes as opposed to determining steepest ascent cuts (see Exercises 11.34 and 11.43). All modern–day commercial software have specialized routines to solve network structured problems that are being continually refined (e.g., the popular software CPLEX has an efficient network simplex code, NETOPT). For extensions to handle convex–arc costs, see Bertsekas et al. [1987], and for the design of relaxation methods for general linear programs (see Tseng and Bertsekas [1987]). Also, for specializations of interior point methods to solve network flow problems and computational comparisons with simplex and relaxation methods, see Mehrotra and Wang [1996].