Now we have

$$x_{ij} = u_{ij} \qquad \text{whenever} \quad y_i + c_{ij} < y_j \tag{21.3}$$

and

$$x_{ij} = 0 \qquad \text{whenever} \quad y_i + c_{ij} > y_j. \tag{21.4}$$

Thus no arc ij satisfies (21.1) or (21.2). As we are about to prove, this circumstance indicates that the current solution is optimal.

ANALYSIS

Let us show at once that a feasible solution $\mathbf{x}$ of an upper-bounded transshipment problem

$$\text{minimize} \quad \mathbf{cx} \qquad \text{subject to} \quad \mathbf{Ax} = \mathbf{b}, \quad \mathbf{0} \le \mathbf{x} \le \mathbf{u} \tag{21.5}$$

is optimal whenever there are numbers $y_1, y_2, \ldots, y_n$ satisfying (21.3) and (21.4). The proof of this claim is simple: by virtue of (21.3) and (21.4), any feasible solution $\tilde{\mathbf{x}}$ of (21.5) satisfies

$$(c_{ij} + y_i - y_j)\tilde{x}_{ij} \ge (c_{ij} + y_i - y_j)x_{ij}$$

for all arcs ij. If $\bar{\mathbf{c}}$ denotes the vector with components $\bar{c}_{ij} = c_{ij} + y_i - y_j$ then $\mathbf{c} = \bar{\mathbf{c}} + \mathbf{yA}$ and so

$$\mathbf{c}\tilde{\mathbf{x}} = \bar{\mathbf{c}}\tilde{\mathbf{x}} + \mathbf{yA}\tilde{\mathbf{x}} = \bar{\mathbf{c}}\tilde{\mathbf{x}} + \mathbf{yb} \ge \bar{\mathbf{c}}\mathbf{x} + \mathbf{yb} = \bar{\mathbf{c}}\mathbf{x} + \mathbf{yAx} = \mathbf{cx}$$

which is the desired conclusion.

Along similar lines, we can explain the choice of the entering arc e as one satisfying $\bar{c}_e < 0$, $x_e = 0$ or $\bar{c}_e > 0$, $x_e = u_e$. In the corresponding pivot, the current feasible solution $\mathbf{x}$ is replaced by a feasible solution $\tilde{\mathbf{x}}$ such that

$$\tilde{x}_{ij} = x_{ij} \qquad \text{for all arcs } ij \notin T \text{ except the entering arc } e.$$

Since

$$\bar{c}_{ij} = 0 \qquad \text{whenever } ij \in T$$

we have

$$\bar{\mathbf{c}}\tilde{\mathbf{x}} = \bar{\mathbf{c}}\mathbf{x} + \bar{c}_e(\tilde{x}_e - x_e).$$

Now $\mathbf{c} = \bar{\mathbf{c}} + \mathbf{yA}$, $\mathbf{Ax} = \mathbf{A}\tilde{\mathbf{x}} = \mathbf{b}$ imply

$$\mathbf{c}\tilde{\mathbf{x}} = \mathbf{cx} + \bar{c}_e(\tilde{x}_e - x_e).$$

Since we set, for some nonnegative t,

$$\tilde{x}_e = x_e + t \qquad \text{in case} \quad \bar{c}_e < 0, \quad x_e = 0$$

and

$$\tilde{x}_e = x_e - t \qquad \text{in case} \quad \bar{c}_e > 0, \quad x_e = u_e$$

it follows that

$$\bar{c}_e(\tilde{x}_e - x_e) = -|\bar{c}_e| \cdot t.$$

In particular, if t is positive, then $\mathbf{c}\tilde{\mathbf{x}} < \mathbf{c}\mathbf{x}$, and so the pivot improves the value of the objective function. Such pivots are called *nondegenerate*; pivots with $t = 0$ are called *degenerate*.

To obtain the new solution $\tilde{\mathbf{x}}$, we locate the unique cycle C in $T + e$ and then adjust x_{ij} on all the arcs $ij \in C$. Each of these arcs is either *forward* (directed in the same sense as e) or *reverse* (directed in the opposite sense). In case $\bar{c}_e < 0$, we set

$$\tilde{x}_{ij} = \begin{cases} x_{ij} + t & \text{for all forward arcs } ij \\ x_{ij} - t & \text{for all reverse arcs } ij. \end{cases}$$

In case $\bar{c}_e > 0$, we set

$$\tilde{x}_{ij} = \begin{cases} x_{ij} - t & \text{for all forward arcs } ij \\ x_{ij} + t & \text{for all reverse arcs } ij. \end{cases}$$

In either case, the adjustments cancel each other out at every node of C, and so $\mathbf{A}\tilde{\mathbf{x}} = \mathbf{b}$. As we have just observed, the value of $\mathbf{c}\tilde{\mathbf{x}}$ decreases when t increases, and so we are led to choosing t as large as possible. The constraints that prevent us from increasing t beyond every bound arise from the need to maintain feasibility. More precisely, the requirement $\mathbf{0} \leq \tilde{\mathbf{x}} \leq \mathbf{u}$ amounts to

$$\left.\begin{array}{ll} x_{ij} + t \leq u_{ij} & \text{for all forward arcs } ij \\ x_{ij} - t \geq 0 & \text{for all reverse arcs } ij \end{array}\right\} \text{ in case } \bar{c}_e < 0$$

and

$$\left.\begin{array}{ll} x_{ij} - t \geq 0 & \text{for all forward arcs } ij \\ x_{ij} + t \leq u_{ij} & \text{for all reverse arcs } ij \end{array}\right\} \text{ in case } \bar{c}_e > 0.$$

It may happen that these formal constraints represent no real restrictions on the value of t. Clearly, this is the case if and only if

$$\tilde{x}_{ij} = x_{ij} + t \quad \text{and} \quad u_{ij} = \infty \quad \text{for all arcs } ij \in C. \tag{21.6}$$

Let us examine this situation more closely. To begin, (21.6) implies $\tilde{x}_e = x_e + t$ for the entering arc e and, since e is a forward arc, we have $\bar{c}_e < 0$. Now it follows that every arc $ij \in C$ must be forward. The change of $\mathbf{x}$ into $\tilde{\mathbf{x}}$ amounts to sending t extra units around the cycle C; since $\mathbf{c}\tilde{\mathbf{x}} < \mathbf{c}\mathbf{x}$, sending a unit around C must cost a negative amount. In other words, the sum of the cost coefficients c_{ij} over all the arcs $ij \in C$ must be negative. To summarize, the cycle C has the following three properties:

(i) All the arcs $ij \in C$ are directed in the same sense.

(ii) There is no finite upper bound u_{ij} on any x_{ij} such that $ij \in C$.

(iii) $\sum_{ij \in C} c_{ij} < 0$.

We shall refer to such cycles as *negative cycles*. The presence of a negative cycle indicates that the problem is unbounded: by sending larger and larger amounts around this cycle, we can make the objective function negative and arbitrarily large in magnitude.

If C is not a negative cycle, then there is at least one upper bound on the value of t, and some arc $f \in C$ provides the most stringent bound. We set t at the corresponding level ($u_f - x_f$ if f is forward, x_f if f is reverse) and define $\tilde{\mathbf{x}}$ as above. If $f \neq e$, then the tree that goes with $\tilde{\mathbf{x}}$ is $T + e - f$. It may happen, as it did in our example, that $e = f$. In that case, $\mathbf{x}$ changes into $\tilde{\mathbf{x}}$, but the tree T and the node numbers $\mathbf{y}$ remain intact. To summarize, each iteration may be of one of the following six types:

(i) $\bar{c}_e < 0, \quad x_e = 0 \quad$ and $\quad e \neq f, \quad \tilde{x}_f = 0.$

(ii) $\bar{c}_e < 0, \quad x_e = 0 \quad$ and $\quad e \neq f, \quad \tilde{x}_f = u_f.$

(iii) $\bar{c}_e < 0, \quad x_e = 0 \quad$ and $\quad e = f, \quad \tilde{x}_e = u_e.$

(iv) $\bar{c}_e > 0, \quad x_e = u_e \quad$ and $\quad e \neq f, \quad \tilde{x}_f = 0.$

(v) $\bar{c}_e > 0, \quad x_e = u_e \quad$ and $\quad e \neq f, \quad \tilde{x}_f = u_f.$

(vi) $\bar{c}_e > 0, \quad x_e = u_e \quad$ and $\quad e = f, \quad \tilde{x}_e = 0.$

Each of these six types has been illustrated on one of the six iterations in our example.

It may happen that some arc $f \in C$ limits the increase of t to zero. In that case, we have

$$f \in T \quad \text{and either} \quad x_f = 0 \quad \text{or} \quad x_f = u_f.$$

Feasible tree solutions $\mathbf{x}$ with this property are called *degenerate*. The corresponding pivot, replacing T by $T + e - f$ but leaving $\mathbf{x}$ unchanged, is also called *degenerate*.

In the presence of degeneracy, the network simplex method may cycle: we have seen an example of this phenomenon even in the case without upper bounds ($u_{ij} = +\infty$ for every arc ij). Nevertheless, Cunningham's cycling-prevention rule, discussed in Chapter 19, extends to the upper-bounded case. We used to call a feasible tree solution *strongly feasible* whenever *every arc $ij \in T$ with $x_{ij} = 0$ was directed away from the root*. Now we shall also require that *every arc $ij \in T$ with $x_{ij} = u_{ij}$ be directed towards the root*. Again, there is an easy procedure that either replaces a feasible tree solution by a strongly feasible one or else decomposes the problem into smaller subproblems. In Chapter 19, the decomposition was induced by a set S of nodes such that:

(i) There was no arc ij with $i \notin S, j \in S$.

(ii) We had $\sum_{k \in S} b_k = 0$.

Intuitively, S was thought of as an autonomous region whose home supply matched the home demand, and which possessed no import channels. In the upper-bounded case, these two conditions get replaced by

$$\sum_{k \in S} b_k = \sum_{\substack{i \notin S \\ j \in S}} u_{ij}.$$

Now S may be thought of as an autonomous region whose total net demand matches the upper bound on the volume of import. Hence in every feasible solution, the region S will import all it can and export nothing:

$$x_{ij} = u_{ij} \quad \text{whenever} \quad i \notin S, \quad j \in S, \qquad \text{and } x_{ij} = 0 \quad \text{whenever} \quad i \in S, \quad j \notin S.$$

It is an easy exercise to couch this argument in algebraic terms.

Having initialized the algorithm by a strongly feasible solution, we break ties in each choice of a leaving arc by the following rule:

> Choose the first candidate encountered when C is traversed, beginning at the join, in the direction of the entering arc e in case $\bar{c}_e < 0$, or in the opposite direction in case $\bar{c}_e > 0$.

Again, it can be proved that this procedure transforms each strongly feasible solution into another strongly feasible solution. Furthermore, it can be proved that in each degenerate pivot of this kind, the quantity $\sum(y_k - y_w)$, for a fixed w and with the summation running through all k, decreases. Hence no tree appears in two different iterations. Now cycling is purged, and so the algorithm terminates.

The only remaining problem is finding a feasible tree solution to begin with. As we did in Chapter 19, we shall get around this difficulty by solving a related *auxiliary problem*. To obtain the auxiliary problem, we first designate some node w as a root; then we add an artificial arc iw for each node i such that $b_i < 0$ and an artificial arc wj for each node j such that $b_j \geq 0$. The demands at the nodes and the upper bounds on the original (nonartificial) arcs remain unchanged, but we replace the cost coefficient c_{ij} on each original arc ij by $p_{ij} = 0$. The artificial arcs ij receive $p_{ij} = 1$ and $u_{ij} = +\infty$. In the resulting auxiliary problem, a feasible tree solution is readily apparent: the tree consists of the $m - 1$ artificial arcs. Hence we can use the network simplex method to find a feasible tree solution $\mathbf{x}^*$ minimizing the new objective function $\mathbf{px}$. If $x_{ij}^* = 0$ for every artificial arc ij, then the remaining components of $\mathbf{x}^*$ describe a feasible solution of the original problem. (In order to initialize the network simplex method on the original problem, we also need a tree T that goes with $\mathbf{x}^*$. If the optimal tree T^* in the auxiliary problem includes no artificial arcs, then we can set $T = T^*$. If T^* does include artificial arcs ij, but we still have $x_{ij}^* = 0$ for all such arcs, then an easy procedure decomposes the original problem into independent subproblems and finds a feasible tree solution in each of them. The details follow the lines of Chapter 19 and will not be repeated here.) On the other hand, if $x_{ij}^* > 0$ for some artificial arc ij then no feasible solution $\mathbf{x}$ of the original problem can exist: such an $\mathbf{x}$, extended by $x_{ij} = 0$ for every artificial arc ij, would yield $\mathbf{px} = 0 < \mathbf{px}^*$, contradicting optimality of $\mathbf{x}^*$. A further analysis of this case leads to the following result.

***THEOREM 21.1* [*D. Gale* (1957)].** An upper-bounded transshipment problem has no feasible solution if and only if there is a set S of nodes such that

$$\sum_{k \in S} b_k > \sum_{\substack{i \notin S \\ j \in S}} u_{ij}. \tag{21.7}$$

PROOF. The "if" part is easy: the left-hand side of (21.7) represents the total net demand of the region S, whereas the right-hand side represents an upper bound on the total volume of import into S. To prove the more difficult "only if" part, consider an optimal solution $\mathbf{x}^*$ of our auxiliary problem. Since the original problem has no feasible solution, there is an artificial arc uv such that $x_{uv}^* > 0$. The corresponding node numbers $y_1, y_2, \ldots, y_m$ must satisfy

$$y_v = y_u + p_{uv} = y_u + 1.$$

Let S consist of all the nodes k such that $y_k \geq y_v$. We propose to show that

$$\sum_{\substack{i \notin S \\ j \in S}} x_{ij}^* \geq x_{uv}^* + \sum_{\substack{i \notin S \\ j \in S}} u_{ij} \tag{21.8}$$

with the right-hand side summation restricted to original arcs ij, and that

$$\sum_{\substack{i \in S \\ j \notin S}} x_{ij}^* = 0. \tag{21.9}$$

Once these two facts are established, the desired inequality (21.7) will follow: it is easy to see intuitively, and to justify rigorously, that

$$\sum_{k \in S} b_k = \sum_{\substack{i \notin S \\ j \in S}} x_{ij}^* - \sum_{\substack{i \in S \\ j \notin S}} x_{ij}^*$$

for an arbitrary set S. [See also equation (19.8).]

To prove (21.8), we need only show that $x_{ij}^* = u_{ij}$ for each original arc ij with $i \notin S$, $j \in S$. But this fact follows directly from the optimality conditions since $y_i + p_{ij} =$

$y_i < y_j$. To prove (21.9), we need only show that $x^*_{ij} = 0$ for each arc ij, original or artificial, such that $i \in S, j \notin S$. Again, this fact follows directly from the optimality conditions since $y_i + p_{ij} \geq y_i > y_j$. ■

In closing, let us note that Theorem 20.1 generalizes easily to the context of upper bounded transshipment problems.

THEOREM 21.2 **(*The Integrality Theorem*).** Consider an upper-bounded transshipment problem

$$\text{minimize} \quad \mathbf{cx} \qquad \text{subject to} \quad \mathbf{Ax} = \mathbf{b}, \quad \mathbf{0} \leq \mathbf{x} \leq \mathbf{u} \tag{21.10}$$

such that all the components of the vector **b** and all the finite components of the vector **u** are integers. If (21.10) has at least one feasible solution, then it has an integer-valued feasible solution. If (21.10) has an optimal solution, then it has an integer-valued optimal solution. ■

PROBLEMS

△ **21.1** Solve the problems

$$\text{minimize} \quad \mathbf{cx} \qquad \text{subject to} \quad \mathbf{Ax} = \mathbf{b}, \quad \mathbf{0} \leq \mathbf{x} \leq \mathbf{u}$$

with

$$\mathbf{A} = \begin{bmatrix} -1 & -1 & & & & & 1 & & \\ 1 & & -1 & & & & & 1 & \\ & 1 & 1 & -1 & & & & & \\ & & & 1 & -1 & -1 & & & \\ & & & & 1 & & -1 & & 1 \\ & & & & & 1 & & -1 & -1 \end{bmatrix}, \quad \mathbf{b} = \begin{bmatrix} 0 \\ 1 \\ 2 \\ -7 \\ 0 \\ 4 \end{bmatrix}$$

and the following choices of **c**, **u**:

(i) $\mathbf{c} = [2, 4, 0, 2, 1, 3, 0, 1, 3]$, $\mathbf{u} = [2, 5, +\infty, +\infty, 5, +\infty, 5, +\infty, 3]^T$.

(ii) $\mathbf{c} = [4, 1, 3, -5, -1, 1, 4, 0, 5]$, $\mathbf{u} = [2, 5, +\infty, +\infty, 5, +\infty, 5, +\infty, 3]^T$.

(iii) $\mathbf{c} = [4, 1, 3, -5, -1, 1, 4, 0, 5]$, $\mathbf{u} = [3, +\infty, +\infty, 5, +\infty, 4, 1, 2, +\infty]^T$.

21.2 Prove in detail that Cunningham's rule prevents cycling when extended to the context of upper-bounded transshipment problems along the lines described in the text.