Upper-bounded transshipment problem

optimality conditions for standard LP compared with optimality conditions for transshipment problem

TP is the problem to maximize \(-c'x\) subject to \(x \geq 0\), \(Ax=b\). The gradients of the constraint functions are the rows of the matrix \(A\). Thus the optimality conditions assert the existence of a vector \(p\) such that

\[-c' + p'A = \begin{cases} 
\leq 0 & \text{if } x_i = 0 \\
= 0 & \text{if } x_i > 0
\end{cases}\]

This is equivalent to the familiar conditions

i. \(x \geq 0\), \(Ax=b\) (in other words, \(x\) is feasible for TP)
ii. \(p'A \leq c'\) (in other words, tension \(\leq\) cost)
iii. \((c' - p'A)x \geq 0\) (in other words, tension = cost in active arcs)

The upper bounded transshipment problem

UTP is the problem to maximize \(-c'x\) subject to \(l \leq x \leq u\), \(Ax=b\). The gradients of the constraint functions are the rows of the matrix \(A\). Thus the optimality conditions assert the existence of a vector \(p\) such that

\[-c' + p'A = \begin{cases} 
\leq 0 & \text{if } x_j = l_j \\
= 0 & \text{if } l_j < x_j < u_j \\
\geq 0 & \text{if } x_j = u_j
\end{cases}\]

In other words,

i. \(l \leq x \leq u\), \(Ax=b\) (in other words, \(x\) is feasible for TP)
ii. tension may equal cost in any arc
iii. tension \(\leq\) cost in any arc \(j\) where where \(x_j = l_j\)
iv. tension \(\geq\) cost in any arc \(j\) where where \(x_j = u_j\)

Our notion of a "bad (nontree) arc" is somewhat different in this context. There are two types of "bad" nontree arcs:

1 - Low arc \(x_j = l_j\) and tension > cost, or
2 - High arc \(x_j = u_j\) and tension < cost.

By a feasible tree solution, we mean a flow \(x\) such that \(l \leq x \leq u\) and \(Ax=b\) and such that in every nontree arc \(j\), either \(x_j = l_j\) or \(x_j = u_j\). If \(x\) is such a flow, there is a unique (up to a constant) potential \(p\) such that tension = cost in all tree arcs. If, in addition, there are no "bad" nontree arcs, then \(x\) is an optimal flow.

The algorithm

0: Start with a feasible tree solution (we will deal later with the problem of finding such a tree). Every nontree arc is either a high arc or a low arc.
1: Calculate the potential (unique up to a constant) such that \(t_j = c_j\) in every tree arc
2: If no "bad" nontree arcs, DONE.
3: Otherwise, choose a bad (nontree) arc
4: Bad arc \(j\) determines a unique circuit with tree.
   (Case 1) If \(j\) is a low arc, increase flow around cycle, in direction of
j, as much as possible. STOP if unbounded.

(Case 2) If j is a high arc, decrease flow around cycle, in direction of j, as much as possible. STOP if unbounded.

"As much as possible" refers to the fact that the flow in each arc cannot drop below the lower bound or rise above the upper bound in any arc of the cycle. The limiting arc might be the bad arc itself.

5: If limiting arc is not the bad arc, then the bad arc enters tree, and the limiting arc leaves the tree. We now have a new feasible tree. Go to step 1.

If the limiting arc is the bad arc, then the tree (and the potential) does not change, but the bad arc flips from high to low, or vice versa.

Why improvement occurs in step 4

A "bad" arc always forms a unique cycle with the tree. For a bad low arc, increasing the flow around the cycle in the direction of the bad arc decreases c'x. For a bad high arc, increasing the flow in the reverse direction decreases c'x.

Finding a feasible tree

How to find a feasible tree solution, or determine that none exists

We set up an auxiliary problem as follows. Label one node (*) as special. For every other node i,

   if i is a supply node, draw an arc from i to *
   if i is a demand node, draw an arc from * to i

(Notice that we draw an extra arc to every other node.)

Every new arc has capacity [0,∞). Define costs for the auxiliary problem as follows: the cost of every new arc is $1. The cost of all original arcs is $0. The demands for the auxiliary problem are the same as the demands for the original problem.

Specify each nontree arc to be a high arc or a low arc.

The auxiliary problem has an obvious feasible tree solution that can be used to initiate the network simplex algorithm. (Be careful to take into account the flows in high and low original arcs when computing these flows).

If the optimal value in the auxiliary problem is positive (greater than zero), then the original problem is infeasible.

If the optimal value in the auxiliary problem is zero, there are two possibilities:

   either the optimal tree uses new arcs, or
   the optimal tree does not use new arcs.

If the optimal tree does not use new arcs then it can be used as the starting tree to initiate the network simplex algorithm on the original problem.

If the optimal tree does use new arcs, a decomposition results.