LP. Lecture 8: Chapter 13: Network flow problems, cont. 1

We continue our study of (MCF): the minimum cost network flow problem.

- We now have an algorithm for calculating the tree solution x for a given spanning tree T.
- Will explain why spanning trees correspond to LP bases.
- Will also study how the other calculations of a pivot can be done.

Let A be the incidence matrix of the directed graph $D = (V, E)$. We have the following result:

Proposition. Assume that D is connected (our standing assumption). Then

$$\mathrm{rank}\,(A) = n - 1$$

where $n = |V|$.

Proof: Since the sum of all the row vectors is the zero vector, the rows in A are linearly dependent. So, if one of the rows is removed the resulting $(n-1) \times m$ matrix $\tilde{A}$ will have full row rank: the reason is that the submatrix B of $\tilde{A}$ consisting of the columns that correspond to a spanning tree will be nonsingular. Actually, such a matrix B will, after appropriate permutations of rows and columns, be triangular and have only ± 1 on the diagonal. And then the determinant must be ± 1 so B is nonsingular. □

Exercise:

1. Check this property with B for a suitably small graph.
2. Prove the property in general.

Consider again the matrix $\tilde{A}$ above, and let r be the node that corresponds to the row we deleted in A; one often calls r the root node (for the spanning trees). Let $\tilde{b}$ arise from the supply/demand vector b by deleting the component corresponding to the root node r. The original flow balance equations $Ax = -b$ are equivalent to

$$\tilde{A}x = -\tilde{b}$$

because we have deleted a redundant equation. (The deleted row in A is a linear combination of the rows in $\tilde{A}$ and because $\sum_v b_v = 0$ b_r is a corresponding linear combination of the other b_v's.)

With this rewriting the (MCF) problem becomes

$$\begin{array}{ll} \min & c^T x \\ \text{subj.to} & \\ & \tilde{A}x = -\tilde{b} \\ & x \geq O. \end{array}$$

and the coefficient matrix $\tilde{A}$ has full rowrank. We are now "in LP business"! Let $N = n - 1$, so $\tilde{A}$ is a $N \times m$ matrix, and here $m \geq N$ because D is connected.

Recall that a basis in $\tilde{A}$ is a non singular $N \times N$ submatrix of $\tilde{A}$; so, this kind of matrix corresponds to N selected columns or edges of the graph and B are a basis just when these edges constitute a spanning tree.

Theorem 13.1 An $N \times N$ square submatrix B of $\tilde{A}$ is a basis if and only if the columns in B correspond to a spanning tree in D.

Proof: If the columns correspond to a spanning tree, we can as mentioned permute into a triangular matrix with ± 1 on the diagonal; this occurs by arranging nodes and edges according to a successive elimination of leaves in the tree. If the columns do not corresponding to a spanning tree, these edges must contain a cycle (because there are $n - 1$ edges), and the corresponding columns in $\tilde{A}$ are linearly dependent (the sum of the rows in the matrix is the zero vector). □

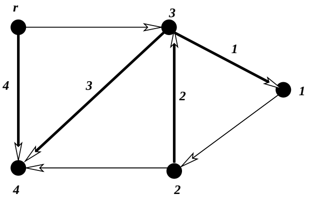
So: there is a one-to-one correspondence between the spanning trees in the graph and LP bases (in $\tilde{A}$).

We also note from the proof that:

- the structure of each basis B (i.e., is triangular with ± 1 on the diagonal) means that the linear equations with B or B^T as coefficient matrix are simple to solve by using backwards substitution and without multiplication or division. Will soon look at the details of this.

This is the main reason why (MCF) problems can be solved very efficiently by the simplex algorithm.

Example: Look at the graph below and the spanning tree T (thick lines) where the numbers denote the numbering of the nodes and the edges (r is the root node) based on leaf elimination.



The corresponding basis B (with rows and columns numbered as mentioned) becomes

$$B = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ -1 & 1 & -1 & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix}.$$

Will now take a closer look at the simplex algorithm for (MCF). First, we partition $\tilde{A}$ by

$$\tilde{A} = [B \;\; N]$$

where B is the basis that corresponds to a spanning tree T. (Actually, it is a column permutation of $\tilde{A}$ which equals the matrix on the right side.) Then $\tilde{A}x = -\tilde{b}$ is equivalent to

$$x_B = B^{-1}(-\tilde{b}) - B^{-1}Nx_N, \quad x_N \text{ free}$$

where the variable vector x is partitioned into the basic variables x_B and the nonbasic variables x_N.

The corresponding basic solution is then given by

$$x_B = B^{-1}(-\tilde{b}), \quad x_N = O.$$

We have already found an algorithm that calculates x_B for a given spanning tree T (and the corresponding basis B):

Algorithm for calculating x_B for a given T:

1. Choose a leaf of the spanning tree T which means a node is incident to precisely one edge e in T.
2. Calculate x_e for this edge e.
3. "Remove" e from T, and go back to step 1 above until all the variables are determined.

But we also need to calculate all the dual variables associated with the basis B. *How do we do that?*

Remember that the dual problem is

$$\begin{array}{ll} \max & -\sum_{v \in V} b_v y_v \\ \text{subj.to} & \\ & \begin{aligned} y_v - y_u + z_{uv} &= c_{uv} \quad ((u,v) \in E) \\ z_{uv} &\geq 0 \qquad ((u,v) \in E). \end{aligned} \end{array}$$

We will now find the dual variables like this:

- Let $y_r = 0$. Actually, we have no dual variable for the root node r because it has been deleted. But since we have only differences of the dual variables in the equations above, $y_v + \Delta$ will also satisfy the equations if y_v satisfies them. Thereby, we can "normalize" in such a way that this works with $y_r = 0$.

- By complimentary slack $z_{uv} = 0$ for every $(u, v) \in E(T)$ (alternatively: every $z_{uv} = 0$ is a nonbasic variable in the dual). So from the equations in the dual we have that $c_{uv} = y_v - y_u + z_{uv} = y_v - y_u$ which means that

$$y_v - y_u = c_{uv} \quad ((u, v) \in E(T))$$

  By starting at the root node r and working our way through T by leaf elimination we can use the equations to calculate the y_v's one by one.

3. The primal network simplex algorithm

We will start with a spanning tree T where the corresponding tree solution x_B is feasible, which means that $x_B \geq O$. We will explain how to find such a tree later.

The algorithm can be summed up like this:

- Check optimality by calculating the dual variables y and z and checking if $z \geq O$.
- If not optimal: perform a pivot. This is done by choosing an edge (u, v) where $z_{uv} < 0$ (which means a negative reduced cost), and finding a new spanning tree T' by adding e and removing a certain other edge (so that T' becomes a spanning tree). Update the tree solution x.

Comments:

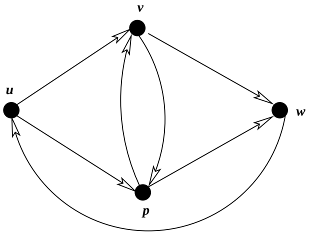
- The calculation of both x and y is done by "leaf elimination" by using the triangularity like we have seen. The calculation of each z_{uv} is done directly from the corresponding equation in the dual problem:

$$z_{uv} = c_{uv} + y_u - y_v.$$

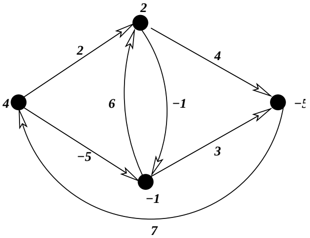
- So for each edge (u, v) outside the tree we perform this (simple) calculation. (This is still the most time consuming part for very large problems in graphs with many edges.)

We will take a closer look at the method through a small example.

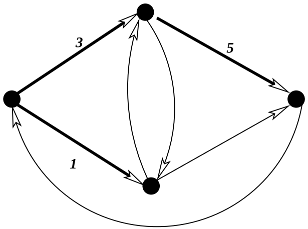
Example. The following figure shows the graph. Let w be the root node.



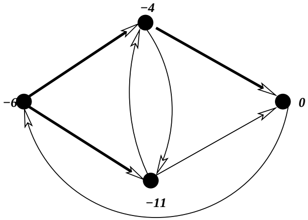
and here the values of b and c are



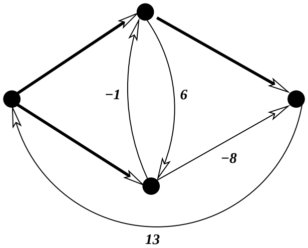
1. iteration: We choose a spanning tree (thick lines in the next figure) and compute x (also denoted) that is feasible.



After this y is computed:



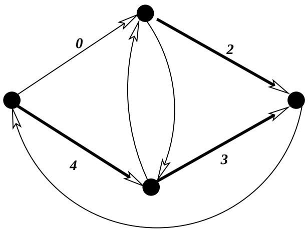
Based on y and the original costs c we find z (we will denote these only for edges outside T; in the tree $z_{uv} = 0$):



The solution is not optimal, e.g. $z_{pw} = -8 < 0$. We now choose to take this variable into the basis. So this edge is going into the tree.

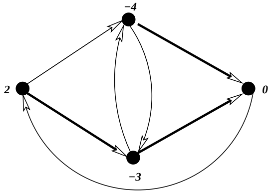
How much can we increase z_{pw}? If we let $z_{pw} = \epsilon$, we find that the basic variables get the values $x_{vw} = 5 - \epsilon$, $x_{uv} = 3 - \epsilon$, $x_{up} = 1 + \epsilon$. Here we have used the equations that express the basic variables as a function of the nonbasic variables.

We can see that the maximal value of ϵ is 3; because a larger value would give a nonfeasible primal basic solution. For $\epsilon = 3$ the new values of the basic variables will be $x_{vw} = 2$, $x_{uv} = 0$, $x_{up} = 4$. As expected, one of the basic variables becomes 0, namely x_{uv}, so we update the basis by replacing x_{uv} with x_{pw} in the basis. This gives us the new spanning tree and a corresponding tree solution as denoted in the next figure:

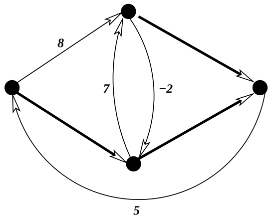


We have completed iteration 1.

2. iteration: Computing y:

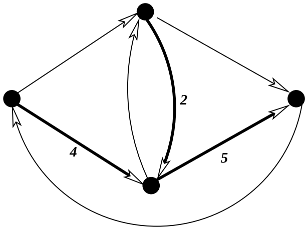


Computing z:

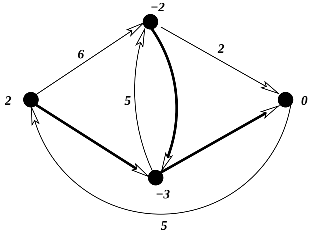


The solution is not optimal because $z_{vp} < 0$.

We now add the edge (v, p) to the tree and get a cycle. Then, we send a flow of ϵ in this cycle in the direction of (v, p). The result is that the maximal ϵ is 2 and then the flow in (v, w) will equal 0. So the tree is updated by (v, w) is replaced by (v, p). The updated tree solution x is:



3. iteration: So, once again we compute y and z which become (shown in the same figure):



Now $z \geq O$, so the solution is optimal (both the primal and the dual solution).

Problem solved!

In the next lecture we will summarize the method and make some concluding comments.