7.1

Generalized network flows

Katta G. Murty, IOE 612 Lecture slides 7

Directed $G = (\mathcal{N}, \mathcal{A}, \ell, k, p, c, b)$. $k \geq \ell \geq 0$. c = cost vector per unit flows entering at tail nodes on arcs. $p = (p_{ij})$ = vector of multipliers associated with arcs, see below. We assume $p_{ij} \neq 0$ $\forall (i, j)$. $b = (b_i)$ is the vector of **requirements** (negative of exogenous flows) at the nodes.

This min cost flow model in GN is a general model that includes all others like max flow as special cases.

If $0 < p_{ij} < 1$ arc lossy (typical example: electric power flows)

If $1 < p_{ij} < \infty$ arc gainy (typical example: money flows in multiperiod investment problms).

In most general case p_{ij} could be $-$ve too. THIS MODEL INCLUDES ALL LPs IN WHICH COEFF. MATRIX HAS ≤ 2 NONZERO ENTRIES IN EACH COLUMN.

Pure network $\Leftrightarrow p_{ij} = 1 \ \forall (i, j)$.

May be inequality constraints too. Introduce slacks. Each slack corresponds to a self-loop. Self-loop called **surplus self-loop** [**slack self-loop**] if coeff. of cooresponding slack in eq. is -1 [$+ 1$]. Define:

A(i) = set of head nodes in after i set, plus i if there is a surplus self-loop at i

B(i) = set of tail nodes in before i set, plus i if there is a slack self-loop at i

The model:

$$
\begin{aligned}
\min \quad & cf \\
\text{s. to} \quad -\sum_{j\in \mathbf{A}(i)} f_{ij} + \sum_{j\in \mathbf{B}(i)} p_{ij} f_{ij} &= b_i, \quad i \in \mathcal{N} \\
\ell \le f &\le k
\end{aligned}
$$

Theorem: If G connected, rank of coeff. matrix A of system is n or $n-1$. If rank $(A) = n-1$, problem can be transformed into a pure network flow problem.

So, in sequel assume rank $(A) = n$. For any oriented cycle C or simple path $\mathcal{P}$ its **loop factor** , or **path factor** is:

$$\frac{\text{product of } p_{ij} \text{ over } (i,j) \text{ reverse}}{\text{product of } p_{ij} \text{ over } (i,j) \text{ forward}}$$

Theorem: Can be transformed into a pure network problem if loop factors of all cycles is 1.

Theorem: The set of col. vectors corresponding to arcs in a cycle is l.d. iff loop factor of cycle is 1.

Theorem: A basis network G_B may have several connected components. Each connected component consists of a tree + an additional arc which may be a self-loop; & so contains a unique loop or cycle; & is called a **quasitree**.

By arranging rows & cols of each connected component consecutively, each basis has block diagonal form:

$$B = \begin{pmatrix} B_1 & & & \\ & B_2 & & \\ & & \ddots & \\ & & & B_t \end{pmatrix}$$

where for each $g = 1$ to t

$$
B_g = \begin{pmatrix}
a_1 & & & & & & \\
-1 & a_2 & & & & & \\
 & -1 & & & & & \\
 & & \ddots & & & & \\
 & & & -1 & a_{r_g-1} & & \\
 & & & & -1 & \boxed{a_{r_g}} & \\
 & & & & & & \\
 & & & & & \text{self loop} & \\
 & & & & & \text{portion} &
\end{pmatrix}
$$

or

$$
B_g = \begin{pmatrix}
a_1 & & & & & & & & \\
-1 & a_2 & & & & & & & \\
 & -1 & & & & & & & \\
 & & \ddots & & & & & & \\
 & & & a_{w-1} & & & & & \\
 & & & -1 & \boxed{\begin{matrix} a_w & & & & -1 \\ -1 & a_{w+1} & & & \\ & -1 & & & \\ & & \ddots & & \\ & & & -1 & a_{r_g} \end{matrix}} \\
 & & & & \text{The cycle portion}
\end{pmatrix} \tag{1}
$$

Near-triangularity

Quasitrees in a basis network:

Tributary trees in a quasitree, rooted loop labeling:

The predecessor path of every node contains the cycle in its quasitree.

How to compute primal & dual basic sols. Corresponding to a given partition (Q, L, U)?

Q = set of n basic arcs, L, U = nonbasic arcs at LB, UB respectively.

In the basic sol. $(\bar{f}_{ij})$: $\bar{f}_{ij} = \ell_{ij}$ $[k_{ij}]$ $\forall (i, j) \in L$ $[(i, j) \in U]$.

Multiply col of f_{ij} by $\bar{f}_{ij}$ for $(i, j) \in L \cup U$ and subract them from RHS constants vector, leaving remaining system in form:

$$B(\bar{f}_{ij} : (i, j) \in Q) = b'$$

Because of block diagonality, basic flows in each quasitree can be determined separately.

Flows on all tributary arcs can be determined using the above.

To find cycle flows, assume flow on one arc in the cycle to be α, then determine flows on other cycle arcs as functions of α by the following. When you go around the cycle, you get an eq. for α. This is how we use **near triangularity**.

To compute Node price vector

The system to determine $\pi = (\pi_i)$ is:

$$p_{ij}\pi_j - \pi_i \;=\; c_{ij} \quad \forall i \neq j, (i,j) \in Q$$

$$a_{ii}\pi_i = c_{ii} \quad \forall (i,i) \in Q$$

First determine π_i $\forall$ cycle nodes using above eqs. and near triangularity as above. Then node prices of non-cycle nodes are computed using above eqs. by going down each tributary tree beginning with the cycle node on it level by level.

Primal simplex opt. criterion

Let $(\bar{f}, \pi)$ be the flow, node price vector pair cooresponding to feasible partition (Q, L, U). The relative cost coeff. of $(i,j) \in \mathcal{A}$ is:

$$\bar{c}_{ij} = \begin{cases} c_{ij} - (p_{ij}\pi_j - \pi_i) & \text{if } i \neq j \\ c_{ii} - a_{ii}\pi_i & \text{if } i = j \end{cases}$$

Optimum if: $\bar{c}_{ij} > 0 \Rightarrow (i,j) \in L$, $\bar{c}_{ij} < 0 \Rightarrow (i,j) \in U$.

If opt. violated, let E = set of eligible to enter arcs which violate opt.

Pivot step: Select an arc, (g, h) say, from E to be **entering arc**.

To get basis representation of entering arc = elementary vector $\bar{y}$, need to solve $B\bar{y} = H(g,h)$ where $H(g,h)$ is original col of entering arc. $H(g,h) = -I_{.g} + p_{gh}I_{.h}$ if $g \neq h$; or $= a_{gg}I_{.g}$ if $g = h$.

Use same method as computing basic flows. Nonzero entries in $\bar{y}$ correspond to arcs in $\mathcal{P}_g(Q) \cup \mathcal{P}_h(Q)$.

Computing min ratio, new flow vector, & selection of dropping arc standard having obtained above.

Updating node labels & node prices in a pivot step

Present partition (Q, L, U), entering arc (g, h), dropping arc $(r, s) \neq (g, h)$.

New basic set of arcs $\hat{Q} = Q \cup \{(g, h)\} \setminus \{(r, s)\}$. $\hat{Q}$ contains a new cycle that Q does not iff $(r, s) \in \mathcal{P}_g(Q) \cap \mathcal{P}_h(Q)$.

If $\mathcal{P}_g(Q) \cap \mathcal{P}_h(Q) = \emptyset$ $[\neq \emptyset]$; g, h belong to different quasitrees [same quasitree].

Removing (r, s) from Q without adding (g, h) creates a unique tree in $\mathcal{P}_g(Q) \cup \mathcal{P}_h(Q)$ which we denote by T. Node prices change only for nodes in T.

Let i, j be the predecessor, successor nodes respectively among r, s. We consider several cases for node label updating.

1. $\mathcal{P}_g(Q) \cap \mathcal{P}_h(Q) = \emptyset$, and $(r, s) \in \mathcal{P}_g(Q)$: $PI(g) = -h$, $SI(h) = g$, $SI(i) = SI(j) = \emptyset$. The predecessor and successor relations on each arc along the path $\mathcal{P}_{jg}(Q)$ (the portion of $\mathcal{P}_g(Q)$ beginning with g upto node j the first time that j appears on this path) get reversed. Make corresponding changes in other labels. Update node prices for nodes in T

beginning with $\hat{\pi}_h = \pi_h$.

2. $\mathcal{P}_g(Q) \cap \mathcal{P}_h(Q) = \emptyset$, and $(r, s) \in \mathcal{P}_h(Q)$: $PI(h) = g, SI(g) = -h$, $SI(i) = SI(j) = \emptyset$. Reverse predecessor, successor relations along $\mathcal{P}_{jh}(Q)$, and make corresponding changes in other labels. Update node prices for nodes in T beginning with $\hat{\pi}_g = \pi_g$.

3. $\mathcal{P}_g(Q) \cap \mathcal{P}_h(Q) \neq \emptyset$, $(r, s) \notin \mathcal{P}_g(Q) \cup \mathcal{P}_h(Q)$: If $(r, s) \in \mathcal{P}_g(Q) \backslash \mathcal{P}_h(Q)$ [$\mathcal{P}_h(Q) \backslash \mathcal{P}_g(Q)$] update node labels, node prices as in case 1. [2.] above.

4. $(r, s) \in \mathcal{P}_g(Q) \cap \mathcal{P}_h(Q)$ but not on the cycle in the quasitree containing g and h: Here g, h belong to the same tributary tree in Q. $T \cup \{(g, h)\}$ forms a new quasitree in the new basic set $\hat{Q}$. The new cycle created containing (g, h) is the cycle in this new quasitree. Update node labels as in 1. Compute node prices for nodes in this new quasitree by procedure described earlier.

5. $(r, s) \in$ cycle in $\mathcal{P}_g(Q) \cap \mathcal{P}_h(Q)$: $T \cup \{(g, h)\}$ forms a new quasitree in $\hat{Q}$. Compute node prices for nodes in this new

quasitree by procedure described earlier.

Let x be the node on $\mathcal{P}_g(Q)$ s. th. $\mathcal{P}_{xg}(Q) = \mathcal{P}_g(Q) \backslash \mathcal{P}_h(Q)$.

Let y be the node on $\mathcal{P}_h(Q)$ s. th. $\mathcal{P}_{yh}(Q) = \mathcal{P}_h(Q) \backslash \mathcal{P}_g(Q)$.

If $(r, s) \in \mathcal{P}_{xy}(\mathbf{Q})$, make PI($h$) $= g$, SI(g) $= -h$; and reverse the predecessor, successor relationships on each arc along the path $\mathcal{P}_{jh}(\mathbf{Q})$. If $(r, s) \in \mathcal{P}_{yx}(\mathbf{Q})$, make PI($g$) $= -h$, SI(h) $= g$; and reverse the predecessor, successor relationships on each arc along the path $\mathcal{P}_{jg}(\mathbf{Q})$. Make corresponding changes in the other labels.

Phase I: If necessary, this is done exactly as in the pure min cost flow problem.