6.1

Min cost flows in Pure networks

Katta G. Murty, IOE 612 Lecture slides 6

Directed $G = (\mathcal{N}, \mathcal{A}, \ell, k, c)$. $k \geq \ell \geq 0$. c = cost vector per unit flows.

Several types of models. Each can be transformed into the other. Problems discussed earlier are special cases of this general problem.

1. **Ship $\bar{v}$ units from s to t at min cost**.
2. **Min cost circulation problem**: To find a min cost circulation. Problem in 1. can be transformed into this by introducing an artificial arc (t, s) with lower bound, capacity both $= \bar{v}$ & unit cost $= 0$.

3. **Min cost flow with an exogenous flow vector:** $V = (V_i)$ exogenous flow. Node i called:

$$\begin{array}{ccc} \textbf{Shipping node} & \text{if} & B_i = \emptyset \\ \textbf{Receiving node} & \text{if} & A_i = \emptyset \\ \textbf{Transshipment node} & \text{if} & \text{both } A_i, B_i \neq \emptyset. \end{array}$$

If there is at least one transshipment node, problem called *Transshipment problem.* E = node-arc incidence matrix. Problem is:

$$\begin{aligned} & \min cf \\ \text{s. to} \quad & Ef = V \\ & \ell \leq f \leq k \end{aligned}$$

Hence a **nec. cond. for feasibility** is $\sum V_i = 0$.

4. **Max. profit flow:** Same as 1., but each unit reaching sink can be sold there for a **premium** (= selling price at sink − buying price at source) of λ. Flow value variable here, need to find flow that maximizes total net profit.

Using Relative costs: In models 1, 2, 3 can replace c by $\bar{c} = (\bar{c}_{ij} = c_{ij} - (\pi_j - \pi_i)) = c - \pi E$ for any **node price vector** π, without changing set of opt. flows.

Two sets of opt. conds.

I. The C. S. conds. (f, π), a feasible flow vector, node price vector pair is an opt. pair if $\quad \forall (i,j) \in \mathcal{A}$

$$\pi_j - \pi_i > c_{ij} \quad \Rightarrow \quad k_{ij} \text{ finite and } f_{ij} = k_{ij}$$

$$\pi_j - \pi_i < c_{ij} \quad \Rightarrow \quad f_{ij} = \ell_{ij}$$

Can be depicted by **C. S. Diagram for arc** (i,j).

II. Opt. conds. in terms of −ve cost residual cycles: $\bar{f}$ feasible flow vector. C a cycle with selected orientation which is a **residual cycle** WRT $\bar{f}$. Define, for the selected orientation,

COST OF $C = \Sigma_{(i,j) \text{forward}}\, c_{ij} - \Sigma_{(i,j) \text{reverse}}\, c_{ij}$

Theorem: Feasible flow vector $\bar{f}$ min cost iff there exists no −ve cost residual cycle WRT it.

Canceling a residual cycle : Given a residual cycle C of residual capacity α, this:

increases flow on forward arcs of C by α
decreases flow on reverse arcs of C by α

Cost of flow vector changes by α(cost of residual cycle canceled).

Two main approaches for finding min cost flow

(a): Cycle canceling approach: Several routines available to find $-$ve cost residual cycles. Start with a feasible flow, and apply one such routine to find a $-$ve cost residual cycle, cancel it; and repeat until a feasible flow with no $-$ve cost residual cycle is obtained.

(b): Augmenting path approach: Start with an opt. flow of some value, and build it up optimally to desired value maintaining optimality throughout.

Building up flow value optimally

Consider min cost flow in $G(\mathcal{N}, \mathcal{A}, 0, k > 0, c, s, t, \bar{v})$. Let $\delta = \min\{k_{ij} : (i,j) \in \mathcal{A}\}$.

Clearly $\forall\ 0 \leq \bar{v} \leq \delta$, optimum sol. is to send a flow of value $\bar{v}$ on each of the arcs of a shortest chain from s to t.

So, consider following approach:

1. Find shortest chain from s to t. If a $-$ve cost circuit found, objective value unbounded below, TERMINATE.

Otherwise, send flow along this shortest chain until either flow value reaches $\bar{v}$ or an arc on chain is saturated.

2. If an arc saturated, no more flow can be sent on it, so delete saturated arcs from further consideration.

Find shortest chain from s to t in remaining network, continue sending flow on this chain now.

Deleting saturated arcs from further consideration, continue same way until flow value reaches $\bar{v}$.

How to obtain a min cost flow of value $> \bar{v}$ from one, $\bar{f}$, of value $\bar{v}$?

Augmentation along a min cost FAC is not guaranteed to preserve optimality.

To get min cost flow of value $> \bar{v}$ we may have to reroute flow in $\bar{f}$ on some arcs.

AMAZINGLY, augmentation along a min cost FAP always preserves optimality!

Define, cost of an FAP $= \Sigma_{(i,j)\text{forward}}\, c_{ij} - \Sigma_{(i,j)\text{reverse}}\, c_{ij}$

Theorem: $\bar{f}$ min cost flow in $G = (\mathcal{N}, \mathcal{A}, \ell, k, c, s, t)$. $G(\bar{f})$ residual network WRT $\bar{f}$.

1. Each chain from s to t in $G(\bar{f})$ corresponds to an FAP from s to t WRT $\bar{f}$ of same cost. So every shortest chain from s to t in $G(\bar{f})$ corresponds to a min cost FAP in G.

2. Let δ be the capacity of a min cost FAP in G. Augmenting λ units along it gives a flow $f(\lambda)$ which is a min cost flow of value $\bar{v} + \lambda \;\forall\;\; 0 \leq \lambda \leq \delta$.

Method for finding min cost flows of increasing values, by successively augmenting along cheapest FAPs, is called **Shortest augmenting path method**.

Out of Kilter (OK) Algo. for min cost flows

By F & F. A cycle canceling type algo. with cycle to cancel found by OK routine.

Highly popular by utilities to opt. natural gas shipments until 1970's when tree label implementations of primal simplex were shown to be faster.

General version: Can be initiated with (f, π) , arbitrary flow vector, node price vector pair.

An infeasibility measure decreases on each arc monotonically. Once f becomes feasible, it remains feasible in subsequent steps.

Feasible flow vector version: Discuss first. Initiate with (f, π) where f feasible. Feasibility maintained, & method tries to reduce violation of C. S. property.

Alternates bet. two subroutines.

Flow change Subroutine: π held constant and only f changed so that on each arc (i, j) the point $(f_{ij}, \pi_j - \pi_i)$ moves horizontally closer to C. S. curve.

Node price change routine: f held constant and only π changes ... moves vertically ...

Kilter Status of arcs: Discuss for feasible flow version now. General version will have many more states related to infeasibility.

In pair (f, π) with f feasible, arc (i, j) said to be:

α-**arc** if $\pi_j - \pi_i < c_{ij}$ and $f_{ij} = \ell_{ij}$

β-**arc** if $\pi_j - \pi_i = c_{ij}$ (& ofcourse $\ell_{ij} \leq f_{ij} \leq k_{ij}$ by feasibility)

Lower bounded β if $\pi_j - \pi_i = c_{ij}$ & $f_{ij} = \ell_{ij}$

Interior β if $\pi_j - \pi_i = c_{ij}$ & $\ell_{ij} < f_{ij} < k_{ij}$

Saturated β if $\pi_j - \pi_i = c_{ij}$ & $f_{ij} = k_{ij}$

γ-**arc** if $\pi_j - \pi_i > c_{ij}$ & k_{ij} finite, $f_{ij} = k_{ij}$

a-**arc** if $\pi_j - \pi_i < c_{ij}$ & $f_{ij} > \ell_{ij}$

b-**arc** if $\pi_j - \pi_i > c_{ij}$ & $f_{ij} < k_{ij}$

α, β, γ arcs satisfy C. S., so called **In-Kilter arcs**. a, b arcs violate C. S. , so called **out-of-kilter arcs**.

Kilter nos.: $KN(i,j)$ is a measure of how far away arc (i,j) is from satisfying C. S. $KN(i,j) = 0$ whenever (i,j) in-kilter, > 0 when (i,j) out-of-kilter.

Values of $KN(i,j)$ not used in algo., but in convergence proofs. Common def. is:

$$KN(i,j) = \begin{cases} 0 & (i,j) \in \{\alpha, \beta, \gamma\} \\ f_{ij} - \ell_{ij} & (i,j) \in a \\ k_{ij} - f_{ij} & (i,j) \in b \end{cases}$$

$\Sigma\, KN(i,j)$ is a measure of how far away (f, π) is from opt. $KN(i,j)$ is monotonic decreasing on every arc (i,j) during algo. until it becomes 0.

Permissible changes

Algo. permits only following changes in flows f_{ij} and tensions $\pi_j - \pi_i$, guaranteeing f remains feasible, and that $KN(i,j) \downarrow 0$ $\forall(i,j)$.

Status of (i,j)	Permissible changes in	
	f_{ij}	$\pi_j - \pi_i$
α	None	$\downarrow$ arbitrarily; $\uparrow$ upto c_{ij}
Interior β	Freely within bounds	None
LB β	"	$\downarrow$ arbitrarily
Saturated β	"	$\uparrow$ arbitrarily
γ	None	$\uparrow$ arbitrarily, $\downarrow$ upto c_{ij}
a	$\downarrow$ upto ℓ_{ij}	$\uparrow$ upto c_{ij}
b	$\uparrow$ upto k_{ij}	$\downarrow$ upto c_{ij}

Distinguished arc, FR Source and Sink

In each step, the OK algo. selects an out-of-kilter arc (called **Distinguished arc**), $(p.q)$ say, and tries to bring it into kilter by flow changes first, and if this is not possible then by a node price change.

If (p, q) is a [b] flow on it has to be $\downarrow$ [$\uparrow$]. To keep feasibility the changed amount has to be routed from p to q [q to p] through some other permissible path called **FRP (Flow Rerouting Path)**. So, for finding FRP, the **rerouting source, sink** are as in figure.

Amount to be rerouted is: $\theta = f_{pq} - \ell_{pq}$ [$k_{pq} - f_{pq}$] if (p, q) is a [b].

Algo. finds shortest (by no. of arcs) FRP by growing a BrFS tree rooted at the FR source using first labeled first scanned labeling routine.

Theorem: Let (p, q) be distinguished arc and $\mathcal{P}$ an FRP. $C = \mathcal{P} \cup (p, q)$ is a cycle, orient it from rerouting source to rerouting sink. Then C is a negative cost residual cycle WRT present flow.

What to do if there is no FRP?

Labeling routine terminates with a **nonbreakthrough**, a set X of labeled nodes containing rerouting source, its complement $\bar{X}$ containing rerouting sink, and no further labeling possible.

So, not possible to lower kilter no. of distinguished arc by flow changes. So, we try node price change. Situation as in following figure.

Let

$$A^1 = \{(i,j) : (i,j) \in (X, \bar{X}) \cap (\{a\} \cup \{\alpha\})$$
$$A^2 = \{(i,j) : (i,j) \in (X, \bar{X}) \cap (\{b\} \cup \{\gamma\})$$

$A^1 \cup A^2 \neq \emptyset$ as distinguished arc in it. on all arcs in A^1 [A^2] it is permissible to $\uparrow$ [$\downarrow$] tension $\pi_j - \pi_i$. So define:

$$\delta = \min\{|\pi_j - \pi_i - c_{ij}| : (i,j) \in A^1 \cup A^2\}$$

and new dual solution π' by:

$$\pi'_i = \begin{cases} \pi_i & \text{if } i \in X \\ \pi_i + \delta & \text{if } i \in \bar{X} \end{cases}$$

Can verify this leads only to permissible tension changes, & that at least one more arc (one attaining min for δ) joins label tree when labeling is resumed.

Selection of initial node price vector π^0, given feasible f

If k finite, can select π^0 arbitrarily.

So, assume that $U = \{(i,j) : k_{ij} = \infty\} \neq \emptyset$. Let N be set of nodes on arcs in U.

If an uncapacitated arc ($k_{ij} = \infty$) is b, it cannot be brought into kilter by any flow change. So algo. selects π^0 to make sure that no arc in U is b. So, π^0 has to satisfy:

$$\pi_j^0 - \pi_i^0 \leq c_{ij} \quad \forall \quad (i,j) \in U$$

If $c \geq 0 \quad \forall \quad (i,j) \in U$, clearly $\pi^0 = 0$ is acceptable.

So, assume $c_{ij} < 0$ for some $(i,j) \in U$.

Assume (N, U) connected, otherwise you need to repeat the following procedure in each connected component of (N, U) separately: Find a shortest chain tree T rooted at some node in (N, U). If this discovers a $-$ve cost circuit in (N, U), objective function unbounded below in original problem, TERMINATE. Otherwise, take π_i^0 for nodes $i \in N$ to be the distance of i in this SC tree from root.

After determining $\pi_i^0 \quad \forall i \in N$ by above procedure, you can select π_i^0 for $i \notin N$ arbitrarily (say $= 0$) to complete the π^0 vector.

The OK Algo.

Step 1: Initialization: Find feasible flow f^0, & initial node price vector π^0 as discussed above. In this process if you discover **infeasibility**, or **unboundedness** (as evidenced by a negative cost circuit consisting of solely uncapacitated arcs) TERMINATE.

Otherwise, let (f^0, π^0) be initial pair. If it satisfies C. S. conds., it is an opt. pair, TERMINATE; or go to Step 2.

Step 2: Select Distinguished arc: Select $(p.q) \in \{a, b\}$ as distiguished arc, go to Step 3.

Step 3: Labeling to find FRP: Let $\bar{s}, \bar{t}$ be rerouting source, sink. Label $\bar{s}$ with $\emptyset$, List $= \{\bar{s}\}$. Go to Step 3.1

Step 3.1: Select node to scan: If list $= \emptyset$ go to Step 5. Otherwise, delete node i from top of list to scan. Go to Step 3.2.

Step 3.2: Scanning: Let (f, π) be present pair, & i the node to scan.

Forward labeling: Label each unlabeled node $j \in \{b, \beta\}$ & satisfying $f_{ij} < k_{ij}$ with $(i, +)$ and include it at bottom of list.

Reverse labeling: Label each unlabeled node $j \in \{a, \beta\}$ & satisfying $f_{ji} > \ell_{ij}$ with $(i, -)$ and include it at bottom of list.

If $\bar{t}$ now labeled, **breakthrough**, FRP found, go to Step 4. Otherwise, go to Step 3.1.

Step 4: Flow rerouting: Find FRP $\mathcal{P}$ by a backward trace of labels from $\bar{t}$. Orient $C = \mathcal{P} \cup \{(p, q)\}$ so that forward arcs on $\mathcal{P}$ remain forward, and cancel it, leading to new flow vector $\hat{f}$. Erase labels on all nodes.

If all arcs in-kilter in $(\hat{f}, \pi)$, it is opt., TERMINATE.

If (p, q) in-kilter now, but other out-of-kilter arcs exist go to Step 2.

If (p, q) still out-of-kilter, leave it distinguished & go to Step 3.

Step 5: Node price change: Let $X, \bar{X}$ be sets of labeled, unlabeled nodes. Obtain new node price vector $\hat{\pi}$ as discussed above.

If all arcs in-kilter in $(f, \hat{\pi})$, it is opt., TERMINATE.

If (p, q) in-kilter now, but other out-of-kilter arcs exist, erase all node labels, go to Step 2.

If (p, q) still out-of-kilter, leave it distinguished, make list $=$ X & go to Step 3.1.

Example

Theorem: This version of OK algo. is finite.

Case 1: ℓ, k, f^0 integral

Case 2: c, π^0 integral

Case 3: All data real.

OK algo. for Parametric value min cost flows

Theorem: Let $(\bar{f}, \bar{\pi})$ be opt. pair of value $\bar{v}$. So all arcs are α, β, or γ in this pair. Any FAP from s to t WRT $\bar{f}$ consisting of β arcs only, has net cost of $\bar{\pi}_t - \bar{\pi}_s$ and is a min cost FAP among all FAPs WRT $\bar{f}$.

The Parametric value algo. Start with an opt. pair of any value. So all arcs are in-kilter, and they will remain in-kilter throughout this algo. Apply OK algo. using only β arcs as permissible for flow change. If δ becomes ∞ in a node price change step, it implies present flow is of maximum value (if trying to increase value); or minimum value (if trying to decrease value).

This parametric algo. is a **Shortest augmenting path method** for min cost flow implemented using OK method.

$g(v)$ = cost of min cost flow of value v, is PL Convex.

Parametric maximum profit flow

$G = (\mathcal{N}, \mathcal{A}, 0, k, c \geq 0, s, t, \lambda)$. Here λ = premium/unit material reaching t from s. It is a parameter varying from 0 to ∞. The flow value v itself is a variable in this model. Need to find a feasible flow which maximizes net profit $= \lambda v - cf$.

Opt. conds.: Pair (f, π) opt. for λ if:

$\pi_t - \pi_s = \lambda$ and

$\pi_j - \pi_i > c_{ij} \quad \Rightarrow \quad k_{ij}$ finite and $f_{ij} = k_{ij}$

$\pi_j - \pi_i < c_{ij} \quad \Rightarrow \quad f_{ij} = 0$

Algorithm: Start with $(f = 0, \ \ \pi = 0, \ \ v = 0, \ \ \lambda = 0)$ opt. for $\lambda = 0$. Apply the OK based parametric value min cost flow algo.; updating the λ at each node price change by adding δ to it.

At some stage if present sols. are $(\bar{f}, \bar{\pi}, \bar{v}, \bar{\lambda})$, & an FAP from s to t WRT $\bar{f}$ of capacity ∞ is found in the β-subnetwork; this FAP has cost $\bar{\lambda}$ and ∞ residual capacity. So, in this case, $\forall \lambda > \bar{\lambda}$, the maximum profit will be $+\infty$.

General version of OK algo. initiated with arbitrary flow vector

Of theoretical interest, needed for developing a polynomially bounded version of OK algo. based on **Scaling of lower bounds and capacities on arcs**.

When initiating OK with (f, π) where f is infeasible (here we assume that flow conservation holds at all nodes, but that the bounds may be violated), we have to define several new out-of-kilter states. For example, if $f_{ij} < \ell_{ij}$ and $\pi_j - \pi_i < c_{ij}$, arc (i, j) is called an a_1 arc, and the permissible flow change on it is to increase the flow upto ℓ_{ij}. Etc.

The algorithm is similar, and finiteness proofs continue to hold.

Polynomially bounded Scaling implementation of OK algo.

Of theoretical interest. Assume ℓ, k are integer vectors. p is the maximum no. of binary digits among the entries in these vectors.

The complexity of this implementation is $O(pmn^2)$.

Primal Network Simplex

$G = (\mathcal{N}, \mathcal{A}, \ell, k, c, V)$. Assume G connected. Specialization of the **Bounded variable Primal Simplex Method** to solve this problem. Each BFS corresponds to a partition of set of arcs into (T, L, U) where:

$T =$ spanning tree consisting of basic arcs

$L =$ nonbasic arcs in which $f_{ij} = \ell_{ij}$ in the basic solution

$U =$ nonbasics with k_{ij} finite & $f_{ij} = k_{ij}$ in basic solution.

Opt. Conds.: Primal feasible partition (T, L, U) associated with pair (f, π) optimal if it is **dual feasible**, i.e., if

$$\bar{c}_{ij} = c_{ij} - (\pi_j - \pi_i) \geq 0 \quad \forall \ (i,j) \in L$$

$$\bar{c}_{ij} = c_{ij} - (\pi_j - \pi_i) \leq 0 \quad \forall \ (i,j) \in U$$

How to compute dual basic sol. for given (T, L, U)?

Select a node, say node n as root node & fix $\pi_n = 0$ (Reason: We treat the flow conservation eq. corresponding to n as redundant & eliminate it.)

$\forall (i, j) \in T$, we have $\pi_j - \pi_i = c_{ij}$.

This system can be solved by backsubstitution beginning at root node n, and going down the tree level by level.

Theorem: $\pi_i =$ cost of predecessor path of i treated as a path from root to i.

$\bar{c}_{ij} =$ net cost of funda. cycle of nonbasic arc (i, j) WRT T, oriented so that (i, j) is forward on it.

How to compute the primal basic sol. for given (T, L, U)?

Backsubstitution. Start at nonroot terminal node, find flow on basic arc incident at it, and repeat.

Updating sols. & tree labels in a pivot step

Simplex algo. begins with a feasible partition & checks it for opt. If violated, it selects one violating arc as **entering arc**, leading to a **pivot step**. In it, entering arc may replace an in-tree arc, this is called **dropping arc**.

Let e be entering arc with its relative cost $\bar{c}_e$.

Pivot cycle = funda. cycle of entering arc.

If $e \in L, \bar{c}_e < 0$ $[e \in U, \bar{c}_e > 0]$ flow on e needs to be increased [decreased] so orient pivot cycle $\mathcal{C}$ so that e is a forward arc [reverse arc].

With this orientation $\mathcal{C}$ is a negative cost cycle.

Min. ratio, dropping arc: Add $+\theta$ $[-\theta]$ to the flow on each forward [reverse] arc on $\mathcal{C}$.

Min ratio in this pivot step = Max $\theta \geq 0$ that keeps all flows on $\mathcal{C}$ within bounds

D = set of arcs on $\mathcal{C}$ which tie for min ratio

If min ratio = ∞, pivot cycle is an uncapacitated $-$ve cost circuit, obj. func. unbounded below, TERMINATE.

If min ratio $= 0$, pivot step is **degenerate**, **nondegenerate** if min ratio > 0. If min ratio finite and > 0, pivot cycle is a $-$ve cost residual cycle. Make $\theta =$ min ratio to get new flow vector.

If $e \in D$, no change in T; e moves from L or U where it is currently, into the other set. No change in dual sol. Go to next step in simplex algo. with new partition.

If $e \notin D$ select one of the arcs in D as the dropping (in-tree) arc to be replaced by e. In this case we will get a new tree, partition, and dual sol. with which the simplex algo. moves to next step.

Predecessor indices: Change only for nodes on **pivot stem**. New $P'(j_1) = \pm i_1$, $P'(j_u) = -\text{Sign}(P(j_{u-1})j_{u-1}$ for $u = 2$ to $t+1$.

Successor indices: change only for nodes on entering arc, pivot stem, & dropping arc.

Brother indices: change only for nodes $S(i_1), YB(j_*), EB(j_*)$ and nodes in the set $H(j_*) = j_*$ and all descendents of j_* in present tree.

Thread index: changes only for nodes on pivot stem and their eldest and youngest children.

Node prices: New $\pi'_i = \begin{cases} \pi_i & \text{if } i \notin H(j_*) \\ \pi_i + \alpha \bar{c}_e & \text{if } i \in H(j_*) \end{cases}$

where $\alpha = +1 \quad [-1] \quad$ if entering arc is $(i_1, j_1) \quad [(j_1, i_1)]$.

Pivot choice (entering arc selection) rule: Most successful one is **outward node - most violated rule**: Examine nodes, looking for eligible arcs in forward star of each examined node. At 1st such node, select the outward arc with most violation as entering arc.

Phase I: Select a spanning tree T_0, and then generate an initial partition (T_0, L_0, U_0). Let f^0 be the basic sol. corresponding to it. Let

$$K_1 = \{(i,j) : f^0_{ij} < \ell_{ij}\} \quad K_2 = \{(i,j) : f^0_{ij} > k_{ij}\}$$

If $K_1 \cup K_2 = \emptyset$, f^0 feasible, go to Phase II with it.

If $K_1 \cup K_2 \neq \emptyset$, $\forall (i,j) \in K_1$ $[K_2]$ change lower bound to f^0_{ij} $[k_{ij}]$ and capacity to ℓ_{ij} $[f^0_{ij}]$, and define c^*_{ij} to be -1 $[+1]$. Define c^*_{ij} to be 0 $\forall (i,j) \notin K_1 \cup K_2$

(T_0, L_0, U_0) feasible to modified problem. Phase I is to min $c^* f$ on modified network beginning with (T_0, L_0, U_0). During Phase I, whenever flow vector changes, if:

For $(i,j) \in K_1$**:** flow increases but still $< \ell_{ij}$, change lower bound on this arc to the new flow amount.

If flow becomes $= \ell_{ij}$, restore original LB, capacities on this arc; take it out of K_1 and make $c^*_{ij} = 0$. You need to recompute Phase I node price vector.

For $(i,j) \in K_2$**:** flow decreases but still $> k_{ij}$, change capacity on this arc to the new flow amount.

If flow becomes $= k_{ij}$, restore original LB, capacities on this arc; take it out of K_2 and make $c^*_{ij} = 0$. You need to recompute Phase I node price vector.

Move to Phase II when $K_1 \cup K_2$ becomes $\emptyset$.

If Phase I terminates without $K_1 \cup K_2$ becoming $\emptyset$, original problem infeasible.

Cycling & its resolution: (T, L, U) associated with BFS $\bar{f}$. Arc (i, j)

Interior	if $\ell_{ij} < \bar{f}_{ij} < k_{ij}$
Lower boundary	if $\bar{f}_{ij} = \ell_{ij}$
Saturated	if $\bar{f}_{ij} = k_{ij}$

(T, L, U) and $\bar{f}$ **primal nondegenerate** if all in-tree arcs are interior, **primal degenerate** otherwise.

If $\hat{f}$ feasible, it is BFS iff set of interior arcs in it form a forest; nondegenerate BFS iff they form a spanning tree.

Under degeneracy primal algo. can cycle. Example by L. Johnson, paper written by his Secretary B. Gassner. After 12 degenerate pivot steps it returns to starting partition completing the cycle.

Conditions for Strong feasibility: Feasible (T, L, U) with a root node selected for T is **Strongly feasible (SF)** if:

in-tree arc lower boundary $\rightarrow$ directed away from root

in-tree arc saturated $\rightarrow$ directed towards root.

Examples: 1. Assignment example with C_1 as root, 2. Same example with R_4 as root.

How to retain SF in a pivot step: THEOREM: Starting from an SF partition, if you choose dropping arc from D in each pivot step to be the first arc in D encountered while traveling the pivot cycle in the direction of its orientation discussed earlier, from apex back to the apex, SF will be preserved.

THEOREM: Resolution of cycling: Starting with an SF partition, if you use above dropping arc choice rule in every pivot step; then in each degenerate pivot step each π_i either stays the same or decreases, and $\sum \pi_i$ strictly decreases. So, cycling can't occur.

How to obtain an initial SF partition: Phase I can be set up so that initial partition is SF.

Or from any feasible partition with β in-tree arcs wrongly oriented for SF, an SF partition can be obtained by carrying out at most $\beta(n-1)$ degenerate pivot steps.

Stalling: A finite but exponentially long sequence of degenerate pivot steps in simplex algo.

Even when cycling is resolved, stalling can occur.

Stalling can be prevented in SF Network simplex method by selecting entering arc using rules such as **LRC**, or any rule which examines each arc periodically (at least once in γm steps for some γ) and enters it if it is eligible.

Polynomially bounded Network simplex method: Recently J. Orlin (MPB Vol. 78, no. 2, 1 Aug. 1997) developed a version of network simplex method with special entering and dropping arc choice rules, and proved that it is polynomially bounded.

Shortest aug. path method

$G = (\mathcal{N}, \mathcal{A}, 0, k, c \geq 0, s, t, \bar{v})$. Given (f, π) where f is feasible of value v, the pair is opt. for that value if:

$$0 < f_{ij} < k_{ij} \quad \Rightarrow \quad \bar{c}_{ij} = c_{ij} - (\pi_j - \pi_i) = 0$$

$$\bar{c}_{ij} < 0 \quad \Rightarrow \quad f_{ij} = k_{ij} < \infty$$

$$\bar{c}_{ij} > 0 \quad \Rightarrow \quad f_{ij} = 0$$

If opt. conds. hold, the cost vector $\bar{c}'$ in residual network $G(f, \pi) = (\mathcal{N}, \mathcal{A}(f), 0, \kappa, \bar{c}')$ is ≥ 0, so shortest chains in it can be found by Dijkstra's method. This method uses this fact.

Shortest aug. path method

Initialization: Start with ($f^0 = 0, \pi^0 = 0$). Optimal for value 0 because $c \geq 0$.

General step: Let (f^r, π^r) be present opt. pair of value $v^r < \bar{v}$.

Find SC tree rooted at s in $G(f^r, \pi^r)$ using Dijkstra, terminate as soon as t permanently labeled.

If Dijkstra terminates before t permanently labeled, no FAP from s to t WRT f^r; so v^r is max flow value in G, so original problem infeasible, TERMINATE.

Otherwise, let μ_i^r = length of shortest chain from s to i in $G(f^r, \pi^r)$ if i is permanently labeled; $= \mu_t^r$ otherwise. Find FAP corresponding to shortest chain from s to t and augment it by $\gamma_r = \min\{\epsilon_r =$ its residual capacity, $\bar{v} - v^r\}$ leading to new flow vector f^{r+1} of value $v^{r+1} = v^r + \gamma_r$. Define $\pi_i^{r+1} = \pi_i^r + \mu_i^r$ $\forall i \in \mathcal{N}$.

If $v^{r+1} = \bar{v}$, (f^{r+1}, π^{r+1}) opt. pair, TERMINATE.

Otherwise go to next step with new pair (f^{r+1}, π^{r+1}).

Classical P-D method for min cost circulation

$G = (\mathcal{N}, \mathcal{A}, \ell, k, c)$. $\ell \leq k < \infty$. Pair (f, π) said to be **admissible** if:

(1) f is bound feasible (i.e., $\ell \leq f \leq k$) but may not satisfy conservation eqs. at nodes.

(2) all arcs are α, β, or γ as in the OK algo.

Define the **Deficit** at node i in f to be $d_i = f(\mathcal{N}, i) - f(i, \mathcal{N})$. It could be $+$ or $-$, and $\Sigma\, d_i = 0$.

Method maintains admissibility throughout, & tries to make all deficits to 0 monotonically by flow change & node price change steps.

P-D algo.

Initialization: Start with an admissible pair.

Flow change step: Let (f, π) be current pair with deficit vector $(d_i) = d$.

If $d = 0$, (f, π) opt., TERMINATE.

Otherwise select a +ve deficit node i_0, root a tree at it and put it in list. Go to scanning step.

Scanning: If list $= \emptyset$, go to node price change step. Otherwise delete a node p from list for scanning.

Forward labeling: $\forall$ unlabeled nodes j satisfying: $(p, j) \in \mathcal{A}$, $(p, j) \in \beta$, and $f_{pj} < k_{pj}$; label j with $(p, +)$ and put it in list.

Reverse labeling: $\forall$ unlabeled nodes j satisfying: $(j, p) \in \mathcal{A}$, $(j, p) \in \beta$, and $f_{jp} > \ell_{jp}$; label j with $(p, -)$ and put it in list.

Breakthrough: If a node q with $-$ve deficit now labeled, a

Flow altering path (FAlP) has been found, and we have a breakthro. Trace it using labels. Its capacity is $\epsilon = \min\{d_{i_0}, -d_q, k_{ij} - f_{ij}$ for forward arcs, $f_{ij} - \ell_{ij}$ for reverse arcs$\}$. Augment flow by ϵ along FAlP, chop down tree and start another flow change step.

If no breakthrough, continue scanning until either list $= \emptyset$ or breakthro' occurs.

Node price change step: Let $X, \bar{X}$ be sets of labeled, unlabeled nodes. Let $A_1 = \{\alpha$ arcs in $(X, \bar{X})\}$, $A_2 = \{\gamma$ arcs in $(\bar{X}, X)\}$.

If $A_1 \cup A_2 = \emptyset$, no feasible circulation in G, TERMINATE.

If $A_1 \cup A_2 \neq \emptyset$, let $\delta = \min\{|c_{ij} - (\pi_j - \pi_i)| : (i,j) \in A_1 \cup A_2\}$.

Define new dual sol. π' by:

$$\pi'_i = \begin{cases} \pi_i + \delta & \text{for } i \in \bar{X} \\ \pi_i & \text{for } i \in X \end{cases}$$

Put all labeled nodes in list and resume tree growth by going to the scanning step.

Strongly polynomial algos.

Consider min cost circulation in $G = (\mathcal{N}, \mathcal{A}, \ell, k, c)$. No. of data elements is $5m + 2n$. An algo. is said to be strongly polynomial if:

(1) no. of arithmetic operations bounded above by a polynomial in no. of data elements even if data real, assuming exact arithmetic

(2) when applied on instances with rational data, size of all intermediate numbers bounded above by a polynomial in size of original instance.

Several strongly polynomial algos. for pure min cost flows. Most based on :

Theorem: Let (f, π) be a feasible circulation, node price vector pair satisfying:

$$\bar{c}_{ij} \geq \alpha \Rightarrow f_{ij} = \ell_{ij}, \qquad \bar{c}_{ij} \leq -\alpha \Rightarrow f_{ij} = k_{ij}$$

for some $\alpha \geq 0$.

If $0 \leq \alpha < 1/n$ and c an integer vector then f is a min cost circulation.

If $\alpha > 0$ and $(p, q) \in \mathcal{A}$ satisfies $|\bar{c}_{pq}| \geq n\alpha$ then in every min cost feasible circulation, the flow on (p, q) must be f_{pq}.

One algo.: Initiate with a feasible circulation. Find a minimum mean cycle. If its cost ≥ 0 circulation opt., TERMINATE. If its cost < 0, cancel that cycle and repeat.

Needs at most $O(nm^2 \log n)$ cancellations. So overall complexity at most $O(n^2 m^3 \log n)$.