An Appropriate Method for The Maximum Capacity Route Problem [I]

Ichibei KUDO, Makoto BITO

ネット・ワークの最大能力の最適方法

工藤市兵衛・尾藤　信

The maximum capacity route problem is to find a route $R = \{(s, i), (i, j), \cdots, (k, m), (m, t)\}$ from the source s to the sink t wnose capacity $c(R) = \min_{(i,j)\in R} c_{ij}$ is maximum, where c_{ij} is the capacity of the directed edge (i, j). It is the route that allows the greatest flow from s to t.

Let $B = (S, F)$ be a subgraph of a given directed graph $G = (V, E)$ such that, for every $i \in S$, B contains a unique route from s to i, that is ,B is a directed tree rooted at s. We shall describe a simple labeling procedure for gradually expanding the directed tree rooted at s. Each time the tree is expanded, the new tree, B, will be assigned a value h(B). When the tree B reaches the sink t, B contains an unique route from s to t which is the maximum capacity route in G with the capacity h(B).

i) Start with $B = (\{s\}, \phi)$ and $h(B) = \infty$

ii) Given a tree $B = (S, F)$ with $t \in S$, from a new tree $B' = (S', F')$ as follows: First, find the value c such that,

(1) $$c = \max\{c_{rj} : r \in s \text{ and } j \in T\}$$

where $T = V - S$. Then, define a subset, D, of the cut (S, T) as

(2) $$D = \{(i, j) / (i, j) \in (S, T) \text{ and } c_{ij} \geq h\},$$

where

(3) $$h = \min\{h(B), c\}.$$

Also let

(4) $$K = \{k \in T / (i, k) \in D\}.$$

Now, for every $k \in K$, we select an edge $(i, k) \in D$ and add to F to from an new F'. And also

$$S' = S \cup K. \quad \text{Then, set}$$

(5) $$h(B') = h.$$

At this point, one may define a capacity transformation

(6) $$c'_{ik} = h \text{ for every } (i, k) \in D,$$

though it is not essential in our algorithm. We, thus, simply repeat this tree expansion until either the tree reaches the sink t, or the cut (S, T) for a tree $B = (S, F)$ is empty.

It is to be noted that, by the way of constructing the tree, $B = (S, F)$, and defining h(B), the route in the tree from s to any $i \in S$ has a capacity equal to or greater than h(B). In particular, when a new tree $B' = (S', F')$ is formed from $B = (S, F)$ by adding new vertices K, the route in the tree from s to any $k \in K$ is equal to h(B'). When the capacity transformation defined in (6) is also performed during procedure, the edge capacities along the route, $\{(s, i), (i, j) \cdots, (m, n)\}$, in the tree $B = (S, F)$ from s to any $n \in S$ from a non-increasing sequence, i.e., $c_{si} \gg c_{ij} \gg \cdots \gg c_{mn}$, and thus the route capacity is determind by the capacity of the last edge, (m, n), along the route. We now show,

Theorem: When the tree B reaches the sink t, the route in the tree from s to t is a maximum capacity route in G with the capacity h(B).

Proof: Since any route, R, from s to t and any cut, C, separating s and t in G, have at least an edge, (p, q), in common, i.e., $R \cap C \neq \phi$, we have

$$c(R) \ll c_{pp} \ll m(C), \text{ where}$$

(7) $$c(R) = \min\{c_{ij} : (i, j) \in R\} \text{ and}$$

(8) $$m(C) = \max\{c_{ij} : (i, j) \in C\}.$$

Hence, if we specify a procedure to find a route, R^*, such that $c(R^*) = m(C^*)$ for some cut, C^*, then R^* is proved to be a maximum capacity route.

Now, let B^* be the tree which reaches the sink t for the first time, then $h(B^*)$ is the capacity of the route R^* in the tree from s to t. However, by the way in which h(B) is determined by the procedure described in (1), (3) and (5), $h(B^*)$ represents the value $c = \max\{c_{rj} : r \in S,\ j \in T\}$ for some cut (S, T) with $s \in S$ and $t \in T$. This completes the proof.

If the procedure stops short of reaching t, the cut (S, T) for some tree B = (S, F) is empty and there is no route from s to t in G. Since the set S expands by at least one vertex at each iteration, the tree will necessarily reach t if there exists at least one route from s to t in G.

In fact, our algorithm provides a new proof to the min-max theorem concerning routes and cuts which was first pointed out by D. R. Fulkerson:

Theorem: let c(R) and m(C) be defined as in (7) and (8), then

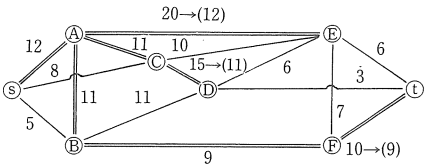
$$\max_{R \in \tilde{R}} c(R) = \min_{C \in \tilde{C}} m(C)$$

where $\tilde{R}$ is the collection of routes from s to t in G, and $\tilde{C}$ is the collection of cuts separating s and t in G.

As a variation of the algorithm, given a subgraph B = (S, F), one may from a new subgraph B′ = (S′, F′) by adding all edges of D defined in (2) to F. Then, the expanding subgraph may not be a directed tree and when the subgraph reaches t for the first time, any route in B from s to t is a maximum capacity route.

Example:

We shall find a maximum capacity route in the following mixed network. The transformed capacities will be indicated by the symbol→( ) along edges. The labeled edges show the tree B when it reached t for the first time:



Step 1: S = {s}; (S, T) = {(s, A), (s, C), (s, B)}
h = min{∞,12} = 12, K = {A}.

Step 2: S = {s, A}; (S,T) = {(s, B), (s, C), (A, B), (A, C), (A,E)},
h = min{12, 20} = 12, K = {E}.

Step 3: S = {s, A, E}; (S, T) = {(s, B), (s, C), (A, B), (A, C), (E, C), (E, D), (E, F), (E, t)},
h = min{12, 11} = 11, K = {C, B}.

Step 4: S = {s, A, B, C, E, }, (S, T) = {(B, F), (B, D), (C, D), (E, D), (E, F), (E, t)},
h = min{11, 15} = 11, K = {D}.

Step 5: S = {s, A, B, C, D, E}; (S, T) = {(B, F), (D, t), (E, F), (E, t)},
h = min{11, 9} = 9, K = {F}.

Step 6: S = {s, A, B, C, D, E, F}; (S, T) = {(D, t), (E, t), (F, t)}.
h = min{9, 10} = 9, K = {t}.
t ∈ S′, terminate. The terminal value O + h = 9.

The maximum capacity route = {(s, A), (A, B), (B, F), (F, t)} with the capacity 9.

Refernces

(1) Fullkerson, D. R., "Flow Networks and Combinatiorial Operations Research,"
(2) Hu, T. C., "The Maximum Capacity Route Problem,"
(3) Pollack, M., "The Maximum Capacity Route Through a Network,"
(4) Moder and Phrillps., "Project Management with CPM and PERT."
(5) Miller R. W., "Schedule, Cost and Profit Control with PERT"
(6) Jucius. M. J., "Elements of Management Action."
(7) Anlill, J. M., "Crilical Pa Melhods in Construction on Practice."
(8) 森 竜雄著, 「PERT」 I, II, 日本能率協会
(9) 須永昭雄著, 「PERT系のプログラミング」朝倉書房
(10) 刀根 薫編, 「PERT議座 I, II, III, IV」東洋経済
(11) 森口博夫訳, 「PERT」同文館