Mean Number Waiting for Service

First-come, first-served	1.7	3.4	8.5	19.1	33.1	103.4
Priority	1.2	2.3	5.0	10.5	17.5	52.8
Utilization	70%	80%	90%	95%	97%	99%

Mean Waiting Time

First-come, first-served	3.7	6.3	14.3	30.1	51.2	156.8
Priority high	1.7	2.1	2.6	2.9	3.0	3.0
Priority low	5.7	10.6	25.9	57.3	99.4	310.5
Utilization	70%	80%	90%	95%	97%	99%

For high utilization, the priority discipline thus gives very different results, in this case, from first-come first-served. The number waiting for service is practically halved, and the waiting time of the customers with high priority (who form three quarters of the arrivals) is very considerably reduced, at the expense of doubling the waiting time of the remainder. The example would tend to show that an alteration of the queue discipline is likely to have considerable effect in many actual situations.

REFERENCES

1. A. Cobham, "Priority Assignment in Waiting Line Problems," *Opns. Res.* **2,** 70–76 (1954).
2. H. Kesten and J. Th. Runnenberg, "Priority in Waiting Line Problems," *Proc. Koeningl. Nederl. Ak.* Series A, 60, No. 3.
3. D. G. Kendall, "Some Problems in the Theory of Queues," *J. Royal Stat. Soc.* **B.13,** 151–185 (1951).

THE MAXIMUM CAPACITY THROUGH A NETWORK

Maurice Pollack

Stanford Research Institute, Menlo Park, California

(Received June 8, 1960)

THE MAXIMUM capacity route is defined to be the route between any two given cities (nodes) that allows the greatest flow. This differs from the maximum capacity problem[1] in that only the single best route is desired; in the maximum capacity problem, the object is to find the maximum flow between two nodes using as many different routes as needed. The maximum capacity route problem arises in automatic teletype networks where, for any given origin-destination pair, there is only one route specified.

The maximum capacity route problem can be formulated as follows. A set of N cities (nodes) is given with every two connected by a road (link). Also given are the capacities of the links that are not in general symmetric—that is, the capacity from node i to j is not necessarily equal to that from j to i. All link capacities are assumed to be nonnegative and there is assumed to be no restriction on the capacity of a node. It is desired to find the route from any one node to any other node that maximizes the capacity of the route. Where links do not exist between nodes, capacities can be thought of as being zero.

For any given route, the capacity of the route is defined to be the minimum of the individual link capacities in the route. Although this problem is different from the shortest-route problem,[2] all of the known computational algorithms for the shortest-route problem can be modified to solve the maximum capacity route problem. In the shortest-route problem, the distance of any given route is the summation of the distances of the individual links. The shortest-route algorithms are thus all based on the following relation. Assume the distance is known from an origin node A to a node B along some route R. Then the distance from node A to C, along the route R and the link BC is equal to the sum of the known distance from A to B and the distance of link BC. The analogous relation for the capacity of a route is as follows. Assume the capacity is known from an origin node A to a node B along some route R. Then the capacity from node A to C, along the route R and the link BC, is equal to the minimum of either the known capacity from A to B or the capacity of link BC.

The four shortest-route algorithms given in reference 2 modified to solve the maximal capacity route problem are given below. Neither the details nor the relative merits of the algorithms will be dealt with here, since the comments would be entirely similar to those given in reference 2. The reader should refer to this reference if he wishes to apply any of them.

The solution described by DANTZIG is modified as follows. Starting with the initial node an arbitrary tree is chosen. Assign to the initial node the capacity 'infinity.' To each node a value is assigned equal to the minimum value of either the preceding node or the capacity of the link joining the preceding node to the node in question. Links not included in the initial tree are introduced one at a time and included in the tree if any node value is increased. Whenever a new link is added to the tree some other link must be removed. This process continues until no further increase in any node value is possible.

The algorithm by MINTY is modified as follows:

1. Label the city A with the capacity 'infinity.' Go to 2.

2. Look at all one-way streets with their 'tails' labeled and their heads unlabeled. For each such street, determine the minimum of the label and the street capacity. Select a street having the maximum value of the minima just found and place a check-mark beside it. Label the 'head' city with this maximum. Return to the beginning of 2.

The algorithm by MOORE using the index number suggested by D'ESOPO is modified as follows.

1. Assign to city A the index 1 and the label 'infinity.' Enter the number 1 in the control register. Go to 2.

2. Locate all nodes connected to the CR-node. To each of these nodes that do not already have index numbers, assign the next available integer as its index. Go to 3.

3. For each node connected to the CR-node, determine the minimum value of either the CR-node label or the street capacity to the connected node. For each connected node there will be three possibilities. (a) If the connected node does not have a label, assign this minimum value as its label and place a check-mark next to the street. Proceed to the next connected node or, if there are no more connected nodes to be examined, go to 4. (b) If the connected node has a previously assigned label that is higher in value than the determined minimum, do nothing to this node. Proceed to the next connected node or, if there are no more connected nodes to be examined, go to 4. (c) If the connected node has a previously assigned label that is smaller than the new minimum, assign the new minimum as the label. Also place a check-mark beside the new street and erase the check-mark that is next to the street associated with the replaced label. Now examine the index of this relabeled node. If the index is higher than the index of the CR-node, proceed to the next connected node or, if none remain, go to 4. If the index is lower than the index of the CR-node, note the value of this lower index number. Proceed to the next connected node or, if none remain, go to 4.

4. If there are any lower index numbers noted in step 3(c), select the lowest one and place it in the control register. If not, add 1 to the number in the control register. In either case go to 2.

The matrix method based on results of SHIMBEL and BELLMAN is modified as follows. Let the capacity from node j to k be represented by c_{jk}^1, an element of the matrix C^1, where j, $k=1, 2, \cdots, N$, and $c_{jj}^1=\infty$ for all values of j. An element of the matrix C^2 is defined as

$$c_{jk}^2=\max_l\,[\min\,(c_{jl}^1, c_{lk}^1)], \qquad (l=1, 2, \cdots, N)$$

so that c_{jk}^2 is the maximum capacity route between j and k by means of a one- or two-link path. The following relation may be used to generate successive matrices.

$$c_{jk}^{m+1}=\max_l\,[\min\,(c_{jl}^1, c_{lk}^m)]. \qquad (l=1, 2, \cdots, N)$$

Here one starts with C^1 and successively obtains C^2, C^3, C^4, etc. The process terminates either when $C^{m+1}=C^m$ or when $m+1=N-1$. It is also true that

$$c_{jk}^{2m}=\max_l\,[\min\,(c_{jl}^m, c_{lk}^m). \qquad (l=1, 2, \cdots, N)$$

When the faster doubling method is used, one starts with C^1 and successively obtains C^2, C^4, C^8, etc. The process terminates either when $C^{2m}=C^m$ or when $2m\geqq N-1$.

To determine the routes associated with the capacities, the R matrix method described in reference 2 can be used as given. An alternative method of determining routes, which uses less storage, is similar to the alternative given in reference 2. To determine the maximum capacity route from j to k first determine from C^{2m} the value of node k and the values of all other nodes that are connected from node k by one link. From C^1 find the link capacities that connect these nodes to node k. From the set of adjacent nodes, eliminate from consideration all nodes whose connected links have a lower capacity than the value of node k. From the remaining set of node values, find a node whose value is not less than the value of node k. This node is the next to last node in a maximum capacity route from

j to k. The process is now repeated for the next to last node, and so on, until one works back to the initial node j.

Rather than finding the maximum capacity route, it may be sufficient in some problems to determine whether there exists a route, between some node i and j, having a capacity greater than or equal to some value V. This question is easily answered if one eliminates from the network all links having capacities less than V. If nodes i and j are still connected after these links are removed, then any route through the remaining network will have a capacity equal to or greater than V. This last relation is interesting since nothing analogous to it exists for the shortest-route problem.

REFERENCES

1. L. R. Ford, Jr. and D. R. Fulkerson, "Maximal Flow Through a Network," *Canadian J. Math.* **8,** 399–404 (1956).
2. M. Pollack and W. Wiebenson, "Solutions of the Shortest-Route Problem—A Review," *Opns. Res.* **8,** 224–230 (1960).