MATHEMATICS OF OPERATIONS RESEARCH
Vol. 4, No. 3, August 1979
Printed in U.S.A.

ON FULKERSON'S CONJECTURE ABOUT CONSISTENT LABELING PROCESSES*

NIMROD MEGIDDO† AND ZVI GALIL‡

Fulkerson conjectured that any consistent labeling process implies a polynomial time bound for the Ford-Fulkerson max-flow algorithm. The conjecture is disproved by means of a sequence of networks requiring an exponential number of augmentations, resulting from a consistent labeling process. On the other hand, it is shown that any consistent labeling process yields an algorithm which runs in time $O[\mathrm{Min}(V!, 3^E)]$. This result is stronger than A. Tucker's which states that consistency implies a finite number of augmentations.

Ford and Fulkerson's method of solving the max-flow problem is by repeated augmentations of flow, from the source s to the sink t. Augmentation is carried out along a path which is found by a labeling process. First, the source is labeled. Second, if x is a labeled vertex and y is unlabeled, then y can receive a label from x either if the edge (x, y) is not saturated, or if the edge (y, x) carries positive flow. See [3] for details. A consistent labeling process is one in which the choice of the labeled node x to be scanned next is determined only by the set of nodes that are currently labeled and unscanned. Then all unlabeled neighbors of x receive a label from x if possible (see [4]).

Following is a summary of known facts.

1. If the labeling process halts and t is unlabled, then the current flow is maximal.
2. If the edge capacities are rational numbers, then a max-flow is found in a finite number of flow-augmentations; this number is not bounded by any function of the size of the network.
3. If the capacities are real numbers, then the procedure might never halt, and furthermore, it might produce a convergent sequence of flows, whose limit is not a max-flow.
4. If the labeling process is consistent, then a max-flow is found in a finite number of flow-augmentations [4]. Tucker has not shown this number to be bounded by any function of the size of the network.
5. If the labeling process is "first-labeled first-scanned" (i.e., breadth-first-search) then a max-flow is found in $O(V^3)$ flow-augmentations [1], [2]. Efficient max-flow algorithms based on this property have been developed recently.

Fulkerson conjectured that consistency of the labeling implies that the number of augmentations is bounded by a polynomial function of the number of vertices. We will disprove Fulkerson's conjecture, but first we modify the definition of consistency —just for the ease of presentation. A consistent (in a broad sense) labeling process progresses as follows. First, the source is labeled. Next, select a labeled vertex x and an unlabeled vertex y, and label y from x if possible. The selection of the pair (x, y) is determined by the set of presently labeled vertices and by the set of pairs tested (without success) for possible labeling so far. A consistent labeling process in the broad sense is equivalent to a consistent labeling process in the narrow sense, but in a modified network, obtained by adding a vertex in the middle of each edge.

* Received April 17, 1978.
AMS 1970 subject classification. Primary 90C35.
IAOR 1973 subject classification. Main: Network programming.
Key words. Max-flows, flow-augmentation algorithm, exponential running time.
† Northwestern University and Tel Aviv University.
‡ Tel Aviv University.

0364-765X/79/0403/0265$01.25

Let G_n be the network shown in figure 1:

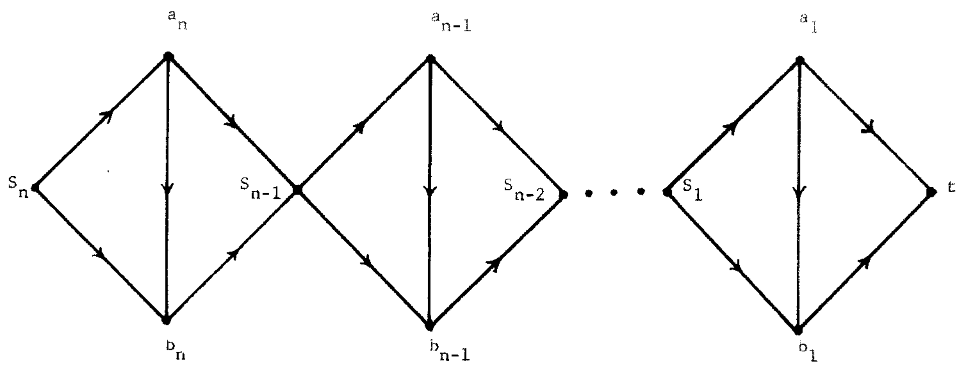

FIGURE 1

The capacities are $c(a_k, b_k) = 2^{k-1}$ $(k = 1, \ldots, n)$ and $c(e) = 2^n$ for every other edge e. The labeling processes introduced below are not well defined for all possible flows in G_n, but are well defined for the particular flows encountered later. The definitions are recursive.

Procedure P_1
begin label s_1; label a_1 from s_1;
if b_1 cannot be labeled from a_1 *then* label t from a_1 *else*
[label b_1 from a_1; label t from b_1]; *end.*

Procedure Q_1
begin label s_1; label b_1 from s_1;
if a_1 cannot be labeled from b_1 *then* label t from b_1 *else*
[label a_1 from b_1; label t from a_1]; *end.*

For $n \geqslant 2$ we define

Procedure P_n
begin label s_n; label a_n from s_n;
if b_n cannot be labeled from a_n *then* [label s_{n-1} from a_n; Q_{n-1}]
else [label b_n from a_n; label s_{n-1} from b_n; P_{n-1}];
end.

Procedure Q_n
begin label s_n; label b_n from s_n;
if a_n cannot be labeled from b_n *then* [label s_{n-1} from b_n; P_{n-1}]
else [label a_n from b_n; label s_{n-1} from a_n; Q_{n-1}]; *end.*

Obviously, P_n and Q_n are consistent in the broad sense. The following proposition can be easily proved by induction on n.

PROPOSITION. (i) *Starting with flow zero, procedure P_n generates in 2^n augmentations the following flow*: $f(s_n, a_n) = 2^n$, $f(s_n, b_n) = 0$, $f(a_k, b_k) = 0$ *for* $k = 1, 2, \ldots, n-1$, *and* $f(e) = 2^{n-1}$ *for every other edge e.* (ii) *Starting from the flow f, procedure Q_n generates in 2^n augmentations the following flow*: $f^*(a_k, b_k) = 0$ *for* $k = 1, \ldots, n$, *and* $f^*(e) = 2^n$ *for every other edge e.*

Thus, consistent labeling may result in an exponential number of flow augmentations, even if capacities are integers. However, we will show that the number of flow augmentations, subject to consistent labeling, is bounded for networks with a fixed number of vertices.

Given a flow f in a network $G = (V, E)$, let $S_1 = \{e \in E : f(e) = 0\}$, $S_2 = \{e \in E : 0 < f(e) < c(e)\}$, and $S_3 = \{e \in E : f(e) = c(e)\}$. The ordered triple (S_1, S_2, S_3) is called the *state* of the flow f. Obviously, there can be at most 3^E distinct states of flow in G. A consistent labeling process generates a sequence of flows with distinct states (see a proof below) and hence cannot generate more than 3^E flows.

LEMMA. *If $f^1, f^2, \ldots$ is a sequence of flows in G, produced by a consistent labeling process, then the states of these flows are pairwise distinct.*

PROOF. Suppose, on the contrary, that there are two flows f^i, f^j $(i < j)$ having the same state. A consistent labeling process will progress identically with respect to both of them. Obviously, f^i and f^{i+1} have distinct states. It follows that there is a pair (x, y) such that y receives a label from x while f^i and f^j are augmented, but y does not receive a label from x while some f^k $(i < k < j)$ is augmented. Without loss of generality, assume that y is the first (in the order of the labeling w.r.t. f^i and f^j) with the property that y participates in such a pair (x, y). Thus, for every l $(i \leqslant l \leqslant j)$ the labeling process progresses identically up to the point where y is considered for receiving a label from x. Since y cannot be labeled from x w.r.t. f^k, it follows that $f^k(x, y) = c(x, y)$ and $f^k(y, x) = 0$. Since x is always labeled before y, x will never receive a label from y. Thus, the flow in (x, y) and (y, x) will never change beyond this point. In particular, $f^j(x, y) = c(x, y)$ and $f^j(y, x) = 0$, so that y cannot receive a label from x w.r.t. f^j. Hence, a contradiction.

Another upper bound on the number of flow-augmentations subject to consistent labeling, namely $V!$, could be derived by induction on V as follows.

Let x be the first vertex that receives a label from s (starting from flow zero). As long as the edge (s, x) is not saturated, x will be the first to receive a label from s. All augmentations made before (s, x) is saturated can be interpreted as being made in the network obtained by contracting the edge (s, x). Hence, there can be at most $(V - 1)!$ of them. If (s, x) never becomes saturated, then the proof is complete. If (s, x) does become saturated, then it will remain saturated forever, and x will never again receive a label from s. Let y be the first vertex that receives a label from s, immediately after (s, x) becomes saturated. A similar argument implies that there can be at most $(V - 1)!$ augmentations since (s, x) has become saturated and before (s, y) becomes saturated. Proceeding in this line, we deduce that there can be at most V phases with at most $(V - 1)!$ augmentations in each one of them. Hence, an upper bound of $V!$ follows.

References

[1] Dinic, E. A. (1970). Algorithm for Solution of a Problem of Maximum Flow in a Network with Power Estimation. *Soviet Math. Dokl.* **11** 1277–1280.

[2] Edmonds, J. and Karp, R. (1972). Theoretic Improvements in Algorithmic Efficiency for Network Flow Problems. *J. Assoc. Comput. Mach.* **19** 248–264.

[3] Ford, L. and Fulkerson, D. R. (1962). *Flows in Networks*. Princeton University Press, Princeton, N.J.

[4] Tucker, A. (1977). A Note on Convergence of the Ford-Fulkerson Flow Algorithm. *Math. Oper. Res.* **2** 143–144.

MEGIDDO: GRADUATE SCHOOL OF MANAGEMENT, NORTHWESTERN UNIVERSITY, EVANSTON, ILLINOIS 60201

DEPARTMENT OF STATISTICS, TEL AVIV UNIVERSITY, TEL AVIV, ISRAEL

GALIL: DEPARTMENT OF MATHEMATICAL SCIENCES, DIVISION OF COMPUTER SCIENCE, TEL AVIV UNIVERSITY, TEL AVIV, ISRAEL