A Note on Yen's Algorithm for Finding the Length of All Shortest Paths in N-Node Nonnegative-Distance Networks

THOMAS A. WILLIAMS

AND

GREGORY P. WHITE

University of Cincinnati, Cincinnati, Ohio

ABSTRACT. An error in Yen's algorithm is pointed out, and an alternative is offered which will produce correct results.

KEY WORDS AND PHRASES: shortest network paths, graph and network theory

CR CATEGORIES: 5.31, 5.32, 5.40, 8.3

In [1], Yen presents a (computer programming) algorithm for finding the lengths of all shortest paths in N-node nonnegative-distance complete networks. Any attempt to program the algorithm as described will produce incorrect results in all but the most specialized cases. The purpose of this note is to point out the mistake in Yen's algorithm, and to offer an alternative which will produce correct results. Given the suggested alternative one can then verify the timing properties of Yen's algorithm.

To demonstrate the problem in Yen's algorithm, we consider a simple illustration involving the network shown in Figure 1.

Applying step I of Yen's algorithm, we set $L = 1$, $K = 4$, $F(1) = 0$, $H(I) = I$, $I = 1, 2, 3, 4$, and $F(I) = \infty$, $I = 2, 3, 4$. Then, in step II, we find that the minimum value of $F(J)$ is 7. This corresponds to $J = 2$; hence, $J^* = 2$. At step III, we set $L = J^* = 2$ and $H(J^*) = H(K) = 4$. Finally, K is reduced to 3 in step IV; therefore, we return to step II.

At the second iteration, we find that $F(J^*) = 8$. This corresponds to $J^* = 4$ (note at this iteration $F(1) = 0$, $F(2) = 7$, $F(3) = 12$, and $F(4) = 8$). Applying step III, we then set $L = 4$ and $H(4) = 3$. Then, in step IV, K is reduced to 2, and hence we return again to step II.

After applying steps II–IV at the third iteration, we find that K is reduced to 1. Thus, we now should have the minimum distance from node 1 to nodes 2, 3, and 4. However, the results given by the algorithm are $F(1) = 0$, $F(2) = 7$, $F(3) = 12$, and $F(4) = 8$. Clearly, $\mathrm{F}(3) = 12$ is incorrect, since by inspection the path that goes from node 1 to node 3 through node 4 yields the correct minimum distance of 10. Thus, the algorithm as stated will produce incorrect results.

The essential problem in the algorithm lies in step III, wherein $H(J^*)$ is incorrectly set to $H(K)$. To correct this problem, consider the following modification of steps II and III.

Authors' address: Department of Quantitative Analysis, University of Cincinnati, Cincinnati, OH 45221.

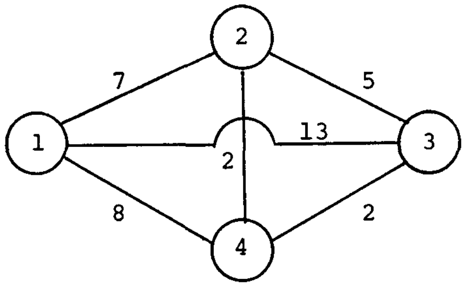

FIG. 1

II. For $I = 2, 3, \cdots, K$, do steps A, B, and C as follows:
   A. Let $J = H(I)$.
   B. Compute $F(J) = \min [F(J), F(L) + D(L, J)]$.
   C. If the value of $F(J)$ is less than the current minimum during this execution of step II, say $F(J^*)$, replace J^* by J and note the corresponding value of I by setting $I^* = \mathrm{I}$.

III. Label $L = J^*$ and $H(I^*) = H(K)$.

A version incorporating these two changes has been programmed in FORTRAN IV and run successfully on an IBM 360/65 at the University of Cincinnati.

REFERENCE

1. YEN, J Y., Finding the lengths of all shortest paths in N-node nonnegative-distance complete networks using $\frac{1}{2}N^3$ additions and N^3 comparisons. *J. ACM 19*, 3 (July 1972), 423–424.

RECEIVED AUGUST 1972; REVISED NOVEMBER 1972

Reply to Williams and White's Note

By Jin Y. Yen

I appreciate that Williams and White, as well as many other readers, have pointed out an error in my paper. The problem lies in an error in step III, in which I have used $H(J^*)$ inappropriately to denote "the location in H in which the node that produced the minimum $F(J)$ is stored." My paper is correct if this point is explained.

The following is a FORTRAN IV program for the algorithm.

```
      PROGRAM SP (INPUT, OUTPUT)
      INTEGER D(100,100), H(100), F(100), X
      ND=100
      L=1
      READ 5,D
    5 FORMAT(1015/)
      NDP2=ND+2
      DO 10 I=1,ND
      H(I)=I
      F(I)=D(L,1)
   10 CONTINUE
      H(L)=1
      H(1)=L
      DO 20 KK=2,ND
      K=NDP2-KK
 (Continued in second column)
      X=9999999999
      DO 30 JJ=2,K
      J=H(JJ)
      IF (F(L)+D(L,J).GE F(J)) GO TO 40
      F(J)=F(L)+D(L,J)
   40 IF (X.LE.F(J)) GO TO 30
      X=F(J)
      LL=J
      JH=JJ
   30 CONTINUE
      L=LL
      H(JH)=H(K)
   20 CONTINUE
      PRINT 50,F
   50 FORMAT (1015/)
      STOP
```