SOLVING FLOW-CONSTRAINED NETWORKS: INVERSE PROBLEM[a]

Discussion by Pramod R. Bhave[3]

Altman and Boulos deserve congratulations for their useful contribution on water distribution networks. They have provided necessary and sufficient conditions for solvability of inverse types of problems in which pipe-system characteristics can be treated as variables and determined to meet designated flow and/or pressure specifications. The necessary condition is that the number of unknown parameters p must be equal to the number of flow constraints c, which, in turn, must not exceed the total number of loops (real and pseudo) in the network; i.e., $e - n$; giving $p = c \leq e - n$. The sufficient condition is that only edges not lying on a junction-spanning tree, i.e., non-JST edges, can be flow-constrained.

These necessary and sufficient conditions developed by the authors apply to situations wherein only the flow-constrained edges can be with unknown characteristics. This type of inverse problem, as briefly indicated elsewhere by the discusser (Bhave 1991, p. 139) can be easily converted to a forward problem. An edge with a targeted flow rate but with an unknown characteristic is removed from the network (this simultaneously removes a known condition and an unknown parameter); the targeted flow rate is treated as additional inflow and outflow at the downstream and upstream nodes of the removed edge, respectively; and the consumptions at these nodes are modified accordingly. Thus, if the initial forward problem is solvable, the inverse problem, converted to a modified forward problem, is also solvable. This procedure of removing a flow-constrained edge with an unknown characteristic can be continued (this always maintains $p = c$) until no part of the modified network gets isolated without any fixed-grade node. (This leads to the necessary and sufficient conditions.) After analyzing the modified network as a forward problem, since the heads at the end nodes of the removed edges are now known, the unknown edge characteristics can be determined. In the modified forward problem, the number of unknowns is $e - p$ instead of $e + p$ as in the method suggested by the authors; thus, the analysis becomes simpler to that extent.

In the illustrative example of Fig. 1, edges 3 and 4 are with unknown characteristics and with targeted flow rates of 600 and 200 L/s, respectively. Therefore, on removal of edges 3 and 4, the modified external demands are $q_1 = 60 + 200 = 260$ L/s, $q_3 = 20 - 200 = -180$ L/s, and $q_2 = 80 + 600 = 680$ L/s. In this illustrative problem, since $p = c = e - n$, the modified network is a branching one. Therefore, the unknown edge flows, the unknown nodal heads, and subsequently the unknown edge characteristics (diameters) for edges 3 and 4 can be determined directly, without resorting to any iterative procedure such as the Newton-Raphson method.

The logic can be extended to networks wherein flow-constrained edges and edges with unknown characteristics are different. In the forward problem, for a network with e edges we get a set of e independent Q equations. Therefore, any edge from a loop (real or pseudo, basic or overlapping) to which this flow-constrained edge belongs can be made with an unknown characteristic to keep the number of unknowns (unknown edge characteristics and unknown edge flows) the same as the number of independent equations, e. Therefore this procedure of introducing flow-constrained edges and edges with unknown characteristics can be continued until the network

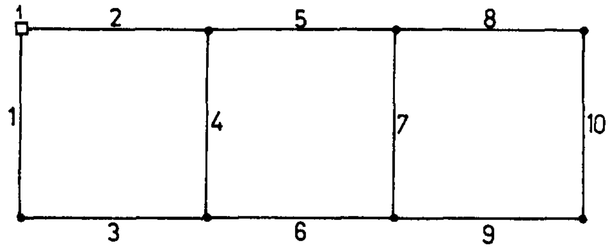

FIG. 2. Illustrative Network

does not contain any loop. This leads to the necessary condition $p = c \leq e - n$. To avoid concentration of flow-constrained edges and the edges with unknown characteristics, the flow-constrained edges should be non-JST edges, as should the edges with unknown characteristics. These two JSTs are different but may be identical if all the flow-constrained edges are with unknown characteristics.

The procedure for checking the solvability of networks with flow-constrained edges and edges with unknown characteristics is as follows:

1. Verify that $p = c \leq e - n$.
2. Verify that all flow-constrained edges and edges with unknown characteristics are non-JST edges.
3. Prepare a list of flow-constrained edges. Remove these flow-constrained edges from the network to get a reduced network. (This reduced network is looped if $c < e - n$, or is a JST if $c = e - n$.)
4. Verify that there exists at least one flow-constrained edge that, when appended to the reduced network, forms a loop (real or pseudo, basic or overlapping) that contains only one edge with an unknown characteristic.
5. Delete this flow-constrained edge from the list and treat the unknown edge characteristic as a known one.
6. Repeat steps 4 and 5, if necessary, until all flow-constrained edges are considered.

If these steps can be carried out, the network is solvable; otherwise it is not.

As an illustration, consider the network of Fig. 2 with node 1 as a fixed-grade node. The edges are labeled as shown in the figure. This network is solvable in the following situations: (1) Any edge is flow-constrained and any edge is with an unknown characteristic; (2) edges 2 and 4 are flow-constrained, and edges 5 and 7 are with unknown characteristics; (3) edges 2, 5, and 7 are flow-constrained; edges 4, 7, and 9 are with unknown characteristics; and so forth. However, if edge 5 is absent in this network, the unknown characteristic of any edge from 6 to 10 cannot be determined when the flow in edge 3 is constrained.

In the network of Fig. 1, when edges 3 and 4 are flow-constrained as in the illustrative example, the unknown characteristic of any edge from 1 to 3, together with an unknown characteristic of an edge from edges 2, 4, and 5 (edge 2 not repeating), can be determined.

The network-solvability rules and procedure are heuristically derived herein and lack rigorous mathematical proof.

Closure by Tom Altman[4] and Paul F. Boulos,[5] Member, ASCE

We thank the discusser for his interest in the paper. The discusser proposes an alternative procedure for solving flow-

[a]May 1995, Vol. 121, No. 5, by Tom Altman and Paul F. Boulos (Technical Note 8362).

[3]Engrg. Consultant, 201 Utkarsha-Vishakha, 42 Bajaj Nagar, Nagpur 440 010, India.

[4]Assoc. Prof., Dept. of Comp. Sci. and Engrg., Univ. of Colorado at Denver, 1200 Larimer St., Denver, CO 80217.

[5]Dir., Water Distribution Technology, Montgomery Watson, 300 North Lake Ave., Ste. 1200, Pasadena, CA 91101.

J. Hydraul. Eng. 1997.123:252-253.

constrained networks wherein the flow-constrained edges and the edges with unknown characteristics may be different. For the most part, the discusser's method is similar to the writers with the exception of the necessary condition for securing the problem solvability. The fundamental difference is due to the fact that both methods are characteristically opposite. While the writers' method maintains the original network structure unchanged and directly (explicitly) solves the inverse problem, the discusser's method first converts the inverse problem to a modified forward problem by removing all flow-constrained edges and altering the demands at the affected nodes, and then solves the reduced network in a forward fashion. In the modified forward problem, the number of unknowns becomes $e - p$ instead of $e + p$. The unknown edge characteristics are then determined based on the calculated heads at their two endpoints.

Both methods are relatively simple to understand and implement, and are geared toward solving practical network-hydraulic engineering applications. The writers agree with the discusser that for the purpose of completeness, it would be desirable if the discusser's heuristic is supplemented with a rigorous mathematical proof [see Boulos and Altman (1993)]. In addition, the discusser's heuristic, as presented, requires the user to exercise special care during the removal of the non-JST edges that may cause the network (some fixed-grade nodes) to be disconnected. In particular, the discusser's suggestion of removing edge 3 in the illustrative example of Fig. 1 will result in fixed-grade node 6 being disconnected (degree 0) from the rest of the network.

Errata. The following correction should be made to the original paper: On page 430, Table 3 should read

TABLE 3. Computational Results

Edge number (1)	Flow rate (L/s) (2)	Velocity (m/s) (3)	Diameter (mm) (4)	Junction number (5)	Piezometric head (m) (6)
1	800.00	1.59	—	1	247.04
2	540.00	4.30	—	2	205.29
3	600.00	1.89	636.4	3	208.71
4	200.00	3.27	279.0	4	208.38
5	140.00	1.11	—	—	—
6	40.00	0.32	—	—	—

APPENDIX. REFERENCE

Boulos, P. F., and Altman, T. (1993). "An explicit approach for modelling closed pipes in water networks." *J. Appl. Math. Modelling*, 17(8), 437–443.

APPLICATION OF NEURAL NETWORKS IN STRATIFIED FLOW STABILITY ANALYSIS[a]

Discussion by Yacoub M. Najjar[2] and Imad A. Basheer[3]

The author is acknowledged for the paper he presented to predict the flow condition when interfacial mixing in stratified estuaries occurs. The use of neural networks to predict the condition of stratification was shown to be an efficient tool. While the discussers encourage the use of the feed-forward back-propagation neural network (BPNN) to model a wide variety of problems in various fields in science and engineering [e.g., Basheer and Najjar (in press, 1996); Basheer et al. 1996], they have some comments regarding the author's use of the BPNN to model the stability condition of the flow.

First, the author developed a network with two input nodes and one output node using a large architecture composed of two hidden layers, each containing 10 nodes. Because of the small number of input and output nodes, the discussers ask whether the author has tried to model the problem using conventional regression methods. Second, the developed networks require one output to represent the condition of the flow, whether it is stable or unstable. Therefore, the problem is of the classification type where discrete inputs are used to train the network and discrete outputs are to be expected from prediction. The author may have used the output vector as 1 for stable flow and 0 for unstable flow. After training the network, normally, an error cannot be avoided, which would tend to deviate the actual output values from their corresponding prediction values (even for the data used in training). Hence, the output cannot always be guaranteed to take a value of either 0 or 1, and real values between 0 and 1 can be expected. The author does not mention, however, how he tackled this problem. For instance, did the author use a modified version for the BPNN where the outputs at each training cycle are adjusted such that the output of each data set is allowed to take only integer values? Or have the predicted outputs been adjusted after convergence (i.e., the last training epoch) to take a 0 or 1 value by using a certain discretization criterion?

APPENDIX. REFERENCE

Basheer, I. A., Reddi, L. N., and Najjar, Y. M. (1996). "Site characterization using neuronets: an application to the landfill siting problem." *Ground Water*, 34(4), 610–617.

Closure by John P. Grubert,[4] Fellow, ASCE

The writer would like to thank Najjar and Basheer for their interest in the paper. The paper was actually condensed from an MSc thesis in computer science that I wrote (Grubert 1994); due to shortage of space, I could not explain everything in detail.

The neural calculations were done using the BrainMaker shell, and its "Edit Network Input" facility was used to obtain trained values of raw data so that they could be converted into Keulegan stability parameter (K) and Reynolds number (R) data for plotting purposes. This was achieved by setting values of $\mathrm{Log}(g')$ and $\mathrm{Log}(R_1)$ then varying $\mathrm{Log}(V)$ by trial and error until the output pattern gave 0.50 for both S and U. With a full range of these results, the logarithmic equations for Keulegan stability parameter (K) and Reynolds number (R), shown in (7) and (8), respectively

$$\mathrm{Log}(K) = 3\,\mathrm{Log}(V) - \mathrm{Log}(g') - \mathrm{Log}(\nu_1) \quad (7)$$

$$\mathrm{Log}(\mathsf{R}) = \mathrm{Log}(V) + \mathrm{Log}(R_1) - \mathrm{Log}(\nu_1) \quad (8)$$

were used to plot the curves since $K = V^3/(g'\nu_1)$ and ($\mathsf{R} = VR_1/\nu_1$. In these calculations, the kinematic viscosity ν_1 was constant.

When this work was started in 1992, conventional multiple

[a]July 1995, Vol. 121, No. 7, by John P. Grubert (Paper 8796).

[2]Asst. Prof., Dept. of Civ. Engrg., Kansas State Univ., Manhattan, KS 66506.

[3]PhD Candidate, Dept. of Civ. Engrg., Kansas State Univ., Manhattan, KS.

[4]Sr. Lect., Dept. of Civ. Engrg., Univ. of Glamorgan, Pontypridd, Mid Glamorgan, CF37 1DL, UK.

J. Hydraul. Eng. 1997.123:252-253.