Acknowledgments: Special thanks to P. M. Smith and U. Mishra for providing the HEMT devices used in these experiments.

J. J. KOMIAK 12th May 1986

General Electric Company
Electronics Laboratory
Electronics Park, Syracuse, NY 13221, USA

SECOND-DERIVATIVE ROUTING ALGORITHMS

Indexing terms: Computers, Communication networks, Computer communications, Routing algorithms

An approximation of Newton's method has been used recently in routing algorithms in computer networks, which is based on the assumption that the matrix of second derivatives of the objective function (Hessian) is diagonal. It is demonstrated that in message-switched networks some specific nondiagonal elements exist and therefore the Hessian is not diagonal. Moreover, it is found that these elements are of the same order as the diagonal ones.

Introduction: There is a strong interest in routing algorithms in computer networks, and especially in algorithms based on methods of nonlinear programming which use second-order derivatives of the objective function; this is because of the higher speed of convergence and better accuracy of the results which are achieved. The most attractive and preferred method seems to be Newton's method or some adequate approximation of it, in order to reduce the large amount of computations required in this kind of problems.[1–3]

A difficulty in Newton's method, however, is the construction and inversion of the Hessian matrix. Since the mathematical formula which gives the second derivatives of the function used in routing problems is very complex,[2] there is a tendency to consider the Hessian to be diagonal. On the contrary, it is shown that in message-switching networks, where the above-mentioned algorithms were applied, this is not true; this is because there is a linear relationship among some of the independent variables at each node of the network. Therefore, the second-order derivatives, taking the variables in pairs, are not always equal to zero and consequently the Hessian is not diagonal. In this letter we find that these nondiagonal elements do exist and they are calculated.

Mathematical analysis: In a network consisting of N nodes (1, 2, 3, …, N) and L directed links it is assumed that the message arrival processes are independent and Poisson-distributed, whereas the message lengths are exponentially distributed.[4] Let the directed link connecting the i and l nodes be denoted by $(i, l) \in L$. Furthermore, in our analysis the following variables and constants are involved: (i) $f_{il(j)}$ is the flow in link (i, l) destined to node j (messages/s), $0 \leq f_{il}$, $\forall(i, l) \in L$ and $f_{il} = 0$ if $i = l$; (ii) C_{il} is the constant capacity of link (i, l) (bit/s), $0 < C_{il}$, $\forall(i, l) \in L$ and $C_{il} = 0$ if $i = l$; (iii) $\gamma_{i(j)}$ is the proper traffic entering the network at node i and destined to node j (messages/s), $\gamma_i \geq 0$, $\forall i \in N$; (iv) $t_{i(j)}$ is the total flow at node i destined to j (messages/s), $t_i \geq 0$, $\forall i \in N$; (iv) $\varphi_{il(j)}$ is the fraction of $t_i(j)$ routed through link (i, l), $0 \leq \varphi \leq 1$, $\forall(i, l) \in L$; (vi) $1/\mu$ is the average message length (bit/message); (vii) $\gamma = \sum_i^N \sum_j^N \gamma_{i(j)}$ is the total arrival flow in the network (messages/s).

The above variables obey the following relationships and conditions:

$$f_{il(j)} = t_{i(j)}\varphi_{il(j)} \tag{1}$$

and

$$F_{il} = \sum_j f_{il(j)} \qquad 0 \leq F_{il} \leq C_{il} \qquad \forall(i, l) \in L \tag{2}$$

$$t_{i(j)} = \gamma_{i(j)} + \sum_l f_{li(j)} = \gamma_{i(j)} + \sum_l t_{l(j)}\varphi_{li(j)} \tag{3}$$

$$\sum_i \varphi_{li(j)} = 1 \tag{4}$$

The criterion of performance is the average message delay, which is given by[4]

$$D = \frac{1}{\gamma}\sum_{(i,l)} \frac{F_{il}}{\mu C_{il} - F_{il}}$$

For notational convenience we will suppress the destination index and concentrate our analysis on a single destination. Our definitions and the mathematical analysis are essentially indentical for each destination, so this notational simplification should not become a source of confusion.

The criterion of performance then becomes

$$\begin{aligned} D &= \frac{1}{\gamma}\sum_{(i,l)} \frac{f_{il}}{\mu C_{il} - f_{il}} \\ &= \frac{1}{\gamma}\sum_{(i,l)} \frac{t_i\varphi_{il}}{\mu C_{il} - t_i\varphi_{il}} \end{aligned} \tag{5}$$

To minimise eqn. 5 with respect to the independent variables φ_{il}, the first and second derivatives must be considered. The first derivatives of D are given by[5]

$$\frac{\partial D}{\partial \varphi_{il}} = t_i\left(D'_{il} + \frac{\partial D}{\partial \gamma_l}\right) \tag{6}$$

$$\frac{\partial D}{\partial \gamma_i} = \sum_l \varphi_{il}\left(D'_{il} + \frac{\partial D}{\partial \gamma_l}\right) \tag{7}$$

where D'_{il} is the first derivative of D with respect to f_{il}, $i \neq l$.

On the other hand, the second derivatives of D are the following:[2]

$$\frac{\partial^2 D}{\partial(\varphi_{il})^2} = t_i^2(D''_{il} + R_l) \tag{8}$$

$$\frac{\partial^2 D}{\partial \varphi_{il}\,\partial \varphi_{il'}} = 0 \tag{9}$$

where D''_{il} is the second derivative of D with respect to f_{il}, and R_l is given by

$$\begin{aligned} R_l &= \left(\frac{\partial^2 D}{\partial \varphi_{il}\,\partial \gamma_l}\right)\Big/ t_i \\ &= \sum_M (\varphi_{lm})^2 D''_{lm} + \sum_{m'} (\varphi_{lm'}\delta_{lm'}) \\ &\quad \times \sum_m \left(\frac{\varphi_{lm}}{\delta_{lm}} R_m\right) \end{aligned} \tag{10}$$

where $\delta_{lm} = D'_{lm} + (\partial D/\partial \gamma_m)$, and m and m' are downstream neighbour nodes from node l.

Now, it is demonstrated here that the term $\partial^2 D/\partial \varphi_{il}\partial \varphi_{il'}$ in eqn. 9 is not always equal to zero. Assuming that node i has two outgoing links l and l', eqn. 4 becomes

$$\varphi_{il} + \varphi_{il'} = 1 \tag{11}$$

It then follows that

$$\frac{\partial^2 D}{\partial \varphi_{il}\,\partial \varphi_{il'}} = \frac{\partial}{\partial \varphi_{il}}\left[t_i\left(D'_{il'} + \frac{\partial D}{\partial \gamma_{l'}}\right)\right] \tag{12}$$

Considering that the t_i is independent of φ_{il}, eqn. 12 becomes

$$\frac{\partial^2 D}{\partial \varphi_{il}\,\partial \varphi_{il'}} = t_i \frac{\partial}{\partial \varphi_{il}}\left[D'_{il'} + \frac{\partial D}{\partial \gamma_{l'}}\right] \tag{13}$$

The first term of eqn. 13 is

$$\begin{aligned} \frac{\partial D'_{il'}}{\partial \varphi_{il}} &= \frac{\partial D'_{il'}}{\partial f_{il'}}\,\frac{\partial f_{il'}}{\partial \varphi_{il}} \\ &= D''_{il'}\,\frac{\partial f_{il'}}{\partial \varphi_{il}} \end{aligned} \tag{14}$$

From eqns. 1 and 11 we have

$$f_{il'} = t_i \varphi_{il'} = t_i(1 - \varphi_{il}) \qquad (15)$$

and therefore eqn. 14 becomes

$$\frac{\partial D'_{il'}}{\partial \varphi_{il}} = -t_i D''_{il'} \qquad (16)$$

On the other hand, the second term of eqn. 13 is

$$\begin{aligned}\frac{\partial}{\partial \varphi_{il}} \frac{\partial D}{\partial \gamma_{l'}} &= \frac{\partial}{\partial \varphi_{il'}} \left[\frac{\partial \varphi_{il}}{\partial \varphi_{il'}}\right]^{-1} \frac{\partial D}{\partial \gamma_{l'}} \\ &= \frac{\partial}{\partial \varphi_{il'}} \left[\frac{\partial (1 - \varphi_{il'})}{\partial \varphi_{il'}}\right]^{-1} \frac{\partial D}{\partial \gamma_{l'}} \\ &= -\frac{\partial^2 D}{\partial \varphi_{il'} \, \partial \gamma_{l'}} \qquad (17)\end{aligned}$$

Using eqn. 10, eqn. 17 becomes

$$\frac{\partial^2 D}{\partial \varphi_{il} \, \partial \gamma_{l'}} = -R_{l'} t_i \qquad (18)$$

Considering eqns. 16 and 18, we obtain

$$\frac{\partial^2 D}{\partial \varphi_{il} \, \partial \varphi_{il'}} = -t_i^2 (D''_{il'} + R_{l'}) \qquad (19)$$

We note that eqn. 19, which is referred to a nondiagonal element, is not only of the same form as eqn. 8, which yields for a diagonal element, but also of the same order.

To generalise our consideration, we now assume that eqn. 11 becomes $\varphi_{il} + \varphi_{il'} + \varphi_{il''} + \ldots = 1$; that is, we consider nodes with more than two outgoing links. That, however, does not affect our results, because the condition given by eqn. 15 holds anyway.

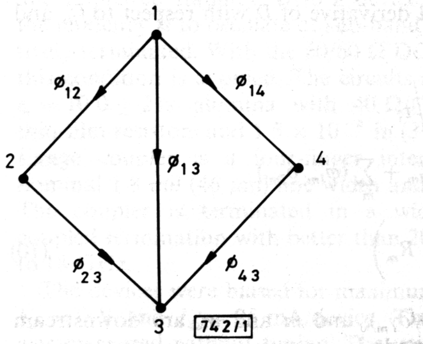

Fig. 1 *Network and routing pattern*

To illustrate the above considerations a simple example is given here. For the network and routing pattern of Fig. 1, the Hessian matrix is shown in Fig. 2.

$$\begin{array}{c|c|c|c|c|c|} & \frac{\partial}{\partial \phi_{12}} & \frac{\partial}{\partial \phi_{13}} & \frac{\partial}{\partial \phi_{14}} & \frac{\partial}{\partial \phi_{23}} & \frac{\partial}{\partial \phi_{43}} \\ \hline \frac{\partial D}{\partial \phi_{12}} & a & -b & -c & 0 & 0 \\ \hline \frac{\partial D}{\partial \phi_{13}} & -a & b & -c & 0 & 0 \\ \hline \frac{\partial D}{\partial \phi_{14}} & -a & -b & c & 0 & 0 \\ \hline \frac{\partial D}{\partial \phi_{23}} & 0 & 0 & 0 & d & 0 \\ \hline \frac{\partial D}{\partial \phi_{43}} & 0 & 0 & 0 & 0 & e \\ \hline \end{array}$$

742/2

Fig. 2 *Hessian matrix*

The determinant Δ of that matrix is

$$\Delta = -4ed\,abc$$

Considering that node 3 could be assumed as the destination node, it follows that $\phi_{23} = 1$ and $\phi_{43} = 1$ and therefore d and e are equal to zero; consequently $\Delta = 0$. That means the Hessian is singular and its inverse does not exist. That leads to the conclusion that a straightforward application of Newton's method seems to be impossible.

Conclusion: It has been demonstrated that in message-switched networks, where the independent variables are related to each other through linear constraint equations, the Hessian matrix of the objective function cannot be diagonal. Indeed, some of the nondiagonal elements can be of the same order as the diagonal ones. Moreover, it is found out that the matrix is singular, and consequently methods using matrix inversion calculations seem to be inappropriate for the solution of the routing problems.

F. PAVLIDOU *25th April 1986*
S. S. KOURIS

Electrical Engineering Department
University of Thessaloniki
Thessaloniki 54006, Greece

References

1 BERTSEKAS, D., GAFNI, E., and GALLAGER, R.: 'Second derivative algorithms for minimum delay distributed routing in networks', *IEEE Trans.*, 1984, **COM-32**, pp. 911–919
2 CHEN, M. S., and MEDITCH, J. S.: 'A distributed adaptive routing algorithm'. Proceedings of ICC '83, 1983, pp. 484–492
3 BERTSEKAS, D., GAFNI, E., and VASTOLA, K.: 'Validation of algorithms for optimal routing of flow in networks'. Proceedings of 17th conf. on decision and control, San Diego, Jan. 1978, pp. 220–227
4 KLEINROCK, L.: 'Queueing systems, vol. II: Computer applications' (Wiley & Sons, New York, 1976)
5 GALLAGER, R.: 'A minimum delay routing algorithm using distributed computation', *IEEE Trans.*, 1977, **COM-25**, pp. 73–84

EFFICIENT SERIAL/PARALLEL INNER-PRODUCT COMPUTATION

Indexing terms: Signal processing, Distributed arithmetic

A class of serial/parallel architectures for inner-product computation is described, based on carry-save accumulator arrays. In their basic form such arrays form carry-save multiply/adders. A simple modification of the coefficient feed allows flexible extension to short vector inner-product computation using distributed arithmetic. These modules may be cascaded to handle longer vectors, forming high-level VLSI digital signal processing subsystems.

Inner-product computers: There are many digital signal processing applications for inner-product (IP) computers, e.g. convolution, linear prediction etc. The IP of two vectors is formed by summing the pairwise products of the vector elements. As a multiplication is a 2-D sum of (weighted) single-bit products, an IP is then a 3-D sum of single-bit products (the 3rd dimension being the vector length). By permuting the summation indices, many different approaches are made possible. Classical multiply-accumulate techniques put the vector index outermost, while distributed arithmetic (DA)[1,2] has the data bit-index outermost. We propose a mixture of these two techniques, with serial-data interfaces.

The serial/parallel multiplier revisited: Acting on n-bit data with m-bit coefficients, the simple 2's complement (2C) serial/parallel (S/P) multiplier[3] forms the product in n clock cycles. Hardware consists mostly of a linear array of m carry-save adder (CSA) cells (each with a resident coefficient bit), which