WF SMYTH

INFORMATION PROCESSING 71 – NORTH-HOLLAND PUBLISHING COMPANY (1972)

# AN IMPROVED METHOD FOR REDUCING THE BANDWIDTH OF SPARSE SYMMETRIC MATRICES

Ilona ARANY, Lajos SZODA
*National Management Development Centre, Budapest, Hungary*

and

W.F.SMYTH
*International Labour Office, Geneva, Switzerland*

The method is a modification of the method of Cuthill–McKee [1]; it is based on the selection of an appropriate *level structure* for the connected undirected graph associated with the given matrix. (Usually, a level structure is a spanning tree with its edges removed.) The method has been programmed in Fortran on the ICL 1905E; comparison with the Cuthill–McKee method indicates a substantial further reduction in bandwidth for most graphs, with little or no increase in computer time.

## 1. INTRODUCTION

We wish to reduce the calculation time and storage requirements associated with operations on sparse matrices. These objectives may often be satisfied by storing the matrix in packed form and using special computer algorithms to perform the required operations [2,3]; the central difficulty with such methods is that the operations usually affect the sparseness in an undesirable way: The algorithm unavoidably creates more non-zero elements than were originally present, and alters their positioning in complex ways.

An alternative, and relatively uninvestigated, approach is to preprocess the matrix into band form. Such an approach is made interesting by the fact that most matrix operations applied to band matrices (see, for example [4, p. 261]) preserve matrix bandwidth. To date, however, only Cuthill–McKee [1] and Rosen [5] have published useful bandwidth reduction methods.

## 2. OUTLINE OF OUR METHOD

We begin with definitions. $A$ is a square matrix of order $n$ with elements $a(i,j)$. The *matrix bandwidth* is the least integer $\delta(A)$ such that

$$a(i,j) = a(j,i) = 0 \quad \text{for all} \quad |i-j| > \delta(A) .$$

$G(A)$, the *graph of* $A$, is an undirected graph whose $n$ *nodes* are numbered from 1 to $n$, and whose *edges* $(i,j)$ are defined as follows: the edge $(i,j)$ exists if and only if $a(i,j) \neq 0$ or $a(j,i) \neq 0$, $i \neq j$. Then the *graph bandwidth* is the least integer $\delta(G)$ such that no edge $(i,j)$ exists if $|i-j| > \delta(G)$. It follows that permuting the node numbers (renumbering the nodes) of $G(A)$ is equivalent to interchanging rows and corresponding columns of $A$; moreover, that the graph bandwidth $\delta(G(A))$ always takes the same value as the corresponding matrix bandwidth $\delta(A)$. We are concerned therefore in what follows to number the $n$ nodes of a graph $G$ in such a way that $\delta(G)$ takes as small a value as possible. $G$ will always be connected, undirected, and contain no loops or multiple edges.

A *level structure* LS = LS($G$) of a graph $G$ is an arrangement of the $n$ nodes of $G$ into $L$ *levels*, numbered $l = 1, \ldots, L$, so that nodes at a given level $l$ are joined (by an edge of $G$) to no nodes of levels other than $l-1$, $l$, and $l+1$. A level $l$ such that $1 < l < L$ is called an *intermediate level*. The number of nodes in a level $l$ is denoted $w(l)$, the *width of level* $l$. The *width of the level structure* is then

$$w = w(\mathrm{LS}) = \max_{1 \leq l \leq L} w(l) .$$

In the example shown in fig. 1, $w(1) = 2$, $w(2) = w(3) = 5$, $w(4) = 1$, so that $w(\mathrm{LS}) = 5$.

Following Cuthill–McKee, we note now that given an LS($G$) of width $w$, we can always number $G$ level by level so that $w \leq \delta(G) \leq 2w - 1$. It is therefore of

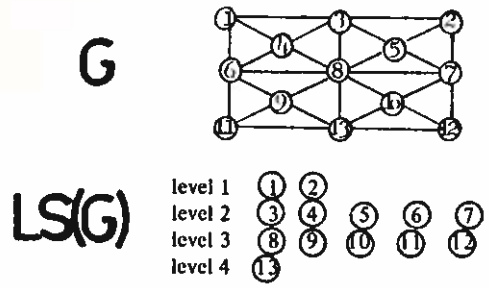



Fig. 1. A level structure of a graph G.

interest to search for LSs of small or minimum width, and in fact we make the following *conjecture*:

The minimum bandwidth $\delta = \delta(G)$ of a given graph $G$ may be attained by numbering $G$ based only on consideration of those LS($G$) which are of minimum width $w_0$.

We restrict our search for these "minimal" LSs by considering only associative level structures. An *associative level structure* (ALS) of $G$ is a level structure whose nodes of level 1 are arbitrarily selected, and whose nodes of level $l > 1$ are determined by the following iterative rule:

The nodes of level $l > 1$ are exactly those nodes joined to the nodes of level $l - 1$ (and not already included in levels $l - 1$ or $l - 2$).

In fact we restrict our search even more. A *reversible level structure* (RLS) of a graph $G$ is an ALS which is unchanged if we begin with the nodes of level $L$ and apply the dual of the above rule:

The nodes of level $l < L$ are exactly those nodes joined to the nodes of level $l + 1$ (and not already included in levels $l + 1$ or $l + 2$).

We note that an RLS is uniquely defined by any one of its levels; we refer to any intermediate level which defines an RLS as the *defining level* of the RLS. Note also that an ALS is an RLS if and only if every node in level $l < L$ is joined to some node in level $l + 1$. The LS of fig. 1 is in fact an RLS.

We conclude our definitions by defining a *maximal reversible level structure* (MRLS) of $G$. To do this, note first that although corresponding to a given defining level, we may always define an RLS, it does not follow that the RLS will contain every node of $G$. For if as we proceed from level $l$ to level $l + 1$, we find that some node of level $l$ is not joined to any node in $l + 1$, then as a consequence of the remark above, level $l$ represents the *upper limit* of the RLS definable by the given defining level. Similarly, we determine the *lower limit* of the RLS. An MRLS then is simply an RLS extended to its upper and lower limits from the given defining level. In most cases of practical interest the MRLS will in fact include every node of $G$; in case it does not, however, we make use of a series of defining levels to define an ALS($G$) consisting of a chain of MRLSs.

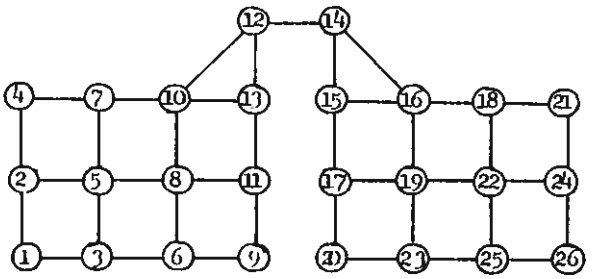


Fig. 2. An MRLS chain.

In fig. 2, choosing the defining level 7–8–9, we find the following MRLS:

level 1 (1)
level 2 (2) (3)
level 3 (4) (5) (6)
level 4 (7) (8) (9)
level 5 (10) (11)
level 6 (12) (13)

At level 6 the MRLS has an upper limit, because node 13 is not joined to node 14. Using another defining level, say 18–19–20, we find another MRLS for the other side:

level 1 (14) (15)
level 2 (16) (17)
level 3 (18) (19) (20)
level 4 (21) (22) (23)
level 5 (24) (25)
level 6 (26)

We are at last in a position to outline our method:

1. [Initialize] Start link 1 of the breakdown of $G$ into a chain of MRLSs $j \leftarrow 1$.

2. [Find link $j$] Execute procedure BREAKDOWN ($G$,$j$).

3. [Remove link $j$ from $G$] Remove the nodes of the MRLSs determined by BREAKDOWN ($G$,$j$) from $G$. If $G$ is now empty, go to 4. Otherwise, increment $j \leftarrow j + 1$, and go to 2.

4. [Number the chain of MRLS] Number the nodes of $G$ based on a level by level numbering of the chains of MRLSs. Select a numbering which gives rise to a least bandwidth $\delta(G)$. [See below, numbering the level structure.]

The procedure BREAKDOWN($G,j$) performs the following processing:

1. [Find possible defining levels] Select some node $i \in G$ which is of maximum degree. Find every possible defining level containing $i$. [See below, finding the defining levels].

2. [Select MRLSs for link $j$] For every defining level found above, determine the corresponding MRLS. There will be $k_j$ MRLSs of least width $w_0$. Store these in a table MRLS($j,k$), $k = 1, ..., k_j$.

We note that this numbering algorithm will yield a *minimum* bandwidth $\delta(G)$ if the following assumptions/conjectures are valid:

1. A node of maximum degree will always belong to an intermediate level of any ALS which is of minimum width.

2. Rather than consider all possible level structures LS($G$) of $G$, it suffices to consider only those ALSs which may be formed from a chain of MRLSs.

3. Further, only ALSs of minimum width need be considered for numbering (if $w(\mathrm{LS}_1) > w(\mathrm{LS}_2)$, then $\delta_1 \geqslant \delta_2$).

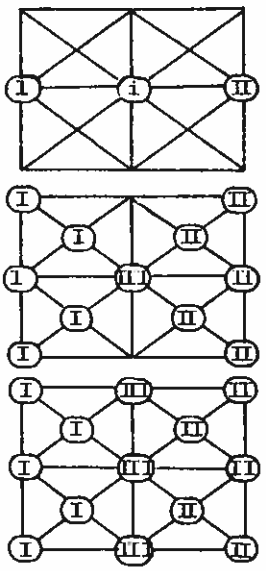
4. Any node contained in an MRLS defined by a defining level containing a given node $i$ (of maximum degree) is also contained in any other such MRLS defined from the same node $i$.

5. The level by level algorithm used to number the LS actually yields a $\delta$ which is the smallest achievable for that LS [see fig. 4].

## 3. FINDING THE DEFINING LEVELS

We discuss first the selection of the node $i$. Cuthill–McKee remark that "usually" nodes in first or final (non-intermediate) levels should have minimum or near-minimum degree; we make the complementary argument that usually when the ALS has minimum width, the nodes of maximum degree will belong to intermediate levels: unlike first and final levels, an intermediate level $l$ has two levels, $l - 1$ and $l + 1$, which together "absorb" the joins to the points of $l$. We take as our starting node $i$, therefore, simply any node of maximum degree.

We note now that a defining level is a rather special set of nodes: For if we think of removing from $G$ the nodes of such a level, together with the edges of which they are end-nodes, then the remaining graph may no longer be connected – it must consist of two or more disjoint non-void subgraphs. Fig. 3 illustrates the algorithm, based on this fact, which generates a defining level containing a given node $i$. To




Choose any two nodes joined to $i$ but to each other. These are the original nodes of type I and type II. $i$ itself is type III.

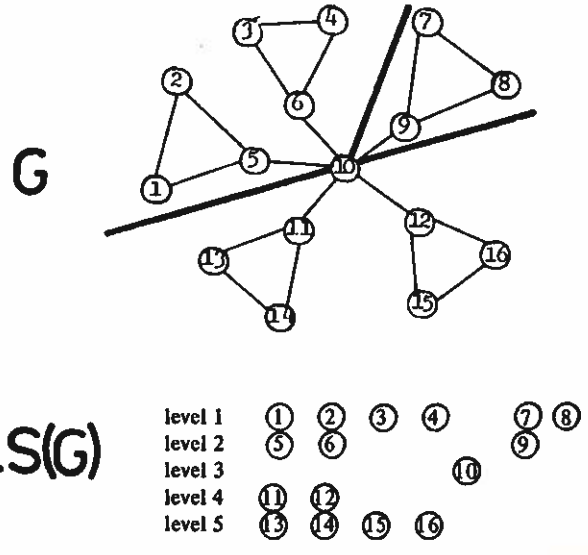
Nodes joined to type I nodes are also type I; to type II nodes, also type II.

Nodes joined to both type I and type II nodes are type III. The set of all type III nodes is a defining level containing $i$.

Fig. 3. Defining levels containing node $i$.

determine all possible defining levels containing $i$, we simply try all possible combinations of unconnected type I and type II nodes which are joined to $i$. Note, however, that the same defining level may result from several different pairs of starting nodes.

It may in fact happen, as shown in fig. 4, that the defining level separates $G$ into more than two disjoint subgraphs. In such a case, the LS-numbering algorithm




[$w$(LS) = 6 , $\delta(G) = 5$] .

Fig. 4. Several disjoint subgraphs.

must take account of "branches" or disjoint sets within a single level of the LS. In the example of fig. 4, there is no chain of MRLSs which has width less than 6; perhaps significantly, however, it is possible to find a non-reversible ALS which has width 5 and which moreover may be numbered level by level in the ordinary way to give $\delta(G) = 5$.

## 4. NUMBERING THE LEVEL STRUCTURE

As indicated by the work of Cuthill–McKee, *natural numbering* (that is, numbering the nodes of each level in the order in which they happen to be stored) can in many cases produce quite satisfactory, even optimum, results. The computer program presently uses a slight modification of natural numbering, which involves a limited look-ahead from level $l$ to level $l + 1$. This method is an improvement on natural numbering in most cases, but not optimal. Several other methods are under consideration, each involving both look-ahead (from level $l$ to level $l + 1$) and look-back (from level $l$ to level $l - 1$) features. These methods all seem to produce near-optimal level by level numbering of the LS, at some cost in terms of computer storage and processing time. The time penalty seems to be slight, however, because in practice processing is often terminated by finding a bandwidth $\delta(G) = w_0$.

In this context, we remark that it seems usually to be possible to number the LS so that $\delta(G) \leqslant w_0 + K$, where $K$ is some small positive integer.

## 5. THE COMPUTER RESULTS

Four different methods have been programmed:

1. [Cuthill–McKee (denoted by CM)] Each node of minimum degree is used as a starting node for a numbering of $G$ based on a spanning tree strategy [1].

2. [Modified Cuthill–McKee (MCM)] Starting nodes include also nodes of "near-minimum" degree.

3. [Our Method (OM)] The method tested here differs in some respects from the method described above. In particular, only one MRLS is determined for $G$, not the chain of MRLSs indicated in fig. 2; the LS is thus in all cases based on only one defining level. Moreover, the sophisticated handling of disjoint subgraphs indicated in fig. 4 has not been programmed. We have already noted the limitations on the numbering system.

4. [Rosen Method (RM)] Given a numbering of $G(A)$, rows and corresponding columns of $A$ are interchanged in a methodical way, in an effort to reduce the bandwidth [5].

The first three of these methods may be termed *constructive methods*, since they develop a numbering generally unrelated to any previous numbering of the nodes. The fourth method, by contrast, is an *iterative method*. Because of this distinction, and also because RM used alone tends to be very much slower and probably less effective than CM or MCM [1], we have followed the example of Cuthill–McKee and used RM only in combination. The six approaches tested are then:

| | |
|---|---|
| CM; | CM followed by RM; |
| MCM; | MCM followed by RM; |
| OM; | OM followed by RM. |

These approaches were tested against 13 different graphs of varying size and complexity. Table 1 describes the characteristics of the graphs tested, and table 2 gives averages of the results obtained.

Table 1
Characteristics of test graphs

| Graph No. | No. of nodes | No. of edges | Orig. bandwidth | Min. bandwidth |
|---|---|---|---|---|
| 1 | 19 | 34 | 18 | 4 |
| 2 | 45 | 85 | 36 | 5 |
| 3 | 26 | 39 | 4 | 3 |
| 4 | 14 | 24 | 7 | 4 |
| 5 | 16 | 16 | 15 | 2 |
| 6 | 14 | 27 | 6 | 3 |
| 7 | 42 | 81 | 37 | 7 |
| 8 | 7 | 11 | 6 | 3 |
| 9 | 14 | 23 | 4 | 4 |
| 10 | 25 | 43 | 6 | 5 |
| 11 | 32 | 42 | 31 | 6 |
| AVE. | 23 | 48 | 15 | 4.2 |
| 12 | 91 | 240 | 37 | 12 |
| 13 | 99 | 169 | 89 | 13 |
| AVE. | 95 | 205 | 63 | 12.5 |

We conclude with a few remarks:

(1) The computer times quoted must be regarded as approximate, since the computer available to the authors had no facility for precise program-controlled subroutine timing. Moreover, the CM + RM and MCM + RM timings were both revised downward to

Table 2
Average test results

| Method | AVE. 1–11 | | AVE. 12–13 | |
|---|---|---|---|---|
| | bandwidth | Time (sec) | bandwidth | Time (sec) |
| CM | 5.1 | 0.45 | 18 | 6.0 |
| CM + RM | 4.4 | 1.52 | 14.5 | 24.5 |
| MCM | 4.8 | 2.02 | 18 | 40.0 |
| MCM + RM | 4.2 | 3.02 | 14.5 | 69.0 |
| OM | 4.5 | 0.70 | 14.5 | 31.0 |
| OM + RM | 4.2 | 2.52 | 12.5 | 42.8 |

make allowance for possible inefficiencies in the programming of these methods.

(2) OM has also been programmed in a somewhat inefficient manner (at the moment *every* level structure, not just those of minimum width, is redundantly numbered). The expected effect of removing inefficiencies and *adding* the more sophisticated features noted above to the program is that the OM + RM results (average bandwidth = minimum bandwidth = 4.2/12.5) will be attained within the present OM time (0.70/31.0 sec).

(3) We have taken the liberty of improving CM and MCM somewhat for the purpose of the computer tests. The numbering system used in each case is the simple look-ahead method used for OM. This gives decreased and more data-independent bandwidth values than would be achieved by the original natural numbering, with negligible increase in computer time.

(4) With these remarks in mind, it is possible to argue that OM represents an improvement over previous methods in the sense that it yields a substantially lower average bandwidth with comparable computer time investment. This argument, however, requires further experimental justification. We note that CM will generally perform more efficiently on graphs containing few nodes of minimum degree, whereas OM is more suited to graphs which give rise to few possible MRLSs.

(5) The behaviour of all methods as graph size increases requires investigation. Based on the presently available evidence, one might conjecture that computer time for every method except CM increases with $n^2$ (or more).

## REFERENCES

[1] E.Cuthill and J.McKee, Reducing the bandwidth of sparse symmetric matrices, Proc. 24th Nat. Conf. ACM (1969) 157–172.
[2] R.P.Tewarson, Computations with sparse matrices, SIAM Review 12 (1970) 527–543.
[3] R.A.Villougby, Sparse matrix algorithms and their relation to problem classes and computer architecture, IBM Research Publication RC 2833 (March 1970) 38.
[4] L.Fox, An Introduction to Numerical Linear Algebra (Clarendon Press, 1964).
[5] R.Rosen, Matrix bandwidth minimization, Proc. 23rd Nat. Conf. ACM (1968) 585–595.