# MINIMIZING THE BANDWIDTH OF SPARSE MATRICES

By

ILONA ARANY, W. F. SMYTH, LAJOS SZÓDA

*(Received June 15, 1973)*

## 1. Introducing the problem

The problem arises in modern computing. There are a great many problems which involve linear systems of large size. It seems to be required for many different applications – solving large systems of linear equations, linear programming, solving systems of differential equations – to perform operations on matrices and to store them.

Consider the following system of linear equations

$$Ax = b$$

where $A$ is an $Nx\,N$ matrix, sparse and symmetric. If the non-zero elements in $A$ are placed in a narrow band near the main diagonal, then the number of operations, hence the computer time and storage required for $A$, are decreased in comparison with a general case. So before performing the operations it seems to be advisable to convert the matrix into such a band form.

This paper describes a new method for reducing the bandwidth of sparse symmetric matrices. A version of this material was presented at IFIP CONGRESS 71 in Ljubljana.

*Definition*

Let $A(a_{ij})$ be an arbitrary square matrix; then

$$\delta(A) = \max_{\substack{a_{ij} \in A \\ a_{ij} \neq 0}} |i-j| \tag{1}$$

is called the *bandwidth* of the matrix (see Figure 1).

Our aim is to minimize $\delta(A)$; that is, to find a suitable rearrangement of the rows and corresponding columns of $A$ so that (1) will take its least value.

The problem can also be approached from the point of view of graph theory. This is what we have chosen in our work.

$$A = \begin{bmatrix} 1 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 \end{bmatrix}$$

Denote $b_i = \max_{\substack{1 \leq j \leq N \\ a_{ij} \neq 0}} |i-j|$,

then $\delta(A) = \max_{1 \leq i \leq N} b_i$

$a_{1j} \neq 0$; $j = 1, 2, 4, 5, 6$

$b_1 = |1-6| = 5$

$a_{2j} \neq 0$; $j = 1, 2, 5, 7, 8$

$b_2 = |2-8| = 6$

$a_{3j} \neq 0$; $j = 3, 4, 5, 7$

$b_3 = |3-7| = 4 \ldots$

It is easy to see that

$$b_i \leq 6$$

(and $b_2 = b_8 = 6$ for elements $a_{28}$, $a_{82}$).

So

$$\delta(A) = 6$$

*Figure 1.*

Let $A(a_{ij})$ be a non-singular $NxN$ matrix for which all the elements in the main-diagonal are non-zero. Then $A$ may be associated with a connected, undirected graph $G(X, E)$ (denoted by $G(A)$, $X$ is the set of nodes, $E$ the set of edges in the graph) as follows:

$$X = \{x_1, x_2, \ldots, x_N\}, \quad \text{where } N \text{ is the number of rows in } A;$$

$$E = \{(x_i, x_j) | \, a_{ij} \in A; \quad a_{ij} \neq 0; \quad i \neq j\},$$

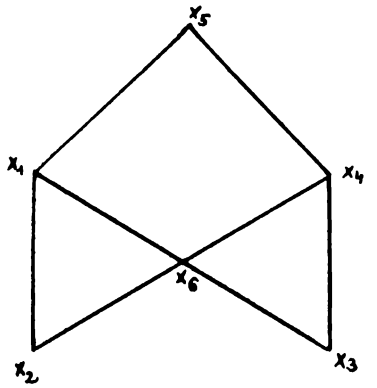
This is a special correspondence between the matrix and the graph that is, for every undirected, connected graph we may uniqualy specify its „connection matrix".

*Definition*

Consider an undirected, connected (these properties will be assumed in what follows) graph $G(X, E)$; $X = \{x_1, x_2, \ldots, x_N\}$. To identify the nodes denote them by the numbers 1 to $N$; this is called a *numbering* of the graph. In other words, an assignment

$$\alpha: X \to \{1, 2, \ldots N-1, N\}$$

yields a numbering of the graph.




$$\begin{bmatrix} 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 0 & 1 \end{bmatrix}$$

*Figure 2.*

*Definition*

Let $\alpha$ be a numbering of the graph $G(X, E)$; then

$$\delta_\alpha = \max_{(x_i, x_j) \in E} |\alpha(x_i) - \alpha(x_j)| \tag{2}$$

is the *bandwidth of the graph* associated with the given numbering $\alpha$. Notice that

$$\delta_\alpha = \delta(A_\alpha)$$

where $A_\alpha$ is the connection matrix of the graph numbered by $\alpha$. It may, of course, happen that a new numbering induces a change in the bandwidth. Our object is to find a numbering $\alpha_0$ for which (2) takes its minimum value

$$\delta = \delta_{\alpha_0} = \min\ \delta_\alpha$$

We present some very simple estimates for the minimum bandwidth $\delta$: Let $d_i$ be the degree of the node $x_i \in X$; that is, the number of edges converging at node $x_i$. Then

$$d_{\max} = \max_{1 \leqq i \leqq N} d_i, \quad d_{\min} = \min_{1 \leqq i \leqq N} d_i$$

are the values of the maximum and minimum degree respectively. It is clear that

$$\delta \geq \left[ \frac{d_{\max}}{2} \right]$$

is satisfied. It may also be seen easily that

$$\delta \geq d_{\min}$$

is also true.

If $D$ is the diameter of the graph, then

$$\delta \geq \left[\frac{N-1}{D}\right],$$

where $N$ is the number of the nodes. (The diameter of the graph is

$$D = \max_{x_i, x_j \in X} \varrho(x_i, x_j)$$

where $\varrho(x_i, x_j)$ means the distance between nodes $x_i$ and $x_j$.)

## 2. Comprehensive survey of methods for reducing the bandwidth

2.1 Types of methods and their main properties
The methods may be broken into two types:

1. Iterative methods.
2. Constructive methods.

The main characteristic of the first type is that it improves the bandwidth of a given matrix. Starting with a given bandwidth (numbering) the method attempts to produce a better one. Using these methods alone tends to be rather more slow and probably somewhat less effective than using other methods. Constructive methods have some advantages over iterative methods. They produce a bandwidth independent of the original or previous numbering. They require rather less computing time in general, and the results are in fact nearer to the minimum. So it is more efficient and more economical to apply them.

2.2 Brief survey of some previous methods

1. R. Rosen' s method [3]
The basic idea is to make an interchange among the rows and corresponding columns of a given matrix sufficient to cause a decrease in bandwidth. At first interchanges are made which directly cause the bandwidth to become smaller; later it is allowed to make some interchanges which do not change the bandwidth, but which lead to a structural change in the matrix, which in turn makes it possible to decrease the bandwidth further. This is a typical iterative method, which is not very efficient and requires too much computing time.

2. G. Alway and D. Martin's method [4]
This is also an iterative method whose efficiency is not too great.

3. E. Cuthill and J. McKee's method [1]
This is the constructive method most well-known in the literature. Cuthill – contrary to Rosen – approaches the problem from the point of view of graph theory.

Theoretically the method consists of wo different parts:

1. Determining a spanning tree of the graph, and arranging the nodes into levels.

2. Numbering the levels.
In practice the two steps are carried out at the same time.

1. Starting from a node of minimum degree $x_i \in X$, determine a spanning tree of the graph. The nodes are arranged into disjoint subsets called levels (denoted by $L_j$) so that the nodes the same distance from $x_i$ belong to the same level.
Using the distance function $\varrho(x_i, x_j)$, we may then define levels as follows:

$$\begin{aligned} L_1 &= \{x_i\} \\ L_2 &= \{x_j | x_j \in X; \quad \varrho(x_i, x_j) = 1\} \\ L_3 &= \{x_j | x_j \in X; \quad \varrho(x_i, x_j) = 2\} \\ &\vdots \\ L_t &= \{x_j | x_j \in X; \quad \varrho(x_i, x_j) = t-1\} \\ [L_t &= \{x_j | x_j \in X; \quad x_j \notin L_{t-2}; \quad \varrho(x_j, L_{t-1}) = 1\}] \\ &\vdots \end{aligned} \tag{3}$$

In the case of a connected graph, after a finite number of steps an index $k$ may be found for which

$$L_{k+1} = 0; \text{ this is equivalent to } \bigcup_{l=1}^{k} L_l = X$$

2. The numbering of nodes takes place level by level in the order they happen to be found. The starting node $x_i$ is numbered 1.
Denote the numbering by $\alpha$

$$\begin{aligned} \alpha(x_i) &= \{l_1 = 1\} \\ \alpha(L_2) &= \{2, 3, \ldots l_2\} \\ \alpha(L_3) &= \{l_2+1, \; l_2+2, \ldots l_3\} \\ &\vdots \\ \alpha(L_k) &= \{l_{k-1}+1, \; l_{k-1}+2, \ldots, l_k = N\} \end{aligned} \tag{4}$$

The numbering of every level is carried out on the basis of the numbers assigned to the previous level. When the nodes in $L_m$ have already been numbered, we consider in increasing sequence of node number those nodes in $L_m$ which are joined to $L_{m+1}$:

$$\alpha^{-1}(l_{m-1}+1), \; \alpha^{-1}(l_{m-1}+2), \ldots \alpha^{-1}(l_m).$$

First we select nodes in $L_{m+1}$ adjacent to $\alpha^{-1}(l_{m-1}+1)$ to be numbered $l_m+1$, $l_m+2$, $l_m+3$,... respectively, in order of increasing degree. This numbering system has to be applied for every node in $L_{m+1}$. Continuing the process, after finite number of steps all the nodes in the graph will be numbered. At the same time we are making sure the bandwidth remains as small as possible.

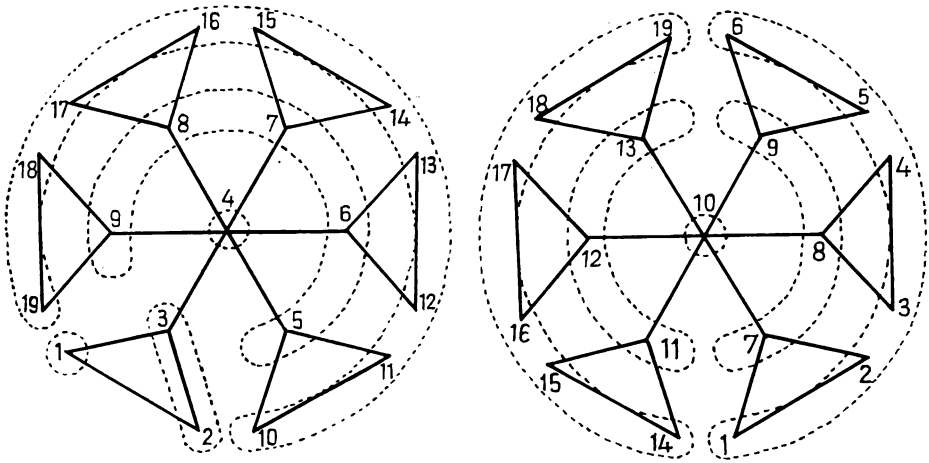
The method described here gives a numbering corresponding to one starting point. Cuthill-McKee suggest repeating the procedure using every node of minimum degree as a starting point, and selecting the numbering for which $\delta$ is the least. This is fast method giving good results in many cases.

A modified Cuthill – Mckee method is also known. This differs from the original method in the selection of the starting nodes: all nodes of minimum and „near-minimum" degree are used as starting points. Nodes of near – minimum degree satisfy

$$d_{\min} \leq d_i \leq d_{\min} + \frac{N}{M} \cdot d_{\max}$$

where Cuthill-Mckee suggest chosing $N = 1$, $M = 2$.

A disadvantage of the method is that it is based on the starting point and hence does not consider any structural properties of the graph. In some cases (Figure 3) the bandwidth given by the Cuthill-McKee method is considerably larger than the theoretical minimum (which is 5 here).




$\delta = 10$ — Cuthill – McKee's nunbering

$\delta = 6$ — Our numbering

*Figure 3.*

## 3. Theoretical basis of our method: description of the algorithm

### 3.1 Introduction

Using the Cuthill-McKee method the nodes are separated into hierarchical levels by determining a spanning tree. This is a partial ordering of the nodes, and it is this ordering which is the real basis of the numbering. Thus the

spanning tree is only a tool for the creating of these levels which avoids using other properties of the graph.

We – like Cuthill – approach the problem from the point of view of graph theory. The essential difference is that we make use of other structural properties of the graph in order to determine the levels.
Our method consists of two important steps:

1. Separating the nodes into disjoint levels.
2. Numbering the nodes.

Before describing the method we provide some theoretical background material:

3.2 The level structure and its basic properties

*Definition*

For a given graph $G(X, E)$ consider

$$\{L_1, L_2, \ldots L_k\} \tag{5}$$

a sequence of subsets of nodes such that

$$L_i \subset X; \qquad i = 1, 2, \ldots, k;$$
$$L_i \cap L_j = 0; \quad i, j = 1, 2, \ldots, k; \quad i \neq j;$$
$$\bigcup_{j=1}^{k} L_j = X.$$

If for arbitrary $x_i, x_j \in X$; $(x_i, x_j) \in E$ implies $x_j \in L_{j*}$; $x_i \in L_{i*}$; such that $|i^* - j^*| \leq 1$, then (5) is called a *level structure* (denoted by LS), and the $L_i$ are its levels.
In other words the level structure is a separation of the nodes into disjoint levels such that every edge is connected only to nodes in the same or neighbouring levels.
Levels $L_1$ and $L_k$ are the *extreme* levels and the other levels, $L_2, L_3, \ldots, L_{k-1}$ are *intermediate* levels.
According to the definition every intermediate level is itself a cut set of the graph.
It is easy to see that the following statements are true:

*a)* Every graph consisting of at least two nodes has a level structure.
Consider $x_1 \in X$; $\{x_1\} \neq X$ then

$$\{\{x_1\},\ X \backslash \{x_1\}\} \tag{6}$$

is in fact an LS.
An LS having no intermediate levels, such that (6) is a trivial LS.

*b)* For every non-complete graph having at least three nodes there exist a non-trivial LS.
In this case there exist $x_i, x_j \in X$ such that $(x_i, x_j) \notin E$; then

$$\left\{\{x_i\},\ X \backslash \{\{x_i\} \cup \{x_j\}\},\ \{x_j\}\right\}$$

satisfies the definition of LS.
Obviously, no complete graph has a non-trivial LS.

*Definition*

For a given LS denote by $W(L_i)$ the number of nodes in $L_i$.
Then

$$W(LS\} = \max_{1 \leq i \leq k} W(L_i) \tag{7}$$

is called the *width of LS*.

Let $W_0$ be the width of the LS for which (7) takes its minimum value:

$$W_0 = \min_{LS} W(LS)$$

*Definition*

A numbering $\alpha$ is called *compatible with an LS* if for every

$$x_i, x_j \in X(x_i \in L_{i*} \text{ and } x_j \in L_{j*}), \text{ then}$$
$$i^* < j^* \text{ implies } \alpha(x_i) < \alpha(x_j). \tag{8}$$

Obviously, the correspondence between the numbering and the LS is not unique, because for every LS a veriety of compatible numberings may be obtained; and conversely, for every numbering we may find several different level structures satisfying (8).

We wish to prove several important results:

*a)* Let $\alpha$ be a numbering compatible with LS and $\delta_\alpha$ the corresponding bandwidth. Then

$$\delta_\alpha \leq 2 \cdot W(LS) - 1 \tag{9}$$

is true.
Consider $(x, \tilde{x}) \in E$ such that $\delta_\alpha = \alpha(\tilde{x}) - \alpha(x)$. Then if $x \in L_i$ we have $\tilde{x} \in L_{i*}$ such that $i^* = \begin{cases} i \\ i+1 \end{cases}$. It follows therefore that

$$\delta_\alpha = \alpha(\tilde{x}) - \alpha(x) \leq \max_{x_j \in L_{i*}} \alpha(x_j) - \min_{x_j \in L_i} \alpha(x_j) = W(L_i) + W(L_{i+1}) - 1 \tag{10}$$

from which (9) follows immediately.

*b)* For every LS there exists a numbering $\alpha$ compatible with LS. We see that every numbering will suffice for which $x_i \in L_{i*}$ implies

$$\sum_{j < i^*} W(L_j) < \alpha(x_i) \leq \sum_{j \leq i^*} W(L_j)$$

A numbering created in such a way will ensure that the bound given in (9) is satisfied.

No better general estimate of the bandwidth than (9) has yet been found. We know therefore that for every LS we can always find a numbering for which

$$\delta_\alpha \leq 2 \cdot W(LS) - 1$$

be satisfied.

For a given LS consider a numbering $\alpha_0$ compatible with LS and such that the minimum bandwidth of the graph is attained. Notice that the upper bound on $\delta$ may be decreased in some measure for the most important practical cases. So

$$\delta_{\alpha_0} \leq c \cdot W(LS)$$

is true, where $c<2$. A suitable value of $c$ has not yet been determined, but the graph in Figure 4 indicates

$$c \geqq \frac{3}{2}.$$

In the greater part of the graphs found in practice

$$\delta_{\alpha_0} = W(LS) + K$$

where $K \ll W(LS)$, and often $K = 0$ or $K = 1$.

*c)* Let $\alpha$ be an arbitrary numbering; then it is possible to find an LS with which $\alpha$ is compatible and such that

$$\delta_\alpha = W(LS)$$

is also true.

Consider $(x_{i*}, x_{j*}) \in E$ where $\delta_\alpha = \alpha(x_{i*}) - \alpha(x_{j*})$.

Create

$$L_l = \{x_j | \alpha(x_{j*}) < \alpha(x_j) \leq \alpha(x_{i*})\},$$

and then

$$W(L_l) = \delta_\alpha$$

Other levels of the LS may be determined from the following recursion relations:

$$L_{t+l} = \{x_j | \underset{x_i \in L_{t+l-2}}{\alpha(x_j) < \alpha(x_j) \leq \max_{\varrho(L_{t+l-1}, x_i)=1} \alpha(x_i)}\}$$

$$L_{t-l} = \{x_j | \underset{x_i \in L_{t-l+2}}{\alpha(x_i) > \alpha(x_j) \geq \min_{\varrho(L_{t-l+1}, x_i)=1} \alpha(x_i)}\}$$

Now consider $x' \in X$ such that $\alpha(x') = \max_{x_i \in L_{t+l}} \alpha(x_i)$; then $\varrho(L_{t+l-1}, x') = 1$ and there exists $x'' \in L_{t+l-1}$ such that $(x', x'') \in E$.

Then

$$W(L_t) = \delta_\alpha \geq \alpha(x') - \alpha(x'') \geq \alpha(x') - \max_{x_i \in L_{t+l-1}} \alpha(x_i) = W(L_{t+l})$$

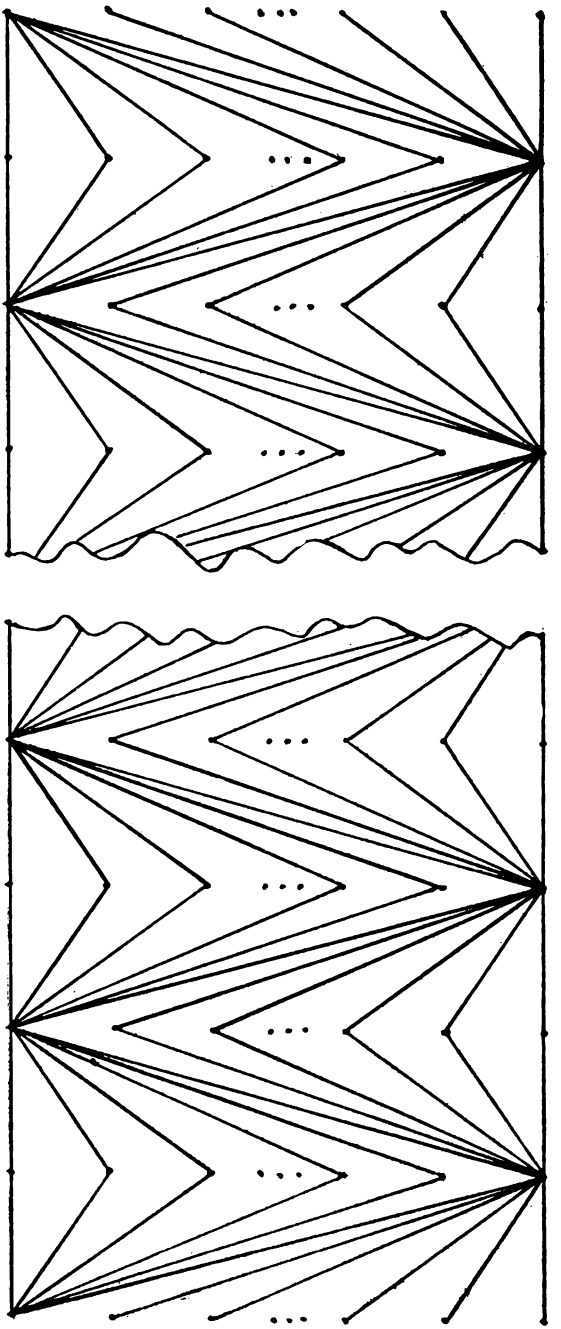


*Figure 4.*

from which it follows, in conjunction with a similar result for $L_{t-l}$, that

$$W(L_t) = W(LS).$$

*d)* Let $\delta$ be the minimum bandwidth. Then as an immediate consequence of *a)* and *c)*

$$W_0 \leq \delta \leq 2 \cdot W_0 - 1 \tag{11}$$

Note that (11) does not guarantee that any numbering compatible with any LS of minimum width will yield the minimum bandwidth.
In view of the above results, for the application of the method we restrict ourselves to the numbering of LS, and moreover to LSs of small width.
The number of possible level structures is very large, so we deal only with a a narrow class of LSs; level structures having a defining level.

*Definition*

In a level structure

$$\{L_1, L_2, \ldots, L_k\}$$

$L_i$ is called a defining level if for every $x_j \in L_{j*}$; $(j* \neq i)$;

if $j* > i$, then there exists $x_{j'} \in L_{j*-1}$ for which $(x_{j'}, x_j) \in E$;

if $j* < i$, then there exists $x_{j''} \in L_{j*+1}$ for which $(x_{j''}, x_j) \in E$.

In other words, $L_i$ is a defining level of the LS if there exists a chain from any node of the graph to $L_i$ such that each node in the chain belongs to a unique different level.
For an LS having a defining level, whenever a node and all its neighbour nodes belong to the same level, then that level must be the defining level.
To illustrate the meaning of the above definition we give some examples.

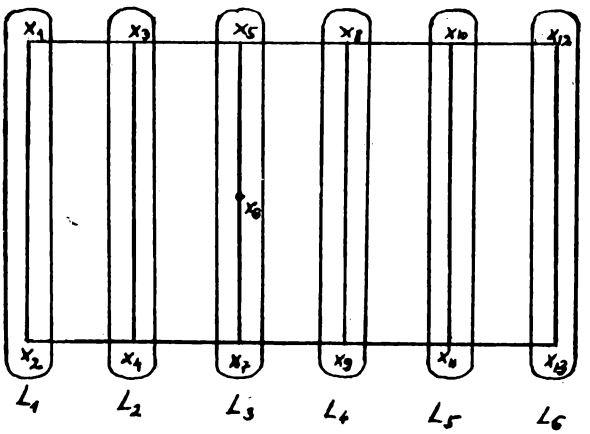



*Figure 5.*

For the graph shown in Figure 5, $L_3$ is the defining level of the given LS. No other level can be a defining level, because all the neighbour nodes of $x_6$ also belong to $L_3$.
For a similar reason the LS in Figure 6a has no defining level.
The LS in Figure 6b is a very special one because all the levels are defining levels of the LS.

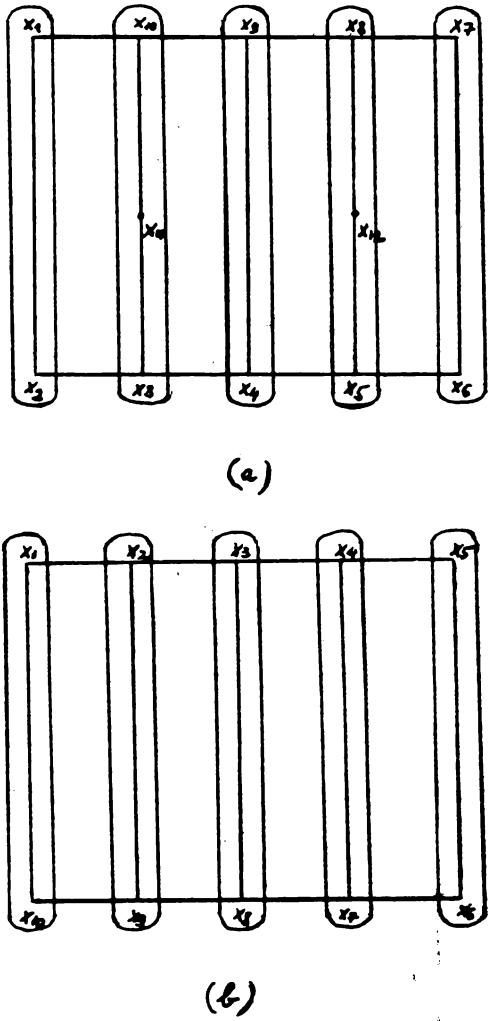


*Figure 6.*

It is easy to see that if $L_i$ is a defining level of an LS and $\alpha$ an arbitrary numbering compatible with the LS, then

$$W_0' \leq \delta_\alpha \leq 2W_0' - 1 ,$$

where

$$W_0' = \max_{j \neq i} W(L_j)$$

The Cuthill-McKee method leads to the creation of a very special LS, for which the defining level is a starting node on which the other levels depend. For an LS such that as this the value $W_0'$ may be larger than necessary, because it often happens that if $j<j'$ then $W(L_j) \leq W(L_{j'})$. As a result the connection matrix associated with a numbering $\alpha$ based on the LS has a very special form (see Figure 7): $W_0'$ and $\delta_\alpha$ have become larger in the righthand lower corner of the band. To compensate for this distortion it seems to be reasonable to create an LS for which the defining level is an intermediate one.

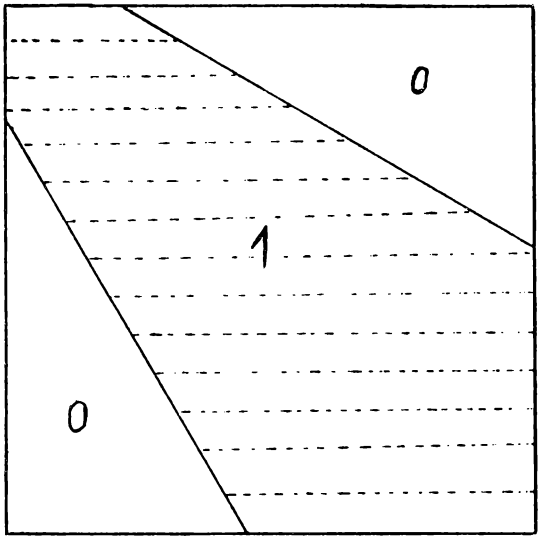



*Figure 7.*

3.2 Survey of our algorithm

I. Determining the level structure

*a)* Determining the defining level
*b)* Constructing the LS generated by the defining level

II Numbering system

I Determining the level strucure

*a)* Our aim is to determine an LS for which the defining level $D$ is an intermediate level. We wish therefore to determine a cut set of the graph which includes a selected starting node $x_{i_0}$ (which we choose to be a node of maximum degree). The cut set separates $x_1$, $x_2$ neighbour nodes of $x_{i_0}$ into disjoint classes, $X_1$ and $X_2$, which are not in connection with each other. We may write then

$$X = X_1 \cup D \cup X_2$$

where

$$x_{i_0} \in D; \qquad x_1 \in X_1; \qquad x_2 \in X_2 .$$

The procedure consists of the following steps:

1. Determining the starting point (that is, find a node of maximum degree) and initialize:

$$D := \{x_{i_0}\}; \qquad X_1 := \Phi; \qquad X_2 := \Phi.$$

2. Assign a "type" number $\tau$ to each node in the graph as follows:

$$\tau(x_i) = k; \quad i = 1, 2, \ldots, N;$$

where $k$ may take the values 0,1,2,3.
At the beginning

$$\tau(x_i) = 0; \qquad i = 1, 2, \ldots, N; \qquad \tau(x_{i_0}) := 3.$$

3. Determine a point-pair $x_1$, $x_2$ satisfying the following conditions:

$$\tau(x_1) = 0; \qquad \tau(x_2) = 0;$$

$$\varrho(x_1, D) = 1; \qquad \varrho(x_2, D) = 1; \text{ and } (x_1, x_2) \notin E$$

Then make the following changes:

$$\tau(x_1) := 1; \qquad \overline{X_1} := \{x_1\};$$

$$\tau(x_2) := 2; \qquad \overline{X_2} := \{x_2\}.$$

4. Consider all the nodes $x_j$ which neighbour $\overline{X_1}$:

if $\tau(x_j) = 0$, then set $\tau(x_j) := 1$ and $\overline{X_1} := \overline{X_1} \cup \{x_j\}$;
if $\tau(x_j) = 2$, then set $\tau(x_j) := 3$ and $D := D \cup \{x_j\}$
if $\tau(x_j) = 3$, then no action.

5. Consider all the nodes $x_j$ which negihbour $\overline{X_2}$:

if $\tau(x_j) = 0$, then set $\tau(x_j) := 2$ and $\overline{X_2} := \overline{X_2} \cup \{x_j\}$;
if $\tau(x_j) = 1$, then set $\tau(x_j) := 3$ and $D := D \cup \{x_j\}$;
if $\tau(x_j) = 3$, then no action.

If during steps 4. and 5. we find no nodes whose type was changed, then the procedure is continued at step 6.; otherwise, we return to step 4.

6. $$X_1 := X_1 \cup \overline{X_1}; \qquad X_2 := X_2 \cup \overline{X_2}.$$

If there are any remaining points of type 0, then the procedure is continued at step 3.; otherwise, $D$ is the defining level.
It is clear that after a finite number of steps every node of the graph is marked by type 1, 2, or 3; and

$$D = \{x_j | \tau(x_j) = 3\}$$

is a cut set of the graph.
Figure 9 shows the essence of this procedure.

*b)* Having found the set $D$ we proceed to construct the corresponding LS. Starting from $D$ separate all nodes into disjoint levels as follows: Given level $L_j$, the next level $L_{j+1}$ (or previous level $L_{j-1}$) consists of the neighbour nodes of $L_j$ not yet assigned to any level. Denoting

$$l = \max_{x_i \in X_1} \varrho(x_i, D) + 1 ,$$

the levels have the form:

$$L_t = \begin{cases} \{x_i | x_i \in X_1 ; \quad \varrho(x_i, D) = l - t\}, & \text{if } t < l ; \\ D \qquad\qquad\qquad\qquad\qquad\quad , & \text{if } t = l ; \\ \{x_i | x_i \in X_2 ; \quad \varrho(x_i, D) = t - l\}, & \text{if } t > l ; \end{cases}$$

Notice that the computer program actually determines only the set $D$. The determination of the LS itself is done as part of the numbering process. Figure 8 shows LS configurations both for Cuthill-Mckee method and for ours. It makes clear the significant difference between the two methods.

Cuthill – McKee method

Our method

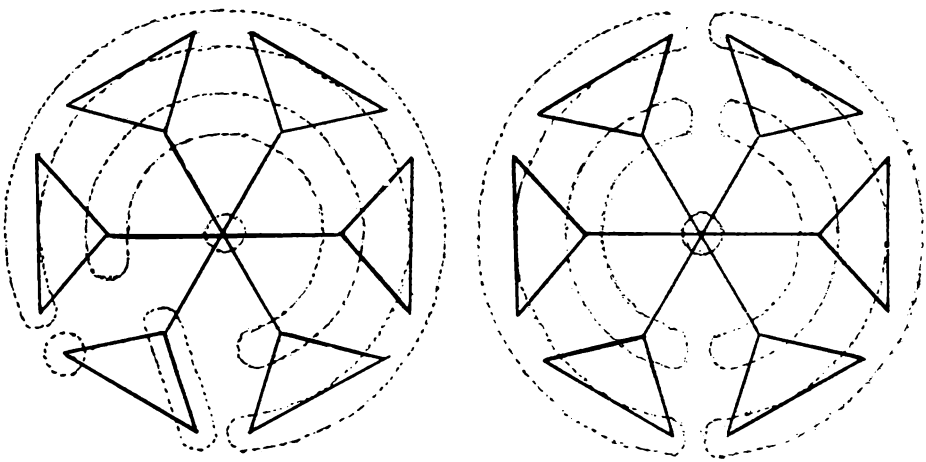
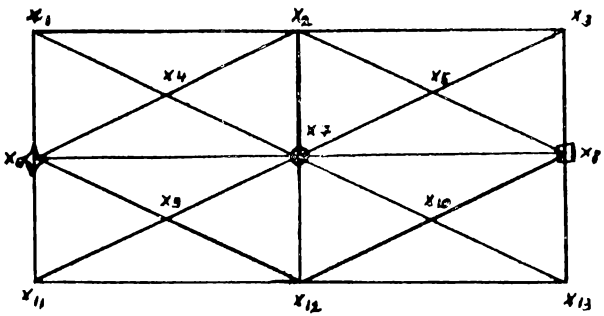


*Figure 8.*

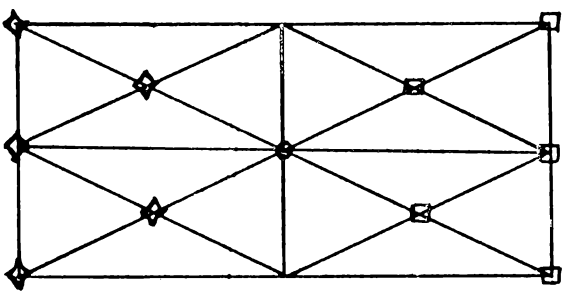
Determining the defining level corresponding to starting point $x_7$ in Figure 9.



1. $\tau(x_j) = 0,$
$i = 1, 2, \ldots, 13.$

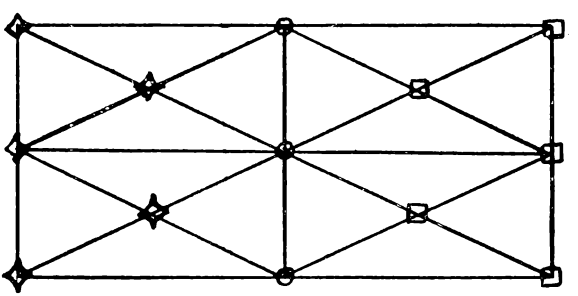
Select two points $x_6$, $x_8$ which neighbour $x_7$ and which are not in connection with each other. Then

$\tau(x_7) := 3; \quad \tau(x_6) := 1,$
$\tau(x_8) := 2.$



2. Nodes of type 0 which neighbour $x_6$ become type 1. Similarly, nodes of type 0 which are neighbour $x_8$ become type 2:

$\tau(x_i) := 1,$
$i = 1, 4, 9, 11;$
$\tau(x_i) := 2,$
$i = 3, 5, 10, 13.$



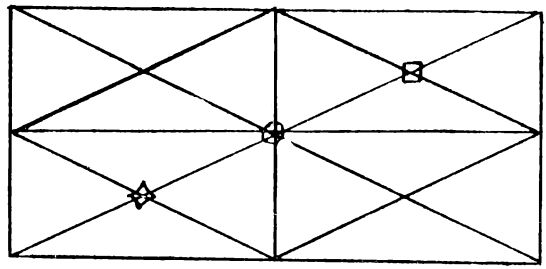
3. Nodes of type 0 which neighbour nodes of type 1 become type 1; and nodes of type 1 having any neighbour node of type 2 become type 3:

$\tau(x_i) := 1, \quad i = 2, 12;$
$\tau(x_i) := 3, \quad i = 2, 12.$

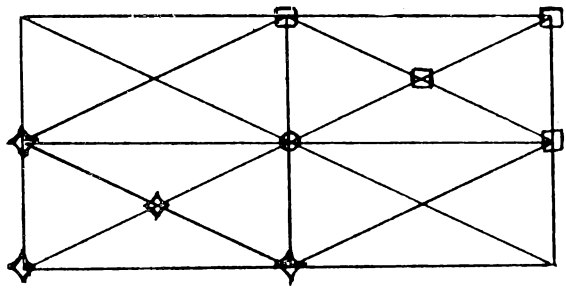
*Figure 9a.*

The defining level is $\{x_2, x_7, x_{12}\}$.
Choosing other neighbour nodes it may lead to another defining level.

1. $\tau(x_i) = 0$,
$i = 1, 2, \ldots, 13$;
$\tau(x_7) := 3$; $\tau(x_9) := 1$,
$\tau(x_5) := 2$.



2. $\tau(x_i) := 1$,
$i = 6, 11, 12$;
$\tau(x_i) := 2$,
$i = 2, 3, 8$.



3. $\tau(x_i) := 1$,
$i = 1, 4, 10, 13$;
$\tau(x_i) := 3$,
$i = 1, 4, 10, 13$.
$D = \{x_1, x_4, x_7, x_{10}, x_{13}\}$

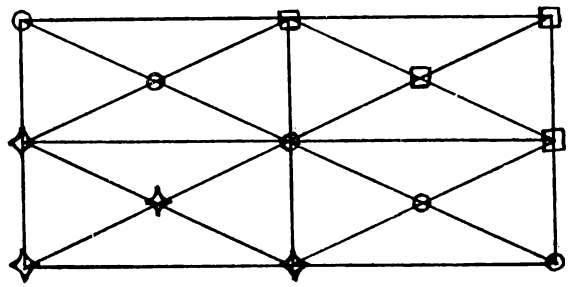


*Figure 9b.*

II Numbering system

We number the graph as if it consists of two disjoint graphs, making use of the numbering system described above for the Cuthill-McKee method (4):

1. We have already seen how to separate the nodes into three classes, so that

$$X = X_1 \cup D \cup X_2$$

Now we are going to assign the nodes in $D$ either to $X_1$ or to $X_2$, depending on the number of connections between them.

For arbitrary $x_1 \in D$, denote by $s_{i1}$ and $s_{i2}$ the number of edges starting at $x_i$ having their end-points in $X_1$ and $X_2$ respectively.

The assignment is made according to the following rule:

$$\text{if } s_{i1} > s_{i2} \text{ then } X_1 := X_1 \cup \{x_i\};$$
$$\text{if } s_{i1} < s_{i2} \text{ then } X_2 := X_2 \cup \{x_i\};$$
$$\text{if } s_{i1} = s_{i2} \text{ then } X_l := X_l \cap \{x_i\}, \quad l = 1, 2, 1, 2, 1, \ldots;$$

This then yields the form

$$X = X_1 \cup X_2$$

we go on now to number $X_1$, and $X_2$ separately, then fit them to each other by means of a suitable transformation.

2. Numbering $X_1$.
First of all, number the nodes in $X_1 \cap D$ in order of increasing value of $s_{i1}$, then the other nodes level by level using the numbering system detailed before. So all the levels $L_{t-1}, L_{t-2}, \ldots, L_1$ are numbered as follows

$$\alpha: \{x_{i1}, x_{i2}, \ldots, x_{im}\} \to \{1, 2, \ldots, m\},$$

where

$$\alpha(x_{i1}) = 1, \qquad x_{i1} \in X_1 \cup D,$$
$$\alpha(x_{im}) = m, \qquad x_{im} \in L_1;$$

3. Numbering $X_2$.
We start by numbering the nodes in $X_2 \cap D$ in order of increasing value of $s_{i2}$ using the numbers from $m+1$ to $N$. $L_{t+1}, L_{t+2}, \ldots, L_k$ are thus numbered just as before we got:

$$\bar{\alpha}: \{x_{j1}, x_{j2}, \ldots, x_{jn}\} \to \{m+1, m+2, \ldots, N\},$$

where

$$\bar{\alpha}(x_{j1}) = m+1, \qquad x_{j1} \in X_2 \cap D,$$
$$\bar{\alpha}(x_{jn}) = N, \qquad x_{jn} \in L_k.$$

4. Figure 10 illustrates the results obtained from steps 2. and 3., showing also the direction of the numbering.

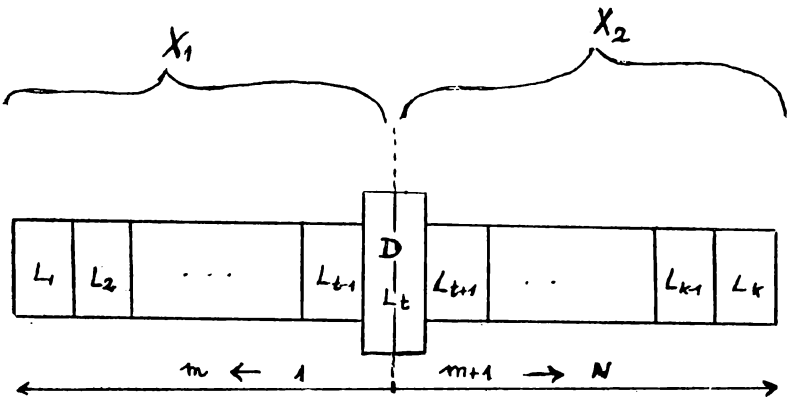



*Figure 10.*

The numbering of $X_1$ must now be reversed in order to satisfy the conditions:

$$\text{if } \tilde{\alpha}(x_i) = 1\,, \qquad \text{then } x_i \in L_1\,;$$
$$\text{if } \tilde{\alpha}(x_i) = m\,, \qquad \text{then } x_i \in X_1 \cap D\,.$$

That is, we need to apply a tansformation

$$\tilde{\alpha}: \{x_{i1}, x_{i2}, \ldots, x_{im}\} \to \{m, m-1, \ldots, 2, 1\}$$

Notice that for certain special graphs we can find an LS such that every level may be regarded as a defining level, so that the numbering of $X_1$ may proceed directly from 1 to m in the order $L_1, L_2, \ldots, L_{t-1}, X_1 \cap D$. In general, however, this will not be possible.

The algorithm described in sections I and II provides a numbering which corresponds to a specific point-pair neighbouring a specific starting node. In practice our method tries out all the nodes of maximum degree as starting nodes, along with all their suitable neighbouring pairs. Finally we select the optimum numbering for which $\delta$ takes its minimum value. Our method is illustrated by the flowchart in Figure 11.

## 4. Brief review of the computer program; computer results

The program is written in FORTRAN for the ICL 1905/E.
Our aim – apart from simply trying out the method – was to compare it with other methods (Cuthill–McKee, Modified Cuthill-McKee, Rosen) in terms of efficiency and time required. In all cases, the methods themselves and their important computational aspects that is, the determination of points of minimum or maximum degree, the selection of suitale neighbour point-pairs, the determination of the defining level, and so on are placed is separate subroutine segments. The MASTER segment is concerned only with control functions. First the paramter list is read. This list in effect controls the course of the program; for example, which methods will be applied to each graph. It is thus possible to apply every method to every graph and even combinations of methods as well.

Figure 12 shows the computer results of a comparison of 13 graphs relatively different in size and geometrical structure. Since using the Rosen' s method alone has been found to be inefficient and expensive in terms of computing time, we apply the constructive methods first, and then Rosen' s method on the result.

We consider therefore the following methods:

Cuthill-McKee (CM)
Modified Cuthill-McKee (MCM)
Our Method (OM)
Rosen's Method (RM)

**Draft of the algorithm**

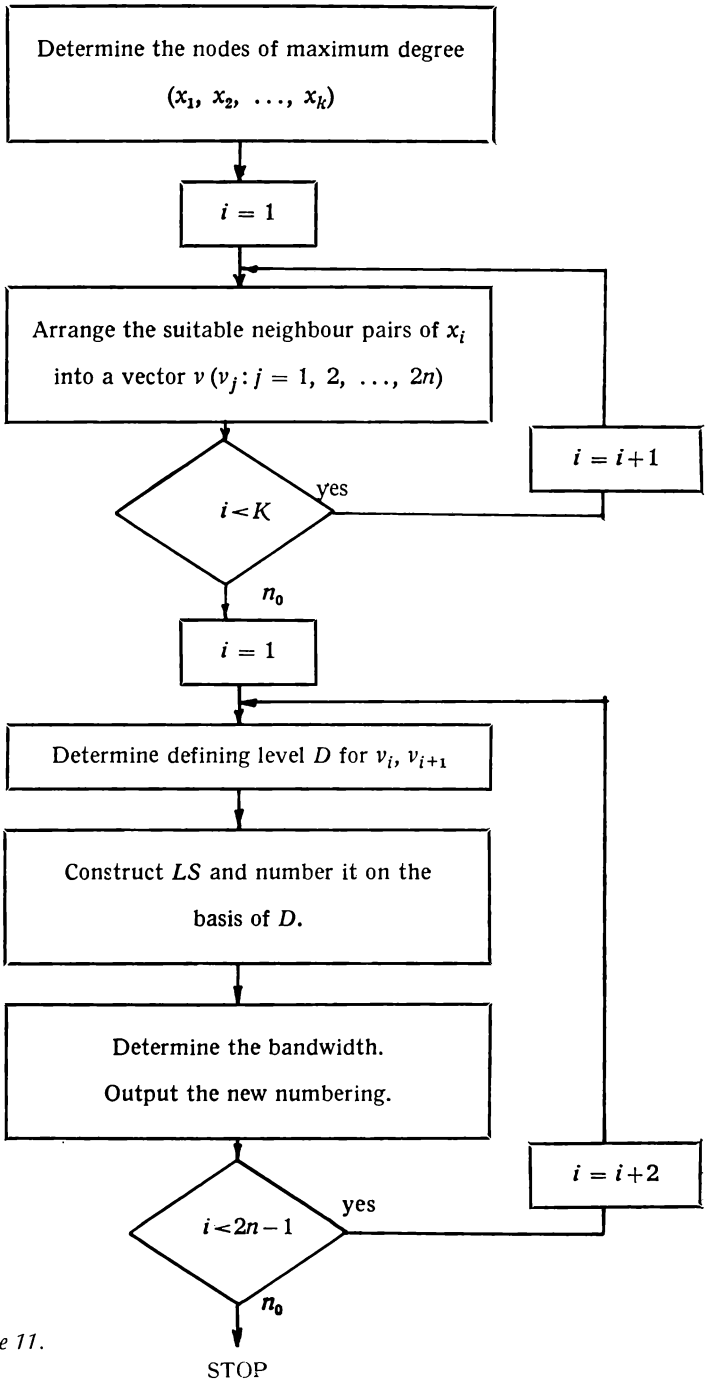



*Figure 11.*

applying them as follows:

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

From Figure 12 we see the computing time fluctuates widely.
For example:

Graph 2 (45 nodes; 85 edges) 1,65 secs (CM), 0,98 secs (OM);
Graph 7 (42 nodes; 81 edges) 0,24 secs (CM), 1,30 secs (OM).

These and other examples illustrate that the time depends on the structure of the graph.
CM seeme to be an efficient, fast method in cases when there are few points of minimum degree in the graph.

Our method produces more nearly optimum results, but it requires somewhat more time. In certain cases, such as the graph shown in Figure 3, OM is markebly better than CM.

To get a really objective comparison of the methods, we have tried to apply them on random graphs (given the number of nodes and edges, the connections are created by a random number generator). The results are shown in Figure 13: OM seems to be the best simple methos.
We should remark that the results are obtained by a slight modification of the original methods: namely, by modification of the condition

$$W_0 < \delta,$$

Let $\delta_k$ be the bandwidth obtained for the $k$th LS. If during the construction of the $(k+1)$th LS we find a level $L_i$ for which

$$\delta_k < W(L_i)$$

then the procedure returns to step I/3; that is, starts again to construct the next (or $(k+2)$th) defining level using the next pair of points. We thus ensure that we are looking only for a better $\delta$ than has already been obtained. Similarly, this midification has been made to Cuthill-McKee method. A significant decrease in computing time was obtained as a result for both methods.

Referring to Figure 12, the original bandwidth of graph 1 was also exactly the minimum possible bandwidth, so of course, no methods could improve it. It would be very useful to find a method which whould give exactly the minimum possible bandwidth.

Here again further experimentation is required.

**Computer results**

| GRAPH | Number of nodes | Number of edges | Original bandwidth | CM | | CM+RM | | MCM | | MCM+RM | | OM | | OM+RM | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | Bandwidth | Time (secs) | Bandwidth | Time (secs) | Bandwidth | Time (secs) | Bandwidth | Time (secs) | Bandwidth | Time (secs) | Bandwidth | Time (secs) |
| 1 | 19 | 34 | 18 | 4 | 0,34 | 4 | 0,44 | 4 | 1,74 | 4 | 1,74 | 4 | 0,48 | 4 | 0,74 |
| 2 | 45 | 85 | 36 | 5 | 1,65 | 5 | 2,82 | 5 | 7,9 | 5 | 9,0 | 5 | 0,98 | 5 | 2,2 |
| 3 | 26 | 39 | 4 | 4 | 0,54 | 3 | 1,16 | 4 | 2,44 | 3 | 2,53 | 4 | 0,74 | 3 | 1,48 |
| 4 | 14 | 24 | 7 | 5 | 0,12 | 4 | 0,28 | 5 | 0,66 | 4 | 1,04 | 5 | 0,68 | 4 | 1,1 |
| 5 | 16 | 16 | 15 | 2 | 0,70 | 2 | 0,83 | 2 | 0,70 | 2 | 0,84 | 2 | 0,3 | 2 | 0,44 |
| 6 | 14 | 27 | 6 | 6 | 0,14 | 5 | 0,46 | 3 | 0,66 | 3 | 0,99 | 3 | 0,6 | 3 | 0,92 |
| 7 | 42 | 81 | 37 | 8 | 0,24 | 7 | 8,4 | 8 | 1,3 | 7 | 3,8 | 7 | 1,3 | 7 | 14,5 |
| 8 | 7 | 11 | 6 | 3 | 0,14 | 3 | 0,16 | 3 | 0,22 | 3 | 0,29 | 3 | 0,34 | 3 | 0,4 |
| 9 | 14 | 23 | 4 | 5 | 0,66 | 4 | 0,90 | 5 | 0,93 | 4 | 1,16 | 4 | 0,48 | 4 | 0,82 |
| 10 | 25 | 43 | 6 | 5 | 0,18 | 5 | 0,51 | 5 | 2,4 | 5 | 4,7 | 6 | 0,78 | 5 | 1,46 |
| 11 | 32 | 42 | 31 | 9 | 0,18 | 8 | 0,74 | 9 | 3,47 | 8 | 7,07 | 7 | 1,06 | 6 | 3,64 |
| 12 | 91 | 240 | 37 | 12 | 4 | 12 | 27 | 12 | 39 | 12 | 39 | 12 | 58 | 12 | 15 |
| 13 | 99 | 169 | 89 | 24 | 8 | 17 | 22 | 24 | 41 | 17 | 80 | 17 | 47 | 13 | 65 |

**Random graphs**

| GRAPH | Number of nodes | Number of edges | Original bandwidth | CM | | MCM | | OM | |
|---|---|---|---|---|---|---|---|---|---|
| | | | | Bandwidth | Time (secs) | Bandwidth | Time (secs) | Bandwidth | Time (secs) |
| 1 | 20 | 60 | 19 | 12 | 0,094 | 11 | 0,578 | 9 | 1,210 |
| 2 | 20 | 60 | 17 | 13 | 0,046 | 10 | 0,554 | 9 | 0,912 |
| 3 | 40 | 100 | 35 | 21 | 0,180 | 17 | 1,284 | 15 | 2,696 |
| 4 | 40 | 100 | 37 | 22 | 0,084 | 17 | 1,032 | 15 | 1,024 |
| 5 | 50 | 162 | 47 | 29 | 0,134 | 26 | 2,118 | 24 | 2,612 |
| 6 | 50 | 162 | 48 | 28 | 0,214 | 27 | 1,974 | 26 | 2,718 |
| 7 | 60 | 150 | 57 | 29 | 0,324 | 27 | 3,188 | 22 | 3,324 |
| 8 | 60 | 150 | 57 | 32 | 0,152 | 26 | 3,004 | 25 | 4,028 |
| 9 | 80 | 216 | 79 | 38 | 0,728 | 36 | 6,018 | 35 | 8,428 |
| 10 | 80 | 216 | 77 | 40 | 0,626 | 35 | 7,210 | 34 | 9,526 |

## REFERENCES

[1] *E. Cuthill* and *J. McKee*, Reducing the Bandwidth of Sparse Symmetric Matrices, Proc. 24th Nat. Conf. ACM (1969) pp. 157–172.
[2] *L. Fox*, An Introduction to Numerical Linear Algebra, Clarendon Press (1964).
[3] *R. Rosen*, Matrix Bandwidth Minimization, Proc. 23rd Nat. Conf. ACM (1968) pp. 585–595.
[4] *G. Alway* and *D. Martin*, An Algorithm for Reducing the Bandwidth of a Matrix of Symmetrical Configuration, Comp. Jour. 8. (1965) pp. 264–272.
[5] *J. Alan George*, Computer Implementation of the Finite Element Method, Ph. D. Thesis (1971).
[6] *W. Smyth*, An Algorithm for Compressing a Sparse Matrix with Symmetrically-placed Elements into a Band, unpublished, (1966).
[7] *I. Arany – W. Smyth – L. Szóda*, An Improved Method for Reducing the Bandwidth of Sparse Symmetric Matrices, Proc. IFIP Cong. 71. (1971) Booklet – TA 1 pp. 6–10

MÜM SZÁMTI
1089 Budapest, Reguly A. u. 57–59.