Memoirs of the School of Engineering, Okayama University Vol.13, February 1979

A GRAPH-THEORETIC STUDY OF THE MINIMUM FILL-IN PROBLEM FOR SPARSE MATRIX METHOD

Takeo TANIGUCHI*

(Received February 1, 1979)

Synopsis

In this paper the minimum fill-in problem which arises at the application of the sparse matrix method for linear sparse systems is discussed from the graph-theoretic viewpoint and the author gives some results which can be directly introduced in the design of, so called, the optimal elimination ordering algorithm which gives the minimum fill-in(the number of zeros in coefficient matrix which become non-zero during the elimination process). Through this investigation only graphs are treated instead of the coefficient matrices for linear systems, and the elimination process for a matrix is equivalated to the vertx eliminations for the graph. Then, the results by the theoretical investigation are summarized as following:

1. Optimal elimination for each subgraph which is subdivided appropriately from whole graph leads to the global optimum.
2. In each subgraph there are only two kind of eliminations.

Furthermore, some numerical experiments show the characteristics of the subset of vertices, which subdivide a subgraph from the residual.

---

* Department of Civil Engineering

1. Introduction

The neccessity to solve linear equations

$$Mx=b \qquad (1)$$

, where M is a n × n sparse¶ symmetric positive definite matrix, arises frequently in structural analysis, and we have a number of efficient solvers which utilize the characteristics of M matrix. The most efficient solver for eq.(1) by elimination is called the sparse matrix method which uses only the non-zero entries in M.

On the other hand it is well known that the elimination process for eq.(1) produces additional non-zero entries in M, that is, some zeros in M become non-zero during eliminations. For example, by use of the i-th row any (j,k) element in M, m_{jk}, is altered to m^*_{jk},

$$m^*_{jk} = m_{jk} - \frac{m_{ji}m_{ik}}{m_{ii}}, \qquad (i < j < k) \qquad (2)$$

Even if $m_{jk} = 0$ in M, m^*_{jk} in the modified matrix,M*, becomes non-zero, when $m_{ij} \neq 0$ and $m_{ik} \neq 0$. Such a new non-zero entry in M* is called "Fill-in". Furthermore, the number of these entries depends on the ordering of eliminations for M. Thus, the utilization of the sparse matrix method requires the optimal elimination ordering which gives the minimum number of fill-in.

With M as in eq.(1) we can associate an undirected graph,G{X,E}, in which X and E denote the sets of vertices and edges in G, respectively, and the elimination process for M is equivalently transformed to the elimination of vertices in G.[1)] Then, eq.(2) is equivalent to following expression: Suppose three distinct vertices, i, j and k, in G, where j and k are adjacent to i but they(j and k) are not adjacent each other (these relations are denoted by $j,k \in$ adj.i and $j \notin$ adj.k). The elimination of i-vertex before the other gives the relation of $j \in$ adj.k in G* which is the graph after i-vertex elimination. As the relation, $x \in$ adj.y, shows the existence of an edge between two vertices, x and y, a fill-in in M shows an additional new edge for G* and, thus, G{X,E} is modified to G*{X,E'} , where $E \subseteq E'$, and the modification is restrictied to the subset of vertices which are adjacent to i-vertex.

¶ We say a matrix M sparse when many of its entries m_{ij} are zero.

Above graph-theoretic interpretation of "fill-in" was given by D.J. Rose whose aim is not for the design of Fill-in Minimization Algorithm but for the mathematical interest.(2) Through the fundamental study of the optimal elimination ordering from the graph-theoretic viewpoint the author intends to give valuable results which may be easily introduced in the design of Fill-in Minimization Algorithm.

2. Vertex Elimination Process

Suppose a Graph $G\{X,E\}$ for $M(n\times n)$, where X is a finite set of $|X| = n$ elements called vertices, and

$$E \subseteq \{\{x,y\} \mid x,y \in X,\ x \neq y\}$$

is a set of $|E|$ vertex pair called edges, which is equal to the number of non-zeros in the upper triangular matrix of M.

For M which represents one independent structural system we can give a connected graph, in which for each pair of distinct vertices, $x,y \in X$, there is a chain of edges from x to y.

According to eq.(1) or its graph-theoretic interpretation in Sec.1 the elimination of a vertex, $x \in X$, may produce some additional edges in the subgraph, $G_s\{X'\}$, where $X' \subset X$ and $X' = adj.x$. The influence of x-vertex elimination is restricted only in G_s, and we denote the subgraph by FVG which is the abbreviation of Frontal Vertex Group and which locates at the front of the eliminated vertices.

The number of vertices in FVG is determined by the number of vertices which are adjacent to the eliminated vertex,x. Furthermore, the subgraph, FVG, constructs a complete graph, in which every pair of vertices is adjacent, and the number of edges is equal to

$$|adj.x|(|adj.x|-1)/2$$

, where $|adj.x|$ is the number of vertices adjacent to x. If we denote the set of these additional edges by F, then the modified graph, G^*, is equal to $G^*\{X, E \cup F\}$.

Suppose a modified graph, G^*, which is obtained after some stages of vertex elimination in accordance with an arbitrary elimination ordering for G. At this stage there may exist several independent FVG's in G^*, and all the vertices in the graph are divided into two sets, one of which is the set of eliminated vertices and another is non-eliminated including, of course, the vertices in FVG's.

If the subgraph, G^*_N, consisting of the latter set of vertices and also all edges connecting them is removed from G*, there remain, in general, a number of connected subgraphs, denoted by G_E. Let's denote G_N for the subgraph which is obtained by removing subgraphs, FVG's, from G_N^*. Now, we begin to investigate the influence of one vertex elimination to G*.

The vertex, x, being eliminated at this stage must belong to one of followings as shown in Fig. 1.

1. $x \in G_N$, $x \notin$ FVG
2. $x \in$ FVG, adj.x $\in G_N$
3. $x \in$ FVG, adj.x $\subset$ FVG
4. $x \in$ FVG's, adj.x $\notin G_N$
5. $x \in$ FVG's, adj.x $\in G_N$

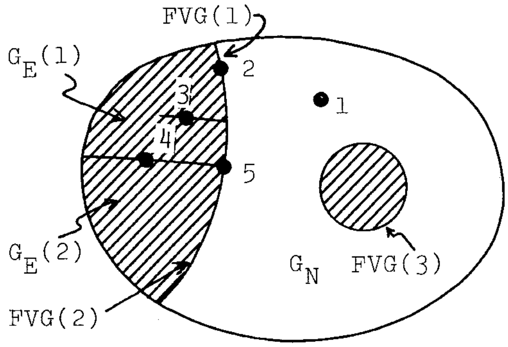

Fig. 1 A Stage of Elimination Process on G

Type 1.

Elimination of $x \in G_N$ makes the subgraph consisting of all vertices adjacent to x complete, and they construct a new FVG. Thus, fill-in, F, is obtained by eq.(3).

$$F = |adj.x| \cdot (|adj.x| - 1)/2 - |E_0| \tag{3}$$

, where |adj.x| is the number of vertices adjacent to the eliminated vertex, x, and $|E_0|$ is the number of edges in FVG $\cap$ G.

Type 2.

This type of vertex elimination does only change the entries of the FVG. That is, adj.x become new entries of FVG instead of x just eliminated.

$$\text{FVG}\{X'\} \xrightarrow{\text{elimination of } x} \text{FVG*}\{(X'-x) \cup adj.x\}$$

Fill-in appearing at the elimination is divided into two parts ; for the completeness of the new entries, $y \in$ adj.x and $y \notin$ FVG, and for the completeness between $\{y \in adj.x\}$ and $\{X' - x\}$. Then, the summation of these two cases gives F, which denotes the number of fill-in of Type 2-vertex elimination.

$$F = \frac{|y| \cdot (|y| - 1)}{2} + |y| \cdot |X' - x| - |E_1| - |E_2| \tag{4}$$

, where |y| and |X' - x| are the number of vertices in {y} and {X'-x}, respectively, and $|E_1|$ and $|E_2|$ are the number of edges in {y}at the previous stage and the edges connecting two subsets, {y} and {X'-x}, respectively.

Type 3.

The x-vertex elimination does only decrease the number of vertices in FVG by one and no ther influence appears. As the vertex, x, is a member of FVG and adj.x $\subset$ X', no fill-in occures at the elimination.

$$F = 0 \tag{5}$$

Type 4.

Typical case of this type is the vertex, x, locating on two FVG's, FVG(1) and FVG(2). Thus, the vertex elimination produces new one FVG by joining FVG(1) and FVG(2). Any vertex, $y \in$ FVG(1) $\cap$ FVG(2), are adjacent to all vertices in FVG(1) $\cup$ FVG(2), and only the set of vertices, $\{z \mid z \in \mathrm{FVG}(1) \cup \mathrm{FVG}(2) - \mathrm{FVG}(1) \cap \mathrm{FVG}(2)\}$, decides the value of fill-in.

$$F = |\mathrm{FVG}(1) - \mathrm{FVG}(1) \cap \mathrm{FVG}(2)| \cdot |\mathrm{FVG}(2) - \mathrm{FVG}(1) \cap \mathrm{FVG}(2)| \tag{6}$$

When we eliminate such a vertex, x, as

$$x \in \bigcap_{i=1}^{n} \mathrm{FVG}(i) \text{ for } n > 2, \tag{7}$$

the elimination requires more fill-in than F in eq.(6). The fact that the influence of any vertex elimination is restricted to the subset of vertices which are adjacent to the vertex and that our aim of this investigation is the minimization of fill-in lead to the conclusion that we should not eliminate this type of vertex, x, as shown in eq.(7).

Type 5.

By the x-vertex elimination all vertices adjacent to x construct a complete graph, that is, all the FVG's in which the vertex, x, belongs are joined, and they form new FVG, and it includes such vertices as $\{y \in \mathrm{adj.x},\ y \notin \mathrm{FVG's}\}$, too.

Comparing the vertex elimination of Type 5 with Type 4, it is evident that the former gives more fill-in than the latter. Therefore, Type 5 need not be considered for our purpose.

As far as our concern is the process of the optimal vertex elimination for the minimum fill-in, we may consider only four types of vertex eliminations for any graph, namely Type 1,2,3 and 4.

3. Vertex Elimination for Minimum Fill-in

This section aims to clarify the general characteristics of modified graph appearing at every stage of optimal vertex elimination process.

In the preceding section we obtained that all the vertices in G are eliminated by use of, at most, five types of vertex eliminations during arbitrary ordering, and also that only four types among them may appear when the elimination ordering is optimal. They are Type 1, 2, 3 and 4.

As far as we treat a connected graph G{X} with $|X| \geqslant 2$, Type 1 is insufficient for eliminations of all vertices in G, and $x \in$ FVG which appear after Type 1 eliminations must be treated by the other Types.

If more than two vertices in G are eliminated by Type 1, then G* has, evidently, as many FVG's as the number of eliminated vertices treated by Type 1, and the process requires Type 4-elimination. But we can eliminate all the vertices in G by use of only Type 1 and Type 2, when Type 1-vertex elimination is applied for only one vertex. Above consideration leads to the conclusion that the maximum number of independent FVG's are decided by the number of the vertices which are eliminated by Type 1.

Here we have to prove that there appear several independent FVG's at a stage of optimal elimination process, that is, the process requires, in general, Type 1, 2 and 4-vertex eliminations.

If Type 1 and 2 are sufficient for any G to give minimum fill-in, the process requires only one FVG and the vertex which should successively eliminated must be always selected among $x \in$ FVG at the stage. Suppose a stage of optimal elimination for G in which there exists only one FVG. If there is a vertex x ($x \in G_N$, adj.x $\in$ FVG, and adj.x $\cap \{G_N - x\} = \phi$), then the optimal process requires the elimination of x before the eliminations of $y \in$ FVG. That is, we find the second FVG. Therefore, we can conclude that the optimal process requires, in general, several independent FVG's as shown in Fig.1.

Every FVG(i) divides the subgraph $G_E(i)$, which consists of only eliminated vertices, in G*, and the vertex eliminations of the subgraph gives no influence to the other region. Therefore, all vertices in the subgraph may be successively eliminated. That is, if a FVG at a stage of the optimal process is obtained, the optimization of vertex elimination for the subdivided area leads to the global optimum.

These results allow us to image the optimal elimination process. The first vertex elimination on G{X} is, of course, Type 1 which produces one FVG. If vertices for successive eliminations are always selected among $x \in$ FVG at every stage, G_E is a connected subgraph and only one FVG exists in G*. These successive eliminations belong to Type 2. After some steps the process does not select a vertex $x \in$ FVG for successive elimination but $y \notin$ FVG. That is, the elimination of $x \in$ FVG does not lead to optimum. Then, the FVG stops its growth and we denote it by FVG(1).

The selection of a vertex, $y \notin$ FVG(1), produces another FVG(2), and some Type 2 vertex elimination steps for vertices in FVG(2) yield to

$$\text{FVG}(1) \cap \text{FVG}(2) \neq \phi \qquad (8)$$

Any vertex $x \in \text{FVG}(1) \cap \text{FVG}(2)$ is selected for the next elimination , and two FVG's are joined into one FVG. Successive eliminations are for these vertices, $x \in \text{FVG}(1) \cap \text{FVG}(2)$ and adj.$x \notin G_N$, and through the steps we have no fill-in, because they belong to Type 3.

Through the elimination process the growth of FVG and joining of FVG's are repeated, and at the last state of the process where $G_N = \phi$, the eliminations are only for $x \in$ FVG's. From this stage we apply only Type 3-elimination.

4. Further Considerations on Frontal Vertex Group

The most important factor through the optimal vertex elimination process is the characteristics of Frontal Vertex Group, especially of the FVG at the stage when it stops its growth, and also when it was produced by joining two FVG's.

For the investigation of the characteristics of FVG's we refer to eq's(3), (4), and (6). These equations suggest that the number of vertices in the newly obtained FVG is the most effective factor for the evaluation of fill-in. That is, in order to minimize fill-in through the elimination process |FVG| should be always kept as small as possible.

For each vertex in G we can obtain the minimum FVG independently from the other region, but as the elimination process modifies the graph at every stage and also some other factors, for example, E_0 in eq.(3) give influence to the value of fill-in, it seems to be very difficult to decide the FVG which stoped its growth.

Fig. 2 show the results of some numerical experiments which are done in order to obtain effective factors for the determination of the maximum FVG's through the optimal elimination process. The term, the maximum FVG, means the FVG which stoped its growth during the process. Each thick line connecting vertices which are shown by ● in the figures indicates the location of one FVG, and every subgraph divided by FVG is eliminated in accordance with the label of the area. All the vertices in each subgraph are eliminated by use of Type 1 and 2, and the vertices on FVG's are by Type 3 and 4.

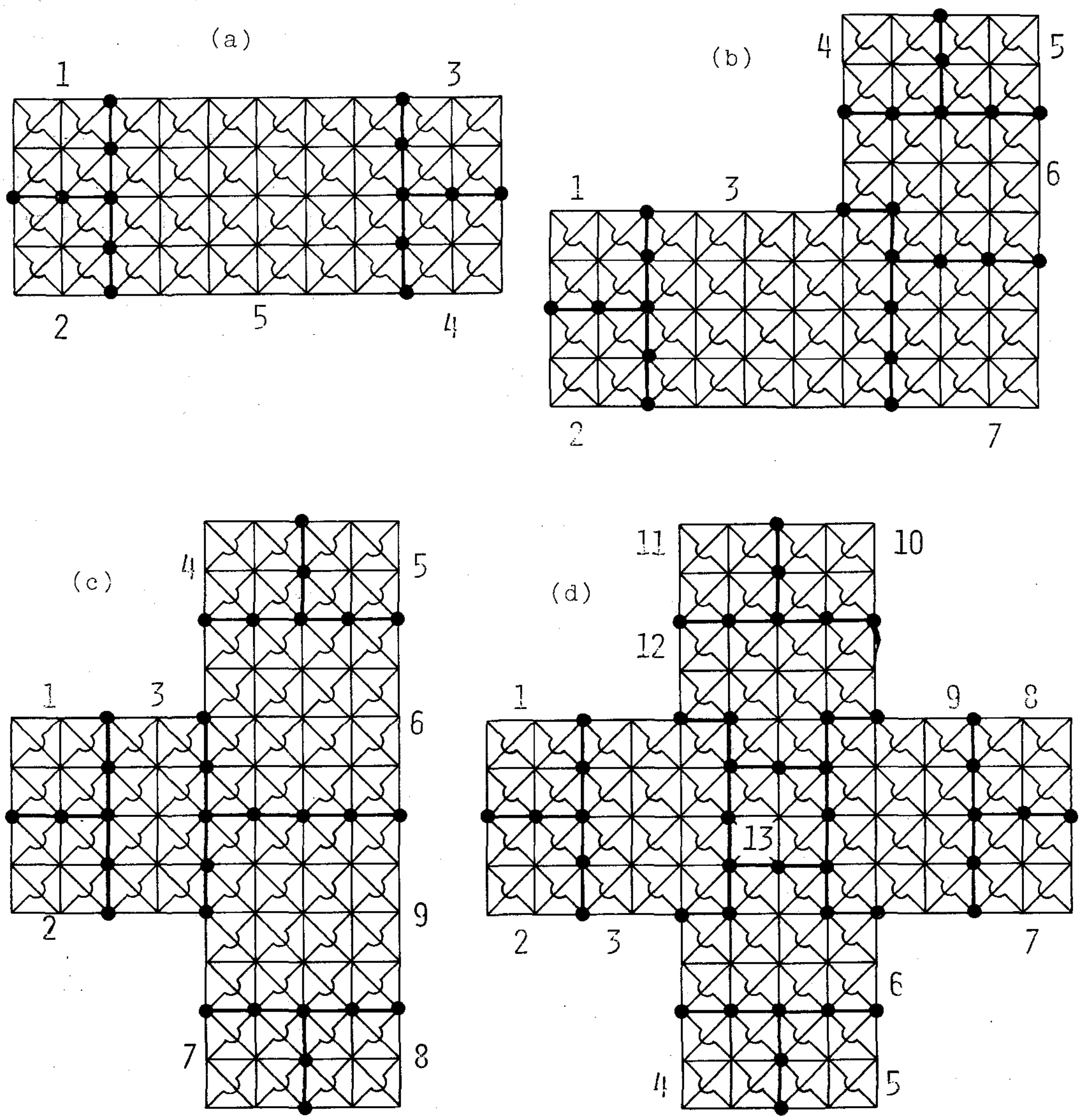

Fig. 2 FVG's on Graphs by Numerical Experiments

The optimal elimination process of Fig.2-a, for example, is as following: After the eliminations of all the vertices in Area 1 and 2, vertices of FVG's locating between these areas are eliminated and residual vertices on these two FVG's form a new FVG. The same procedure is repeated for Area 3 and 4, and vertices on FVG which divides Area 5 from Area 3 and 4 must be eliminated. This is only one of optimal processes and we can easily find another optimal process. For example, after the eliminations of Area 1 and 2, we may treat vertices in Area 5. But, before the elimination of Area 5 two FVG's locating between Area 1 and 2 must be joined into one FVG and the deal of this new FVG precedes that of Area 5. Futhermore, the elimination process for Area 5 must stop at the location of FVG which subdivides Area 5 from Area 3 and 4.

Any FVG obtained in these numerical experiments has following property.

$$|FVG| \leqslant |\text{Minimal } x,y \text{ separator}| \qquad (9)$$

, where $x \in G_E$ and $y \in G_N$. And Minimal separator is a separator, which separates any graph into two connected subgraphs and no, subset of which is also a separator of the graph.[2]

Furthermore, these numerical experiments suggest that the maximum of $|FVG|$ appearing during the vertex elimination process satisfies eq.(10).

$$\text{Max.}|FVG| \leqslant \text{Max. of Min.}\{\text{Minimal } x,y \text{ separator}\} \qquad (10)$$

, where $x,y \in X$ in G and $x \notin \text{adj}.y$.

Eq.(10) shows the location of the FVG which appears at the last stage of elimination process. Furthermore, from these numerical results we can recognize that one FVG stops its growth, when the FVG includes a chain of vertices which subdivides the graph and which locates at the place where "the width" of the graph changes. Eq.(9) is the general expression of this fact.

5. Concluding Remarks

Through this investigation for the minimum fill-in proble, following results were obtained:

1. Optimization of the vertex elimination process for subgraphs obtained by the appropriate subdivision of whole graph,G, leads

to the optimization for G.

2. For the tool of the subdivision of G "the width of G" seems to be useful.
3. Through the optimal vertex elimination process any vertex is eliminated by use of one of four types of eliminations. Two types of them deal the vertices in each subgraph, and the other are used for vertices in the separators(i.e. FVG's) of G.

Above results may surely make the design of the algorithm of optimal vertex elimination ordering ease, and the results obtained by the algorithm will be quite different from the orderings by Minimum Deficiency Algorithm[(3)] which is thought as the best now.

Acknowledgement

The author wishes to acknowledge the assistance of Mr. K. Yasuda, undergraduate student of Okayama University, who did the most of the numerical experiments given in this paper.

References

(1) D.J.Rose: Graph Theory and Computing(edited by R.C.Read), (1972), 183-217
(2) D.J.Rose, Ref. (1), 193
(3) D.J.Rose, Ref. (1), 215