

A Quotient Graph Model for
Symmetric Factorization[†]

by

Alan George
and
Joseph W.H. Liu

Department of Computer Science
University of Waterloo
Waterloo, Ontario, Canada
N2L 3G1

Research Report CS-78-04

[†] Research supported in part by Canadian National Research Council
grant A8111.

Abstract

Consider the computation of the Cholesky factorization LL^T of the symmetric positive definite matrix A . In this paper we present a model of this factorization algorithm based on quotient graphs, and discuss its relationship to existing models. The primary advantage of our model is that its computer implementation is very efficient. In particular, we show how it can be implemented in space proportional to the number of nonzeros in A .

1. Introduction

Consider the symmetric positive definite system of linear equations

$$(1.1) \quad Ax = b ,$$

where the N by N coefficient matrix A is sparse. If Cholesky's method is to be used to solve (1.1), there are compelling reasons to proceed in four distinct steps as follows:

Step 1 (Ordering Step) Find a "good" ordering for the system, and transform it, perhaps only implicitly, to

$$(1.2) \quad (PAP^T)(Px) = Pb .$$

Step 2 (Symbolic Factorization) Determine the location of the nonzeros in L , where $PAP^T = LL^T$, and set up the appropriate data structures for L .

Step 3 (Numerical Factorization) Decompose PAP^T into LL^T , using Cholesky's method.

Step 4 (Triangular Solution) Solve $Ly = Pb$, $L^Tz = y$, and then set $x = P^Tz$.

Our objective in this paper is to present a model of symmetric factorization which is useful in the analysis and implementation of Steps 1 and 2. Sparse matrices normally suffer some fill-in when they are factored, so that $L + L^T$ has nonzeros in positions which are zero in A . The objective of Step 1 is usually to find a P so that this fill-in is acceptably low. Several ordering algorithms, such as the minimum degree algorithm [4, 9], essentially require that the factorization be simulated as the ordering proceeds, because ordering decisions depend

upon the structure of the partially factored matrix. Thus, it is crucial to be able to simulate the factorization efficiently. Similarly, the implementation of Step 2 above also depends upon being able to simulate the factorization efficiently, in terms of both space and time.

An outline of the paper is as follows. In Section 2 we review some existing models of symmetric elimination (factorization), and discuss some of their important features. In Section 3 we present our quotient graph model, and compare it to those of Section 2. In Section 4 we show that the space required for a computer implementation of the model is proportional to the number of nonzeros in A (that is, independent of the fill-in). Section 5 contains a discussion of some applications of the model, along with our concluding remarks.

2. Existing Models or Characterizations of Symmetric Factorization

2.1 Matrix Formulation

As a point of departure, we describe the factorization in terms of the actual numerical computation that is performed. The models we consider subsequently are used to simulate what happens in terms of structural changes to the matrices, without actually involving any numerical computation. Setting $A = A_0 = H_0$, the factorization of A can be described by the following equations.

$$\begin{aligned}
 A_0 &= \begin{pmatrix} d_1 & v_1^T \\ v_1 & \bar{H}_1 \end{pmatrix} = \begin{pmatrix} \sqrt{d_1} & \circ \\ v_1/\sqrt{d_1} & I_{N-1} \end{pmatrix} \begin{pmatrix} 1 & 0 \\ 0 & \bar{H}_1 - v_1 v_1^T/d_1 \end{pmatrix} \begin{pmatrix} \sqrt{d_1} & v_1^T/\sqrt{d_1} \\ \circ & I_{N-1} \end{pmatrix} \\
 &= L_1 A_1 L_1^T \\
 A_1 &= \begin{pmatrix} 1 & & 0 \\ & d_2 & v_2^T \\ 0 & v_2 & \bar{H}_2 \end{pmatrix} = \begin{pmatrix} 1 & & \circ \\ & \sqrt{d_2} & \\ 0 & v_2/\sqrt{d_2} & I_{N-2} \end{pmatrix} \begin{pmatrix} 1 & 0 \\ 0 & 0 \\ & H_2 \end{pmatrix} \begin{pmatrix} 1 & 0 \\ & \sqrt{d_2} & v_2^T/\sqrt{d_2} \\ & \circ & I_{N-2} \end{pmatrix} \\
 &\vdots \\
 A_n &= L_N I_N L_N^T .
 \end{aligned}$$

It is straightforward to show that $A = LL^T$, where

$$L = \left(\sum_{k=1}^N L_k \right) - (N-1)I_N .$$

Here I_k denotes a k by k identity matrix, d_k is a positive scalar, and v_k is a vector of length $N-k$. The matrix H_k is an $N-k$ by $N-k$ symmetric positive definite matrix, which we refer to as the part of A remaining to be factored after the first k steps of the factorization have been performed.

2.2 Some Elementary Graph Theory Terminology

The models we discuss in subsequent parts of this paper rely heavily on graph theory terminology, so in this section we introduce some of the essential definitions and notions. For our purposes, a graph $G = (X, E)$ consists of a finite nonempty set X of nodes together with a prescribed edge set E of unordered pairs of distinct nodes. A graph $G' = (X', E')$ is a subgraph of G if $X' \subset X$ and $E' \subset E$. For $Y \subset X$, $G(Y)$ refers to the subgraph $(Y, E(Y))$ of G , where $E(Y) = \{\{u, v\} \in E \mid u, v \in Y\}$.

Nodes x and y of X are adjacent if $\{x, y\} \in E$. For $Y \subset X$, the set of nodes adjacent to Y is defined and denoted by

$$\text{Adj}(Y) = \{x \in X - Y \mid \{x, y\} \in E \text{ for some } y \in Y\} .$$

If $Y = \{y\}$, we write $\text{Adj}(y)$ rather than $\text{Adj}(\{y\})$. The degree of a node x is simply the number of nodes adjacent to it, denoted by $\text{deg}(x)$. We refer to $y \in \text{Adj}(x)$ as a neighbor of x . A set of nodes $Y \subset X$ which are pairwise adjacent is a clique.

A path of length ℓ is an ordered set of distinct vertices $(v_0, v_1, \dots, v_\ell)$ where $v_i \in \text{Adj}(v_{i-1})$ for $1 \leq i \leq \ell$. A graph G is connected if there is a path connecting each pair of distinct nodes. If G is disconnected, it consists of two or more maximal connected components.

Let $S \subset X$, and $w \in X - S$. The node w is said to be reachable from y through S if there exists a path $(y, y_1, y_2, \dots, y_k, w)$ such that $y_i \in S$, for $1 \leq i \leq k$. We allow k to be zero, so any node $w \in X - S$ and adjacent to y is reachable from y . The reach set

of y through S is then denoted and defined by

$$\text{Reach}(y, S) = \{w \in X-S \mid w \text{ is reachable from } y \text{ through } S\} .$$

We extend this definition to subsets Y as follows. Let $Y \subset X$ and $Y \cap S = \phi$. The reach set of Y through S is then

$$\text{Reach}(Y, S) = \{w \in X-(S \cup Y) \mid w \text{ is reachable from some node } y \in Y \text{ through } S\} .$$

Note that when $S = \phi$, $\text{Reach}(Y) = \text{Adj}(Y)$.

In subsequent sections of this paper, we will be applying these definitions to various graphs. When the graph being referred to is not absolutely clear from context, we will put the appropriate subscript on the definition. Thus, notations of the following type will be used: $\text{Adj}_G(Y)$, $\text{deg}_G(Y)$, $\text{Reach}_G(Y, S)$, etc.

2.3 Elimination Graph Model

In this section we describe the graph theory approach to symmetric elimination which was introduced by Parter [8], and popularized and extended by Rose [9]. Let A be an N by N symmetric matrix. The labelled undirected graph of A , denoted by $G^A = (X^A, E^A)$, is one for which X^A is labelled from 1 to N :

$$X^A = \{x_1, x_2, \dots, x_N\},$$

and $\{x_i, x_j\} \in E^A$ if and only if $A_{ij} \neq 0$. For any N by N permutation matrix P , the unlabelled graphs of A and PAP^T are the same, but the associated labellings differ. Thus, the graph of A is a convenient vehicle for studying the structure of A , since no particular ordering is implied by the graph.

Now consider the symmetric factorization of A into LL^T using the algorithm described in Section 2.1. The sparsity changes (fill-in) can be modelled by a sequence of graph transformations on G^A . Let $G = (X, E)$ be a graph and $y \in X$. The elimination graph of G by y , denoted by G_y , is the graph

$$(X - \{y\}, E(X - \{y\}) \cup \{\{u, v\} \mid u, v \in \text{Adj}(y)\}) .$$

In words, G_y is obtained from G by deleting y and its incident edges, and then adding any edges to the remaining graph so that the set $\text{Adj}(y)$ is a clique. This recipe is due to Parter [8].

With this definition, the process of symmetric elimination on A can be modelled as a sequence of elimination graphs

$G_0, G_1, G_2, \dots, G_{N-1}$, where

$$G_0 = G^A ,$$

and

$$G_i = (G_{i-1})_{X_i} = (X_i, E_i) , \quad i = 1, 2, \dots, N-1 .$$

Here $X_i = \{x_{i+1}, x_{i+2}, \dots, x_N\}$, and it is straightforward to verify that

$$G_i \equiv G^{H_i} ,$$

where H_i (defined in Section 2.1) is the part of the matrix remaining to be factored after step i of the factorization has been completed. Thus, this model is quite explicit; the structure of G_i corresponds directly to the matrix H_i .

2.4 The Element (Clique) Model for Symmetric Elimination

We saw in the previous section that in graph theory terminology, elimination of variable x_i in G_{i-1} in general generates a clique in G_i . In finite difference and finite element applications, the initial graph G_0 of the matrix can be viewed in a natural way in terms of its clique or "element" structure. For these reasons, it is natural to model the factorization process in terms of the clique structure in the elimination graphs G_0, G_1, \dots, G_{N-1} . This approach has been useful in both theoretical and computer applications [1, 2, 7].

As before, let $G^A = (X^A, E^A)$ be the labelled graph of a matrix A , and let $K^0 = \{K_1^0, K_2^0, \dots, K_k^0\}$ be a collection of (not necessarily maximal) cliques of G such that $X^A = \bigcup_{i=1}^{k_0} K_i^0$. Now we want to define a sequence of clique sets K^i , $i = 0, 1, 2, \dots, N-1$ such that K^i correctly reflects the structure of the elimination graph G_i of Section 2.4. It is straightforward to show that K^i is obtained from K^{i-1} according to the following recipe:

- a) Cliques in K^{i-1} not containing x_i remain unchanged in K^i .
- b) Cliques in K^{i-1} containing x_i merge to form a new clique, containing the union of their nodes, except for x_i which is removed from the resulting new clique.

Note that in step b), x_i may belong to only one clique, say K_ℓ^{i-1} , and the net effect of the transformation from K^{i-1} to K^i is to remove x_i from K_ℓ^{i-1} . Note also that if x_i belongs to more than one clique in K^{i-1} , then $|K^i| < |K^{i-1}|$.

Implementations using this model are efficient, compared to those based on the elimination graph model, when the elimination graphs have relatively large cliques, and when the intersection between the cliques is relatively small. Finite element matrix problems, appropriately ordered, typically have these properties [7].

2.5 Characterization of Symmetric Elimination in Terms of Reachable Sets

Again let $G^A = (X^A, E^A)$ be the graph of A , and let $G^F = (X^F, E^F)$ be the graph of $L + L^T$. We call G^F the filled graph of G^A , where $X^F = X^A$ and E^F consists of E^A together with those edges added during the factorization. The relationship between E^A and E^F is contained in the following lemma due to Parter [8].

Lemma 2.2 The unordered pair $\{x_i, x_j\} \in E^F$ if and only if $\{x_i, x_j\} \in E^A$, or there exists some $k < \min\{i, j\}$ for which $\{x_i, x_k\} \in E^F$ and $\{x_j, x_k\} \in E^F$.

□

The above lemma is somewhat unsatisfactory because it is recursive in E^F . The following result, which is essentially a restatement in our notation of a lemma due to Rose et. al. [10], relates E^F directly to E^A . Let $S_i = \{x_1, x_2, \dots, x_i\}$, $i = 1, \dots, N$, with $S_0 = \phi$.

Lemma 2.3 [4] Let $j > i$. Then the unordered pair $\{x_i, x_j\} \in E^F$ if and only if $x_j \in \text{Reach}_G(x_i, S_{i-1})$.

□

Thus, the sets S_i together with the Reach operator precisely characterize the adjacency structure of the elimination graphs G_i . In particular, we have

Lemma 2.4 [4] Let y be a node in the elimination graph $G_i = (X_i, E_i)$. The set of nodes adjacent to y in G_i is given by $\text{Reach}_G(y, S_i)$.

□

Note that in Lemma 2.3, and Lemma 2.4, the Reach operator is applied to the original graph G .

3. The Quotient Graph Model

3.1 Notation and Definitions

Let $G = (X, E)$ be a graph with X the set of nodes and E the set of edges. For a subset $S \subset X$, $G(S)$ will be used to refer to the subgraph $(S, E(S))$ of G , where

$$E(S) = \{\{u, v\} \in E \mid u, v \in S\} .$$

The central notion in the new model is that of a quotient graph [3], which we now review. Let P be a partitioning of the node set X :

$$P = \{Y_1, Y_2, \dots, Y_p\} .$$

That is, $\bigcup_{k=1}^p Y_k = X$ and $Y_i \cap Y_j = \phi$ for $i \neq j$. The quotient graph of G with respect to P is defined to be the graph (P, \mathcal{E}) , where $\{Y_i, Y_j\} \in \mathcal{E}$ if and only if $Y_i \cap \text{Adj}(Y_j) \neq \phi$. This graph will be denoted by G/P .

Consider the example in Figure 3.1. If $P = \{\{a, b, c\}, \{d, e\}, \{g\}, \{f, h\}\}$ is the partitioning, the quotient graph G/P is given as shown.

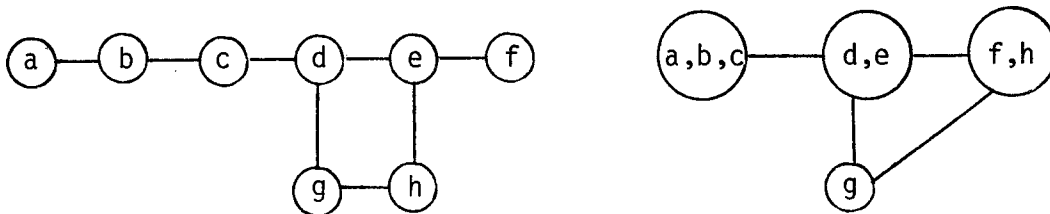


Figure 3.1 A quotient graph

An important type of partitioning is that defined by connected components. Let S be a subset of the node set X . The component partitioning $C(S)$ of S is defined as

$$C(S) = \{Y \subset S \mid G(Y) \text{ is a connected component in the subgraph } G(S)\} .$$

When $S = X$, $C(X)$ simply contains the component sets in the graph G . Therefore, the corresponding quotient graph $G/C(X)$ consists of $|C(X)|$ isolated nodes.

The structure of the quotient graph $G/C(X)$ is not particularly interesting. However, we now study a closely related type of partitioning, which turns out to be quite relevant in the modelling of Gaussian elimination. Again let S be a subset of X . The partitioning on X induced by the subset S is defined to be

$$\bar{C}(S) = C(S) \cup \{\{x\} \mid x \notin S\} .$$

That is, the partitioning $\bar{C}(S)$ consists of the component partitioning of S and the remaining nodes of the graph G .

Consider the graph in Figure 3.1. Let S be the subset $\{a, b, d, f, g\}$. It can be seen that

$$C(S) = \{\{a, b\}, \{d, g\}, \{f\}\} ,$$

so that $\bar{C}(S)$ has six members and is given by

$$\bar{C}(S) = \{\{a, b\}, \{d, g\}, \{f\}, \{c\}, \{e\}, \{h\}\} .$$

In this case, the quotient graph $G/\bar{C}(S)$ is given in Figure 3.2. Here, we use double circles to indicate those partition members in $C(S)$.

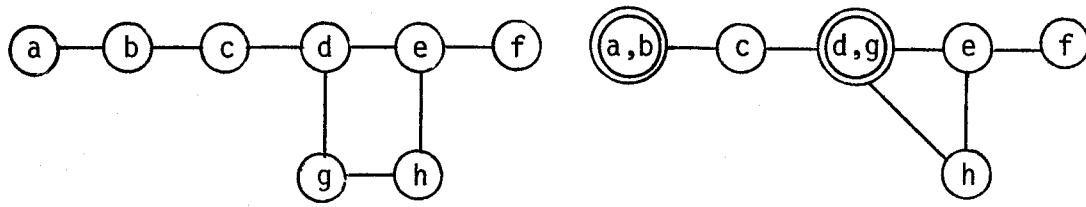


Figure 3.2 The quotient graph $G/\overline{C}(S)$

3.2 The Model

In Section 2, we have reviewed the various existing models of symmetric elimination. These models are used to study the Gaussian elimination process. In this section, we introduce a new model using the notion of quotient graphs. Its relationship with those in Section 2 will also be discussed.

Consider the symmetric factorization of a matrix A into LL^T . Recall from Section 2 that the elimination process applied to A can be interpreted as a sequence of elimination graphs

$$G_0, G_1, \dots, G_{N-1} .$$

The graph G_i precisely reflects the structure of the matrix remaining to be factored after the i -th step of the Gaussian elimination.

The new model represents the process as a sequence of quotient graphs, which may be regarded as implicit representations of the elimination graphs $\{G_i\}$. Let $G = (X, E)$ be the graph and x_1, x_2, \dots, x_N be the sequence of node elimination.

As in Section 2.5, let $S_i = \{x_1, \dots, x_i\}$ for $1 \leq i \leq N$ and $S_0 = \phi$. Consider the partitioning $\overline{C}(S_i)$ induced by S_i , and the corresponding quotient graph $G/\overline{C}(S_i)$. We shall denote this quotient graph by G_i . In this way, we obtain a sequence of quotient graphs

$$G_1, G_2, \dots, G_N .$$

Figure 3.3 contains an example. Partition members in $G(S_i)$ are marked in double circles.

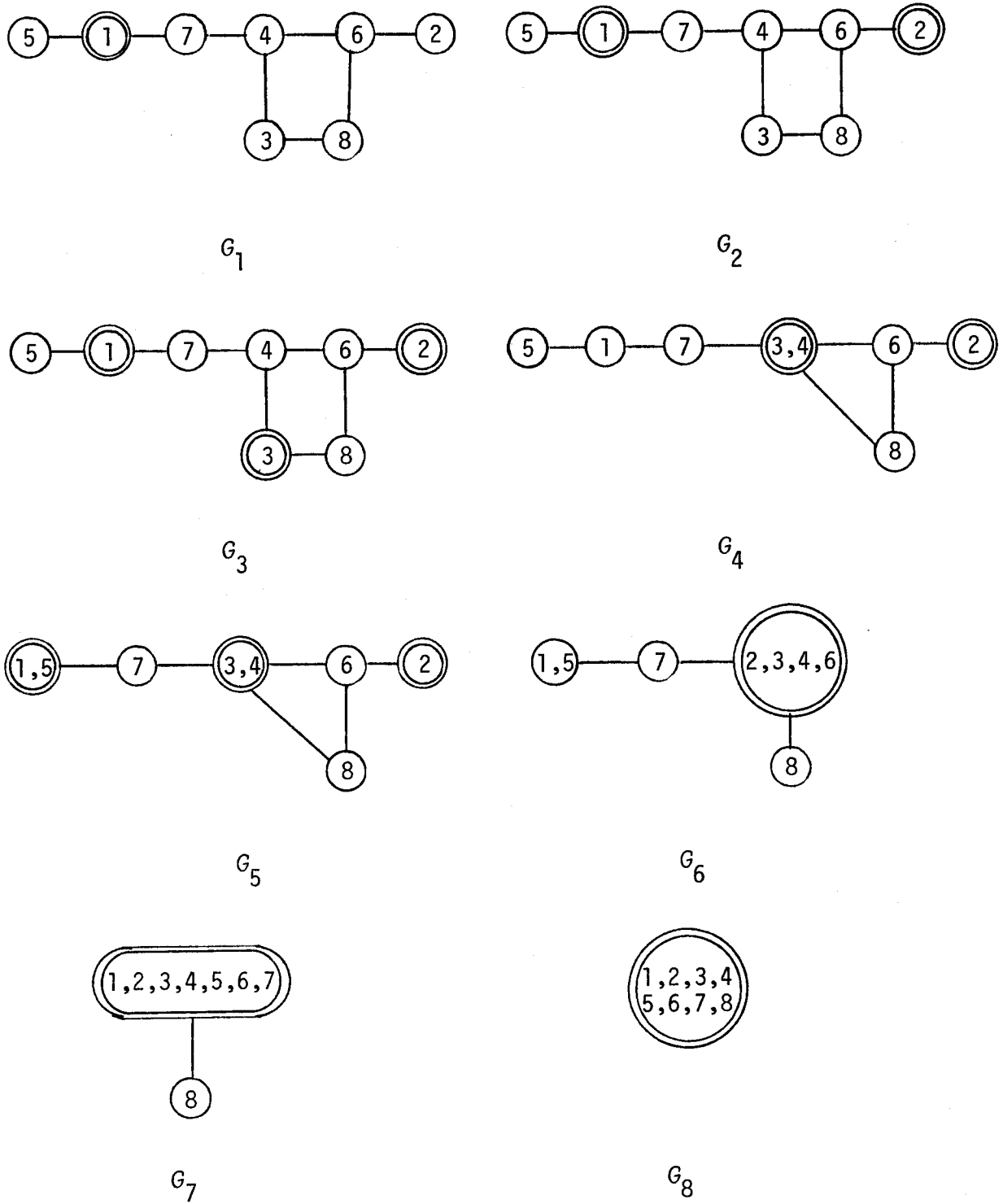


Figure 3.3 A sequence of quotient graphs

As we stated earlier, the quotient graph $G_i = G/\overline{C}(S_i)$ can be regarded as an implicit representation of the elimination graph G_i ; they are related by the following result.

Lemma 3.1 For $y \notin S_i$,

$$\text{Reach}_{G_i}(\{y\}, C(S_i)) = \{\{x\} \mid x \in \text{Reach}_G(y, S_i)\} .$$

Proof Consider $x \in \text{Reach}_G(y, S_i)$. If x and y are adjacent in G , so are $\{x\}$ and $\{y\}$ in G_i . Otherwise, there exists a path y_1, s_1, \dots, s_t, x in G where $\{s_1, \dots, s_t\} \subset S_i$. Let C be the component in $G(S_i)$ containing $\{s_1, \dots, s_t\}$. Then we have a path $\{y\}, C, \{x\}$ in G_i so that

$$\{x\} \in \text{Reach}_{G_i}(\{y\}, C(S_i)) .$$

Conversely, consider any $\{x\} \in \text{Reach}_{G_i}(\{y\}, C(S_i))$. There exists a path

$$\{y\}, C_1, \dots, C_t, \{x\}$$

in G_i where each $C_j \in C(S_i)$. If $t = 0$, then x and y are adjacent in the original graph G . If $t > 0$, by the definition of $C(S_i)$, t cannot be greater than one; that is, the path must be

$$\{y\}, C, \{x\} .$$

Since $G(C)$ is a connected subgraph, we can obtain a path from y to x through $C \subset S_i$ in the graph G . Hence

$$x \in \text{Reach}_G(y, S_i) .$$

□

We can obtain the structure of the elimination graph G_i from G_i as follows.

Step 1 Remove nodes in $C(S_i)$ and their incident edges from the quotient graph G_i .

Step 2 For each $C \in C(S_i)$, add edges to the quotient graph so that all adjacent nodes of C form a clique in the elimination graph.

To illustrate the idea, consider the transformation of G_4 to G_4 for the example in Figure 3.3. The elimination graph G_4 is given in Figure 3.4.

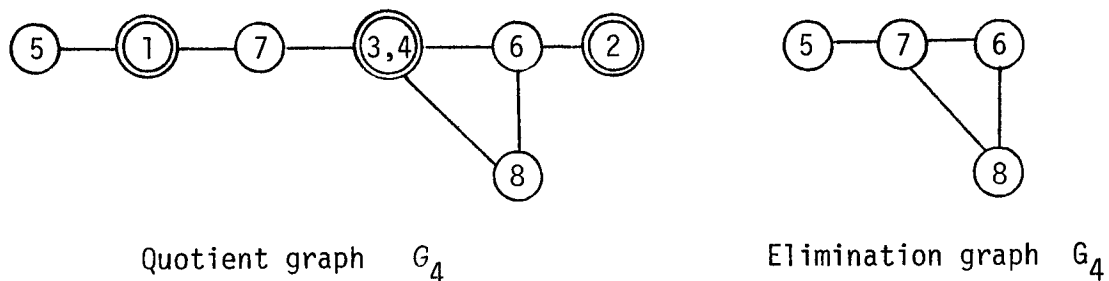
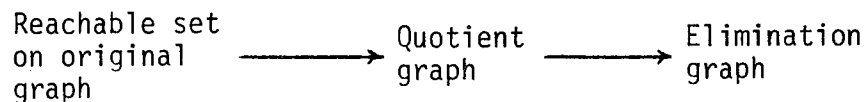


Figure 3.4 From quotient graph to elimination graph.

In terms of implicitness, the quotient graph model lies in between the reachable set model and the elimination graph model, as a vehicle for simulating the elimination process.



Since it is more explicit than the reachable set model in the representation, less effort is usually required to produce the adjacency sets for the elimination graph. On the other hand, the new model has the

advantage over the explicit elimination graph model in that it requires a fixed amount of storage in its computer implementation. This point is elaborated upon in the next section.

The fundamental nature of the elimination process suggests that in general the elimination graphs G_i will contain cliques. An advantage of using the clique model described in Section 2.4 is that the edges within each of the cliques are stored only implicitly via the clique membership. For graphs having large cliques, this can lead to substantial reductions in storage over an implementation which stores elimination graphs explicitly. Our quotient graph model is endowed with a similar advantage in that the cliques are represented by a single node, so that a p -node clique in the graph is represented by $O(p)$ edges, rather than $O(p^2)$ edges.

4. Computer Implementation of the Quotient Graph Model

4.1 Preliminary Results

In this subsection, some simple but important properties of the quotient graph model will be established. These will be used to show that the model can be implemented in-place. We discuss the implementation in Section 4.2. Let $G = (X, E)$ be a given graph.

Lemma 4.1 Let $S \subset X$ where $G(S)$ is a connected subgraph. Then

$$\sum_{x \in S} |\text{Adj}(x)| \geq |\text{Adj}(S)| + 2(|S| - 1) .$$

Proof Since $G(S)$ is connected, there are at least $|S| - 1$ edges in the subgraph. These edges are counted twice in $\sum_{x \in S} |\text{Adj}(x)|$ and hence the result.

□

We now show that the edge set sizes of the quotient graphs G_i cannot increase with increasing i . Let x_1, x_2, \dots, x_N be the node sequence, let

$$G_i = (X_i, E_i) , \quad 0 \leq i \leq N ,$$

be the corresponding elimination graph sequence, and let

$$G_i = (\mathcal{C}(S_i), \mathcal{E}_i) , \quad 1 \leq i \leq N ,$$

be the corresponding quotient graph sequence.

Theorem 4.2 For $1 \leq i \leq N$,

$$|\overline{\mathcal{C}}(S_{i+1})| \leq |\overline{\mathcal{C}}(S_i)|$$

and

$$|\hat{\mathcal{E}}_{i+1}| \leq |\hat{\mathcal{E}}_i| .$$

Proof Since $\mathcal{C}(S_i)$ is the set of components in the subgraph $G(S_i)$, we have

$$|\mathcal{C}(S_{i+1})| \leq |\mathcal{C}(S_i)| + 1 .$$

However, $|\overline{\mathcal{C}}(S_i)| = |\mathcal{C}(S_i)| + N - i$, so that

$$\begin{aligned} |\overline{\mathcal{C}}(S_{i+1})| &= |\mathcal{C}(S_{i+1})| + N - i - 1 \\ &\leq |\mathcal{C}(S_i)| + N - i \\ &= |\overline{\mathcal{C}}(S_i)| . \end{aligned}$$

For the inequality on the edge size, consider $\mathcal{C}(S_{i+1})$. If $\{x_{i+1}\} \in \mathcal{C}(S_{i+1})$, clearly $|\hat{\mathcal{E}}_{i+1}| = |\hat{\mathcal{E}}_i|$. Otherwise, the node x_{i+1} is merged with some components in $\mathcal{C}(S_i)$ to form a new component in $\mathcal{C}(S_{i+1})$. But then Lemma 4.1 applies, so that $|\hat{\mathcal{E}}_{i+1}| < |\hat{\mathcal{E}}_i|$. Hence in all cases,

$$|\hat{\mathcal{E}}_{i+1}| \leq |\hat{\mathcal{E}}_i| .$$

□

The next theorem shows that the degrees of some nodes in the quotient graphs also decrease monotonely with i . The proof follows easily from the definition of the quotient graphs G_i and is omitted.

Theorem 4.3 For $x \notin S_{i+1}$,

$$|\text{Adj}_{G_{i+1}}(\{x\})| \leq |\text{Adj}_{G_i}(\{x\})|.$$

□

The next theorem illustrates the advantage of the quotient graph model over the elimination graph model, and is one of the primary motivations for our introduction of the model.

Theorem 4.4

$$\max_{1 \leq i \leq N} |\mathcal{E}_i| \leq |E| \leq \max_{0 \leq i \leq N} |E_i|.$$

Proof The first inequality follows from Theorem 4.2, and the fact that $|E_0| = |E|$ implies the second one.

□

To illustrate the possible difference between the quantities $\max |\mathcal{E}_i|$ and $\max |E_i|$, we consider the example in Figure 4.1. The corresponding elimination graph and quotient graph sequences are given in Figure 4.2. If we generalize the example in an N-node graph, we have

$$\max |\mathcal{E}_i| = N - 1$$

and

$$\max |E_i| = (N - 1)(N - 2)/2.$$

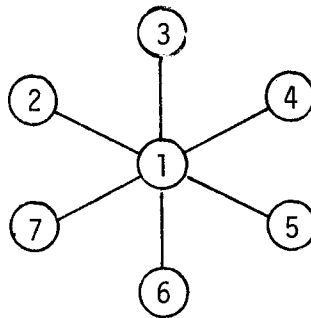


Figure 4.1 A seven-node star graph

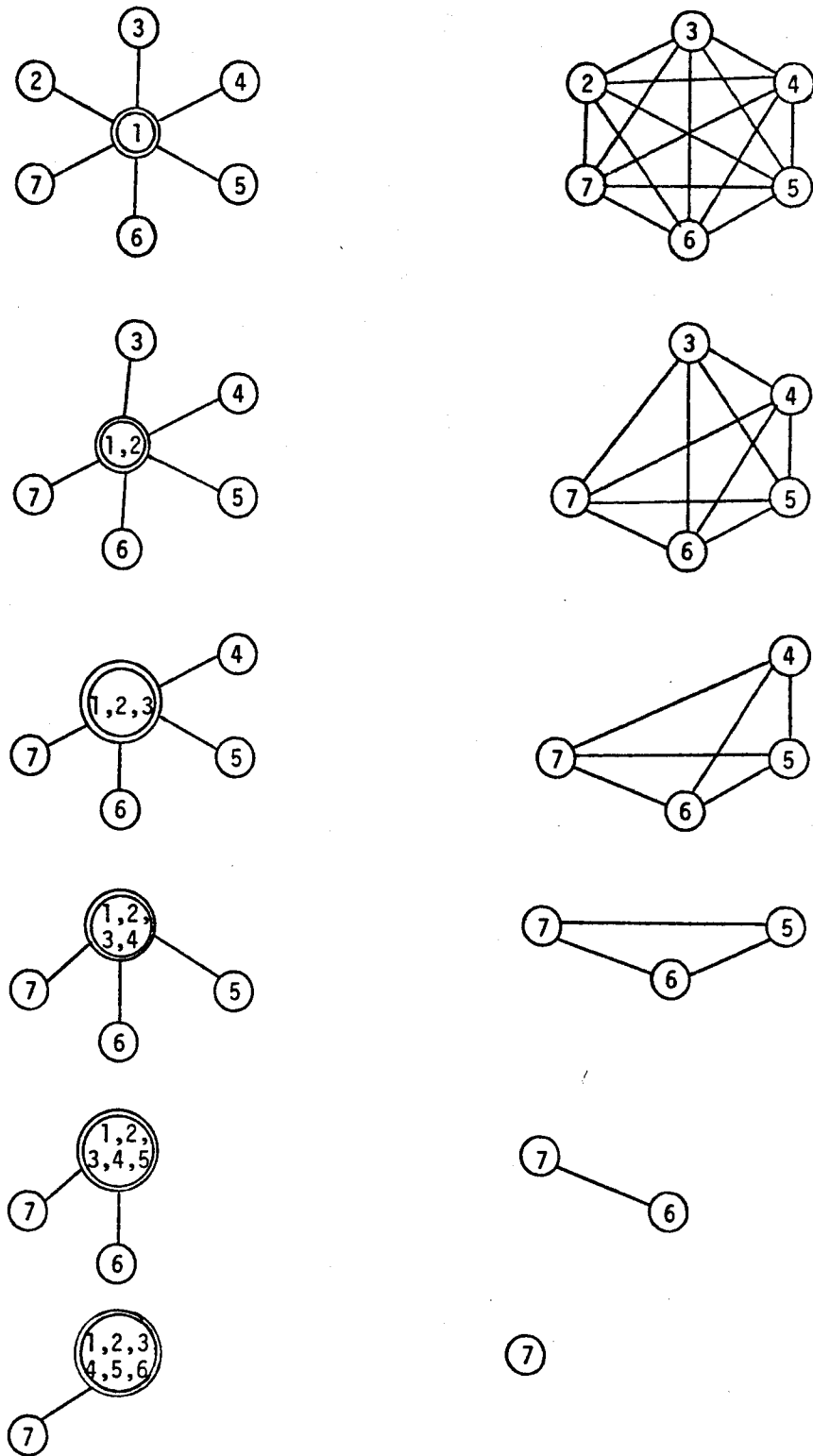


Figure 4.2 The quotient graph and elimination graph sequences for the example in Figure 4.1.

4.2 An In-Place Implementation

A practical way of storing the adjacency structure of a quotient graph is to use representatives for members of the partitioning P . Conceptually, for each partition member $C \in P$, we select a node $x \in C$ as the representative for C , and for convenience, we use x to denote the member it represents. It should be noted that the structure of G/P is independent of the choice of representatives.

In this subsection, we will demonstrate how we can use Lemma 4.1 to achieve an in-place implementation for the sequence of elimination graphs. We first describe the quotient graph transformation in a more general setting.

Let $G = (X, E)$. Given a subset $Y \subset X$ and a node $x \notin Y$ such that the subgraph $G(Y \cup \{x\})$ is connected. The problem is to generate the quotient graph by collapsing the subset $Y \cup \{x\}$ to form a new "super-node". The following is a description of an in-place implementation of the transformation.

Step 1 (Form new adjacent set) Determined the set $\text{Adj}(Y \cup \{x\})$.

Step 2 (In-place implementation) Use the node x as the representative of the new quotient member $Y \cup \{x\}$. Reset

$$\text{Adj}(x) \leftarrow \text{Adj}(Y \cup \{x\}) .$$

Step 3 (Neighbor update) For $z \in \text{Adj}(Y \cup \{x\})$, put

$$\text{Adj}(z) \leftarrow (\text{Adj}(z) - Y) \cup \{x\} .$$

In Step 2, although $|\text{Adj}(Y \cup \{x\})|$ may be greater than $|\text{Adj}(x)|$, we have by Lemma 4.1

$$\sum_{y \in Y \cup \{x\}} |\text{Adj}(y)| \geq |\text{Adj}(Y \cup \{x\})| + 2|Y| .$$

Therefore, there are always enough storage locations for $\text{Adj}(Y \cup \{x\})$ from those for $\text{Adj}(y)$, $y \in Y \cup \{x\}$. In addition, for $Y \neq \phi$, there is a surplus of $2|Y|$ locations, which can be utilized, for example, as links or pointers.

It should also be noted that in Step 3, by Theorem 4.3, the new neighbor set $(\text{Adj}(z) - Y) \cup \{x\}$ in neighbor update step can also be done in place.

In modelling elimination by the sequence of quotient graphs $\{G_i\}$, the graph G_{i+1} can be obtained from G_i by the execution of the above transformation. More specifically, we use the transformation to collapse the node x_{i+1} with those $C \in C(S_i)$ for which $x_{i+1} \in \text{Adj}(C)$.

To provide a concrete example to demonstrate an in-place implementation, we consider the graph of Figure 3.3 and we assume the adjacency structure is represented as shown in Figure 4.3.

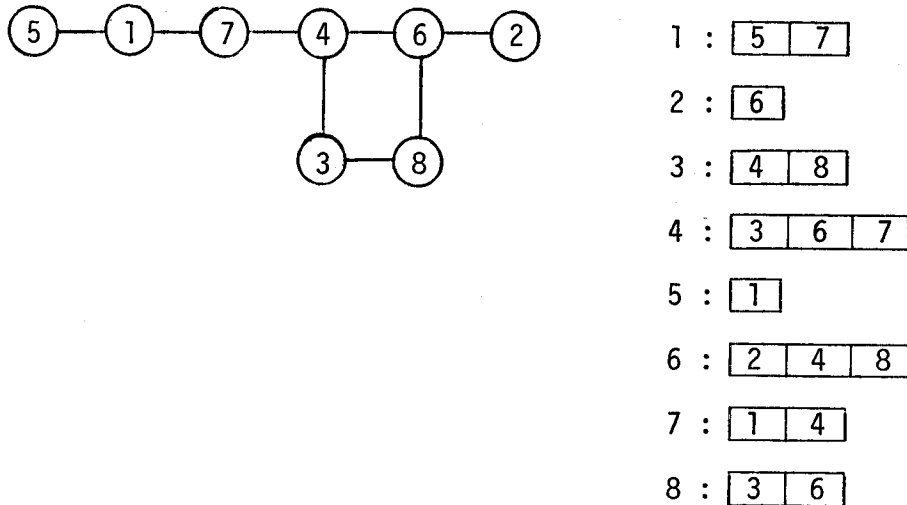
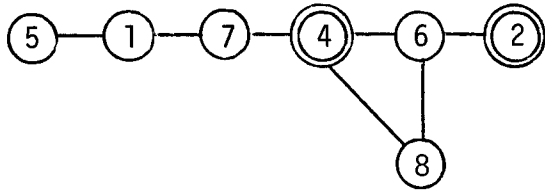


Figure 4.3 A graph and its representation

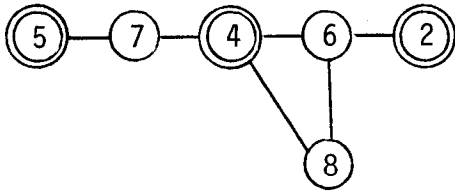
Figure 4.4 shows some important steps in producing quotient graphs for this example. The adjacency structure remains unchanged when the quotient graphs G_1 , G_2 and G_3 are formed. To transform G_3 to G_4 , the nodes 3 and 4 are to be collapsed, so that in G_4 , the new adjacent set of node 4 contains that of the subset $\{3, 4\}$ in the original graph, namely $\{6, 7, 8\}$. Here, the last location for $\text{Adj}_G(4)$ is used as a link. Note also that in the neighbor list of node 8, 3 has been changed to 4 in G_4 since node 4 becomes the representative of the component subset $\{3, 4\}$.

The representations for G_5 and G_6 in this storage mode are also included in Figure 4.4.



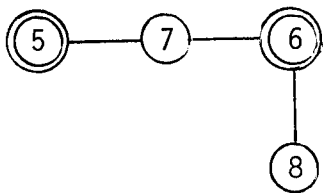
G_4

- 1 : [5 | 7]
- 2 : [6]
- 3 : [:8: | /]
- 4 : [:6: | :7: | :]
- 5 : [1]
- 6 : [2 | 4 | 8]
- 7 : [1 | 4]
- 8 : [:4: | 6]



G_5

- 1 : [:7: | /]
- 2 : [6]
- 3 : [8 | /]
- 4 : [6 | 7 |]
- 5 : [:]
- 6 : [2 | 4 | 8]
- 7 : [:5: | 4]
- 8 : [4 | 6]



G_6

- 1 : [7 | /]
- 2 : [i]
- 3 : [i | i]
- 4 : [i | i | i]
- 5 : []
- 6 : [:7: | :8: | /]
- 7 : [5 | :6:]
- 8 : [:6: | /]

:8: modified
 i ignored
 / end of list

Figure 4.4 An in-place quotient graph transformation

5. Concluding Remarks

As a theoretical tool for analyzing the symmetric factorization process, our model does not appear to be either better than, or inferior to, other models. Its primary advantage, in our opinion, is that it lends itself to very efficient computer implementation. We have given an implementation of the model which requires space only for the original graph of A , and is therefore independent of the fill suffered by A during its factorization.

We should caution the reader that the implementation we suggest in Section 4.2 is only an illustration. Although we have found the basic scheme for representing the G_i in Section 4.2 entirely adequate, the actual method for effecting the transformation depends upon how the model is being used. For example, in [5], where we use the model to develop an optimal algorithm for symbolic factorization, it turns out that Step 3 only needs to be performed for one $z \in \text{Adj}(Y \cup \{x\})$. In [6], we use the model in connection with a fast $O(|E|)$ -space implementation of the minimum degree algorithm.

6. References

- [1] S.C. Eisenstat, M.H. Schultz, and A.H. Sherman, "Applications of an element model for Gaussian elimination", in Sparse Matrix Computations, J.R. Bunch and D.J. Rose, editors, Academic Press, N. Y., 1969.
- [2] Alan George, "Nested dissection of a regular finite element mesh", *SIAM J. Numer. Anal.* 10 (1973), pp. 345-363.
- [3] Alan George and Joseph W.H. Liu, "Algorithms for matrix partitioning and the numerical solution of finite element systems", *SIAM J. Numer. Anal.*, to appear.
- [4] Alan George and Joseph W.H. Liu, "A minimal storage implementation of the minimum degree algorithm", *SIAM J. Numer. Anal.*, to appear.
- [5] Alan George and Joseph W.H. Liu, "An optimal algorithm for symbolic factorization of symmetric matrices", Research Report CS-78-11, Dept. of Computer Science, University of Waterloo, Waterloo, Ontario, March 1978.
- [6] Alan George and Joseph W.H. Liu, "A fast implementation of the minimum degree algorithm using quotient graphs", Research Report CS-78-12, Dept. of Computer Science, University of Waterloo, Waterloo, Ontario, May 1978.
- [7] Alan George and David R. McIntyre, "On the application of the minimum degree algorithm to finite element systems", *SIAM J. Numer. Anal.*, to appear.
- [8] S.V. Parter, "The use of linear graphs in Gauss elimination", *SIAM Rev.* 3 (1961), pp. 364-369.
- [9] D.J. Rose, "A graph theoretic study of the numerical solution of sparse positive definite systems", in Graph Theory and Computing, R.C. Read, editor, Academic Press, 1972.
- [10] D.J. Rose, R.E. Tarjan, and G. Lueker, "Algorithmic aspects of vertex elimination on graphs", *SIAM J. Comput.* 5 (1975), pp. 266-283.