

Copyright © 1976, by the author(s).  
All rights reserved.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission.

## **NODE-TEARING NODAL ANALYSIS**

by

Alberto Sangiovanni-Vincentelli, L-K. Chen  
and Leon O. Chua

Memorandum No. UCB/ERL M582

22 October 1976

*cover*

**NODE-TEARING NODAL ANALYSIS**

by

**A. Sangiovanni-Vincentelli, L.-K. Chen and L. O. Chua**

**Memorandum No. ERL-M582**

**22 October 1976**

**ELECTRONICS RESEARCH LABORATORY**

**University of California, Berkeley  
94720**

## NODE-TEARING NODAL ANALYSIS

Alberto Sangiovanni-Vincentelli,<sup>†</sup> Li-Kuan Chen and Leon O. Chua

Department of Electrical Engineering and Computer Sciences  
and the Electronics Research Laboratory  
University of California, Berkeley, California 94720

### ABSTRACT

An elementary yet novel approach is presented for analyzing large-scale networks with coupled elements by tearing along the nodes rather than branches of the associated network graph. This node-tearing approach leads to a system of nodal equations whose associated nodal admittance matrix assumes either a bordered-block-diagonal form or a bordered-block-triangular form. Compared to conventional diakoptic analysis, the node-tearing nodal analysis is shown to be superior in several important aspects: number of variables, sparsity, susceptibility to ill-conditioning, etc.

Two graph optimization problems associated with the node-tearing approach are formulated for the purpose of developing an optimum node-partitioning algorithm. These problems are then shown to belong to the NP-complete class of hard problems where no polynomial-bounded global optimum algorithms are likely to be found. However, an efficient heuristic algorithm for partitioning the nodes into clusters has been developed and presented. Several examples have verified the validity and usefulness of this algorithm.

Finally, the same tearing approach is extended to loop and cutset analysis thereby obtaining the fundamental loop and cutset equations in a corresponding bordered-block-diagonal or a bordered-block-triangular form.

---

Research sponsored by the Joint Services Electronics Program Contract F44620-76-C-0100 and the National Science Foundation Grant ENG75-22282.

<sup>†</sup>On leave from Instituto di Elettrotecnica ed Elettronica, Politecnico di Milano, Piazza L. da Vinci 32, 20133 Milano, Italy.

## I. Introduction

Recently, much effort has been devoted to tearing methods for the analysis of electrical networks [1-5]. Tearing, usually referred to as diakoptics, was first introduced by Kron [6]. It basically consists of breaking up the original analysis problem into smaller subproblems whose solutions are combined appropriately to give the solution of the original problem. Chua and Chen [4,7] have shown that all previously published tearing methods can be interpreted as special cases of their generalized hybrid analysis involving both voltage and current variables.

Among the numerous diakoptic methods, one is of particular interest; namely, the so-called diakoptic nodal analysis introduced by Wu [5]. Nodal analysis (or its modification [8]) is likely to be the most widely used analysis method because of its simplicity and efficiency [9]. Our main goal in this paper is to derive a new tearing approach based on nodal analysis -- henceforth called the node-tearing nodal analysis -- which uses only the familiar node-to-datum voltage variables, rather than a mixed set of voltage and current variables required in all existing methods of diakoptic analysis. In other words, our method is not a special case of the generalized hybrid analysis of [4] and is therefore not prone to numerical ill-conditioning often caused by the wide disparity in the order of magnitude of the elements of the associated hybrid matrix.

This paper is organized as follows: In Section 2, the node-tearing nodal analysis approach is introduced by simply partitioning the nodes and branches in a particular way. The further imposition of a rather mild branch-coupling condition leads naturally to a nodal analysis having either a bordered-block-diagonal or triangular structure. This node-tearing approach is then given a circuit-theoretic interpretation somewhat reminiscent of that given in [4].

In Section 3, a comparison with Wu's diakoptic nodal analysis is carried out. It is shown that, in almost all cases, node-tearing nodal analysis is superior to diakoptic nodal analysis.

In Section 4, optimization problems related to the choice of the partition of nodes and branches are introduced. They are also given graph-theoretic interpretations. Furthermore, their computational complexities are discussed and shown to belong to a class of hard problems, the so-called NP-complete problems [10,11], where no efficient global solution algorithm can be expected.

In Section 5, an efficient heuristic cluster algorithm is introduced. From test results, the heuristic solutions are found to be very close to the global optimal solutions. In Section 6, some concluding remarks are given.

In Appendix A, the most efficient mathematical methods for solving the tearing equations are discussed. Their computational complexities are evaluated and compared under various sparsity assumptions. In Appendix B, a brief description of NP-complete problems is presented. Also included are the proofs that the optimization problems introduced in Section 4 are indeed NP-complete. Finally, in Appendix C, some other tearing methods based on the same principle as node-tearing nodal analysis are described.

## II. Node-Tearing Nodal Analysis

Our goal in this section is to derive a new tearing approach [4] based on nodal analysis. To simplify the notation, we consider only linear resistive networks. However, the same approach can be easily generalized for nonlinear networks following the procedure given in [4].

Let us briefly recall that, if  $N$  is a connected network whose branches are grouped into the standard composite form (Fig. 1) and if  $G$  is its associated graph, then the standard nodal analysis yields [12]:

$$\underline{A} \underline{G} \underline{A}^T \underline{V}_n = \underline{J}_s \quad (1)$$

where

$$\underline{J}_s \triangleq \underline{A}(\underline{j} - \underline{G}\underline{e}) \quad (2)$$

$\underline{A}$  is the reduced incidence matrix of  $G$ ,  $G$  is the branch conductance matrix,  $\underline{V}_n$  is the node-to-datum voltage vector,  $\underline{j}$  is the branch current-source vector and  $\underline{e}$  is the branch voltage-source vector.  $\underline{J}_s$  can also be interpreted as an equivalent current-source vector representing all the currents injected into each node due to the independent sources.

As pointed out in [4], the tearing approach can be interpreted as partitioning the nodes and branches of  $N$  in such a way that the associated network equations involve a matrix with a special structure. This special structure must lead to a straight-forward decomposition of the system of equations into smaller subsystems, each of which can be solved either independently or in accordance with certain ordering. Furthermore, the decomposed solutions should coincide with the original solutions.

Two structures (or forms) of a square matrix satisfying the above requirement are the bordered-block-diagonal form, henceforth denoted by BBDF, and the bordered-block-triangular form,<sup>1</sup> henceforth denoted by BBTF [13] (also, see Fig. 2 and Appendix A). Hence our aim is to partition the nodes and branches of  $N$  in such a way that the nodal admittance matrix  $\underline{Y} \triangleq \underline{A} \underline{G} \underline{A}^T$  is in BBDF or in BBTF. The resulting nodal equation will then be referred to as the node-tearing nodal analysis, denoted by NTNA.

---

<sup>1</sup>This is also referred to as the bordered-block-lower-triangular form.

Basic Partition:

Let  $\mathcal{V}$  denote the nondatum nodes of  $G$  and let  $\mathcal{B}$  denote the set of branches of  $G$ . Partition the nondatum node set  $\mathcal{V}$  into two arbitrary subsets  $\mathcal{V}_1$  and  $\mathcal{V}_2$ . Partition the branch set  $\mathcal{B}$  into two subsets  $\mathcal{B}_1$  and  $\mathcal{B}_2$  such that:

- (i)  $\mathcal{B}_1$  contains all those branches in  $\mathcal{B}$  that are incident with nodes in  $\mathcal{V}_1$ ;
- (ii)  $\mathcal{B}_2$  contains all the remaining branches of  $\mathcal{B}$ .

Graphically, this basic partition of nodes and branches is illustrated in Fig. 3. Topologically, it yields the following special structure for the reduced incidence matrix  $A$ :

$$A = \begin{bmatrix} & \mathcal{B}_1 & \mathcal{B}_2 \\ \mathcal{N}_1 & \begin{bmatrix} A_{11} & 0_{12} \\ A_{21} & A_{22} \end{bmatrix} \\ \mathcal{N}_2 & \end{bmatrix} \quad (3)$$

Remark 1.  $A_{12} = 0_{12}$  is of fundamental importance for subsequent derivations.

If we rearrange the branch conductance matrix  $G$  with respect to the basic partition as

$$G = \begin{bmatrix} & \mathcal{B}_1 & \mathcal{B}_2 \\ \mathcal{B}_1 & \begin{bmatrix} G_{11} & G_{12} \\ G_{21} & G_{22} \end{bmatrix} \\ \mathcal{B}_2 & \end{bmatrix} \quad (4)$$

then the nodal equation becomes

$$\begin{matrix} \mathcal{N}_1 & \mathcal{N}_2 \end{matrix}$$

$$\begin{matrix} \mathcal{N}_1 \\ \mathcal{N}_2 \end{matrix} \begin{bmatrix} Y_{11} & Y_{12} \\ Y_{21} & Y_{22} \end{bmatrix} \begin{bmatrix} V_{n_1} \\ V_{n_2} \end{bmatrix} = \begin{bmatrix} J_{s_1} \\ J_{s_2} \end{bmatrix} \quad (5)$$

where

$$Y_{11} = A_{11} G_{11} A_{11}^t \quad (6)$$

$$Y_{12} = A_{11} G_{11} A_{21}^t + A_{11} G_{12} A_{22}^t \quad (7)$$

$$Y_{21} = A_{21} G_{11} A_{11}^t + A_{22} G_{21} A_{11}^t \quad (8)$$

$$Y_{22} = A_{21} G_{11} A_{21}^t + A_{22} G_{21} A_{21}^t + A_{21} G_{12} A_{22}^t + A_{22} G_{22} A_{22}^t \quad (9)$$

Let us now look at the conditions that assure  $\mathbf{Y}$  to be in a form suitable for tearing. Basically, they are of two kinds:

- (i) the connection between branches, henceforth called the topological condition;
- (ii) the coupling between branches, henceforth call the branch coupling condition.

#### Topological Condition:

The section graph<sup>2</sup>  $\mathcal{G}(\mathcal{V}_1)$  has "m" ( $m > 1$ ) disconnected components<sup>3</sup>

<sup>2</sup>Given a graph  $\mathcal{G} = (X, U)$  where  $X$  denotes the set of nodes and  $U$  denotes the set of branches, let  $S$  be a subset of nodes, then the section graph [14]  $\mathcal{G}(S)$  is the graph  $(S, U_S)$  where  $U_S = \{b \in U | b \text{ incidents only with } S\}$ . The section graph can also be interpreted as the graph obtained from  $\mathcal{G}$  by removing all  $\{X-S\}$  nodes and all branches with at least one terminal node belonging to  $\{X-S\}$ . Hence, both terminal nodes of each branch in  $\mathcal{G}(S)$  must belong to  $S$ .

<sup>3</sup>In [4] we required the components to be separable whereas here we require the components to be disconnected. In other words, we need a slightly stronger requirement here.

$$\mathcal{G}_1^1 = (\mathcal{V}_1^1, \mathcal{B}_1^{11}), \mathcal{G}_1^2 = (\mathcal{V}_1^2, \mathcal{B}_1^{21}), \dots, \mathcal{G}_1^m = (\mathcal{V}_1^m, \mathcal{B}_1^{m1}).^4$$

It follows from the topological condition that  $\mathcal{A}$  can be further partitioned as:

$$\mathcal{A} = \begin{bmatrix} \mathcal{B}_1^1 & \mathcal{B}_1^2 & \dots & \mathcal{B}_1^m & \mathcal{B}_2 \\ \mathcal{N}_1^1 & \begin{matrix} A_{11}^1 \\ \vdots \end{matrix} & & \begin{matrix} 0 \\ \vdots \end{matrix} & \begin{matrix} \vdots \\ 0 \\ \vdots \\ -12 \end{matrix} \\ \mathcal{N}_1^2 & & \begin{matrix} A_{11}^2 \\ \ddots \end{matrix} & & \vdots \\ \vdots & & & \ddots & \vdots \\ \mathcal{N}_1^m & & & & \begin{matrix} A_{11}^m \\ \vdots \end{matrix} \\ \mathcal{N}_2 & \begin{matrix} A_{21}^1 & A_{21}^2 & \dots & A_{21}^m & \vdots & \dots \end{matrix} & & & & \end{bmatrix} \quad (10)$$

where  $\mathcal{B}_1^k \triangleq \mathcal{B}_1^{k1} \cup \mathcal{B}_1^{k0} \cup \mathcal{B}_1^{k2}$ ,  $\mathcal{B}_1^{k0}$  is the set of branches connecting an  $\mathcal{N}_1^k$  node and the datum node,  $\mathcal{B}_1^{k2}$  is the set of branches connecting an  $\mathcal{N}_1^k$  node and an  $\mathcal{V}_2$  node. The notations are fully illustrated in Fig. 4.

Remark 2. The upper left submatrix  $A_{11}$  of (3) has the block diagonal structure as shown in (10).

As an example, consider the graph of Fig. 5(a). First, let us partition the nondatum node set into  $\mathcal{V}_1 = \{n_1, n_2, n_3, n_4\}$  and  $\mathcal{V}_2 = \{n_5, n_6\}$ . Accordingly,  $\mathcal{B}$  is partitioned into  $\mathcal{B}_1 = \{b_1, b_2, b_3, b_4, b_5, b_6, b_7\}$  and  $\mathcal{B}_2 = \{b_8, b_9, b_{10}\}$ . Note that the section graph  $\mathcal{G}(\mathcal{V}_1)$  has three disconnected components  $\mathcal{G}_1^1, \mathcal{G}_1^2$  and  $\mathcal{G}_1^3$  (Fig. 5(b)).<sup>5</sup> The reduced incidence matrix  $\mathcal{A}$  of this example can then be partitioned as follows:

<sup>4</sup>Observe that  $\mathcal{V}_1^1 \cup \mathcal{V}_1^2 \cup \dots \cup \mathcal{V}_1^m = \mathcal{V}_1$  and  $\{\bigcup_{k=1}^m \mathcal{B}_1^{k1}\} \cup \{\text{branches between an } \mathcal{V}_1 \text{ node and the datum node}\} \cup \{\text{branches between an } \mathcal{V}_1 \text{ node and an } \mathcal{V}_2 \text{ node}\} = \mathcal{B}_1$ .

<sup>5</sup>Observe that  $\mathcal{G}_1^1$  and  $\mathcal{G}_1^2$  contain no branches because each branch connected to either node  $n_1$  or  $n_2$  is also incident with some node not belonging to  $\mathcal{V}_1$ .

$$\begin{array}{c}
 \overbrace{\mathcal{B}_1^1}^{b_1 b_2} \quad \overbrace{\mathcal{B}_1^2}^{b_3 b_4} \quad \overbrace{\mathcal{B}_1^3}^{b_5 b_6 b_7} \quad \overbrace{\mathcal{B}_2}^{b_8 b_9 b_{10}}
 \\[10pt]
 \mathbf{A} = \left[ \begin{array}{cccc|cc|cc}
 \mathcal{N}_1^1 \{n_1\} & 1 & 1 & 0 & & & & \\
 \mathcal{N}_1^2 \{n_2\} & 0 & 1 & 1 & 0 & & & \\
 \mathcal{N}_1^3 \{n_3\} & & & 1 & 1 & & & 12 \\
 \mathcal{N}_1^4 \{n_4\} & & & 1 & 1 & & & \\
 \hline
 \mathcal{N}_2 \{n_5\} & & 1 & 1 & 1 & 1 & 1 & \\
 \mathcal{N}_2 \{n_6\} & 1 & 1 & 1 & 1 & 1 & 1 &
 \end{array} \right] \quad (11)
 \end{array}$$

Before considering the branch coupling condition, let us relate the structure of  $\underline{G}$  with the structure of  $\underline{Y}$  of a network in which the topological condition is satisfied.

Lemma 1. If the topological condition holds, then the submatrix  $\underline{Y}_{11}$  of (5) preserves the nonzero block structure of  $\underline{G}_{11}$  of (4) in the sense that both  $\underline{Y}_{11}$  and  $\underline{G}_{11}$  have identical nonzero blocks as illustrated in Fig. 6.

Proof: Recall from (6) that  $A_{12} = 0$  guarantees the relation  $\underline{Y}_{11} = A_{11} G_{11} A_{11}^t$ . Since the topological condition implies that  $A_{11}$  is block diagonal, then the product  $A_{11} G_{11} A_{11}^t$  can be regarded as a block diagonal transformation of nonzero elements. Thus  $\underline{Y}_{11}$  preserves the nonzero block structure of  $G_{11}$  (Fig. 6).  $\square$

Now, it is straight-forward to give conditions on  $\underline{G}$  such that  $\underline{Y}$  is in a form suitable for tearing.

#### Branch Coupling Condition 1:

Branches in  $\mathcal{B}_1^i$  are not coupled to branches in  $\mathcal{B}_1^j$  for all  $i \neq j$  and  $i, j = 1, 2, \dots, m$ .

According to branch coupling condition 1,  $\underline{G}$  can be recast as

$$G = \begin{bmatrix} \mathcal{B}_1^1 & \mathcal{B}_1^2 & \dots & \mathcal{B}_1^m & \mathcal{B}_2 \\ \mathcal{B}_1^1 & G_{11}^{11} & & 0 & G_{12}^1 \\ \mathcal{B}_1^2 & & G_{11}^{22} & & G_{12}^2 \\ \vdots & 0 & \ddots & & \vdots \\ \mathcal{B}_1^m & & & G_{11}^{mm} & G_{12}^m \\ \mathcal{B}_2 & G_{21}^1 & G_{21}^2 & \dots & G_{21}^m | G_{22} \end{bmatrix} \quad (12)$$

Theorem 1. If the topological condition and the branch coupling condition hold,  $\underline{Y}$  can be put in the bordered-block-diagonal form (BBDF). In particular, we have

$$\underline{Y} = \begin{bmatrix} \mathcal{N}_1^1 & \mathcal{N}_1^2 & \dots & \mathcal{N}_1^m & \mathcal{N}_2 \\ \mathcal{N}_1^1 & Y_{11}^{11} & & 0 & Y_{12}^1 \\ \mathcal{N}_1^2 & & Y_{11}^{22} & & Y_{12}^2 \\ \vdots & 0 & \ddots & & \vdots \\ \mathcal{N}_1^m & & & Y_{11}^{mm} & Y_{12}^m \\ \mathcal{N}_2 & Y_{21}^1 & Y_{21}^2 & \dots & Y_{21}^m | Y_{22} \end{bmatrix} \quad (13)$$

where, for  $k = 1, 2, \dots, m$ ,

$$Y_{11}^{kk} = A_{11}^k G_{11}^{kk} (A_{11}^k)^t \quad (14)$$

$$Y_{12}^k = A_{11}^k G_{11}^{kk} (A_{21}^k)^t + A_{11}^k G_{12}^k A_{22}^t \quad (15)$$

$$Y_{21}^k = A_{21}^k G_{11}^{kk} (A_{11}^k)^t + A_{22}^k G_{21}^k (A_{11}^k)^t \quad (16)$$

$$Y_{22} = \sum_{k=1}^m [A_{21}^k G_{11}^{kk} (A_{21}^k)^t + A_{22}^k G_{21}^k (A_{21}^k)^t + A_{21}^k G_{12}^k A_{22}^t] + A_{22}^k G_{22}^k A_{22}^t \quad (17)$$

Proof: Obvious by Lemma 1. □

It follows from Theorem 1 that the nodal equation (5) now has the following block structure, henceforth called the node-tearing nodal analysis in bordered-block-diagonal form (or simply as NTNA in BBDF):

$$\begin{bmatrix} \underline{Y}_{11}^{11} & & & & \\ & \underline{Y}_{11}^{22} & & & \\ & & \ddots & & \\ & & & \underline{Y}_{11}^{mm} & \\ \hline & \underline{Y}_{21}^1 & \underline{Y}_{21}^2 & \cdots & \underline{Y}_{21}^m \end{bmatrix} \begin{bmatrix} \underline{Y}_{12}^1 & \underline{Y}_{12}^2 & \cdots & \underline{Y}_{12}^m \\ \vdots & \vdots & & \vdots \\ \underline{Y}_{11}^m & \underline{Y}_{12}^m & \cdots & \underline{Y}_{22}^m \end{bmatrix} \begin{bmatrix} \underline{v}_{n_1}^1 & \underline{v}_{n_1}^2 & \cdots & \underline{v}_{n_1}^m \\ \vdots & \vdots & & \vdots \\ \underline{v}_{n_1}^m & \underline{v}_{n_2}^1 & \cdots & \underline{v}_{n_2}^m \end{bmatrix} \begin{bmatrix} \underline{j}_{s_1}^1 & \underline{j}_{s_1}^2 & \cdots & \underline{j}_{s_1}^m \\ \vdots & \vdots & & \vdots \\ \underline{j}_{s_1}^m & \underline{j}_{s_2}^1 & \cdots & \underline{j}_{s_2}^m \end{bmatrix} \quad (18)$$

where  $\underline{v}_{n_1}^k$  and  $\underline{v}_{n_2}^k$  denote respectively the node-to-datum voltage vectors of nodes in  $\mathcal{N}_1^k$  and  $\mathcal{N}_2^k$ ,  $\underline{j}_{s_1}^k$  and  $\underline{j}_{s_2}^k$  denote respectively the equivalent current vectors representing the net current injected into nodes  $\mathcal{N}_1^k$  and  $\mathcal{N}_2^k$  due to all independent sources. Moreover,

$$\underline{j}_{s_1}^k = A_{11}^k (j_1^k - G_{11}^{kk} e_1^k - G_{12}^{kk} e_2^k) \quad (19)$$

$$\underline{j}_{s_2}^k = \sum_{k=1}^m A_{21}^k (j_1^k - G_{11}^{kk} e_1^k - G_{12}^{kk} e_2^k) + A_{22}^k (j_2^k - \sum_{k=1}^m G_{21}^k e_1^k - G_{22}^k e_2^k) \quad (20)$$

where  $j_1^k$  and  $j_2^k$  (resp.;  $e_1^k$  and  $e_2^k$ ) are the branch current-(resp.; voltage-) source vectors of branches in  $\mathcal{B}_1^k$  and  $\mathcal{B}_2^k$ , respectively.

Now, let us focus on the other form that is also suitable for tearing; namely, the bordered block triangular form (BBTF). The following condition guarantees that  $\underline{Y}$  is in BBTF if the topological condition holds.

#### Branch Coupling Condition 2:

Branches in  $\mathcal{B}_1^i$  are not coupled to branches in  $\mathcal{B}_1^j$  for all  $i < j$  and  $i, j = 1, 2, \dots, m$ .

It follows from branch coupling condition 2 that  $G$  can be partitioned as follows:

$$G = \begin{bmatrix} \mathcal{B}_1^1 & \mathcal{B}_1^2 & \dots & \mathcal{B}_1^m & \mathcal{B}_2 \\ \mathcal{B}_1^1 & G_{11}^{11} & & & G_{12}^1 \\ \mathcal{B}_1^2 & G_{11}^{21} & G_{11}^{22} & & G_{12}^2 \\ \vdots & \vdots & \ddots & & \vdots \\ \mathcal{B}_1^m & G_{11}^{ml} & G_{11}^{m2} & \dots & G_{11}^{mm} & G_{12}^m \\ \mathcal{B}_2 & G_{21}^1 & G_{21}^2 & \dots & G_{21}^m & G_{22} \end{bmatrix} \quad (21)$$

Observe that this condition is weaker than the branch coupling condition 1 in that condition 1 implies condition 2, but not vice-versa.

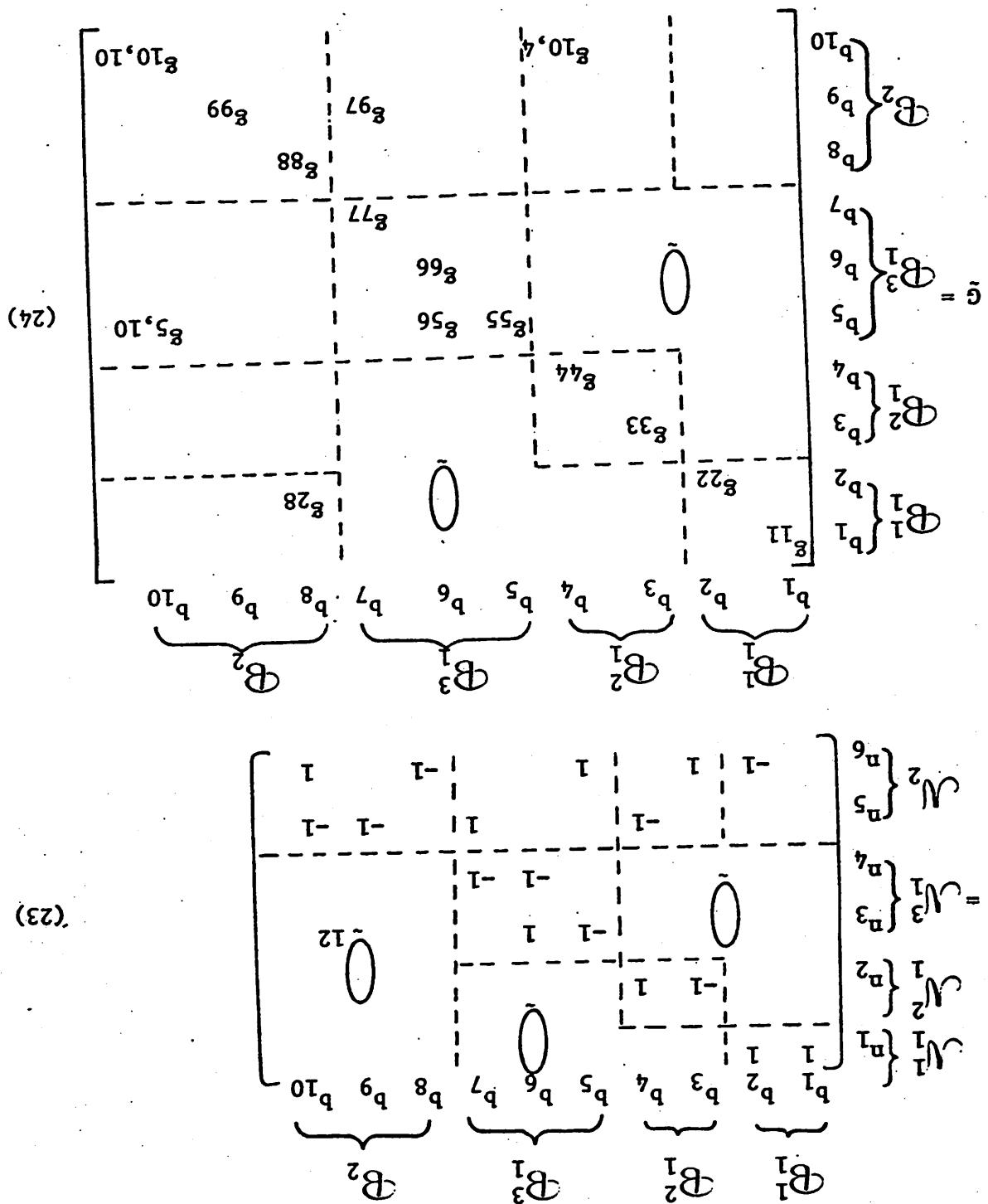
Theorem 2. If the topological condition and the branch coupling condition 2 hold, then  $\underline{Y}$  can be put in bordered-block-triangular form (BBTF). In particular, we have

$$\underline{Y} = \begin{bmatrix} \mathcal{N}_1^1 & \mathcal{N}_1^2 & \dots & \mathcal{N}_1^m & \mathcal{N}_2 \\ \mathcal{N}_1^1 & Y_{11}^{11} & & & Y_{12}^1 \\ \mathcal{N}_1^2 & Y_{11}^{21} & Y_{11}^{22} & & Y_{12}^2 \\ \vdots & \vdots & \ddots & & \vdots \\ \mathcal{N}_1^m & Y_{11}^{ml} & Y_{11}^{m2} & \dots & Y_{11}^{mm} & Y_{12}^m \\ \mathcal{N}_2 & Y_{21}^1 & Y_{21}^2 & \dots & Y_{21}^m & Y_{22} \end{bmatrix} \quad (22)$$

Proof: Obvious from Lemma 1. □

It follows from Theorem 2 that the nodal equation (5) now has the BBTF structure, henceforth called the node-tearing nodal analysis in bordered-block-triangular form (or simply as NTNA in BBTF).

Let us now pause to look at the example shown in Fig. 7. The reason why we choose a network with so many voltage-controlled current sources (i.e., couplings) is to illustrate that our branch coupling conditions are actually very mild.



The nodal equation are shown below, respectively:

Then the reduced incidence matrix  $\hat{A}$ , the branch conductance matrix  $G$  and

shown in Fig. 8. Let us partition the nodes and branches as before.

The directed graph  $G$  associated with the network  $N$  of Fig. 7 is

$$\left[ \begin{array}{|c|c|} \hline y_{11} & 0 \\ \hline \hline 0 & \left[ \begin{array}{|c|c|} \hline 0 & y_{16} \\ \hline y_{25} & y_{26} \\ \hline \end{array} \right] \\ \hline \hline y_{22} & \left[ \begin{array}{|c|c|c|c|} \hline y_{33} & y_{34} & y_{35} & y_{36} \\ \hline y_{43} & y_{44} & y_{45} & 0 \\ \hline 0 & y_{52} & y_{54} & y_{55} & y_{56} \\ \hline y_{61} & y_{62} & y_{63} & y_{64} & y_{65} & y_{66} \\ \hline \end{array} \right] \\ \hline \end{array} \right] = \left[ \begin{array}{|c|c|} \hline v_{n_1} \\ \hline v_{n_2} \\ \hline v_{n_3} \\ \hline v_{n_4} \\ \hline v_{n_5} \\ \hline v_{n_6} \\ \hline \end{array} \right] = \left[ \begin{array}{|c|c|} \hline J_1 \\ \hline 0 \\ \hline J_2 \\ \hline -J_2 \\ \hline 0 \\ \hline 0 \\ \hline \end{array} \right] \quad (25)$$

where

$$\left. \begin{aligned} y_{11} &\triangleq g_{11} + g_{22}; \quad y_{16} \triangleq -g_{22} - g_{28}; \\ y_{22} &\triangleq g_{33} + g_{44}; \quad y_{25} \triangleq -g_{44}; \quad y_{26} \triangleq -g_{33}; \\ y_{33} &\triangleq g_{55} + g_{66} - g_{56}; \quad y_{34} \triangleq g_{56} - g_{66}; \quad y_{35} \triangleq g_{5,10}; \\ y_{36} &\triangleq -g_{55} - g_{5,10}; \quad y_{43} \triangleq -g_{66}; \quad y_{44} \triangleq g_{66} + g_{77}; \\ y_{45} &\triangleq -g_{77}; \quad y_{52} \triangleq -g_{44} - g_{10,4}; \quad y_{54} \triangleq g_{97} - g_{77}; \\ y_{55} &\triangleq g_{44} + g_{77} + g_{99} + g_{10,10} + g_{10,4} - g_{97}; \quad y_{56} \triangleq -g_{10,10}; \\ y_{61} &\triangleq -g_{22}; \quad y_{62} \triangleq g_{10,4} - g_{33}; \quad y_{63} \triangleq g_{56} - g_{55}; \\ y_{64} &\triangleq -g_{56}; \quad y_{65} \triangleq -g_{10,10} - g_{10,4} - g_{5,10}; \\ y_{66} &\triangleq g_{22} + g_{33} + g_{55} + g_{88} + g_{10,10} + g_{28} + g_{5,10}. \end{aligned} \right\} \quad (26)$$

Let us observe the following:

- (i)  $A_{12} = 0_{12}$ .
- (ii)  $G$  as given in (24) has the same block structure as (12) and therefore satisfies the branch coupling condition 1.
- (iii)  $\underline{Y}$  of (25) is in BBDF.

Let us modify the  $G$  matrix in (24) by adding  $g_{31}$ ,  $g_{52}$  and  $g_{74}$ . Physically, we are adding voltage-controlled current sources across branches  $b_3$ ,  $b_5$  and  $b_7$  which depend on voltages of branches  $b_1$ ,  $b_2$  and  $b_4$ , respectively. Then, those elements of the resulting  $\underline{Y}$  that differ from

(25) are listed as follows:

$$\left. \begin{aligned}
 y_{21} &\triangleq -g_{31}; y_{31} \triangleq -g_{52}; y_{36} \triangleq g_{52} - g_{55} - g_{5,10}; y_{42} \triangleq -g_{74}; \\
 y_{45} &\triangleq g_{74} - g_{77}; y_{52} \triangleq -g_{44} + g_{74} - g_{10,4}; y_{55} \triangleq g_{44} - g_{74} + g_{77} - g_{97} \\
 &+ g_{99} + g_{10,10} + g_{10,4}; y_{61} \triangleq g_{52} - g_{22}; y_{66} \triangleq g_{22} + g_{33} + g_{55} + g_{88} \\
 &+ g_{10,10} + g_{28} + g_{5,10} - g_{52}
 \end{aligned} \right\} \quad (27)$$

Observe that both  $\underline{G}$  and  $\underline{Y}$  are in BBTF.

Remark 3. If neither branch coupling condition 1 nor 2 is satisfied relative to the preceding partitioning of  $\underline{G}(\mathcal{N}_1)$  into  $m$  separable components, it is often possible to relax these conditions and still obtain the nodal equation in BBDF or BBTF but with fewer blocks (of course, the dimension of the resulting blocks will be larger). To see how branch coupling condition 1 can be relaxed, let us partition the set of separable components of  $\underline{G}(\mathcal{N}_1)$ , denoted by  $S \triangleq \{G_1^1, G_1^2, \dots, G_1^m\}$ , into "r" disjoint subsets ( $r < m$ )  $S_1, S_2, \dots, S_r$ , each containing one or more separable components. Partition  $\mathcal{N}_1$  and  $\mathcal{B}_1$  each into "r" subsets  $\bar{\mathcal{N}}_1^k$  and  $\bar{\mathcal{B}}_1^k$ ,  $k = 1, 2, \dots, r$  where  $\bar{\mathcal{N}}_1^k$  contains all nodes in  $S_k$  and  $\bar{\mathcal{B}}_1^k$  contains all branches incident with  $\bar{\mathcal{N}}_1^k$  nodes (Fig. 9). The submatrix  $A_{11}$  can then be partitioned with respect to this new partition of nodes and branches. Since Lemma 1 still holds with respect to this new node partition, we can relax branch coupling condition 1 as follows: branches in  $\bar{\mathcal{B}}_1^i$  are not coupled to branches in  $\bar{\mathcal{B}}_1^j$  for all  $i \neq j$  and  $i, j = 1, 2, \dots, r$ . Obviously, Theorem 1 still holds with respect to this new node partition. The same reasoning (mutatis mutandis) can obviously be applied to relax branch coupling condition 2.

As an illustration, let us consider the same example in Fig. 7. Let

us add  $g_{31}$  and  $g_{24}$  to (24) thereby violating the branch coupling condition 1 relative to the original node partition. However, if we choose our new partition as  $\bar{V}_1 = V_1^1 \cup V_1^2$ ,  $\bar{V}_1^2 = V_1^3$ ,  $\bar{B}_1^1 = B_1^1 \cup B_1^2$  and  $\bar{B}_1^2 = B_1^3$ , then those elements of the resulting  $\bar{Y}$  that differ from (25) are listed as follows:

$$\left. \begin{aligned} y_{12} &\triangleq g_{24}; y_{15} \triangleq -g_{24}; y_{21} \triangleq -g_{31}; y_{61} \triangleq g_{31} - g_{22}; \\ y_{62} &\triangleq g_{10,4} - g_{33} - g_{24}; y_{65} \triangleq g_{24} - g_{10,10} - g_{10,4} - g_{5,10} \end{aligned} \right\} \quad (28)$$

Observe that both  $\underline{G}$  and  $\bar{Y}$  are now in BBDF and notice that there are only 2 diagonal blocks in this new partition as compared to the 3 diagonal blocks in (25).

So far, we have shown that these special cases of the well-known nodal analysis are obtained merely by a simple partitioning of the nodes and branches, and by imposing some rather mild branch coupling conditions.

We are now ready to give a circuit-theoretic interpretation of the node-tearing nodal analysis given in (18) in the same spirit as that given for diakoptic analysis in [4]. First, let us recast the  $k$ -th component matrix equation of (18) as

$$\underline{Y}_{11}^{kk} \underline{v}_{n_1}^k + \underline{Y}_{12}^k \underline{v}_{n_2} = \underline{j}_{s_1}^k \quad (29)$$

Substituting (14), (15) and (19) into (29), we obtain

$$\begin{aligned} & \underline{A}_{11}^k \underline{G}_{11}^{kk} (\underline{A}_{11}^k)^t \underline{v}_{n_1}^k + \underline{A}_{11}^k \underline{G}_{11}^{kk} (\underline{A}_{21}^k)^t \underline{v}_{n_2} + \underline{A}_{11}^k \underline{G}_{12}^k \underline{A}_{22}^t \underline{v}_{n_2} \\ &= \underline{A}_{11}^k \underline{j}_1^k - \underline{A}_{11}^k \underline{G}_{11}^{kk} \underline{e}_1^k - \underline{A}_{11}^k \underline{G}_{12}^k \underline{e}_2 \end{aligned} \quad (30)$$

Let us next augment a voltage source between each  $V_2$  node and the datum node and let its terminal voltage waveform be assigned equal to the corresponding node-to-datum voltage. Finally, let us remove all  $B_2$  branches (Fig. 10(a)). Because of the Substitution Theorem [12], this

procedure will not alter the solution of the network. Furthermore, these augmented voltage sources will summarize the "outside" influence due to branches in  $\mathcal{B}_2$  upon each subnetwork  $N_k$ . In other words, these augmented voltage sources effectively "decouple" the original network into "m" subnetworks (Fig. 10(b)). Each "decoupled" subnetwork  $N_k$  can now be analyzed separately as follows:<sup>6</sup>

$$\begin{aligned} \underline{0} &= A_{11-1}^k I^k = A_{11-1}^k i_1^k - A_{11-1}^k j_1^k = A_{11}^k (G_{11-1}^{kk} v_{11-1}^k + G_{12}^k v_{12}^k) - A_{11-1}^k j_1^k \\ &= A_{11-11}^k G_{11}^{kk} (v_{11-1}^k + e_{11-1}^k) + A_{11-12}^k G_{12}^k (v_{12}^k + e_{12}^k) - A_{11-1}^k j_1^k \\ &= A_{11-11}^k \left[ (A_{11}^k)^t v_{n_1}^k + (A_{21}^k)^t v_{n_2}^k + e_{11-1}^k \right] + A_{11-12}^k (A_{22}^t v_{n_2}^k + e_{12}^k) - A_{11-1}^k j_1^k \end{aligned} \quad (31)$$

Observe that (31) is identical to (30).

Similarly, if we augment a voltage source between each  $N_1$  node and the datum node and let its terminal voltage waveform be assigned equal to the corresponding node-to-datum voltage, then upon removing all branches in  $\mathcal{B}_1$ , we can show that the equation governing this augmented network is identical to the last component matrix equation of (18).

In analogy to the diakoptics' tearing approach [4], we can now interpret (18) as the result of a tearing process. Physically, we tear the original network  $N$  apart at the  $N_2$  nodes using augmented equivalent voltage sources to account for the "outside" influence upon each subnetwork  $N_k$ . We analyze each subnetwork separately. Finally, we interconnect the solutions via the  $N_2$ -node subnetwork. It is precisely this node-tearing analogy which prompted us to call this approach as the node-tearing nodal analysis.

---

<sup>6</sup>Notations can be found in Fig. 1.

### III. Comparison with Diakoptic Analysis

As already mentioned in the introduction, another tearing approach based on nodal analysis has been developed and will henceforth be referred to as diakoptic nodal analysis [5]. Our objective in this section is to compare the node-tearing nodal analysis (NTNA) with the diakoptic nodal analysis (DNA). To do this, it is instructive to recall briefly the procedure involved in the derivation of DNA. We assume as usual the standard composite branch format shown in Fig. 1 and partition the nondatum node set  $\mathcal{N}$  arbitrarily into two disjoint subsets  $\hat{\mathcal{N}}_1$  and  $\hat{\mathcal{N}}_2$ .<sup>7</sup> We next partition the branch set  $\hat{\mathcal{B}}$  into two disjoint subsets  $\hat{\mathcal{B}}_1$  and  $\hat{\mathcal{B}}_2$  such that no coupling exists between a  $\hat{\mathcal{B}}_1$  branch and a  $\hat{\mathcal{B}}_2$  branch. The reduced incidence matrix  $\hat{A}$  then assumes the following partitioned form:

$$\hat{A} = \begin{matrix} & \hat{\mathcal{B}}_1 & \hat{\mathcal{B}}_2 \\ \hat{\mathcal{N}}_1 & \left[ \begin{matrix} \hat{A}_{11} & \hat{A}_{12} \\ \hat{A}_{21} & \hat{A}_{22} \end{matrix} \right] \\ \hat{\mathcal{N}}_2 & \end{matrix} \quad (32)$$

Depending on the context, let us partition all current and voltage vectors  $\hat{x}$  with respect to either  $\hat{\mathcal{B}}_1$  and  $\hat{\mathcal{B}}_2$ , or  $\hat{\mathcal{N}}_1$  and  $\hat{\mathcal{N}}_2$ ; namely

$$\hat{x} = \begin{bmatrix} \hat{x}_1 \\ \hat{x}_2 \end{bmatrix} \quad (33)$$

Using this notation, KCL, KVL, and the branch relations assume the following form:

---

<sup>7</sup> In this section, we attach a "hat" to all symbols associated with DNA in order to distinguish them from the corresponding symbols associated with NTNA.

### Kirchhoff Current Law

$$\hat{A}_{11}\hat{i}_1 + \hat{A}_{12}\hat{i}_2 = \hat{0}_1 \quad (34)$$

$$\hat{A}_{21}\hat{i}_1 + \hat{A}_{22}\hat{i}_2 = \hat{0}_2 \quad (35)$$

### Kirchhoff Voltage Law

$$\hat{v}_1 = \hat{A}_{11}^t \hat{v}_{n_1} + \hat{A}_{21}^t \hat{v}_{n_2} \quad (36)$$

$$\hat{v}_2 = \hat{A}_{12}^t \hat{v}_{n_1} + \hat{A}_{22}^t \hat{v}_{n_2} \quad (37)$$

### Branch Relations

$$\hat{i}_1 + \hat{j}_1 = \hat{i}_1 = \hat{G}_1 \hat{v}_{n_1} = \hat{G}_1 (\hat{v}_1 + \hat{e}_1) \quad (38)$$

$$\hat{v}_2 + \hat{e}_2 = \hat{v}_2 = \hat{R}_2 \hat{i}_2 = \hat{R}_2 (\hat{i}_2 + \hat{j}_2) \quad (39)$$

Equations (34)-(39) can be combined into the following matrix equation, henceforth called diakoptic nodal analysis (DNA):<sup>8</sup>

$$\begin{matrix} \hat{N}_1 & \hat{N}_2 & \hat{B}_2 \\ \hat{N}_1 & \left[ \begin{array}{ccc} \hat{A}_{11} \hat{G}_1 \hat{A}_{11}^t & \hat{A}_{11} \hat{G}_1 \hat{A}_{21}^t & \hat{A}_{12} \\ \hat{A}_{21} \hat{G}_1 \hat{A}_{11}^t & \hat{A}_{21} \hat{G}_1 \hat{A}_{21}^t & \hat{A}_{22} \\ \hat{A}_{12} & \hat{A}_{22}^t & -\hat{R}_2 \end{array} \right] & \left[ \begin{array}{c} \hat{v}_{n_1} \\ \hat{v}_{n_2} \\ \hat{i}_2 \end{array} \right] \\ \hat{N}_2 & & = \left[ \begin{array}{c} \hat{j}_{s_1} \\ \hat{j}_{s_2} \\ \hat{E}_2 \end{array} \right] \\ \hat{B}_2 & & \end{matrix} \quad (40)$$

$\underbrace{\qquad\qquad\qquad}_{K}$

where

$$\hat{j}_{s_1} \triangleq \hat{A}_{11} (\hat{j}_1 - \hat{G}_1 \hat{e}_1) \quad (41)$$

$$\hat{j}_{s_2} \triangleq \hat{A}_{21} (\hat{j}_1 - \hat{G}_1 \hat{e}_1) \quad (42)$$

$$\hat{E}_2 \triangleq \hat{e}_2 - \hat{R}_2 \hat{j}_2 \quad (43)$$

<sup>8</sup>When  $\hat{N}_2$  is empty, (40) reduces to the so-called radially attached case [15].

We shall later refer to the matrix in (40) as  $K$ .

Observe that if the removal of all  $\hat{B}_2$  branches reduces the original graph  $G$  into " $m$ " disconnected components  $\hat{G}_1^1, \hat{G}_1^2, \dots, \hat{G}_1^m$  (Fig. 11), then the submatrix  $\hat{A}_{11}$  can be partitioned into a block-diagonal form containing " $m$ " blocks. Consequently, Lemma 1 is also applicable to the product  $\hat{A}_{11} \hat{G} \hat{A}_{11}^t$  of (40).

We are now ready to point out the basic distinctions between NTNA and DNA:<sup>9</sup>

(i) As far as the topological conditions are concerned, the NTNA accomplishes the "tearing" in  $m$  separable components by removing the  $\mathcal{N}_2$  nodes and the datum node. In contrast to this, DNA accomplishes the same task by removing the  $\hat{B}_2$  branches. In other words, NTNA involves "node removal" whereas DNA involves "branch removal."

(ii) NTNA uses only voltage variables in the final network equation whereas DNA uses both voltage and current variables.

Let us pause for a moment to introduce a lemma which relates the two methods.

Lemma 2. Let  $G$  be an undirected graph and let  $\hat{B}_2$  be a subset of branches such that the graph  $\hat{G}_1$  obtained from  $G$  by removing all  $\hat{B}_2$  branches contains " $m$ " disconnected components. Then there always exists a subset  $\mathcal{N}_2$  of nodes of  $G$  such that, the graph  $\hat{G}_1$  obtained from  $G$  by removing all  $\mathcal{N}_2$  nodes contains " $r$ " disconnected components, furthermore<sup>10</sup>

<sup>9</sup>The same can be said about diakoptic methods in general [4].

<sup>10</sup>We let  $|S|$  denote the number of elements in the set  $S$ .

(i)  $r \leq m$ ;

(ii)  $|N_1^k| \leq |\hat{N}_1^k|$  where  $\hat{N}_1^k$  denotes the nodes in  $\hat{G}_1^k$ ;

(iii)  $|N_2| \leq |\hat{B}_2|$ .

Proof: For each branch in  $\hat{B}_2$ , we can choose any one of its end nodes and store it in  $N_2$ . Observe the following:

(i) Since, by assumption, the removal of  $\hat{B}_2$  branches separates the graph  $G$  into  $m$  disconnected components, it follows that the removal of  $N_2$  nodes will separate the remaining graph  $G_1$  into at most " $m$ " disconnected components. If some of the disconnected components of  $G_1$  contain isolated nodes which are all chosen as  $N_2$  nodes, then  $G_1$  contains less than " $m$ " separable components.

(ii) Obvious from the construction of  $N_2$ .

(iii) If a node happens to be the end node of two or more  $\hat{B}_2$  branches (Fig. 12), then only this node needs to be put in  $N_2$ . Hence  $|N_2| \leq |\hat{B}_2|$ .  $\square$

Let us now compare NTNA and DNA with respect to their computational complexities. As shown in Appendix A, the parameters in determining the computational complexity are:

(i) the total dimension of the coefficient matrix;

(ii) the dimension of the border of the coefficient matrix;

(iii) the sparsity of the matrix.

Theorem 3. Let  $N$  be a network whose branches are not coupled to each other, let  $\hat{B}_1$  and  $\hat{B}_2$  be the disjoint subsets of the branch partition satisfying the topological condition of diakoptic nodal analysis. Then there always exists a partition of the set  $N$  of nondatum nodes into  $N_1$  and  $N_2$  which satisfies the topological condition of node-tearing nodal

analysis and the following properties:

- (i) the dimension of  $\underline{Y}$  of (18) is less than the dimension of  $\underline{K}$  of (40);
- (ii) the dimension of the border of  $\underline{Y}$  is less than or equal to the dimension of the border of  $\underline{K}$ ;
- (iii)  $\underline{Y}$  is sparser than  $\underline{K}$ .

Proof: (i)  $|\underline{Y}| = |\underline{U}| = |\hat{\underline{U}}_1| + |\hat{\underline{U}}_2| < |\hat{\underline{U}}_1| + |\hat{\underline{U}}_2| + |\hat{\underline{B}}_2| = |\underline{K}|.$

(ii) The dimension of the border of  $\underline{Y}$  is equal to  $|\hat{\underline{U}}_2|$  whereas the dimension of the border of  $\underline{K}$  is equal to  $|\hat{\underline{U}}_2| + |\hat{\underline{B}}_2|$ . It follows from Lemma 2 that  $|\hat{\underline{U}}_2| \leq |\hat{\underline{B}}_2| \leq |\hat{\underline{U}}_2| + |\hat{\underline{B}}_2|$ .

(iii) We want to show that the number of off-diagonal terms in (18) is always less than those in (40). However, a direct comparison is impossible because different submatrices are involved in  $\underline{Y}$  of (18) and  $\underline{K}$  of (40). Therefore, in order to be able to compare directly, we will repartition the nodal admittance matrix  $\underline{Y}$  with respect to the  $\{\hat{\underline{U}}_1, \hat{\underline{U}}_2; \hat{\underline{B}}_1, \hat{\underline{B}}_2\}$  partition, henceforth referred to as  $\hat{\underline{Y}}$ , so that it involves the same submatrices as those contained in  $\underline{K}$  of (40). Observe that the nodal admittance matrix  $\hat{\underline{Y}}$  with respect to  $\{\hat{\underline{U}}_1, \hat{\underline{U}}_2; \hat{\underline{B}}_1, \hat{\underline{B}}_2\}$  and the nodal admittance matrix  $\underline{Y}$  with respect to  $\{\hat{\underline{U}}_1, \hat{\underline{U}}_2; \hat{\underline{B}}_1, \hat{\underline{B}}_2\}$  (i.e., (18)) are both nodal admittance matrices of the same network, and hence can differ from each other only by a symmetric permutation of rows and columns. In other words, they have the same number of nonzero terms. Therefore, we need only to compare the number of nonzero off-diagonal terms of  $\hat{\underline{Y}}$  with those in  $\underline{K}$  of (40).

Let us now partition the nodal admittance matrix with respect to the  $\{\hat{\underline{U}}_1, \hat{\underline{U}}_2; \hat{\underline{B}}_1, \hat{\underline{B}}_2\}$  partition:

$$\begin{aligned}
 \hat{\mathbf{Y}} &\triangleq \hat{\mathbf{A}} \hat{\mathbf{G}} \hat{\mathbf{A}}^T = \begin{bmatrix} \hat{\mathbf{A}}_{11} & \hat{\mathbf{A}}_{12} \\ \hat{\mathbf{A}}_{21} & \hat{\mathbf{A}}_{22} \end{bmatrix} \begin{bmatrix} \hat{\mathbf{G}}_1 & \hat{\mathbf{0}} \\ \hat{\mathbf{0}} & \hat{\mathbf{G}}_2 \end{bmatrix} \begin{bmatrix} \hat{\mathbf{A}}_{11}^T & \hat{\mathbf{A}}_{21}^T \\ \hat{\mathbf{A}}_{12}^T & \hat{\mathbf{A}}_{22}^T \end{bmatrix} \\
 &= \begin{bmatrix} \hat{\mathbf{A}}_{11} \hat{\mathbf{G}}_1 \hat{\mathbf{A}}_{11}^T + \hat{\mathbf{A}}_{12} \hat{\mathbf{G}}_2 \hat{\mathbf{A}}_{12}^T & \hat{\mathbf{A}}_{11} \hat{\mathbf{G}}_1 \hat{\mathbf{A}}_{21}^T + \hat{\mathbf{A}}_{12} \hat{\mathbf{G}}_2 \hat{\mathbf{A}}_{21}^T \\ \hat{\mathbf{A}}_{21} \hat{\mathbf{G}}_1 \hat{\mathbf{A}}_{11}^T + \hat{\mathbf{A}}_{22} \hat{\mathbf{G}}_2 \hat{\mathbf{A}}_{12}^T & \hat{\mathbf{A}}_{21} \hat{\mathbf{G}}_1 \hat{\mathbf{A}}_{21}^T + \hat{\mathbf{A}}_{22} \hat{\mathbf{G}}_2 \hat{\mathbf{A}}_{22}^T \end{bmatrix} \tag{44}
 \end{aligned}$$

$$\begin{aligned}
 &= \underbrace{\begin{bmatrix} \hat{\mathbf{A}}_{11} \hat{\mathbf{G}}_1 \hat{\mathbf{A}}_{11}^T & \hat{\mathbf{A}}_{11} \hat{\mathbf{G}}_1 \hat{\mathbf{A}}_{21}^T \\ \hat{\mathbf{A}}_{21} \hat{\mathbf{G}}_1 \hat{\mathbf{A}}_{11}^T & \hat{\mathbf{A}}_{21} \hat{\mathbf{G}}_1 \hat{\mathbf{A}}_{21}^T \end{bmatrix}}_{\hat{\mathbf{Y}}_1} + \underbrace{\begin{bmatrix} \hat{\mathbf{A}}_{12} \hat{\mathbf{G}}_2 \hat{\mathbf{A}}_{12}^T & \hat{\mathbf{A}}_{12} \hat{\mathbf{G}}_2 \hat{\mathbf{A}}_{22}^T \\ \hat{\mathbf{A}}_{22} \hat{\mathbf{G}}_2 \hat{\mathbf{A}}_{12}^T & \hat{\mathbf{A}}_{22} \hat{\mathbf{G}}_2 \hat{\mathbf{A}}_{22}^T \end{bmatrix}}_{\hat{\mathbf{Y}}_2} \tag{45}
 \end{aligned}$$

where the last decomposition is made for comparison purposes. Observe that since  $\hat{\mathbf{G}}_1$  and  $\hat{\mathbf{G}}_2$  denote the branch conductance matrix associated with branches in  $\hat{\mathcal{B}}_1$  and  $\hat{\mathcal{B}}_2$ , respectively, it follows that  $\hat{\mathbf{Y}}_1$  corresponds to the  $\hat{\mathcal{B}}_1$  branches whereas  $\hat{\mathbf{Y}}_2$  corresponds to the  $\hat{\mathcal{B}}_2$  branches.

Let us now look at matrix  $\mathbf{K}$  of (40). Observe that the four submatrices in the upper left-hand corner of  $\mathbf{K}$ , which are also due to branches in  $\hat{\mathcal{B}}_1$ , are identical to the submatrix  $\hat{\mathbf{Y}}_1$ . Observe also that the submatrices such as  $\{\hat{\mathbf{A}}_{12}, \hat{\mathbf{A}}_{22}, \hat{\mathbf{A}}_{12}^T, \hat{\mathbf{A}}_{22}^T, \hat{\mathbf{R}}_2\}$  are due to the  $\hat{\mathcal{B}}_2$  branches.

Since the branches in  $\hat{\mathcal{B}}_1$  give rise to the same submatrix  $\hat{\mathbf{Y}}_1$  in both  $\hat{\mathbf{Y}}$  of (45) and  $\mathbf{K}$  of (40), therefore the nonzero terms due to the branches in  $\hat{\mathcal{B}}_1$  need not be considered in the comparison. Furthermore,  $\hat{\mathbf{R}}_2$  is assumed diagonal, hence we need only to compare the number of off-diagonal terms in  $\hat{\mathbf{Y}}_2$  and the nonzero terms in  $\{\hat{\mathbf{A}}_{12}, \hat{\mathbf{A}}_{22}, \hat{\mathbf{A}}_{12}^T, \hat{\mathbf{A}}_{22}^T\}$  which are contributed by the branches in  $\hat{\mathcal{B}}_2$ .

Let  $b_k \in \hat{\mathcal{B}}_2$  be a branch connecting nodes  $n_i$  and  $n_j$ . Then,  $\hat{\mathbf{Y}}_{2_{ij}}$  (i.e., the  $(i,j)$ th element of  $\hat{\mathbf{Y}}_2$ ) and  $\hat{\mathbf{Y}}_{2_{ji}}$  are the only off-diagonal terms in  $\hat{\mathbf{Y}}$  that are due to branch  $b_k$ . However, the column of the submatrix

$\begin{bmatrix} \hat{A}_{12} \\ \hat{A}_{22} \end{bmatrix}$  that corresponds to branch  $b_k$  contains a "1" in the i-th row and a "-1" in the j-th row (or vice-versa).<sup>11</sup> In other words, branch  $b_k$  contributes

4 nonzero terms to  $\{\hat{A}_{12}, \hat{A}_{22}, \hat{A}_{12}^t, \hat{A}_{22}^t\}$ . Hence, the following relations always hold:

$$\begin{aligned} & \text{the number of nonzero off-diagonal terms of } Y \text{ in (18)} \\ = & \text{the number of nonzero off-diagonal terms of } \hat{Y} \text{ in (44)} \\ \leq & \text{the number of nonzero off-diagonal terms of } \hat{Y}_1 \text{ and } \hat{Y}_2 \text{ in (45)} \\ < & \text{the number of nonzero off-diagonal terms of } \hat{Y}_1 \\ + & \text{the number of nonzero terms in } \{\hat{A}_{12}, \hat{A}_{22}, \hat{A}_{12}^t, \hat{A}_{22}^t\} \\ = & \text{the number of nonzero off-diagonal terms of } K \text{ in (40)}. \end{aligned}$$

□

Corollary 1 If the  $\hat{B}_1$  branches are coupled to each other, then (18) is superior to (40) in the sense that properties (i), (ii) and (iii) of Theorem 3 are satisfied.

Remark 4. If the  $\hat{B}_2$  branches are coupled to each other, then properties (i) and (ii) of Theorem 3 are still satisfied while property (iii) also holds except in some special cases. For instance, in the case where the  $\hat{B}_2$  branches are very "strongly coupled" to each other,<sup>12</sup> we can find examples where (iii) is false.

Remark 5. For passive networks, the  $Y$  matrix in (18) is diagonally dominant. Hence, for NTNA, any application of the Gaussian Elimination

<sup>11</sup> When branch  $b_k$  connects node  $n_i$  to the datum node, no off-diagonal term in  $\hat{Y}$  will be contributed by  $b_k$  whereas the column of the submatrix  $\begin{bmatrix} \hat{A}_{12} \\ \hat{A}_{22} \end{bmatrix}$  that corresponds to branch  $b_k$  contains a "1" (or "-1") in the i-th row. In other words, the subsequent inequalities still hold.

<sup>12</sup> By strong coupling, we mean  $\hat{G}_2$  and  $\hat{R}_2$  are full matrices.

method with diagonal pivoting on (18) is guaranteed to be stable.

However, in the case of DNA, we can not make a similar statement about (40).

Remark 6. In the derivation of (40) for DNA, we require that the  $\hat{B}_1$  branches are not coupled to the  $\hat{B}_2$  branches. In the derivation of (18) for NTNA, however, we do allow couplings between  $B_1$  and  $B_2$  branches as in (4).

#### IV. Computational Considerations and Graph Optimization Problems Related to Node Tearing

As already pointed out in the introduction and [4], when  $Y$  is in BBDF or BBTF, there exists several efficient methods for decomposing the solution process. This decomposition step is not only convenient but is in fact essential when a very large system of equations has to be solved. Indeed, even by using advanced sparse matrix techniques [16-19], the computer can not handle such large-scale systems efficiently.

In this paper, we critically analyze three decomposition methods which, to the best of our knowledge, are the most efficient solution techniques available; namely, the LU method [20], the Block LU method [4,5] and the TA method [21].<sup>13</sup> Since each of these three methods decomposes the solution process into several stages each involving a relatively small matrix, we can either apply overlay techniques [22] to "stack" the decomposed solution stages, or apply parallel computation techniques [23] to "speed up" the decomposed solution process.

---

<sup>13</sup>TA stands for Tearing Algorithm.

To avoid drifting too far from the main theme of this paper, the detailed description of these methods and a critical study of their computational complexities -- defined as the number of multiplications required -- are given in Appendix A. We note here for later reference that the computational complexity of these methods for both BBDF and BBTF depends on the number of nodes in  $\mathcal{V}_2$  and on the size, i.e., dimension, of the diagonal blocks of  $\mathbf{Y}_{11}$ . In this section, we give only the final results of the comparisons carried out in details in Appendix A.

Comparisons of the computational complexities associated with these three methods for BBDF and BBTF were obtained under the following two assumptions:

- (i) all the nonzero submatrices of  $\mathbf{Y}$  are full;
- (ii) all the submatrices and/or the vector on the right-hand side of (i) are sparse.

The conclusions are summarized in the following Theorem:

Theorem 4 (Relative efficiency of the LU, Block LU and TA methods): For the BBDF, under assumption (i), all three methods yield the same complexity. Under assumption (ii), LU method is always better than Block LU method which in turn is generally<sup>14</sup> better than TA method. For the BBTF, under both assumptions, TA method is always significantly better than the other two methods. □

In Section 2, we have assumed that a partition satisfying the tearing requirements (namely, the topological and branch coupling conditions) was

<sup>14</sup> The above statement is always true under one rather mild condition that there is no fill-in in the lower border of the matrix considered (i.e.,  $\mathbf{Y}_{21}$ ) during Gaussian Elimination.

given apriori. For some problems, a fairly good partition can be obtained from a good layout. However, in general, this is not the case and a partition has to be found.

Observe that in general there exist many distinct partitions of the nondatum node set  $\mathcal{N}$  and the branch set  $\mathcal{B}$  of  $N$  such that the resulting  $\mathbf{Y}$  matrix is in BBDF or BBTF. We can rank these partitions in terms of the computational effort needed to find the solutions of the derived system of equations and choose the optimal one. Since the computational complexity is directly related to both  $|\mathcal{N}_2|$  and the size of the diagonal blocks of  $\mathbf{Y}_{11}$ , we need to minimize both of them. Thus we have a multi-criteria optimization problem to solve.

There exist two general approaches for solving multi-criteria optimization problems [24]. One approach is to convert all but one criterion into suitable constraints. The other approach is to combine the criteria into a single weighted criterion. Fortunately, the choice of our partition is often constrained by an upper bound for the size of the diagonal blocks of  $\mathbf{Y}_{11}$ . This upper bound is usually determined by the capability of the computer used. Hence, it is meaningful to adopt the first approach and consider the following optimization problem for tearing, henceforth called the tearing optimization problem and denoted by TOP:

Find a partition of nodes and branches in  $N$  such that the dimensions of the diagonal blocks of  $\mathbf{Y}_{11}$  are less than or equal to a prescribed upper bound  $n_{\max}$  and such that  $|\mathcal{N}_2|$  is minimized.

Formally, TOP can be stated as follows:

TOP: Minimize  $|\mathcal{N}_2|$  over the family of all distinct partitions  $\{\mathcal{N}_1, \mathcal{N}_2\}$  of nodes of  $G$  such that

- (i) the topological condition holds;
- (ii) the branch coupling condition (1 or 2) holds;
- (iii)  $\sum_{j=1}^k \leq n_{\max}$ ,  $k = 1, 2, \dots, m$ .

Remark 7. In general,  $n_{\max}$  depends on the type of problem we are dealing with, the computer being used and the sparsity of the matrices involved.

Remark 8. TOP is a combinatorial problem and is often more easily studied if it is given a graph-theoretic interpretation. Observe that if the branches are uncoupled, condition (ii) is automatically satisfied and hence can be dropped. Once (ii) is dropped, TOP reduces to a graph optimization problem. However, in general some branches are coupled to each other and (ii) can not be removed.

Observe that TOP actually consists of two problems, depending on whether branch coupling condition 1 or 2 holds. We will prove shortly that they can be reformulated into two equivalent graph optimization problems.

We want to point out again that the primary purpose of NTNA is to partition the nonzero elements of the nodal admittance matrix  $\mathbf{Y}$  in BBDF or BBTF. It is well-known that, in order to deal with the nonzero elements of a matrix efficiently, it is best to work with its associated graph -- the so-called sparsity graph [13, 25, 26]. We will therefore present a graph-theoretic interpretation of TOP using the concept of sparsity graph.

A sparsity graph associated with an  $n \times n$  matrix  $\mathbf{Y}$  is defined as a directed graph  $\hat{\mathcal{G}}_Y^{15}$  containing "n" nodes and a directed branch from node

---

<sup>15</sup>In this section, a hat is used to distinguish symbols associated with a directed graph from symbols associated with an undirected graph.

$n_i$  to node  $n_j$  whenever  $Y_{ij} \neq 0$  (see Fig. 13(a) and (b) for illustration).

The undirected version of a directed graph  $\hat{G}_Y$  is defined as an undirected graph  $\hat{G}_Y$  containing "n" nodes and an undirected branch between nodes  $n_i$  and  $n_j$  whenever there is a directed branch in  $\hat{G}_Y$  from node  $n_i$  to node  $n_j$  (Fig. 13(c)).

We shall now define two graph optimization problems, henceforth denoted by GOP1 and GOP2, on the sparsity graph associated with the nodal admittance matrix  $\underline{Y}$  as follows:<sup>16</sup>

GOP1: Minimize  $|\mathcal{N}_{Y_2}|$  over the family of all distinct partitions

$\{\mathcal{N}_{Y_1}, \mathcal{N}_{Y_2}\}$  of nodes of  $\hat{G}_Y$  such that

(i) the topological condition holds on  $\hat{G}_Y$ ;

(ii)  $|\mathcal{N}_{Y_1}^k| \leq n_{\max}$ ,  $k = 1, 2, \dots, m$ .

GOP2: Minimize  $|\hat{\mathcal{N}}_{Y_2}|$  over the family of all distinct partitions

$\{\hat{\mathcal{N}}_{Y_1}, \hat{\mathcal{N}}_{Y_2}\}$  of nodes of  $\hat{G}_Y$  such that

(i) the section graph  $\hat{G}_Y(\hat{\mathcal{N}}_{Y_1})$  has "m" ( $m > 1$ ) strongly-connected components  $\hat{G}_{Y_1}^1 = (\hat{\mathcal{V}}_{Y_1}^1, \hat{\mathcal{B}}_{Y_1}^{11}), \hat{G}_{Y_1}^2 = (\hat{\mathcal{V}}_{Y_1}^2, \hat{\mathcal{B}}_{Y_1}^{21}), \dots, \hat{G}_{Y_1}^m = (\hat{\mathcal{V}}_{Y_1}^m, \hat{\mathcal{B}}_{Y_1}^{m1})$ ;

(ii)  $|\hat{\mathcal{N}}_{Y_1}^k| \leq n_{\max}$ ,  $k = 1, 2, \dots, m$ .

Before we relate these two problems to TOP, let us prove the following lemma which gives the relationship between the graph  $\hat{G}$  of a network and the sparsity graph  $\hat{G}_Y$  of the nodal admittance matrix  $\underline{Y}$ .

<sup>16</sup>The following notations are the same as those in Section 2 except the subscript "Y" which is used to emphasize that we are dealing with the sparsity graph associated with matrix  $\underline{Y}$ .

Lemma 3. The undirected version of the sparsity graph  $G_Y$  associated with the nodal admittance matrix  $Y$  of  $N$  is the "union" of the section graph  $G(V_1 \cup V_2)$  and the set of branches, connecting nodes in  $V_1 \cup V_2$ , which are induced by the couplings. In other words,<sup>17</sup>  $G_Y = G(V_1 \cup V_2) + \{\text{branches due to couplings}\}$ .

Proof: Observe that the section graph  $G(V_1 \cup V_2)$  can be obtained from  $G$  by removing the datum node and all branches connected to it. Let branch  $b_{k_1 k_2}$  consist of a conductance  $G_{k_1 k_1}$  in parallel with a voltage controlled-current source  $G_{k_1 k_2} v_{k_2}$  (Fig. 14(a)). If we denote the end nodes of branches  $b_{k_1}$  and  $b_{k_2}$  as  $n_{i_1}, n_{j_1}$  and  $n_{i_2}, n_{j_2}$ , respectively, then  $G_{k_1 k_1}$  would contribute nonzero terms to  $Y_{i_1 j_1}$  and  $Y_{j_1 i_1}$ . Consequently, the sparsity subgraph and its undirected version due to  $G_{k_1 k_1}$  are shown in Figs. 14(b) and (c), respectively. Observe that, for the uncoupled case,  $G_Y = G(V_1 \cup V_2)$ . Since  $G_{k_1 k_2}$  would in general contribute nonzero terms to  $Y_{i_1 i_2}, Y_{i_1 j_2}, Y_{j_1 i_2}$  and  $Y_{j_1 j_2}$ , the associated sparsity subgraph and its undirected version due to  $G_{k_1 k_2}$  are shown in Figs. 14(d) and (e), respectively. It is now obvious that  $G_Y = G(V_1 \cup V_2) + \{\text{branches due to couplings}\}$ .  $\square$

Theorem 5. GOPl is equivalent to TOP with branch coupling condition 1.

Proof: (a) GOPl (i)  $\Rightarrow$  TOP (i) and (ii):

From Lemma 3,  $G(V_1 \cup V_2)$  is a subgraph of  $G_Y$ . GOPl (i) implies that

<sup>17</sup> The symbol "+" in this equation is used in the sense that the branches due to couplings are to be added to the graph  $G(V_1 \cup V_2)$  via soldering-iron entries. In other words, no new node is introduced by this augmentation. This notation is used quite often in the graph theory literature [14].

there is no branch connecting  $G_{Y_1}^i$  and  $G_{Y_1}^j$  for  $i \neq j$ . Hence, there is no branch connecting their respective subgraphs  $G_1^i$  and  $G_1^j$ . Hence,  $G_1^i$  and  $G_1^j$  are separable blocks (i.e., TOP (i)) and are not mutually coupled (i.e., TOP (ii)).

(b) TOP (i) and (ii)  $\Rightarrow$  GOP1 (i):

This follows directly from the construction of  $\hat{G}_Y$ . ■

Theorem 6. GOP2 is equivalent to TOP with branch coupling condition 2.

Proof: (a) GOP2 (i)  $\Rightarrow$  TOP (i) and (ii):

Assume there is a branch from  $\hat{G}_{Y_1}^i$  to  $\hat{G}_{Y_1}^j$  ( $i \neq j$ ), then there is no branch from  $\hat{G}_{Y_1}^j$  to  $\hat{G}_{Y_1}^i$ ; otherwise, it would violate the definition of a strongly-connected component. Therefore, this branch must come from the couplings since each original branch in  $\hat{G}$  corresponding to  $G_{kk}$  would produce two branches in  $\hat{G}_Y$  forming a loop. Since each original branch in  $\hat{G}$  remains in one of the strongly-connected components, there is no original branch connecting nodes in  $\hat{G}_{Y_1}^i$  and  $\hat{G}_{Y_1}^j$ . Hence, TOP (i) is satisfied.

Furthermore, if TOP (ii) is violated, then  $Y$  can become structurally symmetric and GOP2 (i) will be violated.

(b) TOP (i) and (ii)  $\Rightarrow$  GOP2 (i):

This follows directly from the construction of  $\hat{G}_Y$ . ■

It has to be noted that both GOP1 and GOP2 are very difficult graph optimization problems. In fact, it can be shown that they belong to a class of hard problems. This class of problems, called NP-complete [10,11], has the property that if any one of them can be solved (i.e., yielding global optimal solution) in polynomial-bounded time with respect to the dimension of the input, all of them can. However, up

until now, no polynomial-bounded solution algorithm has been found for any of these NP-complete problems. It is widely believed that no such polynomial-bounded algorithm exists. The discussion on NP-complete problems and the proof that GOP1 and GOP2 are NP-complete are given in Appendix B.

In general, once we can prove that a problem is NP-complete, we should avoid trying to find a global solution (unless the size of the problem is so small that exponentially-bounded algorithm is acceptable). Instead, some efficient heuristic algorithm [27] should be developed. In the next section, we shall develop such an algorithm for solving GOP1.

## V. A Heuristic Cluster Algorithm

In this section we give a heuristic algorithm to solve one of the optimization problems introduced in Section 4; namely, GOP1. This problem arises in many different fields and is usually referred to as the cluster problem. For example, it is encountered in computer logic and page partitioning problems [28,29], in power system bus clustering problems [30], in network decomposition problems [4,31], in shortest path decomposition problems [32], in IC placement problems [33] and in statistical data grouping problems [34]. Depending on the nature of the problem, the minimization objectives may be based upon the number of interconnection nodes [32], the number of interconnection branches [4,28-31], the total cost of interconnection branches [33] or the distance between the "centroids" of clusters [34]. The various approaches for solving the cluster problems may be classified into four major categories:

- (i) growing clusters from scratch [28,31];
- (ii) interchanging nodes until some local optimality condition is satisfied [33];
- (iii) transforming the problem into some associated mathematical equation [29,34];
- (iv) finding the "contour" of an associated graph [30].

Given an undirected sparsity graph  $G_Y$ , the cluster algorithm to be presented in this section for minimizing the number of interconnection nodes (i.e.,  $|V_{Y_2}|$ ) is based on the last approach and will henceforth be referred to as the contour approach. Before introducing our algorithm, however, let us first discuss the concept of a contour tableau.

A contour tableau consists of an array of three columns as shown in Fig. 15. The leftmost column is called the iterating set (IS), the middle

column the adjacent set (AS) and the rightmost column the contour number (CN). The entries of the tableau are determined as follows:<sup>18</sup>

#### Contour Tableau Construction Algorithm

- Step 1. Choose an initial iterating node and store it in IS(1).
- Step 2. Store in AS(1) all nodes that are adjacent to the node in IS(1).
- Step 3. Place the cardinality of AS(1) in CN(1).
- Step 4. Let  $i = 1$ .
- Step 5. If  $CN(i) = 0$ , stop!
- Step 6. Choose the next iterating node, denoted by  $n_{i+1}$ , from AS( $i$ ) and place it in IS( $i+1$ ).
- Step 7. Update AS( $i+1$ ) from AS( $i$ ) by deleting the node  $n_{i+1}$  and adding the set  $V$  representing all node adjacent to  $n_{i+1}$  that are not already in AS( $i$ ) or  $\left\{ \bigcup_{j=1}^i IS(j) \right\}$ .
- Step 8.  $CN(i+1) = |AS(i+1)|$ .
- Step 9. Let  $i = i+1$ , go to Step 5.

Let us first clarify Step 7 with the aid of Fig. 16. In AS( $i$ ) and AS( $i+1$ ), we store the adjacent nodes of the sets of iterated nodes  $\left\{ \bigcup_{j=1}^i IS(j) \right\}$  and  $\left\{ \bigcup_{j=1}^{i+1} IS(j) \right\}$ , respectively. Instead of finding AS( $i+1$ ) from scratch at each iteration, we want to find an efficient way of updating AS( $i+1$ ) from AS( $i$ ). Now, let us look at Fig. 16 where the solid lines denote adjacency relations and the dotted lines denote possible adjacency relations. Two sets  $\{IS(i+1)\}$  and  $\{AS(i) - IS(i+1)\}$  are adjacent to  $\left\{ \bigcup_{j=1}^i IS(j) \right\}$ . Since  $\{AS(i) - IS(i+1)\}$  and  $V$  are adjacent to  $\left\{ \bigcup_{j=1}^{i+1} IS(j) \right\}$ , we can therefore update AS( $i+1$ ) from AS( $i$ ) by deleting IS( $i+1$ ) and adding  $V$  which is precisely Step 7.

<sup>18</sup>The graph is assumed to be connected for simplicity.

Now, let us pause to look at an example. Figure 17 shows a graph with 9 nodes. It is clustered into two groups of nodes  $\{n_1, n_2, n_3, n_4\}$  and  $\{n_6, n_7, n_8, n_9\}$  which are separated by the hinged node  $n_5$ . Let us start the construction of our contour tableau by selecting arbitrarily the initial node, say  $n_1$ , and store it in  $IS(1)$ . Since  $\{n_2, n_3, n_4, n_5\}$  are the nodes adjacent to  $n_1$ , they are stored in  $AS(1)$ . Consequently,  $CN(1) = 4$ .

Let us choose arbitrarily an iterating node from  $AS(1)$ , say  $n_3$ , and put it in  $IS(2)$ . Observe that the nodes that are adjacent to  $\{n_1, n_3\}$  are  $\{n_2, n_4, n_5\}$ . So they are put in  $AS(2)$  and hence  $CN(2) = 3$ . Choose the next iterating node as  $IS(3) = n_5$ , then  $AS(3) = \{n_2, n_4, n_6, n_7, n_8, n_9\}$  and hence  $CN(3) = 6$ . The complete tableau is shown in Fig. 18(a).

In order to understand how the preceding algorithm can be used to separate the graph into clusters, let us observe that if  $X$  denotes the set of nodes of a given graph, then the set of  $AS$  nodes always separates  $X$  into 3 subsets; namely,  $Z(i) \triangleq \bigcup_{j=1}^i IS(j)$ ,  $AS(i)$ , and  $W(i) \triangleq X - Z(i) - AS(i)$ , where  $Z(i)$  nodes are not adjacent to  $W(i)$  nodes (Fig. 19). As we construct the tableau, the size of  $AS(i)$  (i.e.,  $CN(i)$ ) in each step varies. It is when  $CN(i)$  is very small, henceforth called bottlenecks, that  $Z(i)$  and  $W(i)$  form clusters. Our aim then is to choose a particular contour tabelau construction algorithm that would yield a good cluster whenever  $CN(i)$  encounters a bottleneck. By using arbitrary choices in Steps 1 and 6 as in the preceding example, the best  $AS(i)$  is  $\{n_2, n_4, n_5\}$  (Fig. 18(a)). However, it is far from the optimal result;namely,  $AS(i) = \{n_5\}$ , which in this case can be obtained by inspection.

In the original contour construction algorithm, there are only two places where choices are made. They are in Step 1 when choosing the initial iterating node,and in Step 6 when choosing the next iterating node.

Let us first examine Step 6. In [30], the strategy chosen is the minimum-fill-in strategy which is quite time-consuming and hence inefficient. In this paper, we propose another approach and call it the greedy strategy;<sup>19</sup> namely, at every iteration, we simply choose the node in AS(i) that yields minimum  $CN(i+1) = |AS(i+1)|$  or, equivalently, we choose the node that yields minimum  $|V|$ . If a tie is encountered, we choose arbitrarily among the ties. To illustrate this strategy, we start with  $n_1$  and eventually construct the tableau shown in Fig. 18(b). Indeed, it yields our desired goal; namely, to separate the 2 clusters  $\{n_1, n_2, n_3, n_4\}$  and  $\{n_6, n_7, n_8, n_9\}$  through the bottleneck  $\{n_5\}$ .

Our main reason for choosing the greedy strategy is that it can be easily implemented. To analyze the efficiency of this strategy, we will shortly derive its computational complexity. Before doing this, however, let us first identify its shortcomings by analyzing the example shown in Fig. 20(a). Suppose after the i-th iteration,  $AS(i) = \{n_1, n_2\}$ . If we choose  $n_1$  to iterate next, we will end up with the cluster shown by the dotted line in Fig. 20(b) which has 2 bottleneck nodes. On the other hand, since  $|V(n_1)| = 3$  and  $|V(n_2)| = 2$ , an application of our greedy strategy would require that  $n_2$  be iterated next. The resulting cluster is shown by the dotted line in Fig. 20(c) which has 5 bottleneck nodes. This result of 5 bottleneck nodes versus the possible 2 bottleneck nodes of course is undesirable.

Let us examine next the choice of the initial iterating node. If we start the tableau construction from  $n_5$  in Fig. 17 and use the greedy strategy, then the resulting tableau is shown in Fig. 18(c). Observe

<sup>19</sup>The term "greedy" is a very common terminology in the graph literatures [32]. It means that the algorithm determines the direction for iteration by simply checking some local conditions.

that the basic contour property for identifying the clusters is lost.

Although there exists no optimal procedure to remedy this situation, a good rule of thumb is to start with a node with the minimum degree. In our example, all nodes except  $n_5$  have degree 4. Observe that if we choose any one of them as the starting node, they will all yield a tableau similar to Fig. 18(b). Besides, this minimum-degree strategy coincides with our greedy strategy since a node with the minimum degree will yield a minimum  $CN(1)$ .

We can now incorporate the minimum-degree initial-node strategy for Step 1 and the greedy strategy for Step 6 into the tableau construction algorithm. Furthermore, we can introduce an algorithm for finding clusters based on the contour concept. The flow-chart of this algorithm is shown in Fig. 21. Basically, if we regard  $CN$  as a function of the iteration step (Fig. 22(a)),<sup>20</sup> this algorithm will separate the nodes into clusters whenever there is a local minimum in  $CN$ .

In our original cluster problem, the number of nodes in each cluster is constrained to be less than or equal to  $n_{\max}$ . In the preceding cluster algorithm, this constraint has not yet been taken into consideration. However, we can easily incorporate it by cutting the contour whenever the number of nodes in the cluster reaches  $n_{\max}$  before a local minimum is attained (Fig. 22(b)).

Another assumption that we have made in the preceding cluster algorithm is that the  $CN$  curves in Figs. 22(a) and (b) are very "smooth." In practice, the  $CN$  curve could be very erratic and may in fact contain

---

<sup>20</sup> Although  $CN$  is actually a discrete function of the iteration step, we will approximate it by drawing a continuous curve through these discrete points as shown in Fig. 22.

many small wiggles as illustrated in Fig. 22(c). Moreover, it may also contain many small clusters as in Fig. 22(d). In such situations, our cluster algorithm would simply yield too many clusters each with a very small dimension. Besides, the total number of bottleneck nodes would become too large.

To overcome the occurrence of small clusters, we can delay our searching for a local minimum until after  $n_{\max}$  nodes have been iterated, where  $\alpha \approx 0.6$  to  $0.8$  (Fig. 22(d)). To overcome the occurrence of small wiggles, we can keep a record of all local minima and choose the smallest local minimum that occurs between  $n_{\max}$  and  $n_{\max}$  as the cut-off point. This is illustrated in Fig. 22(e).

Finally, there is the so-called redundancy phenomenon which we will now illustrate with the help of the example shown in Fig. 23(a). This example shows 3 clusters A, B and C separated by bottleneck nodes D, E and F. Let us start with A and use solid lines to denote adjacency relations and dotted lines to denote possible adjacency relations. Using the preceding cluster algorithm, we will end up with the tableau shown in Fig. 23(b) and the associated CN curve shown in Fig. 23(c). Observe that the bottleneck node F is redundant in the sense that it appeared twice as in  $\{D+F\}$  and  $\{E+F\}$ . Therefore, in selecting the best place to cut the CN curve into clusters, we have an inaccurate information because  $|\{D+F\} \cup \{E+F\}| \neq |\{D+F\}| + |\{E+F\}|$ . The resulting cut may not be the best one that is possible. Moreover, it is unnecessary to iterate on D, E and F in the tableau because, once they are determined to be bottleneck nodes, their adjacency is of no more concern to the remaining graph.

To overcome this redundancy phenomenon, we must resort to the concept of dynamic contour cutting; namely, after we have determined the

cluster A and its bottleneck {D+F}, we throw away {D+F} from any future iteration. The dynamic contour cutting strategy will therefore yield a smaller and more efficient tableau as illustrated in Figs. 23(d) and (e).

We are now ready to present a "refined" cluster algorithm which takes into consideration all of the problems identified in the preceding discussions; namely, the  $n_{\max}$  constraint, the small wiggle and small cluster properties of CN curves and the redundancy phenomenon. The flowchart for this refined cluster algorithm is presented in Fig. 24. Let us now analyze the computational complexity<sup>21</sup> of the cluster algorithm.

Theorem 7. Let "n" and "b" denote the number of nodes and branches of the input graph, then the computational complexity of the cluster algorithm is bounded by  $O(nb)$ .

Proof: The most time-consuming step in the cluster algorithm is the choice of the next iterating node from AS. Applying our greedy strategy, each adjacency list [36] of nodes in AS is scanned once. Let  $\ell_0(n_k)$  denote the length of the original adjacency list of node  $n_k$  and let  $\ell_i(n_k)$  denote the length of the adjacency list of node  $n_k$  in AS(i). The reason for distinguishing  $\ell_0(n_k)$ ,  $\ell_1(n_k), \dots$ , is that the adjacency lists actually become shorter after every iteration. Now, the computational bound can be expressed as

$$\sum_{i=1}^n \sum_{n_k \in AS(i)} \ell_k(n_i) = \sum_{k=1}^n \sum_{n_k \in AS(i)} \ell_0(n_i) = n \cdot 2b \quad (46)$$

The last equality holds because each list appear at most  $n$  times in the

---

<sup>21</sup>The complexity used here is defined to be the number of comparisons involved.

whole tableau. Hence the computational complexity of our cluster algorithm is bounded by  $O(nb)$ . □

A computer program for implementing this cluster algorithm has been developed and the detailed results are given in [35]. We will just mention here that the program employs an efficient data structure -- the edge-oriented adjacency list [36] -- and a novel "flag" system in updating the list structures.

Part of the test results are shown in Table 1 which includes a total of 10 examples. For each example we have listed the number of nodes  $n$ , the number of branches  $b$ , the product  $nb$ , the  $n_{\max}$  constraint, the number of clusters yielded by the cluster algorithm, the total number of bottleneck nodes and the computer time spent. In the sequel, we are going to discuss some of these examples in detail.

Let us now examine Example 1 of Table 1 thoroughly, using the graph shown in Fig. 25(a) with  $n_{\max} = 10$ . The tableau derived from our cluster algorithm is shown in Fig. 25(b). Observe that the resulting three clusters coincide with those enclosed by the three dotted lines shown in Fig. 25(a). The bottleneck is identified as  $\{n_8, n_{12}, n_{14}\}$ . This result is quite good since the optimal solution as obtained by inspection consists of one of the following three possibilities =  $\{n_8, n_{10}\}$ ,  $\{n_8, n_{11}\}$  and  $\{n_{10}, n_{11}\}$ .

Nine more examples, i.e., Examples 2-10 of Table 1, are shown in Figs. 26(a)-(i), respectively, where the initial nodes are identified by arrows and the clusters are encircled by dotted lines.

As a final remark about the computational complexity associated with the cluster algorithm, let us plot the computer times spent of Table 1

Example	n = No. of nodes	b = No. of branches	The product nb	$n_{\max}$	No. of clusters	No. of bottleneck nodes	computer time spent 22
1	25	44	1100	10	3	3	.132
2	46	69	3174	19	3	4	.208
3	14	33	462	8	2	2	.087
4	32	54	1728	12	4	4	.159
5	94	176	16544	27	5	6	.502
6	51	126	6426	15	4	9	.328
7	50	100	5000	20	3	7	.275
8	77	180	13860	30	3	10	.465
9	61	164	10004	17	4	9	.426
10	70	180	12600	25	3	12	.443

Table 1. Testing results of the implementation of cluster algorithm.

<sup>22</sup>The computer used is CDC 6400.

versus the product of  $nb$  in Fig. 27. It is clear that  $\mathcal{O}(nb)$  is an upper bound for the complexity because all the data points are bounded by a straight line.

Before we finish this section, let us look at the practical circuit example shown in Fig. 28(a) where the schematic circuit diagram for each operational amplifier is shown in Fig. 28(b) [37]. Using the Ebers-Moll model (Fig. 28(c) [9], each transistor is replaced by a triangular graph in the induced sparsity subgraph (Fig. 28(d)). Our associated graph optimization problem (i.e., Example 5 in Table 1) contains 94 nodes and 176 branches. Since each operational amplifier contains 19 internal nodes, let us choose  $n_{\max} = 27$ . Applying our cluster algorithm, we obtain 5 clusters shown by the dotted lines in Fig. 28(e), where the 1st operational amplifier is split into 2 clusters. This solution is reasonably good unless we demand that each operational amplifier be included in a single cluster. A careful analysis of the tableau shows that the "local" character of our greedy strategy is responsible for the separation of the 1st operational amplifier into two clusters. On the other hand, if one is adamant about retaining each operational amplifier as an inseparable unit within each cluster, then we should transform this problem into the following weighted cluster problem: Transform each operational amplifier into a "super" node with weight 19 (i.e., the total number of internal nodes) and let all other nodes have weight 1. Find the set  $\bigcup_{Y_2}$  with minimum total weight such that each cluster has weight  $\leq n_{\max}$ .

Observe that with some minor modifications, our cluster algorithm is still applicable in solving the above weighted cluster problem.

Finally, we observe that the formulation of any heuristic algorithm invariably involves various trade-offs. The fact that our cluster algorithm

leads to a very reasonable solution with a computational complexity bounded by  $O(n^b)$  shows that ours is indeed a very good heuristic algorithm [27].

## VI. Concluding Remarks

A new tearing approach to the analysis of large-scale electrical networks has been presented. This method, called the node-tearing nodal analysis (NTNA), is based on the well-known nodal analysis.

The contribution of this paper lies mainly in two areas: circuit theory and graph theory. In the circuit-theoretic part, NTNA has been derived as the result of a simple partition of nodes and branches and a straight-forward manipulation of KCL, KVL and the branch constitutive relations. NTNA has also been proved to be in general superior to Wu's diakoptic nodal analysis. Moreover, it has been pointed out that, for passive networks, NTNA yields a system of equations which involves a diagonally-dominant matrix. This property guarantees that any application of the Gaussian Elimination method with diagonal pivoting is stable. The stability of such a process is often crucial when nonlinear networks are analyzed by computers.

In the graph-theoretic part, graph optimization problems have been related to the problem of finding the most efficient NTNA for a given network. These problems, which involve the partition of nodes and branches of an appropriate graph, have been shown to belong to a class of difficult combinatorial problems, the so-called NP-complete class, where no efficient global solution can be expected. Consequently, an efficient heuristic algorithm has been presented for the solution of a particular graph-theoretic optimization problem, the so-called cluster problem, which arises

also in many other research areas.

Finally, we wish to point out that node-tearing nodal analysis (NTNA) is but a particular formulation in a whole new class of formulations. In particular, a straight-forward extention to loop analysis and cutset analysis in either bordered-block-diagonal form (BBDF), or bordered-block-triangular form (BBTF), is given in Appendix C.

Appendix A. Comparison of Three Decomposition Methods.

Given a system of linear algebraic equations

$$M\mathbf{x} = \mathbf{s}$$

(A.1)

We shall present and compare three decomposition methods for solving the following two specialized matrix structures:

(i) Bordered-Block-Diagonal Form (BBDF)

$$\left[ \begin{array}{ccccc} M_{11} & & & M_{1,m+1} & x_1 \\ & 0 & & M_{2,m+1} & x_2 \\ M_{22} & & & \vdots & \vdots \\ \vdots & & M_{mm} & M_{m,m+1} & x_m \\ 0 & & & \vdots & \vdots \\ M_{m+1,1} & M_{m+1,2} & \cdots & M_{m+1,m} & M_{m+1,m+1} \\ \hline \end{array} \right] \left[ \begin{array}{c} x_1 \\ x_2 \\ \vdots \\ x_m \\ x_{m+1} \end{array} \right] = \left[ \begin{array}{c} s_1 \\ s_2 \\ \vdots \\ s_m \\ s_{m+1} \end{array} \right] \quad (A.2)$$

(ii) Bordered-Block-Triangular Form (BBTF)

$$\left[ \begin{array}{ccccc} M_{11} & & & M_{1,m+1} & x_1 \\ M_{21} & M_{22} & 0 & M_{2,m+1} & x_2 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ M_{m1} & M_{m2} & \cdots & M_{mm} & M_{m,m+1} \\ \hline M_{m+1,1} & M_{m+1,2} & \cdots & M_{m+1,m} & M_{m+1,m+1} \end{array} \right] \left[ \begin{array}{c} x_1 \\ x_2 \\ \vdots \\ x_m \\ x_{m+1} \end{array} \right] = \left[ \begin{array}{c} s_1 \\ s_2 \\ \vdots \\ s_m \\ s_{m+1} \end{array} \right] \quad (A.3)$$

These two structures are frequently encountered in the analysis of large-scale systems, where the dimension of the associated matrix  $M$  is often too large to be analyzed efficiently, even by large computers. Fortunately, these two matrices are endowed with a special structure which allows the original system to be decomposed and solved in several stages, each involving a matrix of a much lower dimension. Consequently, this method of analysis is often called the small computer approach for solving large-scale problems.

We shall discuss and compare three distinct decomposition methods for solving each of the two matrix structures; namely, the LU method, the Block LU method and the TA method. Since matrices associated with large-scale systems are usually very sparse, any meaningful comparison must take into consideration the sparsity of the submatrices  $M_{ij}$  and vectors  $s_i$  in (A.2) and (A.3).

#### A.1. Comparison of the LU, Block LU and TA Methods for Solving Matrices in Bordered-Block-Diagonal Form

We will first present the three methods in a form most convenient for their subsequent comparisons.

##### A.1.1. The Three Methods for BBDF

###### (a) The LU method [20]

Step 1. Factorize  $M$  as

$$\underline{M} = \underline{L}\underline{U} \quad (\text{A.4})$$

where  $\underline{L}$  is a lower-triangular matrix and  $\underline{U}$  is a unit-upper-triangular matrix.

Step 2. Forward substitute for  $\underline{w}$  from

$$\underline{L}\underline{w} = \underline{s} \quad (\text{A.5})$$

where

$$\underline{w} \stackrel{\Delta}{=} \underline{U}\underline{x} \quad (\text{A.6})$$

Step 3. Backward substitute for  $\underline{x}$  from (A.6).

Let us make two observations. First, the triangular matrices  $\underline{L}$  and  $\underline{U}$  associated with (A.2) have the following block structures

$$\underline{L} = \begin{bmatrix} L_{11} & & & & \\ & L_{22} & & & \\ & & \ddots & & \\ & & & L_{mm} & \\ & L_{m+1,1} & L_{m+1,2} & \cdots & L_{m+1,m} & L_{m+1,m+1} \end{bmatrix} \quad (A.7)$$

$$\underline{U} = \begin{bmatrix} U_{11} & & & & U_{1,m+1} \\ & U_{22} & & & U_{2,m+1} \\ & & \ddots & & \vdots \\ & & & U_{mm} & U_{m,m+1} \\ & & & & U_{m+1,m+1} \end{bmatrix} \quad (A.8)$$

where  $\underline{L}_{kk}$  is lower-triangular and  $\underline{U}_{kk}$  is unit-upper-triangular for  $1 \leq k \leq m+1$ . Second, during the factorization of  $M_{kk}$  for  $1 \leq k \leq m$  in Step 1, only the submatrices  $M_{kk}$ ,  $M_{k,m+1}$ ,  $M_{m+1,k}$  and  $M_{m+1,m+1}^k$  are involved, where  $M_{m+1,m+1}^k$  is recursively defined as follows:

$$M_{m+1,m+1}^1 \triangleq M_{m+1,m+1} \quad (A.9)$$

$$M_{m+1,m+1}^{k+1} \triangleq M_{m+1,m+1}^k - L_{m+1,k} U_{k,m+1} \quad (A.10)$$

It follows from the preceding observations that the LU method can be reformulated into the following equivalent form:

Step 1. For  $k = 1, 2, \dots, m$ , factorize the following submatrix

$$\begin{bmatrix} M_{kk} & M_{k,m+1} \\ M_{m+1,k} & M_{m+1,m+1}^k \end{bmatrix} = \begin{bmatrix} L_{kk} & 0 \\ L_{m+1,k} & M_{m+1,m+1}^{k+1} \end{bmatrix} \begin{bmatrix} U_{kk} & U_{k,m+1} \\ 0 & I_{m+1,m+1} \end{bmatrix} \quad (A.11)$$

and factorize

$$\underline{M}_{m+1,m+1}^{m+1} = \underline{L}_{m+1,m+1} \underline{U}_{m+1,m+1} \quad (A.12)$$

Step 2. Forward substitute for  $\underline{w}$  block-by-block from

$$\underline{L}_{kk} \underline{w}_k = \underline{s}_k \quad (A.13)$$

for  $k = 1, 2, \dots, m$ , and

$$\underline{L}_{m+1,m+1} \underline{w}_{m+1} = \underline{s}_{m+1} - \sum_{k=1}^m \underline{L}_{m+1,k} \underline{w}_k \quad (A.14)$$

where  $\underline{w}$ ,  $\underline{s}$ ,  $\underline{x}$  are partitioned into the following compatible blocks:

$$\underline{w} = \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_m \\ \hline \cdots \\ w_{m+1} \end{bmatrix}, \quad \underline{s} = \begin{bmatrix} s_1 \\ s_2 \\ \vdots \\ s_m \\ \hline \cdots \\ s_{m+1} \end{bmatrix}, \quad \underline{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_m \\ \hline \cdots \\ x_{m+1} \end{bmatrix} \quad (A.15)$$

Step 3. Backward substitute for  $\underline{x}$  block-by-block from

$$\underline{U}_{m+1,m+1} \underline{x}_{m+1} = \underline{w}_{m+1} \quad (A.16)$$

and, for  $k=1, 2, \dots, m$ , solve

$$\underline{U}_{kk} \underline{x}_k = \underline{w}_k - \underline{U}_{k,m+1} \underline{x}_{m+1} \quad (A.17)$$

### (b) The Block LU method [4,5]

Step 1. For  $k = 1, 2, \dots, m$ , factorize each submatrix  $\underline{M}_{kk}$  as

$$\underline{M}_{kk} = \underline{L}_{kk} \underline{U}_{kk} \quad (A.18)$$

Step 2. For  $k = 1, 2, \dots, m$ , backward substitute for  $\underline{L}_{m+1, k}$  from

$$\underline{L}_{m+1, k} \underline{U}_{kk} = \underline{M}_{m+1, k} \quad (\text{A.19})$$

Step 3. For  $k = 1, 2, \dots, m$ , forward substitute for  $\underline{U}_{k, m+1}$  from

$$\underline{L}_{kk} \underline{U}_{k, m+1} = \underline{M}_{k, m+1} \quad (\text{A.20})$$

Step 4. Factorize  $\underline{M}_{m+1, m+1}^{m+1}$  as in (A.12) where

$$\underline{M}_{m+1, m+1}^{m+1} = \underline{M}_{m+1, m+1} - \sum_{k=1}^m \underline{L}_{m+1, k} \underline{U}_{k, m+1} \quad (\text{A.21})$$

Step 5. Forward substitute for  $\underline{y}$  block-by-block from (A.13) and (A.14)

Step 6. Backward substitute for  $\underline{x}$  block-by-block from (A.16) and (A.17).

On first sight, it might appear that the LU method is the same as the Block LU method. However, this observation is true only if all the submatrices are full. For sparse submatrices they are actually quite different and we will show shortly that, under this condition, the LU method is always better.

### (c) The TA method [21]

This method is relatively new and we shall briefly describe it here.

Assume the matrix  $\underline{M}$  of (A.1) can be decomposed as

$$\underline{M} = \underline{P} + \underline{H}\underline{K} \quad (\text{A.22})$$

where  $\underline{M}$  and  $\underline{P}$  are  $\ell \times \ell$  nonsingular matrices, and  $\underline{H}$  and  $\underline{K}$  are respectively  $\ell \times q$  and  $q \times \ell$  (where  $q \leq \ell$ ) matrices of rank  $q$ . Instead of solving (A.1) directly, we can first solve

$$\underline{P}\underline{y}^0 = \underline{s} \quad (\text{A.23})$$

Then, by modifying the solution  $\underline{y}^0$  "appropriately," we can obtain  $\underline{x}$ .

In particular, given the decomposition (A.22), let  $\underline{H}_p$  denote the p-th column of  $\underline{H}$  and let  $\underline{H}_0 \triangleq \underline{s}$ , then the following modification algorithm can be used to compute for  $\underline{x}$ :<sup>23</sup>

Step 1. Factorize  $\underline{P}$  as

$$\underline{P} = \hat{\underline{L}} \hat{\underline{U}} \quad (\text{A.24})$$

where  $\hat{\underline{L}}$  is lower-triangular and  $\hat{\underline{U}}$  is unit-upper-triangular.

Step 2. For  $p = 0, 1, 2, \dots, q$ , solve  $\underline{y}^p$  (via substitution) from

$$\hat{\underline{L}} \hat{\underline{U}} \underline{y}^p = \underline{H}_p \quad (\text{A.25})$$

Step 3. Solve  $\underline{z}$  from

$$(I_{qq} + D) \underline{z} = \underline{K} \underline{y}^0 \quad (\text{A.26})$$

where  $I_{qq}$  is a  $qxq$  unit matrix and

$$D \triangleq [K\underline{y}^1; K\underline{y}^2; \dots; K\underline{y}^q] \quad (\text{A.27})$$

Step 4. Obtain the original solution  $\underline{x}$  of (A.1) by

$$\underline{x} = \underline{y}^0 - \sum_{p=1}^q z_p \underline{y}^p \quad (\text{A.28})$$

where  $z_p$  is the p-th component of  $\underline{z}$ .

<sup>23</sup> An alternate formulation as given in [21] is as follows:

Step 1.  $\underline{P} = \hat{\underline{L}} \hat{\underline{U}}$

Step 2. Solve  $\hat{\underline{y}}^p$  from  $\hat{\underline{L}} \hat{\underline{U}} \hat{\underline{y}}^p = -\underline{H}_p$  for  $p = 1, 2, \dots, q$ , and solve  $\underline{y}^0$  from  $\hat{\underline{L}} \hat{\underline{U}} \underline{y}^0 = \underline{s}$ .

Step 3.  $(I_{qq} - \hat{D}) \hat{\underline{z}} = \underline{K} \underline{y}^0$  where  $\hat{D} \triangleq [\hat{K} \hat{\underline{y}}^1; \hat{K} \hat{\underline{y}}^2; \dots; \hat{K} \hat{\underline{y}}^q]$ .

Step 4.  $\underline{x} = \underline{y}^0 + \hat{z}_p \underline{y}^p$  where  $\hat{z}_p$  is the p-th component of  $\hat{\underline{z}}$ .

The validity of (A.28) is proved in [21]. However, a much simpler proof is given below:

Proof of (A.28):

$$\begin{aligned}
 \underline{M}(\underline{y}^0 - \sum_{p=1}^q z_p \underline{y}^p) &= \underline{M}\underline{y}^0 - \sum_{p=1}^q z_p \underline{M}\underline{y}^p \\
 &= \underline{P}\underline{y}^0 + \underline{H}\underline{K}\underline{y}^0 - \sum_{p=1}^q z_p \underline{P}\underline{y}^p - \sum_{p=1}^q z_p \underline{H}\underline{K}\underline{y}^p \\
 &= \underline{s} + \underline{H}\underline{K}\underline{y}^0 - \sum_{p=1}^q z_p \underline{H}\underline{y}^p - H \sum_{p=1}^q z_p \underline{K}\underline{y}^p \\
 &= \underline{s} + \underline{H}\underline{K}\underline{y}^0 - \underline{H}\underline{z} - \underline{H}\underline{D}\underline{z} \\
 &= \underline{s} + \underline{H}(\underline{K}\underline{y}^0 - (\underline{I}_{qq} + \underline{D})\underline{z}) \\
 &= \underline{s} = \underline{M}\underline{x}
 \end{aligned}$$

Premultiplying both sides by  $\underline{M}^{-1}$ , we obtain (A.28).  $\square$

Observe that the effectiveness of the TA method depends on the following two requirements regarding the decomposition given in (A.22):

- (i)  $\underline{P}$  should be easily factorizable into (A.24) and the resulting equation be readily solvable.
- (ii) The border size "q" should be made as small as possible.

Our experience shows that for matrices in the bordered-block-diagonal form, the best decomposition of  $\underline{M}$  is given by [21]

$$\underline{P} = \left[ \begin{array}{cccc|cc}
 \underline{M}_{11} & & & & & & \\
 & \underline{M}_{22} & & & & & \\
 & & \ddots & & & & \\
 & & & \underline{M}_{mm} & & & \\
 \hline
 & & & & \underline{0} & & \\
 & & & & & \underline{0} & \\
 & & & & & & \\
 & & & & & & \\
 & \underline{M}_{m+1,1} & \underline{M}_{m+1,2} & \cdots & \underline{M}_{m+1,m} & \underline{1}_{m+1,m+1} & 
 \end{array} \right], \quad \underline{H} = \left[ \begin{array}{c}
 \underline{M}_{1,m+1} \\
 \underline{M}_{2,m+1} \\
 \vdots \\
 \underline{M}_{m,m+1} \\
 \hline
 \underline{M}_{m+1,m+1} - \underline{1}_{m+1,m+1}
 \end{array} \right] \quad (A.29)$$

$$\underline{K} = \begin{bmatrix} & & \\ & 0 & \\ & & \vdots \end{bmatrix} \quad (A.30)$$

Observe that the matrix  $\underline{P}$  in (A.29) can be easily factorized. In fact, for  $k = 1, 2, \dots, m$ ,  $M_{kk}$  can be factorized individually as in (A.18). Also observe that the resulting equation can be easily solved since each block in  $\underline{P}$  is decoupled from each other. Moreover, the dimension of  $M_{m+1, m+1}$  (i.e.,  $q$ ) is usually very small. Hence both requirements are met for the above decomposition. In addition, we obtain the following two desirable features:

- (i)  $K_y^P$  requires no calculation for  $p = 0, 1, 2, \dots, q$ .
- (ii)  $\underline{z}$  as obtained from (A.26) is equal to  $x_{k+1}$  of (A.15) and hence we can save part of the calculation of (A.28).

#### A.1.2. Computational Complexities of the Three Methods for BBDF

Now that we have presented all three methods, let us compare their computational complexities<sup>24</sup> with respect to the sparsity of the matrix involved. The first case to be considered is when all submatrices are full, i.e., the zero elements in the submatrices are too few to be useful. The second case is when all submatrices are sparse, i.e., the zero elements in the submatrices are abundant and can be efficiently utilized. The third case is when the right-hand-side vector is also sparse.

For simplicity, let us assume throughout that  $M_{11}, M_{22}, \dots, M_{mm}$  all have the same dimension  $n \times n$  while that of  $M_{m+1, m+1}$  is assumed to be  $q \times q$ . Let  $C_{LU}$ ,  $C_{BLU}$  and  $C_{TA}$  respectively denote the computational complexity of the LU method, the Block LU method and the TA method.

##### Case 1. All Submatrices are Full

The derivation for  $C_{LU}$ ,  $C_{BLU}$  and  $C_{TA}$  for this case is given as follows:

<sup>24</sup>Only multiplications are counted.

$$\begin{aligned}
C_{LU} &= \left\{ m \left[ \left( \frac{n^3}{3} - \frac{n}{3} \right) + \frac{qn(n+1)}{2} + \frac{qn(n-1)}{2} + q^2 n \right] + \left( \frac{q^3}{3} - \frac{q}{3} \right) \right\} (\text{the above computations are due to Step 1 of LU method}) + \left\{ m \left[ \frac{n(n+1)}{2} \right] \right. \\
&\quad \left. + \left[ mqn + \frac{q(q+1)}{2} \right] \right\} (\text{Step 2}) + \left\{ \frac{q(q-1)}{2} + m \left[ \frac{n(n-1)}{2} + qn \right] \right\} (\text{Step 3}) \\
&= m \left( \frac{n^3}{3} - \frac{n}{3} + qn^2 + q^2 n + n^2 + 2qn \right) + \frac{q^3}{3} - \frac{q}{3} + q^3 \tag{A.31}
\end{aligned}$$

$$\begin{aligned}
C_{BLU} &= \left\{ m \left( \frac{n^3}{3} - \frac{n}{3} \right) \right\} (\text{Step 1}) + \left\{ mq \frac{n(n-1)}{2} \right\} (\text{Step 2}) + \left\{ mq \frac{n(n+1)}{2} \right\} (\text{Step 3}) \\
&\quad + \left\{ mq^2 n + \frac{q^3}{3} - \frac{q}{3} \right\} (\text{Step 4}) + \left\{ m \frac{n(n+1)}{2} + mqn + \frac{q(q+1)}{2} \right\} (\text{Step 5}) \\
&\quad + \left\{ \frac{q(q-1)}{2} + m \left[ \frac{n(n-1)}{2} + qn \right] \right\} (\text{Step 6}) \\
&= m \left( \frac{n^3}{3} - \frac{n}{3} + qn^2 + q^2 n + n^2 + 2qn \right) + \frac{q^3}{3} - \frac{q}{3} + q^2 \tag{A.32}
\end{aligned}$$

$$\begin{aligned}
C_{TA} &= \left\{ m \left( \frac{n^3}{3} - \frac{n}{3} \right) \right\} (\text{Step 1}) + \left\{ (q+1) m \left[ \frac{n(n+1)}{2} + \frac{n(n-1)}{2} + qn \right] \right\} (\text{Step 2}) \\
&\quad + \left\{ \frac{q^3}{3} + q^2 - \frac{q}{3} \right\} (\text{Step 3}) + \left\{ qmn \right\} (\text{Step 4}) \\
&= m \left( \frac{n^3}{3} - \frac{n}{3} + qn^2 + q^2 n + n^2 + 2qn \right) + \frac{q^3}{3} - \frac{q}{3} + q^2 \tag{A.33}
\end{aligned}$$

As might have been expected, all three methods yield the same computational complexity when all submatrices are full.

#### Case 2. All Submatrices are Sparse

In this case, it is essential to consider the pivoting order used in the factorization of sparse matrices. It is well known [38] that different pivoting order may yield drastically different fill-in patterns. Therefore, it is desirable to choose pivoting order that yields minimum fill-ins. Since the optimal pivoting order for one method is not necessarily optimal for another method, we have to justify using the same optimal pivoting order for all three methods before any meaningful comparisons can be made.

The optimal pivoting order for the LU method minimizes fill-ins in the submatrices  $\hat{M}_{kk}$ ,  $\hat{M}_{k,m+1}$ ,  $\hat{M}_{m+1,k}$  and  $\hat{M}_{m+1,m+1}^k$  for  $k = 1, 2, \dots, m$ . The optimal pivoting order for the Block LU method, on first sight, seems to minimizes fill-ins in the submatrices  $\hat{M}_{kk}$  for  $k = 1, 2, \dots, m$ . However, from [20] we know that the two submatrices  $\hat{L}_{m+1,k}$  and  $\hat{U}_{k,m+1}$  obtained from the Block LU method are identical to those obtained from the LU method. Hence, if we want to also minimize the fill-ins occurring at Steps 2-4 of the Block LU method, the same minimization criterion as that of the LU method should be used.

From the observation that  $\hat{M}_{m+1,m+1}^{m+1}$  is usually a full matrix, there is little reason to keep track of the fill-ins in  $\hat{M}_{m+1,m+1}^k$  at each step. Therefore, a more meaningful optimal pivoting order for both methods is to minimize fill-ins in the submatrices  $\hat{M}_{kk}$ ,  $\hat{M}_{k,m+1}$  and  $\hat{M}_{m+1,k}$ .

With our choice of the decomposition of  $\hat{M}$  as in (A.29) and (A.30), the obvious optimal pivoting order for the TA method is to minimize the fill-ins of all submatrices  $\hat{M}_{kk}$ . However, we will now consider a better, though more subtle, optimal choice. In Step 2 of the TA method, (A.25) is solved  $(m+1)$  times with respect to different vectors on the right-hand side. As we shall see shortly, the zeroes of the right-hand side vectors can be efficiently utilized if they are located at the top of each vector. This requirement somewhat coincides with that requiring minimum fill-ins for all  $\hat{M}_{k,m+1}$ . In other words, a better choice for the optimal pivoting order would be to minimize fill-ins in the submatrices  $\hat{M}_{kk}$  and  $\hat{M}_{k,m+1}$ . Consequently, for symmetric matrices or almost symmetric matrices, the same optimal pivoting order holds for all three methods.

To simplify our subsequent derivation, it is convenient to introduce the following notations<sup>25</sup> (see Fig. A.1):

<sup>25</sup>Notations with a hat denote the number of nonzero elements after fill-in has occurred.

(i)  $(\hat{c}_i^k + 1)$  (resp.;  $(\hat{r}_i^k + 1)$ ) denotes the number of nonzero elements in the first column (resp.; row) of the reduced matrix of order  $(n-i+1)$  of  $\underline{M}_{kk}$ .

(ii)  $(\hat{c}_i^k + 1)$  (resp.;  $(\hat{r}_i^k + 1)$ ) denotes the number of nonzero elements in the first column (resp.; row) of the reduced matrix of order  $(n-i+1)$  during the  $i$ -th step of Gaussian Elimination (or factorization) of  $\underline{M}_{kk}$ .

(iii)  $d_i^k$  (resp.;  $\hat{d}_i^k$ ) denotes the number of nonzero elements in column  $i$  of  $\underline{M}_{m+1,k}$  (resp.;  $\underline{L}_{m+1,k}$ ).

(iv)  $e_i^k$  (resp.;  $\hat{e}_i^k$ ) denotes the number of nonzero elements in row  $i$  of  $\underline{M}_{k,m+1}$  (resp.;  $\underline{U}_{k,m+1}$ ).

For simplicity, we shall omit writing the bounds of the indices over the summation sign  $\sum$ . The actual bounds, unless otherwise given, are defined as follow:

$$\sum_i \triangleq \sum_{i=1}^n, \quad \sum_k \triangleq \sum_{k=1}^m, \quad \sum_p \triangleq \sum_{p=1}^q \quad (A.34)$$

Finally, we shall treat  $\underline{M}_{m+1,m+1}^{m+1}$  as a full matrix.

Using the preceding assumptions and notations, we are now ready to derive the computational complexities  $C_{LU}$ ,  $C_{BLU}$  and  $C_{TA}$  for case 2 as follows:

$$\begin{aligned} C_{LU} &= \left\{ \sum_k \sum_i (\hat{c}_i^k + \hat{d}_i^k + 1) (\hat{r}_i^k + \hat{e}_i^k) + \frac{q^3}{3} - \frac{q}{3} \right\} \text{(Step 1)} + \left\{ \sum_k \sum_i (\hat{c}_i^k + 1) + \frac{q(q+1)}{2} \right. \\ &\quad \left. + \sum_k \sum_i \hat{d}_i^k \right\} \text{(Step 2)} + \left\{ \sum_k \left( \sum_i \hat{r}_i^k + \sum_i \hat{e}_i^k \right) + \frac{q(q-1)}{2} \right\} \text{(Step 3)} \\ &= \sum_k \sum_i \hat{c}_i^k \hat{r}_i^k + \sum_k \sum_i \hat{c}_i^k \hat{e}_i^k + \sum_k \sum_i \hat{d}_i^k \hat{r}_i^k + \sum_k \sum_i \hat{d}_i^k \hat{e}_i^k + 2 \sum_k \sum_i \hat{r}_i^k \\ &\quad + 2 \sum_k \sum_i \hat{e}_i^k + \sum_k \sum_i \hat{c}_i^k + \sum_k \sum_i \hat{d}_i^k + \left( \frac{q^3}{3} + q^2 - \frac{q}{3} + mn \right) \quad (A.35) \end{aligned}$$

$$\begin{aligned}
C_{BLU} &= \left\{ \sum_k \sum_i (\hat{c}_{i+1}^k) \hat{r}_i^k \right\} \text{(Step 1)} + \left\{ \sum_k \left( q \sum_i \hat{r}_i^k \right) \right\} \text{(Step 2)} \\
&\quad + \left\{ \sum_k \left[ q \sum_i (\hat{c}_{i+1}^k) \right] \right\} \text{(Step 3)} + \left\{ \sum_k \sum_i \hat{d}_i^k \hat{e}_i^k + \frac{q^3}{3} - \frac{q}{3} \right\} \text{(Step 4)} \\
&\quad + \left\{ \sum_k \sum_i (\hat{c}_{i+1}^k) + \frac{q(q+1)}{2} + \sum_k \sum_i \hat{d}_i^k \right\} \text{(Step 5)} + \left\{ \frac{q(q-1)}{2} \right. \\
&\quad \left. + \sum_k \left( \sum_i \hat{r}_i^k + \sum_i \hat{e}_i^k \right) \right\} \text{(Step 6)} \\
&= \sum_k \sum_i \hat{c}_i^k \hat{r}_i^k + \sum_k \sum_i \hat{c}_i^k q + \sum_k \sum_i q \hat{r}_i^k + \sum_k \sum_i \hat{d}_i^k \hat{e}_i^k \\
&\quad + 2 \sum_k \sum_i \hat{r}_i^k + \left( \sum_k \sum_i \hat{e}_i^k + mqn \right) + \sum_k \sum_i \hat{c}_i^k + \sum_k \sum_i \hat{d}_i^k \\
&\quad + \left( \frac{q^3}{3} + q^2 - \frac{q}{3} + mn \right) \tag{A.36}
\end{aligned}$$

$$\begin{aligned}
C_{TA} &= \left\{ \sum_k \sum_i (\hat{c}_{i+1}^k) \hat{r}_i^k \right\} \text{(Step 1)} + \left\{ (q+1) \sum_k \left[ \sum_i (\hat{c}_{i+1}^k) + \sum_i \hat{r}_i^k + \sum_i \hat{d}_i^k \right] \right\} \\
&\quad \text{(Step 2)} + \left\{ \frac{q^3}{3} + q^2 - \frac{q}{3} \right\} \text{(Step 3)} + \left\{ mqn \right\} \text{(Step 4)} \\
&= \sum_k \sum_i \hat{c}_i^k \hat{r}_i^k + \sum_k \sum_i \hat{c}_i^k q + \sum_k \sum_i q \hat{r}_i^k + \sum_k \sum_i q \hat{d}_i^k + 2 \sum_k \sum_i \hat{r}_i^k \\
&\quad + 2mqn + \sum_k \sum_i \hat{c}_i^k + \sum_k \sum_i \hat{d}_i^k + \left( \frac{q^3}{3} + q^2 - \frac{q}{3} + mn \right) \tag{A.37}
\end{aligned}$$

In order to compare  $C_{LU}$  and  $C_{BLU}$ , we will make use of the following three inequalities:

$$\sum_k \sum_i \hat{c}_i^k \hat{e}_i^k \leq \sum_k \sum_i \hat{c}_i^k q \tag{A.38}$$

$$\sum_k \sum_i \hat{d}_i^k \hat{r}_i^k \leq \sum_k \sum_i q \hat{r}_i^k \tag{A.39}$$

$$\sum_k \sum_i \hat{e}_i^k \leq mqn \tag{A.40}$$

Using (A.38)-(A.40) and comparing (A.35) with (A.36), we obtain

$$C_{LU} \leq C_{BLU} \quad (A.41)$$

Observe that  $C_{TA}$  and  $C_{BLU}$  (or  $C_{LU}$ ) are not readily comparable because the former expression contains  $d_i^k$  while the latter contains  $\hat{d}_i^k$ . Subtracting (A.37) from (A.36), we obtain

$$C_{BLU} - C_{TA} = \sum_k \sum_i \left[ (\hat{d}_i^k \hat{e}_i^k - q d_i^k) + (\hat{e}_i^k - q) + (\hat{d}_i^k - d_i^k) \right] \quad (A.42)$$

If we assume  $\hat{d}_i^k \approx d_i^k$ , then

$$C_{BLU} \leq C_{TA} \quad (A.43)$$

However, if we assume  $\hat{d}_i^k \approx \hat{e}_i^k \approx q > d_i^k$ , then

$$C_{BLU} \geq C_{TA} \quad (A.44)$$

Observe that if we choose the optimal pivoting order for the TA method to be that which minimizes fill-ins in  $M_{kk}$ ,  $M_{k,m+1}$  and  $M_{m+1,k}$ , then the assumption  $\hat{d}_i^k \approx d_i^k$  is quite reasonable. Actually, if  $\hat{d}_i^k = d_i^k$ , then it has the interpretation that there is no fill-in in  $M_{m+1,k}$ . Now, we can conclude that, for sparse submatrices in (A.2), the LU method is always better than the Block LU method, which in turn is better than the TA method under the assumption of  $\hat{d}_i^k \approx d_i^k$ .

### Case 3. All Submatrices and the Right-hand-side Vector are Sparse

Suppose the leading components of  $s$  in (A.5) of the forward substitution step are zero, then (A.5) can be partitioned as follow:

$$\begin{bmatrix} L_{aa} & 0_{ab} \\ L_{ba} & L_{bb} \end{bmatrix} \begin{bmatrix} w_a \\ w_b \end{bmatrix} = \begin{bmatrix} 0_a \\ s_b \end{bmatrix} \quad (A.45)$$

Observe that  $w_a = 0_a$  and we only need to solve

$$L_{bb} w_b = s_b \quad (A.46)$$

Again, to simplify our derivation, let us introduce the following notations:

- (i)  $f_o^k$  denotes the number of leading zeroes in  $s_k$ .
- (ii)  $f_p^k$  (resp.;  $g_p^k$ ) denotes the number of leading zeroes in column (resp.; row)  $p$  of  $M_{k,m+1}$  (resp.;  $M_{m+1,k}$ )

Using these notations, we can now derive  $C_{LU}$ ,  $C_{BLU}$  and  $C_{TA}$  for case 3 as follows:

$$\begin{aligned} C_{LU} &= \left\{ \sum_k \sum_i \left( \hat{c}_i^k + \hat{d}_i^{k+1} \right) \left( \hat{r}_i^k + \hat{e}_i^k \right) + \frac{q^3}{3} - \frac{q}{3} \right\} \text{(Step 1)} \\ &+ \left\{ \sum_k \sum_{i=f_o^k+1}^n \left( \hat{c}_i^k + 1 \right) + \frac{q(q+1)}{2} + \sum_k \sum_i \hat{d}_i^k \right\} \text{(Step 2)} \\ &+ \left\{ \sum_k \left( \sum_i \hat{r}_i^k + \sum_i \hat{e}_i^k \right) + \frac{q(q-1)}{2} \right\} \text{(Step 3)} \\ &= \sum_k \sum_i \hat{c}_i^k \hat{r}_i^k + \sum_k \sum_i \hat{c}_i^k \hat{e}_i^k + \sum_k \sum_i \hat{d}_i^k \hat{r}_i^k + \sum_k \sum_i \hat{d}_i^k \hat{e}_i^k \\ &+ 2 \sum_k \sum_i \hat{r}_i^k + 2 \sum_k \sum_i \hat{e}_i^k + \sum_k \sum_{i=f_o^k+1}^n \hat{c}_i^k + \sum_k \sum_i \hat{d}_i^k \\ &+ \left[ \frac{q^3}{3} + q^2 - \frac{q}{3} + \sum_k \left( n - f_o^k \right) \right] \quad (A.47) \end{aligned}$$

$$\begin{aligned} C_{BLU} &= \left\{ \sum_k \sum_i \left( \hat{c}_i^k + 1 \right) \hat{r}_i^k \right\} \text{(Step 1)} + \left\{ \sum_k \sum_p \sum_{i=g_p^k+1}^n \hat{r}_i^k \right\} \text{(Step 2)} \\ &+ \left\{ \sum_k \sum_p \sum_{i=f_p^k+1}^n \left( \hat{c}_i^k + 1 \right) \right\} \text{(Step 3)} + \left\{ \sum_k \sum_i \hat{d}_i^k \hat{e}_i^k \right. \\ &\quad \left. + \frac{q^3}{3} - \frac{q}{3} \right\} \text{(Step 4)} + \left\{ \sum_k \sum_{i=f_o^k+1}^n \left( \hat{c}_i^k + 1 \right) + \frac{q(q+1)}{2} \right\} \end{aligned}$$

$$\begin{aligned}
& + \sum_k \sum_i \hat{d}_i^k \} \text{ (Step 5)} + \left\{ \frac{q(q-1)}{2} + \sum_k \left( \sum_i \hat{r}_i^k + \sum_i \hat{e}_i^k \right) \right\} \text{ (Step 6)} \\
= & \sum_k \sum_i \hat{c}_i^k \hat{r}_i^k + \sum_k \sum_p \sum_{i=f_p^k+1}^n \hat{c}_i^k + \sum_k \sum_p \sum_{i=g_p^k+1}^n \hat{r}_i^k \\
& + \sum_k \sum_i \hat{d}_i^k \hat{e}_i^k + 2 \sum_k \sum_i \hat{r}_i^k + \left[ \sum_k \sum_i \hat{e}_i^k + \sum_k \sum_p (n-f_p^k) \right] \\
& + \sum_k \sum_{i=f_o^k+1}^n \hat{c}_i^k + \sum_k \sum_i \hat{d}_i^k + \left[ \frac{q^3}{3} + q^2 - \frac{q}{3} + \sum_k (n-f_o^k) \right] \quad (A.48)
\end{aligned}$$

$$\begin{aligned}
C_{TA} = & \left\{ \sum_k \sum_i (\hat{c}_i^{k+1}) \hat{r}_i^k \right\} \text{ (Step 1)} + \left\{ \sum_{p=0}^m \sum_k \left[ \sum_{i=f_p^k+1}^n (\hat{c}_i^{k+1}) + \sum_i \hat{r}_i^k + \sum_i \hat{d}_i^k \right] \right\} \\
& \text{(Step 2)} + \left\{ \frac{q^3}{3} + q^2 - \frac{q}{3} \right\} \text{ (Step 3)} + \left\{ mqn \right\} \text{ (Step 4)} \\
= & \sum_k \sum_i \hat{c}_i^k \hat{r}_i^k + \sum_k \sum_p \sum_{i=f_p^k+1}^n \hat{c}_i^k + \sum_k \sum_i q \hat{r}_i^k + \sum_k \sum_i q \hat{d}_i^k \\
& + 2 \sum_k \sum_i \hat{r}_i^k + \left[ mqn + \sum_k \sum_p (n-f_p^k) \right] + \sum_k \sum_{i=f_o^k+1}^n \hat{c}_i^k \\
& + \sum_k \sum_i \hat{d}_i^k + \left[ \frac{q^3}{3} + q^2 - \frac{q}{3} + \sum_k (n-f_o^k) \right] \quad (A.49)
\end{aligned}$$

To compare  $C_{LU}$  and  $C_{BLU}$ , let us first subtract (A.48) from (A.47) to obtain

$$\begin{aligned}
C_{LU} - C_{BLU} = & \sum_k \left\{ \left( \sum_i \hat{c}_i^k \hat{e}_i^k - \sum_p \sum_{i=f_p^k+1}^n \hat{c}_i^k \right) + \left( \sum_i \hat{d}_i^k \hat{r}_i^k - \sum_p \sum_{i=g_p^k+1}^n \hat{r}_i^k \right) \right. \\
& \left. + \left[ \sum_i \hat{e}_i^k - \sum_p (n-f_p^k) \right] \right\} \quad (A.50)
\end{aligned}$$

Now that we have separated (A.50) into three terms, we can examine each term one at a time: Consider the first term  $\left( \sum_i \hat{c}_i^k \hat{e}_i^k - \sum_p \sum_{i=f_p^k+1}^n \hat{c}_i^k \right)$ .

Let us focus our attention on column  $\hat{p}$  and note that the first nonzero term is in row  $(f_{\hat{p}}^k + 1)$ . The worst case of fill-ins occurs when all elements in column  $\hat{p}$  from row  $(f_{\hat{p}}^k + 2)$  to the last row become nonzero. Thus its contribution to  $\sum_i \hat{c}_i^k \hat{e}_i^k$ , in the worst case, is  $\sum_{i=f_{\hat{p}}^k+1}^n \hat{c}_i^k$ . However, its

contribution to  $\sum_p \sum_{i=f_p^k+1}^n \hat{c}_i^k$  is always  $\sum_{i=f_p^k+1}^n \hat{c}_i^k$ . Observe that only

the first nonzero term in each column contributes to the first term.

Hence we can conclude that the first term is always  $\leq 0$ . By dual

arguments the second term  $\left( \sum_i \hat{d}_i^k \hat{r}_i^k - \sum_p \sum_{i=g_p^k+1}^n \hat{r}_i^k \right)$  is always  $\leq 0$ .

Finally, the third term  $\left[ \sum_i \hat{e}_i^k - \sum_p \left( n - f_p^k \right) \right]$  can be rewritten as  $\left( \sum_i \hat{e}_i^k - \sum_p \sum_{i=f_p^k+1}^n 1 \right)$  which is a special case of the first term with

$\hat{c}_i^k = 1$  for all  $i$ . Therefore, the third term is always  $\leq 0$ . Combining these observations with (A.50), we obtain

$$C_{LU} \leq C_{BLU} \quad (A.51)$$

To compare  $C_{BLU}$  and  $C_{TA}$ , let us subtract (A.49) from (A.48) to obtain

$$\begin{aligned} C_{BLU} - C_{TA} &= \sum_k \left( \sum_p \sum_{i=g_p^k+1}^n \hat{r}_i^k - \sum_i q \hat{r}_i^k \right) + \sum_k \sum_i \left( \hat{d}_i^k \hat{e}_i^k \right. \\ &\quad \left. - q \hat{d}_i^k \right) + \left( \sum_k \sum_i \hat{e}_i^k - mqn \right) + \sum_k \sum_i \left( \hat{d}_i^k - d_i^k \right) \end{aligned} \quad (A.52)$$

Let us again examine (A.52) term by term. The first term  $\left( \sum_p \sum_{i=g_p^k+1}^n \hat{r}_i^k - \sum_i q \hat{r}_i^k \right)$

$- \sum_i q \hat{r}_i^k$  can be rewritten as  $\sum_p \left( \sum_{i=g_p^k+1}^n \hat{r}_i^k - \sum_{i=1}^{g_p^k} \hat{r}_i^k \right) = - \sum_p \sum_{i=1}^{g_p^k} \hat{r}_i^k \leq 0$ .

The third term  $\left( \sum_k \sum_i \hat{e}_i^k - mqn \right)$  is always  $\leq 0$ . Furthermore, if we assume  $\hat{d}_i^k \approx d_i^k$ , then the second and the fourth term are again  $\leq 0$ . Thus we have

$$C_{BLU} \leq C_{TA} \quad (A.53)$$

We can now summarize our comparisons of  $C_{LU}$ ,  $C_{BLU}$  and  $C_{TA}$  for the

bordered block diagonal case as follows:

- (i) if the submatrices are full, then  $C_{LU} = C_{BLU} = C_{TA}$ ;
- (ii) if the submatrices are sparse and if we assume  $\hat{d}_i^k \approx d_i^k$ , then  
 $C_{LU} \leq C_{BLU} \leq C_{TA}$ ;
- (iii) if, in addition, the right-hand-side vector is also sparse,  
we still have  $C_{LU} \leq C_{BLU} \leq C_{TA}$ .

## A.2. Comparison of the LU, Block LU and TA methods for Solving Matrices in Bordered Block Triangular Form

Let us first present the three methods for solving matrices in BBTF:

### A.2.1. The Three Methods for BBTF

#### (a) The LU method

Step 1. Factorize  $M$  as in (A.4).

Step 2. Forward substitute for  $w$  from (A.5).

Step 3. Backward substitute for  $x$  from (A.6).

Observe that the lower-triangular matrix  $L$  in (A.4) now assumes the following block structure:

$$\begin{bmatrix} L_{11} & & & & \\ L_{21} & L_{22} & & & \\ \vdots & \vdots & \ddots & & \\ L_{m1} & L_{m2} & \cdots & L_{mm} & \\ \hline & & & & \\ L_{m+1,1} & L_{m+1,2} & \cdots & L_{m+1,m} & L_{m+1,m+1} \end{bmatrix} \quad (A.54)$$

The unit-upper-triangular matrix  $U$  in (A.4) still has the same block structure as in (A.8).

(b) The Block LU method

Step 1. For  $k = 1, 2, \dots, m$ , factorize  $\underline{M}_{kk}$  as in (A.18).

Step 2. For  $m+1 \geq k_1 > k_2 \geq 1$ , backward substitute for  $\underline{L}_{k_1 k_2}$  from

$$\underline{L}_{k_1 k_2} \underline{U}_{k_2 k_2} = \underline{M}_{k_1 k_2} \quad (A.55)$$

Step 3. For  $k = 1, 2, \dots, m$ , forward substitute for  $\underline{U}_{k, m+1}$  from

$$\underline{L}_{kk} \underline{U}_{k, m+1} = \underline{M}_{k, m+1} - \sum_{k_2=1}^{k-1} \underline{L}_{kk_2} \underline{U}_{k_2, m+1} \quad (A.56)$$

Step 4. Factorize  $\underline{M}_{m+1, m+1}^{m+1}$  as (A.12) where  $\underline{M}_{m+1, m+1}^{m+1}$  is shown in (A.21).

Step 5. Forward substitute for  $\underline{w}$  from (A.5).

Step 6. Backward substitute for  $\underline{x}$  from (A.6).

(c) The TA method

Our experience shows that the best decomposition for  $\underline{M}$  is obtained by choosing

$$\underline{P} = \left[ \begin{array}{cccc|c} \underline{M}_{11} & & & & 0 \\ \underline{M}_{21} & \underline{M}_{22} & & & 0 \\ \vdots & \vdots & \ddots & & \\ \underline{M}_{m1} & \underline{M}_{m2} & \cdots & \underline{M}_{mm} & \\ \hline & & & & \\ \underline{M}_{m+1,1} & \underline{M}_{m+1,2} & \cdots & \underline{M}_{m+1,m} & \underline{1}_{m+1, m+1} \end{array} \right] \quad (A.57)$$

Hence, the modification matrices  $\underline{H}$  and  $\underline{K}$  are the same as in (A.29) and (A.30), respectively. Observe that the TA method for this case is of the same form as that of the bordered block diagonal case.

### A.2.2. Computational Complexities of the Three Methods for BBTF

For the case where all submatrices are full, we obtain the following expressions

$$\begin{aligned}
 C_{LU} &= \left\{ \sum_k \sum_i \left[ n(m-k+1) + q - i + 1 \right] (n-i+q) + \frac{q^3}{3} - \frac{q}{3} \right\} \text{(Step 1)} \\
 &+ \left\{ \sum_k \sum_i \left[ n(m-k+1) + q - i \right] + \frac{q(q-1)}{2} + nm+q \right\} \text{(Step 2)} \\
 &+ \left\{ \frac{q(q-1)}{2} + \sum_k \sum_i (n-i+q) \right\} \text{(Step 3)} \\
 &= \left( \frac{m^2}{4} + \frac{m}{12} \right) n^3 + \left( \frac{m^2 q}{2} + \frac{mq}{2} + \frac{m^2}{4} + \frac{3m}{4} \right) n^2 + \left( mq^2 + 2mq - \frac{m}{3} \right) n \\
 &+ \left( \frac{q^3}{3} + q^2 - \frac{q}{3} \right) \tag{A.58}
 \end{aligned}$$

$$\begin{aligned}
 C_{BLU} &= \left\{ m \left( \frac{n^3}{3} - \frac{n}{3} \right) \right\} \text{(Step 1)} + \left\{ \frac{m(m-1)}{2} n \frac{n(n-1)}{2} + mq \frac{n(n-1)}{2} \right\} \text{(Step 2)} \\
 &+ \left\{ mq \frac{n(n+1)}{2} + \frac{m(m-1)}{2} n^2 q \right\} \text{(Step 3)} + \left\{ \frac{q^3}{3} - \frac{q}{3} + mnq^2 \right\} \text{(Step 4)} \\
 &+ \left\{ m \frac{n(n+1)}{2} + \frac{m(m-1)}{2} n^2 + \frac{q(q+1)}{2} + mqn \right\} \text{(Step 5)} \\
 &+ \left\{ \frac{q(q-1)}{2} + m \frac{n(n-1)}{2} + mqn \right\} \text{(Step 6)} \\
 &= \left( \frac{m^2}{4} + \frac{m}{12} \right) n^3 + \left( \frac{m^2 q}{2} + \frac{mq}{2} + \frac{m^2}{4} + \frac{3m}{4} \right) n^2 + \left( mq^2 + 2mq - \frac{m}{3} \right) n \\
 &+ \left( \frac{q^3}{3} + q^2 - \frac{q}{3} \right) \tag{A.59}
 \end{aligned}$$

$$\begin{aligned}
 C_{TA} &= \left\{ m \left( \frac{n^3}{3} - \frac{n}{3} \right) \right\} \text{(Step 1)} + \left\{ (q+1) \left[ mn^2 + \frac{m(m-1)}{2} n^2 + mqn \right] \right\} \text{(Step 2)} \\
 &+ \left\{ \frac{q^3}{3} + q^2 - \frac{q}{3} \right\} \text{(Step 3)} + \left\{ mqn \right\} \text{(Step 4)} \\
 &= \frac{m}{3} n^3 + \left( \frac{qm^2}{2} + \frac{m^2}{2} + \frac{qm}{2} + \frac{m}{2} \right) n^2 + \left( mq^2 + 2mq - \frac{m}{3} \right) n + \left( \frac{q^3}{3} + q^2 - \frac{q}{3} \right) \tag{A.60}
 \end{aligned}$$

Comparing (A.58) - (A.60), we find

(i) for  $k = 1$ ,

$$C_{LU} = C_{BLU} = C_{TA} \quad (A.61)$$

(ii) for  $k > 1$ ,

$$C_{LU} = C_{BLU} \gg C_{TA} \quad (A.62)$$

Let us now consider the case where the submatrices and/or the right-hand-side vectors are sparse. Observe first that the main advantage of the TA method is that there is no need to factorize  $M_{k_1 k_2}$  for  $m+1 \geq k_1 > k_2 \geq 1$ . This feature is particularly dominant for the present case when the submatrices and/or the right-hand-side vector are sparse. Moreover, in contrast with the other two methods, no fill-in in  $M_{k_1 k_2}$  would occur in the TA method. Hence, for sparse matrix equations, we conclude that

$$C_{TA} \ll C_{LU} \leq C_{BLU} \quad (A.63)$$

## Appendix B. NP-Complete Problems Associated with Network Tearing

Let NP [11] be the class of decision problems solvable by Nondeterministic algorithms operating in Polynomial time. Roughly speaking, a nondeterministic algorithm is one which is capable of making an arbitrary choice between two alternate routes in which to branch. A nondeterministic algorithm is said to operate in polynomial time if there is a polynomial  $P(\cdot)$  such that, for any input  $x$ , the length of computation corresponding to each decision sequence is bounded by  $P(|x|)$ .<sup>26</sup>

Given two optimization problems  $\Phi_1$  and  $\Phi_2$ , we say  $\Phi_1$  is transformable into  $\Phi_2$ , denoted by  $\Phi_1 \leq \Phi_2$ , if there exists a function  $f$  such that<sup>27</sup>

- (i)  $f$  transforms the input  $x_1$  of  $\Phi_1$  into the input  $x_2$  of  $\Phi_2$ ; (B.1)
- (ii)  $|x_2| = P'(|x_1|)$  where  $P'(\cdot)$  is some polynomial function; (B.2)
- (iii) the transformation preserves the answer, i.e., if  $y_1$  is the output of  $\Phi_1$  with input  $x_1$ , then  $y_1$  will also be the output of  $\Phi_2$  with input  $x_2$ . (B.3)

This transformation is illustrated symbolically in Fig. B.1.

A problem  $\Phi$  is said to be NP-complete if

$$(i) \quad \Phi \in \text{NP}; \quad (\text{B.4})$$

$$(ii) \quad \text{satisfiability problem } \leq \Phi \quad (\text{B.5})$$

where the satisfiability problem is defined in [10,11]. An equivalent definition for NP-completeness is the following:

<sup>26</sup>Here  $|x|$  may take on different meanings depending on the nature of  $x$ . In particular, if  $x$  is an integer, then  $|x| \triangleq x$ . If  $x$  is a set of numbers, then  $|x| \triangleq$  the cardinality of the set. Finally, if  $x$  is a graph, then  $|x| \triangleq \max \{\text{number of nodes, number of branches}\}$ .

<sup>27</sup>A more general definition is available in [11].

Given that problem  $\Phi_1$  is NP-complete, then problem  $\Phi_2$  is NP-complete if

$$(i) \quad \Phi_2 \in \text{NP} \quad (\text{B.6})$$

$$(ii) \quad \Phi_1 \leq \Phi_2 \quad (\text{B.7})$$

The equivalence between these two definitions follows from the transitivity relation of " $\leq$ "; namely,

$$\text{satisfiability problem } \leq \Phi_1 \leq \Phi_2 \quad (\text{B.8})$$

We can interpret (B.7) as follows: If there exists a polynomial-bounded solution for any problem in the NP-complete class, then there exist polynomial-bounded solutions for all problems in this class.

This interpretation is extremely important because once we have established a problem belonging to the NP-complete class, we should avoid trying to find a polynomial-bounded (global) solution. Instead, some efficient heuristic approach should be undertaken

The preceding equivalent definition is also very useful in allowing one to establish the NP-completeness of a given problem by relating it to any convenient problem whose NP-completeness has been previously established. Therefore, in order to prove problem  $\Phi_2 \in \text{NP-complete}$ , it is essential to find a particular problem  $\Phi_1 \in \text{NP-complete}$  such that a transformation  $f$  satisfying (B.1)-(B.3) can be exhibited.

We are now ready to derive the main results in this section. A technical detail which needs to be brought into attention is that the output of all decision problems are assumed to be either "Yes" or "No" [11]. Hence, we shall avoid listing the output statement in the following development.

Theorem B.1. The cluster problem (i.e., GOP1) is NP-complete.

Proof: We can reformulate the cluster problem, henceforth denoted by

$\Phi_2$ , by specifying the input and property statements as follows:

$\Phi_2$ : Cluster problem (GOP1)

Input: An undirected graph  $G_Y$  and positive integers  $q_2$  and  $n_{\max}$  (B.9)

Property:  $G_Y$  has  $q_2$  nodes such that their removal will leave the remaining graph disconnected whereas each component (i.e., each disconnected subgraph) contains no more than  $n_{\max}$  nodes. (B.10)

It is easy to show  $\Phi_2 \in \text{NP}$  with the help of the branching tree shown in Fig. B.2. Since for each decision sequence of  $N_{Y_2}$ , the computation for finding the disconnected components of the section graph  $G(N_{Y_1})$  is linear [39] and therefore polynomial-bounded, it follows that  $\Phi_2 \in \text{NP}$ .

Now, in selecting the problem  $\Phi_1$ , let us choose the following node-cover problem which is known to be NP-complete [10]:

$\Phi_1$ : Node-cover problem

Input: An undirected graph  $G$  and a positive integer  $q_1$ . (B.11)

Property:  $G$  has a subset of nodes  $R$  such that  $|R| \leq q_1$  and every branch of  $G$  is incident with some node in  $R$ . (B.12)

The reason for choosing this node-cover problem as  $\Phi_1$  becomes apparent if we examine the following analogous matrix problem:

Given a symmetric matrix  $M$ , can we find a symmetric permutation of rows and columns such that  $M$  has a bordered-diagonal nonzero structure with a border size no larger than  $q_1$ ?

This structure is illustrated in Fig. B.3. Comparing Fig. B.3 with the structure associated with the matrix analogy of the cluster problem in Fig. B.4, we see that the node-cover problem is in fact a special case of the cluster problem with  $n_{\max} = 1$ .

To complete the proof, we will construct the transformation  $f$  via an example (Fig. B.5(a)). For each node  $n_i$  in  $\mathcal{P}_1$ , construct nodes  $n_{i_1}, n_{i_2}, \dots, n_{i_{n_{\max}}}$  and undirected branches  $b_{i_1 i_2}$  (i.e., a branch between nodes  $n_{i_1}$  and  $n_{i_2}$ ),  $b_{i_2 i_3}, \dots, b_{i_{(n_{\max}-1)} i_{n_{\max}}}$  and  $b_{i_{n_{\max}} i_1}$  for  $\mathcal{P}_2$ . The construction corresponding to  $n_{\max} = 3$  is shown in Fig. B.5(b). For each branch  $b_{ij}$  in  $\mathcal{P}_1$ , construct branches  $b_{i_k j_l}$  for  $k, l = 1, 2, \dots, n_{\max}$ . This procedure yields the input graph for  $\mathcal{P}_2$  having the input integers  $n_{\max}$  and  $q_2 \triangleq q_1 \cdot n_{\max}$ .

Observe that the preceding construction transforms the original graph with "n" nodes and "b" branches into a graph with  $(n \cdot n_{\max})$  nodes and  $(n \cdot n_{\max} + b \cdot n_{\max}^2)$  branches. Thus the construction is polynomial-bounded. Observe that each set of nodes  $\mathcal{N}_i \triangleq \{n_{i_1}, n_{i_2}, \dots, n_{i_{n_{\max}}}\}$  forms a connected component and that the removal of up to  $(n_{\max}-1)$  nodes from the set  $\mathcal{N}_i$  does not change the connectivity of the constructed graph. Therefore, in order to change the connectivity, we have to remove nodes in  $\mathcal{N}_i$  as a single unit. Since  $q_2 = q_1 n_{\max}$ , we are actually removing  $q_1$  sets of  $\mathcal{N}_i$  nodes. Hence, the solution of the associated  $\mathcal{P}_2$  problem is the solution of the original  $\mathcal{P}_1$  problem. Finally, observe that if  $\mathcal{P}_1$  has a solution with less than  $q_1$  nodes, it certainly has a solution with  $q_1$  nodes. This establishes that a solution of the original  $\mathcal{P}_1$  problem is the solution of the associated problem.  $\square$

Theorem B.2. The generalized feedback node problem (i.e., GOP2) is NP-complete.

Proof: We can reformulate the generalized feedback node problem, henceforth denoted as  $\mathcal{P}_2$ , as follows:

$\mathcal{P}_2$ : Generalized feedback node problem (GOP2)

Input: A directed graph  $\hat{G}_Y$  and positive integers  $q_2$  and  $n_{\max}$ . (B.13)

Property:  $\hat{G}_Y$  has  $q_2$  nodes such that their removal will leave the remaining graph containing several strongly-connected components each containing  $\leq n_{\max}$  nodes. (B.14)

To prove that  $\mathcal{P}_2 \in \text{NP}$ , we can use the same branching tree shown in Fig. B2. In this case, for each decision sequence of  $\mathcal{N}_{Y_2}$ , the computation for finding the strongly-connected components of the section graph  $\hat{G}(\mathcal{N}_{Y_1})$  is also linear [39].

Now, in selecting the problem  $\mathcal{P}_1$ , let us choose the following feedback node problem which is known to be NP-complete [11]:

$\mathcal{P}_1$ : Feedback node problem

Input: A directed graph  $\hat{G}$  and a positive integer  $q_1$ . (B.15)

Property:  $\hat{G}$  has a set of  $q_1$  nodes whose removal breaks all directed cycles. (B.16)

The reason for choosing this feedback node problem as  $\mathcal{P}_1$  becomes apparent if we examine the following analogous matrix problem: Given an asymmetric matrix  $M$ , can we find a symmetric permutation of rows and columns such that  $M$  has a bordered-lower-triangular nonzero structure with a border size  $q_1$ ?

This structure is illustrated in Fig. B6. Comparing Fig. B6 with the structure associated with the matrix analogy of the generalized feedback node problem in Fig. B7, we see that the feedback node problem is in fact a special case of the generalized feedback node problem with  $n_{\max} = 1$ .

To complete the proof, we will construct the transformation  $f$  via an example (Fig. B8(a)). For each node  $n_i$  in  $\mathcal{P}_1$ , construct nodes  $n_{i_1}, n_{i_2}, \dots, n_{i_{n_{\max}}}$  and directed branches  $\hat{b}_{i_1 i_2}$  (i.e., a branch from node  $n_{i_1}$  to node  $n_{i_2}$ ),  $\hat{b}_{i_2 i_3}, \dots, \hat{b}_{i_{(n_{\max}-1)} i_{n_{\max}}}$ , and  $\hat{b}_{i_{n_{\max}} i_1}$  for  $\mathcal{P}_2$ . The construction corresponding to  $n_{\max} = 2$  is shown in Fig. B8(b). For each branch  $\hat{b}_{ij}$  in  $\mathcal{P}_1$ , construct branches  $\hat{b}_{i_k j_l}$  for  $k, l = 1, 2, \dots, n_{\max}$ . This procedure yields the input graph for  $\mathcal{P}_2$  having the input integers  $n_{\max}$  and  $q_2 \triangleq q_1 \cdot n_{\max}$ .

It should now be obvious that the rest of the proof is exactly the same as that given for the previous theorem.  $\square$

### Appendix C. Loop Analysis and Cutset Analysis in Tearing Form

Let  $\mathcal{G}$  be the graph of a linear resistive network  $N$ , let  $\mathcal{T}$  denote some tree of  $\mathcal{G}$  and let  $\mathcal{L}$  denote its associated cotree. Partition the tree  $\mathcal{T}$  into two arbitrary subsets  $\mathcal{T}_1$  and  $\mathcal{T}_2$  such that  $\mathcal{T} = \mathcal{T}_1 \cup \mathcal{T}_2$ . Let  $\mathcal{L}_1$  denote any subset of the cotree  $\mathcal{L}$  which forms fundamental loops exclusively with branches in  $\mathcal{T}_1$  and let  $\mathcal{L}_2$  denote the remaining branches so that  $\mathcal{L} = \mathcal{L}_1 \cup \mathcal{L}_2$ . With respect to the above partitioning of branches, the fundamental loop matrix  $\underline{B}$  and the fundamental cutset matrix  $\underline{Q}$  are given respectively by

$$\begin{array}{cccc} \mathcal{L}_1 & \mathcal{T}_1 & \mathcal{L}_2 & \mathcal{T}_2 \\ \hline \underline{B} = & \left[ \begin{array}{cccc} 1 & \frac{B}{Z_1} J_1 & 0 & \frac{0}{Z_1} J_2 \\ -\frac{B}{Z_1} J_1 & 1 & -\frac{B}{Z_1} J_2 & 0 \\ 0 & \frac{B}{Z_2} J_1 & 1 & \frac{B}{Z_2} J_2 \\ -\frac{B}{Z_2} J_1 & -\frac{B}{Z_2} J_2 & 1 & 0 \end{array} \right] & & (C.1) \\ Q = & \left[ \begin{array}{cccc} -B^T & \frac{1}{Z_1} J_1 & -B^T & 0 \\ \frac{1}{Z_1} J_1 & 1 & \frac{1}{Z_2} J_2 & 0 \\ 0^T & 0 & -B^T & \frac{1}{Z_2} J_2 \\ -\frac{1}{Z_1} J_2 & 0 & -\frac{1}{Z_2} J_2 & 1 \end{array} \right] & & (C.2) \end{array}$$

Assuming the composite branch format as shown in Fig. 1, the loop and cutset equations are given respectively by:

$$\underline{B} \underline{R} \underline{B}^T \underline{I}_{\mathcal{L}} = \underline{B} (\underline{e} - \underline{R} \underline{j}) \quad (C.3)$$

$$\underline{Q} \underline{G} \underline{Q}^T \underline{V}_{\mathcal{L}} = \underline{Q} (\underline{j} - \underline{G} \underline{e}) \quad (C.4)$$

where  $\underline{R}$  and  $\underline{G}$  are the branch resistance matrix and the branch conductance matrix, respectively.

Let us consider the following two special cases.

#### Case 1. Loop Analysis in Tearing Form

Upon open-circuiting all branches in  $\mathcal{L}_2 \cup \mathcal{T}_2$ , let us assume that

the remaining branches in  $\mathcal{L}_1 \cup \mathcal{T}_1$  form "m" separable components

$\mathcal{G}_1^1, \mathcal{G}_1^2, \dots, \mathcal{G}_1^m$ . Partition the sets  $\mathcal{L}_1$  and  $\mathcal{T}_1$  into m corresponding components; namely,  $\mathcal{L}_1^1, \mathcal{L}_1^2, \dots, \mathcal{L}_1^m$  and  $\mathcal{T}_1^1, \mathcal{T}_1^2, \dots, \mathcal{T}_1^m$ . Then the fundamental loop matrix  $B$  with respect to this new partition can be written as follows:

$$\underline{B} = \begin{bmatrix} \mathcal{L}_1^1 & \mathcal{T}_1^1 & \mathcal{L}_1^2 & \mathcal{T}_1^2 & \cdots & \mathcal{L}_1^m & \mathcal{T}_1^m & \mathcal{L}_2 & \mathcal{T}_2 \\ \begin{array}{|c|c|} \hline 1 & B \\ \hline \mathcal{L}_1^1 & \mathcal{T}_1^1 \\ \hline \end{array} & \begin{array}{|c|c|} \hline 1 & B \\ \hline \mathcal{L}_1^2 & \mathcal{T}_1^2 \\ \hline \end{array} & \cdots & \begin{array}{|c|c|} \hline 1 & B \\ \hline \mathcal{L}_1^m & \mathcal{T}_1^m \\ \hline \end{array} & & & & \begin{array}{|c|c|} \hline 1 & B \\ \hline \mathcal{L}_2 & \mathcal{T}_2 \\ \hline \end{array} & \cdots \\ \vdots & \vdots \\ \begin{array}{|c|c|} \hline 1 & B \\ \hline \mathcal{L}_2^1 & \mathcal{T}_2^1 \\ \hline \end{array} & \begin{array}{|c|c|} \hline 1 & B \\ \hline \mathcal{L}_2^2 & \mathcal{T}_2^2 \\ \hline \end{array} & \cdots & \begin{array}{|c|c|} \hline 1 & B \\ \hline \mathcal{L}_2^m & \mathcal{T}_2^m \\ \hline \end{array} & & & & \begin{array}{|c|c|} \hline 1 & B \\ \hline \mathcal{L}_2 & \mathcal{T}_2 \\ \hline \end{array} & \cdots \end{bmatrix} \quad (C.5)$$

Observe that the nonzero block structure of  $\underline{B}$  in (C.5) is identical to the nonzero block structure of the reduced incidence matrix  $\underline{A}$  in (2). It follows from Lemma 1 that the product  $\underline{B}\underline{R}\underline{B}^t$  will assume a bordered-block-diagonal form if the analog of branch coupling condition 1 is satisfied, or a bordered-block-triangular form, if the analog of branch coupling condition 2 is satisfied.

#### Case 2. Cutset Analysis in Tearing Form

Upon short-circuiting all branches in  $\mathcal{L}_1 \cup \mathcal{T}_1$ , let us assume that the remaining branches in  $\mathcal{L}_2 \cup \mathcal{T}_2$  form "m" separable components  $\mathcal{G}_2^1, \mathcal{G}_2^2, \dots, \mathcal{G}_2^m$ . Partition the sets  $\mathcal{L}_2$  and  $\mathcal{T}_2$  into m corresponding

components; namely,  $\mathcal{L}_2^1, \mathcal{L}_2^2, \dots, \mathcal{L}_2^m$  and  $\mathcal{T}_2^1, \mathcal{T}_2^2, \dots, \mathcal{T}_2^m$ . Then the fundamental cutset matrix  $Q$  with respect to this new partition can be written as follows:

$$Q = \begin{bmatrix} \mathcal{L}_1 \mathcal{T}_1 & \mathcal{L}_2^1 \mathcal{T}_2^1 & \mathcal{L}_2^2 \mathcal{T}_2^2 & \cdots & \mathcal{L}_2^m \mathcal{T}_2^m \\ \begin{array}{|c|c|c|c|c|} \hline -B^t & \vdots & -B^t & \vdots & -B^t \\ \mathcal{L}_1^1 \mathcal{T}_1^1 & \mathcal{L}_2^1 \mathcal{T}_2^1 & \mathcal{L}_2^2 \mathcal{T}_2^2 & \cdots & \mathcal{L}_2^m \mathcal{T}_2^m \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \end{array} & \begin{array}{|c|c|c|c|c|} \hline -B^t & \vdots & & & 0 \\ \mathcal{L}_2^1 \mathcal{T}_2^1 & \mathcal{L}_2^2 \mathcal{T}_2^2 & \mathcal{L}_2^3 \mathcal{T}_2^3 & \cdots & \\ \vdots & \vdots & \vdots & \ddots & \\ \end{array} & \begin{array}{|c|c|c|c|c|} \hline -B^t & \vdots & & & \\ \mathcal{L}_2^2 \mathcal{T}_2^2 & \mathcal{L}_2^3 \mathcal{T}_2^3 & \mathcal{L}_2^4 \mathcal{T}_2^4 & \cdots & \\ \vdots & \vdots & \vdots & \ddots & \\ \end{array} & \cdots & \begin{array}{|c|c|c|c|c|} \hline -B^t & \vdots & & & \\ \mathcal{L}_2^m \mathcal{T}_2^m & \mathcal{L}_2^{m+1} \mathcal{T}_2^{m+1} & \mathcal{L}_2^{m+2} \mathcal{T}_2^{m+2} & \cdots & \\ \vdots & \vdots & \vdots & \ddots & \\ \end{array} \\ \begin{array}{|c|c|c|c|c|} \hline 0 & & & & \\ \vdots & & & & \\ 0 & & & & \\ \end{array} & \begin{array}{|c|c|c|c|c|} \hline 0 & & & & \\ \vdots & & & & \\ 0 & & & & \\ \end{array} & \begin{array}{|c|c|c|c|c|} \hline 0 & & & & \\ \vdots & & & & \\ 0 & & & & \\ \end{array} & \cdots & \begin{array}{|c|c|c|c|c|} \hline 0 & & & & \\ \vdots & & & & \\ 0 & & & & \\ \end{array} \end{array} \quad (C.6)$$

Observe that the nonzero block structure of  $Q$  in (C.6) is again similar to that of  $A$  in (2) except for a block permutation of rows and columns. It follows from Lemma 1 that the product  $QQ^t$  will likewise assume a BBDF if the analog of branch coupling condition 1 is satisfied or BBTF if the analogue of branch coupling condition 2 is satisfied.

So far, we have implicitly assumed that a tree  $\mathcal{T}$  has been given apriori. Let us now consider the more realistic case where an optimal tree has to be chosen and partitioned relative to some optimal criterion. It suffices to consider the loop analysis case since the dual result would apply to the case of cutset analysis.

Let us recall that the removal of  $\mathcal{L}_2 \cup \mathcal{T}_2$  branches would separate the remaining graph into "m" separable components  $\mathcal{G}_1^1, \mathcal{G}_1^2, \dots, \mathcal{G}_1^m$  with

$G_1^k$  containing  $\mathcal{L}_1^k \cup \mathcal{T}_1^k$  branches. Therefore, we can formulate an analogous graph optimization problem by minimizing the number of branches belonging to  $\mathcal{L}_2 \cup \mathcal{T}_2$  subject to analogous topological constraints.

After  $\mathcal{L}_2 \cup \mathcal{T}_2$  is found, we can determine easily which branches in  $G_1^k$  can be assigned to  $\mathcal{T}_1^k$  and then assign the remaining branches to  $\mathcal{L}_1^k$ . To separate  $\mathcal{L}_2$  from  $\mathcal{T}_2$ , we simply short-circuit all branches belonging to all "m" components  $G_1^1, G_1^2, \dots, G_1^m$ , and find a tree (which is to be assigned as  $\mathcal{T}_2$ ) on the reduced subgraph.

## REFERENCES

- [1] G. Kron, Diakoptics -- Piecewise Solution of Large-Scale Systems, MacDonald, London, 1963.
- [2] H. H. Happ, Diakoptics and Networks, Academic Press, New York, 1971.
- [3] F. H. Branin, "The Relation between Kron's Method and the Classical Methods of Network Analysis," The Matrix and Tensor Quarterly, Vol. 12, No. 3, March 1962, pp. 69-115.
- [4] L. O. Chua and L. K. Chen, "Diakoptic and Generalized Hybrid Analysis," IEEE Trans. on Circuits and Systems, Vol. CAS-23, No. 12, December 1976.
- [5] F. F. Wu, "Solution of Large-Scale Networks by Tearing," University of California, Berkeley, Electronics Research Laboratory, Memorandum No. ERL-M532, July 1975.
- [6] G. Kron, "A Set of Principles to Interconnect the Solution of Physical Systems," Journal of Applied Physics, Vol. 24, No. 8, August 1953, pp. 965-980.
- [7] L. O. Chua and L. K. Chen, "Nonlinear Diakoptics," University of California, Berkeley, Electronics Research Laboratory, Memorandum No. ERL-M495, March 1975.
- [8] C. W. Ho, A. E. Ruehli and P. A. Brennan, "The Modified Nodal Approach to Network Analysis," IEEE Trans. on Circuits and Systems, Vol. CAS-22, No. 6, June 1975.
- [9] L. O. Chua and P. M. Lin, Computer-Aided Analysis of Electronic Circuits: Algorithms and Computational Techniques, Prentice-Hall, Englewood Cliffs, N. J., 1975.

- [10] A. V. Aho, J. E. Hopcroft and J. D. Ullman, The Design and Analysis of Computer Algorithms, Addison-Wesley, Reading, Massachusetts, 1974.
- [11] R. M. Karp, "On the Computational Complexity of Combinatorial Problems," Networks, Vol. 5, No. 1, 1975, pp. 45-68.
- [12] C. A. Desoer and E. S. Kuh, Basic Circuit Theory, McGraw-Hill, New York, 1969.
- [13] L. K. Cheung, Sparse Matrix Techniques: Theory and Application, University of California, Berkeley, Ph.D. Thesis, August 1973.
- [14] F. Harary, Graph Theory, Addison-Wesley, Reading, Massachusetts, 1969.
- [15] F. F. Wu, "Diakoptics and Triangular Factorization," IEEE paper #C75 107-8, presented at IEEE Power Engineering Society Winter Meeting, New York, January 1975.
- [16] R. A. Willoughby, ed., Proceedings of the Symposium on Sparse Matrices and Their Applications, RA1, IBM Research Center, Yorktown Heights, New York, March 1969.
- [17] J. K. Reid, ed., Large Sparse Sets of Linear Equations, Academic Press, New York, 1971.
- [18] D. J. Rose and R. A. Willoughby, ed., Sparse Matrices and Their Applications, Plenum Press, New York, 1972.
- [19] J. R. Bunch and D. J. Rose, ed., Sparse Matrix Computation, Academic Press, New York, 1976.
- [20] D. J. Rose and J. R. Bunch, "The Role of Partitioning in the Numerical Solution of Sparse Systems," Sparse Matrices and Their Applications, D. J. Rose and R. A. Willoughby, ed., Plenum Press, New York, 1972.

- [21] G. Guardabassi and A. Sangiovanni-Vincentelli, "A Two Levels Algorithm for Tearing," IEEE Trans. on Circuits and Systems, Vol. CAS-23, No. 12, December 1976.
- [22] CAL 6400 FORTRAN Guide, University of California, Computer Center, Berkeley, 1969.
- [23] P. H. Enslow, ed., Multiprocessor and Parallel Processing, Wiley, New York, 1974.
- [24] J. L. Cochrane and M. Zeleny, ed., Multiple Criteria Decision Making, University of South Carolina Press, Columbia, South Carolina, 1973.
- [25] S. Parter, "The Use of Linear Graphs in Gaussian Elimination," SIAM Review, Vol. 3, No. 2, April 1961, pp. 119-130.
- [26] R. A. Willoughby, A Survey of Sparse Matrix Technology, IBM Research, Report RC 3872, Yorktown Heights, New York, May 1972.
- [27] S. Lin, "Heuristic Programming as an Aid to Network Design," Networks, Vol. 5, No. 1, 1975, pp. 33-43.
- [28] D. Ferrari, "Improving Locality by Critical Working Sets," Comm. of ACM, Vol. 17, No. 11, November 1974, pp. 614-620.
- [29] W. E. Donath and A. J. Hoffman, "Lower Bounds for the Partitioning of Graphs," IBM J. Res. and Dev., Vol. 17, No. 5, September 1973, pp. 420-425.
- [30] E. C. Ogbuobiri, W. F. Tinney and J. W. Walker, "Sparsity-Directed Decomposition for Gaussian Elimination on Matrices," IEEE Trans. on Power, Apparatus and Systems, Vol. PAS-89, No. 1, January 1970, pp. 141-150.
- [31] F. Luccio and M. Sami, "On the Decomposition of Networks in Minimally Interconnected Subnetworks," IEEE Trans. on Circuit Theory,

Vol. CT-16, No. 2, May 1969, pp. 184-188.

- [32] T. C. Hu, Integer Programming and Network Flow, Addison-Wesley, Reading, Massachusetts, 1969.
- [33] B. W. Kernighan and S. Lin, "An Efficient Heuristic Procedure for Partitioning Graphs," Bell System Tech. J., Vol. 49, No. 2, February 1970, pp. 291-307.
- [34] J. C. Gower, "Comparison of Some Methods of Cluster Analysis," Biometrics, Vol. 54, December 1967, pp. 623-637.
- [35] E. C. Cua, "Fundamental Loops Generation and Clusters Analysis -- Algorithm and Implementation," University of California, Berkeley, Department of Electrical Engineering and Computer Science, Master Thesis Plan II, June 1976.
- [36] L. K. Chen, B. S. Ting and A. Sangiovanni-Vincentelli, "An Edge-Oriented Adjacency List for Undirected Graphs," University of California, Berkeley, Electronics Research Laboratory, Memorandum No. ERL-M589, May 1976.
- [37] Motorola, Linear Integrated Circuits Data Book, 1972, pp. (7-440)-(7-442).
- [38] W. F. Tinney and J. W. Walker, "Direct Solutions of Sparse Network Equations by Optimally Ordered Triangular Factorization," Proc. of IEEE, Vol. 55, November 1967, pp. 1801-1809.
- [39] R. Tarjan, "Depth-First Search and Linear Graph Algorithms," SIAM J. Computing, Vol. 1, No. 2, June 1972, pp. 146-160.

Figure Captions

Fig. 1. The standard composite branch.

Fig. 2. Two useful matrix structures.

(a) BBDF.

(b) BBTF.

Fig. 3. The basic partition of nodes and branches.

Fig. 4. The topological condition.

Fig. 5. An example illustrating the topological condition.

(a) The example graph.

(b) The section graph  $\hat{G}(N_1)$  where  $N_1 = \{n_1, n_2, n_3, n_4\}$ .

Fig. 6. A block-diagonal transformation of nonzero elements which preserves the structures of  $G_{11}$  in  $\hat{Y}_{11}$ .

Fig. 7. A circuit example N for illustrating the branch coupling condition 1.

Fig. 8. The associated directed graph  $\hat{G}$  of N.

Fig. 9. A new partition of nodes and branches.

Fig. 10. The circuit-theoretic interpretation of node-tearing nodal analysis.

(a) Equivalent voltage source substitution.

(b) Substituted voltage sources summarize the outside influence due to elements in  $\hat{B}_2$  thereby decoupling the original network into three separated subnetworks.

Fig. 11. Application of (40) where all branches shown connecting the subnetworks belong to  $\hat{B}_2$ .

Fig. 12. An example showing nodes  $n_a$  and  $n_b$  attached to three  $\hat{B}_2$  branches.

**Fig. 13.** An example of the sparsity graph.

- (a) The example matrix  $\underline{Y}$ .
- (b) The sparsity graph  $\hat{G}_Y$  of  $\underline{Y}$ .
- (c) The undirected version  $G_Y$  of  $\hat{G}_Y$ .

**Fig. 14.** Basic relationship between the graph  $\hat{G}$  and its associated sparsity subgraph.

- (a) A typical branch.
- (b) The sparsity subgraph  $\hat{G}_Y$  due to  $G_{k_1 k_1}$ .
- (c) The undirected version  $G_Y$  due to  $G_{k_1 k_1}$ .
- (d) The sparsity subgraph  $\hat{G}_Y$  due to  $G_{k_1 k_2}$ .
- (e) The undirected version  $G_Y$  due to  $G_{k_1 k_2}$ .

**Fig. 15.** A contour tableau.

**Fig. 16.** A graphic interpretation of Step 7 for updating  $AS(i+1)$  from  $AS(i)$ .

**Fig. 17.** An example for illustrating the contour tableau construction algorithms.

**Fig. 18.** Three different contour tableaus associated with the graph in Fig. 17 by using three different strategies during the construction.

- (a) Arbitrary choice.
- (b) Greedy strategy in choosing the next iterating node.
- (c) Initial iterating node selection.

**Fig. 19.** The graphical interpretation of the role of  $AS(i)$  as a separating set.

**Fig. 20.** An example showing that the greedy strategy may sometimes give undesirable results.

- (a) The example graph.

(b) Cluster obtained by choosing  $n_1$  as the next iterating node.

(c) Cluster obtained by choosing  $n_2$  as the next iterating node.

Fig. 21. Flow-chart for a cluster algorithm.

Fig. 22. An illustration of the various shapes of CN vs. iteration step and some methods for grouping the nodes into clusters.

(a) Smooth curve with well-defined clusters.

(b) A cluster containing  $n_{\max}$  nodes before a local minimum is reached.

(c) A cluster containing many small wiggles.

(d) A cluster containing many small clusters.

(e) Least-local-minimum clustering strategy.

Fig. 23. A graphical illustration of the redundancy phenomenon and the dynamic contour cutting strategy to overcome it.

(a) Example with 3 clusters.

(b) Original contour tableau.

(c) Original CN curve.

(d) Contour tableau with dynamic contour cutting.

(e) CN curve with dynamic contour cutting.

Fig. 24. Flow-chart for the refined cluster algorithm.

Fig. 25. An example illustrates the cluster algorithm.

(a) Example with 3 clusters and  $n_{\max}=10$ .

(b) The resulting contour tableau.

Fig. 26. Nine more examples of the application of the cluster algorithm.

(a) Example 2 with 3 clusters and  $n_{\max}=19$ .

(b) Example 3 with 2 clusters and  $n_{\max}=12$ .

- (c) Example 4 with 4 clusters and  $n_{\max}=12$ .
- (d) Example 5 with 5 clusters and  $n_{\max}=27$ .
- (e) Example 6 with 4 clusters and  $n_{\max}=15$ .
- (f) Example 7 with 3 clusters and  $n_{\max}=20$ .
- (g) Example 8 with 3 clusters and  $n_{\max}=30$ .
- (h) Example 9 with 4 clusters and  $n_{\max}=17$ .
- (i) Example 10 with 3 clusters and  $n_{\max}=25$ .

**Fig. 27.** The "computer time spent" vs. "nb" plot illustrating the  $\mathcal{O}(nb)$  bound. The number in this plot corresponds to the example number of Table 1.

**Fig. 28.** A practical circuit cluster problem.

- (a) A frequency-shift keyer tone generator.
- (b) The operational amplifier circuit schematic.
- (c) The Ebers-Moll model for transistors.
- (d) The induced transistor sparsity subgraph.
- (e) The resulting 5 clusters. Note that due to the greedy strategy, the 1st operational amplifier is broken into 2 clusters.

**Fig. A1.** We use  $c_i^k$  (resp.;  $r_i^k$ ) to denote the number of nonzero elements in the  $i$ -th column below (resp.;  $i$ -th row to the right of) the  $i$ -th diagonal element of  $M_{kk}$ .

**Fig. B1.** Symbolic representation illustrating the transformation of graph problem  $\mathcal{P}_1$  into  $\mathcal{P}_2$ .

**Fig. B2.** A branching tree used for proving  $\mathcal{P}_2 \in \text{NP}$ .

**Fig. B3.** A matrix analogy of the node-cover problem.

**Fig. B4.** A matrix analogy of the cluster problem.

Fig. B5. An example for illustrating the construction of the transformation  $f$  with  $n_{\max}=3$ .

(a) Input graph for  $\mathcal{P}_1$ .

(b) Input graph for  $\mathcal{P}_2$ .

Fig. B6. A matrix analogy of the feedback node problem.

Fig. B7. A matrix analogy of the generalized feedback node problem.

Fig. B8. An example for illustrating the construction of the transformation  $f$  with  $n_{\max}=2$ .

(a) Input graph for  $\mathcal{P}_1$ .

(b) Input graph for  $\mathcal{P}_2$ .

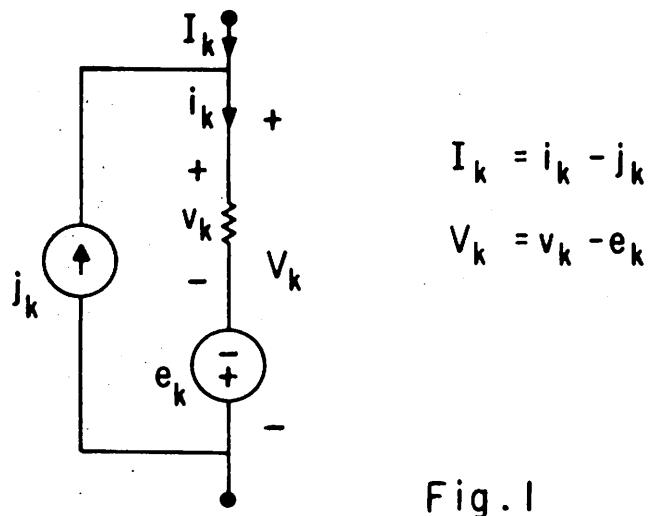


Fig. 1

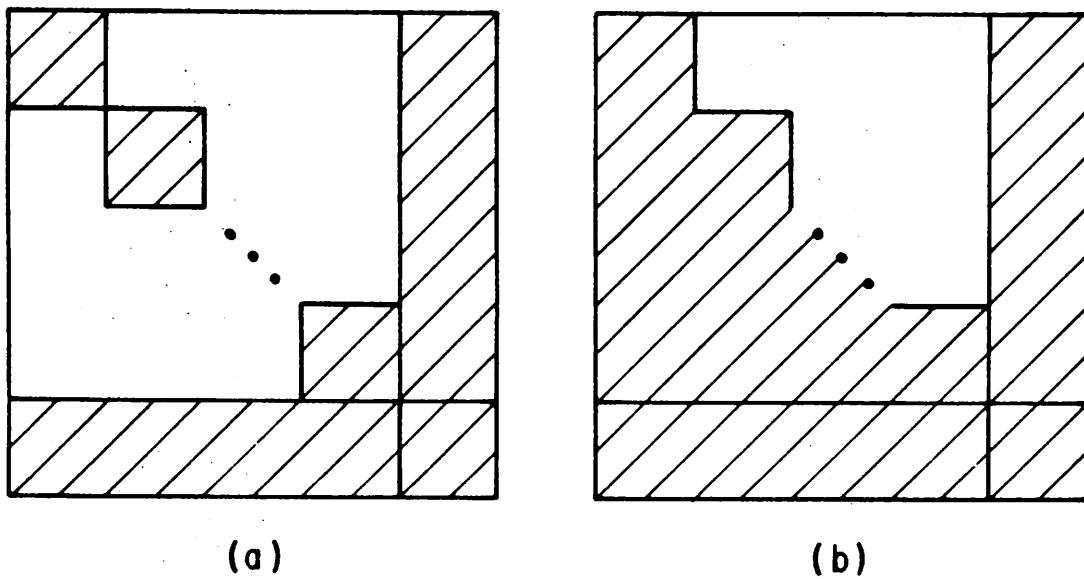


Fig. 2

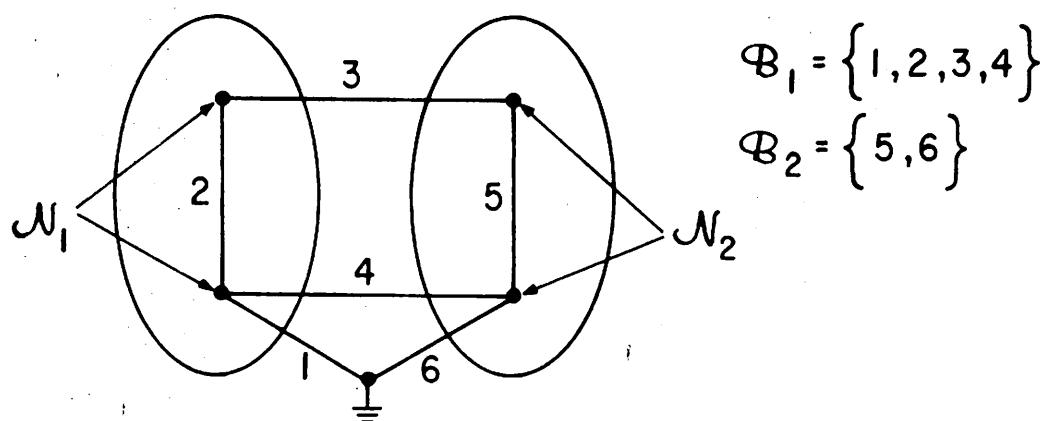


Fig. 3

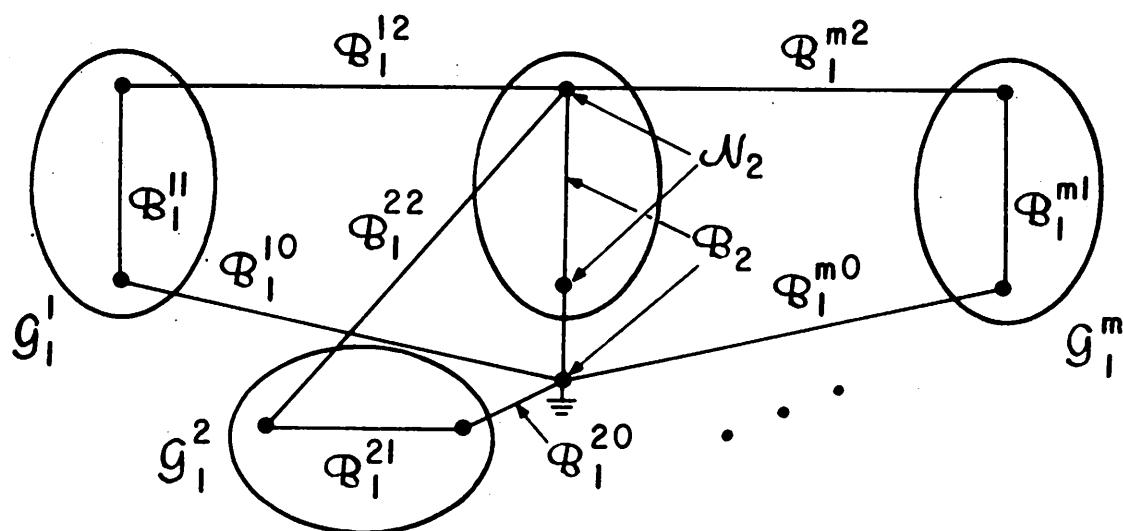
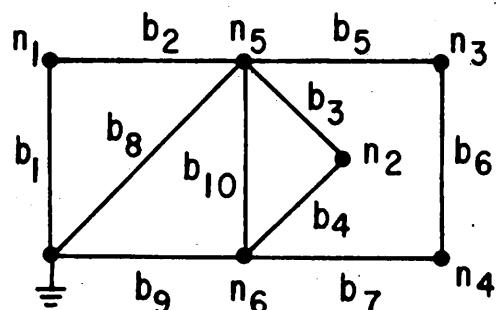
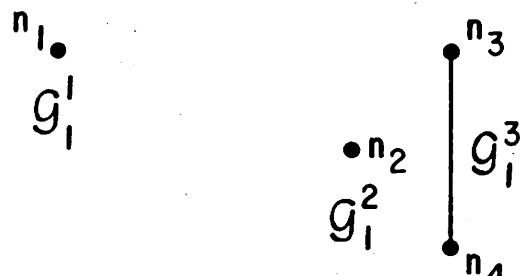


Fig. 4



(a)



(b)

Fig. 5

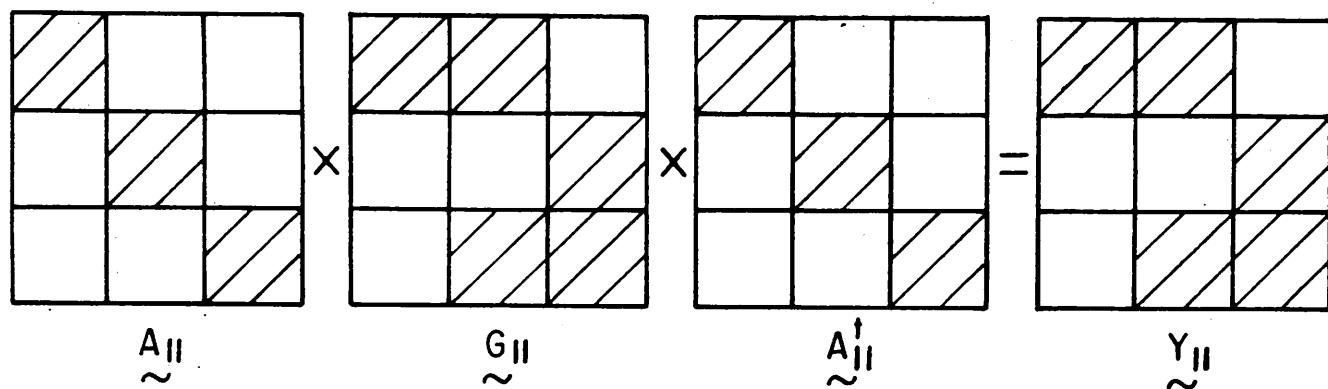


Fig. 6

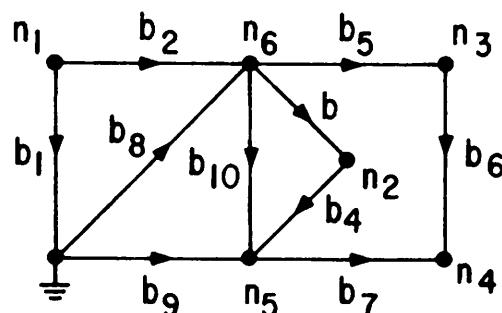
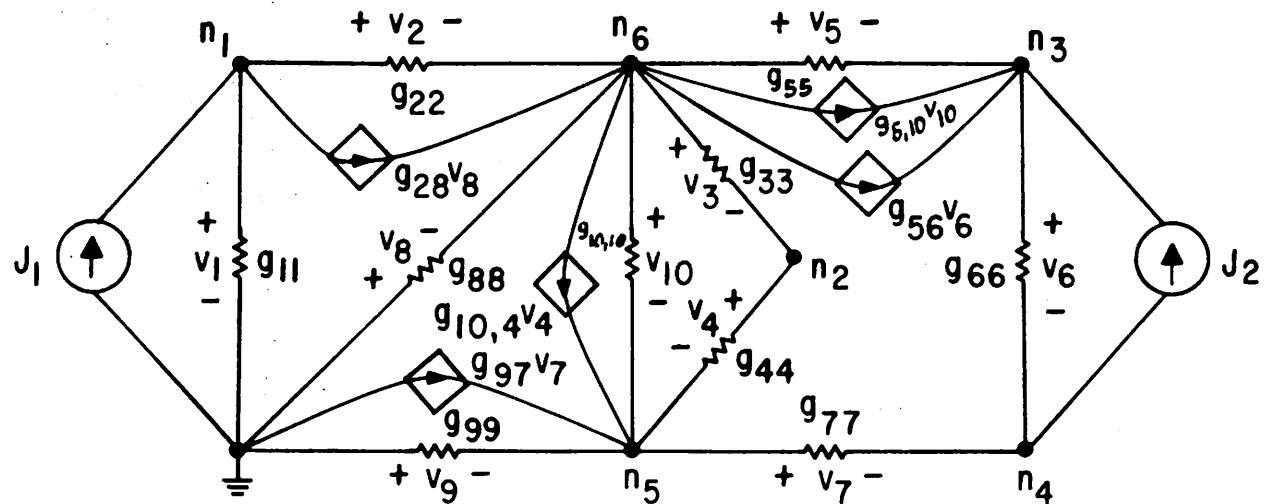


Fig. 8

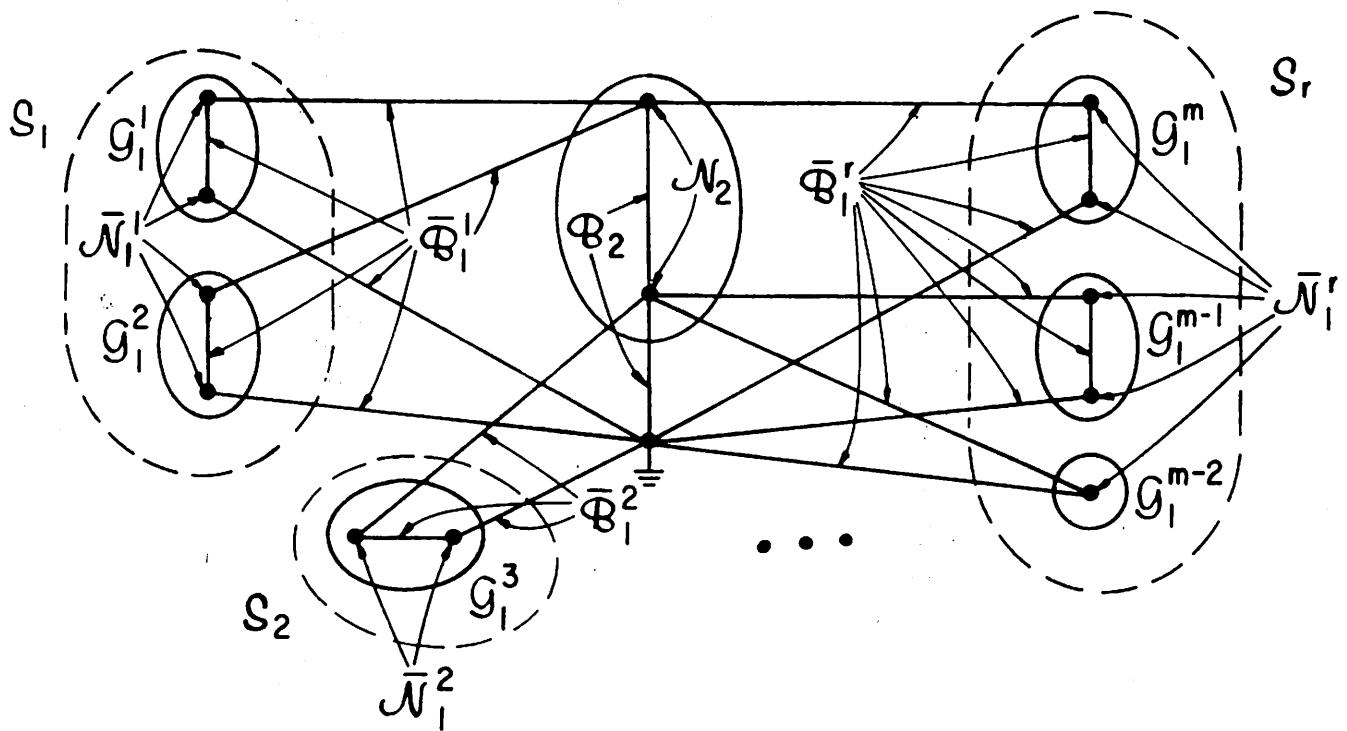


Fig. 9

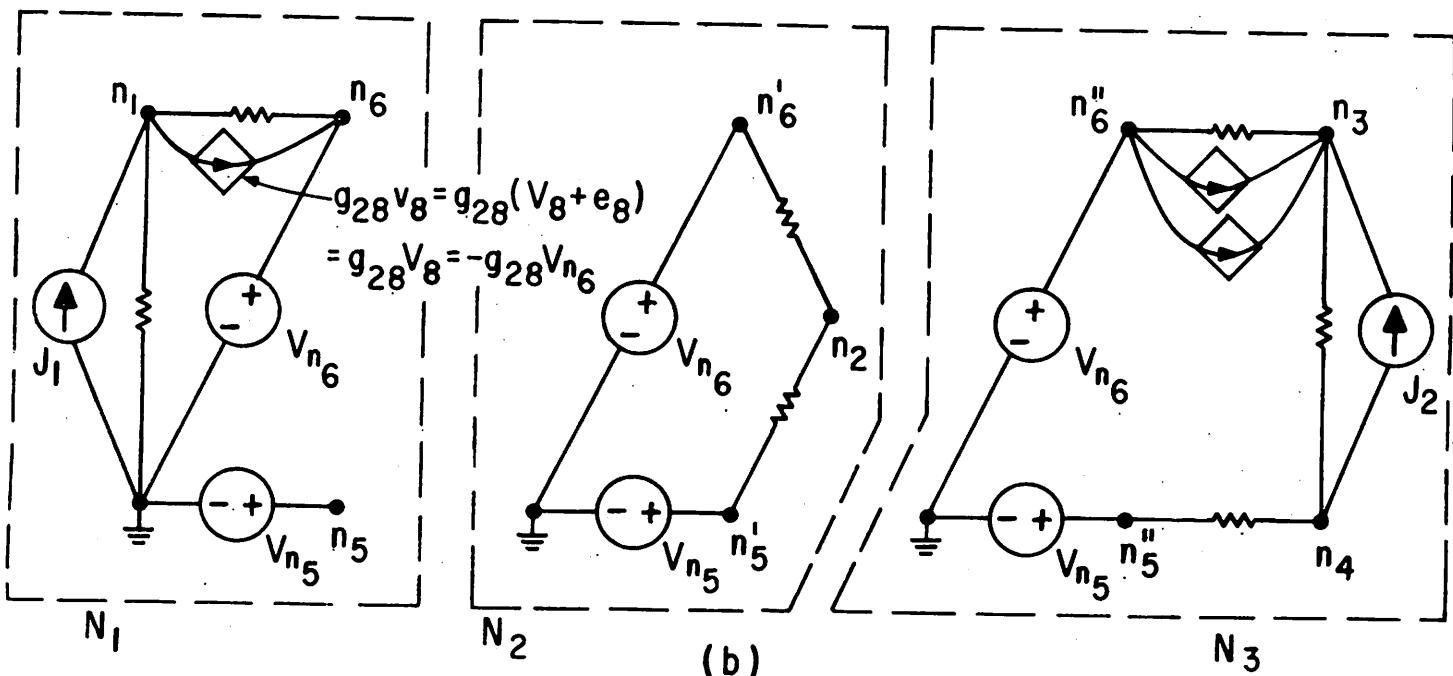
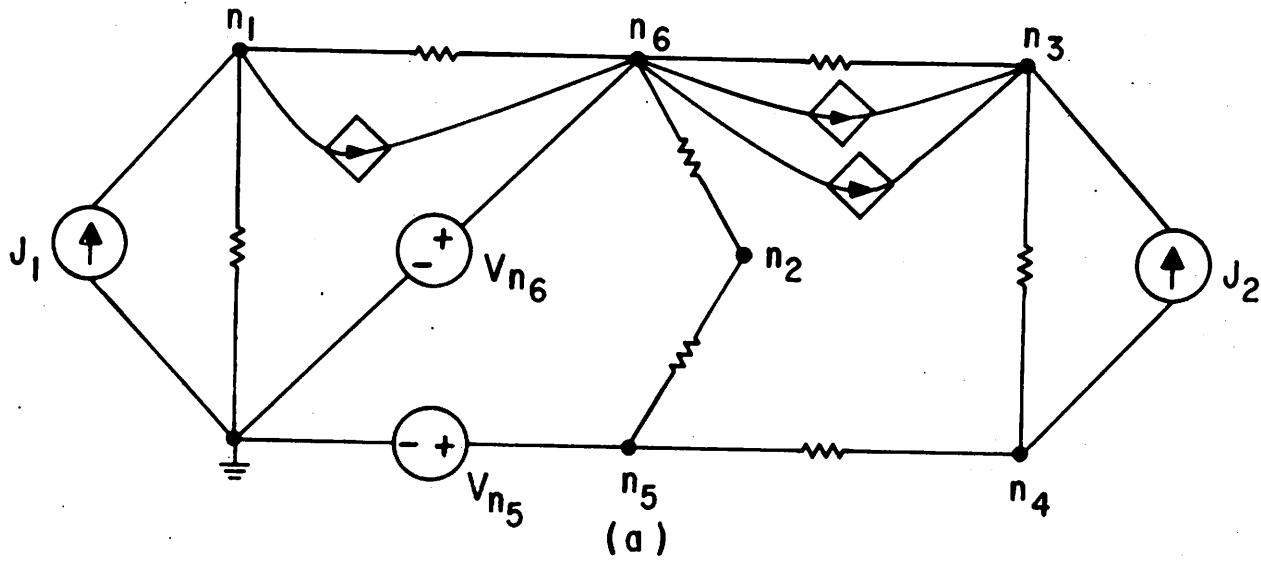


Fig. 10

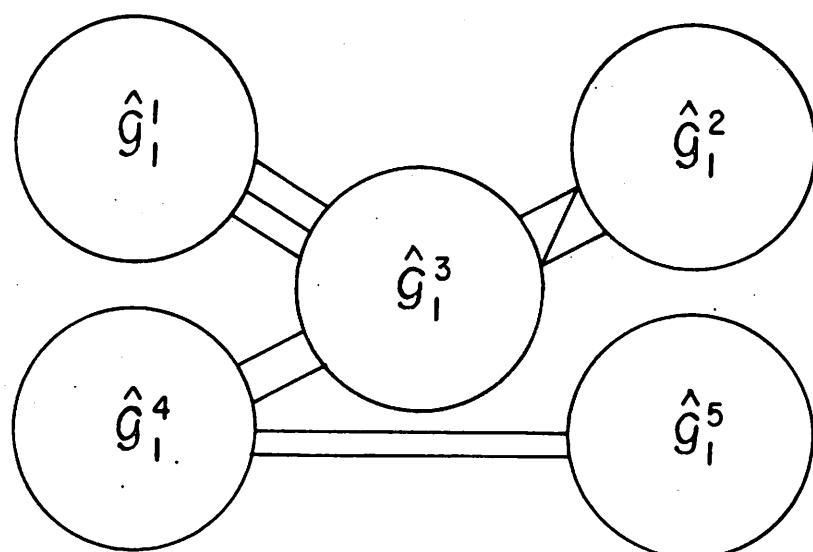


Fig. 11

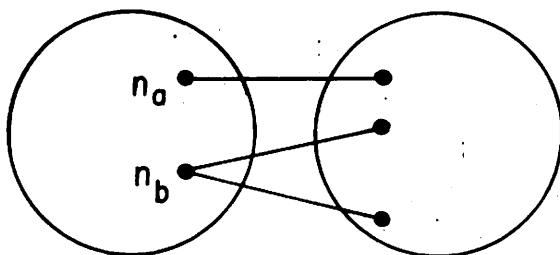
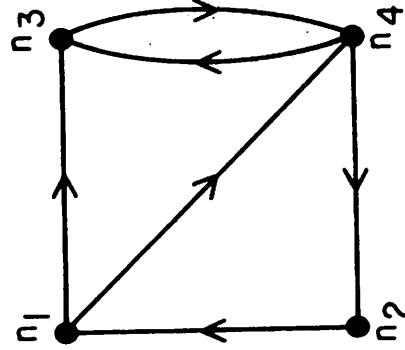


Fig. 12

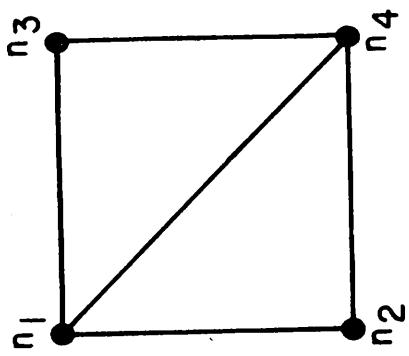
	$n_1$	$n_2$	$n_3$	$n_4$
$n_1$	X		X	X
$n_2$	X	X		
$n_3$			X	X
$n_4$			X	X

(a)

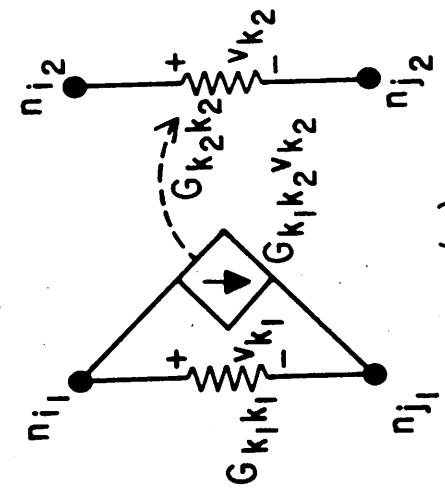


(b)

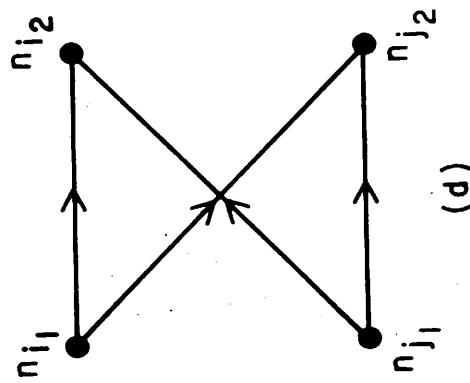
Fig. 13



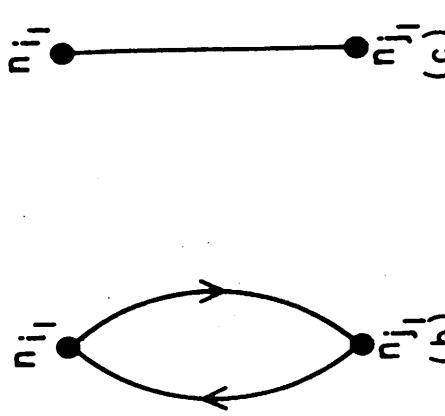
(c)



(a)



(d)



(e)

Fig. 14

IS	AS	CN
IS(1)	AS(1)	CN(1)
IS(2)	AS(2)	CN(2)
IS(3)	AS(3)	CN(3)
IS(4)	AS(4)	CN(4)
• • •	• • •	• • •

Fig.15

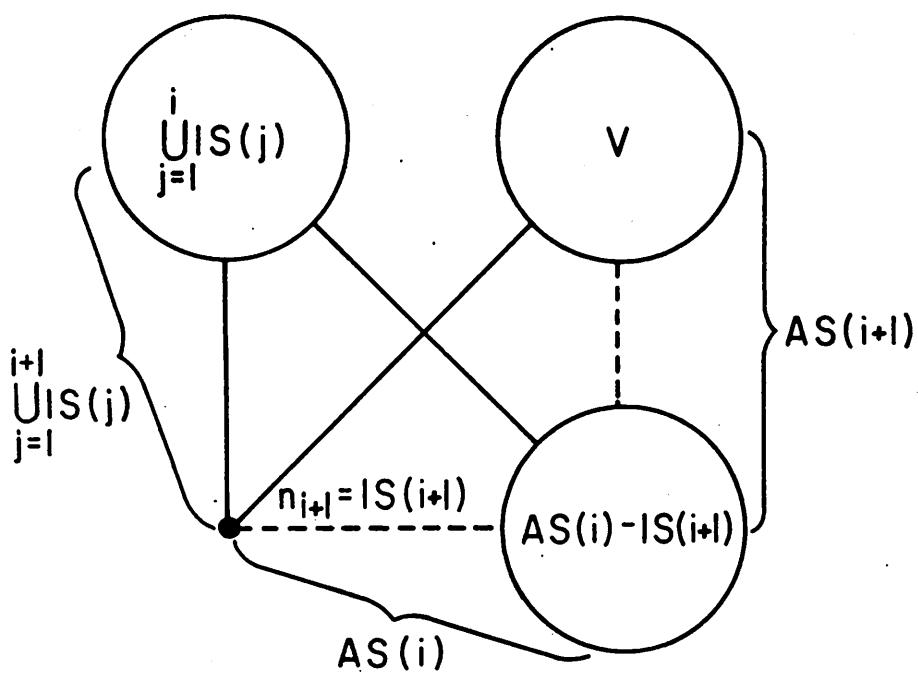


Fig.16

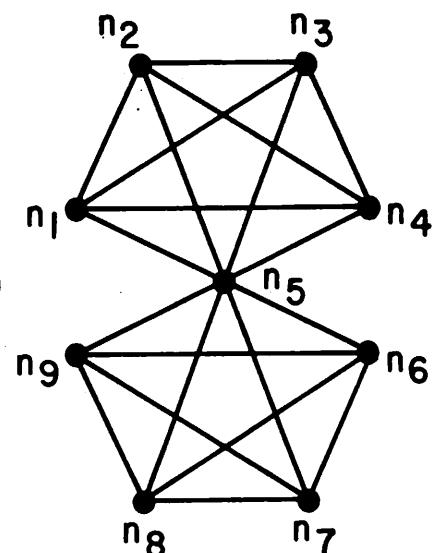


Fig.17

IS	CN	IS	AS	CN	
IS	AS	CN	IS	AS	CN
n <sub>1</sub>	n <sub>2</sub> , n <sub>3</sub> , n <sub>4</sub> , n <sub>5</sub>	4	n <sub>1</sub>	n <sub>2</sub> , n <sub>3</sub> , n <sub>4</sub> , n <sub>5</sub>	4
n <sub>2</sub>	n <sub>2</sub> , n <sub>4</sub> , n <sub>5</sub>	3	n <sub>2</sub>	n <sub>3</sub> , n <sub>4</sub> , n <sub>6</sub> , n <sub>7</sub> , n <sub>8</sub> , n <sub>9</sub>	3
n <sub>3</sub>	n <sub>2</sub> , n <sub>4</sub> , n <sub>5</sub>	2	n <sub>2</sub>	n <sub>3</sub> , n <sub>4</sub> , n <sub>6</sub> , n <sub>7</sub> , n <sub>8</sub> , n <sub>9</sub>	2
n <sub>4</sub>	n <sub>2</sub> , n <sub>4</sub> , n <sub>6</sub> , n <sub>7</sub> , n <sub>8</sub> , n <sub>9</sub>	6	n <sub>3</sub>	n <sub>4</sub> , n <sub>6</sub> , n <sub>7</sub> , n <sub>8</sub> , n <sub>9</sub>	6
n <sub>5</sub>	n <sub>2</sub> , n <sub>4</sub> , n <sub>6</sub> , n <sub>7</sub> , n <sub>8</sub> , n <sub>9</sub>	5	n <sub>4</sub>	n <sub>5</sub>	4
n <sub>6</sub>	n <sub>2</sub> , n <sub>4</sub> , n <sub>7</sub> , n <sub>8</sub> , n <sub>9</sub>	5	n <sub>5</sub>	n <sub>6</sub> , n <sub>7</sub> , n <sub>8</sub> , n <sub>9</sub>	4
n <sub>7</sub>	n <sub>4</sub> , n <sub>7</sub> , n <sub>8</sub> , n <sub>9</sub>	4	n <sub>6</sub>	n <sub>7</sub> , n <sub>8</sub> , n <sub>9</sub>	3
n <sub>8</sub>	n <sub>4</sub> , n <sub>7</sub> , n <sub>8</sub>	3	n <sub>7</sub>	n <sub>8</sub> , n <sub>9</sub>	2
n <sub>9</sub>	n <sub>8</sub>	2	n <sub>8</sub>	n <sub>9</sub>	1
n <sub>10</sub>	φ	0	n <sub>9</sub>	φ	0
n <sub>11</sub>	φ	0	n <sub>9</sub>	φ	0

(c)

(b)

(a)

Fig. 18

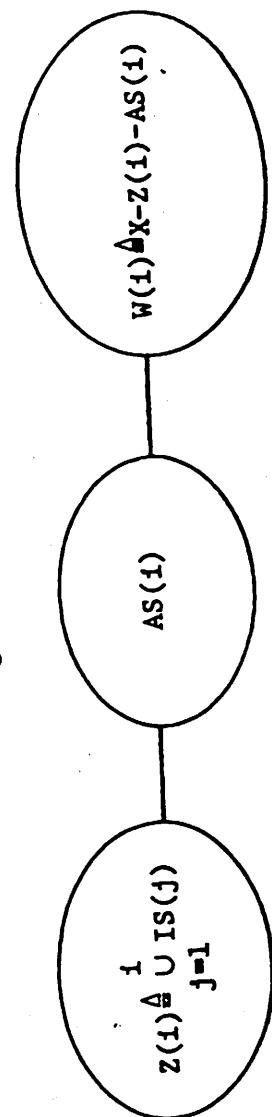
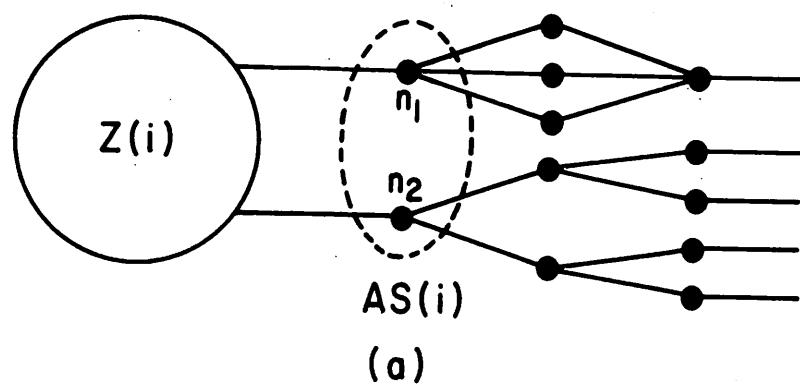
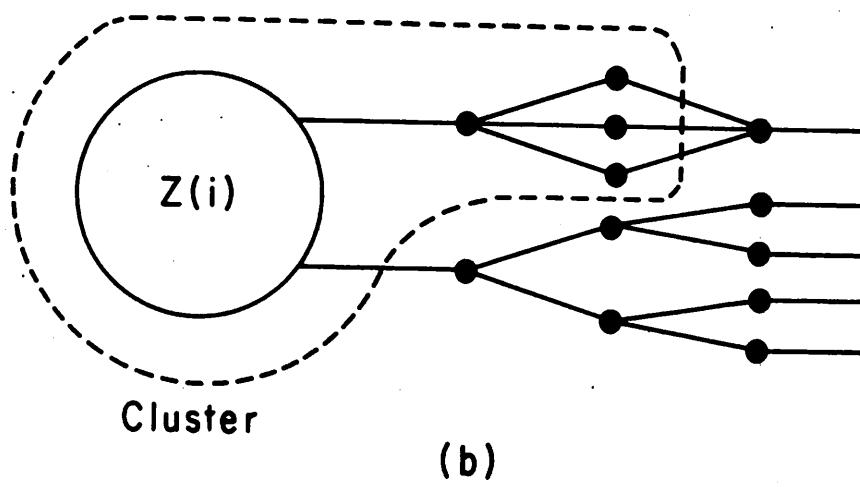


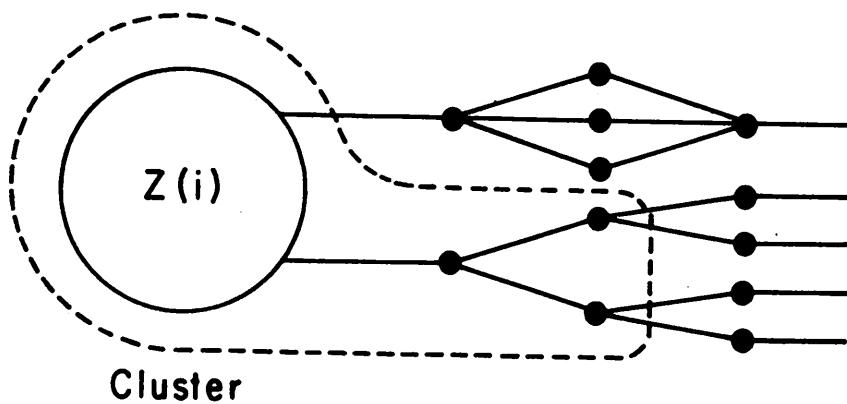
Fig. 19



(a)



(b)



(c)

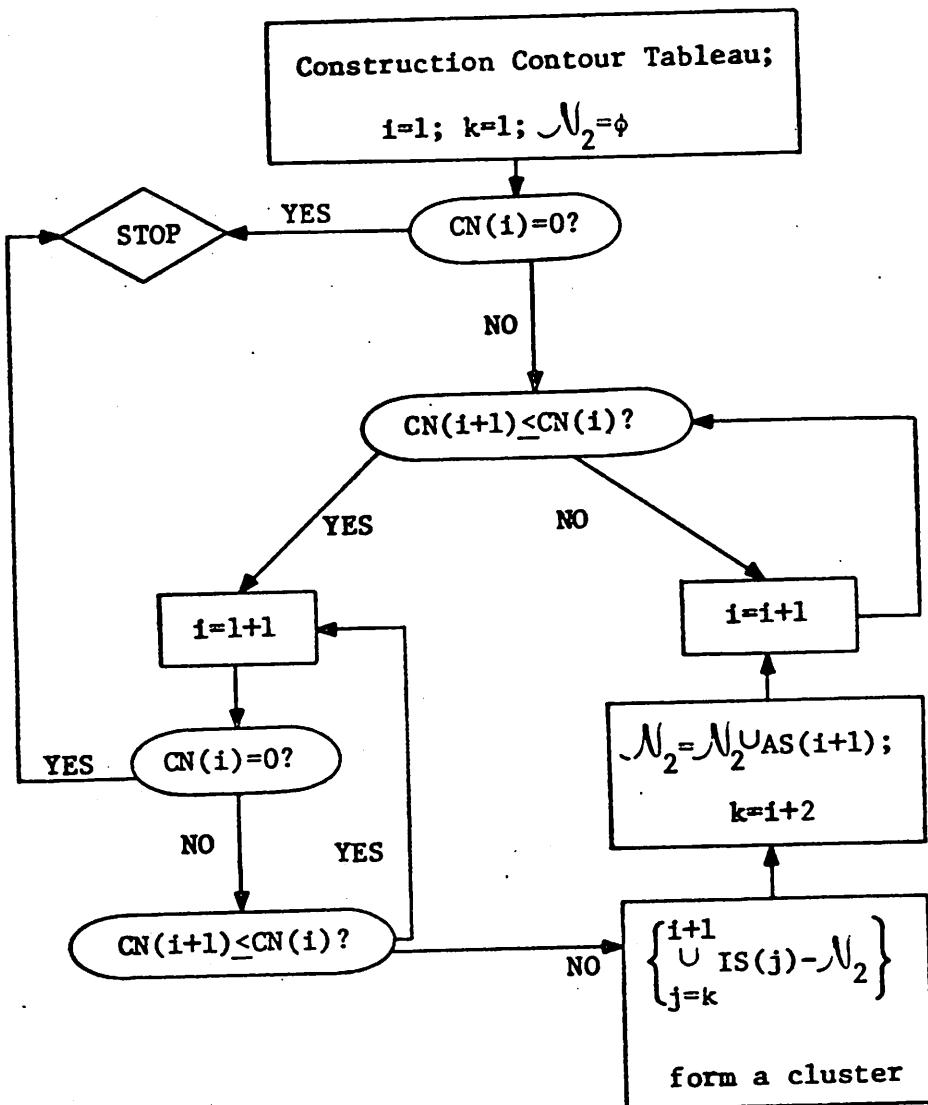


Fig. 21

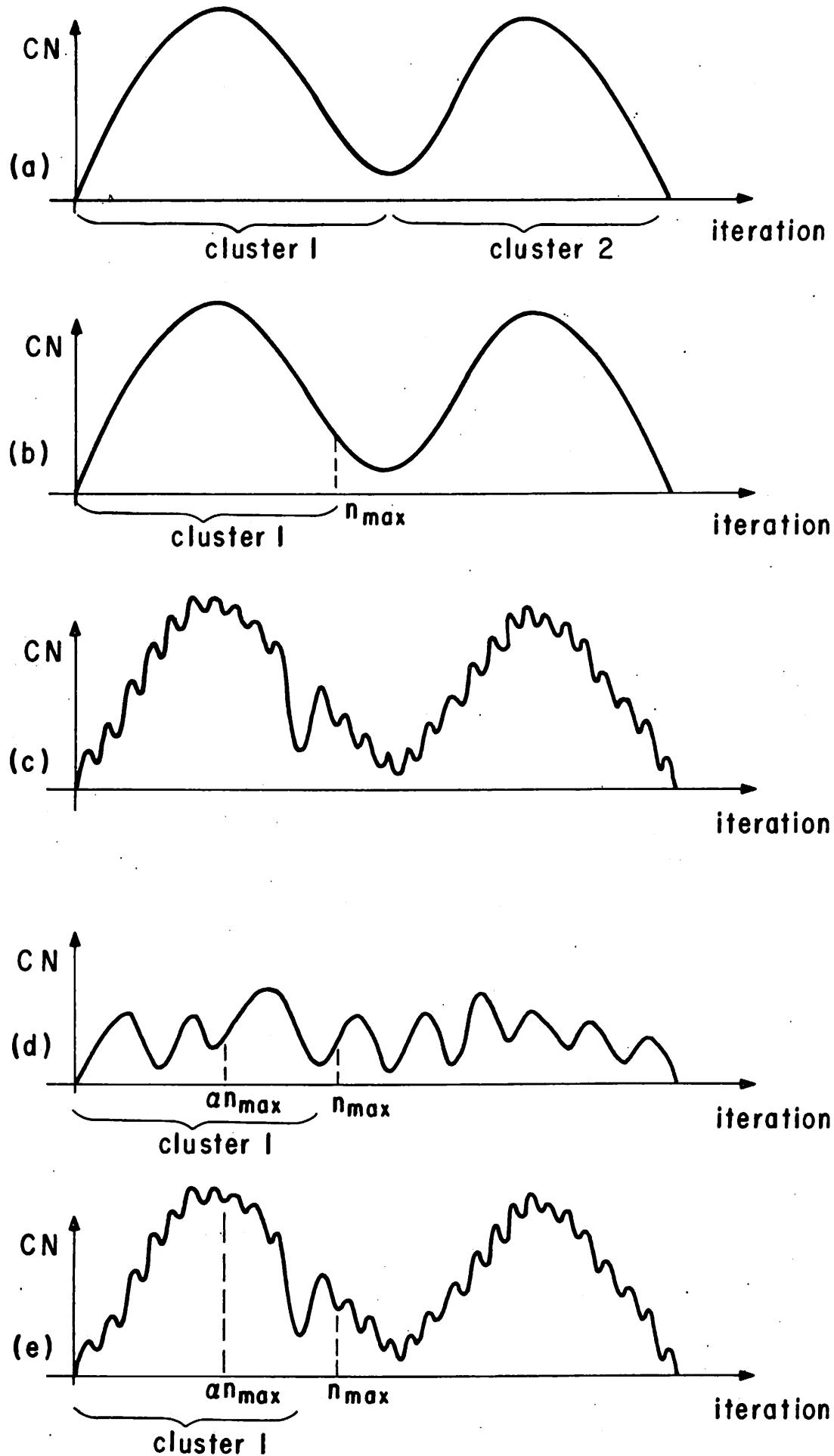
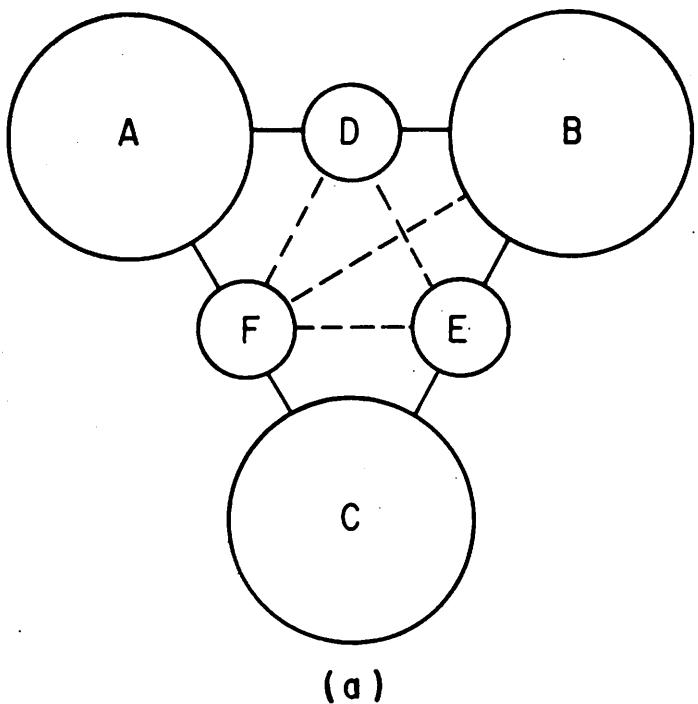
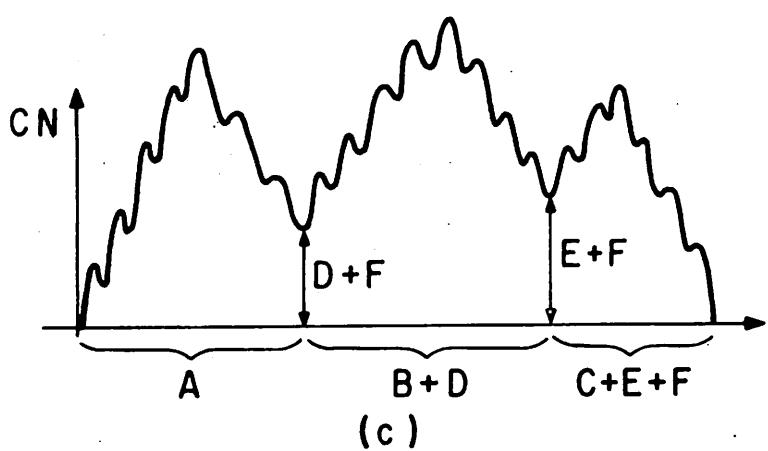


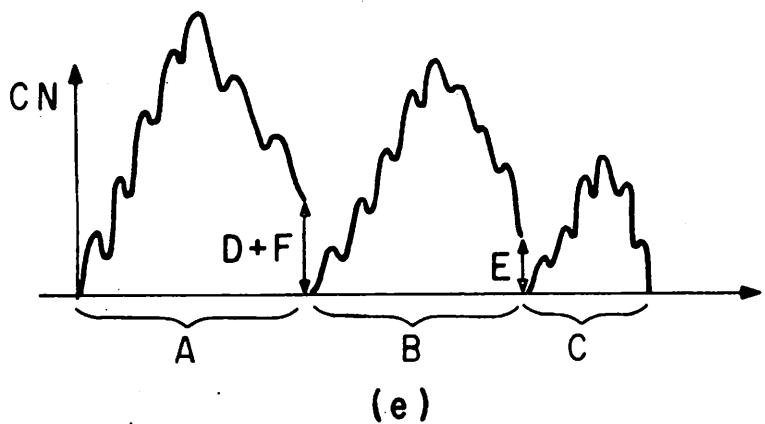
Fig. 22 -92-



(a)



(c)



(e)

IS	AS	CN
A		
B + D	D + F	
C + E + F	E + F	
	$\phi$	0

(b)

IS	AS	CN
A		
B	D + F	
C	E	
	$\phi$	0

(d)

Fig. 23.

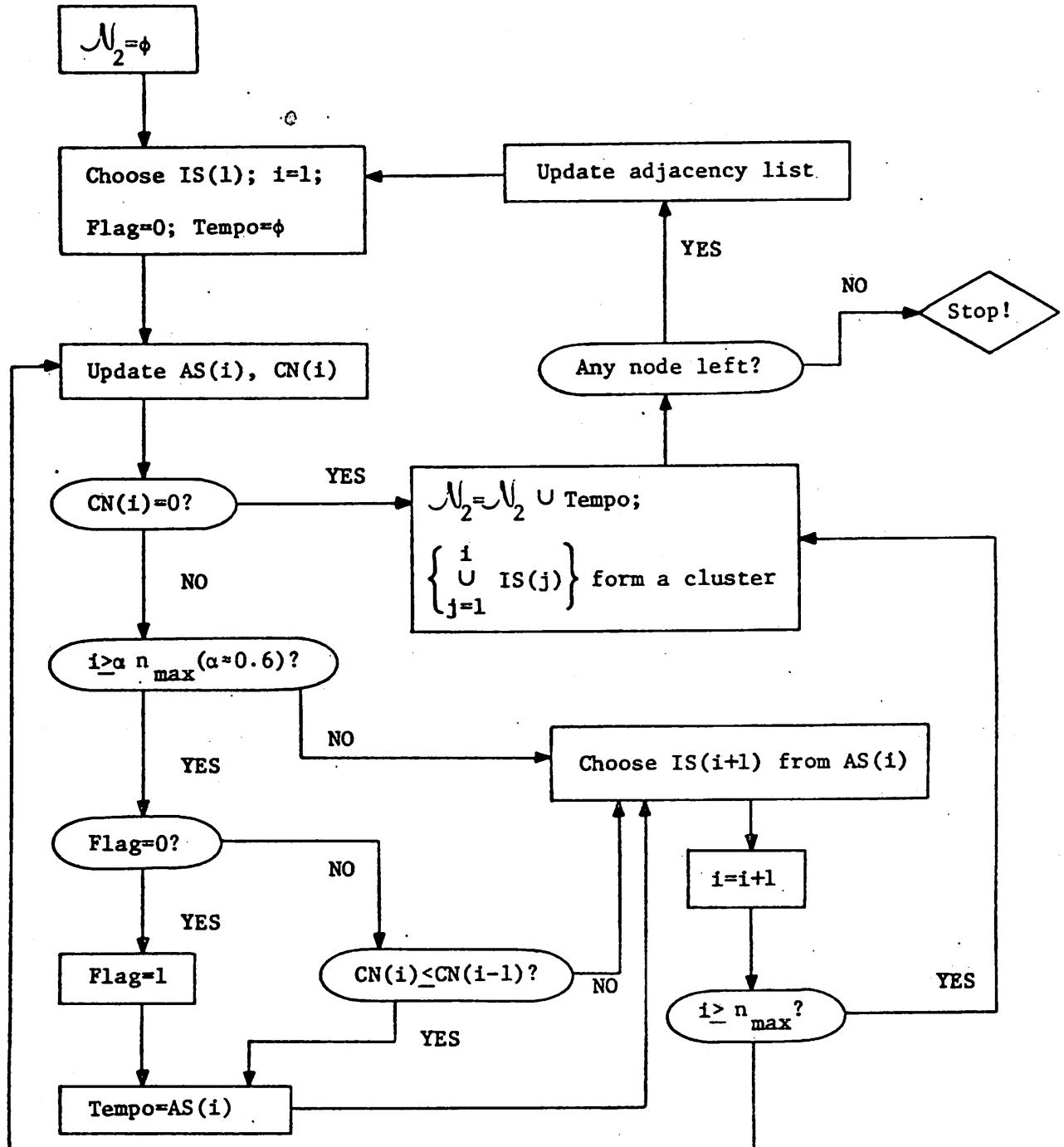
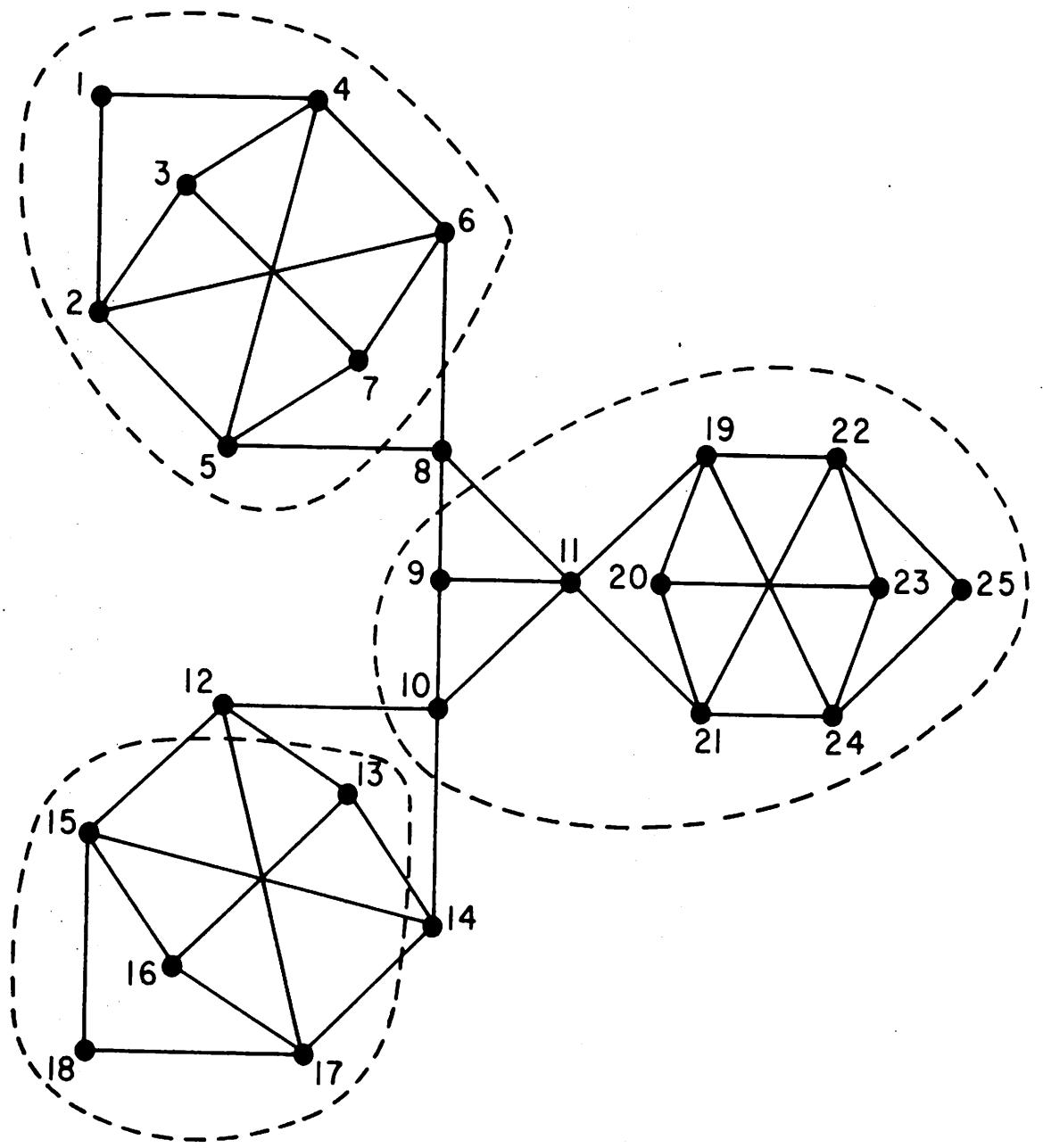


Fig. 24



(a)

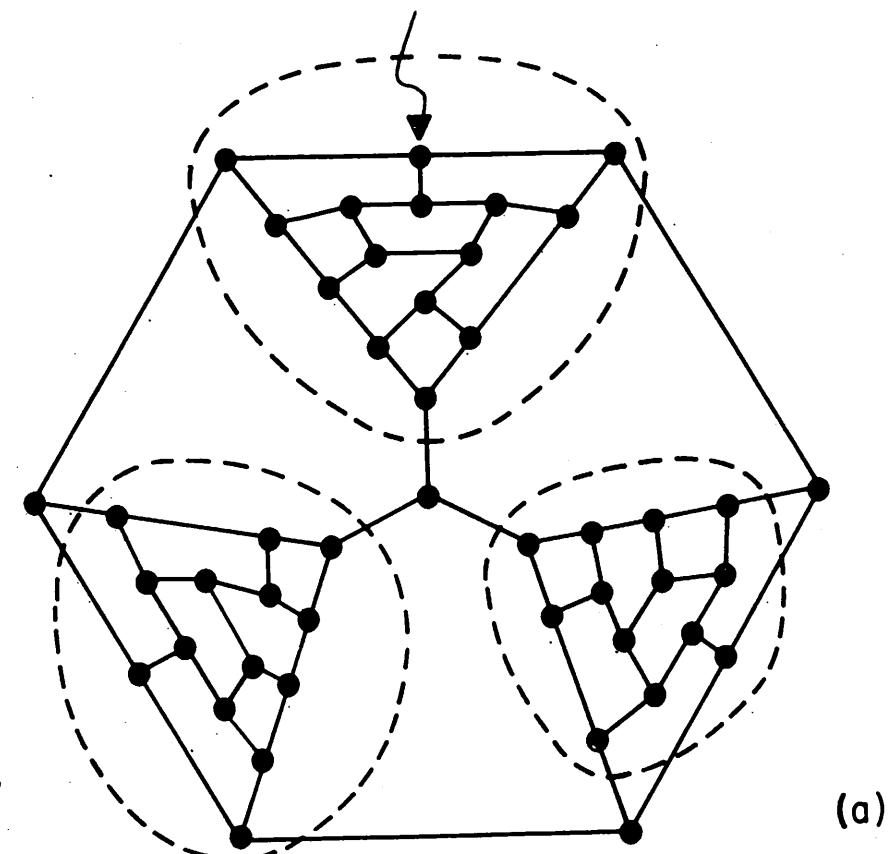
Fig. 25

IS	AS	CN
1	1 2,4	2
2	2 4,3,5,6	4
3	4 3,5,6	3
4	3 5,6,7	3
5	7 5,6	2
6	5 6,8	2
7	6 8	1
8	8 9,11	2
9	9 11,10	2
10	11 10,19,21	3
11	11 9,10,19,21	4
2	9 10,19,21	3
3	10 19,21,12,14	4
4	19 21,12,14,20,22,24	6
5	21 12,14,20,22,24	5
6	20 12,14,22,24,23	5
7	23 12,14,22,24	4
8	22 12,14,24,25	4
9	24 12,14,25	3
10	25 12,14	2
13	16	1
16	15,17	2
15	17,18	2
17	18	1
18	∅	0

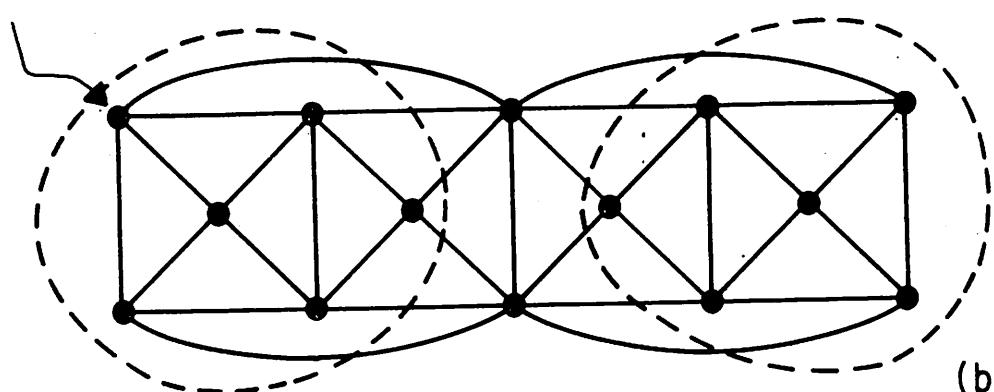
(b)

- ← 1st cluster {1,2,4,3,7,5,6}; throw away cluster and bottleneck bottleneck nodes (i.e., node 8); Start again.
- ← 2nd cluster {11,9,10,19,21,20,23,22,24,25}; throw away cluster and bottleneck nodes (i.e., nodes 12, 14); Start again.
- ← 3rd cluster {13,16,15,17,18}; Stop!

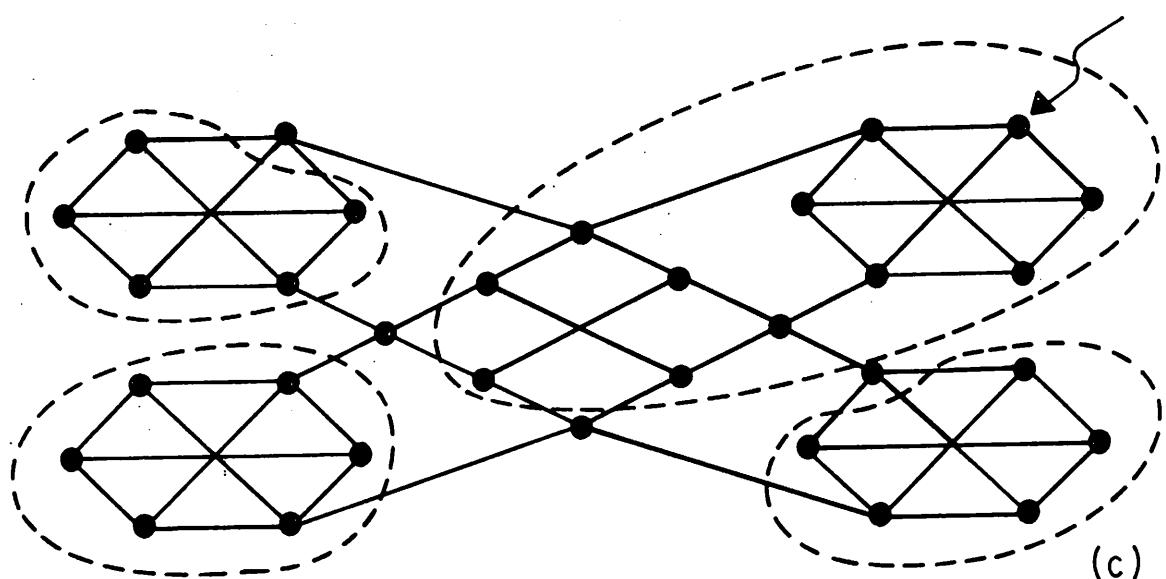
Fig. 25



(a)



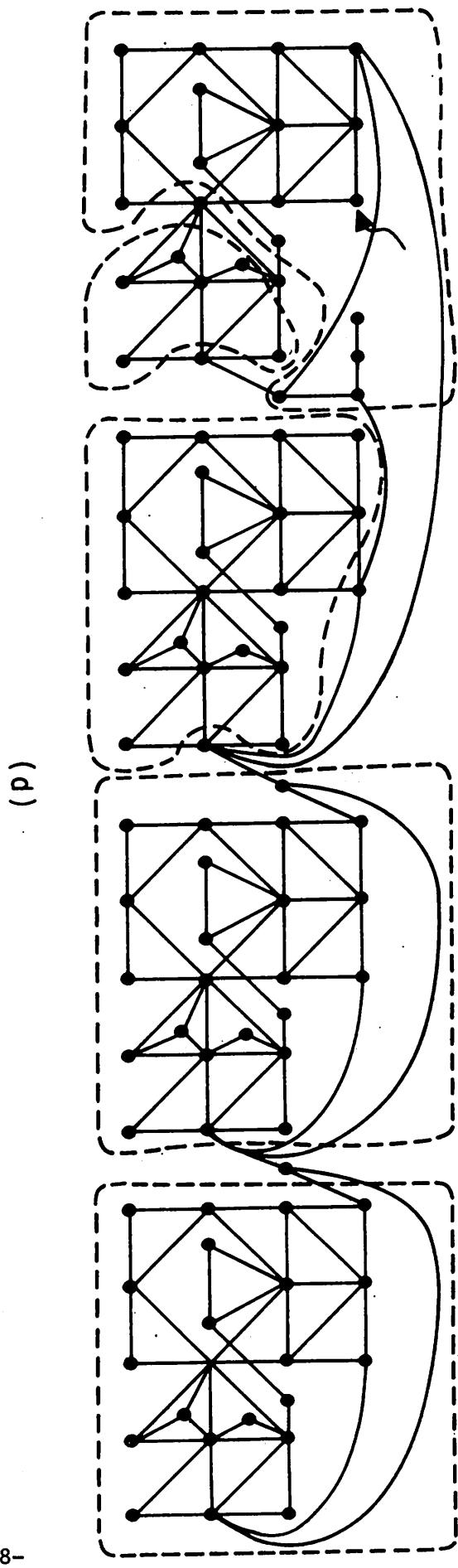
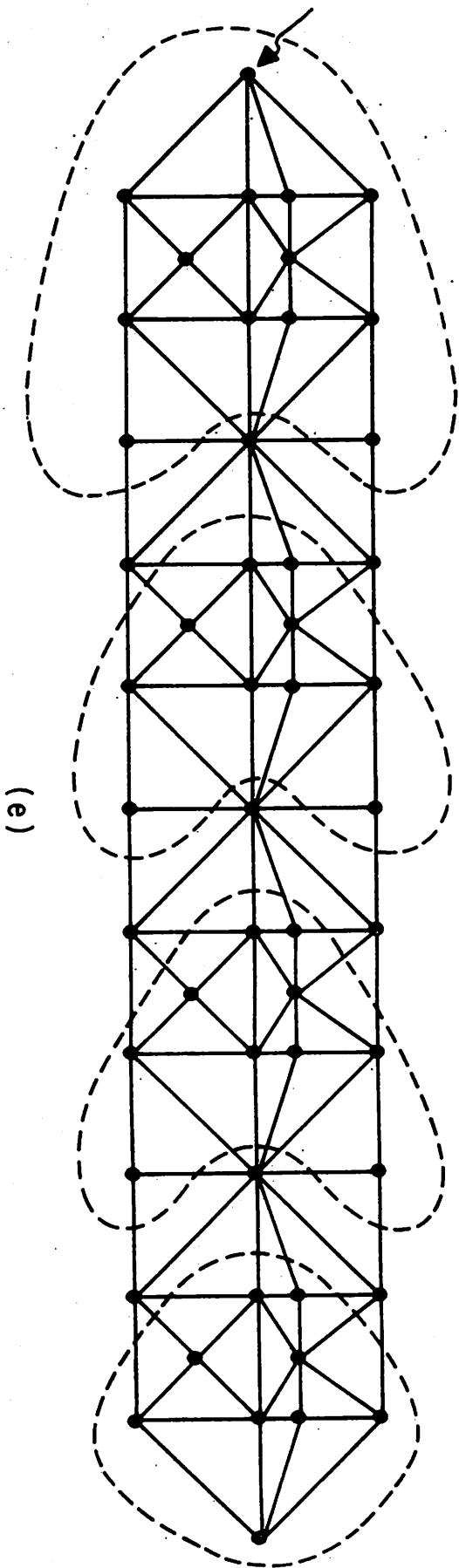
(b)

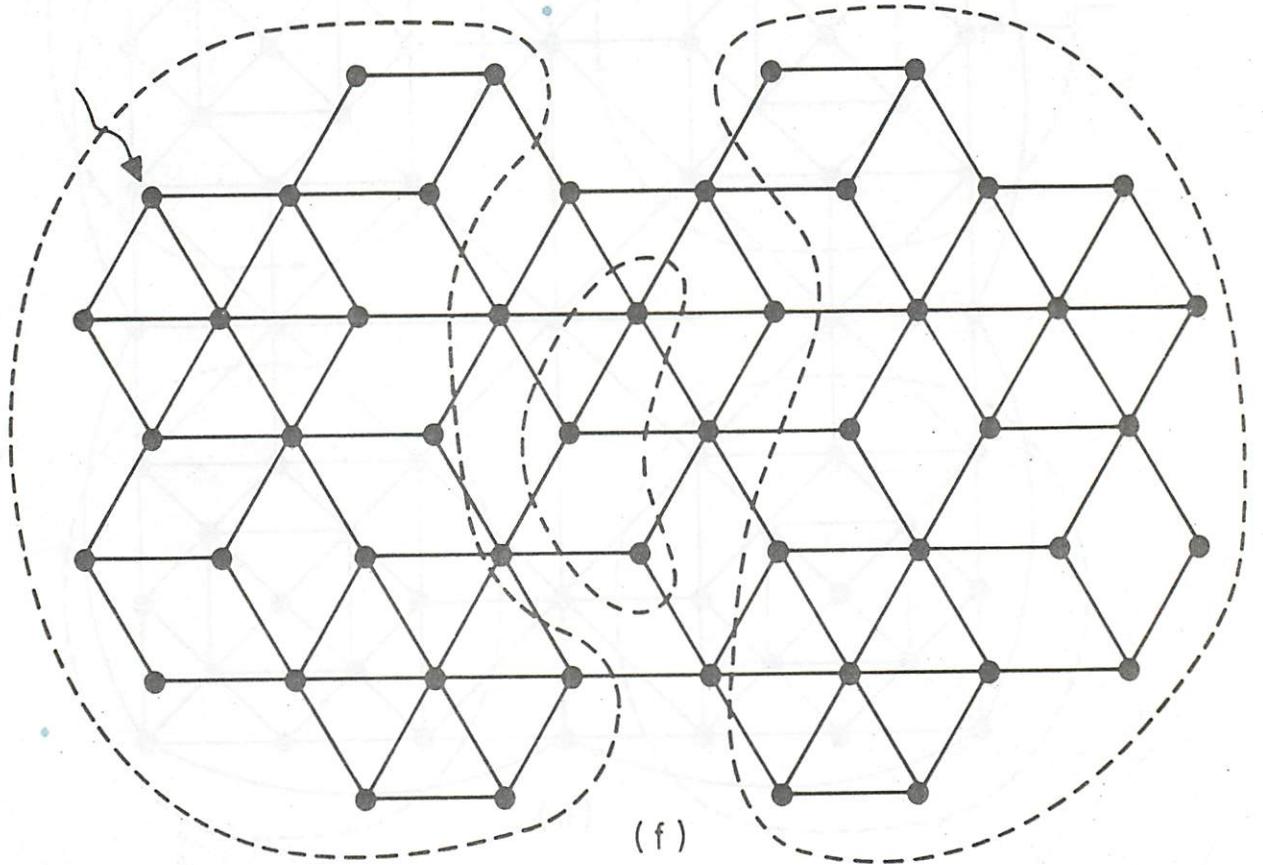


(c)

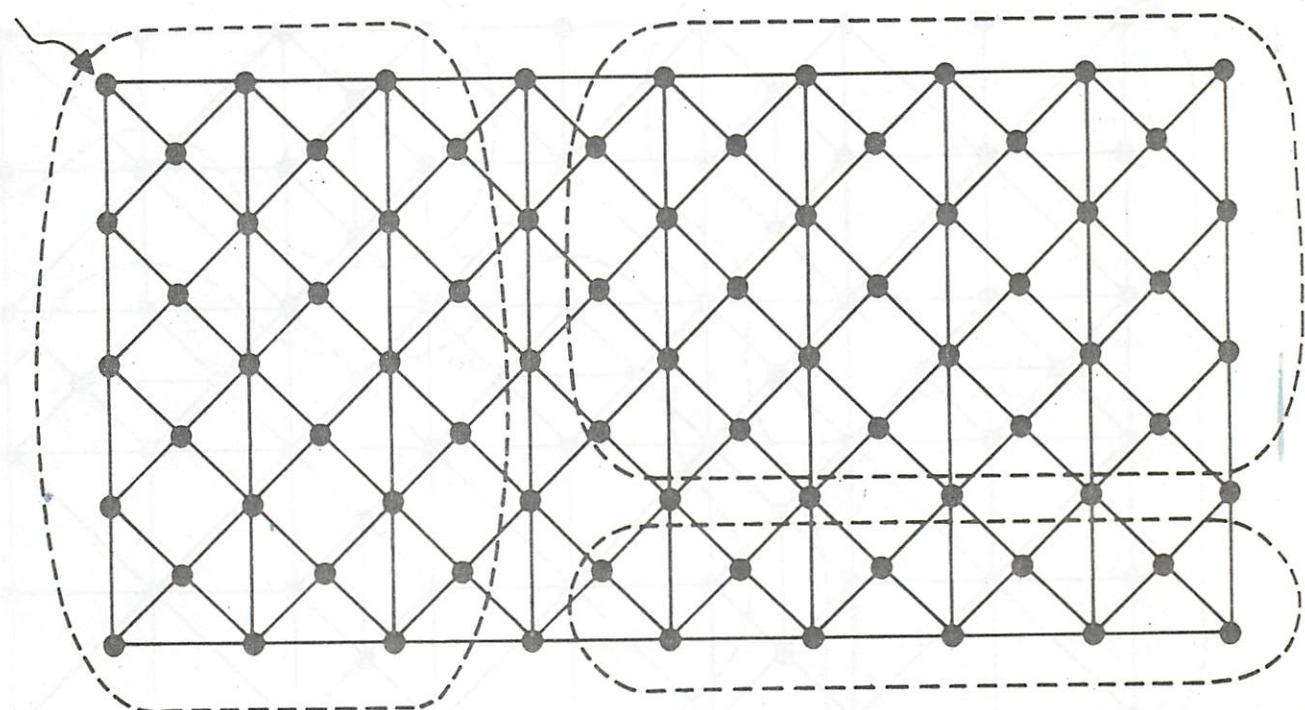
Fig. 26

Fig. 26



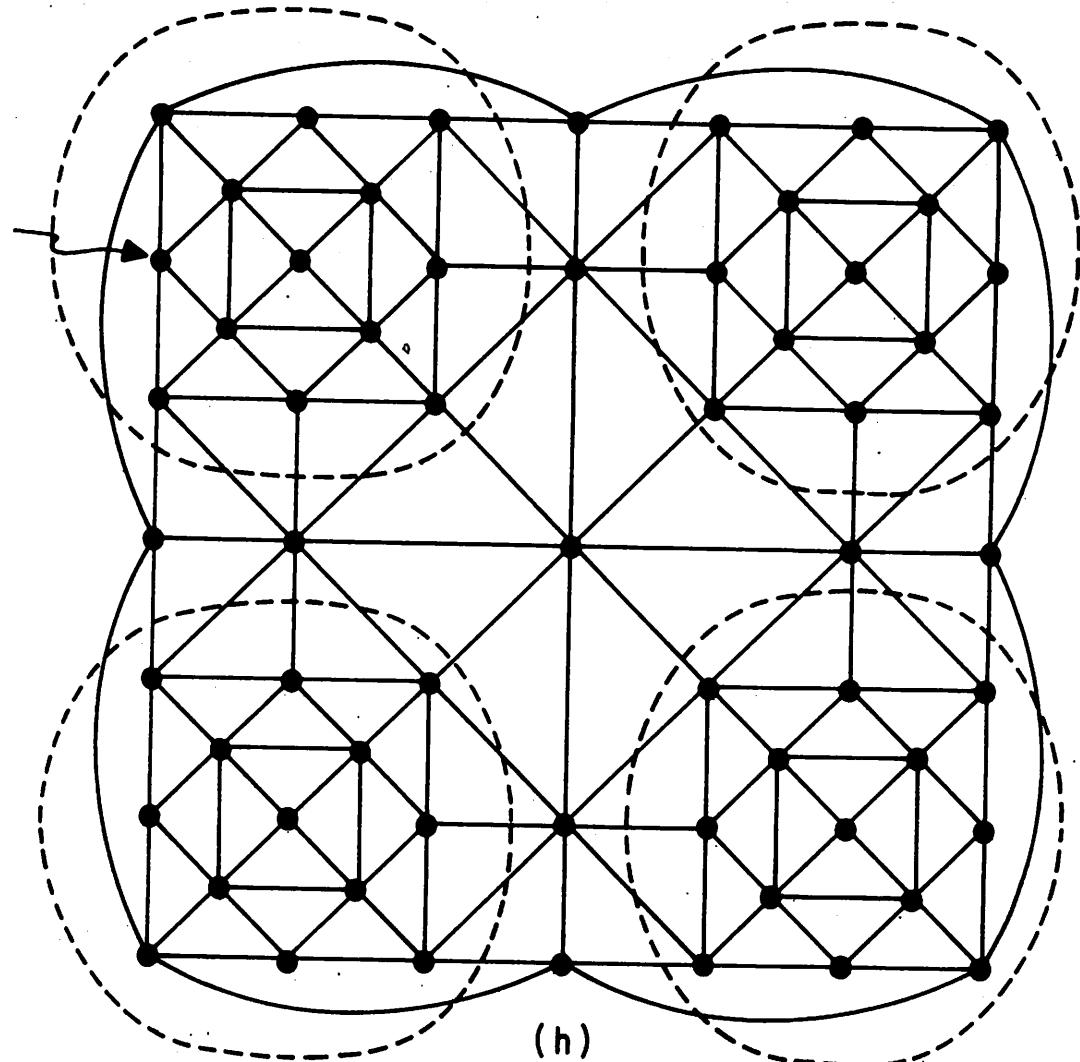


(f)

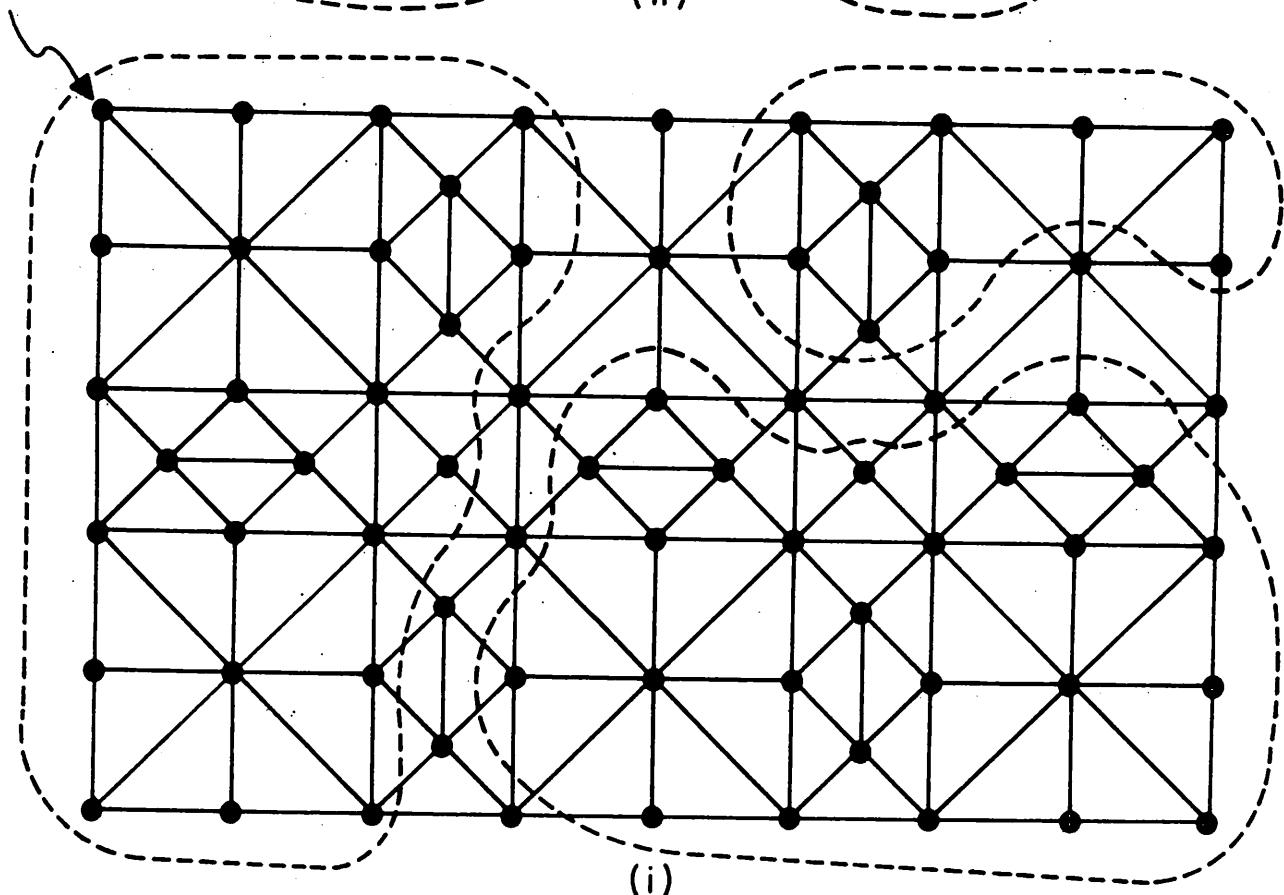


(g)

Fig. 26



(h)



(i)

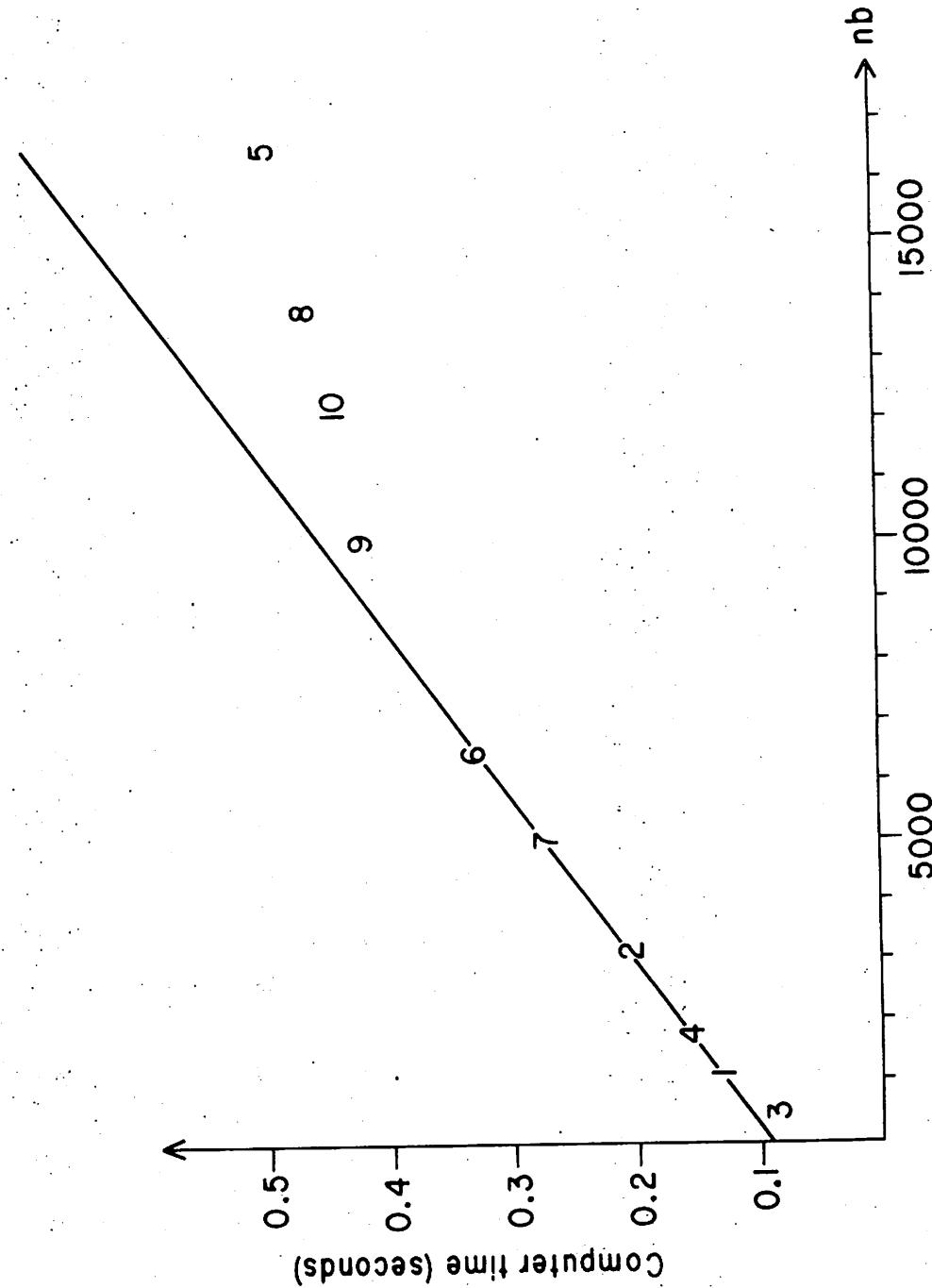
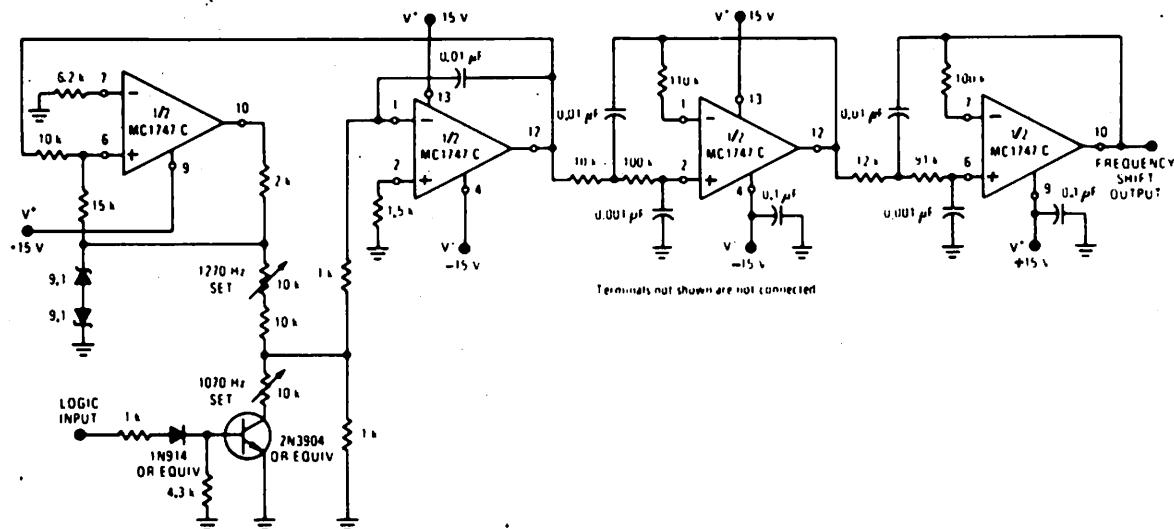
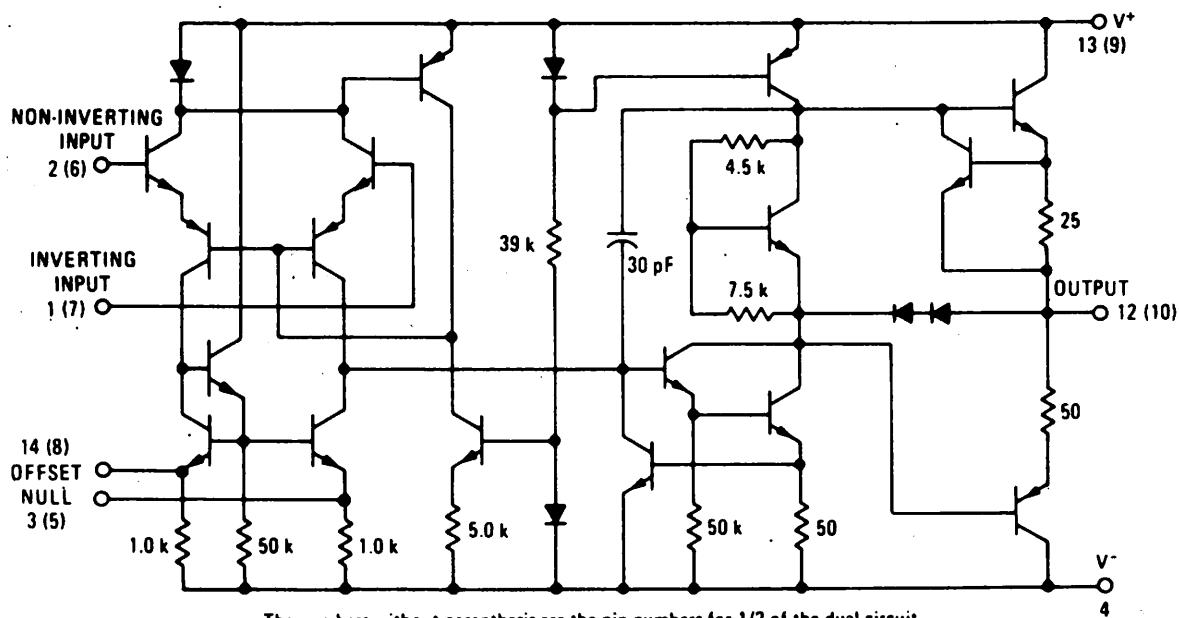


Fig. 27



(a)



(b)

Fig. 28

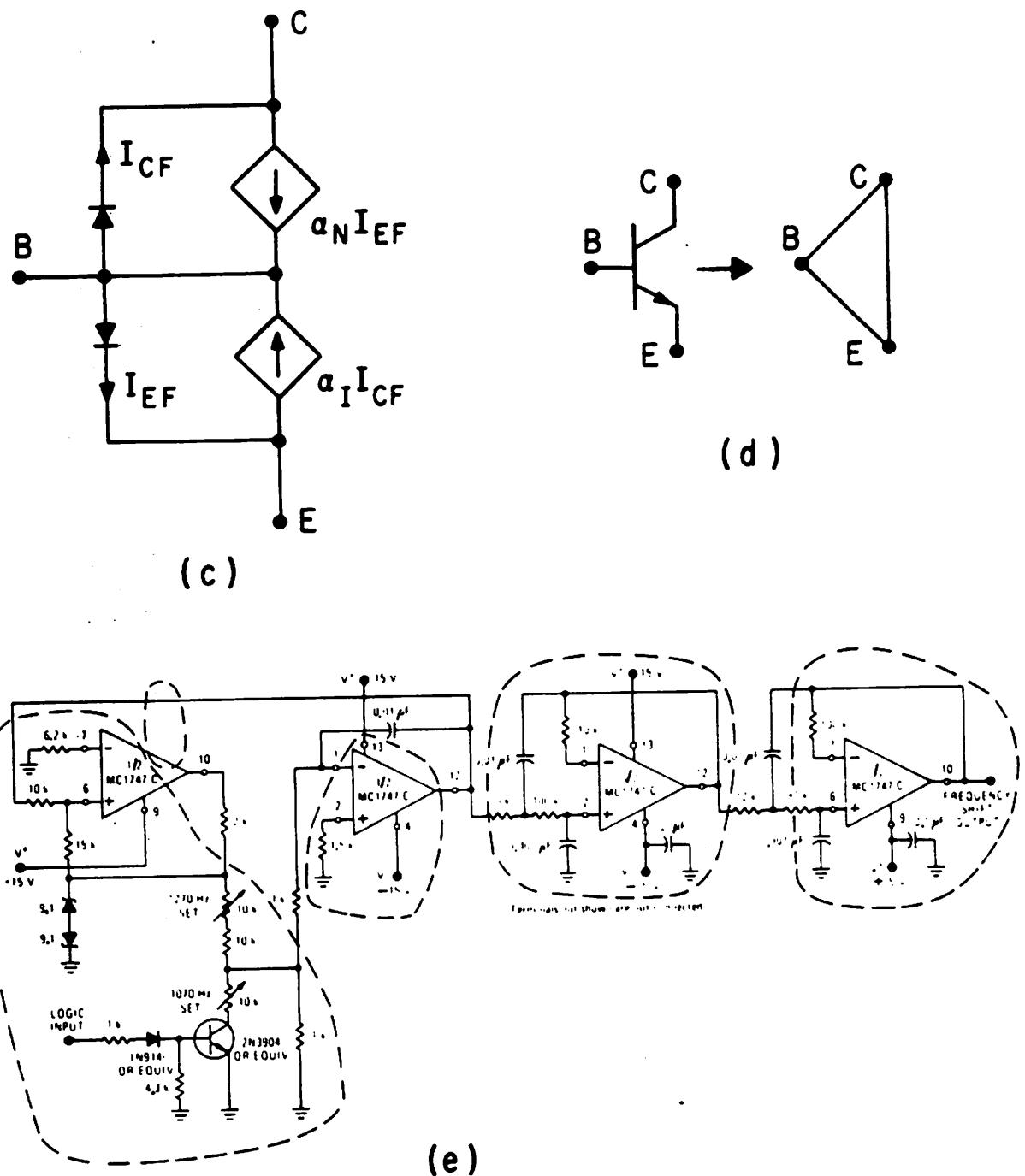


Fig. 28

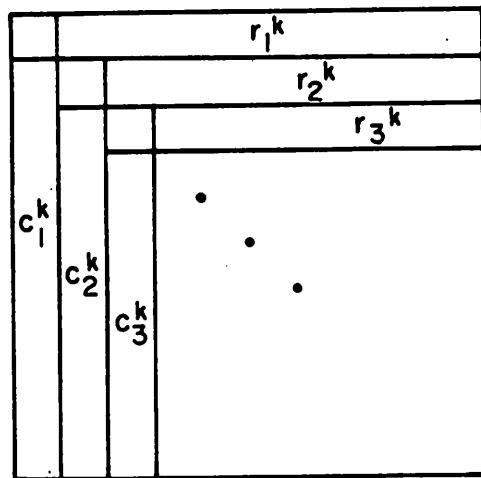


Fig. A1

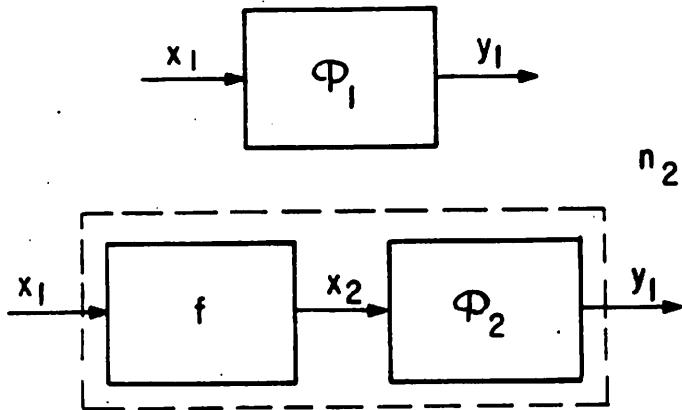


Fig. B1

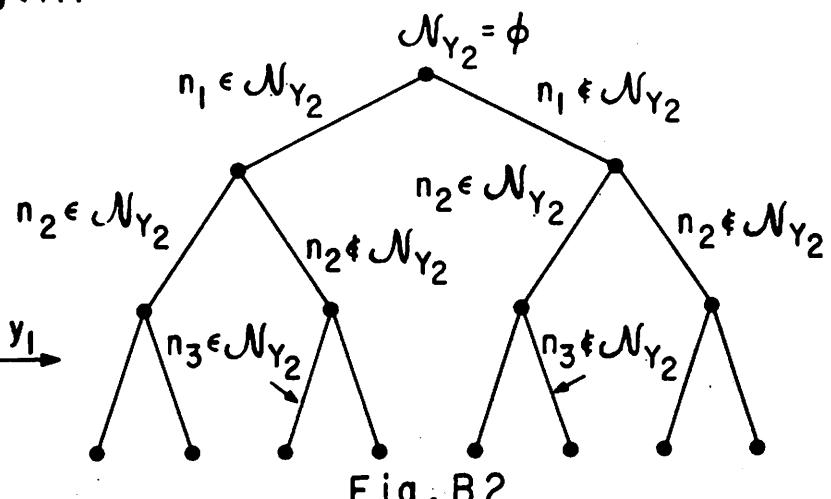


Fig. B2

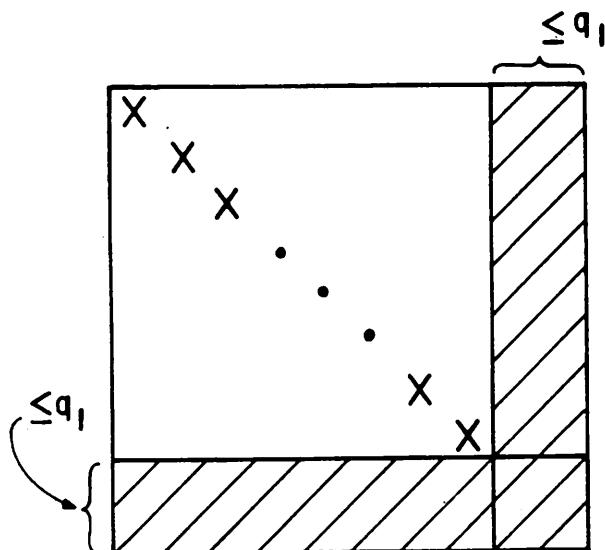


Fig. B3

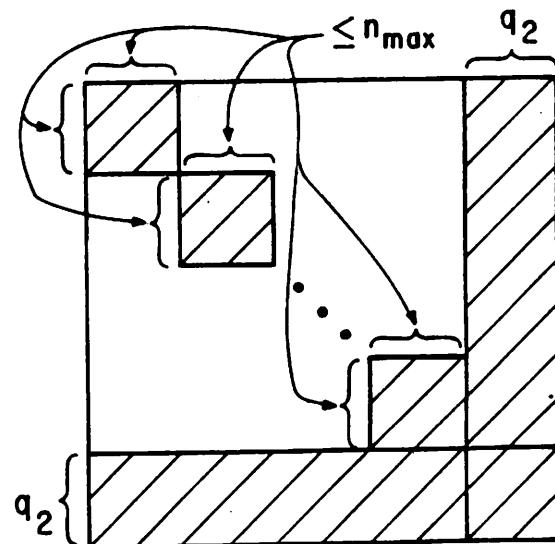
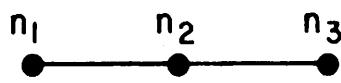
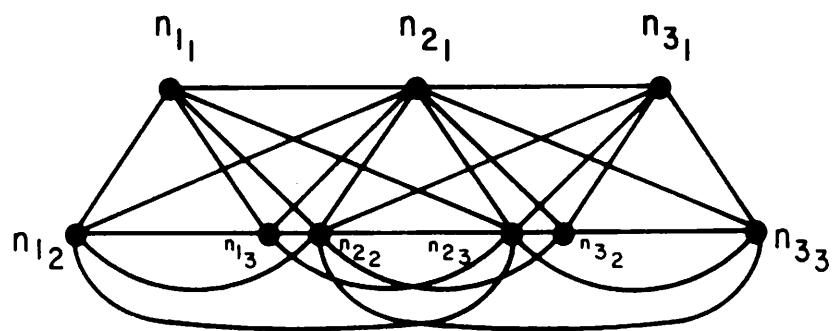


Fig. B4



(a)



(b)

Fig. B5

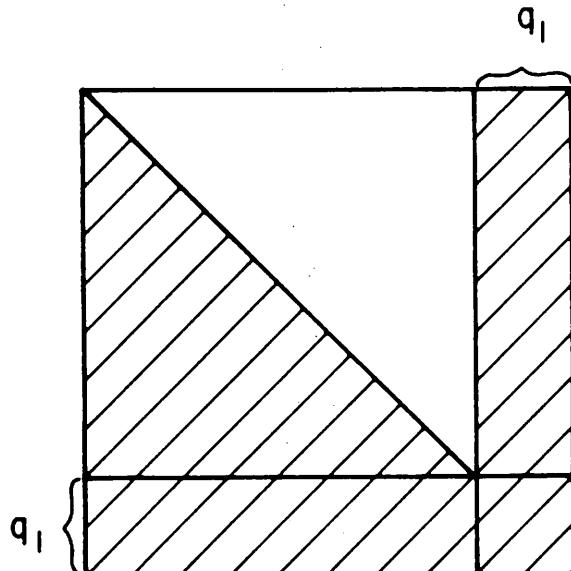


Fig. B6

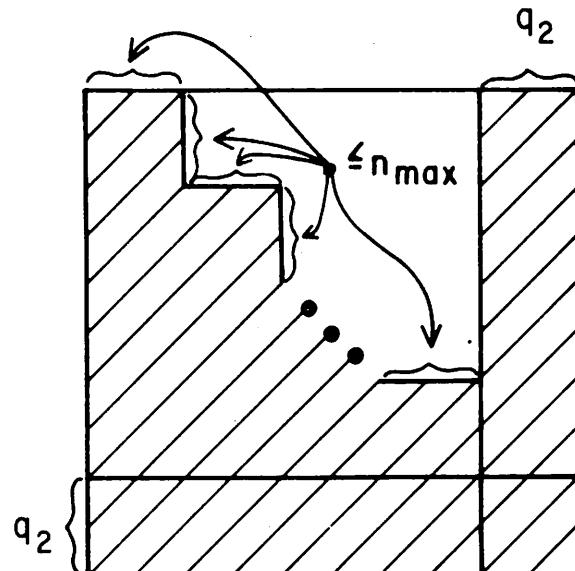
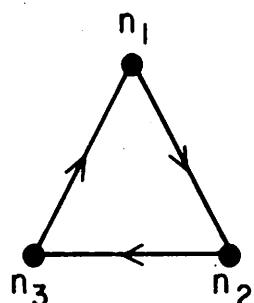
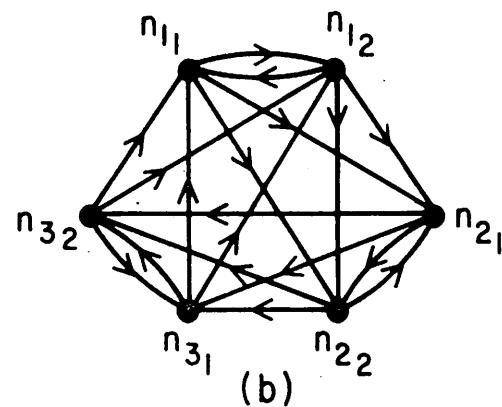


Fig. B7



(a)



(b)

Fig. B8