# A NOTE ON BIPARTITE GRAPHS AND PIVOT SELECTION IN SPARSE MATRICES

by

A. Sangiovanni-Vincentelli

Memorandum No. ERL-M568

21 November 1975

ELECTRONICS RESEARCH LABORATORY

College of Engineering
University of California, Berkeley
94720

# A NOTE ON BIPARTITE GRAPHS AND PIVOT SELECTION IN SPARSE MATRICES

A. Sangiovanni-Vincentelli*

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

ABSTRACT

In the tableau approach to large electrical network analysis, as well as in structure analysis, finite element method, linear programming, etc., a very sparse linear algebraic set of equations $Ax = b$ has to be solved repeatedly. In order to efficiently solve the system via Gaussian Elimination, an optimization problem has to be faced: the selection of a pivot strategy to maintain the sparsity of the matrix A. While a certain number of theoretical results are available when the pivotal elements are chosen on the main diagonal, very few results have been obtained when the selection is done out of the main diagonal. The problem is usually solved via heuristic algorithms. The general structure of these algorithms is such that there is no guarantee that the pivotal elements sequentially chosen were nonzero in the original matrix. In this case, Brayton et al. have shown that Gaussian Elimination is no longer optimal in the sense that unnecessary arithmetic operations as well as unnecessary storage requirements may be produced. In this paper a graph theoretical interpretation of nonsymmetrical pivotal strategies is given and an efficient algorithm which enables to select always nonzero pivotal elements in A, is proposed.

---

*On leave from Istituto di Elettrotecnica ed Elettronica, Politecnico di Milano Italy.

Research sponsored in part by the National Science Foundation grant ENG72-03783.

## I. INTRODUCTION

Sparse matrix techniques [1-4] have been applied in many different technological areas such as structure analysis, finite element method, linear programming and circuit analysis [5-7]. These techniques are concerned with the solution of linear algebraic systems of equations $Ax = b$, when the coefficient matrix A is sparse [1-4]. Their purpose is to fully exploit the sparsity in order to lower the complexity of computer computations. In ordinary Gaussian Elimination (GE) [8], the choice of a pivot strategy is fundamental in order to economize computer storage and time [1-7]. The figure to be minimized is usually the number of fill-ins, i.e., of the nonzero elements introduced during the elimination process [1-7,9-12], or the number of arithmetic operations.

When A is symmetric and positive definite, it is obvious to restrict the pivot choice on the main diagonal [12]. Rose, Ohtsuki et al., Ogbuobiri et al., etc. [12-16], have introduced a graph theoretic interpretation and proved theorems in order to find efficient near-optimum algorithms for the symmetric case.

In the sparse tableau approach to electrical network analysis and design [6], which can be considered as one of the most efficient available techniques, the structure of the coefficient matrix A is highly nonsymmetric and A is not positive definite. In this case, there is no reason to restrict the search of pivot elements on the main diagonal. However the complexity of the optimal pivot selection is by far increased. Some heuristic algorithms [6,7,17-21] have been devised, but very few theoretical results have been obtained. In particular, all the heuristic algorithms available do not assure that the pivotal elements (chosen in a sequential way simulating the elimination process on a structural matrix associated to A) are nonzeros in the matrix before the elimination procedure starts. In this case, Brayton et al. [22] have shown that Gaussian Elimination as well as LU decomposition is not "optimal," i.e., some unnecessary operations and storage requirements may be needed, while if all the selected elements were nonzeros in A, G.E. and LU decomposition are optimal.

As in the symmetric case, a graph theoretic interpretation may be helpful to devise and compare pivot strategies. At the moment, two of them are available:

1) one given by the author [23] and based on simple digraphs and on graph operations on them.

2) the other given by Shirikawa et al. [19] and based on bipartite graphs.

In this paper, a bipartite graph representation similar to the one given in [19] is used in order to build up an algorithm able to select always nonzero elements in A as pivots. It has to be noted that this algorithm can be used as a general framework in which it is possible to insert whatsoever heuristic procedure to minimize the computation time and the storage requirements. In particular, the paper is organized as follows: in Section II, some preliminary remarks and graph theoretic definitions are given. In Section III, the bipartite graph interpretation is introduced and in Section IV the algorithm is described and its complexity is evaluated. In Section V some concluding remarks are given.

## II. PRELIMINARY DEFINITIONS AND REMARKS

A graph theoretical background is presented in this Section. All the undefined terms are to be understood according to Harary [24]. Let $G = (X,U)$ ($G = (X,E)$) be a <u>(di)graph</u> with a set of <u>vertices</u> or <u>nodes</u> $X$ and a set of (directed) <u>edges</u> or <u>arcs</u> $U = \{\{x_i,x_j\} | x_i, x_j \in X\}$ ($E = \{(x_i,x_j) | x_i,x_j \in X\}$).

A simple (directed) path $\mu(x_i,x_j)$ of length $\ell$ is an ordered sequence of distinct vertices.

$$\mu(x_i,x_j) = \langle p_o,p_1,\ldots,p_\ell \rangle$$

such that $p_o = x_i$, $p_\ell = x_j$, $\{p_k,p_{k+1}\} \in U((p_k,p_{k+1}) \in E)$ $k = 0,\ldots,\ell-1$, $x_i \neq x_j$.

A simple(directed) cycle $\eta$ of length $\ell$ is an ordered sequence of distinct vertices: $\eta = \langle p_o,\ldots,p_\ell \rangle$ such that $p_o = p_\ell = \bar{x}$, $\{p_k,p_{k+1}\} \in U((p_k,p_{k+1}) \in E)$ $k = 0,\ldots,\ell-1$. The section (di)graph defined on a subset $Y \subset X$ is $G(Y) = (Y,U(Y))$ ( $= (Y,E(Y))$ where $U(Y) = \{\{x_i,x_j\} \in U | x_i,x_j \in Y\}$ $(E(Y) = \{(x_i,x_j) \in E | x_i,x_j \in Y)$

Given a digraph $G = (X,E)$, the reversion of an arc $(x_i,x_j)$ is performed by replacing it with an edge $(x_j,x_i)$.

Let G be a directed graph. G is said to be strongly connected if for each pair of vectices $x_i,x_j \in X$, there exist a simple path $\mu_1(x_i,x_j)$ and a simple path $\mu_2(x_j,x_i)$. It has to be noted that the trivial graph constituted by one node only is considered to be strongly connected. Let $\pi = \{X_1,\ldots,X_q\}$ be a partition of the nodes X. If the section graphs $G_i = (X_i,E_i) = G(X_i)$, $i = 1,\ldots,q$, are strongly connected and if no $G_i$ is a proper subgraph of a strongly connected subgraph of G, then the $G_i$'s are called the strongly connected components of G.

A bipartite (di)graph $B = (S,T,U)$ $(B = (S,T,E))$ is a (di)graph $B = (X,U)$ $(B = (X,E))$ such that $S \cup T = X$, $S \cap T = \emptyset$ and the section (di)graphs B(S) and B(T) are both vertex graphs. Given a bipartite (di)graph $B = (S,T,U)$ $(B, = (S,T,E))$ a matching I is a set of edges such that no two edges in I are incident to the same node. A node $x \in X$

is said covered if there is an edge in I that incides in it. A complete matching is a matching such that all the nodes of the bipartite (di)graph are covered. A maximum cardinality matching is a matching containing a maximum number of edges. Given a bipartite graph and a matching I, an alternating simple path $\lambda(x_i,x_j)$ is a simple path such that if the edges $\{p_k,p_{k+1}\}$ with k even are in I, the edges with k odd are not in I or vice versa. An alternating simple cycle $\rho$ is an alternating simple path $\lambda(x_i,x_j)$ in which $x_i = x_j$.

A bipartite graph can be conveniently used in order to code the zero-nonzero structure of a matrix. In particular, given a matrix $A = [a_{ij}] \in \mathbb{R}^{n^2}$ a bipartite graph B[A] can be associated to A as follows: $B[A] = (S,T,U)$, with $|S| = |T| = n$, and $\{s_i,t_j\} \in U$ iff (if and only if) $a_{ij} \neq 0$; $i, j = 1,\ldots,n$.

The structural rank of A, rs(A), is given by

(1) $$rs(A) = \max_{A_i \in \mathcal{A}} \text{rank } A_i$$

where

(2) $$\mathcal{A} = \{A_i | A_i \in \mathbb{R}^{n^2} \wedge (B[A_i] = B[A])$$

Then, the following Lemmas can be stated without proof [25,26].

Lemma 2.1. Given a matrix $A \in \mathbb{R}^{n^2}$, $rs(A) = |\bar{I}|$, where $\bar{I}$ is a maximum cardinality matching in B[A].

Lemma 2.2. A system $Ax = b$, $A \in \mathbb{R}^{n^2}$, $x,b \in \mathbb{R}^n$ has a solution only if I of Lemma 2.1 is a complete matching in B[A].

## III. ORDERING STRATEGIES AND BIPARTITE GRAPHS

When G.E. is performed on a given nonsingular matrix $A \in \mathbb{R}^{n^2}$, the matrix is modified step by step during the elimination procedure. Not only the numerical values of its elements are modified but also its structure. In particular, new non zero elements can be added. If the first pivot is taken in position $h_1, k_1 (a_{h_1k_1} \neq 0)$, the modified matrix $A^{(1)} = [a_{ij}^{(1)}]$ is obtained from A as follows:

$$(3) \quad a_{ij}^{(1)} = \begin{cases} a_{ij} & i = h_1 \quad \text{(a)} \\ 0 & j = k_1,\ i \neq h_1 \quad \text{(b)} \\ a_{ij} - (a_{ik_1} / a_{h_1k_1})\, a_{h_1j} & \text{otherwise} \quad \text{(c)} \end{cases}$$

Recursively, if the $\ell$-th pivot is taken in position $h_\ell$, $k_\ell$ $(a_{h_\ell k_\ell}^{(\ell-1)} \neq 0$, $h_\ell \neq h_m$, $k_\ell \neq k_m$, $m = 1,\ldots,\ell-1)$, $A^{(\ell)}$ is obtained from $A^{(\ell-1)}$ as follows:

$$(4) \quad a_{ij}^{(\ell)} = \begin{cases} a_{ij}^{(\ell-1)} & i = h_1,\ldots,h_\ell,\ j = k_1,\ldots,k_{\ell-1} \quad \text{(a)} \\ 0 & j = k_\ell,\ i \neq h_1,\ldots,h_\ell \quad \text{(b)} \\ a_{ij}^{(\ell-1)} - (a_{ik_\ell}^{(\ell-1)} / a_{h_\ell k_\ell}^{(\ell-1)})\, a_{h_\ell j}^{(\ell-1)} & \text{otherwise} \quad \text{(c)} \end{cases}$$

The fill-ins introduced during the $\ell$-th elimination step are

$$(5) \quad F^{h_\ell k_\ell} = \{a_{ij}^{(\ell)} \mid a_{ij}^{(\ell)} \neq 0,\ \ a_{ij}^{(\ell-1)} = 0\}$$

and may occur only in the submatrix of $A^{(\ell)}$, $A_{h_\ell k_\ell}^{(\ell)} \in \mathbb{R}^{(n-\ell)^2}$ defined by 4c. In particular, it has to be noted that $A_{h_\ell k_\ell}^{(\ell)}$ can be obtained from $A_{h_{\ell-1}k_{\ell-1}}^{(\ell-1)}$, deleting row $h_\ell$ and column $k_\ell$, and modifying the other elements according to 4c. When $\ell = n$, the elimination process terminates.

The set of indices $\mathcal{J} = \{(h_1,k_1), (h_2,k_2)\ldots, (h_n,k_n)\}$ individuates the pivot strategy.

If a minimum fill-in policy is followed, it has to be chosen a pivot strategy $\bar{\mathcal{J}}$ such that

$$F(\bar{\mathcal{J}}) = \min_{\mathcal{J}} F(\mathcal{J}) = \min_{\mathcal{J}} \Big| \bigcup_{(h_\ell,k_\ell)\in\mathcal{J}} F^{h_\ell k_\ell} \Big|.$$

Now, a graph theoretic interpretation of the elimination process on A is proposed.

<u>Definition 3.1</u>. Given a bipartite graph $B = (S,T,U)$ a <u>dumb bell</u> is a couple of nodes $d = [s,t]$ such that $s \in S$, $t \in T$ and $\{s,t\} \in U$.

<u>Definition 3.2</u>. Given a bipartite graph $B = (S,T,U)$ and a dumb bell $d_{hk} = [s_h,t_k]$,

(a) the <u>deletion</u> of $d_{hk}$ from B is accomplished removing $s_h$ and $t_k$ with their incident edges. The obtained graph is then $B(X-\{\{s_h\}\cup\{t_k\}\})$

(b) the <u>elimination</u> of $d_{hk}$ is accomplished deleting $d_{hk}$ and adding a set of new edges $U_{hk} = \{\{s_i,t_j\} \mid \{s_i,t_j\} \notin U \wedge \{s_i,t_k\} \in U \wedge \{s_h,t_j\} \in U\}$ $|U_{hk}|$ is called the deficiency of $d_{hk}$, $\tau(d_{hk})$.

<u>Theorem 3.1</u>. Given a matrix $A \in \mathbb{R}^{n^2}$ and a pivot sequence individuated by $\mathcal{J} = \{(h_1,k_1),\ldots,(h_n,k_n)\}$, $B[A^{(n)}_{h_n k_n}] = (\ldots((B[A])_{h_1k_1})_{h_2k_2}\cdots)_{h_nk_n}$ and $F(\mathcal{J}) = \sum_{j=1}^{n} \tau(d_{h_j k_j})$.

<u>Proof</u>. For sake of simplicity, we set $B^o = B[A] = (S^o,T^o,U^o)$ and $B^\ell = B[A^{(\ell)}_{h_\ell k_\ell}] = (S^\ell,T^\ell,U^\ell)$ $\ell = 1,\ldots,n$. At first we prove that $B^1 = (B^o)_{h_1k_1}$ and that $|F^{h_1k_1}| = \tau(d_{h_1k_1})$. $A^{(1)}_{h_1k_1}$ is obtained from A deleting row $h_1$ and column $k_1$. In $B^1$ the nodes $s_{h_1}$ and $t_{k_1}$, corresponding, by definition of bipartite graph associated to a matrix, to row $h_1$ and $k_1$, have been

removed. The condition $a_{h_1k_1} \neq 0$ is correspondent to the condition that $[s_{h_1}, t_{k_1}]$ is a dumb bell in $B^o$. The structure of A is possibly modified adding new elements according to 4c. By (6), fill-ins occur if $a_{ij} = 0$ and $a_{ik_1}$ and $a_{h_1j} \neq 0$, a new edge is introduced in $B^1$ if $\{s_i, t_j\} \notin U$, $\{s_i, t_{k_1}\} \in U$ and $\{s_{h_1}, t_j\} \in U$. Then $|F^{h_1k_1}| = \tau(d_{h_1k_1})$. By the same reasonings it is possible to conclude that $B^\ell = (B^{\ell-1})_{h_\ell k_\ell}$ and that $|F^{h_\ell k_\ell}| = \tau(d_{h_\ell k_\ell})$, $\ell = 2,\ldots,n$. By recursion we obtain the proof of the

¤

Then a pivot strategy $\mathcal{J}$ individuates a sequence of dumb-bells $\mathcal{D} = \langle d_1,\ldots,d_n\rangle, d_j = d_{h_jk_j}$, such that

(7) $d_{h_jk_j} \in U^{j-1}$

The converse is also true: a sequence of dumb-bells $\mathcal{D} = \langle d_1,\ldots,d_n\rangle$ such that (7) holds individuates a pivot strategy. In fact, (7) implies $a^{(j-1)}_{h_jk_j} \neq 0$ and $h_j \neq h_i$, $k_j \neq k_i$, $i \neq j$, $i,j = 1,\ldots,n$.

The successive elimination of the dumb-bells in a sequence $\mathcal{D}^\ell = \langle d_1,\ldots,d_\ell\rangle$ is called <u>dumb bell elimination process</u> and $\mathcal{D}^\ell$ is a <u>dumb bell elimination sequence</u>. An elimination process with $\ell = n$ is said to be <u>complete</u>. Therefore, the minimum fill-in pivot strategy has the following graph theoretic interpretation: determine a complete dumb-bell elimination process in B[A], individuated by a dumb bell sequence $\bar{\mathcal{D}}$, such that

(8) $\tau(\bar{\mathcal{D}}) = \min_{\mathcal{D}} \tau(\mathcal{D})$

<u>Proposition 3.1</u>. Let $B = (S,T,U)$ be a bipartite graph and $\mathcal{D}$ a sequence of $\ell$ dumb bells satisfying (7). $\mathcal{D}$ individuates a matching in $\tilde{B} = (S,T,\tilde{U})$ where $\tilde{U} = U \cup (\bigcup_{d_{hk}\in\mathcal{D}} U_{hk})$

<u>Proposition 3.2</u>. A matching $\bar{I} = \{\{s_{h_1},t_h\},\ldots,\{s_{h_\ell},t_{k_\ell}\}\}$ in $B = (S,T,U)$ individuates a dumb bell set $D_{\bar{I}} = \{(s_{h_1},t_{h_1}),\ldots,(s_{h_e},t_{k_e})\}$ such that $(s_{h_j},t_{k_j})$, $j = 1,\ldots,\ell$, satisfy (7).

It has to be noted that there are no computational feasible algorithms able to find an elimination sequence $\bar{\Phi}$ such that (8) holds. In general [6,7,12-21], the problem is addressed via heuristic algorithms which try to obtain local optimum with a sequential deterministic procedure or step-by-step strategy. The fundamental rules on which the sequential procedures are based can be considered: select the dumb bell $d_{h_i k_i}$ in $B^{i-1}$ at the i-th stage, such that:

(i) $\tau(d_{h_i k_i})$ is minimum (local minimum fill-in strategy) [18,19]

(ii) the product of the number of edges incident to $s_{h_i}$ and $t_{k_i}$ is minimum (Markowitz criterion) [17,18]

(iii) the number of edges incident to $s_{h_i}$ is minimum and among all the dumb bells with this property select one with $t_{k_i}$ of minimum degree [18].

Almost all the available heuristic algorithms are based on these rules, or on combinations and slight modifications of them. However, as pointed out in Section I the pivot elements have to satisfy the following condition:

(9) $a^{(\ell-1)}_{h_\ell k_\ell} \Rightarrow a_{h_\ell k_\ell} \neq 0$

in order to assure the optimality of Gaussian Elimination and LU decomposition in the Brayton's sense [22].

The heuristic algorithms are in general not able to fulfill the condition (9) as shown in Fig. 1 for Markovitz criterion. Sometimes the general structure of the heuristic algorithms have been modified as follows [27]:

STEP 0. $i = 0\ B^o = B[A] = (S,T,U)$

STEP 1. Let $D_1^*$ be the set of dumb-bells in $B^i$ such that the correspondent edges $\in U$. If $D_i^*$ is void, $D_i^*$ is set equal to the set of all the dumb-bells in $B^i$. Select a dumb-bell $d_i^* \in D_i^*$ according to the chosen heuristic rule, $i = i+1$. If $i = n+1$, STOP. Otherwise continue.

STEP 2. Obtain $B^i$ from $B^{i-1}$ performing the elimination of $d_{i-1}^*$. Go to STEP 1.

However, according to STEP 1, sometimes $D_i^*$ is void and then it is necessary to take into account the elements not satisfying (9) in order to complete the elimination process. An example is shown in Fig. 2, where the Markovitz criterion requires the selection of the dumb-bells $[s_4,t_4]$, $[s_5,t_6]$ at the beginning of the elimination procedure. All the pivotal orderings after these steps require the choice of a fill-in as pivotal element.

## IV. THE ALGORITHM NONZERO

Before proving the fundamental theorem on which the algorithm for the selection of a pivotal strategy satisfying (9) is based, we point out the following Remark:

Remark 1. In view of Proposition 3.1 and 3.2, given a complete dumb bell elimination sequence $\mathcal{D}$ in B[A], condition (9) is satisfied iff the dumb-bells in $\mathcal{D}$ individuate a complete matching in B[A].

Remark 2. In order to obtain a complete elimination process satisfying (9) we have to devise an algorithm which is able to select sequentially a complete matching in B[A] without loosing the capability of maintaining the sparsity of A.

The basic idea is,

(1) to begin with a complete matching in B[A],

(2) to consider as possible pivot elements the edges which are in I or which can be inserted in a complete matching

(3) if an element not in I has been chosen according to the heuristic rules adopted, to modify I in order to insert the new element.

The set of elements which can be inserted at a certain step of the selection procedure is identified by the following Proposition which can be considered a consequence of a theorem in [28 pg. 123]

Proposition 1.1. Given a bipartite graph $B = (S,T,U)$ and a complete bipartite matching $I_1$, any other possible bipartite complete matching in B, $I_h$, can be obtained from $I_1$ as follows: individuate one or more disjoint simple alternating cycles (e.g., m) $\rho^j$ w.r.t. $I_1$. Let $I_1'$ be the set of edges in $I_1$ but not incident in the nodes of the alternating cycles, and $\bar{I}$ the set of edges belonging to the alternating cycles but not in $I_1$. Then $I_h = I_1' \cup \bar{I}$. ¤

Suppose now to have a complete matching I in B and direct the edges in B as follows:

$$(10) \qquad \{s_i,t_j\} \in U \begin{cases} \nearrow (t_j,s_i) \in E \text{ iff } \{s_i,t_j\} \in I \\ \searrow (s_i,t_j) \text{ otherwise} \end{cases}$$

Let $\bar{B}_I = (S,T,E)$ be the obtained directed bipartite graph.

Proposition 4.2. There is a one-to-one correspondence between simple directed cycles in $\bar{B}_I$ and simple alternating cycles in B w.r.t. I. ¤

Now it is possible to prove the fundamental theorem.

Theorem 4.1. Let $B = (S,T,U)$ be a bipartite graph and I a complete matching in it. Let $\{s_i,t_j\}$ be an edge not in I. There exists a complete matching I' in B such that $\{s_i,t_j\} \in I'$ iff $(s_i,t_j)$ belongs to a strongly connected component of $\bar{B}_I = (S,T,E)$.

Proof. If part: by definition of strongly connected components there exists a simple path $\mu(t_j,s_i)$ in $\bar{B}_I$. Then $\{s_i,\mu(t_j,s_i)\}$ is a simple cycle in $\bar{B}_I$. By Propositions 4.1 and 4.2, it is possible to obtain a complete matching I' such that $\{s_i,t_j\} \in I'$.

Only if part: If there exists a complete matching I' such that $\{s_i,t_j\} \in I'$, then by Proposition 4.1, there exists an alternating cycle containing $\{s_i,t_j\}$. By Proposition 4.2, there exists in $\bar{B}_I$ a simple directed cycle such that $\exists k$, $p_k = s_i$, $p_{k+1} = t_j$. By definition of strongly connected component, there exists a strongly connected component of $\bar{B}_I$ such that $(s_i,t_j)$ is present in it. ¤

Now, we are able to build up an algorithm for the selection of an elimination process satisfying (9).

Assumption 1. $A \in \mathbb{R}^{n^2}$ is nonsingular

By Lemma 2 we know that there is at least one complete matching in B[A].

Assumption 2. A complete matching I in B[A] is given.

ALGORITHM NONZERO

STEP 0. Set $i = 0$, $B^o = B[A]$. $\bar{B}^o = \bar{B}_I$

STEP 1. Find the strongly connected components of $\bar{B}^i$.

STEP 2. Let $D_i = \{D'_i \cup D''_i\}$ where $D'_i$ is the set of dumb bells corresponding to the edges in the strongly connected components of $\bar{B}^i$ and $D''_i$ is the set of dumb bells corresponding to the edges directed from T to S not in the strongly connected components of $\bar{B}^i$. Select a dumb-bell $d_i \in D_i$ according to the chosen heuristic rule applied to $B^i$.

STEP 3. If $d_i \in D''_i$, go to STEP 6.

STEP 4. If the edge corresponding to $d_i$ is directed from T to S, go to STEP 6.

STEP 5. Let $d_i = [s_{h_i}, t_{k_i}]$. Find a path $\mu(t_{k_i}, s_{h_i})$ and accomplish the reversion of the arcs in the path.

STEP 6. $i = i+1$, if $i = n$, STOP

STEP 7. Delete $d_{i-1}$ from $\bar{B}^{i-1}$ to obtain $\bar{B}^i$. Eliminate $d_{i-1}$ from $B^{i-1}$ to obtain $B^i$. If $d_{i-1} \in D''_i$, go to STEP 2, otherwise go to STEP 1.

¤

The correcteness of the algorithm is proved in the following theorem.

<u>Theorem 4.2</u>. The dumb bells in the sequence $\mathcal{D} = \langle d_o, \ldots, d_{n-1} \rangle$ individuate a complete matching in B[A].

Proof. We prove the theorem proving that the following statement is true:

(S) for every $i = 0, \ldots, n$, there exists a complete matching in B[A] individuated by $d_o, \ldots, d_{i-1}$ and by the arcs directed from T to S in $\bar{B}^i$.

In fact, if the statement (S) is correct, setting $i = n$, the theorem is proved.

(S) will be proved by induction. For $i = 0$, by Assumptions 1 and 2 and by STEP 0, there exists a complete matching in B[A] individuated by the edges directed from T to S in $\bar{B}^o$. Suppose that (S) is true for $i = k$, (S) is true for $i = k+1$. In fact, at the k-th stage, there is a complete matching in B[A] and this is individuated by $d_o,\ldots,d_{k-1}$ and by the edges directed from T to S in $\bar{B}^k$ by hypothesis. Suppose that $d_k$ is selected in STEP 2. Then two cases occur: (1) the edge individuated by $d_k$ is directed from T to S. (2) the edge individuated by $d_k$ is directed from S to T.

In the first case, after STEP 7, the edges corresponding to the dumb-bells $d_o,\ldots,d_k$ and the edges in $\bar{B}^{k+1}$ directed from T to S obviously individuate a complete matching in B[A]. If $d_k \in D_k''$, then the strongly connected components of $\bar{B}^{k+1}$ are equal to the strongly connected components of $\bar{B}^k$ except the trivial ones. Therefore, there is no need to recompute them according to STEP 7.

In the second case, $d_k \in D_k'$ and it belongs to a strongly connected component of $\bar{B}^k$. Then, by Propositions 4.1, 4.2 and Theorem 4.1, at the k+1-th stage, after the execution of STEP 7, the edge corresponding to $d_o,\ldots,d_k$ and the edges directed from T to S in $\bar{B}^{k+1}$ individuate a complete matching in B[A]. ¤

The complexity of Algorithm nonzero is now discussed.[†]

[†]Recall that an algorithm has complexity $O(p^\alpha, q^\beta_\alpha)$ if the computation time and the storage requirements are bounded by $k_1 p^\alpha + k_2 q^\beta$ where $k_1$ and $k_2$ are constants. p and q are parameters depending upon the input of the algorithm [29].

It is immediate to observe that STEPS 1, 5 and 7 are dominant in complexity, so we concentrate our complexity analysis on these steps. STEP 1 can be implemented via Tarjan algorithm [29] or Gustavson algorithm [32]. Both of them are $O(|X|,|E|)$ if $|X|$ is the number of nodes of the considered digraph and $|E|$ is the number of its edges. The data structure used in [29] can be applied in Nonzero as well, while the data structure used in [32] has to be modified with the addition of new arrays. STEP 1 is executed in the worst possible case n-1 times on graphs of decreasing size. For this reason the complexity of STEP 1 is estimated to be $O(n^2, n\,\ell)$ where $\ell$ is the number of nonzero elements in A.

STEP 5 consists mainly in finding a directed path between two vertices. If a depth first search strategy is used on a directed graph stored as in [29], the complexity of STEP5 is $O(|X|,|E|)$. In the worst case this STEP is executed n times. Therefore the overall complexity is $O(n^2, n\,\ell)$.

In STEP 7 the leading term is given by the elimination of $d_{i-1}$. Because of Definition 3.2, in order to eliminate $d_{i-1} = [s_{i-1},t_{i-1}]$, a number of elementary operations proportional to the product of the number of edges incident in $s_{i-1}$ and the number of edges incident in $t_{i-1}$ is needed. Then, its complexity is $O(n(\tau+\ell))$.† However, since the elimination of $d_{i-1}$ is required because almost all the heuristic rules require to simulate the elimination of the selected pivot at each step, it is not necessary to obtain a nonzero pivot selection. Therefore its complexity will not be considered. It is now possible to claim that the complexity of Algorithm Nonzero is $O(n^2,n\ell)$.

---

†It has bo be noted that the algorithm described in [30] whose complexity is $O(n,\tau+\ell)$ does not work in this case because the complete elimination ordering is not known in advance.

Remark 3. The complexity of the selections rule is not taken into account (STEP 2). In fact, it depends on the particular heuristic rule followed.

Remark 4. If we relax Assumption 2, an algorithm developed in [31] can be implemented to compute a complete matching B[A]. Its complexity is $O(n^{0.5}\ell)$.

Remark 5. As already pointed out, the complexity of almost all the heuristic rules is $O(n(\tau+\ell))$. Then, Algorithm Nonzero does not increase significantly the complexity of an algorithm for the selection of a sub-optimal pivotal strategy.

## V. CONCLUDING REMARKS

In this paper, a bipartite graph has been used to code the nonzero structure of a sparse matrix. This representation has been shown to be well suited in order to investigate the problem of the choice of an optimal pivot ordering in Gaussian Elimination, when the pivot elements are not forced to be on the main diagonal. A graph theoretic interpretation of the Gaussian Elimination process as well as of the heuristic rules more frequently used has been proposed.

This graph representation has been used to solve the problem of the selection of pivot elements such that no fill-in is chosen. The problem was introduced in [22], as it was shown that Gaussian Elimination with fill-ins as pivot elements is not optimal in the sense that unnecessary operations as well as unnecessary storage requirements may be needed.

The main result of the paper is an algorithm able to solve the fill-in avoidance problem. Its correctness has been proved and its complexity has been shown to be $O(n^2, n\ell)$ where n is the dimension of the sparse matrix and $\ell$ is the number of nonzero in it. It has to noted that

(i) the algorithm can be used together with almost all the available heuristic rules for the selection of optimal pivot strategies

(ii) its complexity is such that it does not increase significantly the computation time and the storage requirement needed for the application of the heuristic rules alone.

As a final remark, it has to be pointed out that bipartite graphs may be the most promising tools for the study of optimization problems which involve the use of non symmetric permutations of a sparse matrix [33,34].

REFERENCES

[1] D.J. Rose and R.A. Willoughby Eds., Sparse Matrices and Their Applications. New York: Plenum, 1972.

[2] J.K. Reid Ed., Large Sparse Sets of Linear Equations. London: Academic Press, 1970.

[3] R.A. Willoughby Ed., Sparse Matrix Proceedings. IBM Watson Res. Cen., Yorktown Heights, N.Y., 1969.

[4] D.M. Himmemblau Ed., Decomposition of Large Scale Problems. North Holland Publ. Comp., 1973.

[5] R.D. Berry, An Optimal Ordering of Electronic Circuit Equations for a Sparse Matrix Solution, IEEE Trans. Circuit Theory, vol. CT-18, pp. 40-50, June 1971.

[6] G.D. Hachtel, R.K. Brayton, and F.G. Gustavson, The Sparse Tableau Approach to Network Analysis and Design, IEEE Trans. Circuit Theory, vol. CT-18, pp. 101-113, Jan. 1971.

[7] R.S. Norin and C. Pottle, Effective Ordering of Sparse Matrices Arising from Nonlinear Electrical Networks, IEEE Trans. Circuit Theory, vol. CT-18, pp. 139-145, Jan. 1971.

[8] E. Isaacson and H.B. Keller, Analysis of Numerical Methods, New York: J. Wiley & Sons, Inc., 1966.

[9] E.C. Ogbuobiri, W. Tinney, and J. Walker, Sparsity-Directed Decomposition for Gaussian Elimination on Matrices, IEEE Trans. Power App. Syst., vol. PAS-89, pp. 141-150,

[10] W.F. Tinney and J.W. Walker, Direct Solutions of Sparse Network Equations by Optimally Ordered Triangular Factorization, Proc. IEEE, vol. 55, pp. 1801-1809, Nov. 1967.

[11] D.J. Rose, Triangulated Graphs and the Elimination Process, J. Math. Anal. Appl., vol. 32, pp. 597-609, Dec. 1970.

[12] D.J. Rose, A Graph Theoretic Study of the Numerical Solution of Sparse Positive Definition Systems of Linear Equations in R. Read Ed., Graph Theory and Computing, New York: Academic Press, 1971.

[13] R.P. Tewarson, Sparse Matrices, New York: Academic Press, 1973.

[14] L. Haskins and D.J. Rose, Towards Characterization of Perfect Elimination Digraphs, SIAM J. Compt., vol. 2, pp. 217-224, Dec. 1973.

[15] T. Ohtsuki, L.K. Cheung, and T. Fujisawa, Minimal Triangulation of a Graph and Optimal Pivoting Order in a Sparse Matrix to appear in J. Math. Anal. Appl.

[16] T. Ohtsuki, A Graph-Theoretic Algorithm for Optimal Pivoting Order of Sparse Matrices, Proc. of the Sixth Hawaii International Conference on Systems Sciences, Second Supplement, pp. 45-48, Jan. 1973.

[17] H.M. Markovitz, The Elimination Form of the Inverse and its Application to Linear Programming, Man. Sci., vol. 3, n. 3, 1957.

[18] G.B. Dantzig, R.P. Harvey and R.D. McKnight, Sparse Matrix Techniques in Two Mathematical Programming Codes in [3].

[19] I. Shirikawa, S. Kyan, and H. Ozaki, Bipartite Network Associated with Optimal Pivoting Problem, Int. J. Cir. Theor. Appl., vol. 3, no. 1,1975.

[20] H. Hsieh, and M. Ghausi, On Optimal-Pivoting Algorithms in Sparse Matrices, IEEE Trans. Circuit Theory, CT-19, 1972.

[21] M. Nakhla, K. Singhal and J. Vlach, An Optimal Pivoting Order for the Solution of Sparse Systems of Equations, IEEE Trans. Circuits and Systems, CAS-21, no. 2, p. 222, 1974.

[22] R.K. Brayton, F.G. Gustavson, and R.A. Willoughby, Some Results on Sparse Matrices, Math. Comput., vol. 24, no. 112,

[23] A. Sangiovanni Vincentelli, A Graph Theoretic Approach to Nonsymmetric Permutations on Sparse Matrices, Int. Memo LCA, Politecnico di Milano, 1974, submitted for publication.

[24] F. Harary, Graph Theory, Reading, Mass., Addison-Wesley, 1969.

[25] D.V. Steward, On an Approach to Techniques for the Analysis of the Structure of Large Systems of Equations, SIAM Rev., vol. 4, pp. 321-342, 1962.

[26] A Sangiovanni Vincentelli, Bipartite Graphs and Nonsymmetric Permutations on Sparse Matrices, Proc. Third Int. Symp. on Network Theory, Split, 1975.

[27] F.G. Gustavson, private communication, 1975.

[28] C. Berge, Graphs and Hypergraphs, North-Holland, 1973.

[29] R. Tarjan, Depth First Search and Linear-Graph Algorithms, SIAM J. Comp., vol. 1, no. 2, pp. 146-160, 1972.

[30] D.J. Rose and R.E. Tarjan, Algorithm Aspects of Vertex Elimination, Proc. Seventh Annual ACM Symp. on Theory of Computing, 1975.

[31] J.E. Hopcroft and R.M. Karp, An $n^{5/2}$ Algorithm for Maximum Matchings in Bipartite Graphs, SIAM J. Compt., vol. 2, 1972.

[32] F. Gustavson, Finding the Block Lower Triangular Form of a Sparse Matrix, J. Bunch and D.J. Rose eds., Sparse Matrix Computations, Academic Press, to appear.

[33] A. Sangiovanni-Vincentelli, An Optimization Problem Arising from Tearing Methods, ibid.

[34] A. Sangiovanni-Vincentelli, A Graph Algorithm for the Optimal Tearing of a Sparse Matrix, Proc. 13th Allerton Conference on Circuit and System Theory, Monticello, Illinois, 1975.

FIGURE CAPTIONS

Fig. 1. An example of failure of Markowitz criterion in producing a set of pivot elements satisfying (9).

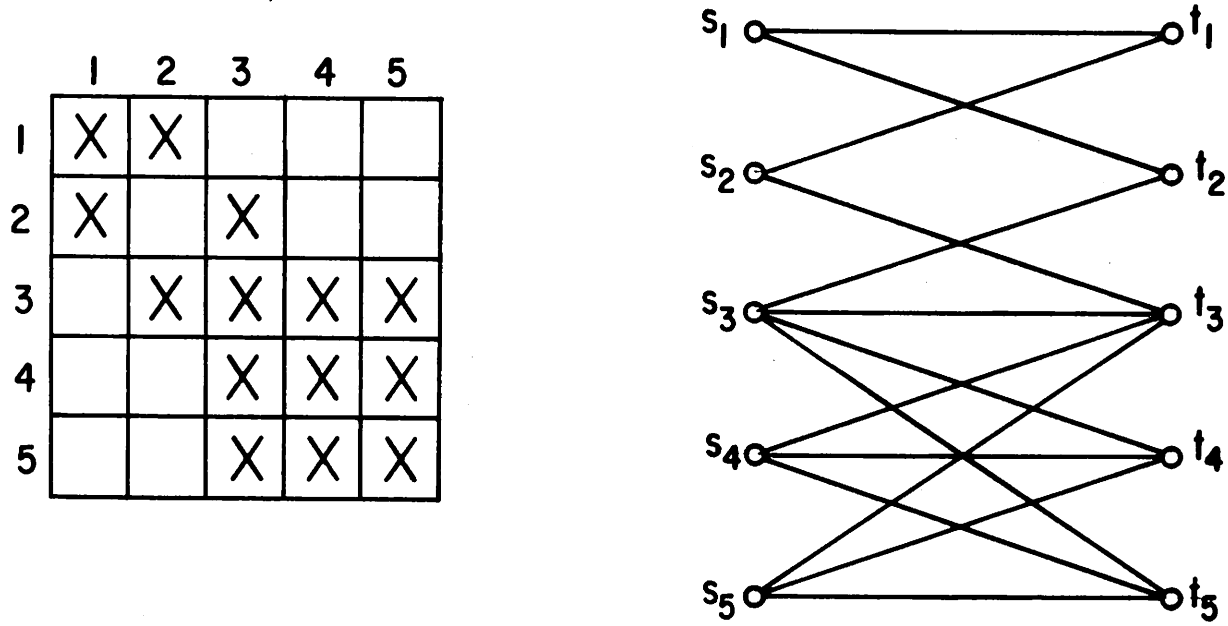
Fig. 1a. A matrix A and its associated bipartite graph B[A].

Fig. 1b. The bipartite graph $(B[A])_{s_1 t_1}$ obtained eliminating the dumb-bell $[s_1,t_1]$ selected by Markowitz criterion (---$\tau(d_{s_1 t_1})$).

Fig. 2. An example of failure of the modified Markowitz criterion in producing a set of pivot elements satisfying (9).

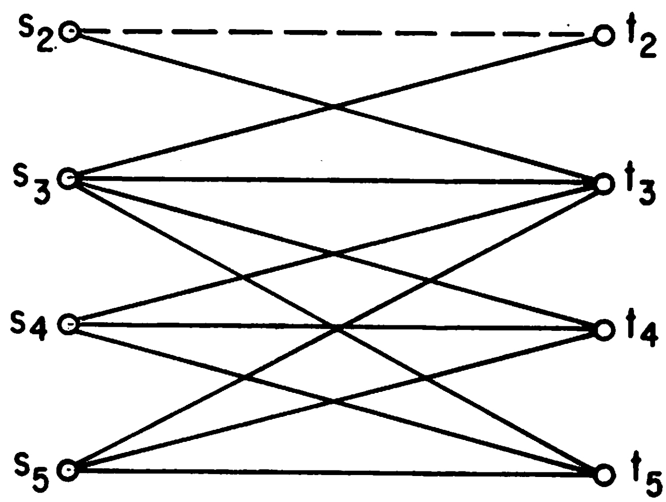
Fig. 2a. A matrix A and its associated bipartite graph B[A].

Fig. 2b. The bipartite graph $((B[A])_{s_4 t_4})_{s_5 t_6}$ obtained eliminating the dumb bells $[s_4,t_4]$ and $[s_5,t_6]$ selected by Markowitz criterion.



1 2 3 4 5
1 X X
2 X X
3 X X X X
4 X X X
5 X X X
$s_1$ $s_2$ $s_3$ $s_4$ $s_5$
$t_1$ $t_2$ $t_3$ $t_4$ $t_5$

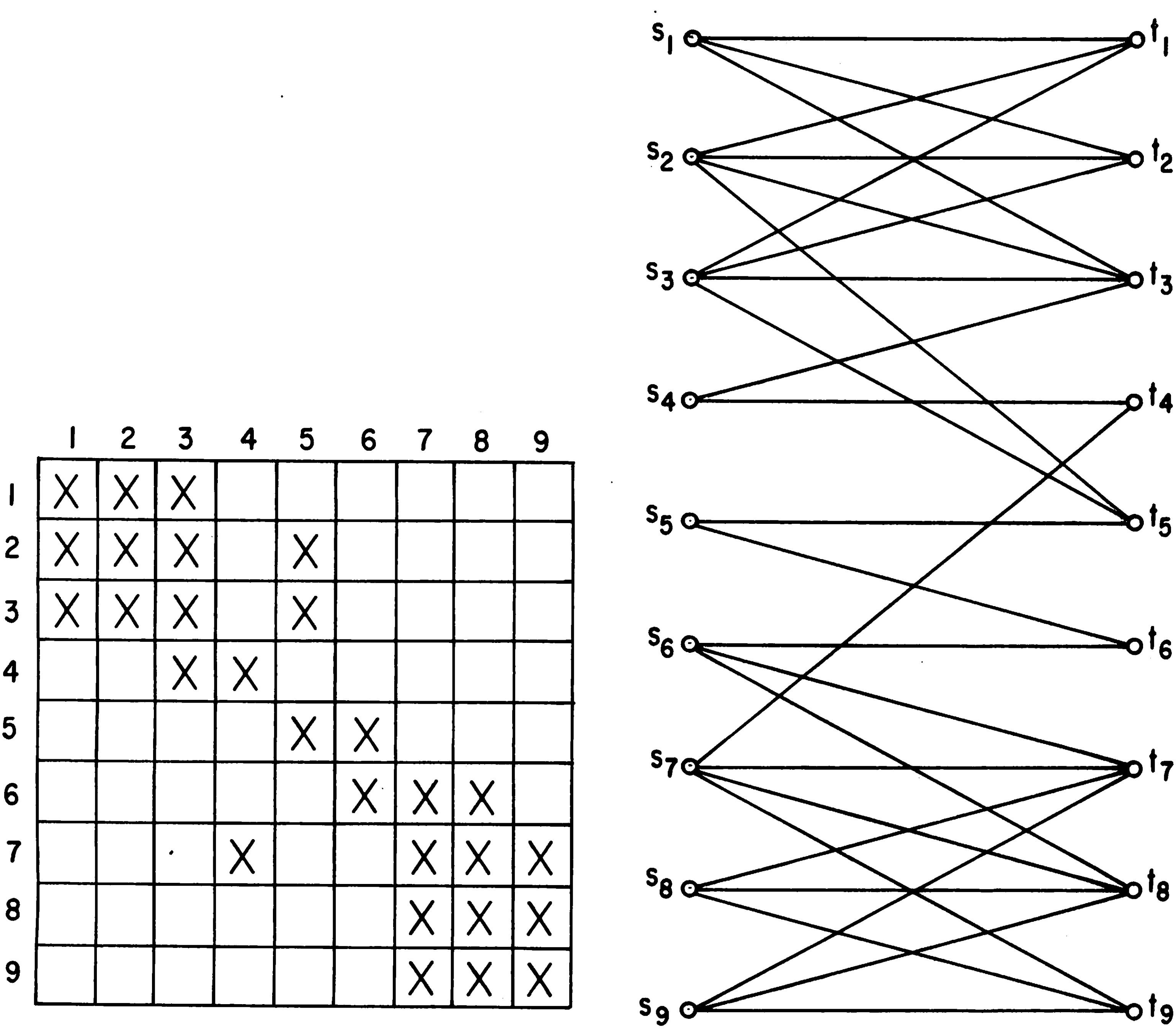

(a)



$s_2$ $s_3$ $s_4$ $s_5$
$t_2$ $t_3$ $t_4$ $t_5$


(b)

FIGURE 1


1 2 3 4 5 6 7 8 9
1 X X X
2 X X X X
3 X X X X
4 X X
5 X X
6 X X X
7 X X X X
8 X X X
9 X X X
$s_1$ $s_2$ $s_3$ $s_4$ $s_5$ $s_6$ $s_7$ $s_8$ $s_9$
$t_1$ $t_2$ $t_3$ $t_4$ $t_5$ $t_6$ $t_7$ $t_8$ $t_9$

(a)

FIGURE 2(a)

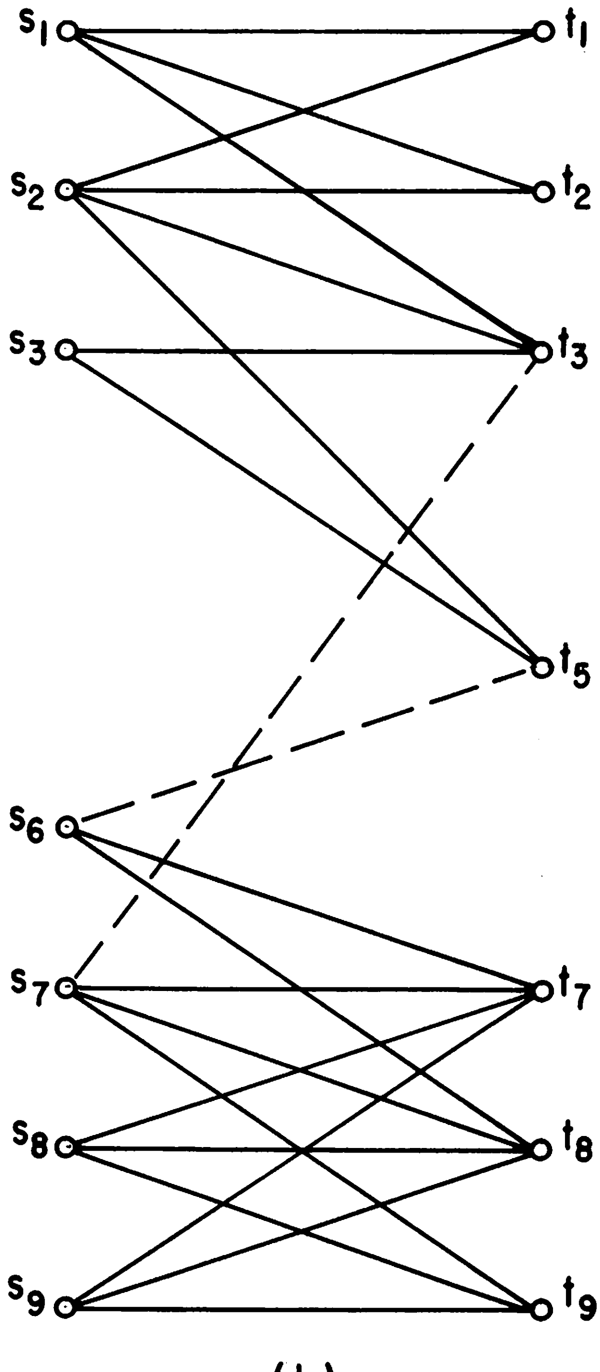



FIGURE 2(b)