On the Determination of Minimum Feedback Arc and Vertex Sets

Abstract—A method presented by Lempel and Cederbaum to find minimum feedback arc and vertex sets in directed graphs is shown to be partly equivalent to the well-known prime implicant problem of switching theory.

I. INTRODUCTION

Lempel and Cederbaum[1] have recently proposed an algorithm for the determination of minimum feedback arc and vertex sets of a directed graph. The algorithm consists of the following steps:

1) The arc cycles (the vertex cycles) of the given graph are determined by taking the permanent of the arc adjacency matrix (the vertex adjacency matrix). A Boolean expression, of the form “product of sums of literals,” listing the set of arc cycles (vertex cycles), is obtained in this step.

2) The Boolean expression is transformed in a “sum of products of literals.” The minimum feedback arc set (vertex set) is simply obtained by inspecting the expression, since it corresponds to the product with the minimum number of literals.

We will show in this correspondence that the subproblem of step 2) is identical to a problem encountered in the synthesis of minimum-cost two-level combinational switching networks, called the “prime implicant selection problem.” This identity has already been noted by one of the authors in a previous paper.[1]

In fact, the technique employed by Lempel and Cederbaum for solving step 2) is well known in the literature on switching theory as “Petrick’s method,” “Petrick’s presence function,” or the “Boolean algebra method.”[10] In addition, it has been recognized long ago that the transformation of the “product of sums” expression into a “sum of products” expression should not be tried on the initial expression, since it entails a great amount of unnecessary computational work. Instead, a more effective procedure consists of simplifying the initial expression by means of reduction rules. Repeated application of the reduction rules sometimes gives a direct solution, while in other cases a simpler “product of sums” expression is obtained, which is then transformed in “sum of products” form.[11]

II. FORMULATION OF THE PROBLEM

In this section, we will formulate in general terms a covering problem, and we will show that the minimum feedback arc problem and the minimum vertex set problem are of identical nature.

Let \( R = (r_1, r_2, \ldots, r_n) \) and \( C = \{ c_1, c_2, \ldots, c_m \} \) be two sets, and let us indicate with the symbol \( r \rightarrow c \) a relation holding between an element of \( R \) and an element of \( C \). If the relation holds between two sets \( R \) and \( C \), we will write:

\[ r_i \rightarrow c_l \]

and we will say that \( r_i \) covers \( c_l \). Assume that sets \( R \) and \( C \) are such that each \( c_l \in C \) is covered by at least one \( r_i \in R \). A solution of the covering problem is a subset \( R_a \subseteq R \) such that for every \( c_l \in C \) there is at least one \( r_i \in R_a \) which covers it. A minimal solution is a solution \( R_a \) such that, for any \( r_i \in R \), the set \( R_a - r_i \) is not a solution. A minimum solution is a solution of minimum cardinality.\(^1\)

Note that:

1) In the minimum feedback arc set problem, set \( C \) is the set of arc cycles (an arc cycle is the set of arcs contained in a directed circuit) of the given directed graph, while set \( R \) is the set of arcs of the graph. An arc \( r \) covers all arc cycles \( c_{i_1}, c_{i_2}, \ldots, c_{i_k} \) to which it belongs.

2) In the minimum feedback vertex set problem, set \( C \) is the set of vertex cycles (a vertex cycle is the set of vertices contained in a directed circuit) of the given directed graph, while set \( R \) is the set of vertices of the graph. A vertex \( r \) covers all vertex cycles \( c_{i_1}, c_{i_2}, \ldots, c_{i_k} \) to which it belongs.

3) In the prime implicant problem, set \( C \) is the set of minterms of the given switching function, while set \( R \) is the set of prime implicants of the function.\(^2\)

III. SOLVING THE COVERING PROBLEM

The direct application of Petrick’s method for solving a covering problem is straightforward. Let:

\[ r_1, r_2, \ldots, r_n \rightarrow c_1 \]
\[ r_1, r_2, \ldots, r_n \rightarrow c_2 \]
\[ \vdots \]
\[ r_1, r_2, \ldots, r_n \rightarrow c_m \]

For every \( r_i \) consider a binary variable \( \rho_i \), and let

\[ \rho_i = 1 \quad \text{if} \quad r_i \quad \text{is selected in the solution of the covering problem} \]
\[ \rho_i = 0 \quad \text{otherwise} \]

The condition that \( c_l \) be covered by at least one \( r_i \) \((1 \leq i \leq n)\) can be expressed by a Boolean sum as follows:

\[ \rho_{a_1} + \rho_{a_2} + \cdots + \rho_{a_n} = 1 \]

Analogous conditions can be written for the other elements \( c_{n+1}, \ldots, c_m \) of set \( C \):

\[ \rho_{b_1} + \rho_{b_2} + \cdots + \rho_{b_m} = 1 \]
\[ \vdots \]
\[ \rho_{c_1} + \rho_{c_2} + \cdots + \rho_{c_m} = 1 \]

Therefore, a solution must satisfy the Boolean condition (presence function)

\[ \pi = \left( \rho_{a_1} + \rho_{a_2} + \cdots + \rho_{a_n} \right) \left( \rho_{b_1} + \rho_{b_2} + \cdots + \rho_{b_m} \right) \cdots \left( \rho_{c_1} + \rho_{c_2} + \cdots + \rho_{c_m} \right) = 1 \]

All the combinations of the \( \rho_i \) variables such that \( \pi = 1 \) are a solution of the covering problem. All the solutions can be found by "multiplying out" the presence function in the sum-of-products form. In this form, every product term corresponds to one solution.

If the theorem

\[ \rho_1 \rho_2 + \rho_4 = \rho_4 \]

1 The terms employed in switching theory for “minimal” and “minimum” are “irredundant” and “minimal,” respectively. However, we have preferred here to retain the terms “minimal” and “minimum” employed by Lempel and Cederbaum.\(^\text{[1]}\)

is consistently applied during the multiplication process, the presence function in sum-of-products form lists the minimal solutions only. The minimum cover corresponds in this expression to the term having the minimum number of literals.

If the problem is less-than-trivial, however, the direct application of the Boolean algebra method, as suggested by Lempel and Cederbaum, brings about formidable computational difficulties. However, a lengthy computation is unnecessary in most cases, since there exist a number of reduction rules which allow a substantial simplification of the original problem. Often, the application of these rules yields directly a solution without recourse to the Boolean algebra method. The Boolean algebra method must be applied only if at some stage in the reduction process a solution has not yet been reached, and all the reduction rules fail to apply. At this stage, the covering problem is said to be cyclic.

In practice, it is cumbersome (although, in principle, possible) to apply the reduction rules to the algebraic expression. Instead, a table is employed (prime implicant table or covering table) which conveys the same information as the presence function. In this table (see examples in Section IV) the rows correspond to the elements of set $R$, and the columns to the elements of set $G$. The covering relation is displayed by crosses on this table: if $r_i \rightarrow g_j$, then a cross is placed at the intersection of row $r_i$ and column $g_j$.

In terms of the covering table, a solution is a set of rows such that for every column there is at least one row in the set having a cross in that column. Note that the covering problem can be solved solely by tabular techniques, without recourse to the Boolean algebra method. That is, after the reduction phase, column branching can be applied to a cyclic table to yield a solution.

A great deal of work has also gone into the problem of performing efficiently the multiplication process. See for example Pynne and McClusky.

There also exist different approaches to the solution of the covering problem. One [1] utilizes a technique known as "consensus taking." Finally, we will conclude by noting that the covering problem has been shown to be equivalent to an integer-restricted linear program.

---

**TABLE I**

**VERTEX CYCLES OF THE GRAPH IN FIG. 1**

<table>
<thead>
<tr>
<th>$c_1$</th>
<th>$c_2$</th>
<th>$c_3$</th>
<th>$c_4$</th>
<th>$c_5$</th>
<th>$c_6$</th>
<th>$c_7$</th>
</tr>
</thead>
<tbody>
<tr>
<td>${a}$</td>
<td>${a, b}$</td>
<td>${a, b, i}$</td>
<td>${a, b, h}$</td>
<td>${a, b, h, i}$</td>
<td>${a, b, h, f}$</td>
<td>${a, b, f}$</td>
</tr>
</tbody>
</table>

---

**TABLE II**

**ARC CYCLES FOR THE GRAPH IN FIG. 3**

![Graph Image]  

**Fig. 3.** Second example: determination of the minimum arc set.

![Matrix Image]  

**Fig. 4.** Covering table for the graph in Fig. 3.
IV. Examples

Consider the graph, Fig. 1. Its vertex cycles are listed in Table I and the covering table is Fig. 2(a). Note that vertices d, e, f, and g do not appear in this table, since they are not contained in any vertex cycle. This table can be reduced to Fig. 2(b) by noting that column 1 has only one cross (in row c). Therefore, row c (called an essential row) must be a member of every solution, and can therefore be selected at this stage. Row c can be erased from the table, together with all the columns in which it has crosses, since there will be no need in further selections to account for these columns, already covered by row c. Hence, by erasing row c and columns 1, 3, 5, 6, and 8, the reduced table, Fig. 2(b), is obtained. Observe in this table that row a has all the crosses of rows b, h, and i. In this example, it is obvious that rows b, h, and i cannot be members of a minimum solution; hence the minimum solution is the set of rows $B' = \{a, c\}$. In general, it can easily be shown that dominated rows can be, at any stage of the reduction process, eliminated from further consideration (and henceforth erased from the table). In fact, at least one minimum solution can always be reached by selecting among the remaining rows.

In the foregoing example, the solution has been directly obtained by very simple tabular operations. The reader can easily check the fact that the application of the Boolean algorithm would have led to (comparatively) long computation.

As a second example, consider the directed graph in Fig. 3. Its arcs are cycles listed in Table II, and the covering table is displayed in Fig. 4(a). In this covering table, row a and row c are dominated by row b; row c and row d can be eliminated from the table, and the covering table, Fig. 4(b) is obtained. In this table, consider columns 1 and 5. Column 1 has all the crosses of column 5, and it is said that column 1 dominates 5. It can easily be shown that dominating columns can be eliminated from the table, at any stage of the reduction process. In fact, if the dominated column is covered at least once, automatically the dominating column will also be covered. In our case, column 1 is eliminated. Also, note that columns 2, 3, and 4 have crosses in the same rows, and we can (arbitrarily) say that one of these columns dominates the other two. Likewise, columns 7 and 8 are equal. By eliminating columns 1, 3, 4, and 8 we obtain the covering table, Fig. 4(c). This table is cyclic, since the reducing rules of row essentiality, row dominance, and column dominance are not applicable. To this table corresponds the presence function

$$\pi = (b + f + k)(b + c)(c + h + 0)(f + h)(i + k),$$

which yields

$$\pi = bih + bhk + bif + cij + cfk + ck.$$  

In this case, every product in the sum-of-products expression corresponds to a minimum solution.

In the second example, it is also clear that the application of the Boolean algorithm method to the unreduced problem would have been clumsy.

L. Diveriti
Istituto di Elettrotecnica ed Elettronica
Politecnico di Milano
Milan, Italy

A. Grasselli
Centro Studi Calcolatrici Elettroniche del CNR
Università di Pisa
Pisa, Italy

References


Authors' Reply

Divietri and Grasselli rightly point out that finding the minimum feedback arc or vertex sets for a given graph may at least partially be presented as covering problems [see Section II, 1) and 2) preceding].

The essential difference, however, between these graph-theoretical problems and the case of simplification of a prime implicat table is that for a given graph the covering relations between the set $R$ of graph elements and the cycle set $C$ are not known a priori.

The known method of finding all the cycles for a given graph is that of the permanent expansion of the arc or vertex adjacency matrix (properties 1 and 2 of our paper). Thus the expansion of the permanent turns out to be a vital part of the procedure which cannot be disposed of.

This point on, the procedure of the paper might be compared with that proposed by Divietri and Grasselli. It needs to be pointed out that the latter procedure does not necessarily yield all the possible minimum feedback sets. Some of them might be eliminated during the simplification procedure. For instance, in the second example of Section IV, the arc set $\{e, h, k\}$ does not appear in the presented solution. Consequently, if one is interested in all the possible solutions, Petrick's method is applicable. Only in the case where more than one solution is possible and any one of them is acceptable would the application of Petrick's method prove more convenient.

In any case, in order to simplify the procedure, certain operations on the given graph have been recommended in our paper. These operations amount to the following.

1) Localize all the self-loops. They belong to every feedback arc set and their vertices to every feedback vertex set. Removing of these elements leaves a simplified subgraph for further examination.

2) Remove all the directed cut-sets and the vertices which may then remain isolated. These elements cannot take part in any feedback sets at all.

These operations have their equivalents in Petrick's method. Applying them, however, preserves the whole set of existing solutions.

A. Lempel
I. Cederbaum
Faculty of Elect. Engrgr.
Technion—Israel Institute of Technology
Haifa, Israel

1 Manuscript received November, 1962.