

T H E U N I V E R S I T Y O F M I C H I G A N

I N F O R M A T I O N S Y S T E M S L A B O R A T O R Y

Department of Electrical Engineering  
College of Engineering

Technical Report ISL-65-3

C O V E R I N G P R O B L E M S : D U A L I T Y R E L A T I O N S

A N D A N E W M E T H O D O F S O L U T I O N

by

Eugene L. Lawler

May, 1965

This research was supported in part by NSF Grant GP-2778,  
and in part by Air Force Contract AF 30(602)-3546.



## Table of Contents

<u>Section</u>		<u>Page</u>
	Abstract	1
1	Introduction	2
2	Definitions and Elementary Results	4
3	Dualization Properties of Cover Mapping	
4	Computations with Minimal Sets; Enumeration of Minimal Covers	11
5	Computation of Optimal Covers: Equal Costs	14
6	Computation of Optimal Covers: Unequal Costs	19
7	A Refinement	24
8	Comparison with Other Methods of Solution	28
	References	30



## Abstract

This paper is concerned with "covering" problems of the following type: Let a strictly positive  $n$ -vector  $c$  and an  $m \times n$  matrix  $A$  of 0's and 1's be given. Let  $\epsilon = (1, 1, \dots, 1)$  be an  $m$ -vector of 1's. Find a solution vector  $y = (y_1, y_2, \dots, y_n)$  which will minimize the inner product  $cy$  subject to  $Ay \geq \epsilon$  and  $y_j = 0$  or  $1$  ( $j=1, 2, \dots, n$ ).

There is an interesting duality between the constraints of covering problems and their solutions. This duality is explored in this paper and is found to furnish the basis for a possible new method of solution which is described in detail.

## 1. Introduction

Let a strictly positive  $n$ -vector  $c$  (the cost vector) and an  $m \times n$  matrix  $A$  of 0's and 1's (the constraint matrix) be given. Let  $\epsilon = (1, 1, \dots, 1)$  be an  $m$ -vector of 1's. A covering problem is formulated as follows: Find a solution vector  $y = (y_1, y_2, \dots, y_n)$  which will

$$\text{minimize } cy = \sum_{j=1}^n c_j y_j$$

subject to

$$Ay \geq \epsilon$$

and

$$y_j = 0 \text{ or } 1 \quad (j=1, 2, \dots, n).$$

Problems of this type arise in a variety of combinatorial contexts, e.g. in switching theory and in operations research. The following interpretation may be suggestive. Suppose a set  $S = \{s_1, s_2, \dots, s_m\}$  of  $m$  elements and an arbitrary collection of  $n$  subsets are given. If  $c = (1, 1, \dots, 1)$ , and  $A = \|\| a_{ij} \|\|$  is defined by the rule

$$\begin{aligned} a_{ij} &= 1 && \text{if } s_i \in S_j \\ &= 0 && \text{otherwise,} \end{aligned}$$

then an optimal solution to the covering problem indicates the smallest collection of subsets that can be chosen, subject to the restriction that each element of  $S$  is a member of at least one of the chosen subsets. This is under the interpretation that

$$\begin{aligned} y_j &= 1 && \text{if } S_j \text{ is chosen} \\ &= 0 && \text{otherwise.} \end{aligned}$$

There is an interesting duality between the constraints of covering problems and their solutions. This duality is explored in this paper, and is found to furnish the basis for a possible new method of solution, which is described in detail. The solution method is iterative in nature, and requires a number of iterations no greater than the cost of an optimal solution. The amount of computation required at each iteration is variable, and may be impracticably large in some cases.

Some conventions adopted in this paper are as follows. With the exception of cost vectors, all vectors and matrices are assumed to be composed of 0's and 1's. The vector  $\epsilon$  is composed entirely of 1's, and is of whatever dimension is appropriate in context. The vector  $\epsilon^k$  is a vector with 1 in the  $k^{\text{th}}$  position and 0's elsewhere:

$$\begin{aligned} (\epsilon^k)_j &= 0 & j \neq k \\ &= 1 & j = k. \end{aligned}$$

The complement of  $\epsilon^k$ , denoted  $\bar{\epsilon}^k$ , is the vector with 0 in the  $k^{\text{th}}$  component and 1's elsewhere.

The inner product of the two  $n$ -vectors  $c$  and  $y$  is denoted  $cy$ :

$$cy = \sum_{j=1}^n c_j y_j.$$

Row  $i$  of matrix  $A$  is denoted  $A_{i.}$  and column  $j$  is denoted  $A_{.j}$ .

The vector  $\underline{a}$  is said to be greater than or equal to the vector  $b$  if and only if  $\underline{a}$  is greater than or equal to  $b$ , component by component:

$$\underline{a} \geq b \iff a_j \geq b_j \quad (j=1,2,\dots,n).$$

The number of elements contained in set  $S$  is given by  $|S|$ .

## 2. Definitions and Elementary Results

A covering problem is unaffected by a reordering of the rows of the constraint matrix  $A$ . Therefore, it is appropriate to conceive of  $A$  as an unordered set of  $n$ -vectors:

$$A = \{a^{(1)}, a^{(2)}, \dots, a^{(m)}\}.$$

We shall adopt this convention for most of the remainder of this paper.

If  $A$  is a set of  $n$ -vectors, a cover of  $A$  is a  $(0,1)$ -vector  $y$  which has a nonzero scalar product with each vector in  $A$ :

$$\begin{aligned} a^{(i)}y &\geq 1 & (i=1,2,\dots,m) \\ y_j &= 0 \text{ or } 1 & (j=1,2,\dots,n). \end{aligned}$$

A cover  $y$  is minimal iff there exists no distinct cover  $y'$  such that  $y' \leq y$ . A cover  $y$  is optimal iff there exists no distinct cover  $y'$  such that  $cy' < cy$ .

Every optimal cover is a minimal cover, and every minimal cover is an optimal cover with respect to some cost vector  $c$ .

We let  $\text{Cov } A$  denote the set of covers of  $A$ . For example, if

$$A = \{(0,1,0,1), (0,1,1,1), (1,0,1,0), (1,1,0,0)\} \quad (2.1)$$

then

$$\begin{aligned} \text{Cov } A = \{ &(0,1,1,0), (0,1,1,1), (1,0,0,1), (1,0,1,1), \\ &(1,1,0,0), (1,1,0,1), (1,1,1,1), (1,1,1,0)\}. \end{aligned}$$

We let the subset of minimal vectors of a set  $S$  be denoted  $\text{Min } S$ . Thus the set of minimal covers of  $A$  is denoted  $\text{Min Cov } A$ . For  $A$  in (2.1),

$$\text{Min Cov } A = \{(0,1,1,0), (1,0,0,1), (1,1,0,0)\}.$$

It is easily shown that the minimal elements of  $A$  determine the covers of  $A$ . Thus, for all  $A$



$$\text{Cov } A = \text{Cov } \text{Min } A. \quad (2.2)$$

Cov A has the property that if  $y$  is contained in Cov A, and  $y' \geq y$ , then  $y'$  is also contained in Cov A. Any set which has this property will be said to be closed. We define the closure of A, denoted Cl A, to be the smallest closed set in which A is included. Or, equivalently,

$$\text{Cl } A = \{a' \mid a' \geq a, \text{ for some } a \in A\},$$

where  $a'$  is required to be a (0,1)-vector. For example, for A in (2.1),

$$\begin{aligned} \text{Cl } A = \{ & (0,1,0,1), (0,1,1,1), (1,0,1,0), (1,0,1,1) \\ & (1,1,0,0), (1,1,0,1), (1,1,1,0), (1,1,1,1) \}. \end{aligned}$$

We have already observed that, for all A

$$\text{Cov } A = \text{Cl } \text{Cov } A. \quad (2.3)$$

To this we can add the identity

$$\text{Cov } A = \text{Cov } \text{Cl } A, \quad (2.4)$$

and also

$$\text{Cl } A = \text{Cl } \text{Min } A. \quad (2.5)$$

The closure operation is consistent with the topological notion of closure, in that

$$\text{Cl } \varnothing = \varnothing \quad (\varnothing \text{ the empty set})$$

$$A \subseteq \text{Cl } A$$

$$\text{Cl } \text{Cl } A = \text{Cl } A$$

$$\text{Cl}(A \cup B) = \text{Cl } A \cup \text{Cl } B. \quad (2.6)$$

Finally, note that every closed set forms a semilattice with respect to the l.u.b. (least upper bound) operation " $\vee$ ", defined as follows:

$$(a \vee b)_j = \text{Max } \{a_j, b_j\}.$$

(A semilattice satisfies the associative, commutative, and idempotent laws

with respect to its single operation. If finite, it has a unique largest element with respect to the partial ordering induced under the definition

$$\underline{a} \leq \underline{b} \quad \text{iff} \quad a \vee b = b.$$

In the case at hand,

$$\underline{a} \leq \underline{b} \quad \text{iff} \quad a_j \leq b_j \quad \text{for } j=1,2,\dots,n.)$$

If vectors are represented by computer words, with a single bit devoted to each vector component, the join operation is easily effected by the LOGICAL OR operation.

### 3. Dualization Properties of Cover Mapping

The Cov operation defines a mapping from sets of n-vectors to closed sets of n-vectors. Three important properties of this mapping are the following:

$$(1) \text{ Cov } A = \text{Cov } B \iff \text{Cl } A = \text{Cl } B$$

$$\iff \text{Min } A = \text{Min } B$$

$$(2) \text{ Cov Cov } A = \text{Cl } A$$

$$(3) \text{ Cov } (\text{Cl } A \cup \text{Cl } B) = \text{Cov } \text{Cl } A \cap \text{Cov } \text{Cl } B$$

$$\text{Cov } (\text{Cl } A \cap \text{Cl } B) = \text{Cov } \text{Cl } A \cup \text{Cov } \text{Cl } B.$$

Suppose the domain of Cov is restricted to  $\mathcal{C}$ , the class of closed sets. Properties (1), (2), and (3) state that

$$\text{Cov: } \mathcal{C} \rightarrow \mathcal{C}$$

is a one-to-one, self-inverse transformation which interchanges the set operations of union and intersection. In other words, Cov defines a "dual" automorphism or dualization transformation quite analogous to ordinary set complementation. Compare:

$$(1) \bar{A} = \bar{B} \iff A = B$$

$$(2) \overline{\bar{A}} = A$$

$$(3) \overline{(A \cup B)} = \bar{A} \cap \bar{B}$$

$$\overline{(A \cap B)} = \bar{A} \cup \bar{B}.$$

(Note, however, that the Boolean laws of operation with respect to 0 and 1 do not hold for Cov. I.e., it is not generally true for closed sets that

$$A \cap \text{Cov } A = \emptyset,$$

$$A \cup \text{Cov } A = \text{Cov } \emptyset.$$

The reader may be interested to observe the effect of the Cov operation on a specific class of closed sets. The table below lists all closed sets of 2-vectors and their images under the Cov transformation:

<u>A</u>		<u>Cov A</u>
$\emptyset$	$\longrightarrow >$	$\{(0,0), (0,1), (1,0), (1,1)\}$
$\{(1,1)\}$	$\longrightarrow >$	$\{(0,1), (1,0), (1,1)\}$
$\{(0,1), (1,1)\}$	$\longrightarrow >$	$\{(0,1), (1,1)\}$
$\{(1,0), (1,1)\}$	$\longrightarrow >$	$\{(1,0), (1,1)\}$
$\{(0,1), (1,0), (1,1)\}$	$\longrightarrow >$	$\{(1,1)\}$
$\{(0,0), (0,1), (1,0), (1,1)\}$	$\longrightarrow >$	$\emptyset$

The dualization properties are now stated and proved as theorems: In each case, the word "vector" is to be interpreted exclusively as an "n-vector of 0's and 1's." A and B are arbitrary (not necessarily closed) sets.

Theorem 3.1

$$\text{Cov } A = \text{Cov } B \iff \text{Cl } A = \text{Cl } B.$$

PROOF:

- (a)  $\longleftarrow \equiv$  is trivial
- (b)  $\Longrightarrow \equiv$  is proved by showing that
- $$\text{Cl } A \neq \text{Cl } B \implies \text{Cov } A \neq \text{Cov } B.$$

Let  $b$  be a vector such that  $b \notin \text{Cl } A$ , but  $b \in \text{Cl } B$ . Define a vector  $y$  as follows:

$$y_j = 0 \quad \text{if } b_j = 1,$$

$$y_j = 1 \quad \text{if } b_j = 0.$$

Clearly,  $b_y = 0$ , so  $y \notin \text{Cov } \text{Cl } B$ . But  $y \in \text{Cov } \text{Cl } A$ . For consider any  $a \in \text{Cl } A$ . It must be the case that there is some  $j$  such that  $a_j=1, b_j=0$ , and  $y_j=1$ . If this were not so, it would be the case that  $a \leq b$ , and  $b$

would be contained in  $\text{Cl } A$ . Hence  $ay \geq 1$ , for all  $a \in \text{Cl } A$ , and so  $y \in \text{Cov } \text{Cl } A$ .

Theorem 3.2

$$\text{Cov } \text{Cov } A = \text{Cl } A.$$

PROOF:

- (a)  $a \in \text{Cl } A \implies ay \geq 1$ , for all  $y \in \text{Cov } A$   
 $\implies a \in \text{Cov } (\text{Cov } A)$ , by definition.

$$\text{Hence } \text{Cl } A \subseteq \text{Cov } \text{Cov } A.$$

- (b)  $a \in \text{Cov } \text{Cov } A \implies ay \geq 1$ , for all  $y \in \text{Cov } A$   
 $\implies a \in \text{Cl } A$ ,

for if this were not so, there would be a set  $B = \text{Cl } A \cup \{a\} \neq \text{Cl } A$ , such that  $\text{Cov } B = \text{Cov } A$ , in violation of Theorem 3.1. Hence,

$$\text{Cov } \text{Cov } A \subseteq \text{Cl } A.$$

Theorem 3.3

(a)  $\text{Cov } (\text{Cl } A \cup \text{Cl } B) = \text{Cov } \text{Cl } A \cap \text{Cov } \text{Cl } B$

(b)  $\text{Cov } (\text{Cl } A \cap \text{Cl } B) = \text{Cov } \text{Cl } A \cup \text{Cov } \text{Cl } B.$

PROOF:

- (a) The first identity is quite simple, and in fact

$$\text{Cov } (A \cup B) = \text{Cov } A \cap \text{Cov } B. \text{ Proof is left to the reader.}$$

- (b) In the case of the second identity, it is easy to establish that

$$\text{Cov } (\text{Cl } A \cap \text{Cl } B) \supseteq \text{Cov } \text{Cl } A \cup \text{Cov } \text{Cl } B.$$

In order to prove the converse, suppose that there is a vector  $y$  such that  $y \in \text{Cov } (\text{Cl } A \cap \text{Cl } B)$ , but  $y \notin \text{Cov } \text{Cl } A$  and  $y \notin \text{Cov } \text{Cl } B$ . This means that there is an  $a \in \text{Cl } A$  and a  $b \in \text{Cl } B$ , such that

$$ay = by = 0.$$

But

$$a \vee b \in \text{Cl } A \cap \text{Cl } B,$$

and

$$(a \vee b)y \geq 1,$$

by assumption. From this it follows that either  $ay \geq 1$  or

$by \geq 1$ , which is a contradiction.

Q.E.D.

#### 4. Computations with Minimal Sets; Enumeration of Minimal Covers

Although it is advantageous to state theorems in terms of closed sets, it is very disadvantageous to carry out actual computations on them. A much better idea is to rely on the identity

$$\text{Min } A = \text{Min } B \iff \text{Cl } A = \text{Cl } B$$

and to represent every set  $A$  by  $\text{Min } A$ .

The most commonly required computations are those of the union and intersection of two sets. The following two theorems show how these may be computed, using minimal representations.

##### Theorem 4.1

$$\text{Min } (\text{Cl } A \cup \text{Cl } B) = \text{Min } (A \cup B).$$

PROOF: Trivial.

Theorem 4.1 states that the minimal elements of  $\text{Cl } A \cup \text{Cl } B$  can be computed by simply removing the nonminimal elements from  $A \cup B$ .

##### Theorem 4.2

$$\text{Min } (\text{Cl } A \cap \text{Cl } B) = \text{Min } (A \vee B)$$

(By definition,  $A \vee B = \{ a \vee b \mid a \in A, b \in B \}$ .)

PROOF: Let  $C = \text{Min } (\text{Cl } A \cap \text{Cl } B)$

$$D = \text{Min } (A \vee B)$$

$$(a) \quad c \in C \implies c \geq a, c \geq b, \text{ for some } a \in A, b \in B.$$

$$\implies c \geq (a \vee b) \geq d, \text{ for some } d \in D.$$

$$(b) \quad d \in D \implies d \geq a, d \geq b, \text{ for some } a \in A, b \in B$$

$$\implies d \in (\text{Cl } A \cap \text{Cl } B)$$

$$\implies d \geq c, \text{ for some } c \in C.$$

$$(c) \quad \text{For every } c^{(1)} \in C, \text{ there is, by (a), a } d^{(1)} \in D \text{ such that } c^{(1)} \geq d^{(1)}.$$

And, by (b), there is a  $c^{(2)} \in C$ , such that  $d^{(1)} \geq c^{(2)}$ , so that

$c^{(1)} \geq d^{(1)} \geq c^{(2)}$ . But  $c^{(1)} = c^{(2)}$ , because  $C$  contains only minimal elements. Hence  $c^{(1)} = d^{(1)}$ , and  $C \subseteq D$ . A similar argument, starting with an element of  $D$  shows that  $D \subseteq C$ . Hence  $C = D$ .

Q.E.D.

Theorem 4.2 states that the minimal elements of  $C1 A \cap C1 B$  can be found by forming the set of all vectors of the form  $a \vee b$ , and then removing all nonminimal vectors from the set.

The minimal covers of a set  $A$  can be computed by applying Theorem 4.2 as follows: Let

$$A = \{a^{(1)}, a^{(2)}, \dots, a^{(m)}\}.$$

Then

$$\begin{aligned} \text{Min Cov } A &= \text{Min Cov } \bigcup_{i=1}^m \{a^{(i)}\}, \text{ by definition} \\ &= \text{Min Cov } \bigcup_{i=1}^m \{a^{(i)}\}, \text{ by (2.4) and (2.5)} \\ &= \text{Min } \bigcap_{i=1}^m \text{Cov } C1 \{a^{(i)}\}, \text{ by Theorem 3.3} \\ &= \text{Min } \bigcap_{i=1}^m C1 \text{Cov } \{a^{(i)}\}, \text{ by (2.3) and (2.4)} \\ &= \text{Min } \bigcap_{i=1}^m C1 \text{Min Cov } \{a^{(i)}\}, \text{ by (2.5)} \\ &= \text{Min } \bigvee_{i=1}^m \text{Min Cov } \{a^{(i)}\}, \text{ by Theorem 4.2.} \end{aligned}$$

The method of computation is completely specified by the above except for the determination of  $\text{Min Cov } \{a^{(i)}\}$ , for  $i=1,2,\dots,m$ . But this is quite simple. Suppose  $a^{(i)}$  contains  $r$  1's. Then  $\text{Min Cov } \{a^{(i)}\}$  contains  $r$  vectors, each containing a "1" in one of the  $r$  positions occupied by 1's in  $a^{(i)}$ .



E.g., if  $a^{(i)} = (1,1,0,0,1,0)$ ,

then

$$\text{Min Cov } \{a^{(i)}\} = \{(1,0,0,0,0,0), (0,1,0,0,0,0), (0,0,0,0,1,0)\}.$$

In practice, the computations would be carried out as follows: Let

$$A^{(k)} = \{a^{(1)}, a^{(2)}, \dots, a^{(k)}\}. \quad (k=1,2,\dots,m)$$

(1) Compute  $\text{Min Cov } A^{(1)} = \text{Min Cov } \{a^{(1)}\}$ .

(2) Set  $k=2$ .

(3) Compute  $\text{Min Cov } \{a^{(k)}\}$ .

(4) Compute

$$\text{Min Cov } A^{(k)} = \text{Min } \{y \vee y' \mid y \in \text{Min Cov } A^{(k-1)}, \\ y' \in \text{Min Cov } \{a^{(k)}\}\}.$$

(5) If  $k < m$ , increment  $k$  and return to step (3). If  $k=m$ , the computation is completed, and  $\text{Min Cov } A = \text{Min Cov } A^{(m)}$ .

As an example,

$$A = \{(1,0,1,1), (0,1,0,1), (1,1,0,0)\}$$

$$\text{Min Cov } \{a^{(1)}\} = \text{Min Cov } A^{(1)} = \{(0,0,0,1), (0,0,1,0), (1,0,0,0)\}$$

$$\text{Min Cov } \{a^{(2)}\} = \{(0,0,0,1), (0,1,0,0)\},$$

$$\text{Min Cov } A^{(2)} = \{(0,0,0,1), (0,1,1,0), (1,1,0,0)\}$$

$$\text{Min Cov } \{a^{(3)}\} = \{(0,1,0,0), (1,0,0,0)\}$$

$$\text{Min Cov } A = \text{Min Cov } A^{(3)} = \{(0,1,0,1), (0,1,1,0), (1,0,0,1), (1,1,0,0)\}.$$

The reader may be able to recognize the equivalence of this algorithm and "Petrick's Method" [6].

## 5. Computation of Optimal Covers: Equal Costs

It is now possible to describe an algorithm for computing optimal covers. Initially, the discussion will be limited to the case of equal unit costs, i.e.  $c = \epsilon$ .

Choose any vector  $\alpha \in A$  as a reducing vector. If  $\text{Cov } A$  is to be non-empty, this vector must be nonzero. Without loss of generality, assume that  $\alpha$  contains exactly  $r$  1's, and that these are in the first  $r$  positions:

$$\begin{aligned} \alpha_j &= 1 && (1 \leq j \leq r) \\ &= 0 && (r + 1 \leq j \leq n). \end{aligned}$$

Every cover  $y$  has the property that either  $y_1=1$  or  $y_2=1$  or ... or  $y_r=1$ ; otherwise, it would be the case that  $\alpha y=0$ . Suppose  $y$  is a cover and  $y_k=1$ , where  $1 \leq k \leq r$ . This insures that  $\alpha y \geq 1$ , and that  $ay \geq 1$  for all  $a \in A$  such that  $a_k=1$ . The remaining components of  $y$  must be such that  $ay \geq 1$  for all  $a \in A$  such that  $a_k=0$ .

Let  $A_{(j)}$  denote the subset of all vectors  $a$  in  $A$  such that  $a_j=0$ :

$$A_{(j)} = \{a \in A \mid a_j=0\}.$$

If  $y$  is a cover of  $A$ ,  $y \wedge \bar{e}^j$  (the vector obtained by setting  $y_j$  to 0), is a cover of  $A_{(j)}$ . Conversely, if  $y$  is a cover of  $A_{(j)}$ , then  $y \vee \epsilon^j$  (the vector obtained by setting  $y_j$  to 1) is a cover of  $A$ . Moreover, if  $y$  is a minimal cover of  $A_{(j)}$  (from which it follows that  $y_j=0$ ), the cost  $y \vee \epsilon^j$  is greater by exactly 1 than the cost of  $y$ :

$$\epsilon(y \vee \epsilon^j) = \epsilon y + 1.$$

Still assuming that the reducing vector  $\alpha$  has 1's in the first  $r$  positions, let  $A^{(1)}$  be a set with property that

$$\text{Cov } A^{(1)} = \bigcup_{j=1}^r \text{Cov } A_{(j)}.$$

Any cover  $y$  of  $A^{(1)}$  is a cover of at least one of the  $A_{(j)}$  ( $1 \leq j \leq r$ ). If it is a cover of  $A_{(k)}$ , then  $y \vee \epsilon^k$  is a cover of  $A$ . Moreover, if  $y$  is an optimal cover of  $A^{(1)}$ , it is an optimal (and hence minimal) cover of  $A_{(k)}$ , and the cost of  $y$  is exactly 1 less than the cost of  $y \vee \epsilon^k$ . Every optimal cover of  $A^{(1)}$  yields an optimal cover of  $A$ , and every optimal cover of  $A$  can be obtained from an optimal cover of  $A^{(1)}$ , whose cost is exactly 1 less.

The strategy for a computational procedure now becomes apparent. From the set  $A^{(0)} = A$ , a reducing vector  $\alpha^{(0)}$  is chosen. Based upon this choice, a set  $A^{(1)}$  is computed. From  $A^{(1)}$  a reducing vector  $\alpha^{(1)}$  is chosen, and a set  $A^{(2)}$  is computed. Sets  $A^{(3)}$ ,  $A^{(4)}$ , ... are computed, until finally a set  $A^{(N)}$  is obtained for which the zero vector is the optimal cover. ( $A^{(N)}$  is the null set.) Then an optimal cover of  $A^{(N-1)}$  can be constructed from the optimal cover of  $A^{(N)}$ , an optimal cover of  $A^{(N-2)}$  can be constructed from an optimal cover of  $A^{(N-1)}$  ..., and an optimal cover of  $A = A^{(0)}$  can be constructed from an optimal cover of  $A^{(1)}$ . The process is clearly finite, and requires exactly as many iterations as the cost of an optimal cover of  $A$ . (Note that not only does the cost of an optimal cover decrease at each iteration, but that

$$\text{Cov } A^{(0)} \subset \text{Cov } A^{(1)} \subset \dots \subset \text{Cov } A^{(N)} = \text{Cov } \emptyset$$

and

$$\text{Cl } A^{(0)} \supset \text{Cl } A^{(1)} \supset \dots \supset \text{Cl } A^{(N)} = \text{Cl } \emptyset = \emptyset.$$

All that remains is to describe the computation of  $A^{(p+1)}$  from  $A^{(p)}$ . Given any vector  $\alpha \in A^{(p)}$ ,  $A^{(p+1)}$  should be such that

$$\begin{aligned}
\text{Cov } A^{(p+1)} &= \bigcup_{\alpha} \text{Cov } A_{(j)}^{(p)}, \text{ by definition} \\
&= \text{Cov } \bigcap_{\alpha} \text{Cl } A_{(j)}^{(p)}, \text{ by Theorem 3.3} \\
&= \text{Cov Min } \bigcap_{\alpha} \text{Cl } A_{(j)}^{(p)} \text{ by (2.2)}
\end{aligned}$$

where each union and intersection is over all  $j$  such that  $\alpha_j=1$ . Hence, let

$$\begin{aligned}
A^{(p+1)} &= \text{Min } \bigcap_{\alpha} \text{Cl } A_{(j)}^{(p)} \\
&= \text{Min } \bigvee_{\alpha} A_{(j)}^{(p)}
\end{aligned}$$

where " $\bigcap$ " and " $\bigvee$ " are again over all  $j$  such that  $\alpha_j=1$ . Thus,  $A^{(p+1)}$  can be computed by taking  $r$ -fold l.u.b.'s of vectors, one vector being drawn from each set  $A_{(j)}^{(p)}$  for which  $\alpha_j=1$ .

The following example may serve to illustrate the computations. For purposes of display, matrix representation rather than set representation is used. At each step, a row vector with as few 1's as possible is chosen as the reducing vector, since this tends to reduce the number of l.u.b. operations.

Let

$$A^{(0)} = \begin{bmatrix} 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} .$$

Choosing  $\alpha^{(0)} = (0,1,0,0,1,0)$ , obtain

$$A_{(2)}^{(0)} = \begin{bmatrix} \overline{1} & 0 & 0 & 1 & 0 & \overline{0} \\ 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 \\ \underline{1} & 0 & 0 & 0 & 0 & \underline{1} \end{bmatrix}$$

$$A_{(5)}^{(0)} = \begin{bmatrix} \overline{0} & 1 & 0 & 1 & 0 & \overline{0} \\ 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 \\ \underline{1} & 0 & 0 & 0 & 0 & \underline{1} \end{bmatrix}$$

and

$$A^{(1)} = \begin{bmatrix} \overline{1} & 0 & 0 & 1 & 0 & \overline{0} \\ 0 & 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 1 \\ \underline{0} & 1 & 1 & 1 & 1 & \underline{0} \end{bmatrix} .$$

Choosing  $\alpha^{(1)} = (1,0,0,1,0,0)$ , obtain

$$A_{(1)}^{(1)} = \begin{bmatrix} \overline{0} & 0 & 1 & 0 & 0 & \overline{1} \\ 0 & 1 & 1 & 1 & 1 & \underline{0} \end{bmatrix}$$

$$A_{(4)}^{(1)} = \begin{bmatrix} \overline{0} & 0 & 1 & 0 & 0 & \overline{1} \\ \underline{1} & 0 & 0 & 0 & 0 & \underline{1} \end{bmatrix}$$

and

$$A^{(2)} = \begin{bmatrix} \overline{0} & 0 & 1 & 0 & 0 & \overline{1} \end{bmatrix} .$$

Choosing  $\alpha^{(2)} = (0,0,1,0,0,1)$ , obtain

$$A_{(3)}^{(2)} = [ \quad ] \text{ (null matrix)}$$

$$A_{(6)}^{(2)} = [ \quad ]$$

and

$$A^{(3)} = [ \quad ] .$$

The optimum cover of  $A^{(3)}$  is  $y^{(3)} = (0,0,0,0,0,0)$ , which is a cover of  $A_{(3)}^{(2)}$ . Hence

$$\begin{aligned}
y^{(2)} &= y^{(3)} \vee \epsilon^3 \\
&= (0,0,1,0,0,0)
\end{aligned}$$

is an optimal cover of  $A^{(2)}$ . Since  $y^{(2)}$  is a cover of  $A_{(1)}^{(1)}$ ,

$$\begin{aligned}
y^{(1)} &= y^{(2)} \vee \epsilon^1 \\
&= (1,0,1,0,0,0)
\end{aligned}$$

is an optimal cover of  $A^{(1)}$ . Since  $y^{(1)}$  is a cover of  $A_{(2)}^{(0)}$ ,

$$\begin{aligned}
y^{(0)} &= y^{(1)} \vee \epsilon^2 \\
&= (1,1,1,0,0,0)
\end{aligned}$$

is an optimum cover of  $A$ , and the problem is solved.

Note that by carrying out the backward construction in all possible ways, one could compute all optimal covers of  $A$ .

## 6. Computation of Optimal Covers: Unequal Costs

Let us consider the case in which the cost coefficients are not necessarily equal. As an example, suppose there is some vector  $\alpha \in A$ , such that

$$\begin{aligned}\alpha_1 &= \alpha_2 = 1 \\ \alpha_3 &= \alpha_4 = \dots = \alpha_n = 0,\end{aligned}$$

and that

$$\begin{aligned}c_1 &= 1 \\ c_2 &= 3.\end{aligned}$$

We obtain the sets  $A_{(1)}$ ,  $A_{(2)}$ , and  $A^{(1)}$  as described above. Then the following situation may occur: The vector  $y^{(1)}$  is an optimal cover of  $A_{(1)}$ , and  $cy^{(1)} = 12$ . The vector  $y^{(2)}$  is an optimal cover of  $A_{(2)}$  and  $cy^{(2)} = 11$ ;  $y^{(2)}$  is therefore an optimal cover of  $A^{(1)}$ . But

$$\begin{aligned}c(y^{(1)} \vee \epsilon^1) &= cy^{(1)} + 1 \\ &= 13\end{aligned}$$

and

$$\begin{aligned}c(y^{(2)} \vee \epsilon^2) &= cy^{(2)} + 3 \\ &= 14.\end{aligned}$$

Thus, the suboptimal cover  $y^{(1)}$  of  $A^{(1)}$  yields an optimal cover of  $A$ , and the optimal cover  $y^{(2)}$  of  $A^{(1)}$  yields a suboptimal cover of  $A$ .

This situation can be corrected by defining certain auxiliary variables, constraints, and cost coefficients in such a way that an optimal cover of  $A^{(1)}$  differs in cost from an optimal cover of  $A$  by a fixed amount. In the case at hand, let (in matrix notation),

$$\begin{aligned} \bar{A}_{(1)} &= [ \quad A_{(1)} \quad \vdots \quad 0 ] \\ \bar{A}_{(2)} &= \left[ \begin{array}{ccc|c} & A_{(2)} & & 0 \\ \hline & 0 & & 1 \end{array} \right] \\ \bar{y} &= (y_1, y_2, \dots, y_n, z_1), \end{aligned}$$

where

$$\begin{aligned} \bar{c} &= (c_1, c_2, \dots, c_n, d_1), \\ d_1 &= c_2 - c_1 = 2. \end{aligned}$$

Let  $\bar{A}^{(1)} = \text{Min} (\bar{A}_{(1)} \vee \bar{A}_{(2)})$ . A vector  $\bar{y} = (y; z)$  is a cover of  $\bar{A}_{(1)}$  iff  $y$  is a cover of  $A_{(1)}$ , but it is a cover of  $\bar{A}_{(2)}$  iff  $y$  is a cover of  $A_{(2)}$  and  $z_1=1$ . The cost coefficient  $d_1$  of  $z_1$  exactly compensates for the difference in costs in converting these two types of covers into covers of  $A$ . It follows that an optimal cover of  $\bar{A}^{(1)}$  yields an optimal cover of  $A$ .

Suppose, without loss of generality,

$$\begin{aligned} \alpha_j &= 1 & (1 \leq j \leq r) \\ &= 0 & (r + 1 \leq j \leq n), \end{aligned}$$

and

$$\text{Min}_{1 \leq j \leq r} \{c_j\} = c_r.$$

A general method of construction can be described as follows (again in matrix notation):



$$\bar{A}^{(j)} = \left[ \begin{array}{ccc|ccc} & & & & & \\ & & & & & \\ & & & & & \\ \hline & & A^{(j)} & & & 0 \\ & & & & & \\ & & & & & \\ \hline 0 & & & & & \epsilon^j \end{array} \right] \quad (j < r)$$

$$\bar{A}^{(r)} = \left[ \begin{array}{ccc|ccc} & & & & & \\ & & & & & \\ & & & & & \\ \hline & & A^{(r)} & & & 0 \\ & & & & & \\ & & & & & \\ \hline & & & & & \end{array} \right]$$

$$\bar{y} = (y; z)$$

$$= (y_1, y_2, \dots, y_n, z_1, z_2, \dots, z_{r-1})$$

$$\bar{c} = (c; d)$$

$$= (c_1, c_2, \dots, c_n, d_1, d_2, \dots, d_{r-1})$$

where

$$\begin{aligned} d_j &= c_j - \text{Min}_{1 \leq j \leq r} \{c_j\} \\ &= c_j - c_r. \end{aligned}$$

Under this construction if  $\bar{y} = (y; z)$  is a cover of  $\bar{A}^{(1)}$  and of  $\bar{A}^{(j)}$ , then  $y \vee \epsilon^j$  is a cover of  $A$ , and

$$\begin{aligned} c(y \vee \epsilon^j) &= cy + dz + c_r \\ &= cy + c_j. \end{aligned}$$

This is not, however, necessarily the most economical way to equalize costs. It is necessary to provide only one auxiliary variable for each distinct cost coefficient  $d_j$ , thereby reducing the size of  $\bar{A}^{(1)}$ .

The following example may clarify the ideas described above. Let

$$A^{(0)} = \begin{bmatrix} 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 1 \end{bmatrix},$$

as before, and

$$c = (6, 5, 4, 3, 2, 1).$$

Choosing  $\alpha^{(0)} = (0, 1, 0, 0, 1, 0)$ , obtain

$$\bar{A}(2) = \left[ \begin{array}{cccccc|c} 1 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 & 0 \\ \hline 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{array} \right]$$

$$\bar{A}(5) = \left[ \begin{array}{cccccc|c} 0 & 1 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 & 0 \end{array} \right]$$

$$\bar{A}(1) = \left[ \begin{array}{cccccc|c} 0 & 1 & 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 & 0 \end{array} \right]$$

and

$$\bar{c}^{(1)} = (6, 5, 4, 3, 2, 1, 3).$$

Choosing  $\alpha^{(1)} = (1, 0, 0, 1, 0, 0, 0)$ , obtain

$$\bar{A}(1) = \left[ \begin{array}{cccccc|c} 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ \hline 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{array} \right]$$

$$\bar{A}(4) = \left[ \begin{array}{cccccc|c} 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \end{array} \right]$$

$$\bar{A}(2) = \left[ \begin{array}{cccccc|c} 1 & 1 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \end{array} \right]$$

and

$$\bar{c}^{(2)} = (6, 5, 4, 3, 2, 1, 3, 3).$$

Choosing  $\alpha^{(2)} = (0, 0, 1, 0, 0, 1, 0, 0)$ , obtain

$$\bar{A}(2) = \left[ \begin{array}{cccccc|c} 1 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \\ \hline 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{array} \right]$$

$$\bar{A}(6) = [ \quad \quad \quad ]$$

$$\bar{A}^{(3)} = [ \quad ]$$

and

$$\bar{c}^{(3)} = (6, 5, 4, 3, 2, 1, 3, 3, 3).$$

Now  $\bar{y}^{(3)} = (0, 0, 0, 0, 0, 0, 0, 0, 0)$  is the optimum cover of  $\bar{A}^{(3)}$ , and is a cover of  $\bar{A}^{(2)}$ . Hence

$$\begin{aligned} \bar{y}^{(2)} &= y^{(3)} \vee \epsilon^6 \\ &= (0, 0, 0, 0, 0, 1, 0, 0) \end{aligned}$$

is an optimal cover of  $\bar{A}^{(2)}$ . Since  $\bar{y}^{(2)}$  is a cover of  $\bar{A}^{(1)}$ ,

$$\begin{aligned} \bar{y}^{(1)} &= y^{(2)} \vee \epsilon^4 \\ &= (0, 0, 0, 1, 0, 1, 0) \end{aligned}$$

is a cover of  $\bar{A}^{(1)}$ . Since  $\bar{y}^{(1)}$  is a cover of  $\bar{A}^{(0)}$ ,

$$\begin{aligned} y^{(0)} &= y^{(1)} \vee \epsilon^5 \\ &= (0, 0, 0, 1, 1, 1) \end{aligned}$$

is an optimal cover of  $A$ , with a cost of 6.

## 7. A Refinement

The algorithm can be made somewhat more efficient, and in such a way that the number of variables is reduced with each iteration. We restrict our discussion to the case of equal costs; the extension to unequal costs is immediate.

It has been observed that if the reducing vector  $\alpha$  is such that

$$\begin{aligned}\alpha_j &= 1 && (i \leq j \leq r) \\ &= 0 && (r + 1 \leq j \leq n),\end{aligned}$$

then for any cover  $y$ , either

$$(1) \quad y_1 = 1$$

or

$$(2) \quad y_2 = 1$$

·  
·  
·

or

$$(r) \quad y_r = 1.$$

These possibilities suggested covering problems with sets  $A_{(1)}, A_{(2)}, \dots, A_{(r)}$ , where  $A_{(j)}$  is obtained from  $A$  by removing those vectors  $a$  for which  $a_j = 1$ .

The possibilities are not, however, mutually exclusive. I.e., a cover of  $A_{(1)}$  can also be a cover of  $A_{(2)}$ .

Now consider the following:

$$(1) \quad y_1 = 1,$$

or

$$(2) \quad y_1 = 0, y_2 = 1,$$

or

$$(3) \quad y_1 = 0, y_2 = 0, y_3 = 1,$$

⋮

or

$$(r) \quad y_1 = 0, y_2 = 0, \dots, y_{r-1} = 0, y_r = 1.$$

These possibilities are disjoint and exhaustive. Moreover, for each one the value of  $y_1$  is fixed, thereby effectively removing it from the problem. We can, therefore, obtain a reduction in dimension from  $n$  for  $A^{(0)}$  to  $n-1$  for  $A^{(1)}$ , and from  $n-k$  for  $A^{(k)}$  to  $n-k-1$  for  $A^{(k+1)}$ . However, we find it simpler to retain  $n$  as the dimension, merely setting to 0 the vector components that correspond to variables whose values are fixed.

Let  $\hat{A}_{(j)}$  be formed from  $A_{(j)}$  by setting to 0 all vector components  $k$  for which  $\alpha_k = 1$  and  $k < j$ . For example, if in matrix representation,

$$\alpha = (1, 0, 1, 1, 0)$$

and

$$A_{(4)} = \begin{bmatrix} 1 & 0 & 1 & 0 & 1 \\ 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 & 1 \end{bmatrix}.$$

then

$$\hat{A}_{(4)} = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 \end{bmatrix}$$

and, of course,

$$\text{Min } \hat{A}_{(4)} = \begin{bmatrix} \bar{0} & 0 & 0 & 0 & \bar{1} \\ 0 & 1 & 0 & 0 & 0 \end{bmatrix} .$$

The set  $\hat{A}^{(1)}$  can be computed from  $\hat{A}_{(1)}, \hat{A}_{(2)}, \dots, \hat{A}_{(r)}$  exactly as before. And from an optimal cover of  $\hat{A}^{(1)}$ , an optimal cover of  $A$  can be constructed exactly as before.

This refinement not only reduces the effective number of variables; it also tends to reduce the size of minimal sets. Consider the following example:

$$\begin{aligned} \alpha &= (1, 1, 1, 0, 0, 0) \\ A &= \begin{bmatrix} \bar{1} & 1 & 1 & 0 & 0 & \bar{0} \\ 1 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 1 & 0 \\ 1 & 1 & 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 0 & 1 & 1 \end{bmatrix} . \\ A_{(1)} &= [0 \ 0 \ 1 \ 1 \ 1 \ 0] \\ A_{(2)} &= \begin{bmatrix} \bar{0} & 0 & 1 & 1 & 1 & \bar{0} \\ 1 & 0 & 0 & 0 & 1 & 1 \end{bmatrix} \\ A_{(3)} &= \begin{bmatrix} \bar{1} & 1 & 0 & 1 & 0 & \bar{1} \\ 1 & 1 & 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 0 & 1 & 1 \end{bmatrix} \\ A^{(1)} &= \begin{bmatrix} \bar{1} & 0 & 1 & 1 & 1 & \bar{1} \\ 1 & 1 & 1 & 1 & 1 & 0 \end{bmatrix} \\ \hat{A}_{(1)} &= [0 \ 0 \ 1 \ 1 \ 1 \ 0] \\ \hat{A}_{(2)} &= \begin{bmatrix} \bar{0} & 0 & 1 & 1 & 1 & \bar{0} \\ 0 & 0 & 0 & 0 & 1 & 1 \end{bmatrix} \\ \hat{A}_{(3)} &= \begin{bmatrix} \bar{0} & 0 & 0 & 1 & 0 & \bar{1} \\ 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 \end{bmatrix} \\ \hat{A}^{(1)} &= [0 \ 0 \ 1 \ 1 \ 1 \ 0] . \end{aligned}$$

Note that  $|\text{Min } \hat{A}^{(1)}| < |\text{Min } A^{(1)}|$ , while  $\text{Cl } \hat{A}^{(1)} \supset \text{Cl } A^{(1)}$ .

For any given problem, this refinement may not improve computations significantly. If the reader cares to rework the example of the previous section, he will note little change.

Questions such as which vector to choose as a reducing vector, how  $r$ -fold l.u.b.'s can be computed most effectively, and so forth, lie chiefly in the realm of heuristics. Anyone with experience programming a computer is likely to have a number of ideas of his own.

## 8. Comparison with Other Methods of Solution

The method of solution described here has not as yet been programmed for a computer or tested in any way, although plans are being made to do this. The method does need to be compared in efficiency with other algorithms, including:

- (1) integer linear programming algorithms
- (2) "alternating path" algorithms
- (3) enumeration methods
- (4) "partitioning" or "branch and bound" methods.

It is evident that covering problems as defined here are simply a special class of integer linear programming problems. Cobham, et al. [1,2], have attempted to apply integer programming algorithms, and most recently Balinski [3] has begun systematic experimentation. Results appear to be fairly encouraging.

"Alternating path" methods stem from the theory of linear graphs, in which there exist fairly efficient algorithms by this name for covering the vertices of a graph with a subset of edges, or covering the edges with a subset of the vertices. Edmonds [4] and Chaudhuri [5] have attempted to extend these algorithms to more general types of covering problems. However, in the author's opinion, these algorithms show little promise of computational efficiency.

Under the heading of enumerative methods, we might include the method of S. R. Petrick [6], which was given an unusual formulation in Section 4. Petrick's method is satisfactory for hand computation with small problems, but is inefficient for machine computation with large problems. This is



because the number of minimal covers that must be enumerated can become quite astronomical.

Partitioning or "branching" methods are most closely associated with the names of J. P. Roth [7] and E. J. McCluskey [8]. This approach decomposes the original problem into two or more subproblems with separate constraint matrices. However, instead of combining these subproblems into a single problem, these problems are solved separately, with further decompositions and simplifications. The approach is very closely related to the method presented here. In fact, some combination of the two approaches might eventually prove to be most fruitful.

Finally, it is necessary to mention that the research reported here was stimulated by a recent paper of J. F. Gimpel [9]. In this paper, Gimpel presented a method for "reducing" the matrix of a covering problem in the case in which some row contained exactly two 1's. The extension of this idea to a general algorithm, and the explication of the underlying duality relations are original to this author.

## References

- [1] A. Cobham, R. Fridshal, and J. H. North, "An Application of Linear Programming to the Minimization of Boolean Functions," Proceedings Second Annual Symposium on Switching Circuit Theory and Logical Design, AIEE Publication S-134, pp. 3-9 (1961).
- [2] A. Cobham and J. H. North, "Extensions of the Integer Programming Approach to the Minimization of Boolean Functions," IBM Research Report RC-915, Yorktown Heights, New York (April, 1963).
- [3] M. L. Balinski, private communication (January, 1965).
- [4] J. Edmonds, "On Minimum Element-Covering from a Finite Class of Weighted Sets," presented at April Meeting of American Mathematical Society, New York, New York, Paper 578-42 (April, 1961).
- [5] D. K. Ray-Chaudhuri, "An Algorithm for a Maximum Cover of an Abstract Complex," Canadian Journal of Mathematics, vol. 15, pp. 11-24 (1963).
- [6] S. R. Petrick, "A Direct Determination of the Irredundant Forms of A Boolean Function from the Set of Prime Implicants," AFCRC-TR-56-110, Air Force Cambridge Research Center (April, 1956).
- [7] J. P. Roth and E. G. Wagner, "Algebraic Topological Methods for the Synthesis of Switching Systems, Part III: Minimization of Nonsingular Boolean Trees," IBM Journal of Research and Development, vol. 4, pp. 326-344 (October, 1959).
- [8] J. B. Pyne and E. J. McCluskey, Jr., "An Essay on Prime Implicant Tables," Soc. Industr. Appl. Math Journal, vol. 9, no. 4; pp. 604-631 (December, 1961).
- [9] J. F. Gimpel, "A Reduction Technique for Prime Implicant Tables," Technical Report number 40, Department of Electrical Engineering, Princeton University (August, 1964).



UNIVERSITY OF MICHIGAN



3 9015 03466 5573