ON THE QUASI-MINIMAL SOLUTION OF THE GENERAL COVERING PROBLEM

by

Ryszard S. Michalski

Proceedings of the V International Symposium on Information Processing (FCIP 69), Vol. A3 (Switching Circuits), pp. 125-128, Yugoslavia, Bled, October 8-11, 1969.

V International Symposium on Information Processing (FCIP 69), Yugoslavia, Bled, October 8-II, 1969, Vol. A3

R. Michalski
Instytut Automatyki
Warszawa, Polska

ON THE QUASI – MINIMAL SOLUTION OF THE GENERAL COVERING PROBLEM

1. Introduction

The so-called covering problem, which we take here to mean the general problem of "covering" a given set of objects with some of its subsets so as to obtain the minimum of a specified "cost" functional, is one of the basic problems in the synthesis of switching circuits. A problem of this type arises in the synthesis of minimal one- or multiple-output combinational switching circuits, which are constructed with NOT-AND-OR, NOR, NAND and other such elements. This problem is also important for the synthesis of any kind of switching circuit, since the synthesis of sequential switching circuits is reduced at the final stage to the synthesis of combinational circuits.

The covering problem also occurs in a number of areas apart from automata theory, such as, e.g. technical diagnostics, communications network control, operations research.

As indicated in recent papers, e.g. [1], even for a small number of variables / about 10-11 / the number of operations required for an exact solution of the covering problem is not feasible even with the fastest computers. As shown by Zhuravlev [2], an exact solution requires, in the general case, an inspection of the irredundant covers / or operations which in one form or another are equivalent to such an inspection /. However, it is known from estimates that the maximum number $T(n)$ of irredundant covers may become very large even for relatively few variables, e.g. $T(n) > 10^{11}$ for $n = 8$ and $T(n) > 10^{44}$ for $n = 15$. Although inspection of some of the irredundant covers may frequently be avoided, the remainder may still be too numerous to be feasible.

Consequently, the present trend of research is to seek methods of obtaining approximate solutions, which involve a drastic reduction in the number of operations. Although several such methods exist e.g.[3-5], they nevertheless have an intrinsic drawback in that they do not yield an estimate of the "distance" between the approximate solution and the minimal solution.

This paper presents a new and the most general formulation of the covering problem, and an algorithm for its solution. This algorithm gives a so-called quasi-minimal cover, which is obtained without inspection of the irredundant covers and with a small total number of operations. Such a cover is either the minimal one or approximately minimal. In the latter case the algorithm gives an estimate of the maximal "distance" / which involves the number of elements and the cost / between the cover obtained and the minimal one.

2. Statement of the General Covering Problem in Terms of a Plane Geometrical Model

The covering problem is usually stated as follows. Given a binary matrix:

$$A = [a_{ij}]_{\substack{i=1,\dots,p \\ j=1,\dots,r}} \qquad (1)$$

containing at least one "1" in each column. Find the minimal set of rows $R \subseteq \{1,\dots,p\}$ such that for every j

$$\sum_{i \in R} a_{ij} \geqslant 1 \qquad (2)$$

In other words, the point is to find the minimal subset of rows which "covers" all columns. In a more general statement of the problem, each row is assigned a natural number called a cost and then the problem is to find a subset of rows which satisfies condition (2) and has a minimum total cost.

This statement of the covering problem, however, has a disadvantage in that knowledge of matrix A is assumed a priori. This matrix may be very large in practical cases and then the calculation of the matrix itself becomes a formidable task.

It turns out that the covering problem can be solved solely on the basis of the knowledge of a rule which enables us to generate individual rows of the matrix, without previously having to construct the entire matrix. Precisely such an approach to solving the covering problem is described in this paper. Consequently, the problem itself is formulated in a different manner, use being made of a plane geometrical model.

Suppose we are given a diagram which is arbitrary rectangle divided into $2^{[\frac{n}{2}]}$ rows and $2^{n-[\frac{n}{2}]}$ columns, where $[\frac{n}{2}]$ is the integer part of the number $\frac{n}{2}$, and n is a specified natural number. For definiteness, we assume that an elementary cell of this diagram, formed by the intersection of any row with any column, does not include the points belonging to any line, which divides the rectangle or to its perimeter. To cells of the diagram we assign in lexical order the consecutive numbers $0,1,\dots,2^n-1$, as in the case $n = 6$ illustrated in Fig.1. The number of cell e will be denoted by $\gamma(e)$, and cell e having number j by e^j. The set of numbers of cells $e \in E$ will be denoted by $\gamma(E)$.

Assume that we are given a mapping f of the set $\gamma = \{0,1,\dots,2^n-1\}$ into set $\{0,1,*\}^m$, that is into the set of all sequences of m elements from the set $\{0,1,*\}$, where $*$ denotes an unspecified / "don't care" / value:

$$f : \gamma \longrightarrow \{0,1,*\}^m \qquad (3)$$

If $m = 1$, f is said to be a one-output, and if $m > 1$, a multiple-output mapping. Now we assume that f is a one-output mapping; we shall consider the case of multiple-output mapping later.

Let us assign a value $f(\gamma(e))$ to each cell e of the diagram.

Definition 1 : The set of all cells of the diagram, along with the values thus assigned from the set $\{0,1,*\}$, we call the image of mapping f and denote by $T(f)$.

The subsets of cells of image $T(f)$ of values $0,1,*$ we denote as F^0, F^1, F^*, respectively. Assume that we are given a function φ which maps the family 2^{γ} of all subsets of into $\{0,1\}$:

$$\varphi : 2^{\gamma} \longrightarrow \{0,1\} \qquad (4)$$

Definition 2 : Any set of cells E such that $\varphi(E) = 1$ is called a complex of cells.

Covering problems in different areas are characterized by different determination of the complex. For instance, in problems of synthesis of switching circuits a complex will be a set of cells which corresponds to a switching function realized by single functor / AND, NAND, THRESHOLD etc. /. In problems of, say, setting up diagnostic tests for machines, the complex will be a set of cells corresponding to the set of machine elements, whose operation can be checked by a single test.

To each complex E we assign a natural number $z(E)$ called the cost of E. The condition we assume in the case of one-output mappings is that the cost of the complexes ordered by the inclusion relation $\subset$ is decreasing. This means that if $E_1 \subseteq E_2$, then $z(E_1) \geqslant z(E_2)$.

Definition 3 : A set of complexes $D(T) = \{E_j\}_{j=1}^{p}$ is called the cover of image $T(f)$, if it satisfies the condition:

$$F^1 \subseteq \bigcup_{j=1}^{p} E_j \subseteq F^1 \cup F^* \qquad (5)$$

This definition implies that if a set of complexes is to be a cover, its set-theoretic sum must completely cover the set F^1 and may also cover some cells of the set F^*, which acts as a "margin". It cannot, on the other hand, cover any cell of the set F^0.

Definition 4 : The cost of cover $D(T) = \{E_j\}_{j=1}^{p}$ is the sum:

$$s(D(T)) = \sum_{j=1}^{p} s(E_j) \qquad (6)$$

Definition 5 : The minimal cover $M(T)$ of image $T(f)$ is a cover which has the minimum number of elements and has a minimum cost for that number.

Definition 6 : A complex E satisfying the conditions:

$$E \subseteq F^1 \cup F^*, \quad E \cap F^1 \neq \emptyset \qquad (7)$$

where $\emptyset$ is the empty set, is called the maximal complex in the image $T(f)$, if it is maximal under inclusion.

Maximal complexes in image $T(f)$ will be denoted by L_i, $i = 1,2,\ldots$. It is readily shown that each minimal cover consists exclusively of maximal complexes. This statement follows directly from the assumption of the cost function, which implies that a maximal complex has the least cost of all complexes which are its subsets.

Definition 7 : A maximal complex L_i in image $T(f)$ is called a core complex, if there exists a cell $e \in L_i \cap F^1$, which is not included in any other maximal complex in image $T(f)$.

Definition 8 : The set of all core complexes in image $T(f)$ is called the core of the cover of image $T(f)$.

Since core complexes are the only complexes covering certain cells of the set F^1, a core must be contained within each minimal cover of image $T(f)$.

Definition 9 : A cover consisting of maximal complexes and which is minimal under inclusion is called an irredundant cover.

In accordance with definition 9, if any maximal complex is removed from an irredundant cover it ceases be a cover. The minimal covers are among the irredundant covers.

The fundamental concept for the algorithm of the synthesis of covers, described in section 2 , is a star of a cell $e \in F^1$.

Definition 10 : The star $G(e)$ of a cell $e \in F^1$ is the set of all maximal complexes covering cell e.

The set of all cells $e_j \in F^1$ covered by the maximal complexes of star $G(e)$ will be denoted by $G^{U}(e)$. The number of elements in any set, say K, will be denoted by $c(K)$.

3. Algorithm A^q of Synthesis of Quasi-Minimal Covers

In this section we shall briefly describe an algorithm A^q for the synthesis of covers of image $T(f)$ of one-output mapping. A full description of this algorithm is given in 6 . This algorithm yields a cover $M^q(T)$, called a quasi-minimal cover, along with an estimate of the maximum possible difference between this cover and the minimal cover, expressed in terms of the number of elements Δ and cost δ :

$$c(M^q(T)) - c(M(T)) \leq \Delta \qquad (8)$$

$$s(M^q(T)) - s(M(T)) \leq \delta \qquad (9)$$

The theoretical possibility of determining the maximum difference between the number of elements in any arbitrary cover and in the minimal cover is a consequence of the following theorem. Assume that we are given a family G^r of stars of cells $e \in F^1$ such that any two stars chosen from it are disjoint sets.

Theorem 1 . The number of elements $c(M(T))$ of the minimal cover of image $T(f)$ satisfies the relation:

$$c(M(T)) \geq \varrho \qquad (10)$$

where $\varrho = c(G^r)$.

The proof of this theorem can be found in [6]. The family G^r will be referred to as a family of disjoint stars. The theorem implies that if we have a cover $D(T)$ and know the number ϱ of elements in a family of disjoint stars, the difference $\Delta = c(D(T)) - \varrho$ can be treated as an estimate of the maximum difference between the number of elements in this cover and in the minimal one. An algorithm R for synthesis of a family of disjoint stars is presented in Fig.2. The stars determined while the execution of the algorithm constitute a family of disjoint stars, which is maximal under inclusion.

The sign := is used as in ALGOL / it denotes that a variable on the left side of the sign takes a new value resulting from the operation written on the right/ . The variable F^P is an auxiliary variable whose values are sets. By $O^P(F^P; e_1)$ we denote the operation of choosing the cell with smallest number from the set specified by the current value of F^P /briefly, from the set F^P/ and assigning the notation e_1 to that cell.

In the algorithm R it is assumed that the mode of generating stars of given cells is known. If we have a concrete covering problem, then we know what is meant by a complex and an algorithm for the generation of stars can always be constructed. If the algorithm R is repeated, generating stars of cells other than those with the smallest number, a value of ϱ may turn out to be larger. Theorem 1 still holds for the new value of ϱ , thus making it possible to improve the previous estimate of Δ .

The algorithm A^q consists of two parts.

Part I .

We successively generate disjoint stars according to the algorithm R . During this process we determine for each star a maximal complex L^q, called a quasi-extremal /see def. 9/ and then perform the following operations:

$$F^1 := F^1 \setminus L^q, \quad F^* := F^* \cup L^q, \quad M^q := M^q \cup \{L^q\} \qquad (11)$$

Here F^1, F^*, M^q denote variables whose values are sets. The starting values of F^1 and F^* are the sets previously defined as F^1 and F^*, and that of M^q is the empty set. Then the value of M^q at any given stage of this process is a set containing all the quasi-extremals which have been determined up to that stage. The performance of operations (11) means that once the complex L^q has been incorporated into the set M^q, the cells of F^1, covered by that complex can henceforth be treated as cells of value $*$, that is cells of the set F^* . The complexes subsequently chosen may, but do not have to, cover these cells.

Definition 9 : A quasi-extremal of star $G(e)$ is a maximal complex $L^q \in G(e)$ which covers the maximum number of cells in the set constituting the current value[1] of variable F^1 and has the minimal cost of all the complexes of $G(e)$ covering the same number of cells in this set.

If a given star contains more then one quasi-extremal, any of them may be chosen. This part of the algorithm ends when we execute these operations for the last disjoint star. If after the first part has been executed, the set F^1 is empty, then the set M^q is a quasi-minimal cover of image $T(f)$. Otherwise, we proceed to execute Part II of the algorithm.

Part II .

In the first step of this part the star of the cell with the smallest number in the set F^1 is determined, a quasi-extremal is chosen from it and operations (11) are performed. The next steps are carried out in a similar fashion until $F^1 = \emptyset$. The set M^q then obtained is the quasi-minimal cover $M^q(T)$ of image $T(f)$. The number of steps performed in this part determines the parameter Δ in Eq. 8 .

A flow diagram of the algorithm is presented in Fig.3 . In order to determine δ in Eq. 9 we perform the following operations. In Part I we calculate for each disjoint star the difference between the cost of the chosen quasi-extremal and the minimal cost complex L^m of that star, and then sum over all disjoint stars. Then to this sum we add the sum of the costs

1/ Henceforth in the description of the algorithm, the sets which constitute the current values of the variables F^1, F^*, M^q will be referred to briefly as sets F^1, F^*, M^q.

of the quasi-extremals chosen in Part II and thus obtaining the value of δ. The final values of the parameters ϱ and x determine a number of elements and a cost, respectively, which are lower bounds for the minimal cover. Then $\frac{\Delta}{\varrho}$ and $\frac{\delta}{x}$ provide an estimate for maximum possible relative defference between the quasi-minimal and minimal covers.

If Δ and δ are considered to be too large after the first execution of the algorithm, then the algorithm may be repeated with other choices of the quasi-extremals or disjoint stars can be generated for cells other than the cells with smallest number in the sets specified by the consecutive values of F^p, as is done in the first execution. If $\Delta = 0$, $\delta = 0$, the quasi-minimal cover is certainly a minimal cover.

It is noteworthy that the core need not be determined at the beginning of the algorithm.

3. Extension of the Algorithm A^q to Multiple-Output Mappings

Now, let f be an m-output mapping / $m > 1$ /. Then to each cell of the diagram there corresponds a sequence of m elements from the set $\{0,1,*\}$. Therefore, let us divide every cell e^j in the diagram into m smaller cells e^{jk}, $k=1,2,..,m$, which we shall call subcells / Fig.4. /. The index j which previously was called the number of the cell e^j is now called the number of the subcell e^{jk}, and index k the subnumber of the subcell. Function f in the case $m > 1$ may be treated as a set of functions $\{f^k\}$, $k=1,2,..,m$ mapping the set γ into the set $\{0,1,*\}$, that is the functions considered previously. Let us assign the value $f^k(j)$ to each subcell e^{jk}.

The image $T(f)$ of mapping f will be the name given to all subcells of the diagram with values so assigned.

As before, by F^1, F^0, F^* we shall denote sets of subcells with values of $1,0,*$, respectively. All concepts - complex, maximal complex, cover, cost of cover, irredundant and minimal cover, star and quasi-extremal - introduced before remain unchanged, except that here we are considering subcells instead of cells. In addition we introduce the concepts of subimages $T(f^k)$, $k=1,2,..,m$ and their covers. A subimage $T(f^k)$ is a set of subcells of image $T(f)$ which have the subnumber k. A cover of subimage $T(f^k)$ is the name given to a set of complexes $D(T^k) = \{E_j\}_{j=1}^{p}$ which satisfies the condition:

$$F^{1,k} \subseteq \bigcup_{j=1}^{p} E_j \subseteq F^{1,k} \cup F^{*,k} \tag{12}$$

where $F^{1,k}$, $F^{*,k}$ are subsets of sets F^1, F^* consisting of cells with subnumber k.

The minimal cover of subimage $T(f^k)$ is the cover $M(T^k)$, which is a subset of the cover $M(T)$ and has minimum number of elements and the minimum cost for that number of elements.

The algorithm A^q can be used directly to determine the cover $M^q(T)$ of image $T(f)$. In order to specify the covers of subimages $T(f^k)$, when the algorithm A^q is being executed there should be assigned to each consecutive quasi-extremal a set $I \subseteq \{1,2,..,m\}$ containing all the subnumbers of its subcells. The set of quasi-extremals with subnumber k in their set I constitutes the cover $M^q(T^k)$ of the image $T(f^k)$.

4. Conclusions

We have presented here an algorithm for the general solution of the covering problem. In order to apply this algorithm to a concrete covering problem, it is necessary to have a rule for determining the complexes and a method based on this rule for generating stars.

In reference [6], such a rule is given e.g. for the case of the synthesing of minimal switching circuits. There was possible to interpret the diagram in such a way that the complexes have a very simple geometrical representation, which is useful for hand realization of the algorithm. In machine realization it is not necessary to use this interpretation and the stars can be obtained using only the numbers of cells of F^1 and F^0. This means that we don't need to record in a machine all of the image $T(f)$ / as a matrix /, but only to record these sets. It allows us to realize the algorithm even for very large number of variables.

The algorithm A^q has been used as a basis of an automatic design system for the synthesis of switching circuits. The program was written in the LYAPAS language for the Odra 1204 computer in the Computer Center of the Polish Academy of Sciences. The system presently operational gives quasi-minimal covers in the synthesis of one- or multiple-output switching circuits with up to 31 input variables.

Experiments have shown this system to work satisfactorily even for very complex covering problems. A detailed description of this system and the experiments will be presented in a separate paper.-

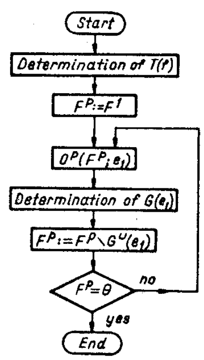
References

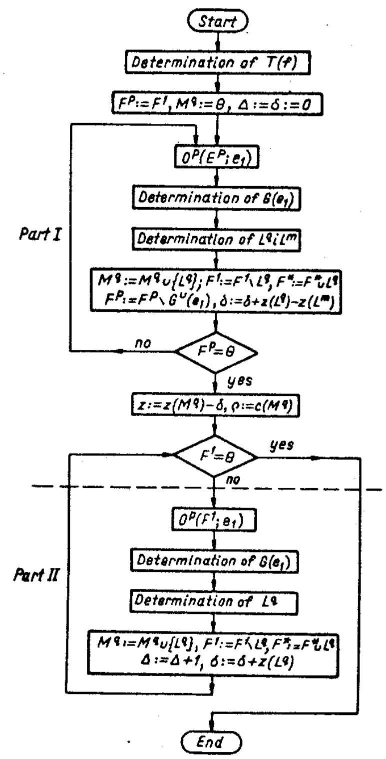
1. Vasilev I.L., Trudnosti minimizatsii bulevykh funktsii na osnove universalnykh podkhodov, Doklady AN SSSR, vol. 171, No 1, 1966.
2. Zhuravlev Yu.I., O nevozmozhnosti postroeniya minimalnykh normalnykh form funktsii algebry logiki v odnom klasse algoritmov, Doklady AN SSSR, vol.132, No 3,pp.504-506,1960.
3. Roth I.P., Algebraic topological methods for synthesis of switching systems, Trans.of the Americ.Mathematic.Society, vol.88, No 2, pp.301-326, July 1958.
4. Kazakov V.D., O minimizatsii logicheskikh funksii bolshego chisla peremennykh, Avtomatika i Telemekhanika, No 9, 1962.
5. Schneider P.R., Dietmeyer D.L., An algorithm for synthesis of multiple-output combinatorial logic, IEEE Trans.on Computers, vol.C-17, pp.117-135, February 1968.
6. Michalski R.S., Synteza wyrażeń minimalnych i rozpoznawanie symetrii funkcji logicznych, to appear in Prace Instytutu Automatyki PAN, 1969.

0	1	2	3	4	5	6	7
8	9	10	.	.	.	.	.
.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.
.	.	.	.	.	53	54	55
56	57	58	59	60	61	62	63

Distribution of cell numbers in the diagram for n = 6

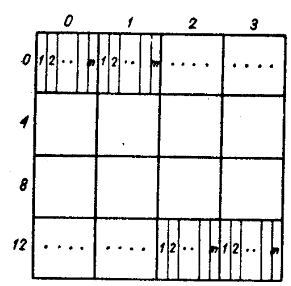
Fig.1.





Algorithm A^q

Fig.3.



Division of cells in the diagram for n = 6 into subcells with subnumbers 1,2,......,m

Fig.4.