

Combinatorial Problems over Logical Matrices in Logic Design and Artificial Intelligence

Arkadij Zakrevskij

Abstract - Logic design and artificial intelligence constitute a rich source of hard combinatorial problems which can be formulated in terms of logical matrices. Research into their classification and working out practically efficient solving algorithms were started in early sixties at the Tomsk University and the Siberian Physiko-Tekhnical Institute and developed later in the Institute of Engineering Cybernetics of the Academy of Sciences of Belarus. A brief review of obtained results is presented in this paper.

case of optimization problems when these algorithms enable to find solutions close to optimal ones.

Naturally, researches in this direction were conducted intensively for many years in different countries, by numerous teams. One of them is the laboratory of logical design of the Institute of Engineering Cybernetics of the Academy of Sciences of Belarus, which continued the research started at Tomsk University and the Siberian Physiko-Tekhnical Institute in early sixties [2-7]. This paper presents a short review of obtained results.

1. INTRODUCTION

Engineering discrete information systems on the base of modern microelectronics technology is quite impossible without their design automation. In its turn, formalization of the regarded design processes is a necessary step towards their automation. Meanwhile, it is well-known that the design of discrete systems is closely connected with solving various hard combinatorial problems, especially at the level of logical design, where the logical structure of the system is elaborated. The quality of designed systems greatly depends on these problems solutions which demand the lion's share of time and memory resources when using modern computers. There arises a problem of combinatorial support for the design automation: classifying combinatorial problems and reducing them to a strict mathematical form, developing methods for their solution and synthesising programs for their implementation, creating an intellectual interface which should facilitate their using in engineering practice.

The nature of combinatorial problems of logical design is close to that of many problems of artificial intelligence. Most of them are NP-hard, i.e. their computational complexity (measured by used volumes of time and memory) depends exponentially on the quantity of input data [1]. They say that in principle there is no efficient algorithm for solving such problems. But only theoretical efficiency is supposed by that: it is accepted that an algorithm is efficient if only its computational complexity is restricted by some polynomial of the input data volume when the latter is unlimited. Meanwhile, it is possible to suggest practically efficient algorithms for solving many NP-hard problems. They can be successfully used in the band of real restrictions imposed on input data and/or in

2. MATRIX REPRESENTATIONS

The majority of combinatorial problems of logical design and artificial intelligence can be formulated in terms of logical matrices, Boolean or ternary ones. Boolean matrices, convenient for visual perception, can be used for description of binary relations of various nature, and ternary matrices can be interpreted specifically as compressed forms of Boolean ones [6,8].

Let A and B be some finite sets, and $\#$ - some binary relation between them: $\# \subseteq A \times B$. And let $[A\#B]$ be the Boolean matrix of this relation, its rows corresponding to elements from A , and columns - to elements from B .

For instance, if $B = \{a, b, c, d, e, f\}$, and A is a set of three subsets from B , namely $B_1 = \{a, b, d\}$, $B_2 = \{b, e, f\}$ and $B_3 = \{c, d, e\}$, then A and B are connected by the relation of belonging $\in$ of elements from B to elements from A , which can be conveniently represented by a Boolean matrix $[A \ni B]$ (below on the left) or a ternary one, using the symbol of uncertainty “-” in case of incomplete information about this relation (below on the right - when it is not known if b belongs to B_1 and d - to B_3).

a	b	c	d	e	f		a	b	c	d	e	f	
1	1	0	1	0	0	B_1	1	-	0	1	0	0	B_1
0	1	0	0	1	1	B_2	0	1	0	0	1	1	B_2
0	0	1	1	1	0	B_3	0	0	1	-	1	0	B_3

Logical matrices are suitable for application in dialogue design systems and are easily jointed with program modules. They are included in the number of basic operands of programming language LYaPAS, intended for logical problems and highly efficient in case of Boolean vectors and matrices processing [5,7,9,10]. They

served as a base for development of efficient means of matrix transformations, which solve numerous problems of logical analysis and synthesis of discrete devices and diagnostics of their failures, as well as many problems of logical recognition of technical and natural objects [11-16].

3. CLASSICAL COMBINATORIAL PROBLEMS

It was the problem of minimization of Boolean functions in the class of normal forms, disjunctive (DNF) or conjunctive (CNF), that served as an initial source of combinatorial problems over Boolean and ternary matrices.

Let X be a set of n Boolean variables, taking values from the set $E=\{0,1\}$, $M=E^n$ - the Boolean space of these variables, and M^I - some its subset, regarded as characteristic for a Boolean function $f: E^n \rightarrow E$. Both the set M^I and the function f can be represented by a Boolean matrix $[M^I \# X]$ of the binary relation $\#$: an element from X takes the value 1 on an element from M^I . The matrix rows can be interpreted by that either as sets of values of the variables or as corresponding complete conjunctive terms constituting a canonical CNF.

The same information can be given in the more compact interval form. Ternary matrices are used by that, with elements choosing values from the set $\{0,1,-\}$, and with rows interpreted as conjunctive terms or as intervals of M , representing characteristic sets of these terms. It is considered that a Boolean matrix is equivalent to a ternary one if each of its rows (but only they) can be obtained from some row of the ternary matrix by changing values “-” for 0 or 1 (a ternary row covers a Boolean one in this case). The problem of minimization of a Boolean function in the DNF class is reduced to looking for a ternary matrix T which is equivalent to a given Boolean matrix B and has the minimum number of rows.

For example, the following matrix pair illustrates a transition from the canonical CNF $f = abc'd' \vee abcd' \vee ab'c'd' \vee ab'cd' \vee abcd \vee abc'd \vee a'bc'd$ to the more compact CNF $f = ad' \vee ab \vee bc'd$.

$$\begin{array}{cc}
 & a \ b \ c \ d \\
 B = & \begin{array}{l} 1 \ 1 \ 0 \ 0 \\ 1 \ 1 \ 1 \ 0 \\ 1 \ 0 \ 0 \ 0 \\ 1 \ 0 \ 1 \ 0 \\ 1 \ 1 \ 1 \ 1 \\ 1 \ 1 \ 0 \ 1 \\ 0 \ 1 \ 0 \ 1 \end{array} \\
 & a \ b \ c \ d \\
 T = & \begin{array}{l} 1 \ - \ - \ 0 \\ 1 \ 1 \ - \ - \\ - \ 1 \ 0 \ 1 \end{array}
 \end{array}$$

Observing some ternary matrix T with n columns, one may put a question: is it equivalent to the Boolean matrix with the same number of columns, which contains all possible 2^n rows? In case of positive answer the matrix T is called regenerated and presents a DNF identical with 1,

i.e. tautology. Checking a ternary matrix on regeneration is equivalent to the well-known satisfiability problem - a basic problem of the theory of computations [1]. Several practically efficient algorithms solving this problem have been suggested and investigated in [11,17,18].

The classical method for minimization of DNF consists (with accuracy to terms) in constructing the set P of all maximal on M^I intervals of the space M and selecting from P a minimum number of intervals containing together all elements of M^I - or, what is the same, finding a shortest row cover for the Quine matrix $[P \# M^I]$, i.e. a minimum combination of rows containing together at least one 1 in each column [19].

The second of these problems (the covering problem) turned out to be very fruitful, having useful interpretations in different applications where not only shortest covers are regarded but irredundant ones as well (not covered by others). The following example illustrates an initial Boolean matrix A and a “reverse” matrix B . Columns of the latter one point out all irredundant covers of matrix A , the shortest covers in their number, one of which marked by 1s in the first column of B is represented also by matrix C .

$$\begin{array}{ccc}
 & 0 \ 0 \ 1 \ 1 \ 0 \ 1 \ 1 & 1 \ 1 \ 1 \ 1 \ 0 \ 0 & 0 \ 0 \ 1 \ 1 \ 0 \ 1 \ 1 \\
 A = & 1 \ 1 \ 1 \ 0 \ 0 \ 0 \ 0, & B = & 1 \ 1 \ 0 \ 0 \ 1 \ 0, & C = & 1 \ 1 \ 1 \ 0 \ 0 \ 0 \ 0 \\
 & 0 \ 1 \ 0 \ 1 \ 1 \ 0 \ 0 & 1 \ 0 \ 1 \ 0 \ 1 \ 1 & 0 \ 1 \ 0 \ 1 \ 1 \ 0 \ 0 \\
 & 0 \ 1 \ 0 \ 0 \ 0 \ 1 \ 0 & 0 \ 0 \ 0 \ 1 \ 1 \ 1 \\
 & 1 \ 0 \ 0 \ 0 \ 0 \ 0 \ 1 & 0 \ 0 \ 1 \ 1 \ 1 \ 1 \\
 & 0 \ 0 \ 1 \ 0 \ 1 \ 0 \ 0 & 0 \ 1 \ 0 \ 1 \ 0 \ 1
 \end{array}$$

Note that computing the “reverse” matrix is equivalent to conversion the set of “maximal 1s” of a monotone Boolean function (upper bound of the set M^I) into the set of “minimum 0s” (lower bound of $M \setminus M^I$). This operation is used when checking conditions for implementation of Boolean functions by threshold elements [20,21].

A lot of publications were dedicated to the covering problem. There were proposed various algorithms for its solution, including trivial “greedy” algorithms and more sophisticated, exact and approximate [22-26].

This problem belongs to a rich class of problems of finding a Boolean matrix minor formed by some rows or columns and possessing some given quality by that. The problem of finding a minimum unconditional diagnostic test is another representative of this class. It is formulated as follows: a Boolean matrix $B=[D \# S]$ is given which represents the binary relation $\#$ between a finite set of failures D and a set of observable attributes S ($d \# s^j$, if failure i has attribute j), and it is needed to choose from S a minimum number of attributes sufficient for recognition of any failure. Solving this problem is reduced to finding in B some minimum column minor with different rows (it is supposed that all rows of B are different). Any obtained result can be represented by a Boolean vector b with 1s marking the chosen columns.

For instance,

$$\begin{aligned}
 & \begin{matrix} 1 & 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ \mathbf{b} = 1 & 0 & 0 & 0 & 1 & 0 & 1 \end{matrix} \\
 \mathbf{B} = &
 \end{aligned}$$

Some exact and approximate algorithms for solving this problem are given in [11,14,15].

Thousands of publications were dedicated to the problem of Boolean functions minimization, [27-29] in their number, were functions of many arguments were considered as well as weakly specified Boolean functions. Such a function is defined by a pair of sets M^1 and M^0 , composed of relative small number of inputs where the function takes accordingly values 1 and 0. Minimization of weakly specified Boolean functions is reduced to finding a minimum set of intervals covering together M^1 but not intersecting with M^0 by that.

4. MATRIX LOGICAL EQUATIONS

Development of modern VLSI technology stimulated researches in optimal silicon implementation of logic circuits. The matrix structure of circuits is adequately reflected by mathematical matrix tools suggested for logic analysis and synthesis. These tools solve the problems of mutual minimization of systems of Boolean functions using as criteria the whole number of conjunctive terms, and also the number of variables and literals - in case of partial functions [13,30-35].

In accordance with these methods, any system of Boolean functions is represented by a pair of matrices $\mathbf{B}$ and $\mathbf{U}$. The matrix $\mathbf{B}$ serves for the enumeration of elements or intervals of the Boolean space M , constituting the area of definition for the functions, and the matrix $\mathbf{U}$ shows the corresponding values of the functions represented by matrix columns.

Some problems which seem quite different at the interpretation level turn out to be very close under more abstract formulation. For instance, such are the problems of diagnostics, minimization of the number of arguments and decomposition of a system of Boolean functions. They are reduced combinatorially to selecting from a Boolean matrix $\mathbf{B}$ a minimum column minor, which satisfies one of the following conditions:

- rows corresponding to *different* rows of the matrix $\mathbf{U}$ should be different (the condition of inside decomposition).
- rows corresponding to *equal* rows of the matrix $\mathbf{U}$ should be different (the condition of outside decomposition).

For example, the result of solving this problem for concrete matrices $\mathbf{B}$ and $\mathbf{U}$ under the condition a is presented by the matrix $\mathbf{C}$, and under the condition b - by the matrix $\mathbf{D}$.

$$\begin{aligned}
 & \begin{matrix} a & b & c & d & e & f \\ 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 0 \\ \mathbf{B} = 0 & 1 & 0 & 1 & 1 & 0 \\ 1 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 \end{matrix} \quad \begin{matrix} u & v & w \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 1 & 0 \\ \mathbf{U} = 0 & 1 & 0 \\ 0 & 1 & 0 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{matrix} \quad \begin{matrix} a & f \\ 0 & 1 \\ 0 & 1 \\ 0 & 0 \\ \mathbf{C} = 0 & 0 \\ 1 & 0 \\ 1 & 1 \\ 1 & 1 \end{matrix} \quad \begin{matrix} a & b \\ 0 & 1 \\ 0 & 0 \\ 0 & 0 \\ \mathbf{D} = 0 & 1 \\ 1 & 1 \\ 1 & 0 \\ 1 & 1 \end{matrix}
 \end{aligned}$$

A number of logic circuits design problems are reduced to solving matrix logical equations $\mathbf{U} = \mathbf{B} \times \mathbf{C}$. Here $\mathbf{U}$, $\mathbf{B}$ and $\mathbf{C}$ are Boolean matrices, and conjunction ($\wedge$), on the one hand, and disjunction ($\vee$) or exclusive disjunction ($\oplus$), on the other hand, serve correspondingly as inside and outside operators when multiplying matrices. Some of these matrices are given by that, and other ones should be found. Regarding more complicated circuits entails solving systems of such equations. This problem is decomposed into a series well-formulated and more easy problems over separate matrices (50 such problems were described in [36]). For instance, the following ones are belonging to their number, where some Boolean matrix $\mathbf{B}$ is regarded:

- 1) the problem about a minimum disjunctive basis [37]: to find for $\mathbf{B}$ such matrix $\mathbf{C}$ with minimum number of rows that any row of $\mathbf{B}$ equals the componentwise disjunction of some rows of $\mathbf{C}$;
- 2) the problem about a minimum linear basis: the problem is the same as the previous one, but exclusive disjunction is regarded instead of disjunction;
- 3) the problem about a minimum disjunctive code [38]: to find for $\mathbf{B}$ all rows of which are different such matrix $\mathbf{C}$ with minimum number of columns that all its rows are also different and each its column equals the componentwise disjunction of some columns of $\mathbf{B}$;
- 4) the problem about a minimum packing: to decompose $\mathbf{B}$ into a minimum number of row minors each of which contains in every column not more than one unit;
- 5) the problem about a compact packing: to find in $\mathbf{B}$ such a row minor which contains in each column exactly one unit.

The following examples illustrate some of these problems:

$$\begin{aligned}
 & \begin{matrix} 1) & 2) & 5) \\ 1 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 1 & 1 & 1 & 1 & 0 & 1 & 1 & 0 & 0 \\ \mathbf{B} = 0 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 1 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \end{matrix} \\
 & \begin{matrix} 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ \mathbf{C} = 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \end{matrix}
 \end{aligned}$$

$$\begin{array}{rcl}
 & \begin{array}{cccccc} 1 & 1 & 1 & 0 & 1 & 0 & 0 & 1 \end{array} & \\
 \mathbf{B} = & \begin{array}{cccccc} 0 & 1 & 1 & 1 & 0 & 1 & 0 & 0 \end{array} & = \mathbf{F} \\
 & \begin{array}{cccccc} 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 \end{array} & \\
 & \begin{array}{cccccc} 0 & 0 & 1 & 1 & 1 & 0 & 1 & 1 \end{array} & \\
 & \begin{array}{cccccc} 1 & 0 & 1 & 1 & 1 & 0 & 1 & 0 \end{array} & \\
 & \begin{array}{cccccc} 0 & 1 & 0 & 0 & 0 & 1 & 1 & 1 \end{array} & \\
 & \begin{array}{cccccc} 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 \end{array} & \\
 \mathbf{K} = & \begin{array}{cccccc} 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 \end{array} & = \mathbf{Z} \\
 & \begin{array}{cccccc} 1 & 0 & 1 & 0 & 0 & 0 & 0 & 1 \end{array} & \\
 & \begin{array}{cccccc} 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 \end{array} &
 \end{array}$$

5. SOME PROBLEMS OF THE AUTOMATA THEORY

When a binary relation represented by a Boolean matrix $\mathbf{B}$ is defined on one set and is symmetrical, it can be interpreted as the relation of incompatibility on the set of partial states of a parallel automaton [51], and solving this problem - as looking for a minimum interval displacing code for these states, what is necessary for the hardware implementation of parallel algorithms for logical control [52-53]. This problem can be regarded also as the problem of covering the graph of incompatibility between partial states by a minimum number of complete bipartite subgraphs which can be represented by ternary vectors with 1s marking components of one part and 0s - marking the other part.

This can be illustrated by the following example of an incompatibility graph. A minimum solution of the regarded problem for this graph includes three complete bipartite subgraphs represented by the corresponding columns of a ternary matrix shown on the right.

$$\begin{array}{rcl}
 & \begin{array}{cccccc} 0 & 1 & 0 & 1 & 0 & 1 \end{array} & \begin{array}{cc} 1 & 1 & 0 \end{array} \\
 & \begin{array}{cccccc} 1 & 0 & 0 & 1 & 1 & 0 \end{array} & \begin{array}{cc} 1 & 0 & 1 \end{array} \\
 & \begin{array}{cccccc} 0 & 0 & 0 & 1 & 0 & 0 \end{array} & \begin{array}{cc} 1 & - & - \end{array} \\
 & \begin{array}{cccccc} 1 & 1 & 1 & 0 & 1 & 1 \end{array} & \begin{array}{cc} 0 & 1 & 1 \end{array} \\
 & \begin{array}{cccccc} 0 & 1 & 0 & 1 & 0 & 1 \end{array} & \begin{array}{cc} 1 & 1 & 0 \end{array} \\
 & \begin{array}{cccccc} 1 & 0 & 0 & 1 & 1 & 0 \end{array} & \begin{array}{cc} 1 & 0 & 1 \end{array}
 \end{array}$$

Practically efficient algorithms for solving this problem were proposed in [54-56].

Rather rich in the sense of possible interpretations is also the problem of compact packing which specifically was applied for checking logical control algorithms for correctness [57]. For example, let a Boolean matrix $\mathbf{B}$ to present the relation of belonging of places of some net of free choice [58] to reachable markings, and the columns of a matrix $\mathbf{C}$ - to show all possible compact packages for $\mathbf{B}$. Then (as follows from the Hack theorem [58]) every row of $\mathbf{C}$ will have at least one unit if the net is live and safe.

6. LOGICAL INFERENCE IN RECOGNITION PROBLEMS

It is worth-while to note that many problems of pattern recognition are also reduced to solving combinatorial tasks. In [59], a logical approach has been suggested to

their solving in the Boolean space of attributes, including inductive inference (obtaining knowledge from data) and deductive inference (using knowledge for computation of some goal attributes values). Data present information concerning individual objects of an experimental selection from a regarded subject area, and knowledge - information about this area as a whole, defining relations between attributes inherent in it. Knowledge is presented by a set of implicative regularities, which can be regarded as a generalization of the functional ones. Both data and knowledge are represented by binary and ternary vectors and matrices, and methods of Boolean functions theory are effectively used for their processing. For instance, many problems of equality transformations and logical inference are reduced to checking CNF for satisfiability.

The methods of the theory of Boolean functions were extended later onto finite predicates [60,61], sectional Boolean vectors and matrices were proposed for representation of data and knowledge, and the suggested approach was generalized for recognition in the space of many-valued attributes [62-64].

The following example illustrates the recognition via deductive inference. Suppose the objects of a regarded area are described in terms of the attributes a, b, c taking values accordingly from the sets $\{1,2,3\}$, $\{1,2,3,4\}$ and $\{1,2\}$, and knowledge is represented by a sectional Boolean matrix $\mathbf{D}$ with rows demonstrating separate regularities-disjuncts (given on the right in the algebraic form).

$$\begin{array}{rcl}
 & \begin{array}{ccc} a & b & c \end{array} & \\
 & \begin{array}{ccccccc} 1 & 2 & 3 & . & 1 & 2 & 3 & 4 & . & 1 & 2 \end{array} & \\
 & \begin{array}{ccccccc} 0 & 0 & 1 & . & 0 & 0 & 1 & 0 & . & 0 & 0 \end{array} & (a=3) \vee (b=3) \\
 \mathbf{D} = & \begin{array}{ccccccc} 0 & 0 & 0 & . & 0 & 0 & 1 & 1 & . & 0 & 1 \end{array} & (b=3) \vee (b=4) \vee (c=2) \\
 & \begin{array}{ccccccc} 0 & 1 & 0 & . & 1 & 1 & 0 & 0 & . & 1 & 0 \end{array} & (a=2) \vee (b=1) \vee (b=2) \vee (c=1) \\
 & \begin{array}{ccccccc} 0 & 0 & 1 & . & 0 & 0 & 0 & 0 & . & 0 & 1 \end{array} & (a=3) \vee (c=2)
 \end{array}$$

Any object of the regarded area has some definite value for every attribute and is represented by a vector with one 1 in each of three sections. It should satisfy all regularities and that means that three columns of $\mathbf{D}$ corresponding to the given values of attributes should cover all rows - have together at least one 1 in each of them. For example, the object presented by the vector 010.0010.01 satisfies this condition, and the object presented by 001.0001.01 - does not satisfy.

Suppose now that some object of the given area possesses the value 1 of the attribute c . Using this information we can reduce the knowledge matrix $\mathbf{D}$, deleting the third row (the corresponding regularity is satisfied) and the last section (corresponding to the attribute c with fixed value). We obtain by that:

$$\begin{array}{rcl}
 & \begin{array}{cc} a & b \end{array} & \\
 & \begin{array}{ccccccc} 1 & 2 & 3 & . & 1 & 2 & 3 & 4 \end{array} & \\
 & \begin{array}{ccccccc} 0 & 0 & 1 & . & 0 & 0 & 1 & 0 \end{array} & \\
 & \begin{array}{ccccccc} 0 & 0 & 0 & . & 0 & 0 & 1 & 1 \end{array} &
 \end{array}$$

0 0 1 . 0 0 0 0

It is easy to conclude that the analysed object has the value 3 of the attribute a and the value 3 or 4 of the attribute b .

A knowledge matrix can be defined either by an expert or be obtained automatically from data - by the algorithm of inductive inference [65]. When data are strongly restricted, it is reasonable to use the syllogistic approach - in this case regularities tie attributes in pairs, and computations are fulfilled much faster than in the general case of arbitrary regularities [66-68]. It is necessary sometimes to take into account the effects of asymmetry - for instance, they could result in impossibility of transforming the regularity "if $a=1$, then $b \neq 3$ " into "if $b=3$, then $a \neq 1$ " [69].

Some expert logical recognition systems were made on the base of the suggested methods [70-72].

7. METHODS OF COMBINATORIAL SEARCH

Solving combinatorial problems over logical matrices is unavoidably connected with exhaustive search, but the latter can be sometimes greatly reduced when taking into account peculiarities of concrete problems.

The most general approach to solving combinatorial problems is based on scanning a search tree that is growing while we look for a solution [73-75]. This process has a recursive character: on each step a considered problem is changed for several alike ones but over reduced data. The descendant problems are regarded in some order and reduced if possible, and less worthwhile of them can be deleted. The processes of decomposing of arising situations are intertwined by that with reducing of them, and it is natural to strive for decomposition into minimum number of more simple situations. In the most effective way this striving can be satisfied only by a profound analysis of concrete problems and developing corresponding formal theories. Some results in this direction are reflected in [76-80].

8. CONCLUSION

Logical matrices can serve as universal means for formulating various problems of logic design and artificial intelligence (and not only for these problems). They are well co-ordinated with features of modern discrete mathematics and peculiarities of computers. They can provide a good base for combinatorial support of processes of logic design and artificial intelligence, enabling to raise essentially the efficiency of solving numerous problems from these areas.

In this article, only some of these problems were described, and next to nothing was said about methods for their solution - because of lack of space. One can find the

appropriate information using the list of publications given below.

REFERENCES

- [1] Aho A., Hopcroft J., Ullman J. The design and analysis of computer algorithms. - Addison-Wesley Publishing Company, 1976.
- [2] Zakrevskij A.D. About the reduction of exhaustive procedures solving some problems of discrete automata synthesis. - Izvestiya VUZov. Radiophysics, 1964, v.7, No 1, pp.166-174 (in Russian).
- [3] Zakrevskij A.D. Machine for the solution of logical problems of the type of the synthesis of relay circuits. - Relay systems and finite automata, Conference Proceedings, Burrough Corp., 1964, pp.544-557.
- [4] Zakrevskij A.D., Toropov N.R. Teaching of pattern recognition in the Boolean space. - Self-learning automatic systems, Moscow: Nauka, 1966, pp.67-72 (in Russian).
- [5] A logical language for representation of algorithms of synthesis of relay devices. - Moscow: Nauka, 1966 (in Russian).
- [6] Zakrevskij A.D. Computations in Boolean spaces. - The logical structure of scientific knowledge, Moscow: Nauka, 1965, pp.292-310 (in Russian).
- [7] Zakrevskij A.D. Algorithmical language LYaPAS and automation of the synthesis of discrete automata. - Tomsk: Tomsk University, 1966 (in Russian).
- [8] Zakrevskij A.D. Programming Boolean computations. - In [9], pp.167-174.
- [9] LYaPAS: A programming language for logic and coding algorithms. (Ed. by M.A.Gavrilov and A.D.Zakrevskij), Academic Press. New-York and London, 1969.
- [10] Zakrevskij A.D., Toropov N.R. A Programming system LYaPAS-M. - Minsk: Nauka i tekhnika, 1978 (in Russian).
- [11] Zakrevskij A.D., Algorithms of discrete automata synthesis. - Moscow: Nauka, 1971 (in Russian).
- [12] Zakrevskij A.D., Balakley L.I. and others. Computer-aided synthesis of asynchronous automata. - Minsk: Nauka i tekhnika, 1975 (in Russian).
- [13] Zakrevskij A.D. Matrix methods for the synthesis of discrete devices on PLA. - VII Triennial World Congress of IFAC, Helsinki, Pergamon Press, v.1, pp.789-794.
- [14] Utkin A.A. Analysis of logic circuits and a technique of Boolean computations. - Minsk: Nauka i tekhnika, 1979.
- [15] Zakrevskij A.D. Logical synthesis of cascade networks. - Moscow: Nauka, 1981 (in Russian).
- [16] Zakrevskij A.D. Logic of recognition. - Minsk: Nauka i tekhnika, 1988 (in Russian).
- [17] Zakrevskij A.D. Testing for identities in Boolean algebra. - In [9], pp.207-213.
- [18] Utkin A.A. Experimental research in satisfiability algorithms. - Avtomatika i vychislitel'naya tekhnika, 1990, No 6, pp.66-74.
- [19] Quine W.V. The problem of simplifying truth functions. - American Mathematical Monthly, 1952, v.59, pp.521-531.
- [20] Dertouzos M. Threshold logic. - The MIT Press, Cambridge, 1965.
- [21] Zakrevskij A.D., Butakov E.A. Some problems of Boolean function implementation by threshold elements. - Izvestiya AN SSSR. Tekhnicheskaya kibernetika, 1964, No 1, pp.39-49 (in Russian).
- [22] Zakrevskij A.D. Coverings of sets. - Trudy SPbTI, 1966, is.48, pp.7-11 (in Russian).
- [23] Zakrevskij A.D. Finding of a set of irredundant coverings. - Trudy SPbTI, 1966, is.48, pp. 73-74 (in Russian).
- [24] Zakrevskij A.D., Ostrovskij V.I. Optimization of the search for a shortest coverage. - Problems of the digital automata synthesis, Moscow: Nauka, 1967, pp.84-95 (in Russian).
- [25] Zakrevskij A.D. Optimal coverage of sets. - In [9], pp.175-191.
- [26] Servit M., Zamazai J. Binat covering problem. Research Report DC-91-05, Czech Technical University, Prague, 1991.
- [27] Zakrevskij A.D. Algorithms for minimization of weakly defined Boolean functions. - Kibernetika, 1965, No 2, pp.53-60 (in Russian).

- [28] Zakrevskij A.D. New algorithms for minimization of weakly specified Boolean functions. - Kibernetika, 1969, No 5, pp.21-28 (in Russian).
- [29] Zakrevskij A.D. Towards minimization of the disjunctive normal forms of Boolean functions. - Izvestiya AN SSSR, Tekhnicheskaya kibernetika, 1970, No 6, pp.102-104 (in Russian).
- [30] Zakrevskij A.D. Matrix means for logical analysis and synthesis of discrete devices on PLAs. - Doklady AN BSSR, 1977, v.21, No 6, pp.991-994 (in Russian).
- [31] Zakrevskij A.D. Logical-combinatorial problems of the cascade circuits synthesis. - Algorithmical methods in some control problems, Kishinev: Shtiintsa, 1979, pp.5-12 (in Russian).
- [32] Zakrevskij A.D. PLAs and matrix logical equations. - Automatentheorie und Ihre Anwendung. Sekt.Math. der Humboldt-Universitet, Berlin, 1984 Januar, s.26-34.
- [33] Zakrevskij A.D. Minimization of the number of literals in a system of Boolean functions. - Doklady AN BSSR, 1978, v.22, No 36, pp.248-250 (in Russian).
- [34] Bochmann D., Zakrevskij A.D., Posthoff Ch. Boolesche Gleichungen. Theorie. Anwendungen. Algorithmen, Berlin, VEB Verlag Technik, 1984.
- [35] Bibilo P.N. Synthesis of combination PLA-structures for VLSI circuits. - Minsk: Nauka i tekhnika, 1992.
- [36] Zakrevskij A.D. Combinatorial theory of logical design. - Avtomatika i vychislitel'naya tekhnika, 1990, No 2, pp.68-79 (in Russian).
- [37] Zakrevskij A.D. Looking for a minimum disjunctive basis of a Boolean matrix. - Doklady AN BSSR, 1978, v.22, No 1, pp.39-41 (in Russian).
- [38] Zakrevskij A.D. On the minimum disjunctive code. - Doklady AN BSSR, 1978, v.22, No 6, pp.516-518 (in Russian).
- [39] Zakrevskij A.D. Logical equations. - Minsk: Nauka i tekhnika, 1975 (in Russian).
- [40] Sasao T. AND-EXOR expressions and their optimization. - In Sasao (ed.) Logic Synthesis and Optimization, Kluwer Academic Publishers, 1993, pp.287-312.
- [41] Mishina A.P., Proskurjakov I.V. Higher algebra. - Moscow: Nauka, 1965 (in Russian).
- [42] Zakrevskij A.D. Implementation of incompletely specified Boolean functions by Zhegalkin polynomials. - Doklady AN B, v.39, No 3, pp.5-8 (in Russian).
- [43] Zakrevskij A.D. Fast algorithm for the synthesis of Zhegalkin polynomials implementing incompletely specified Boolean functions. - Avtomatika i vychislitel'naya tekhnika, 1995, No 4, pp.36-41 (in Russian).
- [44] Zakrevskij A.D., Toropov N.R. Optimizing polynomial implementation of incompletely specified Boolean functions. - Proceedings of the International Conference on Computer-Aided Design of Discrete Devices, Minsk-Szczecin, 1995, v.2, pp.93-98.
- [45] Zakrevskij A.D. Minimum implementation of partial Boolean functions by Zhegalkin polynomials. - Avtomatika i telemekhanika, 1996, No 5, pp.134-140 (in Russian).
- [46] Zakrevskij A. Minimum polynomial implementation of systems of incompletely specified Boolean functions. - Proceedings IFIP WG 10.5 Workshop on Applications of the Reed-Muller Expansion in Circuit Design, 1995, Makuhari, Chiba, Japan, pp.250-256.
- [47] Zakrevskij A.D. Minimization of Zhegalkin polynomial representation of a system of Boolean functions. - Doklady AN B, 1995, v.39, No 6, pp.11-14 (in Russian).
- [48] Zakrevskij A.D. Minimizing AND-EXOR implementation of systems of weakly specified Boolean functions. - Proceedings of the International Conference on Computer-Aided Design of Discrete Devices, Minsk-Szczecin, 1995, v.2, pp.9-16.
- [49] Zakrevskij A.D. Looking for shortest solutions of linear logical equations: theory and applications in logic design. - 2.Workshop "Boolesche Probleme", 19./20. September 1996, Freiberg/Sachsen, pp.63-69.
- [50] Zakrevskij A.D., Jaroszewicz S., Yanushkevich S.N. Minimization of Reed-Muller expansions for systems of incompletely specified MVL functions. - Proceedings of the Third International Symposium on Methods and Models in Automation and Robotics, 10-13 September 1996, Miedzyzdroje, Poland, v.3, pp.1085-1090.
- [51] Zakrevskij A.D. Parallel automaton. - Doklady AN BSSR, 1984, v.28, No 8, pp.717-719 (in Russian).
- [52] Zakrevskij A.D. Towards the theory of parallel logical control algorithms. - Izvestiya AN SSSR. Tekhnicheskaya kibernetika, 1989, No 5, pp.179-191 (in Russian).
- [53] Zakrevskij A. Parallel logical control algorithms: verification and hardware implementation. - Computer Science Journal of Moldova, 1996, v.4, No 1, pp.3-19.
- [54] Pottosin Ju.V. Covering a graph by complete bipartite subgraphs. - Design of discrete systems, Minsk, Institute of Engineering Cybernetics, pp.72-84 (in Russian).
- [55] Zakrevskij A.D. State assignment problem in hardware implementation of parallel algorithms of logical control. - Minsk, Institute of Engineering Cybernetics, 1993, Pre-print N 16.
- [56] Zakrevskij A.D. On minimizing the coding matrix for partial states. - Doklady AN BSSR, 1993, v.37, No 6, pp.8-10 (in Russian).
- [57] Zakrevskij A.D. The analysis of concurrent logic control algorithms. - Lecture Notes in Computer Science, Fundamentals in Computation Theory. Springer Verlag, Vol. 278, 1987. P.497-500.
- [58] Hack M. Analysis of production schemata by Petri nets. - Project MAC TR-94. Cambridge, 1972.
- [59] Zakrevskij A.D. Revealing of implicative regularities in the Boolean space of attributes and the pattern recognition. - Kibernetika, 1982, No 1, pp.1-6 (in Russian).
- [60] Zakrevskij A.D. Disjunctive normal forms of finite predicates. - Problemy bioniki, is.40, 1988, pp.3-9 (in Russian).
- [61] Zakrevskij A.D. Matrix means of logical inference in finite predicates. - Philosophical grounds of non-classic logics, Moscow, Institute of Philosophy, 1990, pp.70-80 (in Russian).
- [62] Zakrevskij A.D. Logical recognition by deductive inference based on finite predicates. - Proceedings of the Second Electrotechnical and Computer Science Conference ERK'93, Slovenia Section IEEE, Ljubljana, v.B, 1993. P.197-200.
- [63] Zakrevskij A.D. Method of logical recognition in the space of multivalued attributes. - Proceedings of the Third Electrotechnical and Computer Science Conference ERK'94, Slovenia Section IEEE, Ljubljana, v.B, 1994. P.3-5.
- [64] Zakrevskij A. Logical recognition in the space of multivalued attributes. - Computer Science Journal of Moldova, 1994, v.2, No 2, pp.169-184.
- [65] Zakrevskij A.D. Knowledge representation and logical inference in the space of many-valued attributes. - Logic and Computer, 2, Moscow: Nauka, 1995, pp.3-15 (in Russian).
- [66] Zakrevskij A.D. Towards the formalization of polysyllogistics. - Logical inference, Moscow: Nauka, 1979, pp.300-309 (in Russian).
- [67] Zakrevskij A.D. Recognition in the space of many-valued attributes: a syllogistic approach. - Doklady AN BSSR, 1995, v.39, No 2, pp.9-12 (in Russian).
- [68] Zakrevskij A.D. Fast algorithms for reliable logical recognition. - Izvestiya AN B, seriya fiziko-tekhnicheskikh nauk, 1995, No 1, pp.46-49 (in Russian).
- [69] Zakrevskij A.D. Appreciating asymmetry of implicative regularities under logical recognition. - Proceedings of the Second International Conference on Applications of Computer Systems. Szczecin, Poland, 1995, pp.75-79.
- [70] Zakrevskij A.D., Levchenko V.I., Pecherskij Yu.N. Instrumental expert diagnostic system EDIP, based on finite predicates. - Application systems of artificial intelligence. Mathematical researches, is.123. - Kishinev: Shtiintsa, 1991, pp.24-40 (in Russian).
- [71] Zakrevskij A.D. EXSYLOR - an expert system for logical recognition. - Upravlyayushchie systemy i mashiny, 1992, No 5/6, pp.118-124 (in Russian).
- [72] Zakrevskij A.D. An expert system of logical recognition as means for training in logical inference. - Research in Logic, is.3, Moscow: Nauka, 1995, pp.178-180 (in Russian).
- [73] Zakrevskij A.D. On approximating methods of the solution of logical problems. - Problems of the digital automata synthesis, Moscow: Nauka, 1967, pp.5-13 (in Russian).
- [74] Nilsson N. Problem-solving methods in artificial intelligence. - New York, 1971.

- [75] Thayse A., Gribomont P. and others. Approche logique de l'intelligence artificielle. - Dunod informatique.
- [76] Zakrevskij A.D., Bibilo P.N., Dudkin A.A., Shneider A.A. A program package for the solution of combinatorial problems over Boolean and ternary matrices. - Upravlyayushchie systemy i mashiny, 1984, No 3, pp.115-118 (in Russian).
- [77] Zakrevskij A.D., Bibilo P.N., Dudkin A.A. A program package for the synthesis of combinational circuits in the basis of PLAs. - Upravlyayushchie systemy i mashiny, 1985, No 1, pp.27-29 (in Russian).
- [78] Zakrevskij A. CAD of discrete devices implementing parallel logical control algorithms. - Proceedings of the Second International Symposium on Methods and Models in Automation and Robotics, 1995, Miedzyzdroje, Poland, pp.803-808.
- [79] Bibilo P.N., Zakrevskij A.D., Kardash S.N., Karkotskaja I.P., Romanov V.I. A system for logical design of custom-made VLSI chips on the base of regular matrix structures. - Mikroelektronika, 1995, v.24, No 5, pp.349-354.
- [80] Bibilo P.N. Silicon compilation of custom-made VLSI circuits. - Minsk, 1996.