INFORMATICA, 2024, Vol. 35, No. 1, 1–20
© 2024 *Vilnius University*
DOI: https://doi.org/10.15388/24-INFOR542

Shadow Minimization Boolean Function Reconstruction

Levon ASLANYAN[1,*], Gyula KATONA[2], Hasmik SAHAKYAN[1]

[1] *Institute for Informatics and Automation Problems of NAS RA,*
1 P. Sevak str., 0014 Yerevan, Armenia
[2] *Alfréd Rényi Institute of Mathematics, Reáltanoda utca 13-15, 1053 Budapest, Hungary*
e-mail: lasl@sci.am, katona.gyula.oh@renyi.hu, hsahakyan@sci.am

Received: November 2023; accepted: February 2024

Abstract. A special class of monotone Boolean functions coming from shadow minimization theory of finite set-systems – KK-MBF functions – is considered. These functions are a descriptive model for systems of compatible groups of constraints, however, the class of all functions is unambiguously complex and it is sensible to study relatively simple subclasses of functions such as KK-MBF. Zeros of KK-MBF functions correspond to initial segments of lexicographic order on hypercube layers. This property is used to simplify the recognition. Lexicographic order applies priorities over constraints which is applicable property of practices. Query-based algorithms for KK-MBF functions are investigated in terms of their complexities.

Key words: monotone Boolean function, reconstruction, lexicographic order, shadow, KK-MBF class.

1. Introduction

Many problems with monotone Boolean functions (MBFs) appear in logical and physical level design of systems (Aslanyan *et al.*, 2019), but also in artificial intelligence (Aslanyan and Sahakyan, 2009), data science and computational learning theory (Aslanyan *et al.*, 2023a), hypergraph theory (Sahakyan, 2023; Sahakyan and Aslanyan, 2017) and other areas (e.g. Carlet *et al.*, 2016; Kulhandjian *et al.*, 2019; Crawford-Kahrl *et al.*, 2022; Kabulov and Berdimurodov, 2021; Zhang *et al.*, 2022). MBFs are used to encode extremely important constructions in various combinatorial optimizations providing a natural way of describing compatible subsets of sets of finite constraints (see e.g. Aslanyan *et al.*, 2023b; Tennakoon *et al.*, 2021; Damásdi *et al.*, 2021).

Let there be n independent constraints in an optimization problem with a target function $\nu(\alpha)$, $\alpha \in P([n])$, where $[n] = \{1, 2, \dots, n\}$, and $P([n])$ is the power set of $[n]$; α encodes a subset of constraints, and the values of ν are valid only on compatible subsets. Sometimes the compatibility check of subsets α becomes a separate intractable problem. The question how to minimize the checks/tests in a global optimization procedure arises

[*] Corresponding author.

naturally. Obviously, if there is a compatible subset α_1 given, and $\alpha_2 \subseteq \alpha_1$, then α_2 is compatible. The overall structure of compatibility is given by these inclusions and then by monotone Boolean functions, and the idea behind the recognition of these functions is to find the maximum compatible subsets of constraints, applying fewer checks and procedures in the optimality search processes.

A number of applications (e.g. wireless sensor networks, dead-end tests of tables, data mining) (Kovalerchuk and Delizy, 2005; Kulhandjian *et al.*, 2019; Aslanyan and Sahakyan, 2009; Aslanyan *et al.*, 2019) use optimization with MBF, where MBFs are represented by hypercube constructions such as chains and anti-chains (Freixas, 2022; Griggs *et al.*, 2023). Other similar applications with MBF can be added to this list (Clements, 1973; Tennakoon *et al.*, 2021; Damásdi *et al.*, 2021; Carlet *et al.*, 2016).

There is a number of known effective tools and methods for analysing MBFs (Korobkov, 1965; Hansel, 1966; Tonoyan, 1979; Gainanov, 1984) and new approaches are constantly being sought, investigated, and applied (e.g. Carlet *et al.*, 2016; Boros and Hammer, 2002; Lange and Vasilyan, 2023; Sahakyan *et al.*, 2022; Balogh *et al.*, 2021; Bezrukov *et al.*, 2023). Open problems in this area include the reconstruction problem of bounded classes of Boolean functions with randomization of queries and functions, and the use of cube-splitting and chain-splitting technique of the Boolean domain (Blum, 2003; Jackson *et al.*, 2011; O'Donnell and Servedio, 2005; Aslanyan and Sahakyan, 2017; Black *et al.*, 2023; Boros *et al.*, 1991).

A well-known approach concerning MBFs recognition is query-based identification – recognition of an unknown MBF of n variables using an oracle and membership queries to it. Hansel's algorithm (Hansel, 1966), based on partitioning the binary cube into a special set of non-intersecting chains, provides optimal reconstruction in the sense of Shannon complexity for the whole class of MBFs. In practical algorithmic implementations, it is even not necessary to build and store all Hansel chains in computer memory (Tonoyan, 1979), which solves the memory limitation problem in applications. But the computational complexity remains.

In order to obtain solutions for bounded classes of MBFs, it is necessary to find a way to the structural properties of these classes (Braverman *et al.*, 2022; Chao and Yu, 2023; Lovász, 2007). The research objective of this paper is related to the well-known Kruskal-Katona theorem (Kruskal, 1963; Katona, 1968, 1987; Sales and Schülke, 2022; Frankl and Katona, 2021), and the class of *KK-MBF* functions, related to this theorem, which describes the exact optimal monotone constructions of shadow minimization (constraint minimization) (Braverman *et al.*, 2022; Chao and Yu, 2023; Jung, 2023; Madden, 2023) and existence of Sperner systems for a given set of parameters. In this way, *KK-MBF* class of MBFs becomes a special and attractive class for recognition, and this is the theoretical value of *KK-MBF* recognition. On the other hand, the practical value of *KK-MBF* recognition consists in the following. *KK-MBF*s appear when compatibility is not only controlled by the inclusion of subsets of constraints. Suppose we are given two constrained, k-subsets α_1 and α_2, such that α_2 alphabetically proceeds α_1. In some applications, constraint compatibility should have the following property: compatibility of α_2 is a simple consequence of the compatibility of α_1 because of the alphabetical priorities of constraints. In this case, the structure of constraint compatibility is given by *KK-MBF* functions.

In this research, we investigate the *KK-MBF* class, focusing on query-based recognition algorithms. Example subclasses of the class are also presented and discussed.

The rest of the paper is organized as follows. Section 2 provides necessary definitions, preliminaries, and basic concepts. The *KK-MBF* class, its basic properties/constraints, as well as recognition procedures, are introduced in Section 3. Section 4 discusses the cardinality issues of the class. Special subclasses are considered in Section 5. The paper ends with the concluding remarks.

2. Preliminaries

2.1. *Monotone Boolean Function Recognition*

Let $B^n = \{(x_1, \dots, x_n) \,|\, x_i \in \{0, 1\},\ i = 1, \dots, n\}$ denote the set of vertices of the n-dimensional binary (unit) cube. Let $\alpha = (\alpha_1, \dots, \alpha_n)$ and $\beta = (\beta_1, \dots, \beta_n)$ be two vertices of B^n. α *precedes* β (by *component-wise order*), denoted as $\alpha \preccurlyeq \beta$, if and only if $\alpha_i \leqslant \beta_i$ for $1 \leqslant i \leqslant n$. α and β are *comparable* if $\alpha \preccurlyeq \beta$ or $\beta \preccurlyeq \alpha$, otherwise, they are *incomparable*. A set of incomparable vertices in B^n is also called a *Sperner* family. A set $\{\alpha^1, \dots, \alpha^k\}$ of elements of B^n is a growing *chain* if $\alpha^i \prec \alpha^j$ for $1 \leqslant i < j \leqslant k$. A chain is *simple* if $\alpha^i \prec \alpha^j$ and there is no α^r such that $\alpha^i \prec \alpha^r \prec \alpha^j$.

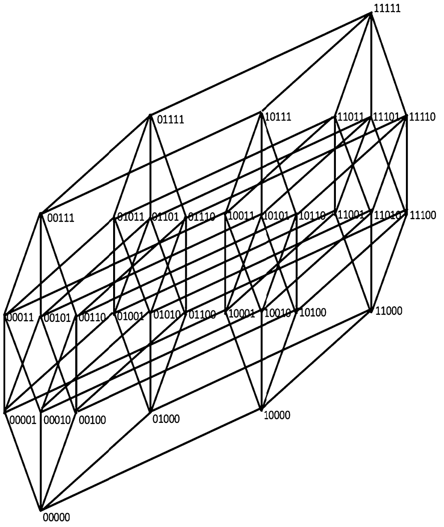
The *serial number* of the vertex $\alpha = (a_1, \dots, a_n)$ is the natural number $a_1 2^{n-1} + a_2 2^{n-2} + \dots + a_n 2^0$, whose binary representation is $a_1 a_2 \dots a_n$.

We will also use the *lexicographic order* of vertices: α *precedes* β lexicographically ($\alpha \preccurlyeq_{\text{lex}} \beta$) if either there exists an integer k, $1 \leqslant k \leqslant n$, such that $a_k < b_k$ and $a_i = b_i$ for $i < k$, or $\alpha = \beta$.

In literature there are different, sometimes confusing definitions of lexicographic order (Schröder, 2016). Mathematical definition is set-theoretical that uses the value 1 to code the presence of an element in a subset: $x <_{\text{lex}} y$ if $\min(x \Delta y) \in x$. Appearance of elements in a subset is coded by a binary vector, and vector α precedes vector β when α precedes β alphabetically. Here $1 < 0$. When we apply to combinatorial settings, the hypercube is considered, and lexicographical order of vertices of B^n, defined alphabetically, uses relation $0 < 1$. We will use the standard set-theoretical definition indicated as *lex* in Fig. 1. *revlex* is the reverse of this, i.e. (00000, 00001, ..., 11111). *colex* in set-theoretical settings means: $x <_{\textit{co-lex}} y$ if $\max(x \Delta y) \in y$. This is convenient when we add a new dimension to the vectors intending to continue the ordered sequence when the initial segment remains unchanged. But as we see in Fig 1, *colex*, unlike the *lex* or *revlex*, preserves neither the order of numerical values nor the order on the layers of the hypercube graphically, in the Hasse diagram. The *sim* (simplicial) order lists hypercube vertices layer after the layer (see the next page for the concept of a layer); in each layer, vertices are ordered according to their serial number. We will use the *revlex* basically, but when there is no confusion we will refer to it simply as *lex*.

Lexicographic, co-lexicographic and simplicial orders are shown in Fig. 1 on the set of vertices of B^5.

lex	#	co-lex	#
11111	31	00000	00
11110	30	10000	16
11101	29	01000	08
11100	28	11000	24
11011	27	00100	04
11010	26	10100	20
11001	25	01100	12
11000	24	11100	28
10111	23	00010	02
10110	22	10010	18
10101	21	01010	10
10100	20	11010	26
10011	19	00110	06
10010	18	10110	22
10001	17	01110	14
10000	16	11110	30
01111	15	00001	01
01110	14	10001	17
01101	13	01001	09
01100	12	11001	25
01011	11	00101	05
01010	10	10101	21
01001	09	01101	13
01000	08	11101	29
00111	07	00011	03
00110	06	10011	19
00101	05	01011	11
00100	04	11011	27
00011	03	00111	07
00010	02	10111	23
00001	01	01111	15
00000	00	11111	31



sim	#
11111	31
11110	30
11101	29
11011	27
10111	23
01111	15
11100	28
11010	26
11001	25
10110	22
10101	21
10011	19
01110	14
01101	13
01011	11
00111	07
11000	24
10100	20
10010	18
10001	17
01100	12
01010	10
01001	09
00110	06
00101	05
00011	03
10000	16
01000	08
00100	04
00010	02
00001	01
00000	00

Fig. 1. Lexicographic, co-lexicographic and simplicial orders of vertices of B^5. The sign # marks the natural order which coincides with the numerical value of binary representation of vertices.

We define also *partition/splitting* of B^n into two $(n-1)$-dimensional sub-cubes according to the values of the binary variables; for arbitrary x_i:

$$B^{n-1}_{x_i=0} = \big\{(x_1, \dots, x_n) \in B^n \,\big|\, x_i = 0\big\} \quad \text{and}$$
$$B^{n-1}_{x_i=1} = \big\{(x_1, \dots, x_n) \in B^n \,\big|\, x_i = 1\big\}.$$

Any subset $\mathcal{M} \subseteq B^n$ will be partitioned into

$$\mathcal{M}_{x_i=1} \subseteq B^{n-1}_{x_i=1}, \quad \text{and} \quad \mathcal{M}_{x_i=0} \subseteq B^{n-1}_{x_i=0}.$$

B^n can also be partitioned according to a set of variables. Partitioning according to variables $x_{i_1}, \dots, x_{i_k}$, we get 2^k number of $(n-k)$-dimensional sub-cubes, where in each of them the values of $x_{i_1}, \dots, x_{i_k}$ are fixed in an appropriate way; for example,

$$B^{n-k}_{x_{i_1}=1,\dots,x_{i_k}=1} = \big\{(x_1, \dots, x_n) \in B^n \,\big|\, x_{i_1} = 1, \dots, x_{i_k} = 1\big\}.$$

Figure 2 illustrates the split of the B^6 according to two variables.

Let $L_k = \big\{(x_1, \dots, x_n) \in B^n \,\big|\, \sum_{i=1}^{n} x_i = k\big\}$. We call L_k the k-th layer of B^n.

The *shadow* $\delta^i\mathcal{M}$ of $\mathcal{M} \subseteq B^n$ is the set of vertices of L_i, which are less than some vertex of $\mathcal{M}$.

In case all the vertices of $\mathcal{M}$ are from the same layer, e.g. from the k-th layer, the lower (upper, respectively) shadow of $\mathcal{M}$ is $\delta^{k-1}\mathcal{M}$ ($\delta^{k+1}\mathcal{M}$, respectively), i.e. the set

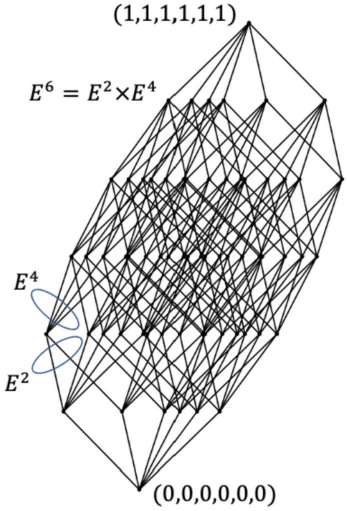


Fig. 2. Split of the B^6 into the $B^4 \times B^2$.

of vertices from the $(k-1)$-th layer, which are less than some vertex of $\mathcal{M}$ (from the $(k+1)$-th layer, which are greater than some vertex of $\mathcal{M}$, respectively).

Boolean function $f : B^n \to \{0, 1\}$ is called *monotone* if for every two vertices $\alpha, \beta \in B^n$, if $\alpha \prec \beta$, then $f(\alpha) \leqslant f(\beta)$. Vertices of B^n, where f takes the value "1", are called *units* or *true points* of the function; vertices, where f takes the value "0", are called *zeros* or *false points* of the function. α^1 is a *lower unit* (or *minimal true point*) of the function if $f(\alpha^1) = 1$, and $f(\alpha) = 0$ for every $\alpha \in B^n$, such that $\alpha \prec \alpha^1$. α^0 is an *upper zero* (or *maximal false point*) of the function if $f(\alpha^0) = 0$, and $f(\alpha) = 1$ for every $\alpha \in B^n$ such that $\alpha^0 \prec \alpha$. $\min T(f)$ and $\max F(f)$ denote the sets of minimal true points and maximal false points, respectively. Obviously, $\min T(f)$ and $\max F(f)$ are Sperner families in B^n.

Formally, the work with MBFs started in 1897, with the issue of counting their number (Dedekind, 1895). The first algorithmic and complexity-related considerations belong to Korobkov (1965), where, in particular, the valuable concept of *resolving subsets* was introduced. The final asymptotic estimate about the number of MBFs of n variables was obtained in Korshunov (1981, 2003). The technique on how to introduce and analyse MBFs, is basically presented in Hansel (1966), Korshunov (2003), Lovász (2007).

The Hansel chain structure (Hansel, 1966) was invented in 1966 and played one of the central roles in MBF-related algorithmic techniques.

The next valuable step towards this was taken by Tonoyan (1979), who introduced a set of simple procedures (chain algebra) that serve all the actual queries about Hansel chains, providing a technical solution to all the problems related to algorithms with Hansel chains, without constructing and keeping them in computer memory. A slightly modified and simplified version of Sokolov (1987) is presented using tools such as: enumeration of all chains, and a procedure of finding the i-th vertex of the j-th chain.

A recurrent step of constructing Hansel chains in B^n is the following (illustrated graphically in Fig. 3). Let l_0 be an arbitrary chain in $B^{n-1}_{x_n=0}$, and let l_1, geometrically, be the same

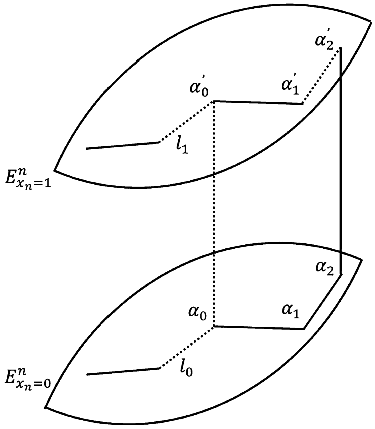


Fig. 3. Recurrent step of constructing Hansel chains in B^n.

chain in $B^{n-1}_{x_n=1}$. Thorough the chain constructions, all pairs of chains like l_0 and l_1 are modified according the following rule. Chain l_1 is shortened by removing the last edge (α_1', α_2') from it, and chain l_0 is updated by adding a new edge (α_2, α_2') to it. This construction provides one of the basic properties of Hansel chains in B^n: the relative complement of three consecutive vertices of extended chain l_0' belongs to the shortened chain l_1'. Relative complement is a vertex α, that together with the growing chain of tree consecutive vertices α_1, α_2, α_3 composes a subcube of dimension 2.

We aim at extending this picture of chain-split to the lexicographic order of vertices in B^n. At least, we note that the longest Hansel chain corresponds to the chain consisting of the first vertices of layers of B^n under the lexicographic ordering. The mirrored chain of the last vertices under the lexicographic ordering will be considered as well.

In the MBF recognition problem using membership queries, the overall goal is to determine an unknown MBF of n variables using as few oracle queries (or tests) as possible. The function can be fully recognized by finding all its upper zeros (and/or lower units) (Korobkov, 1965). The Shannon complexity of finding all upper zeros (lower units) of an arbitrary monotone Boolean function of n variables is $C_n^{\lfloor n/2\rfloor} + C_n^{\lfloor n/2\rfloor+1}$ (Hansel, 1966).

Another recognition structure is used in Sokolov (1987) and in its extensions. For even n, B^n is split according to two variables and the recognition starts in the two middle layer sub-cubes of this construction. For odd n, firstly B^n is split according to one variable, then as each sub-cube now has an even size, the procedure for even sizes is applied. This provides optimal recognition of all MBFs in the sense of Shannon complexity. Unfortunately, while simple and attractive, this approach cannot be used as a practical algorithm. Finally, it is worth to mention the work (Gainanov, 1984) that considers not the Shannon complexity, but the individual complexity of MBF given by its resolving set size.

2.2. *Constraint Monotone Boolean Function Recognition*

In general, tasks related to the recognition of MBFs may have different formulations. One objective is to recognize a particular unknown function, knowing that it belongs to the class of MBFs or to one of its subclasses. Another goal is to start with partial knowledge about the unknown function, trying to complete the information. One more case is when the number of queries is restricted by some number k and the goal is to maximize the recognized part of the function (Sahakyan *et al.*, 2022). Similar problems can be formulated for specific classes of Boolean functions. Examples of classes are as follows:

KK-MBF Kruskal-Katona MBFs arise as a result of the shadow minimization theorem (Kruskal, 1963; Katona, 1968, 1987). *KK-MBF*s are monotone Boolean functions but they also intersect the cube layers along their initial segments of the lexicographic order. The complement of the *KK-MBF* area in B^n has a similar property; it is related to the initial segments of the reverse-lexicographic order, and they are anti-monotone.

Symmetric MBF This is a trivial class of functions that takes a constant value on the cube layers. Trivial, but these functions are practically important. Examples are majority functions, parity functions, and others (Nosov, 2023).

Threshold MBF Functions are defined by a linear inequality of weighted sums of variables.

Matroid MBF Monotone Boolean function f is called a matroid function if for each $\alpha, \beta \in \min T(f)$ with $\alpha_i = 1$, $\beta_i = 0$, there exists a coordinate j with $\alpha_j = 0$, $\beta_j = 1$ such that vertex α', obtained from α by replacing α_i with 0 and α_j with 1, belongs to $\min T(f)$.

The combinatorial complexity of reconstruction in these and other subclasses of MBFs is not well studied. For example, monotone Boolean functions, with zeros and units separated by two middle layers of the cube, are the most difficult functions for query-based reconstruction when only the monotonicity of the function is given. But if it is known that the function belongs to the class of symmetric functions, the reconstruction of this function can be done by n queries. The same function also belongs to the *KK-MBF* class. Knowing more MBF cases that are practically reconstructible enables system proctoring and application problem solving.

3. KK-MBF Recognition

In this section, we consider a special class of monotone Boolean functions, related to the well-known Kruskal-Katona theorem, which describes the exact optimal monotone constructions of shadow minimization (constraint minimization) problem, and the existence problem of Sperner systems for a given set of parameters. Although the reconstruction problem for general MBFs is intractable, the problem itself is extremely important in system design and implementation. The reconstruction problem for such subsets of MBFs is insufficiently investigated, and in this paper, for the first time in our knowledge, the problem of recognition of functions of class *KK-MBF* is investigated.

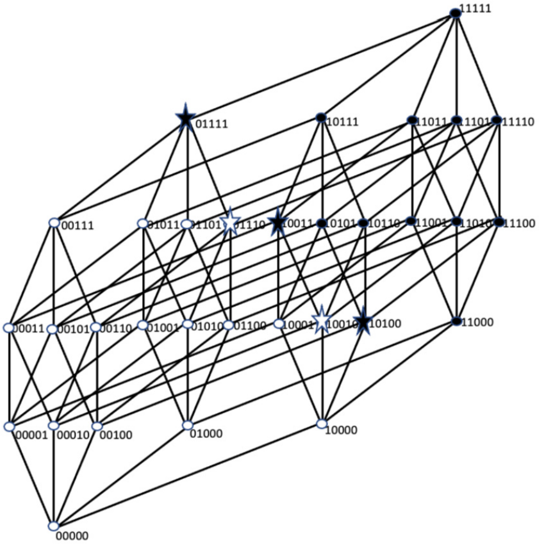


Fig. 4. *KK-MBF* function on B^5 with $p_2 = 2$ (vertices 11000 and 10100), $p_3 = 1$ (vertex 10011), and $p_4 = 1$ (vertex 01111). Uncoloured vertices show zeros, and the blue vertices – units of the function. Stars indicate the corner points.

3.1. *Introduction to KK-MBF Type Functions*

Definition 1. Let f be a monotone Boolean function on B^n. f is called a *KK-MBF* type function if zeros of f on the layers of B^n compose initial segments of the reverse-lexicographic order (an example is given in Fig. 4, where uncoloured vertices correspond to zeros of the function).

The initial formulations of the shadow theorem in Kruskal (1963), Katona (1968, 1987) are given in terms of co-lexicographic order, but this framework was later simplified to the simple lexicographic ordering (Aslanyan, 1979). The basic result was obtained for two neighbour layers of the cube. The name *KK* now refers to an extension of the basic result of the shadow minimization theorem to many layers of B^n, as well as to results on the existence of Sperner families (Clements, 1973; Daykin *et al.*, 1974). Usually, a *KK-MBF* function f is given through its characteristics, $\#\min T(f) = \langle p_{i_1}, p_{i_2}, \dots, p_{i_r}\rangle$, where p_{i_j} is the number of lower units of f on the i_j-th layer. An example is given in Fig. 4.

3.2. *Resolving Sets for KK-MBF*

First, let us introduce some general concepts from the field of reconstruction of Boolean functions (Korobkov, 1965). Suppose we are given a certain class $\mathcal{S}$ of Boolean functions and $f \in \mathcal{S}$.

Definition 2. A set of vertices $G(f, \mathcal{S})$ of B^n is called a *resolving set* for the pair $(f, \mathcal{S})$, if from the fact that:

a) a function g belongs to $\mathcal{S}$, and
b) $g(\alpha) = f(\alpha)$ for $\alpha \in G(f, \mathcal{S})$,

it follows that $g = f$.

It follows from the definition that to reconstruct a function it is sufficient to determine its values on some of its resolving sets.

A resolving set $G(f, \mathcal{S})$ is called a *deadlock resolving set* for $(f, \mathcal{S})$, if no subset of it is resolving for the pair $(f, \mathcal{S})$.

When $\mathcal{S}$ is the set of all monotone Boolean functions, then every function $f \in \mathcal{S}$ has a unique deadlock resolving set included in all its resolving sets; this is the set $G(f) = \min T(f) \cup \max F(f)$ (Korobkov, 1965). Thus, for a general MBF, the concept of deadlock resolving set is given by the set of all upper zeros and lower units of the function, which represent two interrelated Sperner systems.

It should be noted that this is not true for other Boolean functions and classes, for example, for the class of symmetric Boolean functions there are no unique deadlock resolving sets.

In this section, we investigate the existence of a deadlock resolving set for *KK-MBF* functions.

We formulate two simple properties for *KK-MBF* type functions f and call them *horizontal* and *vertical* conditions, denoting them as *Cond-h* and *Cond-v*.

Cond-h:

(1) if $f(\alpha) = 0$ for a vertex α of some layer L_k, then $f(\beta) = 0$ for all $\beta \in L_k$ reverse-lexicographically preceding α ($\beta \preccurlyeq_{\text{revlex}} \alpha$),
(2) if $f(\alpha) = 1$ for a vertex α of some layer L_k, then $f(\beta) = 1$ for all β of L_k lexicographically preceding α ($\beta \preccurlyeq_{\text{lex}} \alpha$).

Cond-v:

(1) if $f(\alpha) = 0$ for a vertex α, then $f(\beta) = 0$ for all β, $\beta \preccurlyeq \alpha$ (component-wise order),
(2) if $f(\alpha) = 1$ for a vertex α, then $f(\beta) = 1$ for all β, $\beta \succcurlyeq \alpha$ (component-wise order).

These conditions, applied recursively, define a *domain* for each vertex $\alpha \in f$; denote it by $d(f, \alpha)$. Domain is downward when $f(\alpha) = 0$ and upward if $f(\alpha) = 1$. A simple characterization of the domains can be given in terms of natural order of vertices (their serial numbers).

Proposition 1. *For $\alpha \in B^n$ and $f \in KK$ - MBF, $d(f, \alpha)$ is composed by*:

- *vertices of B^n with a higher serial number, when $f(\alpha) = 1$, and*
- *vertices of B^n with a smaller serial number, when $f(\alpha) = 0$.*

Proof.

- Suppose that $f(\alpha) = 1$. Consider the partition $B^{n-1}_{x_1=0} \cup B^{n-1}_{x_1=1}$ of B^n by the first variable (see Fig. 4), and apply an inductive inference by the number of variables. Consider 2 cases:

1) $\alpha \in B^{n-1}_{x_1=1}$; then, $d(f, \alpha) \subseteq B^{n-1}_{x_1=1}$, and by the induction hypothesis, the proposition is correct.
2) $\alpha \in B^{n-1}_{x_1=0}$; in this case, $d(f, \alpha) \subseteq B^{n-1}_{x_1=0} \cup B^{n-1}_{x_1=1}$. For the part in $B^{n-1}_{x_1=0}$, all vertices have higher serial number according to the induction hypothesis. As for the part in $B^{n-1}_{x_1=1}$, the proposition simply follows from the evident fact, that the vertex of the smallest serial number in $B^{n-1}_{x_1=1}$ is

   $(1\overbrace{00\ldots0}^{n-1})$ (equals to 2^{n-1}), while the vertex of the highest serial number in $B^{n-1}_{x_1=0}$ is $(0\overbrace{11\ldots1}^{n-1})$ (equals $2^{n-1}-1$).

- Consideration of the case $f(\alpha) = 0$ is similar. □

Note that not all vertices of B^n with a higher (smaller) serial number than α, when $f(\alpha) = 1$ ($f(\alpha) = 0$) are part of the $d(f, \alpha)$.

We also define the notion of *corner points* for *KK-MBF* type functions.

Definition 3. A zero vertex α of a function f is called a *zero-corner point* if:

(1) $f(\beta) = 1$ for all β from the same layer, such that $\beta \prec_{\text{lex}} \alpha$, and
(2) $f(\beta) = 1$ for all β, $\alpha \prec \beta$ (component-wise order).

Similarly, a unit vertex α of a *KK-MBF* type function f is called *one-corner point* if:

(1) $f(\beta) = 0$ for all β from the same layer such that $\beta \succ_{\text{lex}} \alpha$, and
(2) $f(\beta) = 0$ for all β, $\beta \prec \alpha$ (component-wise order).

Let $z(f)$ denote the set of all zero-corner points, and $o(f)$ denote the set of all one-corner points of function f.

Summarizing all the above reasoning, we formulate the following statement.

Proposition 2. *Every monotone Boolean function f of class KK-MBF has a unique deadlock resolving set which is included in all its resolving sets. This deadlock resolving set for f is the set $G(f) = z(f) \cup o(f)$.*

Proof. First note that any zero-corner point α, as a point of $f(\alpha) = 0$ has its domain $d(f, \alpha)$ filled with zero values. But as a corner point, $d(f, \alpha)$ cannot be strongly included in any other domain $d(f, \beta)$ of a point with $f(\beta) = 0$. So the domain of a zero-corner point α is the deadlock domain filled with zero values. The same note is valid for one-corner points. Now, $\bigcup_{\alpha \in z(f) \cup o(f)} d(f, \alpha) = B^n$, otherwise, if there exists $\beta \in B^n$ out of $\bigcup_{\alpha \in z(f) \cup o(f)} d(f, \alpha)$, then β may generate a new domain, or it will properly include some existing domain in it. It follows that $z(f) \cup o(f)$ is a resolving set. Given that the domains of 0- and 1-corner points cannot intersect by definition, and each point $\alpha \in z(f) \cup o(f)$ is represented only by itself, and by its domain, the proof is complete. □

For a general MBF, it is well-known that $|\min T(f) \cup \max F(f)|$ can reach the value $C_n^{\lfloor n/2 \rfloor} + C_n^{\lfloor n/2 \rfloor + 1}$ and as a consequence, recognition of these functions cannot be done with less complexity. Our first notion about the *KK-MBF* recognition complexity is an upper bound obtained for the value $|z(f) \cup o(f)|$. Due to the resolving property of the set $z(f) \cup o(f)$, the estimate will show the number of required tests for reconstruction of *KK-MBF* functions.

Proposition 3. *For monotone Boolean functions f of class KK-MBF $|z(f) \cup o(f)| \leqslant 2(n-1)$.*

Proof. According to the condition *Cond-h*, on each layer of B^n there can be only one pair α, β of neighbour vertices, for which $f(\alpha) = 0$ and $f(\beta) = 1$. Theoretically, they also may be corner points; hence, $|z(f) \cup o(f)| \leqslant 2(n+1)$. Exceptionally, on each of the 0-th and n-th layers there can be only 1 corner point, and in these cases $|z(f) \cup o(f)| = 1$. Therefore, $|z(f) \cup o(f)| \leqslant 2(n-1)$. □

It is still a question if the value $2(n-1)$ is reachable. A simple view of exercises in Fig. 4 shows that the real number of corner points will be smaller, but at this point we aim at mentioning the big difference between the recognition complexities of general MBF and the *KK-MBF*.

Concerning the issue about the size of deadlock resolving sets we may refer to the Theorem 1 of Daykin *et al.* (1974) and to Clements (1973). Here, parametrized subsets of MBF are considered. Let us suppose that there are given numbers $p_0, p_1, \dots, p_n$, denoting the quantities of upper zeros of functions on layers of B^n. Existence theorems of Sperner families, i.e. existence of MBF by the sets $\{p_0, p_1, \dots, p_n\}$ are obtained. The first theorem, Daykin *et al.* (1974), transfers all p_i to the layers upper the middle layer. The second theorem (Clements, 1973; Daykin *et al.*, 1974) gives the necessary and sufficient condition of existence in terms of a formula based on Kruskal's cascade properties (Kruskal, 1963).

Similar theorems could be formulated for *KK-MBF*, where p_i s are the numbers of corner points in layers, and then p_is can be either 0 or 1. Thus, the formulas would obtain simpler Kruskal's cascade forms. But in our case, the upper bound $2(n-1)$ is relatively small and acceptable as a complexity estimation in comparison to $C_n^{\lfloor n/2 \rfloor} + C_n^{\lfloor n/2 \rfloor + 1}$. The problem is how to effectively find the mentioned corner points algorithmically.

On the other hand, let us consider a series of *KK-MBF* functions, which provide a lower bound for $|z(f) \cup o(f)|$.

Define corner points on layers as follows:

On the 1st layer we take two corner points: 1-corner point is the vertex $(1\overbrace{00\dots0}^{n-1})$, the smallest vertex of $B^{n-1}_{x_1=1}$ in *lex* order; and 0-corner point is $(01\overbrace{00\dots0}^{n-2})$, the smallest vertex of $B^{n-2}_{x_1=0,x_2=1}$ in *lex* order. These are two neighbour vertices on the 1st layer.

On the 2nd layer we construct two corner points as follows: 1-corner point is $(01\overbrace{00\dots0}^{n-3}1)$, the smallest vertex of the cube $B^{n-3}_{x_1=0,x_2=1,x_n=1}$; 0-corner point is $(0011\overbrace{00\dots0}^{n-4})$, the smallest vertex of $B^{n-4}_{x_1=0,x_2=0,x_3=1,x_4=1}$, all in *lex* order.

This process is continued until one of the constructed subcubes becomes small and we reach a corner point defined by a 0-size subcube, i.e. a vertex.

Let us explain again the construction. Vertex $(1\overbrace{00\dots0}^{n-1})$ is the smallest vertex on layer 1 of B^n. Vertex $(01\overbrace{00\dots0}^{n-2})$ is its left neighbour (see Fig. 4). Vertex $(1\overbrace{00\dots0}^{n-1})$ defines all vertices of $B^{n-1}_{x_1=1}$ as units of the function by *Cond-h*, and vertex $(01\overbrace{00\dots0}^{n-2})$ defines as zero only the vertex $(\overbrace{00\dots0}^{n})$. Upper area of vertex $(01\overbrace{00\dots0}^{n-2})$ still can be defined arbitrarily – either as a zero or as a unit. But we take the leftmost vertex $(01\overbrace{00\dots0}^{n-3}1)$ of the layer next to the layer of $(01\overbrace{00\dots0}^{n-2})$ defining its value as unit. The left neighbour's value we define as zero. This simple construction, repeated as long as possible, gives a series of *KK-MBF* functions. How many corner points may have these functions?

Starting from the first layer, we reach the last possible corner point on the:

$(n/2)$-th layer, if n is even – in this case we have 2 corner points on each layer starting from 1 to $n/2$; and $(n+1)/2$-th layer, if n is odd – in this case we have 2 corner points on each layer starting from 1 to $(n-1)/2$, and one corner point on the $(n+1)/2$-th layer.

In both cases, we get $|z(f)\cup o(f)| = n$; and thus,

$$n \leqslant \max_{f\in KK\text{-}MBF} \bigl|z(f)\cup o(f)\bigr| \leqslant 2(n-1).$$

3.3. *Identification of KK-MBF Type Functions: Main Procedures*

Hansel chain based MBF recognition methods are global tools and can be applied to recognize any class of monotone Boolean functions, including *KK-MBF*. However, we aim to exploit specific properties of this class of functions to achieve more efficient recognition. To go beyond the chain-based analysis, in this section we consider the recognition of *KK-MBF* functions at the level of analysis of particular layers of B^n.

A useful step and exercise in recognitions is to determine the first and last nontrivial layers (trivial layers are those with all-zeros or with all units of the function). This can be done considering the following two chains of length n: L_1 that is the chain consisting of all first, and L_2 that is the chain consisting of all last elements of the lexicographic order of layers. L_1 is the longest Hansel chain in the chain split, see Fig. 3. Applying bisections on chains L_1 and L_2, we can find two neighbouring layers (k_1, k_1+1) and (k_2, k_2+1) and vertices on chains L_1 and L_2, respectively, where the function's value changes from 0 to 1. This means that the layers from k_2+1 to n, and from 0 to k_1 are trivial, they accept values 0 and 1, correspondingly. Indeed, when first vertex of layer k_1 accepts 0, then the whole layer accepts 0. The layer k_1 and below is filled by values 0 (the trivial 0 layers), but in layer k_1+1 there exists at least one vertex with value 1 so that trivial layers interrupt here. Similar explanation is valid concerning the chain L_2. The bisection procedure of each chain requires only $\log_2(n)$ queries to the oracle.

On the other hand, if the vertices on each layer of B^n are ordered lexicographically, we can apply layer by layer bisections and find a corner vertex candidate α_k on the k-th layer with no more than $\log_2 (C_n^k + 1)$ queries.

In this manner, a *KK-MBF* type function can be recognized by no more than $\sum_{k=1}^{n} \log_2 (C_n^k + 1)$ queries. A very rough estimate of this total value would be $O(n^2)$, as the largest layer is the middle k-th layer with $k = [n/2]$, where $C_n^{[n/2]} \sim \frac{2^{n+1}}{\sqrt{2\pi n}}$ with $n \to \infty$.

Proposition 4. *Time complexity of reconstruction of monotone Boolean functions of class KK-MBF by layer by layer bisections is* $O(n^2)$.

For *KK-MBF* functions the time complexity of Hansel algorithm remains $\sim C_n^{[n/2]}$ so $O(n^2)$ is a valuable reduction of the complexity characteristics. But the techniques of Tonoyan (1979) and Sokolov (1987) for the reduction of memory used in Hansel algorithm can not be applied in the case of our layer bisection approach.

To find a way to reduce the memory complexity we continue to address algorithmic questions on layers from an additional perspective. In order to apply bisections on layers, we need to keep all 2^n vertices (ordered lexicographically, layer-by-layer) in computer memory. Otherwise, we need to find a vertex by its number in lexicographic order, or by a distance from a certain vertex (again, in lexicographic order). There are no explicit formulas for this. Thus, both cases will require appropriate time and memory resource.

One way to avoid this situation is using sub-cube structures on layers when partitioning the cube. This is easily applicable to find the 0–1 border on layers, although it will require more than logarithmic queries on layers.

We will use α_j^k for the j-th element of L_k in the lexicographic order. Then the smallest element of L_k in the lexicographic order is $\alpha_1^k = (\overbrace{11\ldots1}^{k}\overbrace{00\ldots0}^{n-k})$, and the largest one is the element $\alpha_{C_n^k}^k = (\overbrace{00\ldots0}^{n-k}\overbrace{11\ldots1}^{k})$.

Let $B_{x_1=1}^{n-1}$ and $B_{x_1=0}^{n-1}$ be the partitions of B^n according to x_1, and $L_{k,x_1=1}$ and $L_{k,x_1=0}$ denote the parts of L_k in $B_{x_1=1}^{n-1}$ and $B_{x_1=0}^{n-1}$, respectively. Then $\alpha_1 = (0\overbrace{11\ldots1}^{k}\overbrace{00\ldots0}^{n-k-1})$ is the lexicographically smallest element of $L_{k,x_1=0}$, and $\alpha_0 = (1\overbrace{00\ldots0}^{n-k}\overbrace{11\ldots1}^{k-1})$ is the lexicographically largest element of $L_{k,x_1=1}$.

Instead of taking the arithmetical middle vertex of L_k to ask/test the function value, we take either α_1 or α_0 (for certainty, we will take α_1). If $f(\alpha_1) = 1$, then $f(\alpha) = 1$ for all $\alpha \in L_{k,x_1=1}$; therefore, the next vertex that we will take to ask the function value is the largest element of $L_{k,x_1=0,x_2=0}$ (the part of L_k in $B_{x_1=0,x_2=0}^{n-2}$), this is element $\alpha_2 = (00\overbrace{11\ldots1}^{k}\overbrace{00\ldots0}^{n-k-2})$.

If $f(\alpha_1) = 0$, then $f(\alpha) = 0$ for all $\alpha \in L_{k,x_1=0}$; therefore, the next vertex that we will take is the largest element of $L_{k,x_1=1,x_2=0}$ (the part of L_k in $B^{n-2}_{x_1=1,x_2=0}$), this is the element $\alpha_2 = (10\overbrace{11\ldots1}^{k-1}\overbrace{00\ldots0}^{n-k-1})$.

In general, in $B^{n-i}_{x_1=\sigma_1,\ldots,x_i=\sigma_i}$ the largest element of k-th layer is

$\sigma_1\ldots\sigma_i\overbrace{11\ldots1}^{x}\overbrace{00\ldots0}^{y}$, where x is k minus the number of 1s in $\sigma_1\ldots\sigma_i$, and y is $(n-k)$ minus the number of 0s in $\sigma_1\ldots\sigma_i$.

In this way, after each query, we continue in a smaller sub-cube, and hence, the number of queries in each layer can be at most $n-1$. For the heaviest layer, for $k = n/2$, we get the same estimate, but without either keeping all vertices in computer memory or calculating the given j-th vertex in the lexicographic order. Indeed, the number of vertices of middle layer equals $C_n^{[n/2]} \sim \frac{2^{n+1}}{\sqrt{2\pi n}}$ when $n \to \infty$. Logarithm of this, the number of steps in dichotomy, is equivalent to $n-1$, and so the partition by cube structures is simple and effective.

Proposition 5. *Reconstruction of monotone Boolean functions of class KK-MBF can be organized with complexity $O(n^2)$ without keeping the cube structure in computer memory.*

As an example, consider the function given in Fig. 4, and suppose that $k = 3$. Then,
$\alpha_1 = (01110)$, and since $f(01110) = 0$, the next vertex is
$\alpha_2 = (10110)$. $f(10110) = 1$, and it follows that the next is
$\alpha_3 = (10011)$. $f(10011) = 1$.

This way, we found the corner points $f(01110) = 0$ and $f(10011) = 1$ of the third layer.

So far, when recognizing *KK-MBF* functions layer by layer, we have only used the fact that the function satisfies the condition *Cond-h* and have not used the monotonicity of the function. Using the monotonicity will narrow the space for taking the next vertex for testing.

Suppose that α_k is the smallest vertex of L_k in the lexicographic order with $f(\alpha_k) = 1$. Denote by α_{k+1} the lexicographically smallest upper neighbour of α_k in L_{k+1}. Then, at the next step, we need to consider only those vertices of L_{k+1}, lexicographically smaller than α_{k+1}.

For the given example in Fig. 4, (10100) is the smallest vertex of L_2 in the reverse-lexicographic order, where the function value is 1. Then, we will choose the next vertex in the interval [(00111), (10011)] instead of [(00111), (11100)].

But to calculate the length of the interval [reverse-lexicographically first vertex of L_{k+1}, α_{k+1}] we need to find the number of α_{k+1} in lexicographic order. Effective solution of this problem is still open.

Here as well, one can use sub-cube structures, but the benefit is only in the case when $f(\alpha_{k+1}) = 0$.

We conclude the section with the following general note. It addresses an alternative partial order of vertices to the traditional monotony based order, where $\alpha \preccurlyeq \beta$ if α

coordinate-wise precedes β. In terms of *KK-MBF*, each of the vertices α and β creates 2 domains: upper domains $\hat{d}(f, \alpha)$ and $\hat{d}(f, \beta)$, when $f(\alpha) = f(\beta) = 1$, and lower domains $\check{d}(f, \alpha)$ and $\check{d}(f, \beta)$, when $f(\alpha) = f(\beta) = 0$. In this case we define the following order: $\alpha \preccurlyeq_d \beta$ if $d(f, \alpha) \subseteq d(f, \beta)$ with $f(\alpha) = f(\beta) = 0$, and, which is the same, $\alpha \preccurlyeq_d \beta$ if $d(f, \alpha) \supseteq d(f, \beta)$ with $f(\alpha) = f(\beta) = 1$. Otherwise, vertices α and β are incomparable. We call this construction *KK-poset*. Sperner type family may be obtained as a simple extension of this concept to the case of KK-poset. Here, similar to general MBFs, each *KK-MBF* is given by two complementary Sperner families – one is composed by 0-corner points, and the other includes all maximal 1-corner points of the function. We obtain that the *KK-MBF* reconstruction is similar to the MBF reconstruction, just the basic space and poset is a bit different. The technique developed for the general MBF and the experiences may be used in the problem of reconstruction of *KK-MBF* functions. We consider this reduction and transfer from MBF to *KK-MBF* an interesting research topic that is worthy of further investigations.

4. Cardinality of KK-MBF Class

Another important issue is the size of the whole class of *KK-MBF* functions (Korshunov, 2003; BFA, 2023; Jung, 2023). First, let us note that the function, given through $p_{n/2} = C_n^{n/2}$ (with all other layer characteristics equal to 0), belongs to the class *KK-MBF* and is the only function with the largest number of lower units. Therefore, to count the number of *KK-MBF* functions, we need to consider the number of non-negative integer partitions for an arbitrary positive integer p, $1 \leqslant p \leqslant C_n^{n/2}$: $p = p_1 + p_2 + \cdots + p_{n-1}$ (excluding the boundary cases $p = p_0 = 1$ and $p = p_n = 1$), such that $0 \leqslant p_1 \leqslant |[\alpha_1^s, \alpha_1^l]|$, $0 \leqslant p_2 \leqslant |[\alpha_2^s, \alpha_2^l]|, \dots, 0 \leqslant p_{n-1} \leqslant |[\alpha_{n-1}^s, \alpha_{n-1}^l]|$, where $[\alpha_j^s, \alpha_j^l]$ is the feasible interval of vertices on the j-th layer with α_j^s as the smallest and α_j^l as the largest element in the lexicographic order. These smallest and largest elements are defined in the following way. For all intervals, α_j^s is the lexicographically smallest element of the j-th layer. As for the largest elements, – α_1^l is the largest element of the first layer. To find α_2^l, we consider the smallest element of $\delta^{1+1}\mathcal{M}_1$, where $\mathcal{M}_1$ is the final m_1 element on the 1st layer in the lexicographic order, and α_2^l is the vertex previous to it. To find α_3^l, we consider the smallest element of $\delta^{2+1}\mathcal{M}_2$, where $\mathcal{M}_2$ is the next m_2 elements on the 2nd layer in the lexicographic order, after $\delta^{1+1}\mathcal{M}_1$, and α_3^l is the vertex previous to it.

In general, α_j^l is the previous to the smallest element of $\delta^j\mathcal{M}_{j-1}$ in the lexicographic order, where $\mathcal{M}_{j-1}$ is the next p_{j-1} element of the $(j-1)$-th layer after $\delta^{j-1}\mathcal{M}_{j-2}$.

Note that among $p_1, p_2, \dots, p_{n-1}$ at most $n-2$ elements can be positive. Moreover, all $n-2$ elements can be positive only in restricted cases, i.e. when $p_1 = p_2 = \cdots = p_{n-2} = 1$.

Thus, constraints on p_i are imposed not only in the form of $0 \leqslant p_i \leqslant |[\alpha_i^s, \alpha_i^l]|$, but also on their numbers depending on the previous values of $p_1, p_2, \dots, p_{i-1}$. This is one way of counting the number of elements of the class, but it is not easy to obtain an explicit formula in this manner.

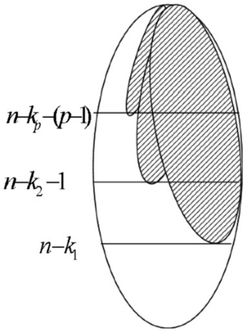


Fig. 5. Structure of $R_t(n)$.

Let C denote a particular subclass of *KK-MBF* functions having one 0-corner point at any layer of the cube.

To construct this subclass we take an arbitrary point $\alpha \in B^n$ and define the function by 0 value in the point α and in the domain $d(f, \alpha)$. f may have one or several 1-corner points. Since we may choose the point α in 2^n different ways, we get that $|C| \geqslant 2^n$.

5. Special Cases

In this section, we address another particular subclass of *KK-MBF* functions, which is related to the cascades of Kruskal.

Let $R_t(n)$ denote the initial t-length segment of the reverse lexicographic ordering on B^n. $R_t(n)$ corresponds to the set of units of a monotone Boolean function, denote it by f_R, the structure of which can be illustrated as in Fig. 5, where $k_1, k_2, \dots, k_p$ are parameters in binary representation of t, $t = 2^{k_1} + 2^{k_2} + \cdots + 2^{k_p}$; $k_1 > k_2 > \cdots > k_p > 0$.

Let $S = (s_1, s_2, \dots, s_n)$ be the associated vector of partitions of $R_t(n)$, composed as $s_i = |R_t(n)_{x_i=1}|$, $i = 1, \dots, n$.

It is known (Sahakyan, 2013, 2015) that f_R is the unique monotone Boolean function (up to variable permutations) with the smallest sum of coordinates of its associated vector of partitions among all monotone Boolean functions with t units. The coordinates of $S = (s_1, \dots, s_n)$ are also calculated.

Numbers of lower units of f_R on the layers $n - k_1, n - k_2 - 1, \dots, n - k_p - 1$ are determined by t.

Equivalently, f_R can be constructed in the following way:

Consider again the binary representation of t:

$$t = \alpha_n 2^n + \alpha_{n-1} 2^{n-1} + \cdots + \alpha_1 2^1 + \alpha_0 2^0.$$

Suppose that the first positive summand is 2^{k_1} ($\alpha_{k_1} = 1$). We construct an interval with the lower unit as the smallest vertex of the $n - k_1$-th layer by the reverse lexicographic order. If the next $\alpha_{k_1-1} > 0$, then we take the next vertex of the reverse lexicographic order as a lower unit, and compose an interval. If a current $\alpha_{k_1-j} = 0$, then we go up (we go up as many times as they are 0), and take the smallest vertex of the reverse lexicographic order (which is not in the upper shadows of the constructed intervals).

By constructing the same function in this manner, we know the numbers of lower units in layers, depending on t.

Thus, for a given t, we constructed a *KK-MBF* function with t units, and we also know its characteristics, $\#\min T(f_R) = \langle p_{i_1}, p_{i_2}, \dots, p_{i_r}\rangle$.

On the other hand, if given characteristics $\#\min T(f_R) = \langle p_{i_1}, p_{i_2}, \dots, p_{i_r}\rangle$ of some *KK-MBF* function, we can count the number of its units.

In general, it is possible to impose restrictions on the numbers/layers of the lower units of *KK-MBF* functions to get this special subclass.

6. Concluding Remarks

Boolean functions are not only a means of computing functional dependencies, but also represent a suitable mathematical apparatus for modelling data science systems. The limitations of models and the structure of their joint collections are reduced to considering Boolean functions that have the property of monotonicity. However, decoding monotone Boolean functions is a multifaceted problem, and there remain many unsolved or inefficiently solved problems in this context. Combinatorial constructions have been considered in some detail, but they are complex and often reduced to enumeration (brute-force). A possible new approach is to bring in a new resource, namely that of artificial intelligence (Valiant, 1984; Sahakyan *et al.*, 2022; Zhuravlev *et al.*, 2019, 2020). In this formulation, the emphasis is placed on solving a large number of problems of the class under consideration, accumulating the results of solutions in the form of a database, training on them, and not solving but recognizing the solution of the problem under consideration by analysing the parameters of the problem and the database information. The problem in this formulation is already becoming popular, and our first results related to it refer to decoding arbitrary monotone Boolean functions and are presented in Sahakyan *et al.* (2022).

Another possible approach continues the first one and seeks ways of refining, and reconstructing the problem constraints, with subtypes of monotone Boolean functions appearing, the decoding of which requires refined approaches, and the associated algorithms, whether combinatorial or based on machine learning, that can be practically implemented. There is a list of practical problems in big data analytics that reduce to the diverse classes of monotone Boolean functions. We begin the study of one of the classes of such functions – shadow minimized Boolean functions, for subsets of finite sets. We proceeded from the well-known solution of the problem for layers, formulated in the form of the Kruskal-Katona theorem, and on the extension of this fact to all layers of the cube, when the existence conditions for Sperner systems are obtained. We were able to show that the class of these functions has the structure of a generating set, which is not a necessary property of arbitrary classes of functions. Basic structures of data analysis of the problem of identification of these functions, details of memory organization in the optimal mode are also given, but we consider the beginning of these investigations as the main step, and we think that subsequent investigations will give acceptable complexity results for solving these problems, both in this and in other systems of functions with constraints.

Acknowledgements

The work is partially supported by grant No. 21T-1B314 of the Science Committee of MESCS RA.

References

Aslanyan, L.H. (1979). The discrete isoperimetry problem and related extremal problems for discrete spaces. *Problemy Kibernetiki*, 36, 85–128.

Aslanyan, L., Sahakyan, H. (2009). Chain split and computations in practical rule mining. In: *Intenational Book Series Information Science and Computing, Book 8, Classification, Forecasting, Data Mining*. Institute of Information Theories and Applications, FOI ITHEA, Bulgaria, pp. 132–135.

Aslanyan, L., Sahakyan, H. (2017). The splitting technique in monotone recognition. *Discrete Applied Mathematics*, 216, 502–512.

Aslanyan, L., Sahakyan, H., Romanov, V., Da Costa, G., Kacimi, R. (2019). Large network target coverage protocols. In: *2019 Computer Science and Information Technologies (CSIT)*, Yerevan, Armenia, 2019, pp. 57–64. https://doi.org/10.1109/CSITechnol.2019.8895058.

Aslanyan, L., Gishyan, K., Sahakyan, H. (2023a). Target class classification recursion preliminaries. *Baltic Journal of Modern Computing*, 11(3), 398–410.

Aslanyan, L., Katona, G., Sahakyan, H. (2023b). Notes on reconstruction of shadow minimization Boolean functions. In: *2023 Computer Science and Information Technologies (CSIT)*, Yerevan, Armenia, 2023, September 25–30, pp. 119–123.

Balogh, J., Katona, G.O., Linz, W., Tuza, Z. (2021). The domination number of the graph defined by two levels of the n-cube, II. *European Journal of Combinatorics*, 91, 103201.

Bezrukov, S., Kuzmanovski, N., Lim, J. (2023). Pull-push method: a new approach to edge-isoperimetric problems. *Discrete Mathematics*, 346(12), 113632.

BFA (2023). Ninth Dedekind number discovered: scientists solve long-known problem in mathematics. In: *The 8th International Workshop on Boolean Functions and their Applications (BFA)*, in memory of Kai-Uwe Schmidt, September 3–8, 2023, Voss, Norway.

Black, H., Chakrabarty, D., Seshadhri, C. (2023). Directed isoperimetric theorems for Boolean functions on the hypergrid and an $O(n\sqrt{d})$ monotonicity tester. In: *STOC 2023: Proceedings of the 55th Annual ACM Symposium on Theory of Computing*, pp. 233–241.

Blum, A. (2003). Learning a function of r relevant variables. In: *Learning Theory and Kernel Machines*, 16th Annual Conference on Learning Theory and 7th Kernel Workshop, COLT, Kernel 2003, Washington, DC, USA, August 24–27, 2003. Springer, Berlin, Heidelberg, pp. 731–733.

Boros, E., Hammer, P. (2002). Pseudo-Boolean optimization. *Discrete Applied Mathematics*, 123(1–3), 155–225.

Boros, E., Hammer, P., Ibaraki, T., Kawakami, K. (1991). Identifying 2-monotonic positive Boolean functions in polynomial time. In: *ISA'91 Algorithms: 2nd International Symposium on Algorithms Taipei*, Republic of China, December 16–18, 1991. Springer, Berlin, Heidelberg, pp. 104–115.

Braverman, M., Khot, S., Kindler, G., Minzer, D. (2022). *Improved monotonicity testers via hypercube embeddings*. Preprint: arXiv:2211.09229v1.

Carlet, C., Joyner, D., Stănică, P., Tang, D. (2016). Cryptographic properties of monotone Boolean functions. *Journal of Mathematical Cryptology*, 10(1), 1–14.

Chao, T.-W., Yu, H.-H.H. (2023). *Kruskal–Katona-type problems via entropy method*. Preprint: arXiv:2307.15379.

Clements, G.F. (1973). A minimization problem concerning subsets of a finite set. *Discrete Mathematics*, 4(2), 123–128.

Crawford-Kahrl, P., Cummins, B., Gedeon, T. (2022). Joint realizability of monotone Boolean functions. *Theoretical Computer Science*, 922, 447–474.

Damásdi, G., Gerbner, D., Katona, G.O., Keszegh, B., Lenger, D., Methuku, A., Nagy, D.T., Pálvölgyi, D., Patkós, B., Vizer, M., Wiener, G., (2021). Adaptive majority problems for restricted query graphs and for weighted sets. *Discrete Applied Mathematics*, 288, 235–245.

Daykin, D.E., Godfrey, J., Hilton, A.J.W. (1974). Existence theorems for Sperner families. *Journal of Combinatorial Theory, Series A*, 17(2), 245–251.

Dedekind, R. (1895). Uber die Begründung der Idealtheorie. *Gött. Nachr*, 106–113.

Frankl, P., Katona, G.O. (2021). On strengthenings of the intersecting shadow theorem. *Journal of Combinatorial Theory, Series A*, 184, 105510.

Freixas, J. (2022). On the enumeration of some inequivalent monotone Boolean functions. *Optimization*, https://doi.org/10.1080/02331934.2022.2154126.

Gainanov, D.N. (1984). On one criterion of the optimality of an algorithm for evaluating monotonic Boolean functions. *Zhurnal Vychislitel'noi Matematiki i Matematicheskoi Fiziki*, 24(8), 1250–1257.

Griggs, J.R., Kalinowski, T., Leck, U., Roberts, I.T., Schmitz, M. (2023). The saturation spectrum for antichains of subsets. *Order*, 40, 537–574. https://doi.org/10.1007/s11083-022-09622-6.

Hansel, G. (1966). Sur le nombre des fonctions Booléennes monotones de n variables. *Comptes Rendus de l'Académie des Sciences*, 262(20), 1088–1090.

Jackson, J.C., Lee, H.K., Servedio, R.A., Wan, A. (2011). Learning random monotone DNF. *Discrete Applied Mathematics*, 159(5), 259–271.

Jung, A. (2023). Shadow ratio of hypergraphs with bounded degree. *Graphs and Combinatorics*, 39(3), 40.

Kabulov, A., Berdimurodov, M. (2021). Parametric algorithm for searching the minimum lower unity of monotone Boolean functions in the process synthesis of control automates. In: *2021 International Conference on Information Science and Communications Technologies (ICISCT)*, Tashkent, Uzbekistan, 2021, pp. 1–4. https://doi.org/10.1109/ICISCT52966.2021.9670370.

Katona, G. (1968). A theorem of finite sets. In: *Theory of Graphs (Proceedings of the Colloquim held at Tihany)*. Academic Press, New York.

Katona, G. (1987). A theorem of finite sets. In: *Classic Papers in Combinatorics*, pp. 381–401.

Korobkov, V.K. (1965). On monotone functions of the algebra of logic. *Problemy Kibernetiki*, 13, 5–28.

Korshunov, A.D. (1981). On the number of monotone Boolean menge. *Problemy Kibernetiki*, 38, 5–109.

Korshunov, A.D. (2003). Monotone Boolean functions. *Russian Mathematical Surveys*, 58(5), 929.

Kovalerchuk, B., Delizy, F. (2005). Visual data mining using monotone Boolean functions. In: *Visual and Spatial Analysis: Advances in Data Mining, Reasoning, and Problem Solving*. Springer, Dordrecht, pp. 387–406.

Kruskal, J.B. (1963). The number of simplices in a complex. *Mathematical Optimization Techniques*, 10, 251–278.

Kulhandjian, M., Aslanyan, L., Sahakyan, H., Kulhandjian, H., D'Amours, C. (2019). Multidisciplinary discussion on 5G from the viewpoint of algebraic combinatorics. In: *2019 Computer Science and Information Technologies (CSIT)*, Yerevan, Armenia, 2019, pp. 69–76, https://doi.org/10.1109/CSITechnol.2019.8895166.

Lange, J., Vasilyan, A. (2023). *Agnostic proper learning of monotone functions: beyond the black-box correction barrier*. Preprint: arXiv:2304.02700.

Lovász, L. (2007). *Combinatorial Problems and Exercises*, 3nd edition. American Mathematical Society.

Madden, G.N. (2023). *Shadows of Colored Complexes and Cycle Decompositions of Equipartite Hypergraphs*. PhD thesis, Illinois State University.

Nosov, M.V. (2023). Estimates of the degrees of separating polynomials for monotone and self-dual functions. *Intelligent Systems. Theory and Applications*, 27(2), 79–82.

O'Donnell, R., Servedio, R. (2005). Learning monotone functions from random examples in polynomial time. *CMU School of Computer Science*. Citeseer.

Sahakyan, H. (2013). (0, 1)-matrices with different rows. In: *Ninth International Conference on Computer Science and Information Technologies Revised Selected Papers*, Yerevan, Armenia, 2013, pp. 1–7. https://doi.org/10.1109/CSITechnol.2013.6710342.

Sahakyan, H. (2015). On the set of simple hypergraph degree sequences. *Applied Mathematical Sciences*, 9(5), 243–253.

Sahakyan, H. (2023). On nonconvexity of the set of hypergraphic sequences. In: *2023 Computer Science and Information Technologies (CSIT)*. NAS RA, pp. 47–52.

Sahakyan, H., Aslanyan, L. (2017). Discrete tomography with distinct rows: relaxation. In: *2017 Computer Science and Information Technologies (CSIT)*. Yerevan, Armenia, 2017, pp. 117–120. https://doi.org/10.1109/CSITechnol.2017.8312153.

Sahakyan, H., Aslanyan, L., Katona, G. (2022). Notes on identification of monotone Boolean functions with machine learning methods. In: *Proceedings of "Middle-European Conference on Applied Theoretical Computer Science" Conference*. Koper, Slovenia, 13–14 October 2022.

Sales, M., Schülke, B. (2022). *A local version of Katona's intersection theorem*. Preprint: arXiv:2206.04278.

Schröder, B. (2016). *Ordered Sets: An Introduction with Connections from Combinatorics to Topology*. Birkhäuser.

Sokolov, N.A. (1987). Optimal reconstruction of monotone Boolean functions. *Computational Mathematics and Mathematical Physics*, 27(6), 181–187.

Tennakoon, R., Suter, D., Zhang, E., Chin, T.-J., Bab-Hadiashar, A. (2021). Consensus maximisation using influences of monotone Boolean functions. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2866–2875.

Tonoyan, G.P. (1979). Chain partitioning of n-cube vertices and deciphering of monotone Boolean functions. *Journal of Computational Mathematics and Mathematical Physics*, 19(6), 1532–1542.

Valiant, L.G. (1984). A theory of the learnable. *Communications of the ACM*, 27(11), 1134–1142.

Zhang, E., Suter, D., Tennakoon, R., Chin, T.-J., Bab-Hadiashar, A., Truong, G., Gilani, S.Z. (2022). Maximum consensus by weighted influences of monotone Boolean functions. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, New Orleans, LA, USA, 2022, pp. 8964–8972. https://doi.org/10.1109/CVPR52688.2022.00876.

Zhuravlev, Y.I., Ryazanov, V.V., Aslanyan, L.H., Sahakyan, H.A. (2019). On a classification method for a large number of classes. *Pattern Recognition and Image Analysis*, 29, 366–376.

Zhuravlev, Y.I., Ryazanov, V.V., Ryazanov, V.V., Aslanyan, L.H., Sahakyan, H.A. (2020). Comparison of different dichotomous classification algorithms. *Pattern Recognition and Image Analysis*, 30, 303–314.

L. Aslanyan graduated from Novosibirsk State University in 1968. He received his PhD in 1976, and his doctorate in 1997. He is a professor since 1997 and a corresponding member of National Academy of Sciences of the Republic of Armenia since 2014. He currently heads the Department of Discrete Modelling, Analysis, and Recognition at the Institute for Informatics and Automation Problems of the NAS RA. Research interests: mathematical logic, discrete mathematics, mathematical theory of pattern recognition and artificial intelligence.

G. Katona, prof., is a doctor of the mathematical sciences. He is an academician of the Hungarian Academy of Sciences, a member of the European Academy of Sciences, a foreign member of the Bulgarian Academy of Sciences, a research professor at Alfred Renyi Institute of Mathematics, Hungarian Academy of Sciences, and a professor at the Eötvös L. University. His main research areas include extreme combinatorial aanalysis.

H. Sahakyan graduated from Yerevan State University. She received her PhD in 2002 and her doctorate in 2018. She is currently a scientific secretary of the Institute for Informatics and Automation Problems of the NAS RA and a leading researcher at the Discrete Mathematics Department. Scientific interests: combinatorics, discrete optimization, discrete tomography, artificial intelligence, and data mining.