Notes on Reconstruction of Shadow Minimization Boolean Functions

Levon Aslanyan
Institute for Informatics and
Automation Problems of NAS RA
Yerevan, Armenia
e-mail: lasl@sci.am

Gyula Katona
Alfréd Rényi Institute of Mathematics
Budapest, Hungary
e-mail: katona.gyula.oh@renyi.hu

Hasmik Sahakyan
Institute for Informatics and
Automation Problems of NAS RA
Yerevan, Armenia
e-mail: hsahakyan@sci.am

***Abstract*—This paper investigates the KK-MBF class of monotone Boolean functions coming from the finite set systems shadow minimization theory. Zeros of these functions correspond to the initial segments of the lexicographic order on the layers of binary cube. In particular, we focus on query-based recognition algorithms of KK-MBF functions, and consider the cardinality estimations of this class.**

***Keywords*—Monotone Boolean functions, reconstruction, shadow, KK-MBF class.**

I. Introduction

Many problems with monotone Boolean functions (MBFs) appear in logical and physical level design of systems, but also in artificial intelligence models, computation learning theory, hypergraph theory, and other areas. MBFs are used to encode extremely important constructions in various combinatorial optimizations: they provide a natural way to describe compatible subsets of sets of finite constraints. A number of applications (e.g., wireless sensor networks, dead-end tests of tables, data mining [1,2]) use optimization with MBF, where MBFs are represented by constructions such as chains and anti-chains [3] in hypercubes. Other similar applications with MBF can be added to this list [4,5,6].

There are a number of known effective tools and methods for analyzing MBFs, and new approaches are constantly being sought, investigated, and applied. Well-known open problems in this area includes 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 [7,1].

A well-known problem concerning MBFs is the query-based identification problem – the recognition of an unknown MBF of n variables by using membership queries. Hansel's algorithm [7], based on partitioning the binary cube into a special set of non-intersecting chains, provides optimal reconstruction in the sense of Shannon complexity. In practical algorithmic implementations, it is even not necessary to build and store all chains in computer memory [8,9,10].

In order to obtain solutions for bounded classes of MBFs, it is necessary to find a way to the structural properties of those classes. As we already mentioned, our research objective here is the well-known Kruskal-Katona theorem [14,15] and KK-MBF functions, that describe the exact optimal monotone constructions of shadow minimisation (constraint minimisation) and existence of Sperner systems for the given set of parameters. In this way, KK-MBF class of MBFs becomes a special and attractive class for recognition.

In this research, we investigate the KK-MBF class, focusing on query-based recognition algorithms, and addressing the cardinality issue of this class of MBFs.

II. Preliminaries

A. Monotone Boolean function recognition

Let $B^n = \{(x_1, \cdots, x_n) \mid x_i \in \{0,1\}, i = 1, \cdots, n\}$ denote the set of vertices of the n-dimensional binary (unit) cube. Let $\alpha = (\alpha_1, \cdots, \alpha_n)$ and $\beta = (\beta_1, \cdots, \beta_n)$ be two vertices of B^n. α *precedes* β (by *component-wise order*), denoted as $\alpha \preccurlyeq \beta$ if and only if $\alpha_i \leq \beta_i$ for $1 \leq i \leq n$. 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.

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

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:

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

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

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

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

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

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

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

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

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 $f(\alpha) \leq f(\beta)$. 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 [16]. The first algorithmic and complexity-related considerations belong to [17], 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 [18]. The technique on how to introduce and analyze MBFs, is basically presented in [7,14,15,1,10,8,9,19].

The Hansel chain structure [7] 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 [10], who invented 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. In continuation, [8] presented a slightly modified and simplified version of [10] by using two tools: enumeration of all chains, and a procedure of finding the i-th vertex of the j-th chain. Then, an optimized procedure is used to propagate newly found values to the chains by a divide-and-conquer manner.

In the MBF recognition problem using membership queries, the goal is to determine an unknown MBF of n variables using as few queries as possible. The function can be fully recognized by finding all its upper zeros (and/or lower units) [17]. The Shannon complexity of finding all upper zeros (lower units) of an arbitrary monotone Boolean function of n variables is $C_n^{\lfloor n/2 \rfloor} + C_n^{\lfloor n/2 \rfloor + 1}$ [7].

Another recognition structure is used in [9]. For even n, B^n is split according to two variables and the recognition in every sub-cube starts from its two middle layers. 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 in practical algorithms for arbitrary functions. Finally, it is to mention the work [19] that considers not the Shannon complexity but the individual complexity of MBF given by its resolving set size.

In general, tasks related to the recognition of MBFs may have different formulations. One task is to recognize a particular unknown function, knowing that it belongs to the class of MBFs or to one of its subclasses. Another task is to start with partial knowledge about the unknown function. 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 [37]. 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 [20,14,15]. KK-MBFs 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 co-lexicographic order.

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.

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

The combinatorial complexity of reconstruction in these and other subclasses of MBF is not well studied. For example, monotone Boolean functions, with zeros and ones 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 KK-MBF class.

III. KK-MBF Recognition

A. Identification of KK-MBF type functions

Definition 1: Let f be a monotone Boolean function over 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 lexicographic order (left parts of layers, uncolored circles in Fig.1).

Initial formulations of shadow theorem in [20,14] are given in terms of co-lexicographic order but this structure was later simplified to the lexicography in [32-34,25], 1977 and [26], 1979. The name KK also refers to an extension of the basic result of shadow theorem to many layers and to the existence of Sperner families. Usually, a KK-MBF function f is given through its characteristics, $\#\min T(f) = \langle p_{i_1}, p_{i_2}, \cdots, p_{i_r} \rangle$, where p_{i_j} is the number of lower units of f on the i_j-th layer. An example is given in Fig.1.

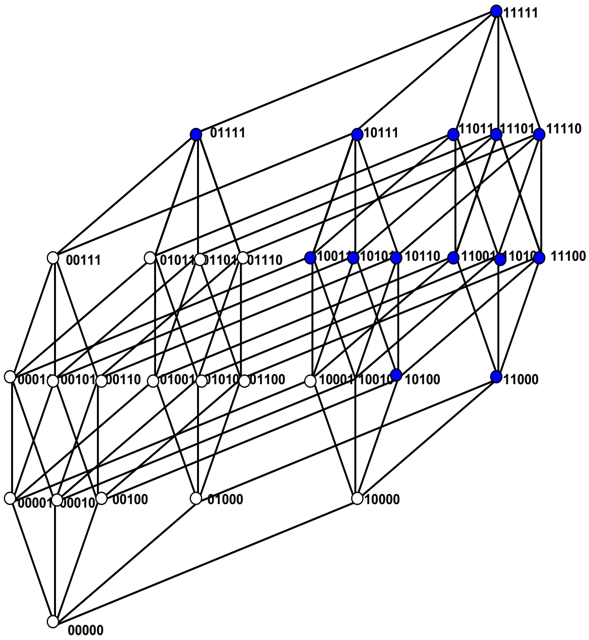

Fig.1 KK-MBF function with $p_2 = 2, p_3 = 1, p_4 = 1$

B. Resolving Sets

Let us introduce some general terms, from Boolean function deciphering (reconstruction) [17]. 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 $g = f$.

To reconstruct a function, it is sufficient to determine its values on some of its resolving sets. 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 $(f, \mathcal{S})$. Each MBF, in class $\mathcal{S}$ =MBF, has a unique deadlock resolving set that is included in its all resolving sets. This deadlock resolving set is $G(f) = \min T(f) \cup \max F(f)$ [17]. Mention that this is not the case for other functions and classes, for instance, for the class of symmetric Boolean functions, there are no unique deadlock resolving sets.

In continuation of our considerations with KK-MBF, it is convenient to take intersections of sets $\min T(f)$ and $\max F(f)$ with layers L_k of B^n, denoting them by $O_k(f)$ and $Z_k(f)$, correspondingly, and their sizes, - by $p_k(f)$ and $q_k(f)$.

We formulate two obvious properties for a KK-MBF type function f, and call them horizontal and vertical conditions.

Cond-h:

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

(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 succeeding α, $(\beta \succcurlyeq_{lex} \alpha)$.

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 $d(f, \alpha)$.

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

(1) $f(\beta) = 1$ for all β from the same layer, such that $\beta \succ_{lex} \alpha$, 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 $\beta \prec_{lex} \alpha$, and

(2) $f(\beta) = 0$ for all β, $\beta \preccurlyeq \alpha$ (component-wise order).

Let $z_k(f)$ denote the set of all zero corner points, and $o_k(f)$ denote the set of all one corner points of f. $f \in$KK-MBF is a monotone function and besides the $z_k(f)$ and $o_k(f)$ we will use for it notations $Z_k(f)$ and $O_k(f)$, and $p_k(f)$ and $q_k(f)$, for the corresponding sets and their sizes as well.

Proposition 1. Each monotone Boolean function f of class KK-MBF has a unique deadlock resolving set that is included in its all resolving sets. This deadlock resolving set for f is the set $g(f) = z(f) \cup o(f)$.

C. Identification Procedures

For a general MBF, it is well-known that $|p_k(f) + q_k(f)|$ 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 in a lesser complexity. Our first notion about the KK-MBF is that for them $|z(f) \cup o(f)|$ cannot become larger than $2n$, that limits the complexity of recognition of such functions.

Concerning the issue about the size of deadlock resolving set we may refer to the Theorem 1 of [26] and to the [36] (this size is not larger than n). But in our case, $2n$ is relatively small and is acceptable as a complexity estimation. The problem is how to effectively find the mentioned corner points.

A useful step and exercise in recognition is determination of the first and last nontrivial layers (trivial layersare all-zero and all-one value layers). This can be done by bisections of two chains - chains of all first and last elements of the lexicographic order of layers, from 0 to n. For instance, in left chain, we seek for two neighbour layers k and $k + 1$ with values 0 and 1 on the chain. $k + 1$ is the lowest layer with all 1 values. The bisection procedure requires $logn$ queries to the oracle. After this we will know the maximum layer with all 0 and minimum layer with all 1 values of function.

If the vertices on layers of B^n are ordered lexicographically, we can find any corner candidate point $\boldsymbol{\alpha_k}$ on the layer with no more than $\log_2(C_n^k + 1)$ queries.

Thus, knowing that f is a KK-MBF type function, we can recognize it by $\sum_{k=1}^{n}\log_2(C_n^k+1)$ queries. A very rough estimate of this would be $O(n^2)$.

In general description of structural properties of KK-MBF functions, it is worth mentioning, that the notion of vertex domain generates a partial order over the vertices of B^n. Corners are analogues of the maximal intervals of Boolean functions in B^n. Sets of all corner point (for zeros and for ones) are analogues of Sperner families in B^n. These sets and relations are an interesting research topic today.

D. Memory minimization

Consider some algorithmic issues. Not to keep all 2^n vertices in computer memory, we can use sub-cube structures of the layers.

We will use α_j^k for the j-th element of L_k in the lexicographic order. Then the largest element of L_k in the lexicographic order is $\alpha_{C_n^k}^k=(\overbrace{11\cdots1}^{k}\overbrace{00\cdots0}^{n-k})$, and the smallest is $\alpha_1^k=(\overbrace{00\cdots0}^{n-k}\overbrace{11\cdots1}^{k})$.

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 $\alpha_1=(0\overbrace{11\cdots1}^{k}\overbrace{00\cdots0}^{n-k-1})$ is the lexicographically largest element of $L_{k,x_1=0}$, and $\alpha_0=(1\overbrace{00\cdots0}^{n-k}\overbrace{11\cdots1}^{k-1})$ is the lexicographically smallest element of $L_{k,x_1=1}$.

Instead of taking the middle vertex of L_k to ask 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 $\alpha_2=(00\overbrace{11\cdots1}^{k}\overbrace{00\cdots0}^{n-k-2})$.

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 $B_{x_1=1,x_2=0}^{n-2}$), this is $\alpha_2=(10\overbrace{11\cdots1}^{k-1}\overbrace{00\cdots0}^{n-k-1})$.

In general, in $B_{x_1=\sigma_1,\cdots x_i=\sigma_i}^{n-i}$, the largest element of k-th layer is $\sigma_1\cdots\sigma_i\overbrace{11\cdots1}^{x}\overbrace{00\cdots0}^{y}$, where x is k minus the number of 1s in $\sigma_1\cdots\sigma_i$, and y is $(n-k)$ minus the number of 0s in $\sigma_1\cdots\sigma_i$.

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 $n-1$. 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.

As an example, consider the function given in Fig.1, 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
$\alpha_3=(10011)$. $f(10011)=1$.
In this way, we found the corner points $f(01110)=0$ and $f(10011)=1$ of the third layer.

IV. Cardinality of KK-MBF class

One more important issue is the size of the whole class of KK functions. 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, $1\le p\le C_n^{n/2}$: $p=p_1+p_2+\cdots+p_{n-1}$ (excluding the boundary cases $p=p_0=1$ and $p=p_n=1$), such that $0\le p_1\le|[\alpha_1^s,\alpha_1^l]|$, $0\le p_2\le|[\alpha_2^s,\alpha_2^l]|$, $\cdots$, $0\le p_{n-1}\le|[\alpha_{n-1}^s,\alpha_{n-1}^l]|$, where $[\alpha_j^s,\alpha_j^l]$ is the feasible interval of vertices on the j-th layer with α_j^s the smallest and α_j^l 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 elements on the 1-st layer in the lexicographic order, and α_2^l is the previous to it vertex. 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 2-nd layer in the lexicographic order, after $\delta^{1+1}\mathcal{M}_1$, and α_3^l is the previous to it vertex.

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} elements of the $(j-1)$-th layer after $\delta^{j-1}\mathcal{M}_{j-2}$.

In accordance with this, taking only those functions that have only 1 corner point in some layer of the cube, we obtain 2^n KK-MBF functions.

V. Concluding remarks

Boolean functions are not only a means of computing functional dependencies, but also represent a suitable mathematical apparatus for modeling 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 environment. Combinatorial constructions have been considered in some detail, but they are complex and often reduced to enumeration (brute-force). One possible new approach is to bring in a new resource, namely that of artificial intelligence [37]. 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 analyzing 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 [37].

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.

ACKNOWLEDGMENT

The work is partially supported by grant No21T-1B314 of the Science Committee of MESCS RA.

REFERENCES

[1] L. Aslanyan and H. Sahakyan, “The splitting technique in monotone recognition”, *Discrete Applied Mathematics*, 216, pp. 502–512, 2017.

[2] L. Aslanyan, H. Sahakyan, V. Romanov, G. Da Costa, and R. Kacimi, “Large network target coverage protocols”, *12th International Conference on Computer Science and Information Technologies, CSIT 2019*, pp. 57–64, 2019.

[3] Bernd SW Schröder, Ordered sets, Springer 29 (2003).

[4] A. Blum, “Learning a function of r relevant variables (open problem)”, *in Proceedings of the 16th Annual Conference on Learning Theory and 7th Kernel Workshop*, pp. 731-733, 2003.

[5] Nader H. Bshouty, and Christino Tamon, “On the Fourier spectrum of monotone functions”, *Journal of the ACM (JACM)*, vol. 43, no. 4, pp. 747-770, 1996.

[6] Jeffrey C. Jackson, Homin K. Lee, Rocco A. Servedio, and Andrew Wan, “Learning random monotone DNF, In Approximation, Randomization and Combinatorial Optimization. Algorithms and Techniques: *11th International Workshop, APPROX 2008, and 12th International Workshop, RANDOM 2008, Boston, MA, USA, August 25-27, 2008.* Proceedings, pp. 483-497. Springer Berlin Heidelberg, 2008.

[7] G. Hansel, “Sur le nombre des fonctions booléennes monotones de n variables,” C. R. Acad. Sci., 262:1088–1090, 1966.

[8] N. A. Sokolov, “On optimal deciphering of monotone functions of logic algebra”, *Journal of Computational Mathematics and Math. Physics*, 22(12):1878–1887, 1982.

[9] N. A. Sokolov, “Optimal reconstruction of monotone Boolean functions”, *USSR Computational Mathematics and Mathematical Physics*, 27(6):181–187, 1987.

[10] G. P. Tonoyan, “Chain partitioning of n-cube vertices and deciphering of monotone Boolean functions”, *Computational Mathematics and Mathematical Physics,* 19(6) (1979).

[11] Mohammad Azad, Igor Chikalov, Shahid Hussain, Mikhail Moshkov, and Beata Zielosko, “Decision Trees with Hypotheses for Recognition of Monotone Boolean Functions and for Sorting”, *arXiv 2203.08894v1, Book series “Synthesis Lectures on Intelligent Technologies”.*

[12] Ryan O’Donnell, and R. Servedio, “Learning monotone functions from random examples in polynomial time”, Citeseer, 2005.

[13] L. Valiant, “A theory of the learnable”, *Communications of the ACM*, 27(11):1134{1142. (1984)

[14] Gyula Katona, “A theorem of finite sets. Theory of Graphs”, *Academic Press - Akadémiai Kiadó, New York; Budapest,* (1968), pp. 187-207.

[15] Gyula Katona, “A theorem of finite sets”, *Classic Papers in Combinatorics*, (2009) pp. 381–401, Springer.

[16] R. Dedekind, „Über Zerlegungen von Zahlen durch ihre größten gemeinsamen Teiler,“ *Festschrift Hoch. Braunschweig u. ges. Werke, II* (1897), 103-148.

[17] V. K. Korobkov, “On monotone functions of the algebra of logic”, *Problemy kibernetiki*, 13:5–28 (1965).

[18] D. Korshunov, “On the number of monotone Boolean functions,” *Problemy kibernetiki*, 38:5–109, 1981.

[19] D. N. Gainanov, “On one criterion of the optimality of an algorithm for evaluating monotonic Boolean functions”, C*omputational Mathematics and Mathematical Physics*, 24 (1984), pp. 176–181.

[20] J. Kruskal, “The number of simplices in a complex”, *Mathematical Optimization Techniques, University of California Press*, 251–278, 1963.

[21] S. V. Yablonsky, “Functional constructions in k-valued logic”, US Joint Publications Research Service, 1960.

[22] Yu. I. Zhuravlev, “On separability of subsets of vertices of the n-dimensional unite cube”, *Collection of works on mathematical logic and its applications to some problems of cybernetics, MIAN SSSR,* M., 1958, pp. 143-157.

[23] L. H. Harper, “Optimal Assignments of Numbers to Vertices”, *J. Soc. Appl. Math.,* vol. 12, no. 1 (March 1964)

[24] L. H. Harper, “Optimal numberings and isoperimetric problems on graphs”, *J. Comb. Th. 1* (1966), pp. 385–393.

[25] L. H. Aslanyan “The discrete isoperimetry problem and related extremal problems for discrete spaces”, *Problemy kibernetiki,* 36:85–128 (1979).

[26] David E. Daykin, Jean Godfrey, and Anthony J. W. Hilton, "Existence theorems for Sperner families", *Journal of Combinatorial Theory, Series A*, vol. 17, no. 2 (1974): 245-251.

[27] E. Boros, P. L. Hammer, T. Ibaraki and K. Kawakami, “Identifying 2-monotonic positive Boolean functions in polynomial time”, *ISA’91, Algorithms, edited by W. L. Hsu and R. C. T. Lee, Springer Lecture Notes in Computer Science*, 557 (1991) 104-115.

[28] Peter Frankl and Gyula OH Katona, “On strengthenings of the intersecting shadow theorem”, *Journal of Combinatorial Theory, Series A*, 184:105510, 2021.

[29] S. Muroga, “Threshold Logic and Its Applications”, *Wiley-Interscience*, 1971.

[30] Ryan O’Donnell, and R. Servedio, “Learning monotone functions from random examples in polynomial time”, Citeseer, 2005.

[31] L. Aslanyan, H. Sahakyan, Chain Split and Computations in Practical Rule Mining, Intenational Book Series, Information Science and Computing, Book 8, Classification, forecasting, Data Mining, pp. 132-135, 2009.

[32] L. Aslanyan, V. Karakhanyan, “Description of all solutions of discrete isoperimetry problem”, *Doklady Akademii Nauk Arm SSR*, 65(4), Yerevan,1977, pp. 211-215.

[33] L. Aslanyan, V. Karakhanyan, B. Torosyan, “Solution of discrete isoperimetry problem”, *Doklady Akademii Nauk Arm SSR*, 65(5), Yerevan,1977, pp. 257-262.

[34] L. Aslanyan, V. Karakhanyan, B. Torosyan, “On compactness of subsets of vertices of n-dimensional unite cube”, Doklady Akademii Nauk SSSR, 241(1), Moscow, 1978 (English translation in Soviet Math. Dokl., American Mathematical Society, vol. 19(1978), no. 4, pp. 781-785), pp. 11-14.

[35] László Lovász, “Combinatorial problems and exercises”, *American Mathematical Soc.*, vol. 361, 2007.

[36] George F. Clements, “A minimization problem concerning subsets of a finite set”, *Discrete Mathematics*, vol. 4, no. 2 (1973): pp. 123-128.

[37] H. Sahakyan, L. Aslanyan, G. Katona, “Notes on Identification of Monotone Boolean Functions with Machine Learning methods”, proceedings of “Middle-European Conference on Applied Theoretical Computer Science” conference, 2022, Slovenia.