Discrete Applied Mathematics 216 (2017) 502–512

Contents lists available at ScienceDirect

Discrete Applied Mathematics

journal homepage: www.elsevier.com/locate/dam

The splitting technique in monotone recognition

L. Aslanyan *, H. Sahakyan

Institute for Informatics and Automation Problems, National Academy of Sciences, P. Sevak str. 1, 0014 Yerevan, Armenia

A R T I C L E I N F O

Article history:
Received 7 March 2015
Received in revised form 12 November 2015
Accepted 16 April 2016
Available online 7 May 2016

Dedicated to the bright memory of Levon Khachatrian

Keywords:
Monotone Boolean function
Grid
Recognition

A B S T R A C T

We consider the problem of query based algorithmic identification/recognition of monotone Boolean functions, as well as of binary functions defined on multi-valued discrete grids. Hansel's chain-split technique of n-cubes is a well known effective tool of monotone Boolean recognition. An extension by Alekseev is already applied to the grid case. The practical monotone recognition on n-cubes is provided by the so called chain-computation algorithms that is not extended to the case of multi-valued grids. We propose a novel split construction based on partitioning the grid into sub-grids and into discrete structures that are isomorphic to binary cubes. Monotonicity in a multi-valued grid implies monotonicity in all induced binary cubes and in multi-valued sub-grids. Applying Hansel's technique for identification of monotone Boolean functions on all appearing binary cubes, and Alekseev's algorithm on all sub-grids leads to different scenarios of reconstruction of monotone functions. On one hand such partitioning technique makes parallel recognition possible, on the other hand — the method can be used in practical identification algorithms due to simple structures and easily calculable quantities appearing after the partition to the n-cubes. Complexity issues of considered algorithms were also elaborated.

1. Introduction and preliminaries

Monotone Boolean functions appear in various ICT applications, such as design of electronic schemes, pattern recognition, discrete optimization, cryptography and others [2,6,21]. Monotone Boolean functions are studied from different viewpoints and they are known as a type of high complexity objects [13,11,14,15,18–20]. Often researchers link this complexity to the Sperner families of partially ordered sets of elements [8]. As a rule, the problem is considered in specific posets — such as the binary cube, and the multi-valued multidimensional grid. A number of results in the domain of structural optimization of monotone Boolean functions and their recognition are obtained by G. Hansel, V. Korobkov, A. Korshunov, G. Tonoyan, N. Zolotykh, V. Alekseev, A. Serjantov and others [11,1,14–16,23,27,24–26].

The exact recognition algorithm, optimal for the n-cube and in the sense of the Shannon complexity criterion, is given by G. Hansel in [11]. The algorithm is based on partitions of the n-dimensional binary cube into disjoint chains, that is effective and very much transparent and understandable. A direct generalization of Hansel's approach to the multi-valued case is obtained by V. Alekseev in [1]. For one particular sub-case this result is improved in [23].

As an example consider an applied problem that can use monotone recognition.

Assume we are given a set of m linear inequalities, $A \cdot X \leq B$ on the set of variables $X = (x_1, x_2, \ldots, x_n)$. In general, this system can be inconsistent. The problem is to find algorithmically one or all maximal consistent subsets of inequalities. If to observe that a subset of a consistent set of inequalities is consistent, and if to code the involvement of individual inequalities

* Corresponding author.
E-mail addresses: lasl@sci.am (L. Aslanyan), hasmik@ipia.sci.am (H. Sahakyan).

http://dx.doi.org/10.1016/j.dam.2016.04.008
0166-218X/

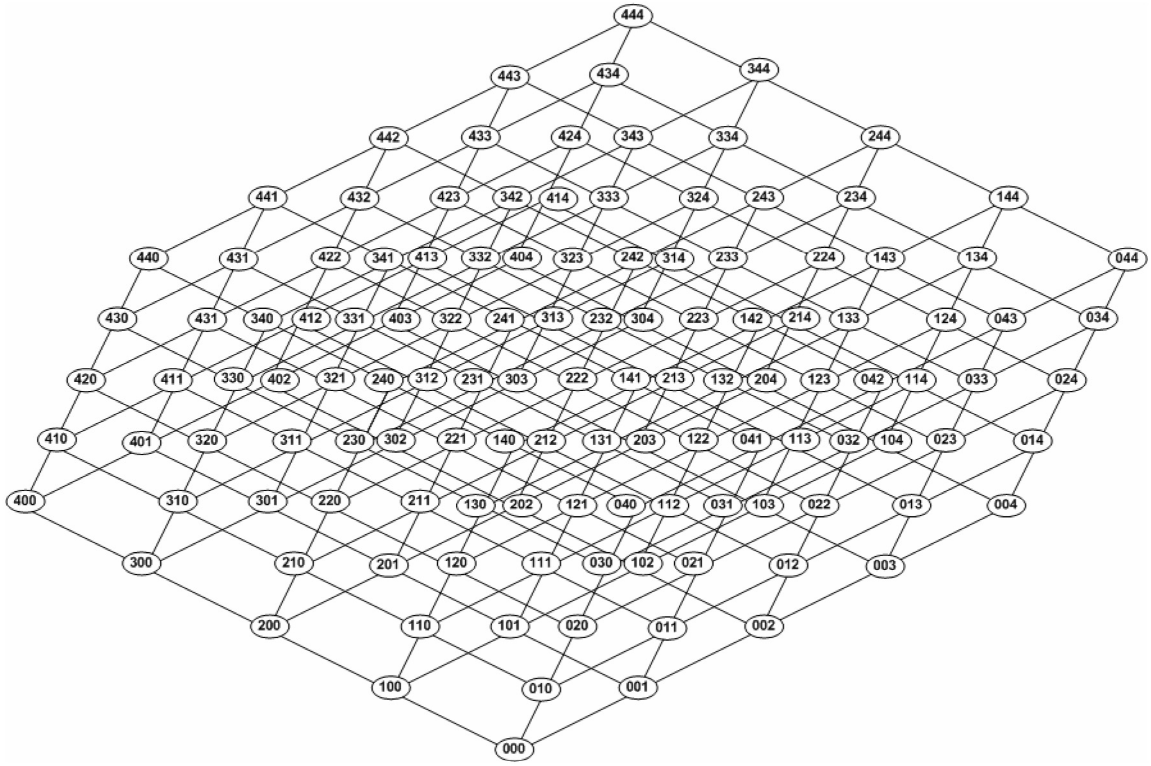

Fig. 1. Hasse diagram of Ξ_5^3, circles correspond to vertices.

into the subsystem under consideration by binary vectors, we get a monotone Boolean function, such that zero values of the function represent all consistent subsets of the system of inequalities. The maximal consistent sets correspond to the upper zeros of the monotone Boolean function.

Other typical applications appear in data mining area [22], where frequent sets of items compose monotone Boolean functions. This is for the case of market basket analysis. If to take into consideration the item quantities in the basket, then we obtain the same problem on multi-valued grids.

In this paper we consider a novel algorithmic resource for elaboration and identification of monotone functions defined on multi-valued grids. We propose a principally new approach based on partitioning of grids into non-intersecting discrete structures, that are isomorphic to binary cubes. Monotonicity of a function in the multi-valued grid implies monotonicity in all induced binary cubes. Monotonicity is retained also in induced multi-valued sub-grids. Hence, applying Hansel's method (and its extensions) for identification of monotone functions in all induced binary cubes and in sub-grids, and then integrating the results, leads to an alternative way of reconstruction of monotone functions defined on the multi-valued grids. The method can be used in practical algorithms of identification due to simple structures and easily calculable quantities in the n-cubes. In a general characterization, the new approach provides a binary cube partition technique vs. the chain partition technique used so far.

Let $\Xi_{m+1} = \{0, 1, \ldots, m\}$ and Ξ_{m+1}^n denote the set of vertices of the n-dimensional $(m+1)$-valued discrete grid defined as:

$$\Xi_{m+1}^n = \{(a_1, \ldots, a_n) : a_i \in \Xi_{m+1} \text{ for all } i \in \overline{1, n} = \{1, 2, \ldots, n\}\}.$$

We place a component-wise partial order on Ξ_{m+1}^n : $(a_1, \ldots, a_n) \le (b_1, \ldots, b_n)$ if and only if $a_i \le b_i$ for all $i \in \overline{1, n}$; and define the rank of an element $(a_1, \ldots, a_n)$ as: $a_1 + \cdots + a_n$. Then, Ξ_{m+1}^n is a ranked partially ordered set. Consider the geometric representation of Ξ_{m+1}^n through the Hasse diagram. The diagram has $m \cdot n + 1$ levels, numbered from 0 (lower level) to $m \cdot n$; the kth level contains all vertices at rank k. Edges connect those vertices in neighbor levels related by a cover relation. Let us demonstrate the Hasse diagram of Ξ_5^3 (see Fig. 1).

Consider a binary function f: $\Xi_{m+1}^n \to \{0, 1\}$. We say that f is *monotone* if for any two vertices $a, b \in \Xi_{m+1}^n$, if $a > b$ then $f(a) \ge f(b)$. For $m = 1$ we get monotone Boolean functions defined on the n-dimensional unit cube $E^n = \{(x_1, \ldots, x_n) : x_i \in \{0, 1\}$ *for all* $i \in \overline{1, n}\}$.

$a^1 \in \Xi_{m+1}^n$ is a *lower unit* of some monotone function f if $f(a^1) = 1$, and $f(a) = 0$ for every $a \in \Xi_{m+1}^n$, which is less than a^1. $a^0 \in \Xi_{m+1}^n$ is an *upper zero* of monotone function f if $f(a^0) = 0$, and $f(a) = 1$ for every $a \in \Xi_{m+1}^n$, which is greater than a^0.

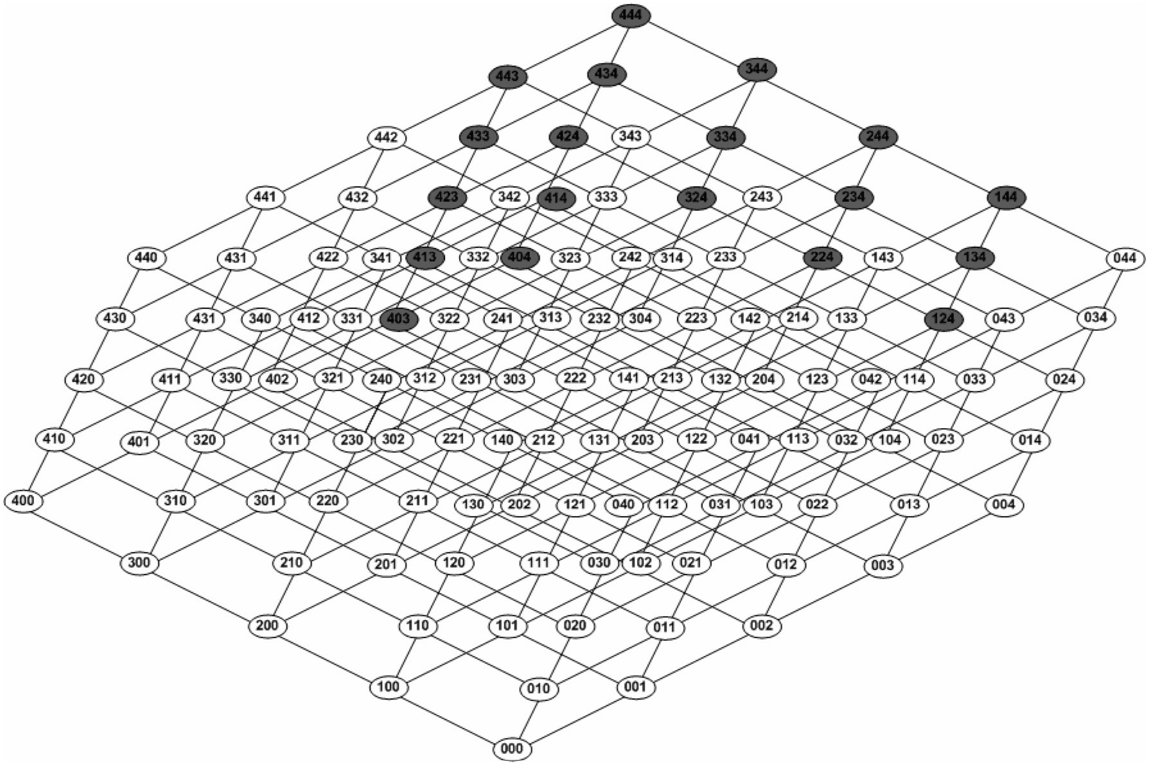

Fig. 2. Highlighted vertices are units of the function in the example.

We denote by $\mathcal{M}(n, m)$ the set of all monotone functions defined in Ξ_{m+1}^n. Consider an example: f is given in Ξ_5^3 in the following way:
$\{(4, 4, 4),$
$(4, 4, 3), (4, 3, 4), (3, 4, 4),$
$(4, 3, 3), (4, 2, 4), (3, 3, 4), (2, 4, 4)$
$(4, 2, 3), (4, 1, 4), (3, 2, 4), (2, 3, 4), (1, 4, 4),$
$(4, 1, 3), (4, 0, 4), (2, 2, 4), (1, 3, 4),$
$(4, 0, 3), (1, 2, 4)\}$
is the set of units of f, and the rest of the vertices of Ξ_{m+1}^n is the set of zeros of f. f is monotone, and $\{(4, 0, 3), (1, 2, 4)\}$ is the set of its lower units. Let us illustrate this function on Hasse diagram (see Fig. 2).

Let f be an arbitrary binary function defined with the help of a certain operator (also called an oracle) such that receiving any vertex a of Ξ_{m+1}^n, the oracle gives $f(a)$, the value of the function. If there is no a-priori information about the function then $(m+1)^n$ accesses to the oracle are needed for identification of the function. If some property of the function is known then it may not be needed to access to the oracle for all vertices of the cube. For example, if $f \in \mathcal{M}(n, m)$, then $f(a) = 1$ for some vertex $a \in \Xi_{m+1}^n$ implies $f(b) = 1$ for all $b > a$. All algorithms of identification of monotone Boolean functions differ from each other by the set of vertices presented to the oracle, and by the method of selecting vertices and diffusing their values to other vertices. The problem is in identification of the function by as far as possible small number of accesses to the oracle. In this context our approach divides the vertex selection process into subprocesses, by partitioning Ξ_{m+1}^n into sub-grids and binary n-cubes.

The paper is organized as follows: a brief overview of existing results on monotone recognition is given in Section 2. In Section 3 we introduce a method for decomposition of Ξ_{m+1}^n into structures isomorphic to binary cubes, such that monotonicity in Ξ_{m+1}^n implies monotonicity in all induced binary cubes. Section 4 concerns the problem of identification of monotone functions based on the decomposition of Ξ_{m+1}^n. The complexity of the considered algorithms was studied. Section 5 brings concluding remarks.

2. Current state/overview

Let A be an algorithm of identification of monotone Boolean functions defined on n-dimensional unit cube E^n. Let $\phi_A(f)$ denote the minimal number of accesses to the oracle which is sufficient to identify a given function f by the algorithm A. The oracle, by an input vertex α returns the value $f(\alpha)$. $\phi_A(n)$ denotes the minimal number of accesses to the oracle which is sufficient to identify an arbitrary monotone Boolean function of n variables by A. Then, $\phi(n) = min\phi_A(n)$, where the minimum is over all algorithms of identification of monotone Boolean functions. Note that we will use similar definitions and notations for the multi-valued case of functions.

For $\phi(n)$ several lower and upper bounds are obtained in [14] first. The complete solution for binary cubes is obtained by G. Hansel in [11]. In particular, it is known that

Theorem 1 *([11]).* $\phi(n) = C_n^{\lfloor n/2 \rfloor} + C_n^{\lfloor n/2 \rfloor + 1}$.

To prove the upper bound in [11] an algorithm is constructed based on a partition of the cube into special disjoint chains. Chains are interrelated by a property of "vertex complement" — when 3 sequential vertices have their complement in a shorter chain in the partition. The algorithm is simple and elegant, however it uses memory for keeping all vertices of E^n by the set of these chains.

In a formal description, a chain in E^n is a sequence of vertices, $\alpha_1 < \alpha_2 < \cdots < \alpha_t$, t denotes the length of the chain (we consider increasing chains). The chain is *continuous* if for each $\alpha_i < \alpha_{i+1}$ there is no $\alpha \in E^n$ such that $\alpha_i < \alpha < \alpha_{i+1}$. The set of all vertices of E^n can be organized as a special set of continuous chains, known as Hansel chains. The chains are disjoint, and the number of these chains is equal to $C_n^{\lfloor n/2 \rfloor}$. The number of the chains of length $n-2p+1$ is $C_n^p - C_n^{p-1}$, $0 \le p \le C_n^{\lfloor n/2 \rfloor}$. And finally, if $\alpha_1 < \alpha_2 < \alpha_3$ is a continuous fragment of some chain of length $n - 2p + 1$, then the *relative complement* to α_2 in the 2-dimensional subcube $[\alpha_1, \alpha_3]$ belongs to the chain of length $n - 2p - 1$. These properties of the Hansel chains compose the base of the Hansel's algorithm. The algorithm starts with the vertices of the chains of minimal length. In a recursive manner, the values of the function f on the chains of length $n - 2p - 1$ is diffused by monotonicity to the larger chains. In this way, at most 2 vertices of chains of length $n - 2p + 1$ may remind undetermined and this provides the upper estimate in Theorem 1.

A similar algorithm but with minimal use of memory is developed by G. Tonoyan in [26]. Here a special algebra is developed that helps to compute the next vertex to be presented to the oracle. The algorithm receives as input not the chains themselves but the information accumulated by the algorithm (vertex set) about the function under recognition.

An algorithm that is optimal with respect to the number of accesses to the oracle, as well as the used memory is obtained later on also by N. Sokolov [24,25].

The problem of identification of monotone functions for the multi-valued case is investigated by V. Korobkov, V. Alekseev, A. Serjantov, and others [14,1,23].

An extension of the model of monotone function recognition is considered in [14] by V. Korobkov. He considered the Cartesian degree of an arbitrary finite partially ordered set R, and functions, accepting values 0, 1. Let $\phi_R(n)$ denote the minimal number of accesses to the oracle for identification of these functions. The following estimate is formulated by V. Korobkov.

Theorem 2 *([14]).*

$$C_1(R)\frac{|R|^n}{\sqrt{n}} \le \phi_R(n) \le C_2(R)\frac{|R|^n}{\sqrt{n}}, \tag{2.1}$$

where $C_1(R)$ and $C_2(R)$ are constants depending on R.

We will specify this result for a particular case in Section 4.

In [1] an algorithm U_0 (generalization of Hansel's algorithm) is constructed by V. Alekseev for identification of monotone binary functions defined on the grid $\Xi_{k_1 k_2 \dots k_n} = \Xi_{k_1} \times \Xi_{k_2} \cdots \times \Xi_{k_n}$, where $\Xi_{k_i} = \{0, 1, \dots, (k_i - 1)\}$.

Theorem 3 *([1]).*

$$\frac{t(U_0)}{t(U_{opt})} \le \frac{1}{2}\lceil log_2(k-1)\rceil,$$

where $t(U_0)$ is the complexity of the algorithm U_0 and $t(U_{opt})$ is the complexity of an optimal algorithm U_{opt}, and $k = maxk_i$.

Let M and N denote the sets of vertices of middle levels of the grid, defined as:

$$M = \left\{(a_1, \dots, a_n) \in \Xi_{k_1} \times \Xi_{k_2} \cdots \times \Xi_{k_n} | a_1 + \cdots + a_n = \left\lceil \frac{1}{2}\sum_{i=1}^{n}(k_i - 1)\right\rceil\right\},$$

$$N = \left\{(a_1, \dots, a_n) \in \Xi_{k_1} \times \Xi_{k_2} \cdots \times \Xi_{k_n} | a_1 + \cdots + a_n = \left\lceil \frac{1}{2}\sum_{i=1}^{n}(k_i - 1)\right\rceil + 1\right\}.$$

It is known also [1] that

$$t(U_{opt}) \ge |M| + |N|$$

and

$$t(U_0) \leq |M| + \lfloor log_2 k \rfloor \cdot |N|.$$

Thus, the complexity of Alekseev's algorithm for monotone functions defined on Ξ^n_{m+1} is:

$$t(U_0) \leq |M'| + \lfloor log_2(m+1) \rfloor \cdot |N'|,$$

where

$$M' = \left\{ (a_1, \ldots, a_n) \in \Xi^n_{m+1} | a_1 + \cdots + a_n = \left\lceil \frac{m \cdot n}{2} \right\rceil \right\},$$
$$N' = \left\{ (a_1, \ldots, a_n) \in \Xi^n_{m+1} | a_1 + \cdots + a_n = \left\lceil \frac{m \cdot n}{2} \right\rceil + 1 \right\}.$$

The notion of cardinalities of two middle levels is unclear and this description passes from publication to publication. Below in Section 4 we will bring clarifications. On the one hand, formulas for the cardinalities of the middle levels of the grid can be brought out by the help of [17,12,3,4], where a formula for the number of ordered partitions of the given integer into the given number of parts, each of size at least 0 but no larger than the given size, is introduced. From the other — we will use the technique of [14,27] deepening this result for a particular case. Combining these analyses, at least in an asymptotic for large n, m yields a simple and closed-form expression for complexities of the recognition (cardinalities of M and N).

3. On a special decomposition of the multi-valued cube

The section introduces a split technique and a special decomposition of Ξ^n_{m+1} through the structures, isomorphic to binary cubes. Our global challenge is in designing the effective constructions in Ξ^n_{m+1} that serve the needs of modeling of monotone relationships and their recognition tasks. Similar studies for binary cubes were started by R. Dedekind, E. Gilbert and others a century ago (see [14]). The major result that solves part of the basic issues is obtained by G. Hansel [11] who invented a perfect partitioning of E^n into the set of "complementary" chains (*chain-split*). In this context we introduce a technique of partitioning Ξ^n_{m+1} into a set of non-intersecting binary cubes. We will call it *cube-split* technique. In this manner, the known effective constructions designed so far for the unit cube domain, can be transferred through the cube-split technique to the area of Ξ^n_{m+1}. In fact, for structuring Ξ^n_{m+1} there are two perspectives:
(1) to develop G. Hansel type chain-splitting algorithms for Ξ^n_{m+1}, and
(2) to reduce constructions for Ξ^n_{m+1} to the binary cube domain and structures, and then use existing approaches for this domain.

The main achievement in (1) is by V. Alekseev [1] who introduced a proper Hansel type chain-split but these chains are specific, and they use *long collinear segments* (the case of sequential increase of the same component) that lead to the increase of algorithmic complexities. If to construct such chains but with an additional "zigzag" property, then the minimal asymptotic algorithmic complexity of monotone recognition will be achieved. In [23] chains with no collinear paths were constructed for one particular sub-case. In this section we will apply the cube-split technique, which is really effective and moves (2) ahead. On a theoretical level this provides the splitting of Ξ^n_{m+1} recursively into smaller domains of the same type objects — multidimensional multi-valued grids (*grid type partition*), and then into binary cubes and chains of smaller sizes. On a practical level of recursion it is important that the cube-split level makes the transfer of effective tools of chain computations [26,5] into the domain of grid computations possible.

The main problem in this domain is that in real size applications it becomes hard to provide the necessary computations over Ξ^n_{m+1}. At least, it is even hard to keep all the set of chain information in memories despite these chains and their algorithms are very effective from a theoretical point of view. In case of binary spaces the problem is solved by computational algebras [26]. Having no such constructions and algebras for Ξ^n_{m+1}, the cube-split reduction can be an effective way of transferring the power of cube–chain-algebras into the Ξ^n_{m+1} area.

Before introducing the special decomposition of grids we first bring some necessary concepts, and distinguish several classes of vertices in Ξ^n_{m+1}.

Middle vertices.
$m_{mid+} = (\lceil m/2 \rceil, \ldots, \lceil m/2 \rceil)$ and $m_{mid-} = (\lfloor m/2 \rfloor, \ldots, \lfloor m/2 \rfloor)$ we call *middle vertices* of Ξ^n_{m+1}. These are exactly the geometric "centers" of the structure Ξ^n_{m+1}. It is obvious that these two vertices coincide for even m.

Upper vertices, **lower vertices**. A vertex $(a_1, \ldots, a_n)$ of Ξ^n_{m+1} is called *upper vertex* if $(a_1, \ldots, a_n) \geq m_{mid+}$. A vertex $(a_1, \ldots, a_n)$ of Ξ^n_{m+1} is called *lower vertex* if $(a_1, \ldots, a_n) \leq m_{mid-}$. $\hat{H}$ and $\check{H}$ denote the sets of all upper and lower vertices, respectively.

$$|\hat{H}| = |\check{H}| = ((m+1)/2)^n \quad \text{for odd } m.$$
$$|\hat{H}| = |\check{H}| = (m/2+1)^n \quad \text{for even } m.$$

Fig. 3 maps the sets $\hat{H}$ and $\check{H}$ in Ξ^3_5.

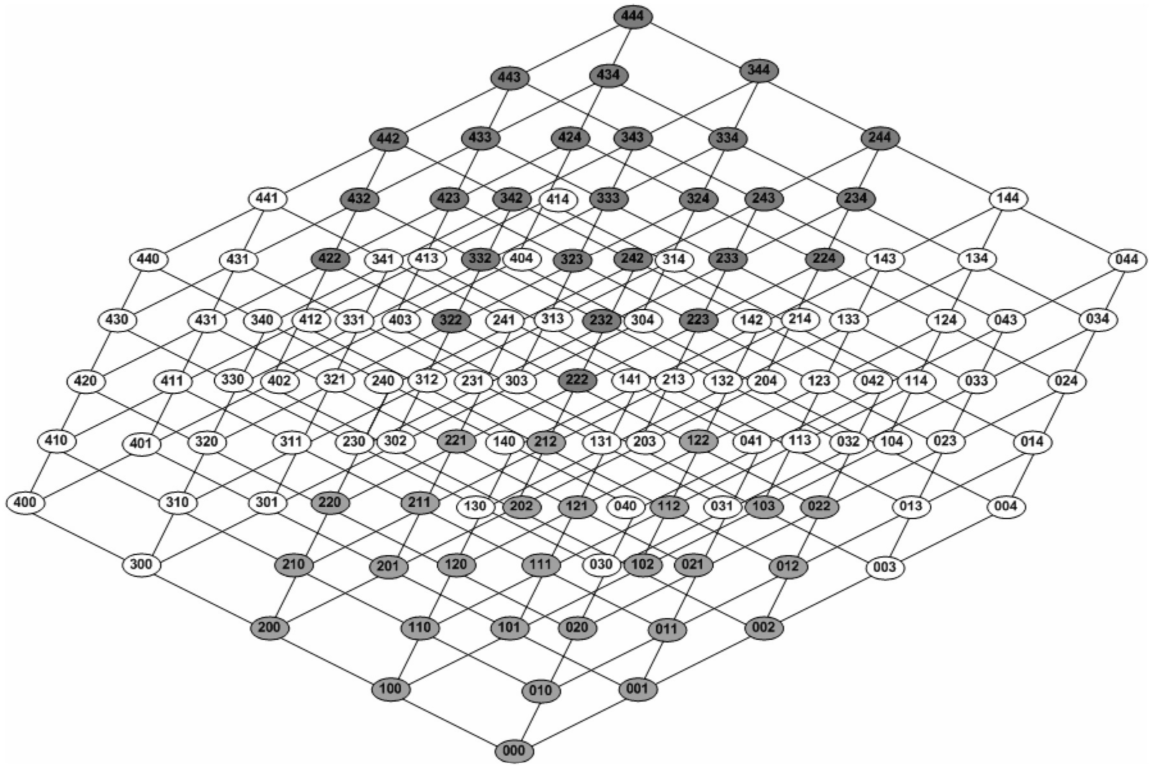

Fig. 3. The set of highlighted vertices composes $\hat{H}$ and $\check{H}$, where the darker part together with the vertex 222 composes $\hat{H}$, and the lighter part with the vertex 222 composes $\check{H}$.

***V*-equivalence**.
Vertices $(a_1, \ldots, a_n)$ and $(b_1, \ldots, b_n)$ of Ξ^n_{m+1} are *vertically equivalent* if $a_i \in \{b_i, m - b_i\}$ for $1 \le i \le n$. $V(a)$ denotes the V-equivalence class of a vertex a. In $V(a)$ we distinguish two vertices $\hat{a}$ and $\check{a}$, with components defined as follows:

$$\hat{a}_i = \begin{cases} a_i, & \text{if } a_i \ge m_{mid+} \\ m - a_i, & \text{if } a_i < m_{mid-} \end{cases}$$

$$\check{a}_i = \begin{cases} m - a_i, & \text{if } a_i > m_{mid+} \\ a_i, & \text{if } a_i \le m_{mid-}. \end{cases}$$

These are the only vertices of $V(a)$ that belong to $\hat{H}$ and $\check{H}$, respectively. Thus all vertices of $V(a)$ can be constructed from the upper and/or lower elements (similarly, from an arbitrary vertex) of $V(a)$ inverting (with respect to m) all groups of components. It is evident that the V-equivalence classes of different vertices of $\hat{H}$ ($\check{H}$) are disjoint.

This provides a complete partition of Ξ^n_{m+1} into $|\hat{H}|$ equivalence classes, which are uniquely defined by elements of $\hat{H}$. For a given $a \in \Xi^n_{m+1}$ consider an integer defined as $k = |\{a_i | a_i \ne (m - a_i)\}|$ ($k = n$ for odd m). Then $|V(a)| = 2^k$. We identify each vertex of $V(a)$ with α, a binary sequence of length n, such that $\alpha_i = 1$ if and only if $a_i = \hat{a}_i$. In this manner, $V(a)$ becomes isomorphic to the k-dimensional binary cube E^k: the 0th level of E^k contains the lower vertex of $V(a)$ belonging to $\check{H}$; the ith level consists of all vertices of $V(a)$ obtained from the lower vertex by applying i number of component inversions. Thus Ξ^n_{m+1} is partitioned into $|\hat{H}|$ disjoint classes — that are identical by structure to binary cubes. This partitioning of Ξ^n_{m+1} we will refer as *cube type* partition. Fig. 4 illustrates the equivalence classes of points (3, 4, 3), (2, 3, 4) and (4, 2, 2) in Ξ^3_5.

It is worth to mention that in the usual chain-split (as is the partition of the binary cube in [11]), vertices in each chain are arranged "continuously", level by level in the mother space, whereas in this case, in case of cube-split, edges of cubes connect, in general, vertices that do not belong to the neighbor levels of Ξ^n_{m+1}.

In case of odd m we obtain $((m+1)/2)^n V$-equivalence classes, and every class/binary-cube has dimension n. The following formula relates sizes and partitions for odd m.

$((m + 1)/2)^n \cdot 2^n = (m + 1)^n$.

The case of even m is not so homogeneous. For vertices of $\hat{H}$ let k be the number of components, not equal to $m/2$, $0 \le k \le n$. Thus $n - k$ components are fixed to $m/2$ and k components are actual, creating $(m/2)^k$-vertex sets in $\hat{H}$. Each of

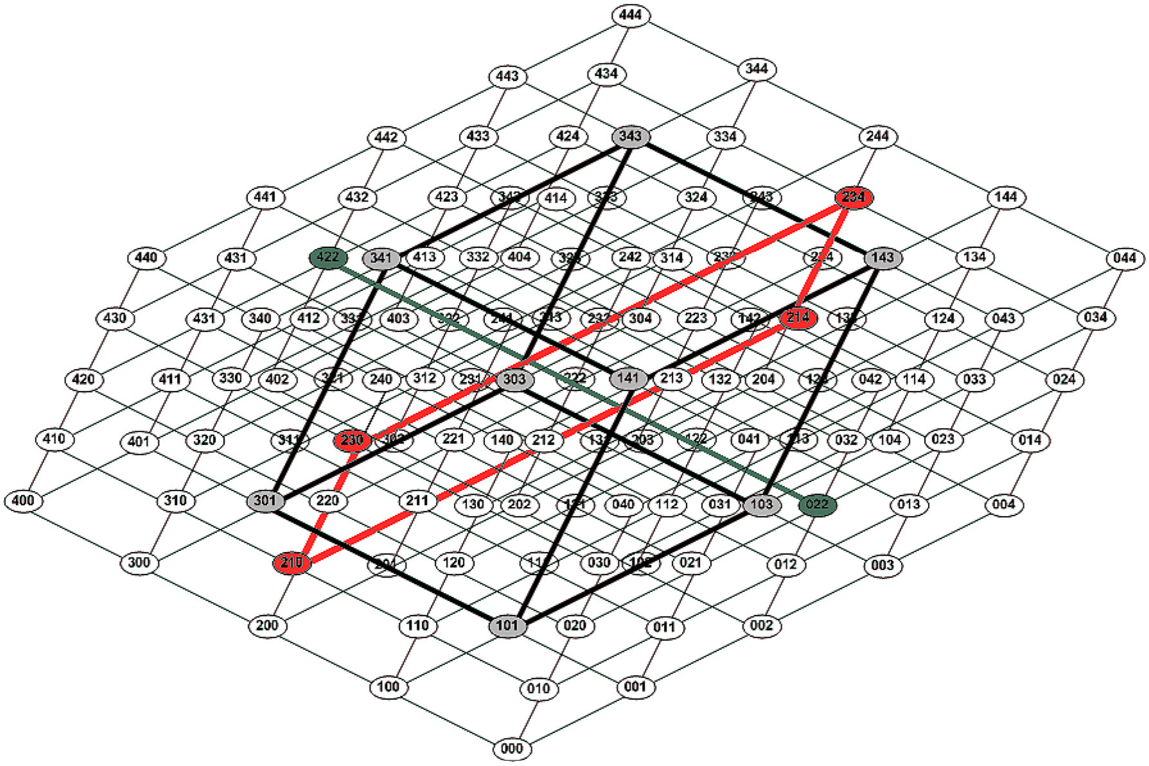

Fig. 4. Highlighted vertices are elements of $V(3, 4, 3)$, $V(2, 3, 4)$ and $V(4, 2, 2)$, *respectively*. $V(3, 4, 3)$ composes a 3-dimensional, $V(2, 3, 4)$ — a 2-dimensional, and $V(4, 2, 2)$ — a 1-dimensional binary cube.

these vertices has its V-equivalence class of size 2^k. Schematically these relations are demonstrated in Fig. 5. The following formula relates sizes and partitions for even m.

$\sum_{k=0}^{n}(C_n^k \cdot 2^k \cdot (m/2)^k) = \sum_{k=0}^{n}(C_n^k \cdot m^k) = (m+1)^n$.

We will apply two types of partitions in Ξ_{m+1}^n: *cube type* and *grid type*, where the *grid type* means an arbitrary partition of Ξ_{m+1}^n into the set of disjoint sub-grids covering Ξ_{m+1}^n.

Define a multi step partitioning of Ξ_{m+1}^n in the following way:

In step i of the partitioning an arbitrary set $\Xi_g(i)$ of current sub-grids is selected for further grid-partitioning (initially we start with Ξ_{m+1}^n), and the reminding set $\Xi_c(i)$ of sub-grids is partitioned into cubes. Let $\mathcal{V}_i$ denote the set of equivalence classes of $\Xi_c(i)$. Let $p \geq 1$ steps be applied, and $\mathcal{V}_1, \mathcal{V}_2, \ldots, \mathcal{V}_p$ are the corresponding sets of vertical equivalence classes.

Consider V, some vertical equivalence class in $\mathcal{V}_i$, and let $E(V)$ denote the corresponding sub-cube in E^n. Splitting of $E(V)$ into Hansel chains will induce the splitting of V: each chain $\alpha_1 < \alpha_2 < \cdots < \alpha_t$ in $E(V)$ will induce a sequence $a_1 < a_2 < \cdots < a_t$ of corresponding vertices in Ξ_{m+1}^n, where two neighbor vertices in the sequence can belong to non neighbor levels. Henceforth these sequences will be referred to as *associated-chains*, a_i as *origin* vertices, and α_i — as *induced* vertices. The above description is summarized by the following lemma.

Lemma 1. *Let Ξ_{m+1}^n is multi-step partitioned, and V be a vertical equivalence class obtained in some step of the partitioning. If $\alpha_1 < \alpha_2 < \cdots < \alpha_t$ is a chain in $E(V)$, then $a_1, a_2, \ldots, a_t$, is the associated chain in Ξ_{m+1}^n, where $a_1, a_2, \ldots, a_t$ are origins of $\alpha_1, \alpha_2, \ldots, \alpha_t$. Conversely, if $a_1 < a_2 < \cdots < a_t$ belong to Ξ_{m+1}^n, and induced vertices $\alpha_1, \alpha_2, \ldots, \alpha_t$ belong to the same binary cube, then $\alpha_1, \alpha_2, \ldots, \alpha_t$ is a chain there.*

A corollary is obtained in the form of:

Theorem 4. *Let Ξ_{m+1}^n be multi-step partitioned, and V be a vertical equivalence class obtained in some step of the partitioning. If $F : \Xi_{m+1}^n \to \{0, 1\}$ is a monotone function then $f : E(V) \to \{0, 1\}$ is monotone, where f is defined as follows: for every $\alpha \in E(V)$, $f(\alpha) = 1$ if and only if $F(a) = 1$, where a is the origin of α in Ξ_{m+1}^n.*

Thus we achieved a partitioning of the multi-valued grid into binary cubes in such a way that monotonicity is retained.

The most valuable practical property of multi-step partitioning is in recursive use of the chain-computation-algebra [26].

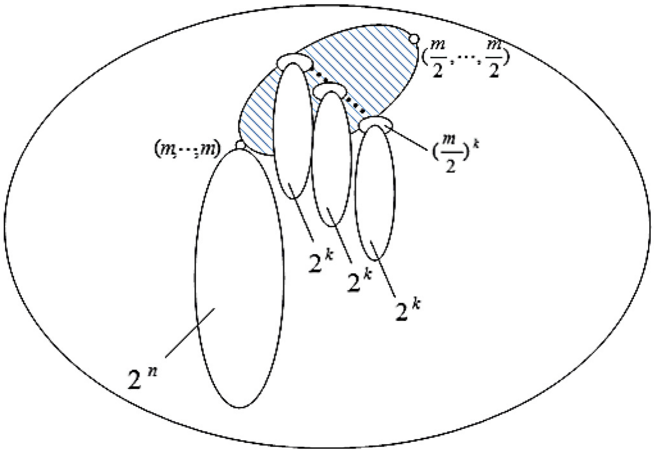

Fig. 5. Scheme of partitioning for even m. The dashed part is $\hat{H}$.

4. Identification of monotone functions

In this section we will use two cube-type partition algorithms for identification of monotone functions defined on Ξ^n_{m+1}. Let $f : \Xi^n_{m+1} \to \{0, 1\}$ be a monotone function defined with the help of an oracle Ω_f.

Algorithm 1. • Partition the grid Ξ^n_{m+1} into binary cubes $E_1, \ldots, E_{|\hat{H}|}$ as it is described in the previous section.
- In each E_i, apply Hansel's algorithm for identification of monotone functions $f_i : E_i \to \{0, 1\}$, defined as follows: for every $\alpha \in E_i, f_i(\alpha) = 1$ if and only if $f(a) = 1$, where a is the origin of α in Ξ^n_{m+1}.
- Integrate the results of $|\hat{H}|$ binary recognitions into f.

In fact, Algorithm 1 (let us denote it by A1) can be considered in an arbitrary step of the multi-step partitioning of Ξ^n_{m+1}. Let p is the number of steps. For each fragment in the ith step, the partition produces a set of collinear binary cubes. It is worth to mention, that in each particular step we consider all induced cubes and treat them independently, although comparison of vertices could improve the result of recognition. The complexity of recognition (by an oracle) for one separate cube is estimated as the sizes of its two middle layers. This is an overestimate in global terms, but we pay this cost for parallelization. There are many other ways to the optimized parallelization. Above the simplest schemes that consider one proper level of p partitions, the Cartesian products of chains of levels $1, 2, \ldots, i \le p$ can be considered. They do not compose a binary cube but are cube like and allow a proper chain split. These constructions are larger, which means that more internal comparisons are involved in individual "sub-cubes" of our consideration. Let us just outline the typical costs for parallelization.

Theorem 5. *Let $f : \Xi^n_{m+1} \to \{0, 1\}$ be a monotone function defined with the help of an oracle Ω_f. Let Δ_{A1} denote the Shannon complexity of* A1. *Then,*

$$\Delta_{A1} = \sum_{k=0}^{n} (C_n^k \cdot (m/2)^k \cdot (C_k^{\lfloor k/2 \rfloor} + C_k^{\lfloor k/2 \rfloor + 1})) \quad \textit{for even } m \textit{ and} \tag{4.1}$$

$$\Delta_{A1} = ((m+1)/2)^n \cdot (C_n^{\lfloor n/2 \rfloor} + C_n^{\lfloor n/2 \rfloor + 1}) \quad \textit{for odd } m. \tag{4.2}$$

Proof. The proof is obtained by Theorem 1, Theorem 4 and results of [1]. Sum in (4.1) corresponds to a partition of the cubes of level $p = 1$ by the number of their coordinates that are equal to $m/2$. These coordinates, evidently, cannot be converted, — they are fixed. And the size of some currently considered cube is determined by the number of the reminder coordinates.

Below we estimate the real numeric value of these complexities. Consider the case of odd m. First we estimate the term $C_n^{\lfloor n/2 \rfloor}$ asymptotically. For this we apply the well known formula from [9]

$$C_n^k \sim \frac{2^{n+1}}{\sqrt{2\pi n}} \cdot \exp\left(-\frac{(2k-n)^2}{2n}\right), \quad \text{when } n, k \to \infty, \text{ and } k - \frac{n}{2} = o(n^{3/4}).$$

Inserting this into (4.2) we obtain that

$$\Delta_{A1} \to ((m+1)/2)^n \cdot 2 \cdot \frac{2^{n+1}}{\sqrt{2\pi n}} = \frac{4(m+1)^n}{\sqrt{2\pi n}} \quad \text{for odd } m \text{ when } n \to \infty.$$

For even m consider the summand of (4.1) with $k = m$. It is easy to see that it is asymptotic to $\frac{4m^n}{\sqrt{2\pi n}}$ when $n \to \infty$. The sub-sum for small k, say up to $n/2$ is small. The maximal summand is located close to n. To check this, we form the fraction of terms $C_n^k \cdot (m/2)^k \cdot \frac{2^{k+1}}{\sqrt{2\pi k}}$ for $k+1$ and k. Equaling this to 1 we get $k = n - \frac{(k+1)\sqrt{1+1/k}}{m}$ which proves the claim. What is important is the order of the complexity which is $\frac{4m^n}{\sqrt{2\pi n}}$.

Simple structures, and easily calculable quantities used in Algorithm 1 can serve as important elements in practical identification algorithms. On the other hand, parallel computation of this recognition is possible, as we deal with disjoint binary cubes.

Algorithm 2.
- Partition Ξ_{m+1}^n into binary cubes $E_1, \ldots, E_{|\hat{H}|}$ as it is described in the previous section and let G be the associated set of sub-grids of this partition.
- In each $g \in G$, apply Alekseev's algorithm for identification of monotone function $f_g : g \to \{0, 1\}$, defined as follows: for every $\alpha \in g, f_g(\alpha) = 1$ if and only if $f(a) = 1$, where a is the origin of α in Ξ_{m+1}^n.
- Integrate the results of $|\hat{H}|$ binary recognitions.

Theorem 6. *Let $f : \Xi_{m+1}^n \to \{0, 1\}$ be a monotone function defined with the help of an oracle. Consider m odd. f can be identified by Algorithm 2 using $(|M_{\hat{H}}| + \lfloor \log_2 m \rfloor \cdot |N_{\hat{H}}|) \cdot (C_n^{\lfloor n/2 \rfloor} + C_n^{\lfloor n/2 \rfloor + 1})$ accesses to the oracle, where $M_{\hat{H}}$ and $N_{\hat{H}}$ denote the sets of vertices of middle levels of $\hat{H}$ (levels which contain vertices of rank $\lfloor mn/4 \rfloor$, $\lfloor mn/4 \rfloor + 1$ respectively). The formula for even m can be derived in a similar way.*

Proof. The claim follows from Theorem 1, Theorem 4 and [1].

Algorithm 2 (let us denote it by A2) can be considered for an arbitrary step $i \le p$ of the cube-split procedure. For each i the partition produces a set of collinear multi-valued-multidimensional grids. We consider grids of one step and treat them independently although there are comparisons between their vertices that can be employed for a better result in recognition. Considering one level we use the fact that the complexity of (oracle) recognition on one separate sub-grid appears as a simple function of the sizes of the two middle layers of that grid. This is an overestimate but we pay this cost for parallelization.

Again, having many other ways to the optimized parallelization let us just outline the typical costs for parallelization in case of Algorithm 2.

For further estimates we refer to two references that contain initial formulas.

[17] considers the number $w_{n,m}(N)$ of ordered partitions of the integer N into n parts each of size at least 0 but not larger than m, supposing that $1 \le n \le N$. The following formula is derived for $w_{n,m}(N)$:

$$w_{n,m}(N) = \sum_{i=0}^{n} (-1)^i C_n^i C_{N+n-1-i(m+1)}^{n-1}. \tag{4.3}$$

The formula, implicitly controls the part of positive summands of (4.3). When $N+n-1-i(m+1)$ is negative or 0, or when it is $< n-1$ the summand becomes 0. It is correct to add these restrictions in an explicit way.

Then [17] derives an asymptotic estimate for $w_{n,m}(N)$ by the use of saddle point evaluation of integrals. Here is the result: Let $N, m \ge 1, n = \frac{2N}{m}(1 + c_1(N^{-1/2}))$ for some absolute constant $c_1 > 0$. Let $\sigma = \frac{m}{6}(1 + \frac{m}{2})$ and $N \to \infty$, then

$$w_{n,m}(N) = \frac{1}{\sqrt{2\pi n\sigma}}(m+1)^n \exp\left(\frac{-(N - \frac{mn}{2})^2}{2n\sigma}\right)(1 - 2\Phi(-n^{-1/10}\sqrt{\sigma}) + O(\exp(-n^{1/5}))). \tag{4.4}$$

Here $\Phi(x)$ is the Gaussian cumulative distribution, and the approximation error rate of the formula is $O(n^{-1/5})$. Formula (4.4) is useful but its asymptotic estimate is hard to use in our case. Rewrite condition $n = \frac{2N}{m}(1 + c_1(N^{-1/2}))$ in an equivalent form $\frac{mn}{2} = N + c_1N^{-1/2}$. This shows that the formula does not work around $N = \frac{mn}{2}$ which is the basic component in Alekseev's algorithm complexity.

Now we refer to [14] to consider its particular, but a deeply investigated sub-case. This result of [14] we have formulated as (2.1).

Surprisingly, this postulation completes estimates of [17] helping to understand clearly the complexity of monotone recognition and thus, — the cost we pay for parallelization. To justify this we need to enter into some elements of the proof of (4.4) in [14].

Consider the set $R = \{r_1, r_2, \ldots, r_s\}$. Map the elements of R to layers on the plane so that comparable elements are placed only in nearby layers, see [7]. Some $r \le s$ layers will be used, and let us denote $N(R, 1, k) = s_k$. Generalizing, consider the Cartesian power R^n of R and note by $N(R, n, k)$ the number of elements of layers $k, k = 0, 1, \ldots, nr$.

Consider (on R^n) n identically distributed random variables ξ_i that attain integer values j, $(0 \le j \le r)$ by probabilities $\frac{s_j}{s}$, respectively. It can be concluded easily that $N(R, n, k) = P\{\sum_{i=1}^n \xi_i = k\} \cdot s^n$. It follows by the use of B. Gnedenko's theorem [10] that uniformly by k and with $n \to \infty$

$$P\left\{\sum_{i=1}^{n} \xi_i = k\right\} \to \frac{1}{\sqrt{2\pi n}\sqrt{D\xi}} \cdot \exp^{-\frac{(k-nM\xi)^2}{2nD\xi}}, \tag{4.5}$$

where $M\xi$ and $D\xi$ denote the expected value and dispersion of the random number ξ. (4.4) is a consequence of this formula. It is necessary to note that the expressions in this formula do not obey the supposition that coefficients $c_1(R)$ are $c_2(R)$ are constants. Consider in detail the case of Ξ_{m+1}^n.

In Ξ_{m+1}^n poset R is just a simple linearly ordered set $\Xi_{m+1} = 0, 1, \ldots, m$. Compute the values of the probabilistic distribution in this case.

$$M\xi = \sum_{k=0}^{m} \frac{k}{m+1} = \frac{m}{2} \quad \text{and} \quad D\xi = \sum_{k=0}^{m} \frac{k^2}{m+1} - \left(\frac{m}{2}\right)^2 = \frac{m(2m+1)}{6} - \frac{m^2}{2} = \frac{m(m+2)}{12}.$$

Substituting $M\xi$ and $D\xi$ into (4.5) we obtain exactly the three first multipliers in expression (4.4), $\frac{1}{\sqrt{2\pi n\sigma}}(m+1)^n \exp(\frac{-(N-\frac{mn}{2})^2}{2n\sigma})$. In this way we eliminated the constraint $c_1 > 0$ in estimate (4.4) and obtain a simple and closed-form estimate of an Alekseev type algorithm of multidimensional multi-valued monotone recognition.

Different versions and combinations of the two given algorithms are possible. It is also possible to combine the partition cubes and their chains together with initial Alekseev chains. The main objective could be resolving the work with chains in one direction hardening Alekseev's algorithm. Concerning the issue of complexity of algorithms it is to mention that the cube-split is not simpler than the chain-split algorithm, but it is extensively parallel and when a large number of processors are available a total computation reduction can be achieved together with simple and interpretable constructions. So we value the simple but effective grid split technique for the domain of monotone recognition. Another value of the technique is for the domain of data mining algorithms, because of the match of frequent subset recognition of association rule mining tasks to the monotone recognition technique considered in this paper.

5. Concluding remarks

Monotone Boolean functions and their extensions appear in diverse applications. The algorithmic and complexity issues of monotone recognition are extensively studied. Unexpected new relations were found between the monotone recognition structures of the multi-valued and the binary cases. The binary case is better investigated and the constructions are optimal in terms of the Shannon criterion. The multi-valued case is harder with respect to constructions and interpretations. So the cube-split technique introduced in this work aims to set up a bridge between these two domains. The outcome is in the form of simple and predictable constructions for multi-valued grids that open further investigation and application perspectives. In a simple but valuable form the introduced cube-split technique provides effective means for parallel computations which is very valuable in our age of high performance computation with supercomputers, clusters, grids and cloud applications.

References

[1] V. Alekseev, On deciphering of some classes of monotone many valued functions, J. Comput. Math. Math. Phys. 16 (1) (1976) 189–198.
[2] L. Aslanyan, The discrete isoperimetry problem and related extremal problems of discrete spaces, Probl. Kibern. 36 (1979) 85–127.
[3] L. Aslanyan, H. Danoyan, On the optimality of the Hash-Coding type nearest neighbour search algorithm, in: CSIT 2013 - 9th International Conference on Computer Science and Information Technologies, Revised Selected Papers, 2013, http://dx.doi.org/10.1109/CSITechnol.2013.6710336.
[4] L. Aslanyan, H. Danoyan, Complexity of hash-coding type search algorithms with perfect codes, J. Next Gener. Inf. Technol. 5 (4) (2014) 26–35.
[5] L. Aslanyan, R. Khachatryan, Association rule mining with n-dimensional unit cube chain split technique, Inf. Theory Appl. (ISSN: 1310-0513) 1 (2) (2008) 108–125.
[6] L. Aslanyan, H. Sahakyan, Numerical characterization of n-cube subset portioning, in: Electronic Notes in Discrete Mathematics, vol. 27, Elsevier B.V., ODSA, 2006, October 2006 — Conference on Optimal Discrete Structures and algorithms.
[7] G. Birkhoff, Lattice Theory, American Mathematical Society Colloquium Publications, revised ed., vol. 25, American Mathematical Society, N.Y, 1948.
[8] Konrad Engel, Sperner Theory, Cambridge University Press, Cambridge, 1997.
[9] W. Feller, An Introduction to Probability Theory and Its Applications, Vol. I, third (revised) ed., ISBN: 9780471257080.
[10] B. Gnedenko, On a local limit theorem of probability theory, Uspekhi Math. Nauk 3 (3(25)) (1948) 187–194.
[11] G. Hansel, Sur le nombre des fonctiones booleennes monotones de n variables, C. R. Acad. Sci. (1966) 262.
[12] N. Katerinochkina, On sets, containing maximal number of pair wise incomparable n-dimensional k-valued vectors, Math. Notes 24 (3) (1978).
[13] D. Kleitman, On Dedekind's problem: the number of monotone Boolean functions, Proc. Amer. Math. Soc. 21 (1969) 677–682.
[14] V. Korobkov, On monotone functions of algebra of logic, Prob. Cyb. 13 (1965).
[15] A. Korshunov, On the number of monotone Boolean functions, Prob. Cyb. 38 (1981).
[16] A. Korshunov, Monotone Boolean functions, Uspekhi Mat. Nauk 58 (5(353)) (2003) 89–162.
[17] J. Ratsaby, Estimate of the number of restricted integer-partitions, Appl. Anal. Discrete Math. 2 (2008) 222–233.
[18] H. Sahakyan, Numerical characterization of n-cube subset partitioning, Discrete Appl. Math. 157 (2009) 2191–2197.
[19] H. Sahakyan, Essential points of the n-cube subset partitioning characterization, Discrete Appl. Math. 163 (2) (2014) 205–213.
[20] H. Sahakyan, On the set of simple hypergraph degree sequences, Appl. Math. Sci. 9 (5) (2015) 243–253.
[21] H. Sahakyan, L. Aslanyan, On structural resource of monotone recogniton, Information Theories and Applications (ISSN: 1310-0513) 15 (4) (2008) 285–289.
[22] H. Sahakyan, L. Aslanyan, Chain split of partially ordered set of k-subsets, New Trends in Information Technologies, ITHEA (2010) 55–65.

[23] A. Serjantov, Optimal algorithm for deciphering of some classes of monotone functions, Journal of Computational Mathematics and Math. Physics 23 (1) (1983) 206–212.
[24] D. Sokolov, On optimal deciphering of monotone functions of logic algebra, Journal of Computational Mathematics and Math. Physics 22 (12) (1982) 1878–1887.
[25] D. Sokolov, Optimal deciphering of monotone Boolean functions, J. Comput. Math. Math. Phys. 27 (2) (1987) 449–461.
[26] G. Tonoyan, Chain partitioning of n-cube vertices and deciphering of monotone Boolean functions, Journal of Computational Mathematics and Math. Physics 19 (6) (1979).
[27] N. Zolotykh, A. Chirkov, On the complexity of deciphering of threshold functions of many-valued logic, in: Proceedings of the XI International Seminar "Discrete Mathematics and its Applications", Commemorate on the 80th Anniversary of Academician Lupanov, 18–23 June, 2012, 2012, pp. 63–77.