アルゴリズム 73－6
(2000. 5. 19)

双対制限されたハイパーグラフ: 部分横断と多重横断の列挙について

Endre Boros[†], Vladimir Gurvich[‡], Leonid Khachiyan[§], 牧野和久[¶]

[†]RUTCOR, Rutgers University, 640 Bartholomew Road, Piscataway NJ 08854.
boros@rutcor.rutgers.edu

[‡]Russian Academy of Sciences, Moscow, Russia. gurvich@rutcor.rutgers.edu

[§]Department of Computer Science, Rutgers University, New Brunswick, New Jersey 08903.
leonid@cs.rutgers.edu

[¶]560-8531 大阪府豊中市待兼山町 1-3 大阪大学大学院基礎工学研究科システム科学分野
makino@sys.es.osaka-u.ac.jp

あらまし　本論文では, データ発掘, 学習理論の分野で現れる有限ハイパーグラフの横断の二つ自然な拡張, 多重横断と部分横断について考える. 我々は, すべての多重横断, あるいは, すべての部分横断から成るハイパーグラフが, その双対 (横断) ハイパーグラフのサイズがそれ自身のサイズと入力ハイパーグラフのサイズの多項式で制限されるという意味で, 双対制限されることを示す. 我々の上界は, 極値的集合理論, 閾ブール理論の新しい不等式に基づいている. 我々は, 与えられたハイパーグラフのすべての多重横断, あるいは, すべての部分横断を列挙する問題が, ハイパーグラフの横断を列挙する問題, すなわち, ハイパーグラフの有名な双対問題に多項式時間還元可能であることも示す. 系として, ハイパーグラフのすべての多重横断, すべての部分横断, また, 線形不等式で表される単調システムのすべての極小な 0-1 解の列挙が逐次擬多項式時間で可能なことも示す.

和文キーワード: ハイパーグラフの横断, 単調論理関数の双対化, 逐次多項式時間アルゴリズム, データ発掘

Dual-Bounded Hypergraphs: Generating Partial and Multiple Transversals

Endre Boros[†], Vladimir Gurvich[‡], Leonid Khachiyan[§] and Kazuhisa Makino[¶]

[†]RUTCOR, Rutgers University, 640 Bartholomew Road, Piscataway NJ 08854.
boros@rutcor.rutgers.edu

[‡]Russian Academy of Sciences, Moscow, Russia. gurvich@rutcor.rutgers.edu

[§]Department of Computer Science, Rutgers University, New Brunswick, New Jersey 08903.
leonid@cs.rutgers.edu

[¶]Division of Systems Science, Graduate School of Engineering Science, Osaka University, Toyonaka, Osaka, 560-8531, Japan. makino@sys.es.osaka-u.ac.jp

abstract　We consider two natural generalizations of the notion of transversal to a finite hypergraph, arising in data-mining and machine learning, the so called *multiple* and *partial* transversals. We show that the hypergraphs of all multiple and all partial transversals are dual- bounded in the sense that in both cases, the size of the dual hypergraph is bounded by a polynomial in the cardinality and the length of description of the input hypergraph. Our bounds are based on new inequalities of extremal set theory and threshold Boolean logic, which may be of independent interest. We also show that the problems of generating all multiple and all partial transversals for a given hypergraph are polynomial-time reducible to the generation of all ordinary transversals for another hypergraph, i.e., to the well-known dualization problem for hypergraphs. As a corollary, we obtain incremental quasi-polynomial-time algorithms for both of the above problems, as well as for the generation of all the minimal Boolean solutions for an arbitrary monotone system of linear inequalities.

英文 **key words**: Transversals of hypergraphs, dualization of monotone Boolean functions, incremental polynomial time algorithms, data mining

1 Introduction

In this paper we consider some problems involving the generation of all subsets of a finite set satisfying certain conditions. The most well-known problem of this type, the generation of all minimal *transversals*, has applications in combinatorics, graph theory [15, 17], artificial intelligence [10], game theory, reliability theory, database theory [1] and learning theory [2].

Given a finite set V of $n = |V|$ points, and a hypergraph (set family) $\mathcal{A} \subseteq 2^V$, a subset $B \subseteq V$ is called a *transversal* of the family $\mathcal{A}$ if $A \cap B \neq \emptyset$ for all sets $A \in \mathcal{A}$; it is called a *minimal transversal* if no proper subset of B is a transversal of $\mathcal{A}$. The hypergraph $\mathcal{A}^d$ consisting of all minimal transversals of $\mathcal{A}$ is called the *dual* (or *transversal*) hypergraph of $\mathcal{A}$. It is easy to see that if $A \in \mathcal{A}$ is not minimal in $\mathcal{A}$, i.e. if $A' \in \mathcal{A}$ for some $A' \subset A$, then $(\mathcal{A} \setminus \{A\})^d = \mathcal{A}^d$. We can assume therefore that all sets in $\mathcal{A}$ are minimal, i.e. that the hypergraph $\mathcal{A}$ is Sperner. (The dual hypergraph $\mathcal{A}^d$ is Sperner by definition.) It is then easy to verify that $(\mathcal{A}^d)^d = \mathcal{A}$ and $\bigcup_{A \in \mathcal{A}} A = \bigcup_{B \in \mathcal{A}^d} B$.

For a subset $X \subseteq V$ let $X^c = V \setminus X$ denote the complement of X, and let $\mathcal{A}^c = \{A^c | A \in \mathcal{A}\}$ be the *complementary* hypergraph of $\mathcal{A}$. Then e.g. $\mathcal{A}^{dc}$ consists of all maximal subsets containing no hyperedge of $\mathcal{A}$, while the hypergraph $\mathcal{A}^{cd}$ consists of all minimal subsets of V which are not contained in any hyperedge of $\mathcal{A}$.

1.1 Dualization

Given a Sperner hypergraph $\mathcal{A}$, a frequently arising task is the generation of the transversal hypergraph $\mathcal{A}^d$. This problem, known as *dualization*, can be stated as follows:

Given a complete list of all hyperedges of $\mathcal{A}$ and a set of minimal transversals $\mathcal{B} \subseteq \mathcal{A}^d$, either prove that $\mathcal{B} = \mathcal{A}^d$, or find a new transversal $X \in \mathcal{A}^d \setminus \mathcal{B}$.

Clearly, we can generate all of the hyperedges of $\mathcal{A}^d$ by initializing $\mathcal{B} = \emptyset$ and recursively solving the above problem $|\mathcal{A}^d|$ times. Note also that in general, $|\mathcal{A}^d|$ can be exponentially large both in $|\mathcal{A}|$ and $|V|$. For this reason, the complexity of generating $\mathcal{A}^d$ is customarily measured in the input and output sizes. In particular, we say that $\mathcal{A}^d$ can be generated in *incremental polynomial time* if the dualization problem can be solved in time polynomial in $|V|$, $|\mathcal{A}|$ and $|\mathcal{B}|$.

The dualization problem can be efficiently solved for many classes of hypergraphs. For example, if the sizes of all the hyperedges of $\mathcal{A}$ are limited by a constant r, then dualization can be executed in incremental polynomial time, (see e.g. [6, 10]). In the quadratic case, i.e. when $r = 2$, there are even more efficient dualization algorithms that run with *polynomial delay*, i.e. in $poly(|V|, |\mathcal{A}|)$ time, where $\mathcal{B}$ is systematically enlarged from $\mathcal{B} = \emptyset$ during the generation process of $\mathcal{A}^d$ (see e.g. [15, 17]). Efficient algorithms exist also for the dualization of 2-monotonic, threshold, matroid, read-bounded, acyclic and some other classes of hypergraphs (see e.g. [4, 8, 9, 19, 20, 24]).

Even though no incremental polynomial time algorithm for the dualization of arbitrary hypergraphs is known, an incremental *quasi-polynomial time* one exists (see [11]). This algorithm solves the dualization problem in $O(nm) + m^{o(\log m)}$ time, where $n = |V|$ and $m = |\mathcal{A}| + |\mathcal{B}|$ (see also [13] for more detail).

In this paper, we consider two natural generalizations of minimal transversals, so called *multiple transversals*, and *partial transversals*. See Section 3 for related hypergraphs in the data-mining and machine learning literature.

1.2 Multiple transversals

Given a hypergraph $\mathcal{A} \subseteq 2^V$ and a non-negative weight b_A associated with every hyperedge $A \in \mathcal{A}$, a subset X is called a *multiple transversal* (or *b-transversal*), if $|X \cap A| \geq b_A$ holds for all $A \in \mathcal{A}$. The family of all minimal b-transversals then can also be viewed as the family of support sets of minimal feasible binary solutions to the system of inequalities

$$Ax \geq b, \tag{1}$$

where the rows of $A = A_{\mathcal{A}}$ are exactly the characteristic vectors of the hyperedges $A \in \mathcal{A}$, and the corresponding component of b is equal to b_A. Clearly, $b = (1, 1, ..., 1)$ corresponds to the case of (ordinary) transversals, in which case (1) is also known as a *set covering* problem.

Generalizing further and giving up the binary nature of A as well, we shall consider the family $\mathcal{F} = \mathcal{F}_{A,b}$ of (support sets of) all minimal feasible binary vectors to (1) for a given $m \times n$-matrix A and a given m-vector b. We assume that (1) is a *monotone* system of inequalities: if $x \in \{0,1\}^n$ satisfies (1) then any vector $y \in \{0,1\}^n$ such that $y \geq x$ is also feasible. For instance, (1) is monotone if A is non-negative. Note that for a monotone system (1) the dual hypergraph $\mathcal{F}^d = \mathcal{F}^d_{A,b}$ is (the complementarity hypergraph of) the collection of (supports of) all maximal infeasible vectors for (1). Note also that we assume that the hypergraph $\mathcal{F}_{A,b}$ is represented by the system (1) and not given explicitly, i.e., by a list of all its hyperedges. In particular, this means that the generation of $\mathcal{F}_{A,b}$ and its dual $\mathcal{F}^d_{A,b}$ are both non-trivial.

Let us consider in general a Sperner hypergraph $\mathcal{F} \subseteq 2^V$ on a finite set V represented in some implicit way, and let GEN($\mathcal{F}$) denote the problem of generating all the hyperedges of $\mathcal{F}$:

Given $\mathcal{F}$ and a list of hyperedges $\mathcal{H} \subseteq \mathcal{F}$, either prove that $\mathcal{H} = \mathcal{F}$ or find a new hyperedge in $\mathcal{F} \setminus \mathcal{H}$.

It is known that problem GEN($\mathcal{F}^d_{A,b}$) is NP-hard even for binary matrices A (see [18]). In contrast to that, we show that the tasks of generating multiple and ordinary transversals are polynomially related.

Theorem 1 *Problem GEN($\mathcal{F}_{A,b}$) is polytime reducible to dualization.*

In particular, for any monotone system of linear inequalities (1), all minimal binary solutions of (1) can be generated in quasi-polynomial incremental time.

Remark 1 *Even though generating all maximal infeasible binary points for (1) is hard, there is a polynomial randomized scheme for nearly uniform sampling from the set of all binary infeasible points for (1). Such a scheme can be obtained by combining the algorithm [16] for approximating the size of set-unions with the rapidly mixing random walk [23] on the binary cube truncated by a single linear inequality. On the other hand, a similar randomized scheme for nearly uniform sampling from within the set of all binary (or all minimal binary) solutions to a given monotone system (1) would imply that any NP-complete problem can be solved in polynomial time by a randomized algorithm with arbitrarily small one-sided failure probability. By using the amplification technique of [14], this can be shown already for systems (1) with two non-zero coefficients per inequality, see e.g. [12] for more detail.*

1.3 Partial transversals

Given a hypergraph $\mathcal{A} \subseteq 2^V$ and a non-negative threshold $k < |\mathcal{A}|$, a subset $X \subseteq V$ is said to be a *partial transversal*, or more precisely, a *k-transversal*, to the family $\mathcal{A}$ if it intersects all but at most k of the subsets of $\mathcal{A}$, i.e. if $|\{A \in \mathcal{A} | A \cap X = \emptyset\}| \leq k$.

Denote by $\mathcal{A}^{d_k}$ the family of all minimal k-transversals of $\mathcal{A}$. Clearly, 0-transversals are exactly the standard transversals, defined above, i.e. $\mathcal{A}^{d_0} = \mathcal{A}^d$. In what follows we assume that the hypergraph $\mathcal{A}^{d_k}$ is represented by a list of all the edges of $\mathcal{A}$ along with the value of $k \in \{0, 1, \cdots, |\mathcal{A}| - 1\}$.

Define a *k-union* from $\mathcal{A}$ as the union of some k subsets of $\mathcal{A}$, and let $\mathcal{A}^{u_k}$ denote the family of all minimal k-unions of $\mathcal{A}$. In other words, $\mathcal{A}^{u_k}$ is the family of all the minimal subsets of A which contain at least k hyperedges of $\mathcal{A}$. By the above definitions, k-union and k-transversal families both are Sperner (even if the input hypergraph $\mathcal{A}$ is not). It is also easy to see that the families of all minimal k-transversals and $(k+1)$-unions are in fact dual, i.e. ,

$$\mathcal{A}^{d_k} = (\mathcal{A}^{u_{k+1}})^d; \quad k = 0, 1, ..., |\mathcal{A}| - 1.$$

The tasks of generating partial and ordinary transversals also turn out to be polynomially equivalent.

Theorem 2 *Problem GEN($\mathcal{A}^{d_k}$) is polytime reducible to dualization.*

It should be mentioned that the dual problem GEN($\mathcal{A}^{u_{k+1}}$) is NP-hard (see [21]).

1.4 Bounding dual hypergraphs

Our proofs of Theorems 1 and 2 make use of the fact that the Sperner hypergraphs $\mathcal{F}_{A,b}$ and $\mathcal{A}^{d_k}$ are *dual-bounded* in the sense that in both cases, the size of the dual hypergraph can be bounded by a polynomial in the size and the length of description of the primal hypergraph.

Theorem 3 *For any monotone system (1) of m linear inequalities in n binary variables,*

$$|\mathcal{F}^d_{A,b}| \leq mn|\mathcal{F}_{A,b}|.$$

Moreover,

$$|\mathcal{H}^d \cap \mathcal{F}^d_{A,b}| \leq mn|\mathcal{H}| \quad \textit{for any} \quad \mathcal{H} \subseteq \mathcal{F}_{A,b}. \tag{2}$$

Theorem 4 *For any hypergraph $\mathcal{A} \subseteq 2^V$ of $m = |\mathcal{A}|$ hyperedges and any threshold $k = 0, \ldots, m-1$, we have*

$$|\mathcal{A}^{u_{k+1}}| \leq 2|\mathcal{A}^{d_k}|^2 + (m-k-2)|\mathcal{A}^{d_k}|.$$

Moreover, for any hypergraph $\mathcal{H} \subseteq \mathcal{A}^{d_k}$,

$$|\mathcal{H}^d \cap \mathcal{A}^{u_{k+1}}| \leq 2|\mathcal{H}|^2 + (m-k-2)|\mathcal{H}|. \tag{3}$$

We derive Theorem 3 from the following lemma.

Lemma 1 *Let $h : \{0,1\}^n \to \{0,1\}$ be a monotone Boolean function such that*

$$h(x) = 1 \;\Rightarrow\; wx \stackrel{\text{def}}{=} \sum_{i=1}^{n} w_i x_i \geq t,$$

where $w = (w_1, \ldots, w_n)$ is a given weight vector and t is a threshold. If $h \not\equiv 0$, then

$$|\max F(h) \cap \{x \mid wx < t\}| \leq \sum_{x \in \min T(h)} ex,$$

where $\max F(h) \subset \{0,1\}^n$ is the set of all maximal false points of h, $\min T(h) \subseteq \{0,1\}^n$ is the set of all minimal true points of h, and e is the vector of all ones. In particular,

$$|\max F(h) \cap \{x \mid wx < t\}| \leq n|\min T(h)|.$$

If the function h is threshold ($h(x) = 1 \Leftrightarrow wx \geq t$), then $|\max F(h)| \leq n|\min T(h)|$ and, by symmetry, $|\min T(h)| \leq n|\max F(h)|$, well-known inequalities (see [4, 9, 24]). Lemma 1 thus extends the above threshold inequalities to arbitrary monotone functions h.

As we shall see, Theorem 4 can be derived from the following combinatorial inequality.

Lemma 2 *Let $\mathcal{A} \subseteq 2^V$ be a hypergraph on $|V| = n$ vertices with $|\mathcal{A}| \geq 2$ hyperedges such that*

$$\begin{array}{ll} |A| \geq k+1 & \textit{for all } A \in \mathcal{A}, \textit{ and} \\ |B| \leq k & \textit{for all } B \in \mathcal{A}^{\cap}, \end{array} \tag{T}$$

where k is a given threshold and $\mathcal{A}^{\cap}$ is the family of all the maximal subsets of V which can be obtained as the intersection of two distinct hyperedges of $\mathcal{A}$. Then

$$|\mathcal{A}| \leq (n-k)|\mathcal{A}^{\cap}|. \tag{4}$$

Note that $\mathcal{A}^{\cap}$ is a Sperner family by its definition, and that condition (T) implies the same for $\mathcal{A}$. Note also that the thresholdness condition (T) is essential for the validity of the lemma – without (T) the size of $\mathcal{A}$ can be exponentially larger than that of $\mathcal{A}^{\cap}$. There are examples of Sperner hypergraphs $\mathcal{A}$ for which $|\mathcal{A}^{\cap}| = n/5$ and $|\mathcal{A}| = 3^{n/5} + 2n/5$ or $|\mathcal{A}^{\cap}| = (n-2)^2/9$ and $|\mathcal{A}| = 3^{(n-2)/3} + 2(n-2)/3$. (Several other inequalities on hypergraphs with restricted intersections can be found in Chapter 4 of [3].)

The remainder of the paper is organized as follows. In Section 2 we discuss the complexity of jointly generating a pair of dual hypergraphs defined via a superset oracle. For a polynomial-time superset oracle the above problem reduces to dualization. This reduction along with the bounds stated in Theorems 3 and 4 yield Theorems 1 and 2. Finally, Section 3 discusses some of the related set families and results, and Section 4 contains our concluding remarks.

2 Joint and separate generation of dual hypergraphs

2.1 Superset oracles

Let $\mathcal{G} \subseteq 2^V$ be a Sperner hypergraph on n vertices. In many applications, $\mathcal{G}$ is represented by a *superset oracle* and not given explicitly. Such an oracle can be viewed as an algorithm which, given an input description $\mathfrak{D}$ of $\mathcal{G}$ and a vertex set $X \subseteq V$, can decide whether or not X contains a hyperedge of $\mathcal{G}$. Equivalently, the oracle can be used to evaluate the monotone Boolean function $f_{\mathcal{G}}(x) \stackrel{\text{def}}{=} \vee_{G \in \mathcal{G}} \wedge_{i \in G} x_i$ at any point $x \in \{0,1\}^n$. Note that the *dual function* $\bar{f}_{\mathcal{G}}(\bar{x}) = f_{\mathcal{G}^d}(x) = \vee_{G \in \mathcal{G}^d} \wedge_{i \in G} x_i$ can also be evaluated via a single call to the oracle. We assume that the length $|\mathfrak{D}|$ of the input description of $\mathcal{G}$ is at least n and denote by $T_s = T_s(|\mathfrak{D}|)$ the worst-case running time of the oracle on any superset query "Does X contains a hyperedge of $\mathcal{G}$?". In particular, $\mathfrak{D}$ is polynomial-time if $T_s \leq poly(|\mathfrak{D}|)$. In what follows, we do not distinguish the superset oracle and the input description $\mathfrak{D}$ of $\mathcal{G}$. As mentioned above, $\mathfrak{D}$ also specifies (a superset oracle for) the dual hypergraph $\mathcal{G}^d$. We list below several simple examples.

1) Multiple transversals. Let (1) be a monotone system of linear inequalities, and let $\mathcal{G} = \mathcal{F}_{A,b}$ be the hypergraph introduced in Section 1.2. Then the input description $\mathfrak{D}$ is (A, b). Clearly, for any input set $X \subseteq V$, we can decide whether X contains a hyperedge of $\mathcal{F}_{A,b}$ by checking the feasibility of (the characteristic vector of) X for (1).

2) Partial transversals. Let $\mathcal{G} = \mathcal{A}^{d_k}$ be the hypergraph of the minimal k-transversals of a family $\mathcal{A}$ (see Section 1.2). Then $\mathcal{G}$ is given by the threshold value k and a complete list of all hyperedges of $\mathcal{A}$, i.e., $\mathfrak{D} \sim (k, \mathcal{A})$. For a subset $X \subseteq V$, determining whether X contains a hyperedge in $\mathcal{A}^{d_k}$ is equivalent to checking if X is intersecting at least $|\mathcal{A}| - k$ hyperedges of $\mathcal{A}$.

3) Monotone Boolean formulae. Let f be a $(\vee, \wedge)$-formula with n variables and let $\mathcal{G} = \mathcal{A}_f$ be the supporting sets of all the minimal true vectors for f. Then $\mathfrak{D} \sim f$ and the superset oracle checks if (the characteristic vector of) $X \subseteq V$ satisfies f. The dual hypergraph $\mathcal{G}^d$ is the set of all the (complements to the support sets of) maximal false vectors of f.

4) Relay circuits. Consider a digraph Γ with a source s and a sink t, each arc of which is assigned a *relay* $r \in V$ (two or more distinct edges may be assigned identical relays). Let $\mathcal{G}$ be the set of *relay s-t paths*, i.e., minimal subsets of relays that connect s and t. Then $\mathfrak{D} \sim \Gamma$, and for a given relay set $X \subseteq V$, the superset oracle can use breadth-first search to check the reachability of t from s via a path consisting of relays in X. Note that the dual hypergraph $\mathcal{G}^d$ is the set of all relay *s-t cuts*, i.e., minimal subsets of relays that disconnect s and t.

5) Helly's systems of polyhedra. Consider a family of n convex polyhedra $P_i \subseteq \mathbb{R}^r$, $i \in V$, and let $\mathcal{G}$ denote the minimal subfamilies with no point in common. Then $\mathcal{G}^{dc}$ is the family of all maximal subfamilies with a nonempty intersection. (In particular, if $P_1, \ldots, P_n$ are the facets of a convex polytope Q, then $\mathcal{G}^{dc}$ corresponds to the set of vertices of Q.) We have $\mathfrak{D} \sim (P_1, \ldots, P_n)$ and, given subsets of polytopes $X \subseteq V$, the superset oracle can use linear programming to check whether $\cap_{i \in X} P_i \neq \emptyset$.

2.2 Joint generation of dual pairs of hypergraphs

In all of the above examples, we have pairs of dual Sperner hypergraphs given by polynomial-time superset oracles. Let $\mathcal{G}, \mathcal{G}^d \subseteq 2^V$ be a pair of dual Sperner hypergraphs given by a superset oracle $\mathfrak{D}$. Consider the problem GEN$(\mathcal{G}, \mathcal{G}^d)$ of *generating jointly* all the hyperedges of $\mathcal{G}$ and $\mathcal{G}^d$:

Given two explicitly listed set families $\mathcal{A} \subseteq \mathcal{G}$ and $\mathcal{B} \subseteq \mathcal{G}^d$, either prove that these families are complete, $(\mathcal{A}, \mathcal{B}) = (\mathcal{G}, \mathcal{G}^d)$, or find a new set in $(\mathcal{G} \setminus \mathcal{A}) \cup (\mathcal{G}^d \setminus \mathcal{B})$.

For the special case when $\mathcal{A} = \mathcal{G}$ and $\mathfrak{D}$ is a list of all the sets in $\mathcal{G}$, we obtain the dualization problem as stated in Section 1.1. In fact, as observed in [5, 12], for any polynomial-time superset oracle $\mathfrak{D}$ problem GEN$(\mathcal{G}, \mathcal{G}^d)$ can be reduced in polynomial time to dualization. This can be done via the following Algorithm $\mathcal{J}$:

Step 1. Check whether each element of $\mathcal{B}$ is a minimal transversal to $\mathcal{A}$, i.e., $\mathcal{B} \subseteq \mathcal{A}^d$. (Recall that $\mathcal{A}$ and $\mathcal{B}$ are given explicitly.) Note that each set $X \in \mathcal{B}$ is a transversal to $\mathcal{A}$ because $\mathcal{A} \subseteq \mathcal{G}$ and $\mathcal{B} \subseteq \mathcal{G}^d$. If some transversal $X \in \mathcal{B}$ is not *minimal* for $\mathcal{A}$ then we can easily find a proper subset Y of X such that Y is also a transversal to $\mathcal{A}$. Since Y is a proper subset of X, and X is a minimal transversal to $\mathcal{G}$, Y must miss some hyperedges of $\mathcal{G}$. Hence Y^c, the complement of Y, contains a hyperedge of $\mathcal{G}$. By querying the superset oracle $\mathfrak{O}$ at most $|Y^c|$ times we can find such a hyperedge $Z \in \mathcal{G}$. Note that $Z \cap Y = \emptyset$ whereas $A \cap Y \neq \emptyset$ for all hyperedges $A \in \mathcal{A}$. This means that Z is a new hyperedge of $\mathcal{G}$. Thus, if the inclusion $\mathcal{B} \subseteq \mathcal{A}^d$ is not satisfied, we can obtain an element in $\mathcal{G} \setminus \mathcal{A}$ and halt.

Step 2 is similar to *Step 1.* Check whether $\mathcal{A} \subseteq \mathcal{B}^d$. If $\mathcal{A}$ contains a non-minimal transversal to $\mathcal{B}$, find a new element in $\mathcal{G}^d \setminus \mathcal{B}$ and halt.

Step 3. Suppose that $\mathcal{B} \subseteq \mathcal{A}^d$ and $\mathcal{A} \subseteq \mathcal{B}^d$. Then $\mathcal{B} = \mathcal{A}^d \Rightarrow (\mathcal{A}, \mathcal{B}) = (\mathcal{G}, \mathcal{G}^d)$. (This is because any hyperedge $X \in \mathcal{G} \setminus \mathcal{A}$ would be a transversal for $\mathcal{B}$, which would then imply that X contains some hyperedge of $\mathcal{A} = \mathcal{B}^d$, contradiction. By symmetry, the duality of $\mathcal{A}$ and $\mathcal{B}$ also implies the emptiness of $\mathcal{G}^d \setminus \mathcal{B}$.) Hence $(\mathcal{A}, \mathcal{B}) = (\mathcal{G}, \mathcal{G}^d) \Leftrightarrow \mathcal{B} = \mathcal{A}^d$. The condition $\mathcal{B} = \mathcal{A}^d$ can be checked by solving the dualization problem for $\mathcal{A}$ and $\mathcal{B}$. If $\mathcal{B} \neq \mathcal{A}^d$, we obtain a new minimal transversal $X \in \mathcal{A}^d \setminus \mathcal{B}$, see Section 1.1. By definition, X contains no hyperedge in $\mathcal{B}$ and X^c contains no hyperedge in $\mathcal{A}$. Due to the duality of $\mathcal{G}$ and $\mathcal{G}^d$, either (i) X^c contains a hyperedge of $\mathcal{G}$, or (ii) X contains a hyperedge of $\mathcal{G}^d$, but not both. We can call the superset oracle to decide which of the two cases holds. In case (i) we obtain a new hyperedge in $\mathcal{G} \setminus \mathcal{A}$ by querying the superset oracle at most $|X^c|$ times. Similarly, in case (ii) we get a new hyperedge in $\mathcal{G}^d \setminus \mathcal{B}$ in at most $|X|$ calls to the oracle. □

Algorithm $\mathcal{J}$ readily implies the following result.

Proposition 1 ([5, 12]) *Problem GEN($\mathcal{G}$, $\mathcal{G}^d$) can be solved in $n\,(poly(|\mathcal{A}|, |\mathcal{B}|) + T_s(|\mathfrak{O}|)) + T_{dual}$ time, where T_{dual} denotes the time required for solving the dualization problem with $\mathcal{A}$ and $\mathcal{B}$.*

In particular, for any (quasi-)polynomial-time oracle $\mathfrak{O}$, problem GEN$(\mathcal{G}, \mathcal{G}^d)$ can be solved in quasi-polynomial time. Thus, for each of the 5 examples above we can jointly generate all the hyperedges of $(\mathcal{G}, \mathcal{G}^d)$ in incremental quasi-polynomial time. Note, however, that separately generating all the hyperedges of $\mathcal{G}$ or all the hyperedges of $\mathcal{G}^d$ may be substantially harder. For instance, as shown in [12], both problems GEN$(\mathcal{G})$ and GEN$(\mathcal{G}^d)$ are NP-hard for examples 3-5 above. In fact, in example 3 these problems are NP-hard already for $\vee, \wedge$-formulae of depth 3; if the depth is 2 then the formula is either CNF or DNF and we get exactly dualization.

2.3 Dual-bounded hypergraphs

Algorithm $\mathcal{J}$ may not be efficient for solving either of the problems GEN$(\mathcal{G})$ or GEN$(\mathcal{G}^d)$ separately for the simple reason that we do not control which of the families $\mathcal{G} \setminus \mathcal{A}$ and $\mathcal{G}^d \setminus \mathcal{B}$ contains each new hyperedge produced by the algorithm. Suppose, we want to generate $\mathcal{G}$, and the family $\mathcal{G}^d$ is exponentially larger than $\mathcal{G}$. Then, if we are unlucky, we can get hyperedges of $\mathcal{G}$ with exponential delay, while getting large subfamilies of $\mathcal{G}^d$ (which are not needed at all) in between.

Such a problem will not arise and simultaneous generation of $(\mathcal{G}, \mathcal{G}^d)$ can be used to produce $\mathcal{G}$ efficiently, in some sense, if the size of $\mathcal{G}^d$ is polynomially limited in the size of $\mathcal{G}$ and in the input size $|\mathfrak{O}|$, i.e. when there exists a polynomial p such that

$$|\mathcal{G}^d| \leq p(|V|, |\mathfrak{O}|, |\mathcal{G}|). \tag{5}$$

We call such Sperner hypergraphs $\mathcal{G}$ *dual-bounded.*

If $\mathcal{G}$ is dual-bounded, we can generate both $\mathcal{G}$ and $\mathcal{G}^d$ in $|\mathcal{G}^d| + |\mathcal{G}| \leq poly(|V|, |\mathfrak{O}|, |\mathcal{G}|)$ rounds of Algorithm $\mathcal{J}$, and hence obtain all the hyperedges of $\mathcal{G}$ in *total quasi-polynomial* time.

This approach, however, may still be inefficient incrementally, i.e., for obtaining a single hyperedge of $\mathcal{G}$ as required in problem GEN$(\mathcal{G})$. It is easy to see that the decision problem: "Given a family $\mathcal{A} \subseteq \mathcal{G}$, determine whether $\mathcal{A} = \mathcal{G}$?" is polynomially reducible to dualization for any dual-bounded hypergraphs represented by a polynomial-time superset oracle. If $\mathcal{A}$ is much smaller than $\mathcal{G}$, however, getting a new hyperedge in $\mathcal{G} \setminus \mathcal{A}$ may require exponentially many (in $|\mathcal{A}|$) rounds of $\mathcal{J}$.

2.4 Uniformly dual-bounded hypergraphs

Let us call a Sperner hypergraph $\mathcal{G}$ *uniformly dual-bounded* if

$$|\mathcal{H}^d \cap \mathcal{G}^d| \leq p(|V|, |\mathfrak{D}|, |\mathcal{H}|) \text{ for any } \mathcal{H} \subseteq \mathcal{G}.$$

Note that for $\mathcal{H} = \mathcal{G}$ the above condition gives (5).

Proposition 2 *Problem GEN($\mathcal{G}$) is poly-time reducible to dualization for any uniformly dual-bounded hypergraph $\mathcal{G}$ defined by a polynomial-time superset oracle.*

Theorems 3 and 4 state that the hypergraphs $\mathcal{F}_{A,b}$ and $\mathcal{A}^{d_k}$ are both uniformly dual-bounded. In view of Proposition 2, this means that Theorems 3 and 4 imply Theorems 1 and 2, respectively.

3 Related set-families

The notion of frequent sets appears in the data-mining literature, see [1, 22], and can be related naturally to the families considered above. More precisely, following a definition of [25], given a $(0,1)$-matrix and a threshold k, a subset of the columns is called *frequent* if there are at least k rows having a 1 entry in each of the corresponding positions. The problems of generating all *maximal frequent* sets and their duals, the so called *minimal infrequent* sets (for a given binary matrix) were proposed, and the complexity of the corresponding decision problems were asked in [25]. Results of [21] imply that it is NP-hard to determine whether a family of maximal frequent sets is incomplete, while our results prove that generating all minimal infrequent sets polynomially reduces to dualization.

Since the family $\mathcal{A}^{d_k}$ consists of all the minimal k-transversals to $\mathcal{A}$, i.e. subsets of V which are disjoint from at most k hyperedges of $\mathcal{A}$, the family $\mathcal{A}^{cd_k}$ consists of all the minimal subsets of V which are contained in at most k hyperedges of $\mathcal{A}$. It is easy to recognize that these are the *minimal infrequent sets* in a matrix, the rows of which are the characteristic vectors of the hyperedges of $\mathcal{A}$. Furthermore, the family $\mathcal{A}^{d_k c}$ consists of all the maximal subsets of V, which are supersets of at most k hyperedges of $\mathcal{A}$.

Due to our results above, all these families can be generated e.g. in incremental quasi-polynomial time.

In the special case, if $\mathcal{A}$ is a quadratic set-family, i.e. if all hyperedges of $\mathcal{A}$ are of size 2, the family $\mathcal{A}$ can also be interpreted as the edge set of a graph G on vertex set V. Then, $\mathcal{A}^{d_k c}$ is also known as the family of the so called *fairly independent sets* of the graph G, i.e. all the vertex subsets which induce at most k edges (see [25].)

As it was defined above, the family $\mathcal{A}^{u_k}$ consists of all the minimal k-unions of $\mathcal{A}$, i.e. all minimal subsets of V which contain at least k hyperedges of $\mathcal{A}$, and hence the family $\mathcal{A}^{cu_k}$ consists of all the minimal subsets which contain at least k hyperedges of $\mathcal{A}^c$. Thus, the family $\mathcal{A}^{cu_k c}$ consists of all the maximal k-intersections, i.e. maximal subsets of V which are subsets of at least k hyperedges of $\mathcal{A}$. These sets can be recognized as the *maximal frequent sets* in a matrix, the rows of which are the characteristic vectors of the hyperedges of $\mathcal{A}$. Finally, the family $\mathcal{A}^{u_k c}$ consists of all the maximal subsets of V which are disjoint from at least k hyperedges of $\mathcal{A}$.

As it follows from the mentioned results (see e.g. [21]), generating all hyperedges for each of these families is NP-hard , unless k (or $|\mathcal{A}|-k$) is bounded by a constant.

4 General closing remarks

In this paper we considered the problems of generating all partial and all multiple transversals. Both problems are formally more general than dualization, but in fact both are polynomially equivalent to it because the corresponding pairs of hypergraphs are uniformly dual-bounded.

It might be tempting to look for a common generalization of these notions, and of these results. However, the attempt to combine partial and multiple transversals fails. For instance, generating all the minimal partial binary solutions to a system of inequalities $Ax \geq b$ is NP-hard, even if A is binary and $b = (2, 2, ..., 2)$. To show this we can use arguments analogous to those of [18, 21]. Consider the well-known NP-hard problem of determining whether a given graph $G = (V, E)$ contains an independent vertex set of size t, where $t \geq 2$ is a given threshold. Introduce $|V| + 1$ binary variables x_0 and x_v, $v \in V$, and write t inequalities $x_u + x_v \geq 2$ for each edge $e = (u, v) \in E$, followed by the inequalities $x_0 + x_v \geq 2$, $v \in V$. It is easily seen that the characteristic vector of any edge $e = (u, v)$ is a minimal binary solution satisfying at least t inequalities of the resulting system. Deciding whether there are other minimal binary solutions satisfying $\geq t$ inequalities of the system is equivalent to determining whether G has an independent set of size t.

For further generalizations of Lemma 2 we refer the interested reader to the technical report [7].

Acknowledgments. The authors are thankful to József Beck for helpful discussions.

References

[1] R. Agrawal, H. Mannila, R. Srikant, H. Toivonen and A. I. Verkamo, Fast discovery of association rules, In U. M. Fayyad, G. Piatetsky-Shapiro, P. Smyth and R. Uthurusamy eds., *Advances in Knowledge Discovery and Data Mining*, 307-328, AAAI Press, Menlo Park, California, 1996.

[2] M. Anthony and N. Biggs, *Computational Learning Theory*, Cambridge University Press, 1992.

[3] L. Babai and P. Frankl, Linear Algebra Methods in Combinatorics, Manuscript, The University of Chicago, September 1992.

[4] P. Bertolazzi and A. Sassano, An $O(mn)$ time algorithm for regular set-covering problems, *Theoretical Computer Science* 54 (1987) 237-247.

[5] J. C. Bioch and T. Ibaraki, Complexity of identification and dualization of positive Boolean functions, *Information and Computation* 123 (1995) 50-63.

[6] E. Boros, V. Gurvich, and P.L. Hammer, Dual subimplicants of positive Boolean functions, *Optimization Methods and Software*, 10 (1998) 147-156. (RUTCOR Research Report 11-93.)

[7] E. Boros, V. Gurvich, L. Khachiyan and K.Makino, Dual-Bounded Hypergraphs: Generating Partial and Multiple Transversals, DIMACS Technical Report 99-62, Rutgers University, 1999.

[8] E. Boros, P. L. Hammer, T. Ibaraki and K. Kawakami, Polynomial time recognition of 2-monotonic positive Boolean functions given by an oracle, *SIAM Journal on Computing* 26 (1997) 93-109.

[9] Y. Crama, Dualization of regular Boolean functions, *Discrete Applied Mathematics*, 16 (1987) 79-85.

[10] T. Eiter and G. Gottlob, Identifying the minimal transversals of a hypergraph and related problems, *SIAM Journal on Computing*, 24 (1995) 1278-1304.

[11] M. L. Fredman and L. Khachiyan, On the complexity of dualization of monotone disjunctive normal forms. *J. of Algorithms*, 21 (1996) 618-628.

[12] V. Gurvich and L. Khachiyan, On generating the irredundant conjunctive and disjunctive normal forms of monotone Boolean functions, *Discrete Applied Mathematics*, 97 (1999) 363-373.

[13] V. Gurvich and L. Khachiyan On the frequency of the most frequently occurring variable in dual DNFs, *Discrete Mathematics* 169 (1997) 245-248.

[14] M. R. Jerrum, L. G. Valiant and V. V. Vazirani, Random generation of combinatorial structures from a uniform distribution *Theoretical Computer Science* 43 (1986) 169-188.

[15] D. S. Johnson, M. Yannakakis and C. H. Papadimitriou, On generating all maximal independent sets, *Information Processing Letters*, 27 (1988) 119-123.

[16] R. Karp and M, Luby, Monte-Carlo algorithms for enumeration and reliability problems, in *Proc. 24th IEEE Symp. on Foundations of Computer Science* (1983) 56-64.

[17] E. Lawler, J. K. Lenstra and A. H. G. Rinnooy Kan, Generating all maximal independent sets: NP-hardness and polynomial-time algorithms, *SIAM Journal on Computing*, 9 (1980) 558-565.

[18] K. Makino and T. Ibaraki, Interor and exterior functions of Boolean functions, *Discrete Applied Mathematics*, 69 (1996) 209-231.

[19] K. Makino and T. Ibaraki, The maximum latency and identification of positive Boolean functions, *SIAM Journal on Computing*, 26 (1997) 1363-1383.

[20] K. Makino and T. Ibaraki, A fast and simple algorithm for identifying 2-monotonic positive Boolean functions, *Journal of Algorithms*, 26 (1998) 291-305.

[21] K. Makino and T. Ibaraki, Inner-core and outer-core functions of partially defined Boolean functions, *Discrete Applied Mathematics*, 96-97 (1999), 307-326.

[22] H. Mannila and K. J. Räihä, Design by example: An application of Armstrong relations, *Journal of Computer and System Science* 22 (1986) 126-141.

[23] B. Morris and A. Sinclair, Random walks on truncated cubes and sampling 0-1 knapsack problem, in *Proc. 40th IEEE Symp. on Foundations of Computer Science* (1999) 230-240.

[24] U. N. Peled and B. Simeone, An $O(nm)$-time algorithm for computing the dual of a regular Boolean function, *Discrete Applied Mathematics* 49 (1994) 309-323.

[25] R. H. Sloan, K. Takata, G. Turan, On frequent sets of Boolean matrices, *Annals of Mathematics and Artificial Intelligence* 24 (1998) 1-4.