Output-Sensitive Algorithms for Enumerating Minimal Transversals for Some Geometric Hypergraphs

Khaled Elbassioni[1], Kazuhisa Makino[2], Imran Rauf[1]

[1] Max-Planck-Institut für Informatik, Saarbrücken, Germany
{elbassio,irauf}@mpi-inf.mpg.de
[2] Graduate School of Information Science and Technology, University of Tokyo, Tokyo, Japan
makino@mist.i.u-tokyo.ac.jp

Abstract. We give a general framework for the problem of finding all minimal hitting sets of a family of objects in $\mathbb{R}^d$ by another. We apply this framework to the following problems: (i) hitting hyper-rectangles by points in $\mathbb{R}^d$; (ii) stabbing connected objects by axis-parallel hyperplanes in $\mathbb{R}^d$; and (iii) hitting half-planes by points. For both the covering and hitting set versions, we obtain incremental polynomial-time algorithms, provided that the dimension d is fixed.

1 Introduction

Let $\mathcal{V}$ and $\mathcal{F}$ be two finite sets of geometric objects in $\mathbb{R}^d$. A subset of objects $\mathcal{X} \subseteq \mathcal{V}$ is said to be a *hitting set* (or *transversal* or *cover*) for $\mathcal{F}$ if for every $O \in \mathcal{F}$, there exists an $O' \in \mathcal{X}$ such that $O \cap O' \neq \emptyset$. A hitting set is *minimal* if none of its proper subsets is also a hitting set.

In this paper, we are interested in finding all minimal hitting sets of one family of objects by another. For such generation problems, we measure the time complexity in terms of both input and output length. An algorithm is said to run in *incremental polynomial-time*, if the time required to find k minimal transversals is polynomial in $|\mathcal{V}|$, $|\mathcal{F}|$, and k.

When $\mathcal{V}$ is a finite set of points and each object in $\mathcal{F}$ is an arbitrary finite subset of $\mathcal{V}$, we obtain the well-known *hypergraph transversal* or *dualization* problem [2], which calls for finding all minimal hitting sets for a given hypergraph $\mathcal{G} \subseteq 2^V$, defined on a finite set of vertices V. Denote by $\mathrm{Tr}(\mathcal{G})$ the set of all minimal hitting sets of $\mathcal{G}$, also known as the *transversal hypergraph* of $\mathcal{G}$. The problem of finding $\mathrm{Tr}(\mathcal{G})$ has received considerable attention in the literature (see, e.g., [3, 12, 13, 19, 29, 31]), since it is known to be polynomially or quasi-polynomially equivalent with many problems in various areas, such as artificial intelligence (e.g., [12, 24]), database theory (e.g., [30]), distributed systems (e.g., [23]), machine learning and data mining (e.g., [1, 7, 20]), mathematical programming (e.g., [5, 25]), matroid theory (e.g., [26]), and reliability theory (e.g., [9]).

The currently fastest known algorithm [17] for solving the hypergraph transversal problem runs in quasi-polynomial time $|V|N^{o(\log N)}$, where N is the combined input and output size $N = |\mathcal{G}| + |\operatorname{Tr}(\mathcal{G})|$. Several quasi-polynomial time algorithms with some other desirable properties also exist [8, 16, 18, 33]. While it is still open whether the problem can be solved in polynomial time for arbitrary hypergraphs, polynomial time algorithms exist for several classes of hypergraphs, e.g. hypergraphs of bounded edge-size [4, 12], of bounded-degree [11, 13], of bounded-edge intersections [4], of bounded conformality [4], of bounded treewidth [13], and read-once (exact) hypergraphs [15].

Almost all previously known polynomial-time algorithms for the the hypergraph transversal problem assume that at least one of the hypergraphs $\mathcal{G}$ or $\operatorname{Tr}(\mathcal{G})$ either (i) has bounded size $\min\{|\mathcal{G}|, |\operatorname{Tr}(\mathcal{G})|\} \leq k$, (ii) is k-conformal[3], or (iii) is k-degenerate[4], for a constant k. One can verify that all the special classes mentioned above belong to one of these categories.

In this paper, we shall extend these polynomially solvable classes to include hypergraphs arising in geometry. More precisely, we consider the following problems $\textsc{Hit}(\mathcal{V}, \mathcal{F})$:

- **Hitting hyper-rectangles by points:** Given a finite set of points $\mathcal{V} \subseteq \mathbb{R}^d$ and a finite collection $\mathcal{F}$ of axis-parallel hyper-rectangles (also called orthotopes or boxes) in $\mathbb{R}^d$, find all minimal sets of points from $\mathcal{V}$ that hit every hyper-rectangle in $\mathcal{F}$;
- **Hitting (Stabbing) connected objects by axis-parallel hyperplanes:** Given a finite set of axis-parallel hyperplanes $\mathcal{V} \subseteq \mathbb{R}^d$ and a finite collection $\mathcal{F}$ of connected objects in $\mathbb{R}^d$, find all minimal sets of hyperplanes from $\mathcal{V}$ that stab every object in $\mathcal{F}$;
- **Hitting half-spaces by points:** Given a finite set of points $\mathcal{V} \subseteq \mathbb{R}^d$ and a finite collection $\mathcal{F}$ of half-spaces in $\mathbb{R}^d$, find all minimal sets of points from $\mathcal{V}$ that hit every half-space in $\mathcal{F}$.

We show that the first two problems can be solved in incremental polynomial time, if the dimension d of the underlying space is bounded, and that the last problem can be solved in incremental polynomial time, if $d = 2$.

To construct efficient algorithms for the above problems, we first propose a general framework to solve the hypergraph transversal problem, which can be regarded as a generalization of the algorithms given in [13, 28], and apply it to the above problems. We remark that when we apply the framework to the problem of hitting half-planes by points, we need to run a backtracking algorithm at the base level of the recursion in the framework. While such an algorithm is inefficient in general, as it requires solving an NP-hard problem as a subroutine,

[3] A hypergraph is said to be k-conformal [2] if any set $X \subseteq V$ is contained in a hyperedge of $\mathcal{G}$ whenever each subset of X of cardinality at most k is contained in a hyperedge of $\mathcal{G}$.

[4] A hypergraph $\mathcal{G}$ is said to be k-degenerate [13] if for every set $X \subseteq V$, the minimum degree of a vertex in the induced hypergraph $\mathcal{G}_X$ on X is at most k.

we exploit the geometry to show that it can be made to work in the case of hitting half-planes by points.

We also consider the covering versions $\textsc{Cover}(\mathcal{V}, \mathcal{F})$ $(= \textsc{Hit}(\mathcal{F}, \mathcal{V}))$ of the above problems. For example, we consider the problem of finding all minimal sets of hyper-rectangles from $\mathcal{F}$ that hit all points in $\mathcal{V}$. We propose incremental polynomial-time algorithms for finding all minimal covers for the first two problems, by exploiting the fact that the geometric hypergraphs arising in the first two problems have the bounded *Helly property* [2], and show that minimal covers for the last problem can be generated in incremental polynomial time, by using *geometric duality* from the corresponding result for minimal hitting sets.

The enumeration of minimal geometric hitting sets, as the ones described above, arises in various areas such as computational geometry, machine learning, and data mining [14]. Moreover, our efficient enumeration algorithm might be useful in developing exact algorithms, fixed-parameter tractable algorithms, and polynomial-time approximation schemes for the corresponding optimization problems (see, e.g., [22]).

The rest of this paper is organized as follows. In the next section, we give a general framework for finding all minimal hitting sets for a given hypergraph. In Sections 3, 4 and 5, we apply this framework to hitting hyper-rectangles by points, stabbing connected objects by axis-parallel hyperplanes, and hitting half-planes by points.

2 A Framework for Computing Transversal Hypergraphs

Let $\mathcal{G} \subseteq 2^V$ be a hypergraph, and σ be an ordering of vertices in V. For $i = 1, \ldots, n$, let $\mathcal{G}_i$ be the sub-hypergraphs of $\mathcal{G}$ defined as $\mathcal{G}_i = \{G \in \mathcal{G} : G \subseteq \{v_{\sigma(1)}, \ldots, v_{\sigma(i)}\}\}$. Let us denote the set of hyperedges in $\mathcal{G}_i$ which are not contained in $\mathcal{G}_{i-1}$ as Δ_i, i. e., $\Delta_i = \mathcal{G}_i \setminus \mathcal{G}_{i-1}$ and for a set $X \subseteq V$, let $\Delta_i[X] = \{G \in \Delta_i : G \cap X = \emptyset\}$. Given a hypergraph $\mathcal{G}$, Eiter et.al. [13] describe an algorithm to generate $\mathrm{Tr}(\mathcal{G})$, the hypergraph consisting of all minimal transversals of $\mathcal{G}$. The algorithm proceeds inductively, for $i = 1, \ldots, n$, by extending each minimal transversal X in $\mathrm{Tr}(\mathcal{G}_{i-1})$ to a set in $\mathrm{Tr}(\mathcal{G}_i)$ by finding $\mathrm{Tr}(\Delta_i[X])$, each set of which is combined with X to obtain a minimal transversal of $\mathcal{G}_i$. In Figure 1, we present a generic algorithm which recursively reduces the problem of finding $\mathrm{Tr}(\mathcal{G}_i)$ into the smaller subproblems of computing $\mathrm{Tr}(\Delta_i[X])$ for $i \in [n]$ and $X \in \mathrm{Tr}(\mathcal{G}_{i-1})$.

The algorithm uses a sequence of permutations $\Sigma = \sigma_1 \ldots \sigma_k$ as a part of an input. When called initially as DUALIZE-INC$(\mathcal{G}, \Sigma, 1)$, it dualizes $\mathcal{G}$ by using the above mentioned approach where σ_j is used for partitioning in the j-th level of the recursion. The operator minimal$(\mathcal{H})$ in Step 9 returns the hypergraph obtained from a given hypergarph $\mathcal{H}$ by removing the non-minimal edges.

After k levels of recursion, procedure DUALIZE-SIMPLE() is used directly to solve the problem. As we will see in the later sections, for several classes of geometric hypergraphs, the subproblem after k levels can be solved easily, where k depends only on the dimension of the geometric space under consideration.

Procedure DUALIZE-INC$(\mathcal{G}, \Sigma, j)$**:**
Input: A hypergraph $\mathcal{G}$ over $n = |V(\mathcal{G})|$ vertices, an index $j\,(\leq k)$
and a sequence $\Sigma = (\sigma_1, \ldots, \sigma_k)$ of permutations of vertices in $V(\mathcal{G})$.
Output: The hypergraph $\mathcal{G}^d$.

1. $\mathcal{G}_0 \leftarrow \emptyset,\ \mathcal{X}_0 \leftarrow \{\emptyset\},\ \mathcal{X}_i \leftarrow \emptyset\ \forall i = 1, \ldots, n$
2. **for** $i = 1, \ldots, n$ **do**
3. Let $\mathcal{G}_i \leftarrow \{G \in \mathcal{G}\ :\ G \subseteq \{v_{\sigma_j(1)}, \ldots, v_{\sigma_j(i)}\}\}$
4. **for** each $X \in \mathcal{X}_{i-1}$ **do**
5. Let $\Delta_i[X] = \{G \in \mathcal{G}_i \setminus \mathcal{G}_{i-1}\ :\ G \cap X = \emptyset\}$
6. **if** $j \geq k$ or $|\Delta_i[X]| \leq 1$ **then**
7. $\mathcal{A} \leftarrow$ **DUALIZE-SIMPLE**$(\Delta_i[X])$
8. **else** $\mathcal{A} \leftarrow$ **DUALIZE-INC**$(\Delta_i[X], \Sigma, j+1)$
9. $\mathcal{X}_i \leftarrow \text{minimal}\,(\mathcal{X}_i \bigcup \{X \cup Y\ :\ Y \in \mathcal{A}\})$
10. **return** $\mathcal{X}_n$

Fig. 1. A generic sequential method for finding minimal transversals

As an illustration, consider the problem of dualizing an *interval* hypergraph: Let $v_1, v_2, \ldots, v_n$ be a set of points on the line ordered from left to right, and let $\mathcal{G} \subseteq 2^V$ be a *Sperner*[5] hypergraph, in which each edge $G \in \mathcal{G}$ consists of consecutive points from V. Denote by σ the left-to-right ordering of the vertices, and consider the execution of the algorithm when called as DUALIZE-INC$(\mathcal{G}, \Sigma, 1)$ with $\Sigma = \sigma$. The algorithm incrementally dualizes the hypergraphs $\mathcal{G}_i = \{G \in \mathcal{G} : G \subseteq \{v_1, \ldots, v_i\}\}$ for $i = 1 \ldots n$. Note that the subproblem $\Delta_i = \mathcal{G}_i \setminus \mathcal{G}_{i-1}$ contains at most one edge because of our assumption that $\mathcal{G}$ is Sperner and thus can be solved trivially.

The correctness of the procedure follows from the following statement.

Proposition 1 ([13]). *For* $i = 1, \ldots, n$, $\mathrm{Tr}(\mathcal{G}_i) = \mathit{minimal}(\{X \cup Y\ :\ X \in \mathrm{Tr}(\mathcal{G}_{i-1}),\ Y \in \mathrm{Tr}(\Delta_i[X])\})$.

It is shown in [13] that the intermediate hypergraphs obtained in the algorithm never get too large, more specifically, $|\,\mathrm{Tr}(\Delta_i[X])| \leq |\,\mathrm{Tr}(\mathcal{G}_i)| \leq |\,\mathrm{Tr}(\mathcal{G})|$. Consequently, we get the following bound on the worst-case running time.

Theorem 1. *Let* $\mathcal{G} \subseteq 2^V$ *be a hypergraph over vertex set* V *and* $\Sigma = (\sigma_1, \ldots, \sigma_k)$ *be a sequence of permutation functions of vertices of* $\mathcal{G}$*. Then the procedure DUALIZE-INC*$(\mathcal{G}, \Sigma, 1)$ *computes* $\mathrm{Tr}(\mathcal{G})$ *in* $\mathcal{O}((nm')^k(mm' + T))$ *time, where* $n = |V|$, $m = |\mathcal{G}|$, $m' = |\,\mathrm{Tr}(\mathcal{G})|$ *and* T *is time required by DUALIZE-SIMPLE in each* k*-th level recursion of the procedure.*

3 Points and Hyper-rectangles in $\mathbb{R}^d$

Let $\mathcal{V}$ be a set of points and $\mathcal{F}$ be a collection of axis-parallel hyper-rectangles in $\mathbb{R}^d$. In this section, we consider the problem of enumerating all minimal hitting

[5] A hypergraph $\mathcal{H}$ is said to be *Sperner* if no hyperedge of $\mathcal{H}$ contains another.

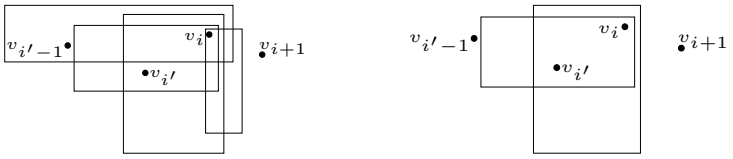

Fig. 2. An example of points and rectangles in $\mathbb{R}^2$. Left: The set Δ_i consists of all rectangles that contain v_i and other points only from the subset $\{v_1, \ldots, v_i\}$. Right: The subproblem in the recursive call considers all rectangles which contain both v_i and $v_{i'}$ and no points from the strict left of $v_{i'}$ nor from the strict right of v_i

sets for $\mathcal{F}$ from $\mathcal{V}$ as well as the related problem of enumerating all minimal covers of $\mathcal{V}$ by $\mathcal{F}$. Let $\mathcal{G} \subseteq 2^V$ be the hypergraph defined by $V = \mathcal{V}$ and $\mathcal{G} = \mathcal{G}(\mathcal{V}, \mathcal{F}) \stackrel{\text{def}}{=} \{\{v \in \mathcal{V} \ : \ v \cap F \neq \emptyset\} \ : \ F \in \mathcal{F}\}$. Then the transposed hypergraph $\mathcal{G}^T \subseteq 2^{\mathcal{F}}$ is defined as $\mathcal{G}^T = \mathcal{G}(\mathcal{F}, \mathcal{V})$. Clearly, a minimal set of points from $\mathcal{V}$ hitting every hyper-rectangle in $\mathcal{F}$ corresponds to a minimal hitting set of $\mathcal{G}$, while a minimal set of hyper-rectangles from $\mathcal{F}$ covering every point in $\mathcal{V}$ corresponds to a minimal hitting set for $\mathcal{G}^T$. For a hypergraph $\mathcal{G}$, we will denote by $V(\mathcal{G})$ the vertex set of $\mathcal{G}$.

3.1 Minimal Hitting Sets

To illustrate the idea, let us first consider the problem in $\mathbb{R}^2$. The algorithm is based on the framework presented in Figure 1. We order the points in $\mathcal{V}$ from *left* to *right* and if their x-coordinates are equal, we sort them from *bottom* to *top*. Let $v_1, v_2, \ldots, v_n$ be the corresponding ordering of the vertices of the hypergraph $\mathcal{G} \subseteq 2^{\mathcal{V}}$, defined above. Note that because of our ordering of the vertices, no rectangle in the hypergraph $\mathcal{G}_i$ contains any point strictly to the right of v_i and by definition, every rectangle in $\Delta_i \subseteq \mathcal{G}_i$ contains v_i.

Consider the subproblem of dualizing $\Delta_i[X]$ for each $i \in [n]$ and $X \in \mathrm{Tr}(\mathcal{G}_{i-1})$ and let the primed variables denote the corresponding variables in the recursive call of the algorithm. We order the vertices of $\mathcal{G}' = \Delta_i[X]$ in the *reverse* order i. e., from *right* to *left*, breaking ties by sorting them from *top* to *bottom*.

Consider now a further recursive call of the procedure on a hypergraph $\mathcal{G}'' = \Delta'_{i'}[X']$, where $i' \in \{1, \ldots, |V(\mathcal{G}')|\}$ and $X' \in \mathrm{Tr}(\mathcal{G}'_{i'-1})$. The crucial observation is that each rectangle corresponding to an edge in the hypergraph $\Delta'_{i'}[X']$ contains both the points v_i and $v_{i'}$, and because of our ordering, no rectangle in the subproblem contains a point from the left of $v_{i'}$ nor from the right of v_i (see Figure 2). Hence, as can be easily seen, only the y-coordinates matter when deciding whether a given point $v \in \{v_{i'} \ldots v_i\}$ intersects a rectangle from $\Delta'_{i'}[X']$. So we can project the subproblem $\Delta'_{i'}[X']$ on the y-axis and reduce it to a problem of dualizing an interval hypergraph, which can be solved in polynomial time, as seen in Section 2.

The above algorithm can be extended to higher dimensions in an obvious manner. In dimension d, we use the orderings $\Sigma = (\sigma_1, \ldots, \sigma_{2d-2})$, where σ_i is

the *lexicographical ordering* of the points using their last $d - \lfloor \frac{i-1}{2} \rfloor$ coordinates. Moreover, we define the ordering σ_i to be *increasing* when i is odd and *decreasing* otherwise. To generate all minimal transversals of $\mathcal{G}$, we call DUALIZE-INC$(\mathcal{G}, \Sigma, 1)$, and use the dualization procedure for interval hypergraphs in place of DUALIZE-SIMPLE(). After the second recursive call the subproblems we obtain contain all hyper-rectangles that intersect two given points, say $v_{i'}$ and v_i with $v_{i'}$ being lexicographically smaller than v_i, and have the property that they contain no point that is lexicographically smaller then $v_{i'}$ or lexicographically greater than v_i. Hence after two levels of recursion, the first coordinate of the points can be ignored and thus the problem reduces to $d-1$-dimensional subproblems.

3.2 Minimal Covers

As mentioned above, to generate all minimal covers of the given points $\mathcal{V}$ by hyper-rectangles from $\mathcal{F}$, we consider the *transposed* hypergraph $\mathcal{G}^T(\mathcal{V}, \mathcal{F})$ and compute its transversal hypergraph.

Halman [21] showed that any hypergraph $\mathcal{G} = \mathcal{G}(\mathcal{V}, \mathcal{F})$ defined by a set of hyper-rectangles $\mathcal{F}$ and a set of points $\mathcal{V}$ in $\mathbb{R}^d$ is $2d$-*Helly*, that is, for any $\mathcal{G}' \subseteq \mathcal{G}$, the following holds: if every $2d$ edges in $\mathcal{G}'$ have a non-empty intersection then all edges in $\mathcal{G}'$ have a non-empty intersection. Thus the transposed hypergraph $\mathcal{G}^T$ is $2d$-conformal (cf. [2]), and hence can be dualized by the algorithm of Khachiyan et al. [27] in incremental polynomial time.

Theorem 2. *Both problems* $\textsc{Hit}(\mathcal{V}, \mathcal{F})$ *and* $\textsc{Cover}(\mathcal{V}, \mathcal{F})$ *can be solved in time* $O((|\mathcal{V}||\mathcal{F}|k)^{O(d)})$*, when* $\mathcal{V}$ *and* $\mathcal{F}$ *are, respectively, a set of points and a set of axis-parallel hyper-rectangles in* $\mathbb{R}^d$*, and* k *is the size of the output.*

4 Stabbing Connected Objects in $\mathbb{R}^d$

4.1 Minimal Stabbing Sets

Given a collection of connected objects $\mathcal{F}$ and a set of axis-parallel hyperplanes $\mathcal{V}$, both in $\mathbb{R}^d$, we are interested in the problem of finding all minimal sets of hyperplanes from $\mathcal{V}$ such that every object in $\mathcal{F}$ is stabbed by at least one of the hyperplanes in the set. Let $\mathcal{G} = \mathcal{G}(\mathcal{V}, \mathcal{F})$ be the corresponding hypergraph with vertex set $\mathcal{V}$ and each object $F \in \mathcal{F}$ defining an edge consisting of all hyperplanes from $\mathcal{V}$ which intersect F.

We consider the simple case first, that is, when all hyperplanes in $\mathcal{V}$ are parallel to each other. This turns out to be equivalent to the interval hypergraph case since we can project the problem on any line L that is perpendicular to the hyperplanes in $\mathcal{V}$. The projection maps hyperplanes in $\mathcal{V}$ into points on L, and because of the connectivity, the objects in $\mathcal{F}$ are mapped to intervals on L.

More generally, for $d > 1$, assuming there is at least one hyperplane perpendicular to every principal axis, there are exactly d groups of axis-parallel hyperplanes in $\mathbb{R}^d$ such that every group contains hyperplanes that are parallel

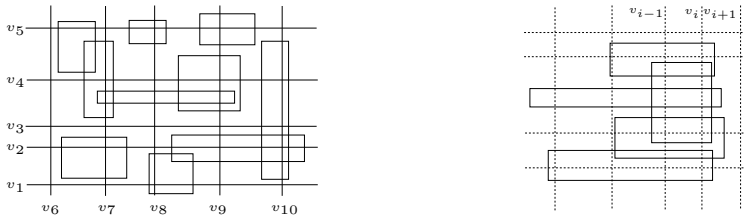

Fig. 3. Left: An example of lines and objects in $\mathbb{R}^2$ and a valid ordering of lines. Right: The set $\Delta_i[X]$ consists of new objects that intersect v_i and only lines from the subset $\{v_1, \ldots, v_{i-1}\}$.

to one principal axis. This assumption can be made without loss of generality, since if there is no hyperplane along a particular principal axis, say z, then we can orthogonally project all other hyperplanes and objects on the hyperplane $z = 0$ and reduce the dimension of the problem by one.

The dual of $\mathcal{G}$ can be found incrementally by following the algorithm of Figure 1. Fixing the order of principal axes, we order the hyperplanes sequentially: starting with hyperplanes perpendicular to the first principal axis, sorted in increasing order, we continue with the hyperplanes perpendicular to the second principal axis and so on. For an example in $\mathbb{R}^2$, the set of lines $\{x = 1, y = -1, x = 0, y = 1\}$ would be ordered as $x = 0, x = 1, y = -1, y = 1$ assuming that x-axis comes before y-axis in our fixed ordering of principal axes. Let σ be an ordering of hyperplanes in $\mathcal{G}$ as defined above and let $j_0^{\mathcal{G}} < j_1^{\mathcal{G}} < \ldots < j_d^{\mathcal{G}}$ be indices with $j_0^{\mathcal{G}} = 0$ and $j_d^{\mathcal{G}} = n$, such that the hyperplanes in the group $\sigma(j_{r-1}^{\mathcal{G}} + 1), \ldots, \sigma(j_r^{\mathcal{G}})$ are parallel to each other and perpendicular to r-th principal axis for $r \in [d]$.

Consider the subproblem of dualizing $\mathcal{G}_i$ for $i = 1, \ldots, n$ as defined in the algorithm, where $\mathcal{G}_i$ contains only those edges which form subsets of vertices from $v_{\sigma(1)}, \ldots, v_{\sigma(i)}$. As discussed above, the problem reduces to dualizing an interval hypergraph when $1 \leq i \leq j_1^{\mathcal{G}}$. Now consider the case when $j_{r-1}^{\mathcal{G}} < i \leq j_r^{\mathcal{G}}$ for $r \in [d]$. Consider the subproblem of dualizing $\mathcal{G}' = \Delta_i[X]$ for $X \in \mathrm{Tr}(\mathcal{G}_{i-1})$, and let the primed variables denote the corresponding variables in the recursive call of the algorithm. Note that the subproblem for $\Delta_i[X]$ contains all objects that do not intersect any hyperplane "above" $v_{\sigma(i)}$. Let σ' be an ordering of vertices of $\mathcal{G}'$ defined similarly as σ for the hyperplanes perpendicular to first $r-1$ principal axes except the r-th group of hyperplanes which are sorted in decreasing order. As before, let $j_0^{\mathcal{G}'} < \ldots < j_r^{\mathcal{G}'}$ be indices with $j_0^{\mathcal{G}'} = 0$ and $j_r^{\mathcal{G}'} = n'$, where $n' = |V(\mathcal{G}')|$ and the hyperplanes in the group $\sigma'(j_{r'-1}^{\mathcal{G}'} + 1), \ldots, \sigma'(j_{r'}^{\mathcal{G}'})$ are parallel to each other and perpendicular to r'-th principal axis for $r' \in [r]$.

In the recursive call, we use σ' as our ordering and dualize $\mathcal{G}'$ incrementally by considering $\mathcal{G}'_{i'}$ for $1 < i' \leq i$. Note that for $i' \leq j_{r-1}^{\mathcal{G}'}$, $V(\mathcal{G}'_{i'}) \subseteq V(\mathcal{G}_{i-1})$ and hence the subproblem $\mathcal{G}'_{i'}$ is already taken care of when $\mathrm{Tr}(\mathcal{G}_{i-1})$ is computed[6].

[6] Note that the set of all minimal transversals of the sub-hypergraph $\mathcal{H}_S$ induced by vertices in S is equivalent to the set $\mathrm{Tr}(\mathcal{H}_S) = \mathrm{minimal}(\{H \cap S : H \in \mathrm{Tr}(\mathcal{H})\})$

Alternatively, when $j_{r-1}^{\mathcal{G}'} < i' \leq i$ then similar to the case in Section 3.1, the subproblems $\Delta'_{i'}$ we get contain all objects that intersect both v_i and $v_{i'}$ with the property that no hyperplane above v_i or below $v_{i'}$ stabs any of them. Note that both $\{v_i\}$ and $\{v_{i'}\}$ as well as all hyperplanes between them are trivially hitting sets for the subproblems for hypergraphs of the form $\Delta'_{i'}[\cdot]$. The other hitting sets can be found recursively by observing that they do not involve any of the hyperplanes parallel to v_i and $v_{i'}$. Thus we are able to reduce the dimension of the problem by 1 after two levels of recursion.

In summary, to generate $\mathrm{Tr}(\mathcal{G})$, we call DUALIZE-INC$(\mathcal{G}, \Sigma, 1)$ with $\Sigma = (\sigma_1, \sigma_2, \ldots, \sigma_{2d-1})$ and a trivial dualization procedure for DUALIZE-SIMPLE(). For odd r, $1 \leq r < 2d$, the ordering σ_r sorts each group of parallel hyperplanes in increasing order (of the points of intersection with the common orthogonal line), whereas for even r, $1 < r < 2d$, σ_r is defined by sorting each group of parallel hyperplanes in increasing order except the last group, which is sorted in decreasing order (of the points of intersection with the common orthogonal line). As we noted above, at every r-th level of recursion for even r, the dual hypergraphs $\mathrm{Tr}(\mathcal{G}_i)$ such that $v_{\sigma(i)}$ does not belong to the last group of hyperplanes, can be easily computed from the corresponding dual hypergraph in the recursion level $r-1$. This observation can be used to avoid redundant computations by solving those subproblems directly instead of following the algorithm of Figure 1.

4.2 Minimal Covers

We use again the algorithm of [27] for dualizing conformal hypergraphs.

Lemma 1. *Let $\mathcal{F}$ be the set of connected objects and $\mathcal{V}$ be set of axis-parallel hyperplanes in $\mathbb{R}^d$. Then the corresponding hypergraph $\mathcal{G}(\mathcal{F}, \mathcal{V})$ is $2d$-Helly.*

Theorem 3. *Both problems* $\textsc{Hit}(\mathcal{V}, \mathcal{F})$ *and* $\textsc{Cover}(\mathcal{V}, \mathcal{F})$ *can be solved in time $O((|\mathcal{V}||\mathcal{F}|k)^{O(d)})$, when $\mathcal{V}$ and $\mathcal{F}$ are, respectively, a set of axis-parallel hyperplanes and a set of connected objects in $\mathbb{R}^d$, and k is the size of the output.*

5 Hitting and Covering with Half-planes

In this section we consider the case when the hypergraph $\mathcal{G} = \mathcal{G}(\mathcal{V}, \mathcal{F})$ is defined by a finite set of points $\mathcal{V} \subseteq \mathbb{R}^2$ and a set of half-planes $\mathcal{F}$. In the following, we will refer to the elements of $\mathcal{V}$ as points or vertices interchangeably.

5.1 Minimal hitting sets

Let $\Sigma = (\sigma'_1, \sigma''_1, \sigma'_2, \sigma''_2)$, where for $i \in \{1, 2\}$, σ'_i and σ''_i are respectively the ascending and descending orderings of the points in $\mathcal{V}$ along the ith coordinate. We call the procedure DUALIZE-INC$(\mathcal{G}, \Sigma, 1)$.

When the procedure reaches the base level, we end-up with an instance on a hypergraph $\mathcal{G}' = \mathcal{G}(\mathcal{V}', \mathcal{F}')$ in which all points $\mathcal{V}$ lie inside a rectangle R, the boundary of which contains four (not necessarily distinct) points $p_1, p_2, p_3, p_4 \in$

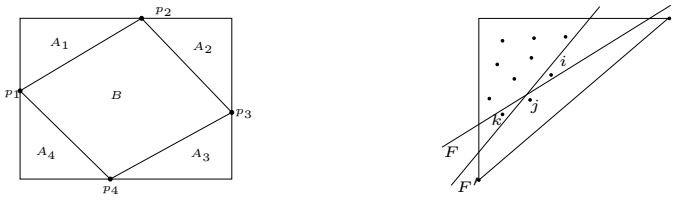

Fig. 4. Left: All hyperplanes in the subproblem contain the points $\{p_1, p_2, p_3, p_4\}$. Right: One of the subproblems consists of all points in the right-angle triangle and all halfplanes containing the longest side of it.

$\mathcal{V}$, one on each side of R, and such that for each $F \in \mathcal{F}'$, $F \supseteq \{p_1, p_2, p_3, p_4\}$. Consider the polygon connecting these four points: it partitions the rectangle R into 5 regions (some possibly empty); see Figure 4, Left. Let $B \subseteq \mathcal{V}'$ be the set of points inside the polygon and A_1, A_2, A_3, A_4 be the sets of points inside the other four regions. Note that, without loss of generality, we can assume that any line defining a half-plane in $\mathcal{F}'$ intersects *exactly one* of the 4 regions (otherwise, there is only one hyperplane and $\mathrm{Tr}(\mathcal{G}') = \{\{v\} : v \in \mathcal{V}'\}$). For $i = 1, 2, 3, 4$, let $\mathcal{F}_i$ be the half-planes that hit the ith region, and $\mathcal{H}_i = \mathcal{G}(A_i, \mathcal{F}_i)$ be the corresponding hypergraph.

Lemma 2. $\mathrm{Tr}(\mathcal{G}') = \mathit{minimal}\ (\mathcal{T}' \cup \mathcal{T}'' \cup \bigcup_{i=1}^{4} \mathrm{Tr}(\mathcal{H}_i))\ \mathit{where}\ \mathcal{T}' = \{\{v\} : v \in B\}$ *and* $\mathcal{T}'' = \{\{v, v'\}\ :\ v \in A_i,\ v' \in A_j,\ i \neq j\ \ \mathit{and}\ \{v, v'\}\ \mathit{is\ transversal\ to}\ \mathcal{G}'\}$.

5.2 Backtracking Method

Given a hypergraph $\mathcal{H} \subseteq 2^V$ and a subset $S \subseteq V$ of vertices, [6] gave a criterion to decide if S is a *sub-transversal* of $\mathcal{G}$, i. e., if there is a minimal transversal $T \in \mathrm{Tr}(\mathcal{G})$ such that $T \supseteq S$. In general, this is an NP-hard problem even if $\mathcal{G}$ is a graph (see [4]). However, if $|S|$ is bounded by a constant, or if the hypergraph is read-once [15], then such a check can be done in polynomial time (see [6]). Here we give another instance in which such a check can also be performed in polynomial time.

For a subset $S \subseteq V$, and a vertex $v \in S$, let $\mathcal{H}_v(S) = \{H \in \mathcal{H} \mid H \cap S = \{v\}\}$. A selection of $|S|$ hyperedges $\{H_v \in \mathcal{H}_v(S) \mid v \in S\}$ is called *covering* if there exists a hyperedge $H \in \mathcal{H}_0 = \{H \in \mathcal{H}\ :\ H \cap S = \emptyset\}$ such that $H \subseteq \bigcup_{v \in S} H_v$.

Proposition 2 ([6]). *A non-empty subset $S \subseteq V$ is a sub-transversal for $\mathcal{H} \subseteq 2^V$ if and only if there is a non-covering selection $\{H_v \in \mathcal{H}_v(S) \mid v \in S\}$ for S.*

The algorithm is given in Figure 5, and is based on the standard backtracking technique for enumeration (see e.g. [32, 15]). The procedure is called initially with $S_1 = S_2 = \emptyset$ and $i = 1$. It is easy to verify that the algorithm outputs all elements of the transversal hypergraph $\mathrm{Tr}(\mathcal{H})$, without repetition (and in lexicographic ordering according to the input permutation σ). Since the algorithm essentially builds a backtracking tree whose leaves are the minimal transversals of $\mathcal{G}$, the

Procedure DUALIZE-BT$(\mathcal{H}, \sigma, i, S)$**:**
Input: A hypergraph $\mathcal{H} \subseteq 2^V$, an ordering σ on V,
an integer $i \in \{1, \ldots, |V|\}$ and a subset $S \subseteq \sigma([i-1]) \stackrel{\text{def}}{=} \{\sigma(j) : j \in [i-1]\}$
Output: The set $\{T \in \mathrm{Tr}(\mathcal{H}) : T \supseteq S, T \cap (\sigma([i-1]) \setminus S) = \emptyset\}$
1. **if** $S \in \mathrm{Tr}(\mathcal{H})$ **then**
2. **output** S and **return**
3. **if** $\exists T \in \mathrm{Tr}(\mathcal{H})$ s.t. $S \cup \{\sigma(i)\} \subseteq T$ and $(\sigma([i-1]) \setminus S) \cap T = \emptyset$ **then**
4. **DUALIZE-BT**$(\mathcal{H}, \sigma, i+1, S \cup \{\sigma(i)\})$
5. **DUALIZE-BT**$(\mathcal{H}, \sigma, i+1, S)$

Fig. 5. The backtracking method for finding minimal transversals

time required to produce each new minimal transversal is bounded by the depth of the tree (at most $\min\{|V|, |\mathcal{H}|\}$) times the maximum time required at each node. The efficiency of such procedure depends on being able to perform the test in Step 3, which is addressed in the next subsection.

5.3 Solving the Special Instance

Without loss of generality we concentrate on finding $\mathrm{Tr}(\mathcal{H}_1)$, where $\mathcal{H}_1$ is the hypergraph defined in Section 5.1. The other 3 sets of transversals can be found similarly. In other words, we may assume that all points lie inside a right-angle triangle of which the hyperplanes contain the longest side (see Figure 4, Right).

We use the backtracking method with σ being the following lexicographic order of the points: if $p = (p_1, p_2)$ and $q = (q_1, q_2)$ then $p \prec_\sigma q$ if and only if $p_1 < q_1$ or $p_1 = q_1$ and $p_2 < q_2$. Without loss of generality, we assume $V(\mathcal{H}_1) = \{1, \ldots, n\}$ and reorder the points such that they are numbered from 1 to n according to σ, i. e., assume that σ is the identity permutation.

Now we show that the sub-transversal test in Step 3 of the backtracking procedure in Figure 5 can be performed in polynomial time. Given $i \in [n]$ and $S \subseteq [i-1]$, we would like to check if

$$\exists T \in \mathrm{Tr}(\mathcal{H}_1) \text{ such that } S \cup \{i\} \subseteq T \text{ and } ([i-1] \setminus S) \cap T = \emptyset. \tag{1}$$

Lemma 3. *Fix $i \in [n]$ and let $S \subseteq [i-1]$ be a sub-transversal of $\mathcal{H}_1$. Suppose that $F, F' \in \mathcal{H}_1$ satisfy: $F \cap (S \cup \{i\}) = \{j\}$ for $j \neq i$ and $F' \cap (S \cup \{i\}) = \{i\}$. Then $F \setminus [i-1] \subseteq H'$.*

Proof. If there is a point with index $k \in F \setminus ([i-1] \cup F')$ then k comes before i with respect to the first coordinate (see Figure 4, Right), in contradiction to the fact that we process the points according to the order imposed by σ. □

Now the sub-transversal criterion of Proposition 2 reduces to a simple check.

Lemma 4. *Given $i \in [n]$ and $S \subseteq [i-1]$. Then (1) holds if and only if there exists $F \in \mathcal{H}_1$ such that $F \cap (S \cup \{i\}) = \{i\}$ and for all $F' \in \mathcal{H}_1$ for which $F' \cap (S \cup \{i\}) = \emptyset$, we have $(F' \setminus [i]) \setminus (F \setminus [i]) \neq \emptyset$.*

Proof. We apply the sub-transversal criterion for $S \cup \{i\}$ in the restricted hypergraph $\mathcal{H}'_1 = \{H \setminus ([i-1] \setminus S) \ : \ H \in \mathcal{H}_1\}$. By Proposition 2, (1) holds if and only if there exists $F_1, \ldots, F_i \in \mathcal{H}_1$ such that $F_j \cap (S \cup \{i\}) = \{j\}$, for $j = 1, \ldots, i$, and $(F' \setminus [i-1]) \not\subseteq \bigcup_{j=1}^{i} (F_j \setminus [i-1])$ for all $F' \in \mathcal{H}_1$ such that $F' \cap (S \cup \{i\}) = \emptyset$. By Lemma 3, the union $\bigcup_{j=1}^{i} (F_j \setminus [i-1])$ is equal to $F_i \setminus [i-1]$. □

5.4 Minimal Covers

By *geometric* duality (see e.g. [10], Chapter 8), the problem of finding minimal set covers reduces to finding all minimal hitting sets. Indeed, we may assume by rotating the given instance if necessary that all points are in general position. If we use the mapping $(p_1, p_2) \mapsto y = p_1 x + p_2$ that maps the point $p = (p_1, p_2)$ in the original space to the line $\{(x, y) : y = p_1 x + p_2\}$, then one can easily see that minimal set covers in one space correspond to minimal hitting sets in the other space and vice versa.

Theorem 4. *Both problems* $\textsc{Hit}(\mathcal{V}, \mathcal{F})$ *and* $\textsc{Cover}(\mathcal{V}, \mathcal{F})$ *can be solved in time* $O((|\mathcal{V}||\mathcal{F}|k)^{O(1)})$*, when* $\mathcal{V}$ *and* $\mathcal{F}$ *are, respectively, a set of points and a set of half-planes in* $\mathbb{R}^2$*, and* k *is the size of the output.*

References

1. M. Anthony and N. Biggs. *Computational learning theory: an introduction.* Cambridge University Press, New York, NY, USA, 1992.
2. C. Berge. *Hypergraphs.* Elsevier-North Holand, Amsterdam, 1989.
3. J. C. Bioch and T. Ibaraki. Complexity of identification and dualization of positive boolean functions. *Information and Computation*, 123(1):50–63, 1995.
4. E. Boros, K. Elbassioni, V. Gurvich, and L. Khachiyan. Generating maximal independent sets for hypergraphs with bounded edge-intersections. In *LATIN '04*, pages 488–498, 2004.
5. E. Boros, K. Elbassioni, V. Gurvich, L. Khachiyan, and K. Makino. Dual-bounded generating problems: All minimal integer solutions for a monotone system of linear inequalities. *SIAM J. Computing*, 31(5):1624–1643, 2002.
6. E. Boros, V. Gurvich, and P.L. Hammer. Dual subimplicants of positive boolean functions. *Optim. Methods Softw.*, 10:147–156, 1998.
7. E. Boros, V. Gurvich, L. Khachiyan, and K. Makino. On the complexity of generating maximal frequent and minimal infrequent sets. In *STACS '02*, pages 133–141, 2002.
8. E. Boros and K. Makino. A fast and simple parallel algorithm for the monotone duality problem. In *ICALP '09, to appear*, 2009.
9. C. J. Colbourn. *The Combinatorics of Network Reliability.* Oxford University Press, NY, USA, 1987.
10. M. de Berg, M. van Kreveld, M. Overmars, and O. Schwarzkopf. *Computational Geometry, Algorithms and Applications.* SpringerVerlag, Amsterdam, 1997.
11. C. Domingo, N. Mishra, and L. Pitt. Efficient read-restricted monotone cnf/dnf dualization by learning with membership queries. *Machine Learning*, 37(1):89–110, 1999.

12. T. Eiter and G. Gottlob. Identifying the minimal transversals of a hypergraph and related problems. *SIAM J. Computing*, 24(6):1278–1304, 1995.
13. T. Eiter, G. Gottlob, and K. Makino. New results on monotone dualization and generating hypergraph transversals. *SIAM J. Computing*, 32(2):514–537, 2003.
14. T. Eiter, K. Makino, and G. Gottlob. Computational aspects of monotone dualization: A brief survey. *Discrete Applied Mathematics*, 156(11):2035–2049, 2008.
15. Thomas Eiter. Exact transversal hypergraphs and application to Boolean μ-functions. *Journal of Symbolic Computation*, 17(3):215–225, 1994.
16. K. Elbassioni. On the complexity of the multiplication method for monotone CNF/DNF dualization. In *ESA '06*, pages 340–351, 2006.
17. M. L. Fredman and L. Khachiyan. On the complexity of dualization of monotone disjunctive normal forms. *Journal of Algorithms*, 21:618–628, 1996.
18. D. R. Gaur and R. Krishnamurti. Average case self-duality of monotone boolean functions. In *Canadian AI '04*, pages 322–338, 2004.
19. G. Gottlob. Hypergraph transversals. In *FoIKS '04: Proc. of the 3rd International Symposium on Foundations of Information and Knowledge Systems*, pages 1–5, 2004.
20. D. Gunopulos, H. Mannila, R. Khardon, and H. Toivonen. Data mining, hypergraph transversals, and machine learning (extended abstract). In *PODS '97*, 1997.
21. N. Halman. On the power of discrete and of lexicographic Helly-type theorems. In *FOCS '04*, pages 463–472, 2004.
22. F. Hüffner, R. Niedermeier, and S. Wernicke. Techniques for practical fixed-parameter algorithms. *Comput. J.*, 51(1):7–25, 2008.
23. T. Ibaraki and T. Kameda. A theory of coteries: Mutual exclusion in distributed systems. *IEEE Trans. on Parallel and Distributed Systems*, 4(7):779–794, 1993.
24. D. J. Kavvadias, C. H. Papadimitriou, and M. Sideri. On horn envelopes and hypergraph transversals. In *ISAAC '93*, pages 399–405, 1993.
25. L. Khachiyan. Transversal hypergraphs and families of polyhedral cones. In *Advances in Convex Analysis and Global Optimization, honoring the memory of K. Carathéodory*, pages 105–118. Kluwer, 2000.
26. L. Khachiyan, E. Boros, K. Borys, K. Elbassioni, V. Gurvich, and K. Makino. Enumerating spanning and connected subsets in graphs and matroids. In *ESA '06*, pages 444–455, 2006.
27. L. Khachiyan, E. Boros, K. Elbassioni, and V. Gurvich. A global parallel algorithm for the hypergraph transversal problem. *Information Processing Letters*, 101(4):148–155, 2007.
28. E. Lawler, J. K. Lenstra, and A. H. G. Rinnooy Kan. Generating all maximal independent sets: NP-hardness and polynomial-time algorithms. *SIAM J. Computing*, 9:558–565, 1980.
29. L. Lovász. Combinatorial optimization: some problems and trends. DIMACS Technical Report 92-53, Rutgers University, 1992.
30. H. Mannila and K. J. Räihä. Design by example: An application of armstrong relations. *Journal of Computer and System Sciences*, 33(2):126–141, 1986.
31. C. Papadimitriou. NP-completeness: A retrospective. In *ICALP '97*, 1997.
32. R. C. Read and R. E. Tarjan. Bounds on backtrack algorithms for listing cycles, paths, and spanning trees. *Networks*, 5:237–252, 1975.
33. H. Tamaki. Space-efficient enumeration of minimal transversals of a hypergraph. Technical Report IPSJ-AL 75, 2000.