

Dual-Bounded Generating Problems: All Minimal Integer Solutions for a Monotone System of Linear Inequalities^{*†}

E. Boros[‡] K. Elbassioni[§] V. Gurvich[‡] L. Khachiyan[§]
K. Makino[¶]

Abstract

We consider the problem of enumerating all minimal integer solutions of a monotone system of linear inequalities. We first show that for any monotone system of r linear inequalities in n variables, the number of maximal infeasible integer vectors is at most rn times the number of minimal integer solutions to the system. This bound is accurate up to a *polylog*(r) factor and leads to a polynomial-time reduction of the enumeration problem to a natural generalization of the well-known dualization problem for hypergraphs, in which dual pairs of hypergraphs are replaced by dual collections of integer vectors in a box. We provide a quasi-polynomial algorithm for the latter dualization problem. These results imply, in particular, that the problem of incrementally generating all minimal integer solutions to a monotone system of linear inequalities can be done in quasi-polynomial time.

^{*}An extended abstract of this paper containing some of the results appeared in the *Proceedings of the 28th International Colloquium on Automata, Languages and Programming (ICALP2001)*, Crete, Greece, July 8-12, 2001.

[†]This research was supported by the National Science Foundation (Grant IIS-0118635), and by the Office of Naval Research (Grant N00014-92-J-1375). The second and third authors are also grateful for the partial support by DIMACS, the National Science Foundation's Center for Discrete Mathematics and Theoretical Computer Science. The fifth author is also grateful for the partial support by the Scientific Grant in Aid of the Ministry of Education, Culture, Sports, Science and Technology of Japan.

[‡]RUTCOR, Rutgers University, 640 Bartholomew Road, Piscataway NJ 08854-8003; ({boros, gurvich}@rutcor.rutgers.edu).

[§]Department of Computer Science, Rutgers University, 110 Frelinghuysen Road, Piscataway, New Jersey 08854-8004; ({elbassio@paul, 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)

1 Introduction

Consider a system of r linear inequalities in n integer variables

$$Ax \geq b, \quad x \in \mathcal{C} = \{x \in \mathbb{Z}^n \mid 0 \leq x \leq c\}, \quad (1)$$

where $A \in \mathbb{R}^{r \times n}$ is a given $r \times n$ -matrix, $b \in \mathbb{R}^r$ is an r -vector, $c \in \mathbb{R}_+^n$ is a non-negative n -vector some or all of whose components may be infinite, and where \mathbb{Z} , \mathbb{R} and \mathbb{R}_+ denote, respectively, the set of integers, the set of reals and the set of non-negative reals. A vector $x \in \mathcal{C}$ is called a *feasible solution* of (1), if $Ax \geq b$. We shall assume that (1) is a *monotone system*, i.e. if $x \in \mathcal{C}$ is feasible, then all vectors $y \in \mathcal{C}$ with $y \geq x$ are also feasible. For instance, (1) is monotone if the matrix A is non-negative, though the non-negativity of A is not necessary, e.g. if $r = 1$, $n = 2$, and $c = (1, 2)$, then the system $5x_1 - x_2 \geq 3$ is monotone.

Let us denote by $\mathcal{F}_{A,b,c}$ the set of all minimal feasible solutions for the monotone system (1):

$$\mathcal{F}_{A,b,c} = \left\{ y \mid \begin{array}{l} y \in \mathcal{C}, \quad Ay \geq b \text{ and} \\ \nexists \text{ feasible solution } x \text{ of (1)} \\ \text{such that } x \leq y \text{ and } x \neq y \end{array} \right\}.$$

In this paper, we are concerned with the problem of generating all vectors of $\mathcal{F}_{A,b,c}$ for a monotone system (1). Since the number of minimal feasible solutions of a monotone system may be exponential in the size of the input, the efficiency of such generation is measured customarily in terms of the sizes of both the input and the output (see e.g. [19]). More precisely, we focus on the problem of generating a “next” element of $\mathcal{F}_{A,b,c}$, sequentially:

GEN(A, b, c, \mathcal{X}): *Given a monotone system (1) defined by (A, b, c) , and a subset $\mathcal{X} \subseteq \mathcal{F}_{A,b,c}$ of its minimal feasible vectors, either find a new minimal integral vector $x \in \mathcal{F}_{A,b,c} \setminus \mathcal{X}$, or show that no such vector exists, i.e., $\mathcal{X} = \mathcal{F}_{A,b,c}$.*

Clearly, the entire set $\mathcal{F}_{A,b,c}$ can be generated by initializing $\mathcal{X} = \emptyset$ and iteratively solving the above problem $|\mathcal{F}_{A,b,c}| + 1$ times. Accordingly, we say that the family $\mathcal{F}_{A,b,c}$ can be generated in *incremental q -time* if the problem GEN(A, b, c, \mathcal{X}) can be solved in $q(n, r, |\mathcal{X}|)$ operations, for every subset $\mathcal{X} \subseteq \mathcal{F}_{A,b,c}$. In particular, we say that $\mathcal{F}_{A,b,c}$ can be generated in *incremental polynomial* or *quasi-polynomial* time, if q is a polynomial or quasi-polynomial expression, respectively. These complexity notions are stated here for the unit-cost model of computation in which the $q(n, r, |\mathcal{X}|)$ bound applies to the number of comparisons (\leq, \geq) and arithmetic and rounding operations ($+, -, \times, /, \lfloor \cdot \rfloor$) required to solve problem GEN(A, b, c, \mathcal{X}) for *real* input (A, b, c) . All of our results in this paper will also be valid, with the same bound for the number of operations, for *rational* input (A, b, c) in the bit model of computation, where

the number of bits used in the operations depends linearly on the binary size of the input data.

Let us note here that the above definition for the complexity of sequential generation, based on $\text{GEN}(A, b, c, \mathcal{X})$, does indeed grasp the essence of sequential generation correctly. Namely, if there is any algorithm \mathbf{A} generating all the elements of $\mathcal{F}_{A,b,c}$ sequentially, say in the order $\mathcal{F}_{A,b,c} = \{x^1, x^2, \dots, x^K\}$ and $x^{K+1} = \text{STOP}$ such that $\{x^1, \dots, x^k\}$ is generated by $q(n, r, k-1)$ operations, for every $k = 1, \dots, K+1$, then using the same algorithm \mathbf{A} , problem $\text{GEN}(A, b, c, \mathcal{X})$ can also be solved in $O(q(n, r, |\mathcal{X}|))$ operations for every subset $\mathcal{X} \subseteq \mathcal{F}_{A,b,c}$.

The problem of generating the family $\mathcal{F}_{A,b,c}$ of all minimal feasible solutions to a monotone system (1) includes, as special cases, several well-known problems from the literature.

First of all, if A is integral and $c = +\infty$, the generation of $\mathcal{F}_{A,b,c}$ can be regarded as the computation of the Hilbert basis for the polynomial ideal which has $\{x \in \mathbb{Z}^n \mid Ax \geq b, x \geq 0\}$ as its Newton polyhedron.

If A is a binary matrix, $b = \mathbf{e}_r$ and $c = \mathbf{e}_n$, where \mathbf{e}_d denotes the vector of all ones of dimension d , then $\mathcal{F}_{A,b,c}$ is the set of (characteristic vectors of) all minimal transversals to the hypergraph defined by the rows of A . In this case, problem $\text{GEN}(A, b, c, \mathcal{X})$ turns into the well-known *hypergraph dualization problem*, that is the incremental enumeration of all minimal transversals (or equivalently, all maximal independent sets) of a given hypergraph (see e.g. [12]). For several special classes of hypergraphs $\text{GEN}(A, \mathbf{e}_r, \mathbf{e}_n, \mathcal{X})$ can be solved efficiently, including 2-monotonic, threshold, matroid, read-bounded, and acyclic hypergraphs (see [4, 8, 9, 13, 16, 21, 23, 24, 26, 27]), and most notably, when the hyperedges are all of size 2, that is when the hypergraph is the edge set of a graph (see [19, 20, 30]). The latter result has been extended to an incremental polynomial solution for all hypergraphs with bounded edge sizes (see e.g., [6, 12, 13]), in which case even an efficient parallel solution exists (see [5]). There is no incremental polynomial time algorithm known for the dualization of an arbitrary hypergraph. The best known result is an incremental *quasi-polynomial time* algorithm, proposed by Fredman and Khachiyan [14], which solves $\text{GEN}(A, \mathbf{e}_r, \mathbf{e}_n, \mathcal{X})$ in $O(nt) + t^{o(\log t)}$ time, where $t = \max\{r, |\mathcal{X}|\}$. Recent improvements on this procedure can be found in [15, 29].

If A is binary, $c = \mathbf{e}_n$ and $b = Ac - \mathbf{e}_r$, then the vectors in $\mathcal{F}_{A,b,c}$ correspond (in a one-to-one way) to the maximal matchings of the hypergraph, formed by the columns of A . It was shown in [20] that $\text{GEN}(A, b, c, \mathcal{X})$ can be solved in this case in polynomial time, providing thus an efficient generation of all maximal matchings. More generally, if A is binary, $c = \mathbf{e}_n$ and b is arbitrary, then the vectors in $\mathcal{F}_{A,b,c}$ are the characteristic vectors of the so-called *multiple transversals* of the hypergraph formed by the rows of A , and an incremental quasi-polynomial generation is provided in [7] for this case.

If $c = \mathbf{e}_n$, and $r = 1$, then (1) is also known as a *binary knapsack problem*, for which $\mathcal{F}_{A,b,c}$ can be generated in incremental polynomial time with an amortized

complexity of $O(n^2)$ time per element, as it was shown by Lawler, Lenstra and Rinnoy Kan [20]. Improving on this result, Uno [31] showed recently that in this special case $\mathcal{F}_{A,b,c}$ can be generated in $O(n)$ time per element, in the worst case.

More generally, Lawler, Lenstra and Rinnoy Kan considered the case of a non-negative matrix A with b arbitrary, and $c = \mathbf{e}_n$, and conjectured that no efficient generation of $\mathcal{F}_{A,b,\mathbf{e}_n}$ is possible in this case, unless $P=NP$.

As our main result, we shall show that problem $\text{GEN}(A, b, c, \mathcal{X})$ can in fact be solved in quasi-polynomial time for any monotone system (1), that is all minimal integer solutions of (1) can be generated in incremental quasi-polynomial time, both in the unit-cost and the bit models of computation.

Theorem 1 *Problem $\text{GEN}(A, b, c, \mathcal{X})$ can be solved in $t^{o(\log t)}$ time, where $t = \max\{n, r, |\mathcal{X}|\}$.*

This result implies that $\text{GEN}(A, b, c, \mathcal{X})$ cannot be NP-hard, unless all NP-complete problems can be solved in quasi-polynomial time. We mention in passing that if c is bounded and the number of non-zero coefficients per inequality in (1) is fixed, the results of [5] also imply that problem $\text{GEN}(A, b, c, \mathcal{X})$ can be efficiently solved in parallel.

2 Uniformly Dual Bounded Monotone Systems

Let us note first that, without any loss of generality, the finiteness of \mathcal{C} can always be assumed. Namely, defining $J^* = \{j \mid c_j = \infty\}$, $J_* = \{1, \dots, n\} \setminus J^*$, and

$$\bar{c}_j = \begin{cases} \max_{i: a_{ij} > 0} \left\lceil \frac{b_i - \sum_{k \in J_*} \min\{0, a_{ik}\} c_k}{a_{ij}} \right\rceil & \text{for } j \in J^*, \text{ and} \\ c_j & \text{for } j \in J_*, \end{cases} \quad (2)$$

we have $\mathcal{F}_{A,b,c} = \mathcal{F}_{A,b,\bar{c}}$. To see this equality, let us consider an arbitrary vector $x = (x_1, \dots, x_n) \in \mathcal{F}_{A,b,c}$ such that $x_j > 0$ for some $j \in J^*$. We claim that $x_j \leq \bar{c}_j$. For this, let us observe first that the restriction of A on J^* must be non-negative: $a_{ij} \geq 0$ for all $i \in \{1, \dots, r\}$ and $j \in J^*$, for otherwise the monotonicity of (1) could be violated by sufficiently increasing x_j , the j^{th} component of x . Thus, we have

$$\begin{aligned} a_{ij}x_j + \sum_{k \in J_*} \min\{0, a_{ik}\}c_k &\leq a_{ij}x_j + \sum_{k \in J_*} \min\{0, a_{ik}\}x_k \\ &\leq a_{ij}x_j + \sum_{k \in J_*} a_{ik}x_k \\ &\leq \sum_{k \in J^*} a_{ik}x_k + \sum_{k \in J_*} a_{ik}x_k = a^i x, \end{aligned}$$

since the restriction of A on J^* is non-negative, where a^i denotes the i^{th} row of A . Let us consider now the vector x' obtained from x by decreasing its j^{th} component by 1. Then $x' \in \mathcal{C}$, and by the minimality of x , x' is infeasible for (1). Hence, $b_i - a_{ij} \leq a^i x - a_{ij} = a^i x' < b_i$ for some $i \in \{1, \dots, r\}$, implying $a^i x < b_i + a_{ij}$. Thus, we can conclude that

$$a_{ij}x_j + \sum_{k \in J_*} \min\{0, a_{ik}\}c_k < b_i + a_{ij}$$

from which our claim follows.

Since the bounds in (2) are easy to compute, and since appending these bounds to (1) does not change the set $\mathcal{F}_{A,b,c}$ by the above observation, we shall assume in the sequel that all components of the non-negative vector c are finite, even though this may not be the case for the original system. This assumption does not entail any loss of generality and allows us to consider $\mathcal{F}_{A,b,c}$ as a system of integral vectors in a finite box.

Let us also note that the input monotone system (1) can be assumed to be feasible and non-trivial, that is $\mathcal{F}_{A,b,c} \neq \emptyset$ and $\mathcal{F}_{A,b,c} \neq \{0\}$. For a finite and non-negative c this is equivalent to $Ac \geq b$ and $0 \not\geq b$.

Generalizing systems of monotone inequalities, we shall consider arbitrary monotone subsets of the finite integral box \mathcal{C} . For a collection of integral vectors $\mathcal{A} \subseteq \mathcal{C}$ let us denote, respectively, by $\mathcal{A}^+ = \{x \in \mathcal{C} \mid x \geq a \text{ for some } a \in \mathcal{A}\}$ and $\mathcal{A}^- = \{x \in \mathcal{C} \mid x \leq a \text{ for some } a \in \mathcal{A}\}$ the *ideal* and *filter* generated by \mathcal{A} . Generalizing standard terminology of the theory of hypergraphs, elements in $\mathcal{C} \setminus \mathcal{A}^+$ will be called *independent* of \mathcal{A} , and let us denote by $\mathcal{I}(\mathcal{A})$ the set of all *maximal independent* elements of \mathcal{A} . Then, for a finite box \mathcal{C} we have:

$$\mathcal{A}^+ \cap \mathcal{I}(\mathcal{A})^- = \emptyset, \quad \text{and} \quad \mathcal{A}^+ \cup \mathcal{I}(\mathcal{A})^- = \mathcal{C}. \quad (3)$$

In particular, if $\mathcal{A} = \mathcal{F}_{A,b,c}$ is the set of all minimal feasible integral vectors for (1) and \mathcal{C} is finite, then the ideal $\mathcal{F}_{A,b,c}^+$ is the set of all feasible solutions of (1), while the filter $\mathcal{C} \setminus \mathcal{F}_{A,b,c}^+$ is the collection of all *infeasible* vectors. This filter is generated by the set $\mathcal{I}(\mathcal{F}_{A,b,c})$ consisting of all *maximal infeasible* integral vectors for (1):

$$\mathcal{C} \setminus \mathcal{F}_{A,b,c}^+ = \mathcal{I}(\mathcal{F}_{A,b,c})^- = \{x \in \mathcal{C} \mid Ax \not\geq b\} = \bigcup_{y \in \mathcal{I}(\mathcal{F}_{A,b,c})} \{y\}^-.$$

Let us return finally to the generation of minimal feasible solutions of a monotone system (1). Our approach for the solution of this problem will be based on the *joint generation* of the union $\mathcal{F}_{A,b,c} \cup \mathcal{I}(\mathcal{F}_{A,b,c})$. The following statement, ensuring the efficiency of such an approach, is instrumental in our proofs, and may also be of independent interest.

Theorem 2 *If the monotone system (1) is feasible, then for any non-empty subset $\mathcal{X} \subseteq \mathcal{F}_{A,b,c}$ we have*

$$|\mathcal{I}(\mathcal{X}) \cap \mathcal{I}(\mathcal{F}_{A,b,c})| \leq rn|\mathcal{X}|. \quad (4)$$

In particular, for $\mathcal{X} = \mathcal{F}_{A,b,c}$ this implies the inequality

$$|\mathcal{I}(\mathcal{F}_{A,b,c})| \leq rn|\mathcal{F}_{A,b,c}|.$$

Using the terminology introduced in [7], the above statement claims that $\mathcal{F}_{A,b,c}$ for a monotone system (1) is *uniformly dual bounded*. This property, more precisely inequality (4), guarantees that if we manage to generate the elements of the union $\mathcal{F}_{A,b,c} \cup \mathcal{I}(\mathcal{F}_{A,b,c})$ one-by-one, in some efficient way, then at any given moment during this generation the number of produced maximal infeasible vectors is polynomially limited by the number of obtained minimal feasible vectors. Thus, by simply disregarding the maximal infeasible vectors in this process, we get an efficient algorithm for the generation of minimal feasible vectors.

Let us remark here that such an approach would certainly be too wasteful for the generation of maximal infeasible solutions, as the following example shows. Let us consider the monotone system (A, b, c) consisting of the following r inequalities $x_1 + x_2 \geq 1, x_3 + x_4 \geq 1, \dots, x_{2r-1} + x_{2r} \geq 1$ in $n = 2r$ variables, with $c = \mathbf{e}_{2r}$. This system has 2^r minimal feasible binary vectors and only r maximal infeasible ones, i.e.,

$$|\mathcal{F}_{A,b,c}| \geq 2^{|\mathcal{I}(\mathcal{F}_{A,b,c})|}.$$

Thus, $|\mathcal{F}_{A,b,c}|$ cannot be bounded by a polynomial in r, n , and $|\mathcal{I}(\mathcal{F}_{A,b,c})|$, in general. Therefore, in the joint generation process we may get unlucky in the worst case, and get first exponentially many minimal feasible vectors, before obtaining the first maximal infeasible one.

In fact, not only such a joint generation approach is inefficient for the generation of maximal infeasible vectors, but more generally, a result analogous to that of Theorem 1 is highly unlikely to hold.

Proposition 1 *Let us consider a monotone system (1), where A is an $r \times n$ binary matrix, $c = \mathbf{e}_n$, and all the r components of b , but one, are equal to 1. Further, let $\mathcal{X} \subseteq \mathcal{I}(\mathcal{F}_{A,b,c})$. Then, it is NP-complete to decide if $\mathcal{I}(\mathcal{F}_{A,b,c}) \setminus \mathcal{X} \neq \emptyset$.*

Proof. We can use arguments analogous to those of [22]. Consider the well-known NP-hard problem of determining whether a given graph $G = (V, E)$ contains an independent set of size t , where $t \geq 2$ is a given threshold. Let us introduce $|V|$ binary variables $x_v, v \in V$, and write $|E|$ inequalities: $x_v + x_{v'} \geq 1$, one for every edge $e = (v, v') \in E$, followed by the inequality: $\sum_{v \in V} x_v \geq |V| - t$. It is easily seen that if x is the characteristic vector of an edge $e \in E$, then $\mathbf{e} - x$ is a maximal infeasible binary vector for the above system of inequalities. Deciding

whether there are other such vectors is equivalent to determining whether the given graph G has an independent set of size at least t . \square

According to our earlier remark, Proposition 1 implies that no incrementally efficient generation of maximal infeasible vectors is possible, regardless of the order of their generation, unless all NP-hard problems can be solved with the same efficiency. On the other hand, it is easy to see that if the right hand side vector b is bounded by a constant and the matrix A is integral and non-negative then listing all maximal infeasible binary vectors of (1) can be done in polynomial time.

Note also that the enumeration of all maximal infeasible binary vectors to a system of non-negative linear inequalities $a^1x \geq b_1, \dots, a^rx \geq b_r$, with integer coefficients, is equivalent to the enumeration of all maximal solutions satisfying at least one of r knapsack inequalities, $a^1x \leq b_1 - 1, \dots, a^rx \leq b_r - 1$. An exponential algorithm for the latter problem and its relation to the facet enumeration for some 0, 1-polytopes [1] are discussed in [20].

Let us remark finally that the bounds in Theorem 2 are sharp for $r = 1$: for instance, the single inequality $x_1 + \dots + x_n \geq n$ in binary variables has only one maximal infeasible vector and exactly n minimal feasible ones. For larger values of r , these bounds are accurate up to a factor poly-logarithmic in r . To see this, let us consider the input (A, b, c) consisting of $r = 2^k$ inequalities of the form

$$x_{i_1} + x_{i_2} + \dots + x_{i_k} \geq 1, \quad i_1 \in \{1, 2\}, \quad i_2 \in \{3, 4\}, \dots, \quad i_k \in \{2k-1, 2k\},$$

in $n = 2k$ variables, where $x = (x_1, \dots, x_n) \in \mathcal{C} = \{x \in \mathbb{Z}^n \mid 0 \leq x \leq c\}$. For any positive integer vector c , we have 2^k maximal infeasible integer vectors and only k minimal feasible integer vectors in this system, that is

$$|\mathcal{I}(\mathcal{F}_{A,b,c})| = \frac{rn}{2(\log r)^2} |\mathcal{F}_{A,b,c}|.$$

3 Joint Generation and Dualization

As we have indicated, maximal infeasible vectors play a crucial role in our analysis. More precisely, for $\mathcal{X} \subseteq \mathcal{F}_{A,b,c}$ we shall use the equivalence

$$\mathcal{X} = \mathcal{F}_{A,b,c} \iff \mathcal{X}^+ \cup \mathcal{I}(\mathcal{F}_{A,b,c})^- = \mathcal{C}$$

to verify the “completeness” of \mathcal{X} . In other words, our generation of minimal feasible solutions will in fact involve all maximal infeasible solutions, as well. For this reason, we shall consider the *joint generation* of minimal feasible and maximal infeasible vectors by repeatedly solving the following problem:

GEN($A, b, c, \mathcal{X}, \mathcal{Y}$): Given a monotone system (1) defined by (A, b, c) , and subsets $\mathcal{X} \subseteq \mathcal{F}_{A,b,c}$ and $\mathcal{Y} \subseteq \mathcal{I}(\mathcal{F}_{A,b,c})$, either find a new integral vector $x \in (\mathcal{F}_{A,b,c} \cup \mathcal{I}(\mathcal{F}_{A,b,c})) \setminus (\mathcal{X} \cup \mathcal{Y})$, or show that no such vector exists, i.e., $\mathcal{X} = \mathcal{F}_{A,b,c}$ and $\mathcal{Y} = \mathcal{I}(\mathcal{F}_{A,b,c})$.

Extending the results of [3] and [17] from the Boolean case $\mathcal{C} = \{0, 1\}^n$ to arbitrary integer boxes, we show that for a monotone system, $\text{GEN}(A, b, c, \mathcal{X}, \mathcal{Y})$ is polynomially equivalent to the following *dualization* problem:

DUAL($\mathcal{C}, \mathcal{A}, \mathcal{B}$): Given a family of vectors $\mathcal{A} \subseteq \mathcal{C}$, and a subset $\mathcal{B} \subseteq \mathcal{I}(\mathcal{A})$ of its maximal independent vectors, either find a new maximal independent vector $x \in \mathcal{I}(\mathcal{A}) \setminus \mathcal{B}$, or prove that no such vector exists, i.e., $\mathcal{B} = \mathcal{I}(\mathcal{A})$.

More specifically, we present a simple algorithm which, given a proper subset \mathcal{Z} of the (disjoint) union $\mathcal{F}_{A,b,c} \cup \mathcal{I}(\mathcal{F}_{A,b,c})$, finds a new vector in the union by performing $\text{poly}(n, r)$ comparisons and operations $+$, $-$, \times , $/$, $\lfloor \cdot \rfloor$ and by solving, if necessary, problem $\text{DUAL}(\mathcal{C}, \mathcal{A}, \mathcal{B})$ for $\mathcal{A} = \mathcal{Z} \cap \mathcal{F}_{A,b,c}$ and $\mathcal{B} = \mathcal{Z} \cap \mathcal{I}(\mathcal{F}_{A,b,c})$.

Next, by invoking Theorem 2, which we shall prove separately in Section 5, we can argue that since the number of maximal infeasible integer vectors for (1) is relatively small, we can efficiently generate all elements of $\mathcal{F}_{A,b,c}$ by generating all vectors in the union $\mathcal{F}_{A,b,c} \cup \mathcal{I}(\mathcal{F}_{A,b,c})$ and discarding the elements belonging to $\mathcal{I}(\mathcal{F}_{A,b,c})$. This leads to the following result:

Theorem 3 *Problem $\text{GEN}(A, b, c, \mathcal{X})$ can be reduced in strongly polynomial time to $\text{DUAL}(\mathcal{C}, \mathcal{A}, \mathcal{B})$.*

To complete the proof of Theorem 1, we need to show that $\text{DUAL}(\mathcal{C}, \mathcal{A}, \mathcal{B})$, the *dualization problem over integer boxes*, can indeed be solved efficiently. As mentioned earlier, for $\mathcal{C} = \{0, 1\}^n$, problem $\text{DUAL}(\mathcal{C}, \mathcal{A}, \mathcal{B})$ turns into the well-known hypergraph dualization problem. In Section 7 we extend the hypergraph dualization algorithms of [14] to problem $\text{DUAL}(\mathcal{C}, \mathcal{A}, \mathcal{B})$ and show that the latter problem can be solved in quasi-polynomial time:

Theorem 4 *Given two sets \mathcal{A} , and $\mathcal{B} \subseteq \mathcal{I}(\mathcal{A})$ in an n -dimensional box $\mathcal{C} = \{x \in \mathbb{Z}^n \mid 0 \leq x \leq c\}$, problem $\text{DUAL}(\mathcal{C}, \mathcal{A}, \mathcal{B})$ can be solved in $\text{poly}(n) + m^{o(\log m)}$ time, where $m = |\mathcal{A}| + |\mathcal{B}|$.*

As before, Theorem 4 is stated in the unit-cost model of computation: the bound of the theorem applies to the number of comparisons between components of \mathcal{A} and \mathcal{B} , required to solve the dualization problem. Other applications of the dualization problem on boxes can be found in [2, 10, 25].

Clearly, Theorem 1 follows from Theorems 3 and 4. The special cases of Theorems 2 and 3 for Boolean systems can be found in [7].

4 Bounded Number of Inequalities

Even though generating all maximal infeasible vectors for (1) is NP-hard by Proposition 1, this problem can be solved efficiently if the number of inequalities in (1) is a fixed constant. Specifically, for $r = \text{const}$, $|\mathcal{F}_{A,b,c}|$ can be bounded by a polynomial in n and $|\mathcal{I}(\mathcal{F}_{A,b,c})|$ and consequently, all elements of $\mathcal{I}(\mathcal{F}_{A,b,c})$ can be generated in quasi-polynomial time. In fact, for $r = \text{const}$ the problem of generating $\mathcal{I}(\mathcal{F}_{A,b,c})$ as well as the problem of generating $\mathcal{F}_{A,b,c}$ can be solved separately in incremental polynomial time.

For a subset $\mathcal{Y} \subseteq \mathcal{C}$, let $\mathcal{I}^{-1}(\mathcal{Y})$ denote the set of all minimal integral vectors of the ideal $\mathcal{C} \setminus \mathcal{Y}^-$.

Theorem 5 *Suppose that the monotone system (1) is nontrivial, i.e., $0 \notin \mathcal{F}_{A,b,c}$. Then for any non-empty subset $\mathcal{Y} \subseteq \mathcal{I}(\mathcal{F}_{A,b,c})$ we have*

$$|\mathcal{I}^{-1}(\mathcal{Y}) \cap \mathcal{F}_{A,b,c}| \leq \left(n|\mathcal{Y}|\right)^r. \quad (5)$$

In particular, for $\mathcal{Y} = \mathcal{I}(\mathcal{F}_{A,b,c})$ we get

$$|\mathcal{F}_{A,b,c}| \leq \left(n|\mathcal{I}(\mathcal{F}_{A,b,c})|\right)^r.$$

Generalizing the results of [20, 31] for knapsack problems, we can show that both minimal feasible and maximal infeasible solutions can be generated efficiently, if the number of inequalities in the input monotone system is fixed.

Theorem 6 *If the number of inequalities in (1) is bounded, then both $\mathcal{F}_{A,b,c}$ and $\mathcal{I}(\mathcal{F}_{A,b,c})$ can be generated in incremental polynomial time.*

The remainder of the paper consists of the proofs of Theorems 2, 3, 4, 5, and 6 in Sections 5, 6, 7, 8 and 9, respectively.

5 Proof of Theorem 2

We first need some notation and definitions.

Let $\mathcal{C} = \{x \in \mathbb{Z}^n \mid 0 \leq x \leq c\}$ be a box and let $f : \mathcal{C} \rightarrow \{0, 1\}$ be a discrete binary function. The function f is called *monotone* if $f(x) \geq f(y)$ whenever $x \geq y$ and $x, y \in \mathcal{C}$. We denote by $T(f)$ and $F(f)$ the sets of all true and all false vectors of f , i.e.,

$$T(f) = \{x \in \mathcal{C} \mid f(x) = 1\} = (\text{MIN}[f])^+, \quad F(f) = \{x \in \mathcal{C} \mid f(x) = 0\} = (\text{MAX}[f])^-,$$

where $\text{MIN}[f]$ and $\text{MAX}[f]$ are the sets of all minimal true and all maximal false vectors of f , respectively.

Let $\sigma \in \mathbb{S}_n$ be a permutation of the coordinates and let x, y be two n -vectors. We say that y is a *left-shift* of x and write $y \succeq_\sigma x$ if the inequalities

$$\sum_{j=1}^k y_{\sigma_j} \geq \sum_{j=1}^k x_{\sigma_j}$$

hold for all $k = 1, \dots, n$. A discrete binary function $f : \mathcal{C} \rightarrow \{0, 1\}$ is called *2-monotonic with respect to σ* if $f(y) \geq f(x)$ whenever $y \succeq_\sigma x$ and $x, y \in \mathcal{C}$. Clearly, $y \geq x$ implies $y \succeq_\sigma x$ for any $\sigma \in \mathbb{S}_n$, so that any 2-monotonic function is monotone.

The function f will be called *regular* if it is 2-monotonic with respect to the identity permutation $\sigma = (1, 2, \dots, n)$. Any 2-monotonic function can be transformed into a regular one by appropriately re-indexing its variables. To simplify notation, we shall state Lemma 1 below for regular functions, i.e., we fix $\sigma = (1, 2, \dots, n)$ in this lemma.

For a given subset $\mathcal{A} \subseteq \mathcal{C}$ let us denote by \mathcal{A}^* all the vectors which are left-shifts of some vectors of \mathcal{A} , i.e., $\mathcal{A}^* = \{y \in \mathcal{C} \mid y \succeq x \text{ for some } x \in \mathcal{A}\}$. Clearly, $T(f) = (\text{MIN}[f])^*$ for a regular function f (in fact, the subfamily of *right-most* vectors of $\text{MIN}[f]$ would be enough to use here).

Given monotone discrete functions f and g , we call g a *regular majorant* of f , if $g(x) \geq f(x)$ for all $x \in \mathcal{C}$, and g is regular. Clearly, $T(g) \supseteq (\text{MIN}[f])^*$ must hold in this case, and the discrete function h defined by $T(h) = (\text{MIN}[f])^*$ is the unique minimal regular majorant of f .

For a vector $x \in \mathcal{C}$, and for an index $1 \leq k \leq n$, let the vectors $x[k]$ and $x[k]$ be defined by

$$x_j[k] = \begin{cases} x_j & \text{for } j \leq k, \\ 0 & \text{otherwise,} \end{cases}$$

and

$$x_j[k] = \begin{cases} x_j & \text{for } j \geq k, \\ 0 & \text{otherwise.} \end{cases}$$

Let us denote by \mathbf{e}^j the j^{th} unit vector in \mathbb{R}^n , for $j = 1, \dots, n$, and let $p(x)$ denote the number of positive components of the vector $x \in \mathcal{C}$.

Lemma 1 *Given a monotone discrete binary function $f : \mathcal{C} \rightarrow \{0, 1\}$ such that $f \not\equiv 0$, and a regular majorant $g \geq f$, we have the inequality*

$$|F(g) \cap \text{MAX}[f]| \leq \sum_{x \in \text{MIN}[f]} p(x). \quad (6)$$

Proof. Let us denote by h the unique minimal regular majorant of f . Then we have $F(g) \cap \text{MAX}[f] \subseteq F(h) \cap \text{MAX}[f]$, and hence it is enough to show the statement for $g = h$, i.e. when $T(g) = (\text{MIN}[f])^*$.

For a vector $y \in \mathcal{C} \setminus \{c\}$ let us denote by $l = l_y$ the index of the last component which is less than c_l , i.e., $l = \max\{j \mid y_j < c_j\} \in \{1, \dots, n\}$. We claim that for every $y \in F(h) \cap \text{MAX}[f]$ there exists an $x \in \text{MIN}[f]$ such that

$$y = x[l-1] + (x_l - 1)\mathbf{e}^l + c[l+1], \quad (7)$$

where $l = l_y$. To see this claim, first observe that $y \neq c$ because $y \in F(f)$ and $f \not\equiv 0$. Second, for any j with $y_j < c_j$ we have $y + \mathbf{e}^j \in T(f)$, by the definition of a maximal false point. Hence there exists a minimal true-vector $x \in \text{MIN}[f]$ such that $x \leq y + \mathbf{e}^l$ for $l = l_y$. We must have $x(l-1) = y(l-1)$, since if $x_i < y_i$ for some $i < l$, then $y \geq x + \mathbf{e}^i - \mathbf{e}^l \succeq x$ would hold, i.e. $y \succeq x$ would follow, implying $y \in (\text{MIN}[f])^*$ and yielding a contradiction with $y \in F(h) = \mathcal{C} \setminus (\text{MIN}[f])^*$. Finally, the definition of $l = l_y$ implies that $y[l+1] = c[l+1]$. Hence, our claim and the equality (7) follow.

The above claim implies that

$$F(h) \cap \text{MAX}[f] \subseteq \{x(l-1) + (x_l - 1)\mathbf{e}^l + c[l+1] \mid x \in \text{MIN}[f], x_l > 0\},$$

and hence (6) and thus the lemma follow. \square

Lemma 2 *Let $f : \mathcal{C} \rightarrow \{0, 1\}$ be a monotone discrete binary function such that $f \not\equiv 0$ and*

$$x \in T(f) \Rightarrow \alpha x \stackrel{\text{def}}{=} \alpha_1 x_1 + \dots + \alpha_n x_n \geq \beta, \quad (8)$$

where $\alpha = (\alpha_1, \dots, \alpha_n)$ is a given real vector and β is a real threshold. Then

$$|\{x \in \mathcal{C} \mid \alpha x < \beta\} \cap \text{MAX}[f]| \leq \sum_{x \in \text{MIN}[f]} p(x).$$

Proof. Suppose that some of the weights $\alpha_1, \dots, \alpha_n$ are negative, say $\alpha_1 < 0, \dots, \alpha_k < 0$ and $\alpha[k+1] \geq 0$. Since $\alpha x \geq \beta$ for any $x \in T(f)$ and since f is monotone, we have $x \in T(f) \Rightarrow \alpha[k+1]x \geq \beta - \alpha[k]c[k]$. By the negativity of the weights $\alpha_1, \dots, \alpha_k$, we also have $\{x \in \mathcal{C} \mid \alpha x < \beta\} \subseteq \{x \in \mathcal{C} \mid \alpha[k+1]x < \beta - \alpha[k]c[k]\}$. Hence it suffices to prove the lemma for the non-negative weight vector $\alpha[k+1]$ and the threshold $\beta - \alpha[k]c[k]$. In other words, we can assume without loss of generality that the original weight vector α is non-negative.

Let $\sigma \in \mathbb{S}_n$ be a permutation such that $\alpha_{\sigma_1} \geq \alpha_{\sigma_2} \geq \dots \geq \alpha_{\sigma_n} \geq 0$. Then the threshold function

$$g(x) = \begin{cases} 1 & \text{if } \alpha x \geq \beta, \\ 0 & \text{otherwise.} \end{cases}$$

is 2-monotonic with respect to σ . By (8), we have $g \geq f$ for all $x \in \mathcal{C}$, i.e., g is a majorant of f . In addition, $F(g) = \{x \in \mathcal{C} \mid \alpha x < \beta\}$, and hence Lemma 2 follows from Lemma 1. \square

Proof of Theorem 2. We are now ready to show inequality (4) and finish the proof of Theorem 2. For this we shall prove in fact the following stronger inequality:

$$|\mathcal{I}(\mathcal{X}) \cap \mathcal{I}(\mathcal{F}_{A,b,c})| \leq r \sum_{x \in \mathcal{X}} p(x). \quad (9)$$

Given a non-empty set $\mathcal{X} \subseteq \mathcal{F}_{A,b,c}$, consider the monotone discrete function $f : \mathcal{C} \rightarrow \{0, 1\}$ defined by the condition $\text{MIN}[f] = \mathcal{X}$. Since (1) is monotone, any true vector of f also satisfies (1):

$$x \in T(f) \Rightarrow a_{k1}x_1 + \dots + a_{kn}x_n \geq b_k$$

for all $k = 1, \dots, r$. In addition, $f \not\equiv 0$ because $\mathcal{X} \neq \emptyset$. Thus, by Lemma 2 we have the inequalities

$$|\{x \mid a_{k1}x_1 + \dots + a_{kn}x_n < b_k\} \cap \text{MAX}[f]| \leq \sum_{x \in \mathcal{X}} p(x) \quad (10)$$

for each $k = 1, \dots, r$. Now, from $\text{MAX}[f] = \mathcal{I}(\mathcal{X})$ we deduce that

$$\mathcal{I}(\mathcal{F}_{A,b,c}) \cap \mathcal{I}(\mathcal{X}) \subseteq \bigcup_{k=1}^r \{x \mid a_{k1}x_1 + \dots + a_{kn}x_n < b_k\} \cap \text{MAX}[f],$$

and hence (9) follows by (10), implying thus (4) and completing the proof of the theorem. \square

6 Generating Minimal Feasible Solutions

As mentioned in the Introduction, the proof of Theorem 3 has two ingredients. First, we show that given a monotone system (1), the sets $\mathcal{I}(\mathcal{F}_{A,b,c})$ and $\mathcal{F}_{A,b,c}$ can be jointly enumerated by iteratively solving the dualization problem $\text{DUAL}(\mathcal{C}, \mathcal{A}, \mathcal{B})$. Second, we argue that due to Theorem 2, the number of maximal infeasible vectors for (1) is relatively small, and hence the generation of $\mathcal{F}_{A,b,c}$ polynomially reduces to the joint generation of $\mathcal{I}(\mathcal{F}_{A,b,c})$ and $\mathcal{F}_{A,b,c}$.

6.1 Joint generation of dual subsets in an integral box

As it was observed in [3, 17], for $c = \mathbf{e}_n$ problem $\text{GEN}(A, b, c, \mathcal{A}, \mathcal{B})$ can be reduced in polynomial time to the dualization problem $\text{DUAL}(\mathcal{C}, \mathcal{A}, \mathcal{B})$. Extending this observation, we prove it here for any integral vector c .

Proposition 2 *Problem $\text{GEN}(A, b, c, \mathcal{A}, \mathcal{B})$ can be solved in $\text{poly}(n, |\mathcal{A}|, |\mathcal{B}|) + T^*$ time, where T^* denotes the time required to solve problem $\text{DUAL}(\mathcal{C}, \mathcal{A}, \mathcal{B})$.*

Proof. In fact, we can prove a more general statement. Let us consider an arbitrary *antichain* $\mathcal{F} \subseteq \mathcal{C}$ (i.e., a family \mathcal{F} of vectors such that $x \not\leq y$ for any two distinct elements $x, y \in \mathcal{F}$), assume that there is a polynomial time membership oracle $\mathfrak{D}(\mathcal{F}^+)$ for the monotone set \mathcal{F}^+ , and consider the following problem:

GEN($\mathfrak{D}(\mathcal{F}^+), \mathcal{A}, \mathcal{B}$): *Given subsets $\mathcal{A} \subseteq \mathcal{F}$ and $\mathcal{B} \subseteq \mathcal{I}(\mathcal{F})$, either find a new element $x \in (\mathcal{F} \cup \mathcal{I}(\mathcal{F})) \setminus (\mathcal{A} \cup \mathcal{B})$, or show that no such vector exists, i.e., $\mathcal{A} = \mathcal{F}$ and $\mathcal{B} = \mathcal{I}(\mathcal{F})$.*

We can show that this more general problem also reduces in polynomial time to $\text{DUAL}(\mathcal{C}, \mathcal{A}, \mathcal{B})$.

Our proof uses two subroutines, the first of which takes as input a vector $x \in \mathcal{F}^+$ and returns a minimal vector x^* in $\mathcal{F}^+ \cap \{x\}^-$. Such a vector $x^* = \min_{\mathcal{F}}(x)$ can, for instance, be computed by coordinate descent:

$$\begin{aligned} x_1^* &\leftarrow \min\{y_1 \mid (y_1, y_2, \dots, y_{n-1}, y_n) \in \mathcal{F}^+ \cap \{x\}^-\}, \\ x_2^* &\leftarrow \min\{y_2 \mid (x_1^*, y_2, \dots, y_{n-1}, y_n) \in \mathcal{F}^+ \cap \{x\}^-\}, \\ &\dots \\ x_n^* &\leftarrow \min\{y_n \mid (x_1^*, x_2^*, \dots, x_{n-1}^*, y_n) \in \mathcal{F}^+ \cap \{x\}^-\}. \end{aligned}$$

The second subroutine is to compute, for a given vector $x \in \mathcal{I}(\mathcal{F})^-$, a maximal vector $x^* \in \mathcal{I}(\mathcal{F})^- \cap x^+$. Similarly, this problem can be done by coordinate descent. Note that each of the n coordinate steps in the above procedures can be reduced via binary search to at most $\log(\|c\|_\infty + 1)$ membership queries for the monotone family \mathcal{F}^+ . Though this bound depends on the size of the box \mathcal{C} , in general, in our case when $\mathcal{F} = \mathcal{F}_{A,b,c}$ is the set of minimal integer solutions for an explicitly given monotone system (1), both of the above coordinate descends can be clearly performed in $O(nr)$ comparisons (\leq, \geq) and arithmetic operations ($+, -, \times, /, \lfloor \rfloor$), regardless of the box size.

Using these subroutines, our proof of Proposition 2 can now be completed by the following algorithm.

Algorithm \mathcal{J}

Input: An oracle $\mathcal{D}(\mathcal{F}^+)$ and subsets $\mathcal{A} \subseteq \mathcal{F}$ and $\mathcal{B} \subseteq \mathcal{I}(\mathcal{F})$.

Step 1. Check whether $\mathcal{B} \subseteq \mathcal{I}(\mathcal{A})$. Since $\mathcal{B} \subseteq \mathcal{I}(\mathcal{F})$ and $\mathcal{A} \subseteq \mathcal{F}$, each vector $x \in \mathcal{B}$ is independent of \mathcal{A} and we only have to check the maximality of x for $\mathcal{I}(\mathcal{A})$. In other words, we have to check whether or not $x + \mathbf{e}^j \geq y$ for some unit vector \mathbf{e}^j , $j \in \{1, \dots, n\}$ and some vector $y \in \mathcal{A}$. Since both \mathcal{A} and \mathcal{B} are explicitly given, this can be done in $\text{poly}(n, |\mathcal{A}|, |\mathcal{B}|)$ comparisons. If there is an $x \in \mathcal{B} \setminus \mathcal{I}(\mathcal{A})$, then $x \notin \mathcal{F}^+$ because $x \in \mathcal{B} \subseteq \mathcal{I}(\mathcal{F})$. This and the inclusion $\mathcal{A} \subseteq \mathcal{F}$ imply that $x \notin \mathcal{A}^+$. Since $x \notin \mathcal{I}(\mathcal{A})$, we can find a coordinate $j \in \{1, \dots, n\}$ for which $y = x + \mathbf{e}^j \notin \mathcal{A}^+$. By the maximality of x in $\mathcal{C} \setminus \mathcal{F}^+$, y belongs to \mathcal{F}^+ . Now using the first subroutine and letting $y^* = \min_{\mathcal{F}}(y)$ we can conclude that $y^* \in \mathcal{F} \setminus \mathcal{A}$, i.e., y^* is a new minimal integral vector in \mathcal{F} . Otherwise, if $\mathcal{B} \subseteq \mathcal{I}(\mathcal{A})$, we continue with the next step.

Step 2. Similar to the previous step, we check whether $\mathcal{A} \subseteq \mathcal{I}^{-1}(\mathcal{B})$, where $\mathcal{I}^{-1}(\mathcal{B})$ denotes the set of integral vectors which are minimal in $\mathcal{C} \setminus \mathcal{B}^-$. If \mathcal{A} contains an element that is not minimal in $\mathcal{C} \setminus \mathcal{B}^-$, we can find a new vector in $\mathcal{I}(\mathcal{F}) \setminus \mathcal{B}$ and stop. Otherwise we continue with the next step.

Step 3. In this case we have $\mathcal{B} \subseteq \mathcal{I}(\mathcal{A})$ and $\mathcal{A} \subseteq \mathcal{I}^{-1}(\mathcal{B})$, and thus the following equivalence holds:

$$(\mathcal{A}, \mathcal{B}) = (\mathcal{F}, \mathcal{I}(\mathcal{F})) \quad \Leftrightarrow \quad \mathcal{B} = \mathcal{I}(\mathcal{A}).$$

To see this, assume that $\mathcal{B} = \mathcal{I}(\mathcal{A})$, and suppose on the contrary that there is an $x \in \mathcal{F} \setminus \mathcal{A}$. Since $x \notin \mathcal{A} = \mathcal{I}^{-1}(\mathcal{B})$ and $x \notin \mathcal{B}^- \subseteq \mathcal{I}(\mathcal{F})^-$, there must exist a $y \in \mathcal{I}^{-1}(\mathcal{B}) = \mathcal{A} \subseteq \mathcal{F}$ such that $y \leq x$. Hence we get two distinct elements $x, y \in \mathcal{F}$ such that $y \leq x$, which contradicts the definition of \mathcal{F} . The existence of an $x \in \mathcal{I}(\mathcal{F}) \setminus \mathcal{B}$ leads to a similar contradiction.

To check the stopping criterion $\mathcal{B} = \mathcal{I}(\mathcal{A})$, we solve problem $\text{DUAL}(\mathcal{C}, \mathcal{A}, \mathcal{B})$. If $\mathcal{B} \neq \mathcal{I}(\mathcal{A})$, we obtain a new point $x \in \mathcal{I}(\mathcal{A}) \setminus \mathcal{B}$. By (3), either $x \in \mathcal{F}^+$, or $x \in \mathcal{I}(\mathcal{F})^-$ and we can decide which of these two cases holds by asking the oracle $\mathfrak{D}(\mathcal{F}^+)$, or in our special case by checking the feasibility of x for (1). In the first case, we conclude that $x^* = \min_{\mathcal{F}}(x)$ is a new vector in $\mathcal{F} \setminus \mathcal{A}$. In the second case, we can extend $\mathcal{I}(\mathcal{F}) \setminus \mathcal{B}$ by using the second subroutine to compute a maximal vector in $x^+ \cap \mathcal{I}(\mathcal{F})^-$. \square

As we noted, the above procedure can be used for any antichain $\mathcal{F} \subseteq \mathcal{C}$ defined by a polynomial-time membership oracle for \mathcal{F}^+ , since the coordinate descend subroutines used by the algorithm can always be implemented in $n \log(\|c\|_\infty + 1)$ membership tests. Accordingly, Proposition 2 also holds for an arbitrary antichain defined by a polynomial-time membership oracle, provided that the polynomial term in the proposition includes a multiplicative factor of $\log(\|c\|_\infty + 1)$.

6.2 Uniformly dual-bounded antichains

Let $\mathcal{F} \subseteq \mathcal{C}$ be an antichain defined by a polynomial-time membership oracle $\mathfrak{D}(\mathcal{F}^+)$ for \mathcal{F}^+ . Given an input description \mathcal{D} of \mathcal{F} and a vector $x \in \mathcal{C}$, such an oracle checks the membership of x in \mathcal{F}^+ in time bounded by a polynomial in the size of x and the length $|\mathcal{D}|$ of the input description of \mathcal{F} . For instance, the antichain $\mathcal{F}_{A,b,c}$ of minimal feasible integer vectors for (1) is defined by the triple $\mathcal{D} = (A, b, c)$, and the membership test for a given $x \in \mathcal{C}$ simply checks if $Ax \geq b$. The generation problem can thus be considered more generally for arbitrary antichains:

GEN($\mathfrak{D}(\mathcal{F}^+), \mathcal{X}$): *Given an antichain $\mathcal{F} \subseteq \mathcal{C}$ defined by a polynomial time membership oracle $\mathfrak{D}(\mathcal{F}^+)$, and a subset $\mathcal{X} \subseteq \mathcal{F}$, either find a new vector $x \in \mathcal{F} \setminus \mathcal{X}$, or show that no such vector exists, i.e., $\mathcal{X} = \mathcal{F}$.*

Extending the notion of dual-bounded hypergraphs defined in [7], we say that an antichain $\mathcal{F} \subseteq \mathcal{C}$ with an input description \mathcal{D} is *uniformly dual-bounded* if there exists a polynomial $q(x, y)$ such that

$$|\mathcal{I}(\mathcal{F}) \cap \mathcal{I}(\mathcal{X})| \leq q(|\mathcal{D}|, |\mathcal{X}|),$$

for any nonempty subset $\mathcal{X} \subseteq \mathcal{F}$ (see [7] for further details and examples). As we show below, the uniform dual-boundedness of an antichain is a sufficient condition to be able to reduce polynomially generation to dualization.

Proposition 3 *Suppose \mathcal{F} is uniformly dual-bounded and defined by a polynomial-time membership oracle $\mathfrak{D}(\mathcal{F}^+)$ for \mathcal{F}^+ . Then problem $GEN(\mathfrak{D}(\mathcal{F}^+), \mathcal{X})$ is polynomial time reducible to at most $q(|\mathcal{D}|, |\mathcal{X}|)+1$ instances of problem $DUAL(\mathcal{C}, \mathcal{A}, \mathcal{B})$.*

Proof. Given a set $\mathcal{X} \subseteq \mathcal{F}$, we repeatedly run Algorithm \mathcal{J} , starting with $\mathcal{A} = \mathcal{X}$ and $\mathcal{B} = \emptyset$, until it either produces a new element in $\mathcal{F} \setminus \mathcal{X}$ or proves that $\mathcal{X} = \mathcal{F}$ by generating the entire family $\mathcal{I}(\mathcal{F})$. By Step 1, either $\mathcal{B} \subseteq \mathcal{I}(\mathcal{X})$ is maintained during the execution of the algorithm, or a new element $x \in \mathcal{F}$ can be found. Thus, as long as Algorithm \mathcal{J} outputs elements of $\mathcal{I}(\mathcal{F})$, these elements also belong to $\mathcal{I}(\mathcal{X})$, and hence the total number of such elements does not exceed $q(|\mathcal{D}|, |\mathcal{X}|)$. \square

Proof of Theorem 3. By Theorem 2, the antichain $\mathcal{F} = \mathcal{F}_{A,b,c}$ of minimal integer solutions to the monotone system (1), for which we have $\mathcal{D} = (A, b, c)$, is uniformly dual-bounded with $q(|\mathcal{D}|, |\mathcal{X}|) = rn|\mathcal{X}|$. Furthermore, the reduction of Proposition 3 is strongly-polynomial. For this reason, Theorem 3 follows from Theorem 2 and Proposition 3. \square

7 Dualization in Products of Chains

In this section, we prove Theorem 4. Let $\mathcal{C} \stackrel{\text{def}}{=} \mathcal{C}_1 \times \dots \times \mathcal{C}_n$ be an integer box defined by the product of n chains $\mathcal{C}_i = [l_i : u_i]$ where $l_i, u_i \in \mathbb{Z}$ are, respectively, the lower and upper bounds of chain \mathcal{C}_i . For given antichains $\mathcal{A} \subseteq \mathcal{C}$ and $\mathcal{B} \subseteq \mathcal{I}(\mathcal{A})$, we say that \mathcal{B} is *dual to* \mathcal{A} if $\mathcal{B} = \mathcal{I}(\mathcal{A})$, i.e., \mathcal{B} contains all the maximal elements of $\mathcal{C} \setminus \mathcal{A}^+$. If \mathcal{C} is the unit cube, we obtain the familiar notion of dual hypergraphs, where $\mathcal{I}(\mathcal{A})$ corresponds to the family of complementary sets of the minimal transversals of \mathcal{A} . In the following two subsections, we will show how to extend the hypergraph dualization algorithms of [14] to arbitrary antichains \mathcal{A} of integral vectors in a box \mathcal{C} . Note that, by (3), our problem can be stated as of checking whether $\mathcal{C} = \mathcal{A}^+ \cup \mathcal{B}^-$, and if not, find an element $x \in \mathcal{C} \setminus (\mathcal{A}^+ \cup \mathcal{B}^-)$.

As in [14], we shall analyze the running time of the algorithms in terms of the *volume* $v = v(\mathcal{A}, \mathcal{B}) \stackrel{\text{def}}{=} |\mathcal{A}||\mathcal{B}|$ of the input problem. In general, a given problem will be decomposed into a number of subproblems of smaller volume, which we will solve recursively. Since we have assumed that $\mathcal{B} \subseteq \mathcal{I}(\mathcal{A})$, (3) implies that the following condition holds for the original problem and all subsequent subproblems:

$$a \not\leq b \quad \text{for all } a \in \mathcal{A}, b \in \mathcal{B}. \quad (11)$$

Let $C(v) = C(v(\mathcal{A}, \mathcal{B}))$ denote the number of subproblems that have to be solved in order to solve the original problem (the maximum number of recursive calls on a problem of volume $\leq v$), and let $m \stackrel{\text{def}}{=} |\mathcal{A}| + |\mathcal{B}|$, and $[n] \stackrel{\text{def}}{=} \{1, \dots, n\}$. We start with the following proposition that provides the base case for recursion.

Proposition 4 *Suppose $\min\{|\mathcal{A}|, |\mathcal{B}|\} \leq \text{const}$, then problem $DUAL(\mathcal{C}, \mathcal{A}, \mathcal{B})$ is solvable in time $\text{poly}(n, m)$.*

Proof. Let us assume without loss of generality that $\mathcal{B} = \{b^1, \dots, b^k\}$, for some constant k . Then $\mathcal{C} = \mathcal{A}^+ \cup \mathcal{B}^-$ if and only if for every $t = (t_1, \dots, t_k) \in [n]^k$ for which

$$b_{t_j}^j \neq u_{t_j}, \text{ for all } j \in [k], \quad (12)$$

there exists an $a \in \mathcal{A}$ such that

$$a_i \leq \max(\{b_i^j + 1 \mid j \in [k], t_j = i\} \cup \{l_i\}), \text{ for all } i \in [n]. \quad (13)$$

To see this, assume first that $\mathcal{C} = \mathcal{A}^+ \cup \mathcal{B}^-$ and consider any $t \in [n]^k$ such that (12) holds. Let $x \in \mathcal{C}$ be defined by taking $x_i = \max(\{b_i^j + 1 \mid j \in [k], t_j = i\} \cup \{l_i\})$, for $i = 1, \dots, n$. Then $x \in \mathcal{C} \setminus \mathcal{B}^-$ and hence $x \in \mathcal{A}^+$, implying that there is an $a \in \mathcal{A}$ satisfying (13). On the other hand, let us assume that for every $t \in [n]^k$ satisfying (12), there is an $a \in \mathcal{A}$ for which (13) holds. Consider an arbitrary $x \in \mathcal{C} \setminus \mathcal{B}^-$. Then, there must exist an index $t_j \in [n]$, for every $j \in [k]$, such that $x_{t_j} \geq b_{t_j}^j + 1$. The vector $t = (t_1, \dots, t_k) \in [n]^k$ constructed in this way then satisfies (12), and therefore, there is an $a \in \mathcal{A}$ such that $a_i \leq \max(\{b_i^j + 1 \mid j \in [k], t_j = i\} \cup \{l_i\}) \leq x_i$, for all $i = 1, \dots, n$, implying thus $x \in \mathcal{A}^+$.

The condition given in (12) and (13) can clearly be checked in $\text{poly}(n, m)$ time, for any constant k . If the condition does not hold for some $t \in [n]^k$ then the element x , defined by $x_i = \max(\{b_i^j + 1 \mid j \in [k], t_j = i\} \cup \{l_i\})$ for $i = 1, \dots, n$, belongs to $\mathcal{C} \setminus (\mathcal{A}^+ \cup \mathcal{B}^-)$. \square

Note that, to complete the solution of the dualization problem, we need to find an element *maximal* in $\mathcal{C} \setminus (\mathcal{A}^+ \cup \mathcal{B}^-)$. This can be easily done in polynomial time, and even independently of the chain sizes, as shown in the next proposition. Therefore, in the following subsections, we shall only show how to obtain an element $x \in \mathcal{C} \setminus (\mathcal{A}^+ \cup \mathcal{B}^-)$, when such an element exists.

Proposition 5 *Given an $x \in \mathcal{C} \setminus (\mathcal{A}^+ \cup \mathcal{B}^-)$, it can be extended to a maximal element with the same property in $O(n^2m)$ time.*

Proof. Note that, for $i = 1, \dots, n$, the i^{th} component of any maximal element in $\mathcal{C} \setminus \mathcal{A}^+$ must belong to the set $\{a_i - 1 \mid a \in \mathcal{A}\} \cup \{u_i\}$. Thus a new element $x' \geq x$ maximal in $\mathcal{C} \setminus (\mathcal{A}^+ \cup \mathcal{B}^-)$ can be found as follows. For $i = 1, \dots, n$, we find iteratively the set $\mathcal{A}_i \stackrel{\text{def}}{=} \{a \in \mathcal{A} \mid a_1 \leq x'_1, \dots, a_{i-1} \leq x'_{i-1}, a_i > x_i, a_{i+1} \leq x_{i+1}, \dots, a_n \leq x_n\}$ in $O(nm)$ time, and then set $x'_i \leftarrow \min(\{a_i - 1 \mid a \in \mathcal{A}_i\} \cup \{u_i\})$. \square

7.1 Algorithm A

Assume that \mathcal{A}, \mathcal{B} satisfy (11), and let $\mathcal{C} = \mathcal{C}_1 \times \dots \times \mathcal{C}_n$, where $\mathcal{C}_i = [l_i : u_i]$. We say that a coordinate $i \in [n]$ is *essential* for a point $a \in \mathcal{A}$ ($b \in \mathcal{B}$), if $a_i > l_i$ (respectively, $b_i < u_i$). Let us denote by $E(x)$ the set of essential coordinates of a point $x \in \mathcal{A} \cup \mathcal{B}$. The following lemma generalizes a well-known fact for dual Boolean functions (cf. [14]).

Lemma 3 *Let $\mathcal{A}, \mathcal{B} \subseteq \mathcal{C}$ be given subsets. Then, (i) either there exists an element $y \in \mathcal{A} \cup \mathcal{B}$ with few essential coordinates: $|E(y)| \leq \log m$, where $m = |\mathcal{A}| + |\mathcal{B}|$; (ii) or, if no such element y exists then $\mathcal{B} \neq \mathcal{I}(\mathcal{A})$ and an element $x \in \mathcal{C} \setminus (\mathcal{A}^+ \cup \mathcal{B}^-)$ can be found in $\text{poly}(n, m)$ time.*

Proof. Let $z \in \mathcal{C}$ be the vector obtained by picking each coordinate z_i randomly from $\{l_i, u_i\}$, $i = 1, \dots, n$, and consider the random variable $N(z) \stackrel{\text{def}}{=} |\{a \in \mathcal{A} \mid z \geq a\}| + |\{b \in \mathcal{B} \mid z \leq b\}|$. Then the expected value of $N(z)$ is given by

$$\begin{aligned} \mathbb{E}[N(z)] &= \sum_{a \in \mathcal{A}} \Pr\{z \geq a\} + \sum_{b \in \mathcal{B}} \Pr\{z \leq b\} \\ &= \sum_{a \in \mathcal{A}} \prod_{i \in E(a)} \Pr\{z_i = u_i\} + \sum_{b \in \mathcal{B}} \prod_{i \in E(b)} \Pr\{z_i = l_i\} \quad (14) \\ &= \sum_{a \in \mathcal{A}} \left(\frac{1}{2}\right)^{|E(a)|} + \sum_{b \in \mathcal{B}} \left(\frac{1}{2}\right)^{|E(b)|}. \end{aligned}$$

If we have $\mathbb{E}[N(z)] \geq 1$, then by letting $r = \min\{|E(z)| : z \in \mathcal{A} \cup \mathcal{B}\}$, we get by (14) that

$$1 \leq \mathbb{E}[N(z)] \leq (|\mathcal{A}| + |\mathcal{B}|) \left(\frac{1}{2}\right)^r = m \left(\frac{1}{2}\right)^r,$$

from which part (i) of the lemma follows.

On the other hand, if $\mathbb{E}[N(z)] < 1$, then we can find an element $x \in \mathcal{C} \setminus (\mathcal{A}^+ \cup \mathcal{B}^-)$ (in fact a corner of the finite box \mathcal{C}) in polynomial time, proving part (ii) of the lemma. To see this, let us consider for each $i = 1, \dots, n$, the variables $z_j \in \{l_j, u_j\}$ random for $j > i$, as above, compute the expectations of $N(x_1, \dots, x_{i-1}, l_i, z_{i+1}, \dots, z_n)$ and $N(x_1, \dots, x_{i-1}, u_i, z_{i+1}, \dots, z_n)$ analogously to (14), and select the value for $x_i \in \{l_i, u_i\}$ so as to minimize the corresponding expectation. Clearly, $N(x) < 1$ will hold in this case, thus $N(x) = 0$, implying that $x \notin \mathcal{A}^+ \cup \mathcal{B}^-$. \square

Next we show that, for any dual pair $(\mathcal{A}, \mathcal{B})$, an essential coordinate with high frequency exists for either \mathcal{A} or \mathcal{B} .

Lemma 4 *Let $\mathcal{A}, \mathcal{B} \subseteq \mathcal{C}$ be a pair of dual subsets, $\mathcal{B} = \mathcal{I}(\mathcal{A})$ for which $|\mathcal{A}||\mathcal{B}| \geq 1$. Then there exists a coordinate $i \in [n]$, which is essential in \mathcal{A} or in \mathcal{B} with frequency at least $1/\log m$, where $m = |\mathcal{A}| + |\mathcal{B}|$.*

Proof. By Lemma 3, the set $\mathcal{A} \cup \mathcal{B}$ must contain an element y with a logarithmically small number of essential coordinates. Suppose without loss of

generality that $y \in \mathcal{A}$. From our assumptions and condition (11), we know that for every $b \in \mathcal{B}$, there is an $i \in E(b) \cap E(y)$ such that $b_i < y_i$. Letting $\mathcal{B}_i^y \stackrel{\text{def}}{=} \{b \in \mathcal{B} \mid b_i < y_i\}$ for $i \in E(y)$, we conclude that

$$|\mathcal{B}| = \left| \bigcup_{i \in E(y)} \mathcal{B}_i^y \right| \leq \sum_{i \in E(y)} |\mathcal{B}_i^y|,$$

and therefore there is an $i \in [n]$ which is essential for at least $|\mathcal{B}|/|E(y)| \geq |\mathcal{B}|/\log m$ many elements of \mathcal{B} . \square

We are now ready to state the first dualization algorithm. Given an integral box \mathcal{C} , and subsets $\mathcal{A}, \mathcal{B} \subseteq \mathcal{C}$ that satisfy (11), in time $\text{poly}(n, m) + m^{O(\log^2 m)}$ the algorithm will either prove that $\mathcal{C} = \mathcal{A}^+ \cup \mathcal{B}^-$ or find an $x \in \mathcal{C} \setminus (\mathcal{A}^+ \cup \mathcal{B}^-)$.

Step 1. If $|\mathcal{A}||\mathcal{B}| \leq 1$, then the dualization problem can be solved in $O(n)$ time.

Step 2. For each $k \in [n]$: if $a_k < l_k$ for some $a \in \mathcal{A}$ ($b_k > u_k$ for some $b \in \mathcal{B}$), set $a_k \leftarrow l_k$ (respectively, set $b_k \leftarrow u_k$). (Note that $\mathcal{A}, \mathcal{B} \subseteq \mathcal{C}$ holds initially, but might not hold after decomposing \mathcal{C} , see Step 4 below.) Note that condition (11) continues to hold after such replacements. Thus in $O(nm)$ time, we can assure that $\mathcal{A}, \mathcal{B} \subseteq \mathcal{C}$ hold.

Step 3. Check if there is a $y \in \mathcal{A} \cup \mathcal{B}$ with at most $\log m$ essential coordinates. If no such y can be found, a new point in $\mathcal{C} \setminus (\mathcal{A}^+ \cup \mathcal{B}^-)$ can be obtained as described in the proof of Lemma 3 part (ii). Otherwise, assume without loss of generality, that $y = a^o \in \mathcal{A}$ and find an $i \in E(a^o)$ for which $|\{b \in \mathcal{B} \mid b_i < a_i^o\}| \geq |\mathcal{B}|/\log m$. Let us suppose, again without any loss of generality, that $i = 1$ and set $\mathcal{C}'_1 \leftarrow [a_1^o : u_1]$, $\mathcal{C}''_1 \leftarrow [l_1 : a_1^o - 1]$. Let further

$$\mathcal{A}'' = \{a \in \mathcal{A} \mid a_1 < a_1^o\}, \quad \mathcal{B}' = \{b \in \mathcal{B} \mid b_1 \geq a_1^o\}.$$

Observe that $|\mathcal{A}''| \leq |\mathcal{A}| - 1$ since $a^o \notin \mathcal{A}''$, and that $|\mathcal{B}'| \leq (1 - 1/\log m)|\mathcal{B}|$ by our choice of a^o and i .

Step 4. Denoting by $\mathcal{C}' = \mathcal{C}'_1 \times \mathcal{C}_2 \times \dots \times \mathcal{C}_n$, and $\mathcal{C}'' = \mathcal{C}''_1 \times \mathcal{C}_2 \times \dots \times \mathcal{C}_n$ the two sub-boxes of \mathcal{C} induced by the above partitioning, it is then easy to see that \mathcal{A} and \mathcal{B} are dual in \mathcal{C} if and only if

$$\mathcal{A}, \mathcal{B}' \text{ are dual in } \mathcal{C}' \tag{15}$$

and

$$\mathcal{A}'', \mathcal{B} \text{ are dual in } \mathcal{C}''. \tag{16}$$

Thus by applying the algorithm recursively to these two subproblems, we reduce the computation on a problem of size $v = |\mathcal{A}||\mathcal{B}|$ to computing the solution for two subproblems (15)-(16) of volumes

$$\begin{aligned} v(\mathcal{A}, \mathcal{B}') &= |\mathcal{A}||\mathcal{B}'| \leq |\mathcal{A}||\mathcal{B}|(1 - \epsilon) = (1 - \epsilon)v, \text{ and} \\ v(\mathcal{A}'', \mathcal{B}) &= |\mathcal{A}'||\mathcal{B}| = (|\mathcal{A}| - 1)|\mathcal{B}| \leq v - 1, \end{aligned}$$

where $\epsilon = 1/\log m$. This leads to the recurrence

$$C(v) \leq 1 + C((1 - \epsilon)v) + C(v - 1),$$

which was shown in [14] to evaluate to $C(v) \leq (3 + 2v\epsilon)^{\log v/\epsilon}$, implying that the running time of the algorithm is $\text{poly}(n) + m^{O(\log^2 m)}$.

In the next subsection, we shall give an algorithm that solves the problem in $\text{poly}(n, m) + m^{o(\log m)}$ time.

7.2 Algorithm B

Algorithm A of the previous subsection decomposes problem $\text{DUAL}(\mathcal{C}, \mathcal{A}, \mathcal{B})$ into two subproblems (15) and (16). As we shall see below, subproblems (15) and (16) are not independent, and we can utilize their dependence to get a more efficient dualization algorithm. Given an integral box $\mathcal{C} = \mathcal{C}_1 \times \dots \times \mathcal{C}_n$, where $\mathcal{C}_i = [l_i : u_i]$, and subsets of integral vectors \mathcal{A}, \mathcal{B} that satisfy the necessary condition (11), we proceed as follows:

Step 1. If $\min\{|\mathcal{A}|, |\mathcal{B}|\} \leq 2$, the duality of \mathcal{A} and \mathcal{B} can be tested in $O(n^3 m)$ using Proposition 4.

Step 2. For each $k \in [n]$:

- 2.1. if $a_k > u_k$ for some $a \in \mathcal{A}$ ($b_k < l_k$ for some $b \in \mathcal{B}$), then a (respectively, b) can be clearly discarded from further consideration;
- 2.2. if $a_k < l_k$ for some $a \in \mathcal{A}$ ($b_k > u_k$ for some $b \in \mathcal{B}$), we set $a_k \leftarrow l_k$ (respectively, $b_k \leftarrow u_k$).

Thus we may assume for next steps that $\mathcal{A}, \mathcal{B} \subseteq \mathcal{C}$.

Step 3. Let $a^o \in \mathcal{A}$, $b^o \in \mathcal{B}$. By (11), there exists an $i \in [n]$, such that $a_i^o > b_i^o$. Let us assume, without any loss of generality, that $i = 1$ and set $\mathcal{C}'_1 \leftarrow [a_1^o : u_1]$, $\mathcal{C}''_1 \leftarrow [l_1 : a_1^o - 1]$. (Alternatively, we may set $\mathcal{C}'_1 \leftarrow [l_1 : b_1^o]$ and $\mathcal{C}''_1 \leftarrow [b_1^o + 1 : u_1]$.) Let \mathcal{C}' and \mathcal{C}'' be the two resulting sub-boxes as defined in Step 4 of the previous subsection. Define further

$$\begin{aligned} \mathcal{A}'' &= \{a \in \mathcal{A} \mid a_1 < a_1^o\}, & \mathcal{A}' &= \mathcal{A} \setminus \mathcal{A}'', & \epsilon_1^{\mathcal{A}} &= \frac{|\mathcal{A}'|}{|\mathcal{A}|}, \\ \mathcal{B}' &= \{b \in \mathcal{B} \mid b_1 \geq a_1^o\}, & \mathcal{B}'' &= \mathcal{B} \setminus \mathcal{B}', & \epsilon_1^{\mathcal{B}} &= \frac{|\mathcal{B}''|}{|\mathcal{B}|}. \end{aligned}$$

Observe that $\epsilon_1^{\mathcal{A}} > 0$ and $\epsilon_1^{\mathcal{B}} > 0$ since $a^o \in \mathcal{A}'$ and $b^o \in \mathcal{B}''$.

Step 4. Define

$$\epsilon(v) = 1/\chi(v), \quad \text{where } \chi(v)^{\chi(v)} = v = v(\mathcal{A}, \mathcal{B}).$$

If $\min\{\epsilon_1^A, \epsilon_1^B\} > \epsilon(v)$, we use the decomposition rule given above, which amounts to solving recursively two subproblems (15), (16) of respective volumes:

$$\begin{aligned} v(\mathcal{A}, \mathcal{B}') &= |\mathcal{A}||\mathcal{B}'| = |\mathcal{A}|(1 - \epsilon_1^B)|\mathcal{B}| = (1 - \epsilon_1^B)v(\mathcal{A}, \mathcal{B}), \\ v(\mathcal{A}'', \mathcal{B}) &= |\mathcal{A}''||\mathcal{B}| = (1 - \epsilon_1^A)|\mathcal{A}||\mathcal{B}| = (1 - \epsilon_1^A)v(\mathcal{A}, \mathcal{B}). \end{aligned}$$

This gives rise to the recurrence

$$C(v) \leq 1 + C((1 - \epsilon_1^B)v) + C((1 - \epsilon_1^A)v) \leq 1 + 2C((1 - \epsilon(v))v). \quad (17)$$

Step 5. Let us now suppose that $\epsilon_1^B \leq \epsilon(v)$. In this case, we begin by solving subproblem (15). If $\mathcal{A}, \mathcal{B}'$ are not dual in \mathcal{C}' , we get a point x maximal in $\mathcal{C}' \setminus [(\mathcal{A}^+ \cup (\mathcal{B}')^-]$, and we are done. Otherwise we claim that

$$\mathcal{A}'', \mathcal{B} \text{ are dual in } \mathcal{C}'' \iff \forall a \in \tilde{\mathcal{A}} : \mathcal{A}'', \mathcal{B}'' \text{ are dual in } \mathcal{C}''(a), \quad (18)$$

where $\tilde{\mathcal{A}} = \{a \in \mathcal{A} \mid a_1 \leq a_1^o\}$, and $\mathcal{C}''(a) = \mathcal{C}_1'' \times [a_2 : u_2] \times \dots \times [a_n : u_n]$.

Proof of (18): The forward direction does not use (15). Suppose that there is an $a \in \tilde{\mathcal{A}}$ such that \mathcal{A}'' and \mathcal{B}'' are not dual in $\mathcal{C}''(a)$, i.e., there is an $x \in \mathcal{C}''(a) \setminus [(\mathcal{A}'')^+ \cup (\mathcal{B}'')^-]$. Then $x_i \geq a_i$, for $i = 2, \dots, n$. If $x \in (\mathcal{B}'')^-$, i.e., $x \leq b$ for some $b \in \mathcal{B}'$, then by the definition of \mathcal{B}' , $b_1 \geq a_1^o$. On the other hand, $a \in \tilde{\mathcal{A}}$ implies that $a_1 \leq a_1^o$. But then,

$$(a_1, a_2, \dots, a_n) \leq (a_1^o, x_2, \dots, x_n) \leq (b_1, b_2, \dots, b_n),$$

which contradicts the assumed condition (11). This shows that $x \in \mathcal{C}'' \setminus [(\mathcal{A}'')^+ \cup (\mathcal{B}' \cup \mathcal{B}'')^-]$ and hence \mathcal{A}'' and \mathcal{B} are not dual in \mathcal{C} .

For the other direction, let $x \in \mathcal{C}'' \setminus [(\mathcal{A}'')^+ \cup \mathcal{B}^-]$. Since $x \notin (\mathcal{B}')^-$ and $x = (x_1, x_2, \dots, x_n) < y \stackrel{\text{def}}{=} (a_1^o, x_2, \dots, x_n)$, the vector y also satisfies $y \in \mathcal{C}' \setminus (\mathcal{B}')^-$. We conclude therefore, assuming (15), that $y \in \mathcal{A}^+$, i.e., there is an $a \in \mathcal{A}$ such that $a \leq y$. But this implies that $a \in \tilde{\mathcal{A}}$ and hence that $x \in \mathcal{C}''(a) \setminus [(\mathcal{A}'')^+ \cup (\mathcal{B}'')^-]$ for some $a \in \tilde{\mathcal{A}}$. \square

It follows by (18) that, once we discover that (15) holds, we can reduce the solution of subproblem (16) to solving $|\tilde{\mathcal{A}}|$ subproblems, each of which has a volume of $v(|\mathcal{A}''|, |\mathcal{B}''|) \leq \epsilon_1^B v(\mathcal{A}, \mathcal{B})$. Thus we obtain the recurrence

$$C(v) \leq 1 + C((1 - \epsilon_1^B)v) + |\mathcal{A}|C(\epsilon_1^B v). \quad (19)$$

Step 6. Finally, if $\epsilon_1^A \leq \epsilon(v) < \epsilon_1^B$, we solve subproblem (16), and if we discover that $\mathcal{A}'', \mathcal{B}$ are dual in \mathcal{C}'' , we obtain the following decomposition rule, symmetric to (18):

$$\mathcal{A}, \mathcal{B}' \text{ are dual in } \mathcal{C}' \iff \forall b \in \tilde{\mathcal{B}} : \mathcal{A}', \mathcal{B}' \text{ are dual in } \mathcal{C}'(b),$$

where $\tilde{\mathcal{B}} = \{b \in \mathcal{B} \mid b_1 \geq a_1^o - 1\}$, and $\mathcal{C}'(b) = \mathcal{C}'_1 \times [l_2 : b_2] \times \dots \times [l_n : b_n]$. This reduces our original problem into one subproblem of volume $\leq (1 - \epsilon_1^A)v$, plus $|\tilde{\mathcal{B}}|$ subproblems, each of volume at most $\epsilon_1^A v$, thus giving the recurrence

$$C(v) \leq 1 + C((1 - \epsilon_1^A)v) + |\mathcal{B}|C(\epsilon_1^A v), \quad (20)$$

which is the symmetric version of (19).

Using induction on $v \geq 9$, it can be shown that recurrences (17), (19) and (20) imply $C(v) \leq v^{\chi(v)}$ (see [14]). As $\chi(m^2) < 2\chi(m)$ and $v(\mathcal{A}, \mathcal{B}) < m^2$, we get $\chi(v) < \chi(m^2) < 2\chi(m) \sim 2 \log m / \log \log m$. Let us also note that every step above can be implemented in $O(n^3 m)$ time, independently of the sizes of the chains $|\mathcal{C}_i|$. This establishes the bound stated in Theorem 4.

8 Bounding the Size of $\mathcal{F}_{A,b,c}$

To prove inequality (5) of Theorem 5, let us consider an arbitrary non-empty antichain $\mathcal{Y} \subseteq \mathcal{I}(\mathcal{F}_{A,b,c})$. For any $y \in \mathcal{I}(\mathcal{F}_{A,b,c})$ we can find an index $i = \rho(y) \in [r] \stackrel{\text{def}}{=} \{1, \dots, r\}$ such that y violates the i^{th} inequality of the system, i.e., $a^i y < b_i$, where a^i and b_i denote the i^{th} row and component of A and b , respectively.

Consider a vector $x \in \mathcal{I}^{-1}(\mathcal{Y}) \cap \mathcal{F}_{A,b,c}$ and let x_l be a positive component of x . Then there exists a vector $y^l \in \mathcal{Y}$ such that $y^l \geq x - \mathbf{e}^l$. Let $i = \rho(y^l)$ and assume without loss of generality that

$$a_1^i \geq a_2^i \geq \dots \geq a_n^i. \quad (21)$$

We claim that $x(l) = z^l(l)$, where

$$z^l = y^l(l) + \mathbf{e}^l. \quad (22)$$

It follows from $y^l \geq x - \mathbf{e}^l$ that $z^l(l) \geq x(l)$. If $z^l_i > x_l$, then $y^l_i \geq x_l$, which implies $y^l \geq x$, a contradiction. Thus $z^l_i = x_l$ holds. Moreover, if $z^l_j > x_j$ for some $j < l$, then we have $y^l_j \geq x_j + 1$. By (21), $a^i(y^l - \mathbf{e}^j + \mathbf{e}^l) < b_i$, i.e., $y^l - \mathbf{e}^j + \mathbf{e}^l$ is infeasible for (1). However, $y^l - \mathbf{e}^j + \mathbf{e}^l \geq x$ by $y^l \geq x - \mathbf{e}^l$ and hence $y^l - \mathbf{e}^j + \mathbf{e}^l$ must be feasible. This shows that $x(l) = z^l(l)$ and consequently leads to the representation

$$x = \bigvee_{l \in [n]: x_l > 0} z^l, \quad (23)$$

where for vectors $v, u \in \mathcal{C}$ we let $v \vee u$ denote the component-wise maximum of v and u .

Not all of the vectors z^l are necessary for this representation. Suppose that $\rho(y^l) = \rho(y^{l'}) = i$ for some positive components x_l and $x_{l'}$ of x , and $l' < l$.

Then (23) remains valid if we drop $z^{l'}$, the vector with the smaller index l' . In other words, by sorting the i^{th} row of A and then selecting among the vectors $y^l \in \rho^{-1}(i)$ the one with the highest $l = l_i$, we obtain at most r vectors z^{l_i} such that

$$x = \bigvee_{i \in [r]} z^{l_i}. \quad (24)$$

The latter representation readily implies (5). \square

9 Polynomial Generation of $\mathcal{F}_{A,b,c}$ and $\mathcal{I}(\mathcal{F}_{A,b,c})$

Theorem 5 implies that for $r \leq \text{const}$ the antichain $\mathcal{I}(\mathcal{F}_{A,b,c})$ is uniformly dual-bounded and consequently, $\mathcal{I}(\mathcal{F}_{A,b,c})$ can be generated in incremental quasipolynomial time via Algorithm \mathcal{J} presented in Section 6. In this section we show that for bounded r , the antichains $\mathcal{I}(\mathcal{F}_{A,b,c})$ and $\mathcal{F}_{A,b,c}$ can, in fact, be generated in incremental polynomial time. Since the sizes $|\mathcal{F}_{A,b,c}|$ and $|\mathcal{I}(\mathcal{F}_{A,b,c})|$ are (uniformly) polynomially related by Theorems 2 and 5, the required result will follow from Proposition 2, provided that Step 3 of Algorithm \mathcal{J} , the dualization step, can be done in polynomial time. Thus, it is enough to show that problem $\text{DUAL}(\mathcal{C}, \mathcal{A}, \mathcal{B})$ can be solved in polynomial time, if $\mathcal{A} \subseteq \mathcal{F}_{A,b,c}$ is a subset of the minimal solutions of a monotone system (1) with bounded r .

For $i \in [r] = \{1, \dots, r\}$, let $\sigma_{(i)} = (\sigma_{(i)1}, \dots, \sigma_{(i)n}) \in \mathbb{S}_n$ be a permutation of the coordinates such that

$$a_{\sigma_{(i)1}}^i \geq a_{\sigma_{(i)2}}^i \geq \dots \geq a_{\sigma_{(i)n}}^i. \quad (25)$$

Given $\mathcal{A} \subseteq \mathcal{F}_{A,b,c}$ and $\mathcal{B} \subseteq \mathcal{I}(\mathcal{F}_{A,b,c})$, we may assume that $0 \notin \mathcal{A}$ (otherwise, $\mathcal{F}_{A,b,c} = \{0\}$ and $\mathcal{I}(\mathcal{F}_{A,b,c}) = \emptyset$) and $\mathcal{B} \neq \emptyset$ (otherwise, Proposition 4 can be used to generate a point $x \in \mathcal{C} \setminus (\mathcal{A}^+ \cup \mathcal{B}^-)$). Now we proceed in two basic steps:

Step 1. For every $y \in \mathcal{B}$ and for each pair of indices $\sigma_{(i)j}$ and $\sigma_{(i)l}$ with $y_{\sigma_{(i)j}} > 0$, $y_{\sigma_{(i)l}} < c_{\sigma_{(i)l}}$ and $j < l$, check if there exists a $y' \in \mathcal{B}$ such that

$$y' \geq y - \mathbf{e}^{\sigma_{(i)j}} + \mathbf{e}^{\sigma_{(i)l}}, \quad (26)$$

where $i = \rho(y)$ is the index of an infeasible inequality for y , as defined in the previous section. Note that $y - \mathbf{e}^{\sigma_{(i)j}} + \mathbf{e}^{\sigma_{(i)l}}$ is infeasible, and hence is an independent element of \mathcal{A} . If no such y' can be found, we generate a new maximal independent vector $y' \in \mathcal{I}(\mathcal{A}) \setminus \mathcal{B}$, satisfying (26), and halt.

Step 2. For every collection $(y^i \in \rho_B^{-1}(i) \mid i \in [r])$, where $\rho_B^{-1}(i) \stackrel{\text{def}}{=} \{y \in \mathcal{B} \mid \rho(y) = i\}$, and for every $(l_i \mid i \in [r], y_{l_i}^i < c_{l_i}) \in [n]^r$, construct the vector $x = \bigvee_{i \in [r]} z^{l_i}$, where z^{l_i} is given by (22) (according to the permutation $\sigma_{(i)}$) and

using $y^l = y^{l_i}$.) If $x \notin \mathcal{A}^+ \cup \mathcal{B}^-$, then a new maximal independent vector can be generated.

Clearly, the above two steps run in $\text{poly}(n, m) + O((n|\mathcal{B}|)^r)$ time, which is polynomially bounded for constant r . It is also clear that if the algorithm outputs a point $x \in \mathcal{C}$, then $x \notin \mathcal{A}^+ \cup \mathcal{B}^-$, so it remains to verify that the algorithm indeed outputs such a point if $\mathcal{A}^+ \cup \mathcal{B}^- \neq \mathcal{C}$. To see this, let x be a minimal vector in $\mathcal{C} \setminus (\mathcal{A}^+ \cup \mathcal{B}^-)$. From our assumptions, it follows that $x \neq 0$, and thus, there exists an index l with $x_l > 0$. By the minimality of x , there exists a vector $y^l \in \mathcal{B}$ such that $y^l \geq x - \mathbf{e}^l$. Let $i = \rho(y^l)$, assume without any loss of generality that (21) holds, and consider an index $j < l$. If $y_j^l > x_j$, we get $y^l - \mathbf{e}^j + \mathbf{e}^l \geq x$, and therefore a new maximal independent point $x' \geq x$ must have been output in Step 1 of the algorithm (c.f. (26)). On the other hand, if for every $l \in [n]$ such that $x_l > 0$ and for every $y^l \in \mathcal{B}$ such that $y^l \geq x - \mathbf{e}^l$, we have $x(l) = z^l(l)$ (in the ordering implied by $\sigma(i)$, where $i = \rho(y^l)$), then we can conclude that x satisfies (24), and consequently, it must have been created in Step 2. \square

10 Concluding Remarks

We mention in closing that in Section 5, we actually proved Theorem 2 for arbitrary systems of 2-monotonic inequalities in integer variables. Consequently, the set of minimal feasible integer vectors for any system of 2-monotonic inequalities is uniformly dual-bounded. By Proposition 3, this implies that all minimal integer solutions to a system of 2-monotonic inequalities can be generated in incremental quasi-polynomial time, provided that the system has a polynomial-time feasibility oracle. Theorems 5 and 6 also hold for arbitrary systems of 2-monotonic inequalities. These generalize results known for a single Boolean 2-monotonic inequality, discussed in [4, 8, 9, 18, 24, 26, 27].

References

- [1] E. Balas and E. Zemel, All the facets of zero-one programming polytopes with positive coefficients, Management Sciences Research Report 374, Carnegie-Mellon University, Pittsburgh, 1975.
- [2] J. C. Bioch, Dualization, decision lists and identification of monotone discrete functions, *Annals of Mathematics and Artificial Intelligence* 24 (1998) 69-91.
- [3] J. C. Bioch and T. Ibaraki, Complexity of identification and dualization of positive Boolean functions, *Information and Computation* 123 (1995) 50-63.

- [4] P. Bertolazzi and A. Sassano, An $O(nm)$ algorithm for regular set-covering problems, *Theoretical Computer Science* 54 (1987), 237-247.
- [5] E. Boros, K. Elbassioni, V. Gurvich and L. Khachiyan, An efficient incremental algorithm for generating all maximal independent sets in hypergraphs of bounded dimension, *Parallel Processing Letters*, **10** (4) (2000) pp. 253-266.
- [6] E. Boros, V. Gurvich, and P.L. Hammer, Dual subimplicants of positive Boolean functions, *Optimization Methods and Software*, 10 (1998) 147-156.
- [7] E. Boros, V. Gurvich, L. Khachiyan and K.Makino, Dual-bounded generating problems: Partial and multiple transversals of a hypergraph, *SIAM Journal on Computing* **30**(6) (2001) pp. 2036-2050.
- [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] Y. Crama, P. L. Hammer and T. Ibaraki, Cause-effect relationships and partially defined boolean functions, *Annals of Operations Research* 16 (1988) 299-326.
- [11] C. Domingo, N. Mishra and L. Pitt, Efficient read-restricted monotone CNF/DNF dualization by learning with membership queries, *Machine learning*, 37 (1999) 89-110.
- [12] T. Eiter and G. Gottlob, Identifying the minimal transversals of a hypergraph and related problems, *SIAM Journal on Computing*, 24 (1995) 1278-1304.
- [13] T. Eiter, G. Gottlob, and K. Makino, New results on monotone dualization and generating hypergraph transversals, *Proceedings of the 34th ACM Symposium on Theory of Computing (STOC-02)*, May 19-21, 2002, Montreal, Quebec, Canada.
- [14] M. L. Fredman and L. Khachiyan, On the complexity of dualization of monotone disjunctive normal forms, *Journal of Algorithms*, 21 (1996) 618-628.
- [15] D. R. Gaur. *Satisfiability and self-duality of monotone Boolean functions*. Ph.D. thesis, School of Computing Science, Simon Fraser University, January 1999.
- [16] D. R. Gaur, R. Krishnamurti. Self-duality of bounded monotone Boolean functions and related problems. In: *Proc. 11th International Conference on Algorithmic Learning Theory (ALT)*, pp. 209-223, Springer LNCS 1968, 2000.

- [17] V. Gurvich and L. Khachiyan, On generating the irredundant conjunctive and disjunctive normal forms of monotone Boolean functions, *Discrete Applied Mathematics*, 96-97, (1999) 363-373.
- [18] P. L. Hammer, U.N. Peled, and M.A. Pollatschek, An algorithm to dualize a regular switching function, *IEEE Trans. on Computers* **28** (1979), 238-243.
- [19] D. S. Johnson, M. Yannakakis and C. H. Papadimitriou, On generating all maximal independent sets, *Information Processing Letters*, 27 (1988) 119-123.
- [20] 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.
- [21] K. Makino. Efficient dualization of $O(\log n)$ -term monotone disjunctive normal forms. Technical Report 00-07, Discrete Mathematics and Systems Science, Osaka University, 2000; to appear in *Discrete Applied Mathematics*.
- [22] K. Makino and T. Ibaraki, Interior and exterior functions of Boolean functions, *Discrete Applied Mathematics*, 69 (1996) 209-231.
- [23] K. Makino and T. Ibaraki. The maximum latency and identification of positive Boolean functions. *SIAM Journal on Computing*, 26:1363–1383, 1997.
- [24] K. Makino and T. Ibaraki, A fast and simple algorithm for identifying 2-monotonic positive Boolean functions, *Journal of Algorithms*, 26 (1998) 291-305.
- [25] O. L. Mangasarian, Mathematical programming in machine learning, in G. Di. Pillo and F. Giannessi eds, *Nonlinear Optimization and Applications* (Plenum Publishing, New York, 1996) 283-295.
- [26] U. N. Peled and B. Simeone, Polynomial-time algorithm for regular set-covering and threshold synthesis, *Discrete Applied Mathematics* 12 (1985) 57-69.
- [27] 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.
- [28] R. C. Read and R. E. Tarjan, Bounds on backtrack algorithms for listing cycles, paths, and spanning trees, *Networks* 5 (1975) 237-252.
- [29] H. Tamaki. Space-efficient enumeration of minimal transversals of a hypergraph. *IPJSJ-AL* 75 (2000) 29-36.

- [30] S. Tsukiyama, M. Ide, H. Ariyoshi and I. Shirakawa, A new algorithm for generating all maximal independent sets, *SIAM Journal on Computing*, 6 (1977) 505-517.
- [31] T. Uno, Private communication.