Computing the Integer Points of a Polyhedron, II: Complexity Estimates

Rui-Juan Jing[1,2(✉)] and Marc Moreno Maza[2]

[1] KLMM, UCAS, Academy of Mathematics and Systems Science, Chinese Academy of Sciences, Beijing, China
[2] University of Western Ontario, London, Canada
rjing8@uwo.ca, moreno@csd.uwo.ca

Abstract. Let K be a polyhedron in $\mathbb{R}^d$, given by a system of m linear inequalities, with rational number coefficients bounded over in absolute value by L. In this series of two papers, we propose an algorithm for computing an irredundant representation of the integer points of K, in terms of "simpler" polyhedra, each of them having at least one integer point. Using the terminology of W. Pugh: for any such polyhedron P, no integer point of its grey shadow extends to an integer point of P. We show that, under mild assumptions, our algorithm runs in exponential time w.r.t. d and in polynomial w.r.t m and L. We report on a software experimentation. In this series of two papers, the first one presents our algorithm and the second one discusses our complexity estimates.

1 Introduction

In the first paper of that series of two, we have presented an algorithm, called IntegerSolve, for decomposing the set of integer points of a polyhedron. See Sect. 4 of the first paper. This second paper is dedicated to complexity estimates considering both running time and output size. Our main result is Theorem 1, which states an exponential time complexity[1] for IntegerSolve, under Hypothesis 1, that we call *Pugh's assumption*. Before discussing this hypothesis and stating the theorem, we set up some notations.

Notation 1. *Recall that we consider a polyhedral set $K \subseteq \mathbb{R}^d$ given by an irredundant intersection $K = \cap_{i=1}^{i=m} H_i$ of closed half-spaces $H_1, \ldots, H_m$ such that, for each $i = 1, \ldots, m$, the half-space H_i is defined by $\mathbf{a}_i^T \mathbf{x} \leq b_i$, with $\mathbf{a}_i \in \mathbb{Z}^d$ and $b_i \in \mathbb{Z}$. The conjunction of those inequalities forms a system of linear inequalities that we denote by $\mathbf{A}\mathbf{x} \leq \mathbf{b}$. Let L (resp. h) be the maximum absolute value (resp. maximum bit size) of a coefficient in either $\mathbf{A}$ or $\mathbf{b}$. Thus $h = \lfloor \log_2(L) \rfloor + 2$.*

Hypothesis 1. We assume that during the execution of the function call IntegerSolve(K), for any polyhedral set K', input of a recursive call, each facet of the dark shadow[2] of K' is parallel to a facet of the real shadow of K'.

[1] To be precise, in the EXP complexity class.
[2] The notions of real shadow, dark shadow and grey shadow are presented in Sect. 3 of the first paper.

V.P. Gerdt et al. (Eds.): CASC 2017, LNCS 10490, pp. 242–256, 2017.
DOI: 10.1007/978-3-319-66320-3_18

The figure in the introduction of the first paper shows a polyhedron for which each facet of its dark shadow is parallel to a facet of its real shadow. This property is commonly observed in practice, see Sect. 5. In [9], W. Pugh observes that it is possible to build polyhedra K that challenge the Omega Test by generating many recursive calls when searching for integer points of K that extend integer points of its grey shadow. But he notices that, in practice, this combinatorial explosion is rare, due to the fact that the grey shadow of K is often empty (or at least for most of the recursive calls of the Omega test, when searching for integer points in K). This experimental observation leads us to Hypothesis 1 which is less strong than the property observed by W. Pugh, while being sufficient to guarantee that our algorithm runs in exponential time.

We believe that this running estimate could still hold with the following even weaker hypothesis: during the execution of the function call $\mathsf{IntegerSolve}(K)$, for any polyhedral set K', input of a recursive call, the number of facets of the dark shadow of K' is in "big-O" of the number of facets of its real shadow. Investigating this question is left for future work.

To state our main result, we need a notation for the running time of solving a linear program. Indeed, linear programming is an essential tool for removing redundant inequalities generated by Fourier-Motzkin elimination, see [7].

Notation 2. *For an input linear program with total bit size H and with d variables, we denote by $\mathsf{LP}(d, H)$ an upper bound for the number of bit operations required for solving this linear program. For instance, in the case of Karmarkar's algorithm [6], we have $\mathsf{LP}(d, H) \in O(d^{3.5}H^2 \cdot \log H \cdot \log\log H)$.*

Theorem 1. *Under Hypothesis 1, the call function $\mathsf{IntegerSolve}(K)$ runs within $O(m^{2d^2}d^{4d^3}L^{4d^3}\mathsf{LP}(d, m^d d^4(\log d + \log L)))$ bit operations.*

The running time estimate in Theorem 1 is exponential w.r.t. d but polynomial w.r.t m and L. Since our algorithm transforms the Omega Test from a decision procedure into a system solving algorithm, our result also holds for the original Omega Test. To our knowledge, this is the first complexity estimate for the whole Omega Test procedure.

The proof follows from a series of results established in Sects. 2, 3 and 4. We believe that some of them are interesting on their own.

Section 2 deals with the following problem. Let F be a k-dimensional face of K, for $0 \leq k < d$. What is the computational cost of projecting F onto a k-dimensional linear subspace of $\mathbb{R}^d$?

Section 3 gives complexity estimates for Fourier-Motzkin elimination (FME). While it is known that FME can run in single exponential time [5,7], we are not aware of running time estimates for FME in the literature. Thanks to Hypothesis 1, our FME estimates applies to the DarkShadow sub-routine of IntegerSolve.

Section 4 gathers results for completing the proof of Theorem 1. The recursive nature of this algorithm leads us to give upper bounds for three quantities: the number of nodes in the tree of the recursive calls, the number of facets of each polyhedron input of a recursive call, the maximum absolute value of a coefficient in a linear system defining such a polyhedron.

2 Properties of the Projection of Faces of a Polyhedron

This section gathers preliminary results towards the complexity analysis of the IntegerSolve algorithm. Some of these results are probably not new, but we could not find a reference for them in the literature.

Definition 1. *Let I be a subset of $\{1, \ldots, m\}$ and denote by $\mathbf{B}_I$ the affine space $\{\mathbf{x} \in \mathbb{R}^d \mid \mathbf{a}_i^T \mathbf{x} = b_i \text{ for } i \in I\}$. If $\mathbf{B}_I \cap K$ is not empty, then $\mathbf{B}_I \cap K$ is a face of K. We call such an index set I a* defining index set *of the face $\mathbf{B}_I \cap K$.*

Let F be a k-dimensional face of K for an integer $0 \leq k < d$ and let I be a defining index set of F with maximum cardinality. Consider the set O_I given by:

$$O_I \quad = \quad \mathbf{B}_I \cap \{\mathbf{x} \mid \mathbf{a}_i^T \mathbf{x} < b_i, i \notin I\}. \tag{1}$$

Proposition 1. *The set O_I is not empty.*

Proof. The assumption on I implies that for all $i \notin I$ the equality $\mathbf{a}_i^T \mathbf{x} = b_i$ is not an implicit equation of F. Indeed, if $\mathbf{a}_i^T \mathbf{x} = b_i$ were an implicit equation of F, then the set $\{i\} \cup I$ would be a defining index set of F as well, a contradiction. From Sect. 8.1 of [10] and since no equation $\mathbf{a}_i^T \mathbf{x} = b_i$ for $i \notin I$ is an implicit equation of F defined by I, we deduce that the set O_I is not empty. □

Using Gaussian elimination, we can compute a parametric representation of O_I where $\dim(\mathbf{B}_I)$ variables are treated as parameters; we denote by $\mathbf{x}'$ those parameters. The other $d - \dim(\mathbf{B}_I)$ variables are referred as *main variables* or *leading variables*, following the terminology of the theory of regular chains [2]. Once we substitute the main variables by their linear forms in the parameters (solved from $\mathbf{B}_I$) into the system $\{\mathbf{x} \mid \mathbf{a}_i^T \mathbf{x} < b_i, i \notin I\}$, we obtain a consistent strict inequality system in the parameters, whose solution set, that we call O_o, is of dimension $\dim(\mathbf{B}_I)$ in the parameter space.

Proposition 2. *We have* $\dim(\mathbf{B}_I) = \dim(O_I) = \dim(O_o) = \dim(F) = k$.

Proof. Note that the set O_o is the image of O_I in the standard projection onto the parameter space and that O_o is open in that space (equipped with the Euclidean topology). Hence, we have $\dim(\mathbf{B}_I) = \dim(O_I) = \dim(O_o)$. In fact, this elimination-and-substitution process shows that O_I is the solution set of a so-called *regular semi-algebraic system* [4] where the regular chain part is given by a regular chain of height $d - \dim(\mathbf{B}_I)$. Meanwhile, we have $\dim(\mathbf{B}_I) \geq \dim(F) \geq \dim(O_I)$, since $\mathbf{B}_I \supseteq F \supseteq O_I$ holds by definition. Moreover, we have $\dim(O_I) \geq \dim(O_o) = \dim(\mathbf{B}_I)$ since O_o is the image of O_I. Finally, since $\dim(F) = k$ holds by assumption, we deduce $\dim(\mathbf{B}_I) = \dim(F) = k$. □

The following lemma was found by the authors independently of the work of Imbert [5] but it is likely that our result could be derived from that paper.

Lemma 1. *Let F be a k-dimensional face of K for some integer $0 \leq k < d$. Then, the face F admits a defining index set with cardinality $d - k$.*

Proof. First, we shall prove that there exists a defining index set with cardinality at least $d-k$. Assume that I is a defining index set of F with maximum cardinality. From Proposition 2, we have $\dim(\mathbf{B}_I) = k$, hence I has at least $d-k$ elements. Assume $I = \{i_1, i_2, \ldots, i_t\}$, with $t \geq d-k$. Since $\dim(\mathbf{B}_I) = k$ holds, one can easily deduce that the rank of the matrix

$$(\mathbf{a}_{i_1}^T, \mathbf{a}_{i_2}^T, \ldots, \mathbf{a}_{i_t}^T)$$

and the rank of the matrix

$$(\mathbf{a}_{i_1}^T, \mathbf{a}_{i_2}^T, \ldots, \mathbf{a}_{i_t}^T, (b_{i_1}, b_{i_2}, \ldots, b_{i_t})^T)$$

are both $d-k$. Thus, we can further assume w.l.o.g. that

$$(\mathbf{a}_{i_1}^T, \mathbf{a}_{i_2}^T, \ldots, \mathbf{a}_{i_{d-k}}^T)$$

has rank $d-k$. Then clearly, the set $I^* = \{i_1, i_2, \ldots, i_{d-k}\}$ is also a defining index set of F. That is, the k-dimensional face F admits a defining index set with cardinality $d-k$. □

Corollary 1 follows immediately from Lemma 1.

Corollary 1. *Let $0 \leq k < d$ be an integer. Let $f_{d,m,k}$ be the number of k-dimensional faces of K. Then, we have*

$$f_{d,m,k} \leq \binom{m}{d-k}.$$

Therefore, we have

$$f_{d,m,0} + f_{d,m,1} + \cdots + f_{d,m,d-1} \leq m^d.$$

Note that, from now on, when we say *a defining index set of a k-dimensional face of K*, we shall always refer to one with cardinality $d-k$. Let F_P be the closure of O_o in the Euclidean topology. Then, F_P is the projection of F on the coordinates $\mathbf{x}'$, where $\mathbf{x}'$ stand for the parameters introduced above. Thus, F_P is a polyhedron and Corollary 2 gives upper-bound estimates on a representation of F_P with a system of linear inequalities.

Corollary 2. *One can compute a matrix $\mathbf{C}$ over $\mathbb{Z}$ and a vector $\mathbf{d}$ over $\mathbb{Z}$ such that the integer points of F_P are given by $\mathbf{C}\mathbf{x}' \leq \mathbf{d}$ and the maximum absolute value of a coefficient in either $\mathbf{C}$ or $\mathbf{d}$ is no more than $(d-k+1)^{\frac{d-k+1}{2}} L^{d-k+1}$, where L is the maximum absolute value of a coefficient in either $\mathbf{A}$ or $\mathbf{b}$.*

Proof. Without loss of generality, assume $I = \{1, \ldots, d-k\}$. From Proposition 2, we have $\dim(F_P) = k$. From the proof of Lemma 1, the rank of the matrix $(\mathbf{a}_1^T, \mathbf{a}_2^T, \ldots, \mathbf{a}_{d-k}^T)$ and the rank of matrix $(\mathbf{a}_1^T, \mathbf{a}_2^T, \ldots, \mathbf{a}_{d-k}^T, (b_1, b_2, \ldots, b_{d-k})^T)$ are both equal to $d-k$. Without loss of generality, assume that the first $d-k$ rows of each of the above two matrices are linearly independent. Therefore, we have $\mathbf{x}' = [x_{d-k+1}, \ldots, x_d]^T$. It follows that F_P can be defined by the inequality

system $\mathbf{C}\mathbf{x}' \leq \mathbf{d}$ obtained by Fraction-Free Gaussian Elimination, where $\mathbf{C}$ and $\mathbf{d}$ are given by:

$\mathbf{C} = (c_{i,j})_{d-k<i,j\leq d}$, where $c_{i,j}$ is the determinant of $\begin{pmatrix} a_{11} & \cdots & a_{1,d-k} & a_{1,j} \\ \vdots & \vdots & \vdots & \vdots \\ a_{d-k,1} & \cdots & a_{d-k,d-k} & a_{d-k,j} \\ a_{i,1} & \cdots & a_{i,d-k} & a_{i,j} \end{pmatrix}$,

$\mathbf{d} = [d_{d-k+1}, \ldots, d_d]^T$, where d_i is the determinant of $\begin{pmatrix} a_{11} & \cdots & a_{1,d-k} & b_1 \\ \vdots & \vdots & \vdots & \vdots \\ a_{d-k,1} & \cdots & a_{d-k,d-k} & b_{d-k} \\ a_{i,1} & \cdots & a_{i,d-k} & b_i \end{pmatrix}$.

Using Hadamard's inequality, the absolute value of any $c_{i,j}$ and d_j can be bounded by $(d-k+1)^{\frac{d-k+1}{2}} L^{d-k+1}$. □

3 Complexity Estimates for Fourier-Motzkin Elimination

Proposition 4 states a running time estimate for computing the linear inequality system $\mathsf{proj}(K; x_1, \ldots, x_d)$ defined in Sect. 2 of the first paper. Note that the article [7] states that Fourier-Motzkin elimination can be run in single exponential time but without giving a running time estimate. Let $k < d$ be a positive integer. Following the notations of Sect. 2 of the first paper, we denote by $\Pi^{x_{k+1},\ldots,x_d}$ the standard projection from $\mathbb{R}^d$ to $\mathbb{R}^{d-k}$ mapping $(x_1, \ldots, x_d)$ to $(x_{k+1}, \ldots, x_d)$.

Proposition 3. *Assume that K is full-dimensional. Then, we have:*

(i) The projected polyhedron $\Pi^{x_{k+1},\ldots,x_d} K$ admits at most $\binom{m}{d-k-1}$ facets.
(ii) Any facet of $\Pi^{x_{k+1},\ldots,x_d} K$ can be given by a system consisting of one linear equation and $m-k-1$ linear inequalities, all in $\mathbb{R}^{d-k}$, such that the absolute value of any coefficient in those constraints is at most $(k+1)^{\frac{k+1}{2}} L^{k+1}$.

Proof. Let G be a facet of $\Pi^{x_{k+1},\ldots,x_d} K$. There exists a face F of K such that G is the projection of F. Since K is full-dimensional, it is clear that $\Pi^{x_{k+1},\ldots,x_d} K$ is full-dimensional as well. Hence, we have $\dim(G) = d-k-1 \leq \dim(F)$. Clearly, choosing F with minimum dimension implies $d-k-1 = \dim(F)$. With Corollary 1, we deduce (i). Now we prove (ii). It follows from Lemma 1 that one can choose a defining index set I of F with cardinality $d-(d-k-1) = k+1$. Thus, we have $\mathbf{B}_I \cap K = F$, with $\mathbf{B}_I$ given in Definition 1. Consider, then, the set O_I given by (1). We know from Proposition 1 that O_I is not empty and from Proposition 2 that $\dim(\mathbf{B}_I) = d-k-1$. Consider now the system of linear equations given by:

$$G_I \quad = \quad \mathbf{B}_I \cap \{\mathbf{x} \mid \mathbf{a}_i^T \mathbf{x} = b_i, i \notin I\}. \tag{2}$$

Using Fraction-Free Gaussian Elimination on G_I and since $\dim(\mathbf{B}_I) = d-k-1$ holds, one can use the $k+1$ equations defining $\mathbf{B}_I$ to eliminate $x_1, \ldots, x_k$ from the

inequalities $\{\mathbf{x} \mid \mathbf{a}_i^T \mathbf{x} < b_i, i \notin I\}$ and, in addition, obtain one equation involving the variables $x_{k+1}, \ldots, x_d$ only. Clearly, the resulting inequalities and equation exactly define G. Using Hadamard's inequality as in the proof of Corollary 2, we deduce (ii). □

Definition 2. *Let θ be an inequality in the irredundant representation of the projected polyhedron $\Pi^{x_{k+1},\ldots,x_d} K$. Let G be the facet of $\Pi^{x_{k+1},\ldots,x_d} K$ associated with θ. There exists a $(k+1)$-dimensional face G' of K such that $\Pi^{x_{k+1},\ldots,x_d} G' = G$ holds. We call* defining index set *of θ any defining index set of G'.*

Lemma 2. *Let $\mathbf{v} \in \mathbb{R}^d$ and $s \in \mathbb{R}$ such that h is also the maximum bit size of any coefficient in $\mathbf{v}$ and s. Hence, the total bit size of the linear program $\sup\{-(\mathbf{vx} - s) \mid \mathbf{Ax} \leq \mathbf{b}\}$ is $H \in O(h\,m\,d)$. Moreover, deciding whether the inequality $\mathbf{vx} \leq s$ is implied by $\mathbf{Ax} \leq \mathbf{b}$ or not can be done within $O(\mathsf{LP}(d, H))$ bit operations.*

Proof. The estimate $H \in O(h\,m\,d)$ clearly holds. On the other hand, the inequality $\mathbf{vx} \leq s$ is implied by $\mathbf{Ax} \leq \mathbf{b}$ if only if $\sup\{-(\mathbf{vx} - s) \mid \mathbf{Ax} \leq \mathbf{b}\}$ is zero. □

Proposition 4. *Within $O(d^2\, m^{2d}\, \mathsf{LP}(d, 2^d h d^2 m^d))$ bit operations, the projected representation $\mathsf{proj}(K; x_1, \ldots, x_d)$ of K can be computed.*

Proof. Following the notations of Sect. 2 of the first paper, the process of eliminating x_1 in $\mathbf{Ax} \leq \mathbf{b}$ generates at most $\frac{m^2}{4}$ new inequalities. Augmenting $\mathbf{A}^{<x_1}$ with all these new inequalities and, making this augmented system irredundant, we obtain a total number of inequalities that we denote by c_2. We define $c_1 := m$, $m_1 := c_1$ and $m_2 := c_1 + c_2$. We observe that:

1. generating all the new inequalities (irredundant or not) amounts to at most $O(\frac{m_1^2}{4}\, d\, h_1^2)$ bit operations, and
2. removing the redundant ones amounts to at most to $O(\frac{m_1^2}{4}\ \mathsf{LP}(d, h_1\, d\, m_1))$ bit operations, thanks to Lemma 2.

Similarly, during the process of eliminating x_2, we observe that:

1. generating all the new inequalities (irredundant or not) amounts to at most $O(\frac{c_2^2}{4}\, d\, h_2^2)$ bit operations, and
2. removing the redundant ones amounts to at most to $O(\frac{c_2^2}{4}\ \mathsf{LP}(d, h_2\, d\, m_2))$ bit operations and yields a total number of c_3 inequalities in x_3.

Continuing in this manner, we deduce that for successively eliminating $x_1, \ldots, x_{d-1}$,

1. generating all the new inequalities (irredundant or not) amounts to at most

$$O(\frac{c_1^2}{4}\, d\, h_1^2 + \cdots + \frac{c_{d-1}^2}{4}\, d\, h_{d-1}^2), \tag{3}$$

2. removing the redundant ones amounts to at most to

$$O(\frac{c_1^2}{4}\mathsf{LP}(d, h_1\, d\, m_1) + \cdots + \frac{c_{d-1}^2}{4}\,\mathsf{LP}(d, h_{d-1}\, d\, m_{d-1})), \tag{4}$$

where $m_i := c_1 + \cdots + c_i$, for $1 \leq i < d$, as well as $h_0 := h$ and $h_{i+1} \leq 2\, h_i + 1$, for $0 \leq i < d$. We observe that c_i is bounded over by the number of facets of $\Pi^{x_i,\ldots,x_d}K$, for $1 \leq i < d$. Observe also that, for $1 < i < d$, each facet of $\Pi^{x_i,\ldots,x_d}K$ is the projection of a face of $\Pi^{x_{i-1},\ldots,x_d}K$. Using Lemma 1, we deduce that, for all $1 \leq i < d$, we have: $c_i \leq m^d$. Therefore, the running time estimates of (3) and (4) can be bounded over by $O(d^2\, m^{2d}\, d\, (2^d h)^2)$ and $O(d^2\, m^{2d}\, \mathsf{LP}(d, (2^d h)d(dm^d)))$. The latter dominates the former; the conclusion follows. □

4 Proof of Theorem 1

We use Fig. 1 and Notation 3 to provide further explanation on Algorithm $\mathsf{IntegerSolve}_0$, presented in the first paper.

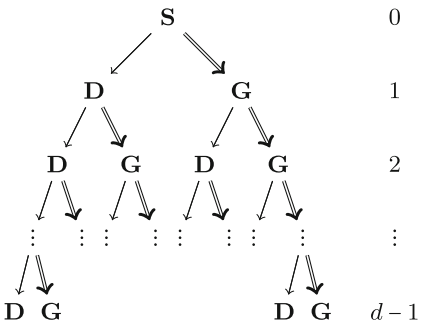

Fig. 1. Diagram

Notation 3. *Fig. 1 illustrates the tree of recursive calls for the* $\mathsf{IntegerSolve}_0$ *procedure. The root of the tree is labelled with* **S***, which stands for the input system. The left (resp. right) child of a node, other than a leaf, is labelled by* **D** *(resp.* **G***) which stands for the output of the* DarkShadow *procedure (resp. the* GreyShadow *procedure). Since the* DarkShadow *procedure generates one input system for* IntegerSolve*, we use a simple* $\rightarrow$ *arrow as an edge to a* **D***-node. However, the* GreyShadow *procedure may generate several linear inequality systems, leading to several recursive calls to* $\mathsf{IntegerSolve}_0$*. Thus, we use a* $\Rightarrow$ *arrow as an edge to a* **G***-node. The numbers on the right-hand side of Fig. 1 stand for the levels in the tree.*

Let $\mathbf{Ax} \leq \mathbf{b}, m, d$ be as in Notation 1. Let L and h denote the maximum absolute value and height of any coefficient in either $\mathbf{A}$ or $\mathbf{b}$.

Notation 4. *Recall that Fig. 1 depicts the tree of recursive calls in Algorithm* IntegerSolve$_0$. *Let* $\mathbf{N}$ *denote any node in that tree, whether it is labelled* $\mathbf{S}$, $\mathbf{D}$ *or* $\mathbf{G}$. *If* $\mathbf{N}$ *is labelled with* $\mathbf{S}$ *or* $\mathbf{D}$, *it is associated with a single linear system denoted by* $\mathbf{M_N t_N} \leq \mathbf{v_N}$. *If* $\mathbf{N}$ *is labelled with* $\mathbf{G}$, *it is associated with a sequence of linear systems produced by the* GreyShadow *procedure and we denote by* $\mathbf{M_N t_N} \leq \mathbf{v_N}$ *any of those systems. For any linear system* $\mathbf{M_N t_N} \leq \mathbf{v_N}$ *(whether* $\mathbf{N}$ *is labelled* $\mathbf{S}$, $\mathbf{D}$ *or* $\mathbf{G}$*), we denote by* $m_\mathbf{N}$ *and* $d_\mathbf{N}$ *the number of rows and columns of* $\mathbf{M_N}$. *We denote by* $L_\mathbf{N}$ *(resp.* $\ell_\mathbf{N}$*) be the maximum absolute value of any coefficient in* $\mathbf{M_N}$ *(resp. in either* $\mathbf{M_N}$ *or* $\mathbf{v_N}$*). We denote by* $h_\mathbf{N} = \lfloor \log_2 \ell_\mathbf{N} \rfloor + 1$ *the maximum bit size of a coefficient in either* $\mathbf{M_N}$ *or* $\mathbf{v_N}$. *The system* $\mathbf{M_N t_N} \leq \mathbf{v_N}$ *encodes a polyhedron* $K_\mathbf{N}$ *in* $\mathbb{R}^{d_\mathbf{N}}$ *and we denote by* $F_\mathbf{N}$ *an arbitrary facet of* $K_\mathbf{N}$. *Every path from the root to a leaf* $\mathbf{N}_r$ *in the tree depicted in Fig. 1 can be labelled* $\mathbf{S} \to \mathbf{N}_1 \to \cdots \to \mathbf{N}_r$ *for some* $r \leq d-1$. *Note that a leaf (that is, a node with no children) may have level less than* $d-1$. *For simplicity, for the node* $\mathbf{N}_r$, *we write* d_r, L_r, ℓ_r, h_r, $\mathbf{t}_r$, $\mathbf{M}_r$, $\mathbf{v}_r$, K_r, F_r *instead of* $d_{\mathbf{N}_r}$, $L_{\mathbf{N}_r}$, $\ell_{\mathbf{N}_r}$, $h_{\mathbf{N}_r}$, $\mathbf{t}_{\mathbf{N}_r}$, $\mathbf{M}_{\mathbf{N}_r}$, $\mathbf{v}_{\mathbf{N}_r}$, $K_{\mathbf{N}_r}$, $F_{\mathbf{N}_r}$ *respectively, when there is no ambiguity.*

In particular, let d_0, L_0, ℓ_0, h_0, $\mathbf{t}_0$, $\mathbf{M}_0$, $\mathbf{v}_0$, K_0, F_0 *denote the corresponding values of node* $\mathbf{S}$.

Without loss of generality, we assume the polyhedron K is full-dimensional, that is, $\dim(K) = d$ and, thus, that the input system $\mathbf{S}$ has no implicit equations. Then, each call to the DarkShadow or GreyShadow procedures at level 1 reduces the dimension of the ambient space by one. Similarly, at every level, we assume that the input system of inequalities of IntegerSolve$_0$ (that is, the fourth argument of this procedure) is full-dimensional. Hence at Line 23 of Algorithm IntegerSolve$_0$, the output of IntegerNormalize($\mathbf{Ax} \leq \mathbf{b}$) is $(\varnothing, \varnothing, \mathbf{Ax} \leq \mathbf{b})$.

This full-dimensionality assumption has two consequences. First, along any path $\mathbf{S} \to \mathbf{N}_1 \to \cdots \to \mathbf{N}_r$ we have $d_{k+1} = d_k - 1$, for $1 \leq k < r$, and thus, we have $d_k = d - k$. Second, at node $\mathbf{N}_k$, the input system is $\mathbf{M}_{k-1}\mathbf{t}_{k-1} \leq \mathbf{v}_{k-1}$ (while the output is $\mathbf{M}_k\mathbf{t}_k \leq \mathbf{v}_k$).

It is easy to see that this full-dimensionality assumption is a worst case scenario as far as running time is concerned. Indeed, when this assumption does not hold, for at least one path $\mathbf{S} \to \mathbf{N}_1 \to \cdots \to \mathbf{N}_r$, implicit equations will be discovered at Line 23 of Algorithm IntegerSolve$_0$ in the first paper, and dimension will drop by more than one at one node of that path.

To prove Theorem 1 we shall establish a series of intermediate results. Lemmas 5, 7, 8 provide upper bounds for the absolute values of any coefficient in the systems $\mathbf{M_N t_N} \leq \mathbf{v_N}$ while Lemmas 3, 4, 9, 10 deal with running time estimates. We start with Lemmas 3 and 4, which give running time estimates for the DarkShadow and GreyShadow procedures at level k. The proof of Lemma 3 follows that of Proposition 4.

Lemma 3. *For any non-negative integer* $k < d-1$, *the* DarkShadow *procedure at level* $k+1$ *runs within* $O(\frac{m_k^2}{4}\mathsf{LP}(d_k, d_k h_k m_k))$ *bit operations.*

Proof. The input system of $\mathbf{D}_{k+1}$ is $\mathbf{M}_k\mathbf{t}_k \leq \mathbf{v}_k$, which has m_k inequalities and h_k as maximum coefficient size in either $\mathbf{M}_k$ or $\mathbf{v}_k$. The process of

(E) eliminating the first variable of $\mathbf{t}_k$ in $\mathbf{M}_k\mathbf{t}_k \leq \mathbf{v}_k$,
(A) adding the at most $\frac{m_k^2}{4}$ resulting inequalities to those of $\mathbf{M}_k\mathbf{t}_k \leq \mathbf{v}_k$ where the first variable of $\mathbf{t}_k$ does not appear, and
(R) removing all redundant inequalities,

yields $\mathbf{M}_k\mathbf{t}_k \leq \mathbf{v}_k$, see Algorithm DarkShadow. Observe that Steps (E) and (R) amount to at most $O(\frac{m_k^2}{4}\, d_k\, h_k^2)$ and $O(\frac{m_k^2}{4}\ \mathsf{LP}(d_k, h_k\, d_k\, m_k))$ bit operations, respectively. The latter dominates the former. The conclusion follows. □

Lemma 4. *For any non-negative integer $k < d-1$, the* GreyShadow *procedure at level $k+1$ runs within $O(m_k^2 d_k^{3+\varepsilon} h_k^3)$ bit operations.*

Proof. For the GreyShadow procedure at level $k+1$, we need to call at most m_k times the IntegerNormalize procedure. Then, the lemma follows from Proposition 1 in the first paper. □

Lemma 5. *Consider a path of the form*

$$\mathbf{S} \to \mathbf{D}_1 \to \cdots \to \mathbf{D}_r, \tag{5}$$

where all nodes, except the first one, are labelled by $\mathbf{D}$. *Then, for all $1 \leq k \leq r$, we have $m_k \leq m^{k+1}$ and the maximum absolute value L_k of any coefficient in $\mathbf{M}_k$ is no more than $(k+1)^{\frac{k+1}{2}} L^{k+1}$.*

Proof. Let $1 \leq k \leq r$. Under Hypothesis 1, each facet of K_k is parallel to a facet of the real shadow of K_{k-1}. Inductively, each facet of K_k is parallel to a facet of the projection $\Pi^{x_{k+1},\ldots,x_d} K$. By Proposition 3, we have $m_k \leq m^{k+1}$ and $L_k \leq (k+1)^{\frac{k+1}{2}} L^{k+1}$. □

Next, we will consider an arbitrary path:

$$\mathbf{S} \to \mathbf{D}_1 \to \cdots \to \mathbf{D}_{j_1-1} \to \mathbf{G}_{j_1} \to \cdots \to \mathbf{G}_{j_s} \to \mathbf{D}_{j_s+1} \to \cdots \to \mathbf{D}_r. \tag{6}$$

In the path (6), only the subscripts $j_1, j_2, \ldots, j_s$ correspond to the GreyShadow procedures.

To make things simpler, instead of setting $\Theta_2 := \Upsilon \cup \mathbf{Mt} \leq \mathbf{v} \cup \{c\alpha - a\gamma > -(c-1)(a-1)\}$ in Line 8 of Algorithm GreyShadow in the first paper, we let $\Theta_2 := \mathbf{Mt} \leq \mathbf{v}$. This simplification cannot guarantee that $V_{\mathbb{Z}}(\mathbf{t} = \mathbf{P}_k\mathbf{u}_k + \mathbf{q}_k \cup \mathbf{M}_k\mathbf{u}_k \leq \mathbf{v}_k,\ \mathbf{t})$ for $k = 1, \ldots, s$ form a disjoint union. However, it will endow $\mathbf{M}_k$ with good structural properties, as we will see later. Actually, since all the inequalities in Υ and the negation of $c\alpha - a\gamma > -(c-1)(a-1)$ can be obtained by the DarkShadow procedure and since we are doing the worst case complexity analysis, all the coming conclusions apply to our algorithm as it was originally stated in the first paper.

First, we consider the sub-path of (6): $\mathbf{S} \to \mathbf{D}_1 \to \cdots \to \mathbf{D}_{j_1-1} \to \mathbf{G}_{j_1}$. We assume the variable order is $x_1 > x_2 > \cdots > x_d$. Thus, we can denote the variable set $\mathbf{t}_{j_1-1}$ for the input system of node $\mathbf{D}_{j_1-1}$ as: $\mathbf{t}_{j_1-1} = [x_{j_1}, x_{j_1+1}, \ldots, x_d]^T$ since $\mathbf{t}_{j_1-1} \subset \mathbf{x}$. For the node $\mathbf{G}_{j_1}$, we need to add one equation based on the

output system $\mathbf{M}_{j_1-1}\mathbf{t}_{j_1-1} \le \mathbf{v}_{j_1-1}$ of node $\mathbf{D}_{j_1-1}$. Without loss of generality, we assume the new equation is $\mathbf{mt}_{j_1-1} = v + i$ for some non-negative integer $i \le L_{j_1-1}$, where $\mathbf{mt}_{j_1-1} \le v$ is the first inequality in the system $\mathbf{M}_{j_1-1}\mathbf{t}_{j_1-1} \le \mathbf{v}_{j_1-1}$. Let I be the defining index set of $\mathbf{mt}_{j_1-1} \le v$, which has cardinality j_1.

Recall that $\mathbf{M}_0\mathbf{t}_0 \le \mathbf{v}_0$ is the input system of node $\mathbf{D}_1$. Let $\mathbf{M}_0^{(1)}$ and $\mathbf{M}_0^{(2)}$ be the sub-matrices of $\mathbf{M}_0$ consisting of the first $j_1 - 1$ columns and the last $d - j_1 + 1$ columns, respectively. Denote by $(\mathbf{v}_0)_I$ (resp. $(\mathbf{M}_0)_I$) the sub-vector (resp. sub-matrix) of $\mathbf{v}_0$ (resp. $\mathbf{M}_0$) with index (resp. row index) I. Let $\mathbf{Q}_{j_1-1}$ be a matrix whose columns consist of a $\mathbb{Z}$-basis of the space $\{\mathbf{x} : (\mathbf{M}_0)_I\,\mathbf{x} = \mathbf{0}\}$. We have assumed that the input polyhedron K is full-dimensional, which implies that the rank of $\mathbf{M}_0$ is d. By the definition of defining index set I, we can easily deduce that the rank of $(\mathbf{M}_0)_I$ is j_1, that is, $\mathbf{Q}_{j_1-1}$ is an integer matrix with d rows and $d - j_1$ columns. Let $\mathbf{Q}'_{j_1-1}$ be the sub-matrix consisting of the last $d-j_1+1$ rows of $\mathbf{Q}_{j_1-1}$. Let $\mathbf{V}_1 := [\mathbf{e}_1, \ldots, \mathbf{e}_{j_1-1}, \binom{\mathbf{0}}{\mathbf{Q}'_{j_1-1}}] \in \mathbb{Z}^{d\times(d-1)}$. Let $\mathbf{S}_1$ be a node associated with the system $\mathbf{M}_0^{(1)}[x_1, \ldots, x_{j_1-1}] + \mathbf{M}_0^{(2)}\mathbf{Q}'_{j_1-1}\mathbf{t}_{j_1+1} \le \mathbf{v}'_0$, *i.e.* $\mathbf{M}_0\mathbf{V}_1[x_1, \ldots, x_{j_1-1}, \mathbf{t}_{j_1+1}]^T \le \mathbf{v}'_0$. For $j_1 \le k < j_2$, let $\mathbf{M}'_k\mathbf{t}'_k \le \mathbf{v}'_k$ be the output system of the node $\mathbf{D}_{k-1}$ in the path: $\mathbf{S}_1 \to \mathbf{D}_1 \to \cdots \to \mathbf{D}_{k-1}$.

Lemma 6. *With the above notations, we have* $\mathbf{M}_{j_1} = \mathbf{M}'_{j_1}$. *Consequently,* $\mathbf{M}_k = \mathbf{M}'_k$ *for* $j_1 \le k < j_2$.

Proof. The second statement will follow once the first lemma is valid.

Following the algorithm DarkShadow in the first paper, there exists a matrix $U \in \mathbb{Z}^{m_{j_1-1}\times m_0}$, such that $U\mathbf{M}_0^{(1)} = \mathbf{0}$ and $\mathbf{M}_{j_1-1} = U_d U\mathbf{M}_0^{(2)}$, where $U_d = \text{DiagonalMatrix}(\frac{1}{gcd_1}, \ldots, \frac{1}{gcd_{m_{j_1-1}}})$ and gcd_i is the gcd of all the coefficients in the i-th row of $U\mathbf{M}_0^{(2)}$ for $1 \le i \le m_{j_1-1}$. Let $\mathbf{u} \in \mathbb{Z}^{m_0}$ be the first row of U. Then, $\mathbf{m} = \frac{1}{gcd_1}\mathbf{u}\mathbf{M}_0^{(2)}$ since $\mathbf{mt}_{j_1-1} \le v$ is the first inequality of $\mathbf{M}_{j_1-1}\mathbf{t}_{j_1-1} \le \mathbf{v}_{j_1}$. Then, $\mathbf{m} = \frac{1}{gcd_1}\mathbf{u}_I(\mathbf{M}_0^{(2)})_I$. Solving the equation $\mathbf{mt}_{j_1-1} = v+i$ by Lemma 2 in the first paper, we have $\mathbf{t}_{j_1-1} = \mathbf{P}_{j_1-1}\,\mathbf{t}_{j_1} + \mathbf{q}_{j_1-1}$, where $\mathbf{P}_{j_1-1} \in \mathbb{Z}^{(d-j_1+1)\times(d-j_1)}$ whose columns consist of a $\mathbb{Z}$-basis for $\{\mathbf{y} : \mathbf{my} = 0\} = \{\mathbf{y} : \mathbf{u}_I(\mathbf{M}_0^{(2)})_I\mathbf{y} = 0\}$. Therefore, $\mathbf{M}_{j_1}\mathbf{t}_{j_1} \le \mathbf{v}_{j_1}$ comes from $\mathbf{M}_{j_1-1}\mathbf{P}_{j_1-1}\mathbf{t}_{j_1} \le \mathbf{v}_{j_1-1} - \mathbf{M}_{j_1-1}\mathbf{q}_{j_1-1}$, *i.e.* $U_d U\mathbf{M}_0^{(2)}\mathbf{P}_{j_1-1}\mathbf{t}_{j_1} \le \mathbf{v}_{j_1-1} - \mathbf{M}_{j_1-1}\mathbf{q}_{j_1-1}$.

Next, we will show that $\mathbf{P}_{j_1-1}$ can be replaced by $\mathbf{Q}'_{j_1-1}$ introduced above. Since $\mathbf{u}_I(\mathbf{M}_0^{(1)})_I = 0$, we have any $\mathbf{y} \in \mathbb{Z}^d$ satisfying $\mathbf{u}_I(\mathbf{M}_0)_I\mathbf{y} = \mathbf{0}$ is equivalent to $\mathbf{u}_I(\mathbf{M}_0^{(2)})_I\mathbf{y}^{(2)} = 0$, where $\mathbf{y}^{(2)}$ is the last $d - j_1 + 1$ elements of $\mathbf{y}$. Thus, $[\mathbf{e}_1, \ldots, \mathbf{e}_{j_1-1}, \binom{\mathbf{0}}{\mathbf{P}_{j_1-1}}]$ is a $\mathbb{Z}$-basis for the space $\{\mathbf{y} : \mathbf{u}(\mathbf{M}_0)_I\mathbf{y} = 0\}$. For any row vector $\mathbf{y} \in \mathbb{Z}^d$ such that $\mathbf{u}_I(\mathbf{M}_0)_I\mathbf{y} = 0$, either $\mathbf{0} \ne (\mathbf{M}_0)_I\mathbf{y} \in \{\mathbf{z} : \mathbf{u}_I\mathbf{z} = 0\}$ or $(\mathbf{M}_0)_I\mathbf{y} = \mathbf{0}$. For the first case, $\mathbf{e}_1, \ldots, \mathbf{e}_{j_1-1}$ is a $\mathbb{Z}$-basis for the solutions of $\mathbf{y}$, where $\mathbf{e}_k \in \mathbb{Z}^d$ is the k-th standard basis for $1 \le k \le j_1 - 1$. For the second case, columns of $\mathbf{Q}_{j_1-1}$ consisting of a $\mathbb{Z}$-basis for the solutions of $\mathbf{y}$. Thus, $[\mathbf{e}_1, \ldots, \mathbf{e}_{j_1-1}, \mathbf{Q}_{j_1-1}]$ is a $\mathbb{Z}$-basis for the space $\{\mathbf{y} : \mathbf{u}_I(\mathbf{M}_0)_I\mathbf{y} = 0\}$. Consequently, $\mathbf{P}_{j_1-1}$ is equivalent to $\mathbf{Q}'_{j_1-1}$, which is the last $d - j_1 + 1$ rows of

$\mathbf{Q}_{j_1-1}$. That is, the integer solutions to $\mathbf{m}\,\mathbf{t}_{j_1-1} = v + i$ can be represented by $\mathbf{t}_{j_1-1} = \mathbf{Q}'_{j_1-1}\,\mathbf{t}_{j_1} + \mathbf{q}_{j_1-1}$, where $|\mathbf{Q}'_{j_1-1}| \leq j_1^{j_1+1} L^{2j_1}$. Therefore, we can make $\mathbf{M}_{j_1} = U_d U \mathbf{M}_0^{(2)} \mathbf{Q}'_{j_1-1}$.

Remember that $\mathbf{S}_1$ is associated with the system $\mathbf{M}_0\mathbf{V}_1[x_1, \ldots, x_{j_1-1}, \mathbf{t}_{j_1+1}]^T \leq \mathbf{v}_0$, where $\mathbf{V}_1 := [\mathbf{e}_1, \ldots, \mathbf{e}_{j_1-1}, \binom{\mathbf{0}}{\mathbf{Q}'_{j_1-1}}]$, and $\mathbf{M}'_k\mathbf{t}'_k \leq \mathbf{v}'_k$ is the output system of the node $\mathbf{D}_{k-1}$ in the path: $\mathbf{S}_1 \to \mathbf{D}_1 \to \cdots \to \mathbf{D}_{k-1}$. We have $\mathbf{M}'_{j_1} = U_d U \mathbf{M}_0^{(2)} \mathbf{Q}'_{j_1}$. Consequently, $\mathbf{M}'_k = \mathbf{M}_k$ for any integer $k : j_1 \leq k < j_2$. □

Then, we have the following lemma:

Lemma 7. *For $j_1 \leq k < j_2$, the maximum absolute value of any coefficient in $\mathbf{M}_k$ can be bounded over by $d^k k^{2k^2} L^{3k^2}$. Moreover, we have $m_k \leq m^{k+1}$.*

Proof. By Lemma 5, the maximum absolute value of any coefficient in $\mathbf{M}'_k = \mathbf{M}_k$ can be bounded over by $|\mathbf{M}_k| \leq k^{\frac{k}{2}}(d - j_1 + 1)^k j_1^{kj_1+k} L^{2kj_1+k} \leq d^k k^{2k^2} L^{3k^2}$.

Moreover, $m_k \leq m^{k+1}$ follows from the equivalent path $\mathbf{S}_1 \to \mathbf{D}_1 \to \cdots \to \mathbf{D}_{k-1}$ for integer $k : j_1 \leq k < j_2$. □

For any $1 \leq t \leq s$, we assume that the new equation is $\mathbf{m}_t\mathbf{t}_{j_t-1} = v_t + i_t$ for some non-negative integer $i_t \leq L_{j_t-1}$, where $\mathbf{m}_t\mathbf{t}_{j_t-1} \leq v_t$ comes from the input system $\mathbf{M}_{j_t-1}\mathbf{t}_{j_t-1} \leq \mathbf{v}_{j_t-1}$ of the node $\mathbf{G}_{j_t}$. Let I_t be the defining index set of the inequality $\mathbf{m}_t\mathbf{t}_{j_t-1} \leq v_t$, with cardinality j_t. Let $\mathbf{Q}_t \in \mathbb{Z}^{d\times(d-j_t)}$ consist of the columns of a $\mathbb{Z}$-basis of space $\{\mathbf{y} : (\mathbf{M}_0)_{I_t}\mathbf{y} = 0\}$. For any $1 \leq t \leq s$, we define $\mathbf{V}_t = [\mathbf{e}_1, \ldots, \mathbf{e}_{j_1-1}, \mathbf{Q}_1^{(1)}, \ldots, \mathbf{Q}_{t-1}^{(t-1)}, \mathbf{Q}_t^{(t)}] \in \mathbb{Z}^{d\times(d-t)}$ and $\mathbf{t}_t := [x_1, \ldots, x_{j_1}, \mathbf{t}_{j_1-1}^{(1)}, \ldots, \mathbf{t}_{j_t-1}^{(t)}]^T$ as follows:

1. When $k < t$, we let $\mathbf{Q}'_k$ be the sub-matrix consisting of the last $d - j_k + 1$ rows and $(j_{k+1} - j_k - 1)$ columns of $\mathbf{Q}_k$. Let $\mathbf{Q}_k^{(k)} \in \mathbb{Z}^{d\times(j_{k+1}-j_k-1)}$ be the matrix $\binom{\mathbf{0}}{\mathbf{Q}'_k}$, where $\mathbf{0}$ is a zero matrix which has $j_k - 1$ rows and $j_{k+1} - j_k - 1$ columns.
2. When $k = t$, we let $\mathbf{Q}'_t$ be the sub-matrix consisting of the last $d - j_t + 1$ rows of $\mathbf{Q}_t$. Let $\mathbf{Q}_t^{(t)} \in \mathbb{Z}^{d\times(d-j_t)}$ be the matrix $\binom{\mathbf{0}}{\mathbf{Q}'_t}$, where $\mathbf{0}$ is a zero matrix which has $j_t - 1$ rows and $d - j_t$ columns.
3. Denote by $\mathbf{t}_{j_k-1}^{(k)}$ (resp. $\mathbf{t}_{j_t-1}$) the set of $j_{k+1} - j_k - 1$ (resp. $d - j_t$) variables. and the variables in $\mathbf{t}_t$ are independent variables.

Let $\mathbf{S}_t$ be the system represented by $\mathbf{M}_0\mathbf{V}_t\mathbf{t}_t \leq \mathbf{v}_0$ for $1 \leq t \leq s$.

Lemma 8. *The maximum absolute value of any coefficient in $|\mathbf{M}_k|$ (resp. $|\mathbf{v}_k|$) can be bounded by $d^k k^{2k^2} L^{3k^2}$ (resp. $d^{3k^2} k^{4k^3} L^{6k^3}$) for $1 \leq k \leq r$. Moreover, we have $m_k \leq m^{k+1}$.*

Proof. Similar to the notations defined before Lemma 6, for $1 \leq t \leq s$ and $j_t \leq k < j_{t+1}$, let $\mathbf{M}'_k\mathbf{t}'_k \leq \mathbf{v}'_k$ be the output system of the path: $\mathbf{S}_t \to \mathbf{D}_1 \to \cdots \to \mathbf{D}_{k-t}$, where j_{s+1} is defined as $r + 1$.

We claim $\mathbf{M}_{j_t} = \mathbf{M}'_{j_t}$ for any $1 \le t \le s$, where $\mathbf{M}_{j_t}\mathbf{t}_{j_t} \le \mathbf{v}_{j_t}$ is the output system of the path: $\mathbf{S} \to \mathbf{D}_1 \to \cdots \to \mathbf{D}_{j_1-1} \to \mathbf{G}_{j_1} \to \cdots \to \mathbf{G}_{j_t}$. This claim is valid if $t = 1$ by Lemma 6. We suppose it is valid for $t = 1, \ldots, s-1$. Then, we have $\mathbf{M}_{j_s-1} = \mathbf{M}'_{j_s-1}$, where $\mathbf{M}'_{j_s-1}\mathbf{t}'_{j_s-1} \le \mathbf{v}'_{j_s-1}$ is the output system of the node $\mathbf{D}_{j_s-s}$ in the path: $\mathbf{S}_{s-1} \to \mathbf{D}_1 \to \cdots \to \mathbf{D}_{j_s-s}$. Let $\mathbf{M}''_{j_s}\mathbf{t}''_{j_s} \le \mathbf{v}''_{j_s}$ be the output system of node $\mathbf{G}_{j_s-s+1}$ in the path: $\mathbf{S}_{s-1} \to \mathbf{D}_1 \to \cdots \to \mathbf{D}_{j_s-s} \to \mathbf{G}_{j_s-s+1}$. Note that $\mathbf{S}_{s-1}$ is associated with $\mathbf{M}_0\mathbf{V}_{s-1}\mathbf{t}_{s-1} \le \mathbf{v}_0$ and the input system of node $\mathbf{G}_{j_s-s+1}$ is $\mathbf{M}'_{j_s-1}\mathbf{t}'_{j_s-1} \le \mathbf{v}'_{j_s-1}$, *i.e.* $\mathbf{M}_{j_s-1}\mathbf{t}'_{j_s-1} \le \mathbf{v}'_{j_s-1}$. We have $\mathbf{M}''_{j_s} = \mathbf{M}_{j_s}$ immediately, since both of them come from the output system of node $\mathbf{G}_{j_s-s+1}$ of the path: $\mathbf{S}_{s-1} \to \mathbf{D}_1 \to \cdots \to \mathbf{D}_{j_s-s} \to \mathbf{G}_{j_s-s+1}$. By the proof of Lemma 6, $\mathbf{M}''_{j_s}$ can be obtained from the output system of the path: $\mathbf{S}'_s \to \mathbf{D}_1 \to \cdots \to \mathbf{D}_{j_s-s}$, where $\mathbf{S}'_s$ is associated with $\mathbf{M}_0[\mathbf{V}_{s-1}[\mathbf{e}_1, \ldots, \mathbf{e}_{j_s-s}], \mathbf{Q}_s]\mathbf{t}_s \le \mathbf{v}_0$, *i.e.* $\mathbf{M}_0\mathbf{V}_s\mathbf{t}_s \le \mathbf{v}_0$, which associates to the label $\mathbf{S}_s$. Then, we have $\mathbf{M}''_{j_s} = \mathbf{M}'_{j_s}$. The claim is valid. That is, $\mathbf{M}_{j_s}$ can be obtained from the output system of the path: $\mathbf{S}_s \to \mathbf{D}_1 \to \cdots \to \mathbf{D}_{j_s-s}$.

By Proposition 1 of the first paper, we know that the maximum absolute value of any coefficient in $\mathbf{M}_0\mathbf{V}_t$ can be bounded by $dj_t^{j_t+1}L^{2j_t+1}$. Thus, by Lemma 5, for any $1 \le t \le s$ and $j_t \le k < j_{t+1}$, we have the maximum absolute value of any coefficient in $\mathbf{M}_k$ can be bounded by $(k-t)^{\frac{k-t}{2}}(dj_t^{j_t+1}L^{2j_t+1})^{k-t} \le d^k k^{2k^2} L^{3k^2}$. The first statement is valid.

Let $1 \le k \le r$. For the node $\mathbf{D}_k$, we have $|\mathbf{v}_k| \le L_{k-1}^2 + 2L_{k-1}|\mathbf{v}_{k-1}|$. For the node $\mathbf{G}_k$, we have $|\mathbf{v}_k| \le 2d_k L_{k-1}^2|\mathbf{v}_{k-1}|$ since we only need to solve one equation. That is, for any node $\mathbf{N}_k$, we will have $|\mathbf{v}_k| \le 2d_k L_{k-1}^2|\mathbf{v}_{k-1}|$. Thus, $|\mathbf{v}_k| \le 2^k d^k L_{k-1}^2 \cdots L_1^2|\mathbf{v}_0|^2 \le d^{3k^2}k^{4k^3}L^{6k^3}$ for any $1 \le k \le r$. □

Until now, we can safely say that any coefficient in $\mathbf{M}_r$ (resp.$\mathbf{v}_r$) produced by each path in Fig. 1 can be bounded over by $L_r \le d^r r^{2r^2} L^{3r^2}$ (resp. $\ell_r \le d^{3r^2} r^{4r^3} L^{6r^3}$). That is, the coefficient size associated with the node $\mathbf{N}_r$ can be bounded over by $h_r \le 6r^3(\log d + \log L)$. Moreover, we can have at most m_r inequalities in $\mathbf{M}_r\mathbf{t}_r \le \mathbf{v}_r$. The following lemma shows the complexity estimates for implementing each path of the tree in Fig. 1:

Lemma 9. *The path (6) can be implemented within* $O(m^{2r+2}d^{3+\varepsilon}r^{10}(\log d + \log L)^3) + O(rm^{2r+2}\mathsf{LP}(d, dm^r r^3(\log d + \log L)))$ *bit operations.*

Proof. By Lemma 3 (resp. Lemma 4), each node $\mathbf{D}_k$ (resp. $\mathbf{G}_k$) can be implemented with $O(\frac{m_k^2}{4}\mathsf{LP}(d_k, d_k h_k m_k))$ (resp. $O(m_k^2 d_k^{3+\varepsilon} h_k^3)$) bit operations. Thus, the path (6) can be implemented within

$$\begin{aligned} & r \cdot O(m_r^2 d_r^{3+\varepsilon} h_r^3) + r \cdot O(\frac{m_r^2}{4}\mathsf{LP}(d, dh_r m_r)) \\ \le\ & O(m^{2r+2}d^{3+\varepsilon}r^{10}(\log d + \log L)^3) + O(rm^{2r+2}\mathsf{LP}(d, dm^{2r+2}r^3(\log d + \log L))) \end{aligned}$$

bit operations. □

Let T_r be the total number of nodes in the r-th level. In particular, we have $T_0 = 1$, $T_1 \le mL$. We have the following lemma:

Lemma 10. *We have:* $T_{r+1} \leq m^{r+1} d^r r^{2r^2} L^{3r^2} T_r$ *for* $r = 0, \ldots, d-2$. *Thus, we have* $T_{d-1} \leq m^{d^2} d^{3d^3} L^{3d^3}$.

Proof. By Lemma 8, each node can have at most m^{r+1} inequalities as the input and each inequality has coefficient bound L_r. Following the Algorithm $\mathsf{IntegerSolve}_0$ and Fig. 1, each node can give out at most $m^{r+1} L_r$ branches. Considering we have T_r nodes in the r-th level, we can easily deduce that $T_{r+1} \leq m^{r+1} L_r T_r \leq m^{r+1} d^r r^{2r^2} L^{3r^2} T_r$. The second statement follows easily. □

Now we give the proof for Theorem 1:

Proof. Under Hypothesis 1, by Lemmas 9 and 10, the complexity estimates for $\mathsf{IntegerSolve}(K)$ can be bounded over

$$
\begin{aligned}
&T_{d-1} O(m^{2r+2} d^{3+\varepsilon} r^{10} (\log d + \log L)^3) + \\
&T_{d-1} O(m^{2r+2} r \mathsf{LP}(d, d m^{r+1} r^3 (\log d + \log L))) \\
\leq &O(m^{2d^2} d^{4d^3} L^{4d^3} \mathsf{LP}(d, m^d d^4 (\log d + \log L))) \text{ bit operations, since } r < d.
\end{aligned}
$$

The theorem is valid. □

5 Experimentation

We have implemented the algorithm presented in the first paper within the `Polyhedra` library in MAPLE. This library is publicly available in source on the download page of the `RegularChains` library at www.regularchains.org.

We have used test-cases coming from various application areas: regular polytopes (first 5 examples in Table 1), examples from Presburger arithmetic (next 5 examples in Table 1), random polytopes (next 5 examples in Table 1), random unbounded polyhedra (next 5 examples in Table 1), examples from text-books (next 3 examples in Table 1) and examples from research articles on automatic parallelization of for-loop nests (last 4 examples in Table 1).

For each example, Table 1 gives the number of defining inequalities (Column m), the number of variables (Column d), the maximum absolute value of an input coefficient (Column L), the number of polyhedra returned by IntegerSolve (Column m_o), the maximum absolute value of an output coefficient (Column L_o) and whether Hypothesis 1 holds or not (Column ?Hyp).

Recall from Sect. 4.1 of the first paper that Step ($S4$) of the IntegerNormalize procedure can use either the HNF method introduced in Lemma 2 of the first paper, or the method introduced by Pugh in [9]. We implemented both of them. It is important to observe that Pugh's method does not solve systems of linear equations according to our prescribed variable order, in contrast to the HNF method. In fact, Pugh's method determines a variable order dynamically, based on coefficient size considerations. In Table 1, the columns t_{H} and t_{P} correspond to the timings for the HNF and Pugh's method, respectively.

Table 1. Implementation

Example	m	d	L	m_o	L_o	?Hyp	t_H	t_P
Tetrahedron	4	3	1	1	1	Yes	0.695	0.697
Cuboctahedron	14	3	2	1	2	Yes	1.855	1.846
Octahedron	8	3	1	1	1	Yes	1.357	1.357
TruncatedOctahedr.	14	3	3	1	1	Yes	1.995	1.977
TruncatedTetrahedr.	8	3	1	1	1	Yes	1.461	1.468
Presburger 1	3	2	2	1	1	Yes	0.083	0.082
Presburger 2	3	2	20	1	20	Yes	0.184	0.182
Presburger 3	3	2	18	3	4	Yes	0.287	0.260
Presburger 4	3	4	5	2	12	Yes	0.706	0.871
Presburger 6	4	5	89	6	35	Yes	0.893	0.746
Bounded 5	6	3	19	4	224	Yes	16.433	15.091
Bounded 7	8	3	19	3	190	No	138.448	239.637
Bounded 8	4	3	25	5	67	Yes	6.462	3.821
Bounded 9	6	3	18	6	74	No	23.574	16.763
Bounded 10	4	3	15	1	176	Yes	0.559	0.558
Unbounded 2	3	4	10	61	2255	No	0.547	0.600
Unbounded 3	4	4	20	1	20	No	0.981	0.987
Unbounded 4	6	5	2	1	2	No	0.722	0.510
Unbounded 5	5	4	8	1	8	No	1.321	1.319
Unbounded 6	10	4	8	1	8	No	1.494	1.479
P91	12	3	96	5	96	No	19.318	15.458
Sys_1	6	3	15	2	67	Yes	2.413	1.915
Sys_3	8	3	1	1	1	Yes	1.481	1.479
Automatic	8	2	999	1	999	Yes	0.552	0.549
Automatic2	6	4	1	1	2	Yes	1.115	1.113
Automatic3	3	4	1	1	1	Yes	0.130	0.135
Automatic4	3	5	1	1	1	Yes	0.227	0.232

From Table 1, we make a few observations:

1. Hypothesis 1 holds for most examples while it usually does not hold for random ones.
2. For 16 out of 27 examples, IntegerSolve produces a single component, which means that each such input polyhedron has no integer points in its grey shadow; this is, in particular, the case for regular polytopes and for examples from automatic parallelization.
3. When a decomposition consists of more than one component, most of those components are points; for example, the decomposition of Unbounded 2 has 61 components and 46 of them are points.

4. Coefficients of the output polyhedra are usually not much larger than the coefficients of the corresponding input polyhedron.
5. Among the challenging problems, some of them are solved faster when IntegerNormalize is based on HNF (e.g. Bounded 7) while others are solved faster when IntegerNormalize is based on Pugh's method (e.g. Bounded 9) which suggests that having both approaches at hand is useful.

To the best of our knowledge, there are two other published software libraries which are capable of describing the integer points of a polyhedron: one is `4ti2` [1] and the other is `Normaliz` [3]. Both softwares rely on Motzkin's theorem [8] which expresses any rational polyhedron as the Minkowski sum of a rational polytope and a rational cone. Hence, they do not decompose a polyhedron in the sense of our algorithm `IntegerSolve`.

Acknowledgements. The authors would like to thank IBM Canada Ltd (CAS project 880) and NSERC of Canada (CRD grant CRDPJ500717-16), as well as the University of Chinese Academy of Sciences, UCAS Joint PhD Training Program, for supporting their work.

References

1. 4ti2 team. 4ti2–a software package for algebraic, geometric and combinatorial problems on linear spaces. www.4ti2.de
2. Aubry, P., Lazard, D., Moreno Maza, M.: On the theories of triangular sets. J. Symb. Comput. **28**, 105–124 (1999)
3. Bruns, W., Ichim, B., Römer, T., Sieg, R., Söger, C.: Normaliz. Algorithms for rational cones and affine monoids. https://www.normaliz.uni-osnabrueck.de
4. Chen, C., Davenport, J.H., May, J.P., Moreno Maza, M., Xia, B., Xiao, R.: Triangular decomposition of semi-algebraic systems. J. Symb. Comput. **49**, 3–26 (2013)
5. Imbert, J.-L.: Fourier's elimination: which to choose? pp. 117–129 (1993)
6. Karmarkar, N.: A new polynomial-time algorithm for linear programming. In: Proceedings of the Sixteenth Annual ACM Aymposium on Theory of Computing. STOC 1984, pp. 302–311. ACM, New York, NY, USA (1984)
7. Khachiyan, L.: Fourier-motzkin elimination method. In: Floudas, C.A., Pardalos, P.M. (eds.) Encyclopedia of Optimization, pp. 1074–1077. Springer, Heidelberg (2009). doi:10.1007/978-0-387-74759-0_187
8. Motzkin, T.S.: Beiträge zur Theorie der linearen Ungleichungen. Azriel Press, Jerusalem (1936)
9. Pugh, W.: The omega test: a fast and practical integer programming algorithm for dependence analysis. In: Martin, J.L. (ed.), Proceedings Supercomputing 1991, Albuquerque, NM, USA, 18–22 November 1991, pp. 4–13. ACM (1991)
10. Schrijver, A.: Theory of Linear and Integer Programming. Wiley, New York (1986)