Divide and Decide:
How to solve equations of greatest common divisor

Rémy Défossez - supervised by Alessio Mansutti
IMDEA Software Institute - Campus Montegancedo - 28223 Pozuelo de Alarcón (Madrid) Spain

August 18th, 2023

The following is the report produced by the internship of Rémy Défossez at the IMDEA Software Institute, from February 2023 to June 2023, under the supervision of Alessio Mansutti, researcher at the IMDEA who had been working in formal logics since his PhD. The internship was fully financed by the IMDEA Software Institute. I would like to thank the help of Emanuele Giunta and Felix Ridoux, for the productive exchanges they had all along the internship; and of course to thank Alessio Mansutti, for his caring supervision, his sympathy and the many valuable lessons on research and logics he gave me and that could not fit in the following pages.

1 The EPAD problem

One of the main questions in formal logics is the decidability of a given system of logics, i.e. the existence of an algorithm that can determine if a given statement written in that system of logics is true or not. For example, Presburger arithmetics, i.e. first-order logic enriched with addition and order, has been proven to be decidable by Presburger in 1929 and in **2EXPSPACE** by Berman in 1980 [3]. On the other hand, Peano arithmetics, i.e. first-order logic with addition and multiplication, is commonly known to be undecidable.

Developing algorithms deciding formal logics, even if it is often a very theoretical work, has concrete applications, especially in formal testing and verification. For automatic bug-finding, a common technic is symbolic execution: it consists in running the program on a symbolic input x, and extracting a logical constraint $\varphi(x)$ which is true for any x if and only if the program is valid on any input. Depending on the syntax of the program, the constraint φ would be written in a particular logic. Since the global efficiency of the automatic tool depends on the choice of that logic, it is necessary to have the best algorithms of decidability of formal logics, in order to progress in the development of automatic tools.

One operation that can appear in these logical formulae after symbolic execution is the divisibility relation: the program may only succeed if the first component of the input divides its second component. Since this cannot be expressed in Presburger arithmetics, one needs to study the decidability of Presburger enriched with divisibility. Unfortunately, this can be proven to be undecidable (this follows from the undecidability of Peano arithmetics). Therefore, we need a weaker version of Presburger, keeping the addition. The solution would be to remove quantifiers.

Removing all quantifiers would lose too much expressiveness (since we cannot have variables anymore), therefore we chose to remove only the universal quantifier and the negation, obtaining a segment of Presburger Arithmetics called Existential Presburger Arithmetics. Deciding Existential Presburger is similar to solving systems of linear equations with multiple variables, and is thus

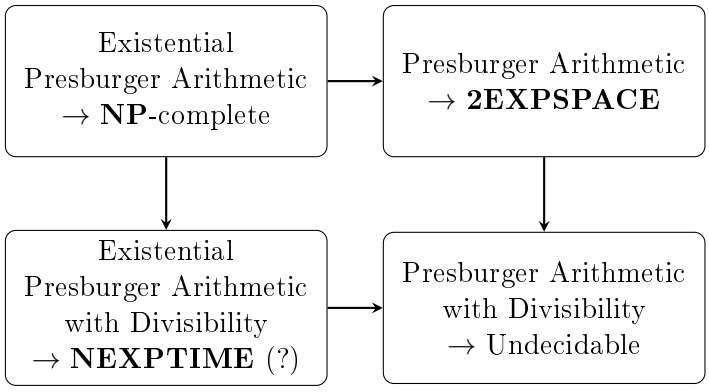

Figure 1: A summary of known decidabilities of EPAD and similar problems. An arrow indicates that which logics are fragments of which ones.

NP-complete. From that, we can define the Existential Presburger Arithmetic with Divisibility (EPAD), whose exact class of complexity remains an open problem. As said earlier, determining the exact class of complexity of EPAD would allow to optimize a lot of algorithms of verification of programs, but would also have applications in other domains, such as counter automata or word equations [12].

EPAD was first proven to be decidable by Leonard Lipshitz in 1978 [13], but his proof remained badly understood (due to being a bit too concise) until 2015, when Antonia Lechner, Joël Ouaknine and James Worrell extracted an algorithm from this proof in [11]. In the same paper, they proved that their algorithm functioned in **NEXPTIME**. To this day, it is not known if this upper bound can be optimized or if EPAD is **NEXPTIME**-complete. This open problem formed the mathematical context in which I started my internship.

2 My internship at the IMDEA Software Institute

When I arrived at the IMDEA Software Institute in Madrid, Alessio Mansutti, my supervisor during this five-month internship, presented me the EPAD problem, on which he had been working since August last year. Of course, there was no expectation of solving this problem that had remained opened for fifty years in five months, so we decided to focus our efforts on a seemingly easier problem (but still unsolved): Existential Presburger Arithmetic with GCD instead of Divisibility. Here, GCD stands for greatest common divisor: it is added to Existential Presburger with the binary relation $\gcd(x, y) = c$, which is true if and only if c is the greatest common divisor of x and y. Here, c is constant, whereas x and y can be quantified upon. This segment of logics is more expressive than Existential Presburger and less expressive than EPAD. The objective of the internship was to prove that it belonged to **NP** with a small model property, i.e. by proving that each satisfiable formula of this logic had at least one solution of polynomial size compared to the size of the formula, and therefore could be solved in **NP**. Alessio Mansutti had already proven that Existential Presburger with GCD was decidable in **NP**, but with an exponential bound on the size of the minimal solution. Bringing down that bound to a polynomial size was the goal of the internship. It was a success, and this constitutes the Theorem 1 of the Section 3.

To understand this problem, I spent the first month of my internship reading the litterature on the subject, and especially the algorithm of Antonia Lechner, Joël Ouaknine and James Worrell solving EPAD in **NEXPTIME**. After trying a few approaches, Alessio and I came to a conjecture

regarding this algorithm: if we could regroup the variables of the formula in blocks following some *increasing property* (a term yet to be defined), and then, if the number of blocks obtained this way was bounded by a constant (for example, gcd formulae could be converted in divisibility formulae whose number of blocks was bounded by 3), then the minimal solution of the formula was of polynomial size compared to the size of the formula. This conjecture is formally detailed in Section 3, as the Theorem 4.

From that point, the work became focused on proving this conjecture, which is also a direct upgrade of the algorithm computed by Lechner, Ouaknine, Worrell. I spent the second month of the internship trying to find all the improvements to this algorithm that would allow to get a polynomial bound, testing variations of all the concepts used in the proof. Finally, after more than a month of work, we had a first verion of an algorithm that constructed solutions of polynomial bit length for any satisfiable formula, subject to a small lemma that seemed intuitive at the time.

This lemma (Theorem 3 in the Section 3) extends on the Chinese Remainder Theorem, a fundamental lemma in number theory, which gives an upper bound to systems of congruences equations with one variable. The question of our lemma was: what bound do we get if we replace congruences with non-congruences? Since non-congruences equations have more solutions than congruences (there are more solutions to "$x \not\equiv 3 \mod 4$" than to "$x \equiv 3 \mod 4$"), one would expect the upper bound to be lower. The problem is easy to state, so I expected that this had already been solved and was a known result. Surprisingly, it was not. It was not an open problem neither: the problem just seemed too uninteresting to have any results published on this, so I had to prove this lemma myself. Unfortunately, it was way more complicated than the Chinese Remainder Theorem. I spent the third month on my internship on it, before finally proving its equivalence to a theorem of 1915: the Brun's pure sieve. With that being solved, we had finally a first proof of our algorithm.

The fourth month was essentially to write down the complete proof of that algorithm, without overlooking any passage that seemed trivial at first. As anyone with a background in math would expect, most of the parts that seemed trivial were not: we had to spend days, sometimes weeks, to prove lemmas that seemed obvious. The first draft was five pages long. In the end, the complete proof is about twenty-five pages long.

We spent the fifth month writing an article out of this proof, working together with Christoph Haase and Guillermo Perez, colleagues of Alessio Mansutti at respectively the University of Oxford and the University of Antwerp, that had worked with him on the EPAD problem. This month allowed me to see the whole process of building an article ready for submission out of a result, a process far from being easy. The submission date was coincidentally set on the last day of my internship, but we are far from being done with that article. On the day I redact this report, the article is submitted to the ACM-SIAM Symposium on Discrete Algorithms (SODA). If it gets accepted, I will be able to witness to whole publishing process, and present it next January to the SODA conference in Alexandria, United States.

3 Overview of main results

The following section is a more formal overview of the results of the internship, without going too much into detail (since the complete paper is sixty pages long). It is the ten first pages of the article that was submitted at the SODA conference, that covers exactly my internship and the results we got with Alessio Mansutti. As said earlier, variations of Existential Presburger arithmetics can be seen as systems of linear inequalities with multiple variables, i.e., an instance of integer programming. This is how we chose to present it in the following.

3.1 Introduction

Integer programming, the problem of finding an (optimal) solution over the integers to a systems of linear inequalities $A \cdot \boldsymbol{x} \leq \boldsymbol{b}$, is a central problem computer science and operations research. Feasibility of its 0-1 variant constituted one of Karp's 21 seminal NP-complete problems [9]. In the 1970s, membership of the unrestricted problem in NP was established independently by Borosh and Treybig [4], and von zur Gathen and Sieveking [21]. To show membership in NP, both groups of authors established a small witness property: if an instance of integer programming is feasible then there is a solution whose bit length is polynomially bounded in the size of the instance. Reductions to integer programming have become a standard tool to show membership of numerous problems in NP. In this paper, we study a non-linear generalization of integer programming which additionally allows to constrain the numerical value of the greatest common divisor (GCD) of two linear terms.

Throughout this paper, denote by $\mathbb{R}$ the set of real numbers, $\mathbb{Z}$ the set of integers, $\mathbb{N}$ the set of non-negative integers including zero, and $\mathbb{P}$ the set of all prime numbers. For $R \subseteq \mathbb{R}$, denote by $R_+ \coloneqq \{r \in R : r > 0\}$. Formally, an instance of integer programming with GCD constraints (IP-GCD) is a mathematical program of the following form:

$$\begin{aligned} \text{minimize} \quad & \boldsymbol{c}^\intercal \boldsymbol{x} \\ \text{subject to} \quad & A \cdot \boldsymbol{x} \leq \boldsymbol{b} \\ & \gcd(f_i(\boldsymbol{x}), g_i(\boldsymbol{x})) \sim_i d_i, \qquad\qquad 1 \leq i \leq k\,, \end{aligned}$$

where $\boldsymbol{c} \in \mathbb{Z}^n$, $A \in \mathbb{Z}^{m \times n}$, $\boldsymbol{b} \in \mathbb{Z}^m$, $d_i \in \mathbb{Z}_+$, $\boldsymbol{x} = (x_1, \ldots, x_n)$ is a vector of unknowns, the f_i and g_i are linear polynomials with integer coefficients, and $\sim_i \in \{\leq, =, \neq, \geq\}$. We call $\boldsymbol{a} \in \mathbb{Z}^n$ a solution if setting $\boldsymbol{x} = \boldsymbol{a}$ respects all constraints; $\boldsymbol{a}$ is an optimal solution if the value of $\boldsymbol{c}^\intercal \boldsymbol{a}$ is minimal among all solutions. We will first and foremost focus on the feasibility problem of IP-GCD and discuss finding optimal solutions later on in this paper. The main result of this paper is to establish a small witness property for IP-GCD and consequently membership of the problem in NP.

Theorem 1. *If an instance of IP-GCD is feasible then it has a solution (and an optimal solution, if one exists) of polynomial bit length. Hence, IP-GCD feasibility is* NP*-complete.*

We remark that IP-GCD feasibility is NP-hard even for a single variable, in contrast to classical integer programming, which is polynomial-time decidable for any fixed number of variables [8]. It is shown in [1, Theorem 5.5.7] that deciding a univariate system of non-congruences $x \not\equiv a_i \pmod{m_i}$, $1 \leq i \leq k$, is an NP-hard problem. Hardness of IP-GCD then follows from observing that a non-congruence $x \not\equiv a \pmod{m}$ is equivalent to $\gcd(x - a, m) \neq m$.

3.2 The NP upper bound at a glance

Even decidability of the IP-GCD feasibility problem is far from obvious, but can be approached by observing that deciding a GCD constraint is a *"Diophantine problem 'in disguise'"* [10]. It follows from Bézout's identity that $\gcd(x, y) = d$ if and only if there are $a, b, u, v \in \mathbb{Z}$ such that $u \cdot d = x$, $v \cdot d = y$, and $d = a \cdot x + b \cdot y$. While arbitrary systems of quadratic Diophantine equations are undecidable [15], we see that the unknowns a, b, u, v are only used to express divisibility properties. Hence, those equations can equivalently be expressed in the existential fragment of the first-order theory of the structure $L_{\text{div}} = (\mathbb{Z}, 0, 1, +, \leq, |)$, where $m \mid n$ holds whenever there exists a unique[1]

[1]This definition implies that $0 \mid n$ does not hold for any $n \in \mathbb{Z}$, 0 included. Throughout this paper, we assume wlog. that $f \neq 0$ for any divisibility $f \mid g$. For GCD, we instead use the standard interpretation where $\gcd(0, n) = n$ for any $n \in \mathbb{N}$; this mismatch between the interpretation of divisibility and GCD is for technical convenience only.

integer q such that $n = q \cdot m$:

$$u \cdot d = x \wedge v \cdot d = y \wedge d = a \cdot x + b \cdot y \iff \exists s\, \exists t\colon d \mid x \ \wedge\ d \mid y \ \wedge\ x \mid s \ \wedge\ y \mid t \ \wedge\ d = s + t\,.$$

The full first-order theory of L_{div} is easily seen to be undecidable [16]. However, decidability of its existential fragment was independently shown by Lipshitz [13, 14] and Bel'tyukov [2], and later also studied by van den Dries and Wilkie [20], Lechner et al. [11], and Starchak [18, 19]. The precise complexity of the existential fragment is a long-standing open problem. It is known to be NP-complete for a fixed number of variables [14, 11], and membership in NEXP has only more recently been established [11]. In particular, the bit length of smallest solutions can be exponential [11], as demonstrated by the family of formulae $\Phi_n \coloneqq x_n > 1 \wedge \bigwedge_{i=0}^{n-1} x_i > 1 \wedge x_i \mid x_{i+1} \wedge x_i + 1 \mid x_{i+1}$, for which any solution satisfies $x_n \geq 2^{2^n}$. From those results, it is possible to derive that IP-GCD feasibility is decidable in NEXP. However, IP-GCD does not require the full expressive power of L_{div}. In fact, the first-order theory of L_{div} can be seen to be equivalent to the theory of $(\mathbb{Z}, 0, 1, +, \leq, \gcd)$ in which the divisibility predicate is replaced by a full ternary relation $\gcd(x, y) = z$. In contrast, IP-GCD only requires countably many binary predicates $(\gcd(\cdot,\cdot) = d)_{d \in \mathbb{Z}_+}$ and $(\gcd(\cdot,\cdot) \geq d)_{d \in \mathbb{Z}_+}$ with the obvious interpretation. Several expressiveness results concerning (fragments of) the existential theory of the structure $(\mathbb{Z}, 0, 1, +, \leq, (\gcd(\cdot,\cdot) = d)_{d \in \mathbb{Z}_+})$ have recently been provided by Starchak [17]. The question of whether this theory admits solutions of polynomial bit length is explicitly stated as open in [17]. Theorem 1 answers this question positively.

Our starting point for establishing Theorem 1 is Lipshitz' [13, 14] decision procedure for the existential theory of L_{div} that was later refined by Lechner et al. [11]. Given a system of divisibility constraints $\Phi(\boldsymbol{x}) \coloneqq \bigwedge_{i=1}^{m} f_i(\boldsymbol{x}) \mid g_i(\boldsymbol{x})$ for linear polynomials f_i and g_i, Lipshitz' algorithm first computes from Φ an equi-satisfiable formula Ψ in so-called *increasing form*. Informally speaking, Ψ is in increasing form whenever Ψ is a system of divisibility constraints augmented with constraints imposing a total (semantic) ordering on the values of the variables in Ψ, and whenever the largest variable with respect to that ordering occurring in any non-trivial divisibility $f \mid g$ implied by Ψ only appears in the right-hand side g. For instance, the system $x < y \wedge x + 1 \mid y - 2$ is in increasing form, but adding $x + 1 \mid x + y$ results in a non-increasing system, since $x + 1 \mid y - 2 \wedge x + 1 \mid x + y$ implies $x + 1 \mid x + y - (y - 2)$, i.e., $x + 1 \mid x + 2$. Such implied divisibilities are captured in [11] by the notion of a *divisibility module* that we later formalize in Section 3.4. One conceptual contribution of this paper is to identify a weaker notion of formulae in increasing form that is syntactic in nature, as it does not explicitly enforce a particular ordering among the variables. Informally speaking, a system of divisibility constraints Ψ is *r-increasing* whenever there exists a partial order $\prec$ over the free variables of Ψ whose longest chain is of length at most $r - 1$, and for any non-trivial divisibility $f \mid g$ implied by Ψ, the set of variables occurring in $f \mid g$ has a $\prec$-maximal variable that only appears in the right-hand side g. Referring to the previous example, we observe that $x + 1 \mid y - 2$ is 2-increasing, witnessed by the (total) order $x \prec y$. This concept is fundamental for establishing Theorem 1, since, as we discuss below, for fixed r, any satisfiable r-increasing formula Ψ of L_{div} has a smallest solution of polynomial bit length, and L_{div} formulae resulting from IP-GCD instances are 3-increasing.

Returning to Lipshitz' approach, the key property of existential L_{div} formulae in increasing form is that they enable appealing to a local-to-global property: Lipshitz shows that any Φ in increasing form has a solution over $\mathbb{Z}$ if and only if Φ has a solution in the p-adic integers $\mathbb{Z}_p$ for every prime p belonging to a finite set of difficult primes $\mathbf{P}_+(\Phi)$, the other primes being "easy" in the sense that a p-adic solution for them always exists and that they do not influence the bit length of the minimal solution of Φ. In order to combine the p-adic solutions to an integer solution of Φ, Lipshitz invokes (a generalized version of) the Chinese Remainder Theorem (CRT):

Theorem 2 (CRT). *Let $M = \{m_1, \ldots, m_k\}$, $b_1, \ldots, b_k \in \mathbb{Z}$ be such that m_i and m_j are coprime for all $1 \leq i \neq j \leq k$. The system of simultaneous congruences $x \equiv b_i \bmod m_k$, $1 \leq i \leq k$, has a solution, and all solutions lie on the shifted lattice $a + \mathbb{Z} \cdot \Pi M$ for some $a \in \mathbb{Z}$.*

Here and below, for a finite set $M \subseteq \mathbb{Z}$, we denote by ΠM the product of all elements in M. It follows that the smallest non-negative solution of the system of congruences is of polynomial bit length. As a key technical contribution of this paper, required to establish Theorem 1, we develop the following Chinese-remainder-style theorem that includes additional non-congruences and yields a bound for the smallest solution that is, in certain settings, substantially better than the one that can be achieved by the CRT. For a finite set S, we write $\#S$ for its cardinality.

Theorem 3. *Let $d \in \mathbb{Z}_+$, $M \subseteq \mathbb{Z}_+$ finite, and $Q \subseteq \mathbb{P}$ be a non-empty finite set of primes such that the elements of $M \cup Q$ are pairwise coprime, $M \cap Q = \emptyset$, and $\min(Q) > d$. Consider the univariate system of simultaneous congruences and non-congruences $\mathcal{S}$ defined by*

$$\begin{aligned} x &\equiv b_m \pmod{m} && m \in M \\ x &\not\equiv c_{q,i} \pmod{q} && q \in Q,\ 1 \leq i \leq d\,. \end{aligned}$$

Then, for every $k \in \mathbb{Z}$, $\mathcal{S}$ has a solution in the interval $\{k, \ldots, k + \Pi M \cdot \mathfrak{f}(Q,d)\}$, where

$$\mathfrak{f}(Q,d) \coloneqq \big((d+1) \cdot \#Q\big)^{4(d+1)^2(3+\ln\ln(\#Q+1))}.$$

The strength of Theorem 3 can be seen as follows. While it is possible to deduce from the classical CRT that the solutions of $\mathcal{S}$ are periodic with period $\Pi Q \cdot \Pi M$, we have $\Pi Q \gg \mathfrak{f}(Q,d)$ as the magnitude of the primes in Q grows, as in particular $\mathfrak{f}(Q,d)$ only depends on $\#Q$ and d. We further discuss some results used to establish Theorem 3 in Section 3.3 below.

Another key technical contribution towards establishing Theorem 1 is to propose a refinement of the set of difficult primes $\mathbf{P}_+(\Phi)$. The definition of this set was changed from [13] to [11] to decrease its bit length from doubly to singly exponential. We refine the definition once more, and show that we obtain a set of polynomially many primes of polynomial bit length. This result is achieved by an in-depth analysis of how the integer solution for Φ is constructed starting from the p-adic solutions. The bound on $\mathbf{P}_+(\Phi)$ also enables us to derive an NP algorithm for increasing formulae. It is shown in [7] that, for every prime $p \in \mathbb{P}$, the existential theory of the p-adic integers with linear p-adic valuation constraints is decidable in NP. Deciding an increasing Φ thus reduces to a polynomial number of independent queries to an NP algorithm and is hence in NP. It is worth mentioning that the family of formulae Φ_n above is increasing only for the ordering $x_1 \prec x_2 \prec \cdots \prec x_n$ (i.e., it is n-increasing but not $(n-1)$-increasing). Hence, even though the smallest solution of Φ_n has exponential bit length, our bound on $\mathbf{P}_+(\Phi)$ enables us to witness the *existence* of a solution in NP.

Moreover, this bound leads to a further main result of this paper, showing that we can construct an integer solution for Φ from the relevant p-adic solutions that is asymptotically smaller when compared to the existing local-to-global approaches [13, 11]. These improved bounds also crucially rely on Theorem 3. To formally state this result, we require some further definitions. Given $\boldsymbol{v} \in \mathbb{Z}^d$, denote by $\|\boldsymbol{v}\|$ the maximum absolute value of the components of $\boldsymbol{v}$, and by $\langle\cdot\rangle$ the bit length encoding an object under some reasonable standard encoding in which numbers are encoded in binary. Furthermore, for a system of divisibility constraints $\Phi \coloneqq \bigwedge_{i=1}^{m} f_i \mid g_i$, denote by $\mathbb{P}(\Phi)$ the set of all primes that are less or equal than m or that divide some number occurring in Φ. For $p \in \mathbb{P}$ and $a \in \mathbb{Z} \setminus \{0\}$, we write $v_p(a)$ for the largest $k \in \mathbb{N}$ such that $a = p^k b$ for some $b \in \mathbb{Z}$, and $v_p(0) \coloneqq \infty$. We say that Φ has a solution modulo p if there is some $\boldsymbol{b}_p \in \mathbb{Z}^d$ such that $f_i(\boldsymbol{b}_p) \neq 0$ and $v_p(f_i(\boldsymbol{b}_p)) \leq v_p(g_i(\boldsymbol{b}_p))$ for all $1 \leq i \leq m$. Note that every integer solution is a solution modulo p for all $p \in \mathbb{P}$, and therefore if Φ does not have a solution modulo some prime p, then Φ is unsatisfiable

over $\mathbb{Z}$. The following theorem now gives bounds on the bit length of an integer solution of Φ in terms of solutions modulo p for primes in $\mathbb{P}(\Phi)$.

Theorem 4. *Let $\Phi(\boldsymbol{x})$ be an r-increasing system of divisibility constraints such that Φ has a solution $\boldsymbol{b}_p \in \mathbb{Z}^d$ modulo p for every prime $p \in \mathbb{P}(\Phi)$. Then Φ has infinitely many solutions, and a solution $\boldsymbol{a} \in \mathbb{N}^d$ such that $\langle\|\boldsymbol{a}\|\rangle \leq (\langle\Phi\rangle + \max\{\langle\|\boldsymbol{b}_p\|\rangle : p \in \mathbb{P}(\Phi)\})^{O(r)}$.*

The bound achieved in Theorem 4 primarily improves upon existing upper bounds by being exponential only in r, as opposed to exponential in $\mathrm{poly}(d)$ as established in [11], where d is the number of variables of Φ. In particular, for r fixed, as is the case for systems of divisibility constraints resulting from IP-GCD systems, Theorem 4 yields small solutions of polynomial bit length. Observe that Theorem 4 does not explicitly invoke the set of difficult primes $\mathbf{P}_+(\Phi)$, but rather the set $\mathbb{P}(\Phi)$. The latter is the subset of those primes p in $\mathbf{P}_+(\Phi)$ for which solutions modulo p might not exist, and one of the initial steps in the proof Theorem 4 is to compute solutions modulo q for every prime $q \in \mathbf{P}_+(\Phi) \setminus \mathbb{P}(\Phi)$. We give further details on the proof of Theorem 4 in Section 3.4 and then outline in Section 3.5 how it can be used to obtain the NP upper bound for Theorem 1. But first, we continue with the promised discussion on some details on Theorem 3.

3.3 Small solutions to systems of congruences and non-congruences

Let us introduce some notation. Given $a, b \in \mathbb{Z}$, we define $[a, b] \coloneqq \{a, a+1, \ldots, b\}$. We write $\mathrm{div}(a) \subseteq \mathbb{N}$ for the (positive) divisors of a and $\mathbb{P}(a)$ for $\mathbb{P} \cap \mathrm{div}(a)$. A function $m\colon \mathbb{Z}_+ \to \mathbb{R}_+$ is *multiplicative* if $m(a \cdot b) = m(a) \cdot m(b)$ for all $a, b \in \mathbb{N}$ coprime (so, $m(1) = 1$).

The proof of Theorem 3 is based on an abstract version of Brun's pure sieve [5]. Similarly to other results in sieve theory, Brun's pure sieve considers a finite set $A \subseteq \mathbb{Z}$ and a finite set of primes Q, and (subject to some conditions) derives bounds on the cardinality of the set $A \setminus \bigcup_{q \in Q} A_q$, where A_q is the subset of the elements in A that are divisible by q. In other words, the sieve studies the number of $x \in A$ satisfying $x \not\equiv 0 \pmod{q}$ for every $q \in Q$. In comparison, Theorem 3 requires x to be non-congruent modulo q to multiple integers, instead of non-congruent to just 0. The key insight in overcoming this difference is to notice that Brun's result can be established for arbitrary sets A_q, as long as a simple *independence* property holds together with Brun's *density* property (a formal statement is given below). A second technical issue concerns the bounds obtained from Brun's sieve. In its standard formulation (see e.g. [6, Ch. 6]), given an arbitrary $u \in \mathbb{Z}_+$, the sieve gives an estimate on the cardinality of the set $A \setminus \bigcup_{q \in Q \cap [2,u]} A_q$ that depends on u; and to estimate $\#\big(A \setminus \bigcup_{q \in Q} A_q\big)$ one sets u as the largest prime in Q. The resulting bound is, however, inapplicable in our setting as we seek to be independent of the bit length of the primes in Q. This issue is overcome by revisiting the analysis of Brun's pure sieve from [6], and by requiring an additional hypothesis: the multiplicative function $m\colon \mathbb{Z}_+ \to \mathbb{R}_+$ used to express Brun's *density* property must satisfy $m(q) \leq q - 1$ for all $q \in Q$. Those insights and requirements lead us to the following sieve.

Lemma 1. *Let $A \subseteq \mathbb{Z}$ and $Q \subseteq \mathbb{P}$ be non-empty finite sets, and let $n \coloneqq \Pi Q$ and $d \in \mathbb{Z}_+$. Consider a multiplicative function $m\colon \mathbb{Z}_+ \to \mathbb{R}_+$ satisfying $m(q) \leq q - 1$ on all $q \in Q$, and an (error) function $\sigma\colon \mathbb{N} \to \mathbb{R}$. Let $(A_r)_{r \in \mathrm{div}(n)}$ be a family of subsets of A satisfying the following two properties:*

independence: *$A_{r \cdot s} = A_r \cap A_s$, for every $r, s \in \mathrm{div}(n)$ coprime, and $A_1 = A$;*

density: *$\#A_r = \#A \cdot \frac{m(r)}{r} + \sigma(r)$, for every $r \in \mathrm{div}(n)$.*

Assume $|\sigma(r)| \leq m(r)$, and $m(q) \leq d$, for every $r \in \mathrm{div}(n)$ and $q \in Q$. Then,

$$\frac{1}{2} \cdot \#A \cdot W_m(Q) - \mathfrak{g}(Q, d) \;\leq\; \#\Big(A \setminus \bigcup\nolimits_{q \in Q} A_q\Big) \;\leq\; \frac{3}{2} \cdot \#A \cdot W_m(Q) + \mathfrak{g}(Q, d),$$

where $W_m(Q) \coloneqq \prod_{q\in Q}\left(1-\frac{m(q)}{q}\right)$ *and* $\mathfrak{g}(Q,d) \coloneqq (d\cdot \#Q)^{4(d+1)^2(2+\ln\ln(\#Q+1))+2}$.

Note that setting $A_r = \{a \in A : r \mid a\}$ for every $r \in \mathrm{div}(n)$, as usually done in sieve theory, results in a family of subsets of A satisfying the *independence* property. We defer the proof of Lemma 1 and only sketch here how to establish Theorem 3.

Proof sketch of Theorem 3. Below, the set of primes Q and $d \in \mathbb{Z}_+$ defined in the statement of Theorem 3 coincide with their homonyms in Lemma 1. Let $n \coloneqq \Pi Q$. By the CRT, the system of congruences $\forall m \in M$, $x \equiv b_m \pmod m$ has a solution set S_M that is a shifted lattice with period ΠM. Fix some $k \in \mathbb{Z}$. We consider the parametric set $B(z) \coloneqq [k, k+z] \cap S_M$, and find a small value for $z \in \mathbb{N}$ ensuring that $B(z)$ contains at least one solution to $\mathcal{S}$. To do so we rely on Lemma 1: we set $A \coloneqq B(z)$, and for every $q \in Q$, define $A_q \coloneqq \{a \in A : \text{there is } i \in [1,d] \text{ s.t. } a \equiv c_{q,i} \pmod q\}$. By definition, the sieved set $A \setminus \bigcup_{q\in Q} A_q$ corresponds to the set of solutions of $\mathcal{S}$ that belong in $[k, k+z]$. The definition of A_q is extended to every $r \in \mathrm{div}(n)$ not prime as $A_r \coloneqq A \cap \bigcap_{q\in\mathbb{P}(r)} A_q$. We establish that these sets satisfy the *independence* and *density* properties of Lemma 1, subject to the following multiplicative function: $m(r) \coloneqq \prod_{q\in\mathbb{P}(r)} \#\{c_{q,i} \mod q : i \in [1,d]\}$, i.e., $m(r)$ is the product of the number of distinct values $(c_{q,i} \mod q)$, for every $q \in \mathbb{P}(r)$. By hypothesis $\min(Q) > d$, hence $m(q) \le d \le q-1$ for every $q \in Q$. Furthermore, we show that m and the error function $\sigma(r) \coloneqq \#A_r - \#A \cdot \frac{m(r)}{r}$ satisfy the assumption $|\sigma(r)| \le m(r)$, for all $r \in \mathrm{div}(n)$. Hence, by Lemma 1, we obtain a lower bound on the sieved set $A \setminus \bigcup_{q\in Q} A_q$. Lastly, we show that taking $z = \mathfrak{f}(Q,d)$ makes the lower bound strictly positive, concluding the proof.

3.4 Small solutions to r-increasing systems of divisibility constraints

We now provide an overview on the technical machinery underlying Theorem 4. Our main goal here is to formalize the notion of difficult primes $\mathbf{P}_+(\Phi)$ and to sketch the proof of Theorem 4. We first need several key definitions and auxiliary notation. Subsequently, $\mathbb{Z}[x_1,\ldots,x_d]$ denotes the set of *linear* polynomials $f(x_1,\ldots,x_d) = a_1\cdot x_1 + \cdots + a_d \cdot x_d + c$, often written as $f(\boldsymbol{x}) = \boldsymbol{a}^\intercal\boldsymbol{x} + c$; when clear from the context, we omit the vector of variables $\boldsymbol{x}$ and write f instead of $f(\boldsymbol{x})$. The integers $a_1,\ldots,a_d$ are the *coefficients* of f, c is its *constant*. A polynomial f is *primitive* if it is non-zero and $\gcd(f) = 1$, where $\gcd(f) \coloneqq \gcd(a_1,\ldots,a_d,c)$. For any $b \in \mathbb{Z}$, we write $b\cdot f \coloneqq b\cdot \boldsymbol{a}^\intercal\boldsymbol{x} + b\cdot c$, and $\mathbb{Z}f \coloneqq \{b\cdot f : b \in \mathbb{Z}\}$. The *primitive part* of a polynomial g is the unique primitive polynomial f such that $g = \gcd(g)\cdot f$. Let $\Phi(\boldsymbol{x}) \coloneqq \bigwedge_{i=1}^m f_i(\boldsymbol{x}) \mid g_i(\boldsymbol{x})$ be a system of *divisibility constraints*. We let $\mathrm{terms}(\Phi) \coloneqq \{f_i, g_i : 1 \le i \le m\}$, and, given a finite sequence $\{(n_i, x_i)\}_{i\in I}$ of integer-variable pairs, write $\Phi[n_i \,/\, x_i : i \in I]$ for the system obtained from Φ by evaluating x_i as n_i, for all $i \in I$.

Divisibility modules and r-increasing form. As stated in Section 3.2, when dealing with a system of divisibility constraints $\Phi(\boldsymbol{x})$ one has to consider all divisibility constraints that are implied by Φ. This is done by relying on the notion of divisibility module. The *divisibility module* of a primitive polynomial f with respect to Φ, denoted by $\mathrm{M}_f(\Phi)$, is the smallest set such that (i) $f \in \mathrm{M}_f(\Phi)$; (ii) $\mathrm{M}_f(\Phi)$ is a $\mathbb{Z}$-module, i.e., $\mathrm{M}_f(\Phi)$ is closed under integer linear combinations; and (iii) if $g \mid h$ is a divisibility constraint in Φ and $b\cdot g \in \mathrm{M}_f(\Phi)$ for some $b \in \mathbb{Z}$, then $b\cdot h \in \mathrm{M}_f(\Phi)$. The following property holds: for every $g \in \mathrm{M}_f(\Phi)$ and solution $\boldsymbol{a}$ to Φ, the integer $f(\boldsymbol{a})$ divides $g(\boldsymbol{a})$. The divisibility module $\mathrm{M}_f(\Phi)$ is a vector subspace, hence it is spanned by linear polynomials $h_1,\ldots,h_\ell \in \mathbb{Z}[x_1,\ldots,x_d]$, that is $\mathrm{M}_f(\Phi) = \mathbb{Z}h_1 + \cdots + \mathbb{Z}h_\ell$; where $+$ is the Minkowski sum.

We can now formalize the key concept of r-increasing formula. Let $\prec$ be a syntactic order on variables $\boldsymbol{x} = (x_1,\ldots,x_d)$. Given $f \in \mathbb{Z}[x_1,\ldots,x_d]$, we write $\mathrm{LV}_\prec(f)$ for the *leading variable* of f, that is the variable with non-zero coefficient in f that is maximal wrt. $\prec$; if f is constant then

$\mathrm{LV}_\prec(f) \coloneqq \bot$, and we postulate $\bot \prec x_i$ for all $1 \leq i \leq d$. We omit the subscript $\prec$ when it is clear from the context. A system of divisibility constraints Φ is in *increasing form (wrt. $\prec$)* whenever $\mathrm{M}_f(\Phi) \cap \mathbb{Z}[x_1, \ldots, x_k] = \mathbb{Z}f$ for every primitive polynomial f with $\mathrm{LV}(f) = x_k$, for every $1 \leq k \leq d$. Given a partition $X_1, \ldots, X_r$ of the variables $\boldsymbol{x}$, we write $(X_1 \prec \cdots \prec X_r)$ for the set of all orders $\prec$ on $\boldsymbol{x}$ with the property that for any two x, x', if $x \in X_i$ and $x' \in X_j$ for some $i < j$ then $x \prec x'$.

Definition 1. *A system of divisibility constraints $\Phi(\boldsymbol{x})$ is* r-increasing *if there exists a partition $X_1, \ldots, X_r$ of $\boldsymbol{x}$ such that Φ is in increasing form wrt. every ordering $\prec$ in $(X_1 \prec \cdots \prec X_r)$.*

Observe that for any $\prec$ from $(X_1 \prec \cdots \prec X_r)$, we have that for every primitive linear polynomial f and $g \in \mathrm{M}_f(\Phi)$, if $g \notin \mathbb{Z}f$ then $\mathrm{LV}_\prec(f) \in X_i$ and $\mathrm{LV}_\prec(g) \in X_j$ for some $i < j$.

The elimination property and S-terms. To handle systems in increasing form, two more concepts are required in the context of the local-to-global property. First, to compute the "global" integer solution starting from the "local" solutions modulo primes, the divisibility modules of all primitive parts of polynomials in a system of divisibility constraints Φ need to be taken into account. One way to do this, introduced in [11], is to add bases for these modules directly to Φ. This leads to the notion of elimination property: $\Phi(\boldsymbol{x})$ has the *elimination property* for the order $x_1 \prec \cdots \prec x_d$ of the variables in $\boldsymbol{x}$ whenever for every primitive part f of a polynomial appearing in the left-hand side of some divisibility in Φ, and for every $0 \leq k \leq d$, $\{g : \mathrm{LV}(g) \preceq x_k \text{ and } f \mid g \text{ appears in } \Phi\}$ is a set of linearly independent polynomials that forms a basis for $\mathrm{M}_f(\Phi) \cap \mathbb{Z}[x_1, \ldots, x_k]$, where $x_0 \coloneqq \bot$. We show that closing a formula under the elimination property can be done in polynomial time.

Lemma 2. *There is a polynomial-time algorithm that, given a system of divisibility constraints $\Phi(\boldsymbol{x}) \coloneqq \bigwedge_{i=1}^{m} f_i \mid g_i$ and an order $x_1 \prec \cdots \prec x_d$ for $\boldsymbol{x}$, computes $\Psi(\boldsymbol{x}) \coloneqq \bigwedge_{i=1}^{n} f'_i \mid g'_i$ with the elimination property for $\prec$ that is equivalent to $\Phi(\boldsymbol{x})$, both over $\mathbb{Z}$ and modulo each $p \in \mathbb{P}$.*

In a nutshell, for every primitive part f of a polynomial appearing in the left-hand side of a divisibility in Φ, the algorithm first computes a finite set S spanning $\mathrm{M}_f(\Phi)$. The algorithm then uses the Hermite normal form of a matrix, whose entries are the coefficients and constant of the elements of S, to obtain linearly independent polynomials $h_1, \ldots, h_\ell$ with different leading variables with respect to $\prec$. The system Ψ is then obtained by replacing divisibility constraints of the form $f \mid g$ appearing in Φ with the divisibilities $f \mid h_1, \ldots, f \mid h_\ell$.

The second concept is related to how Theorem 4 is proven. In a nutshell, in the proof we iteratively assign values to the variables in a way that guarantees the system of divisibility constraints to stay in increasing form. To do that, additional polynomials need to be considered. For an example, consider the following system of divisibility constraints Φ in increasing form for the order $u \prec v \prec x \prec y \prec z$, and with the elimination property for that order:

$$\Phi \;\coloneqq\; v \mid u + x + y \;\wedge\; v \mid x \;\wedge\; y + 2 \mid z + 1 \;\wedge\; v \mid z\,.$$

From the first two divisibility constraints, we have $(u + y) \in \mathrm{M}_v(\Phi)$; i.e., $(u - 2) + (y + 2) \in \mathrm{M}_v(\Phi)$. Therefore, if u were to be instantiated as 2, the resulting formula Φ' would satisfy $(y + 2) \in \mathrm{M}_v(\Phi')$ and hence $(z + 1) \in \mathrm{M}_v(\Phi')$, from the third divisibility constraint. Then, $1 \in \mathrm{M}_v(\Phi')$ would follow from the last divisibility, violating the constraints of the increasing form. The reason why increasingness is lost when setting $u = 2$ stems from the fact that in Φ' we have an implied divisibility $v \mid y + 2$, where $y + 2$ is a left-hand side that was not present in $\mathrm{M}_v(\Phi)$. We can avoid this problem by considering the polynomial $u - 2$ and forcing it to be non-zero. The main issue is then to identify all such problematic polynomials, which is done with the following notion of S-terms. Less refined versions of this notion, as considered in [13, 11], result in exponentially larger sets of polynomials.

Given polynomials $f(\boldsymbol{x})$ and $g(\boldsymbol{x})$ with $\mathrm{LV}(f) = x_l$ and $\mathrm{LV}(g) = x_k$, we define their *S-polynomial* $S(f,g) \coloneqq b_k \cdot f - a_l \cdot g$, where a_l and b_k are coefficients of x_l in f and x_k in g, respectively. For constant f (resp. g), i.e., $\mathrm{LV}(f) = \bot$, above $a_l \coloneqq f$ (resp. $b_k \coloneqq g$). Note that if f and g are non-constant and $\mathrm{LV}(f) = \mathrm{LV}(g)$ then $\mathrm{LV}(S(f,g)) \prec \mathrm{LV}(f)$. For any $X \subseteq \mathbb{Z}[x_1, \ldots, x_n]$, we define $S(X) \coloneqq X \cup \{S(f,g) : f, g \in X\}$. Given a system of divisibility constraints Φ with the elimination property for $\prec$ and a primitive polynomial f, we define the set of *S-terms for* f, denoted as $\mathrm{S}_f(\Phi)$, to be the smallest set such that (i) $\mathrm{terms}(\Phi) \subseteq \mathrm{S}_f(\Phi)$, and (ii) if $f \mid g$ occurs in Φ and $h \in \mathrm{S}_f(\Phi)$ with $\mathrm{LV}(g) = \mathrm{LV}(h)$, then $S(g,h) \in \mathrm{S}_f(\Phi)$. We write $\Delta(\Phi)$ for the set of all S-terms for f, where f is any primitive part of a polynomial in $\mathrm{terms}(\Phi)$.

The set of difficult primes. We now turn towards identifying a small set of difficult primes $\mathbf{P}_+(\Phi)$ of polynomial bit length. There are two categories of difficult primes: those for which a solution to Φ modulo p is not guaranteed to exist, and those for which such a solution always exists, but which still influences the size of the minimal integer solution for Φ. The former is the set $\mathbb{P}(\Phi)$ defined in Section 3.2. The next lemma shows that Φ has a solution modulo any prime not in $\mathbb{P}(\Phi)$.

Lemma 3. *Let* $\Phi(\boldsymbol{x}) \coloneqq \bigwedge_{i=1}^{m} f_i \mid g_i$ *and* $p \in \mathbb{P} \setminus \mathbb{P}(\Phi)$. *Then,* Φ *has a solution* $\boldsymbol{b} \in \mathbb{N}^d$ *modulo* p *such that* $v_p(f_i(\boldsymbol{b})) = 0$ *for every* $1 \leq i \leq m$, *and* $\|\boldsymbol{b}\| \leq p - 1$.

In a nutshell, $v_p(f_i(\boldsymbol{b})) = 0$ holds if and only if $f_i(\boldsymbol{b}) \not\equiv 0 \pmod{p}$, meaning that the solution $\boldsymbol{b}$ can be computed by considering a system of at most m non-congruences; one for each left-hand side of Φ. Consider an ordering $\prec$ of the variables in $\boldsymbol{x}$. Since $p \notin \mathbb{P}(\Phi)$, p does not divide any coefficient or constant appearing in some f_i. This means that if $f_i(\boldsymbol{x}) = f_i' + a \cdot x$, with $x = \mathrm{LV}_{\prec}(f_i)$, we can rewrite $f_i(\boldsymbol{x}) \not\equiv 0 \pmod{p}$ as $x \not\equiv -a^{-1} f_i' \pmod{p}$, where a^{-1} is the inverse of a modulo p. Then, since $p > m$, one can find $\boldsymbol{b}$ by picking suitable residues in $\{0, \ldots, p-1\}$; this can be done inductively, starting from the $\prec$-minimal variable.

Extending $\mathbb{P}(\Phi)$ into $\mathbf{P}_+(\Phi)$, hence capturing the second of the two categories above, is a delicate matter. In fact, while $\mathbb{P}(\Phi)$ is defined for an arbitrary system of divisibility constraints, the set $\mathbf{P}_+(\Phi)$ can only meaningfully be defined on systems that have the elimination property for an order $\prec$. For systems without the elimination property, one must first appeal to Lemma 2. Let Φ be a system of divisibility constraints with the elimination property. The set of *difficult primes* $\mathbf{P}_+(\Phi)$ is the set of primes $p \in \mathbb{P}$ satisfying at least one the following conditions:

(P1) $p \leq \#S(\Delta(\Phi))$,

(P2) p divides any non-zero coefficient or constant of a polynomial in $S(\Delta(\Phi))$, or

(P3) p divides the smallest (in absolute value) non-zero $\lambda \in \mathbb{Z}$ such that $\lambda \cdot g \in \mathrm{M}_f(\Phi)$ for some primitive polynomial f occurring in Φ and $g \in \mathrm{S}_f(\Phi)$ (if such a λ exists).

Note that (P1) and (P2) imply $\mathbb{P}(\Phi) \subseteq \mathbf{P}_+(\Phi)$. The following lemma establishes bounds on these two sets that are central to the proof of Theorem 4.

Lemma 4. *Consider a system of divisibility constraints* $\Phi(\boldsymbol{x})$ *in* d *variables. Then, the set of primes* $\mathbb{P}(\Phi)$ *satisfies* $\log_2(\Pi\mathbb{P}(\Phi)) \leq m^2(d+2) \cdot (\langle\|\Phi\|\rangle + 2)$. *Furthermore, if* Φ *has the elimination property for an order* $\prec$ *on* $\boldsymbol{x}$, *then the set of primes* $\mathbf{P}_+(\Phi)$ *satisfies* $\log_2(\Pi\mathbf{P}_+(\Phi)) \leq 64 \cdot m^5(d+2)^4(\langle\|\Phi\|\rangle + 2)$.

Note that $\langle S \rangle = O(\log_2(\Pi S))$ for any finite set S of positive integers, and therefore the above lemma bounds $\langle \mathbb{P}(\Phi) \rangle$ and $\langle \mathbf{P}_+(\Phi) \rangle$ polynomially.

Proof sketch of Theorem 4. Recall that Theorem 4 establishes a local-to-global property for r-increasing systems of divisibility constraints $\Phi(\boldsymbol{x})$: if such a system has a solution $\boldsymbol{b}_p \in \mathbb{Z}^d$ modulo p for every prime $p \in \mathbb{P}(\Phi)$, then it has infinitely many integer solutions, and a solution $\boldsymbol{a} \in \mathbb{N}^d$ such that $\langle \|\boldsymbol{a}\| \rangle \leq (\langle \Phi \rangle + \max\{\langle \|\boldsymbol{b}_p\| \rangle : p \in \mathbb{P}(\Phi)\})^{O(r)}$. We give a high-level overview of the proof of this result, focusing on the part of the statement that constructs a solution over $\mathbb{N}$. Fix an order $\prec$ in $X_1 \prec \cdots \prec X_r$. We compute a map $\boldsymbol{\nu} \colon \left(\bigcup_{j=1}^{r} X_j\right) \to \mathbb{Z}_+$ such that $\boldsymbol{\nu}(\boldsymbol{x})$ is a solution for Φ by induction on r, populating $\boldsymbol{\nu}$ according the order $\prec$.

If $r = 1$, the system Φ is of the form $\bigwedge_{i=1}^{\ell} c_i \mid g_i(\boldsymbol{x}) \wedge \bigwedge_{j=\ell+1}^{m} f_j(\boldsymbol{x}) \mid a_j \cdot f_j(\boldsymbol{x})$, with $c_i \in \mathbb{Z} \setminus \{0\}$ and $a_j \in \mathbb{Z}$, and $\boldsymbol{\nu}$ can be computed using the CRT. Given $p \in \mathbb{P}(\Phi)$, one considers the natural number $\mu_p \coloneqq \max\left\{v_p(f(\boldsymbol{b}_p)) : f(\boldsymbol{x}) \text{ left-hand side of a divisibility in } \Phi\right\}$, which determines up to what power of p the integer solution given by $\boldsymbol{\nu}$ has to agree with the solution $\boldsymbol{b}_p$. Then, the CRT instance to be solved is $x_k \equiv b_{p,k} \pmod{p^{\mu_p+1}}$ for every $p \in \mathbb{P}(\Phi)$ and $1 \leq k \leq d$, where $x_1 \prec \cdots \prec x_d$ are the variables in Φ and $b_{p,1}, \ldots, b_{p,d}$ are their related values in $\boldsymbol{b}_p$.

When $r \geq 2$, the construction is much more involved. The goal is to define $\boldsymbol{\nu}$ for the variables in X_1 in such a way that the formula $\Phi' \coloneqq \Phi[\boldsymbol{\nu}(x) \,/\, x : x \in X_1]$ is increasing for $X_2 \prec \cdots \prec X_r$, and has solutions modulo p for every $p \in \mathbb{P}(\Phi')$. This allows us to invoke Theorem 4 inductively, obtaining a solution $\boldsymbol{\xi} \colon \left(\bigcup_{j=2}^{r} X_j\right) \to \mathbb{Z}_+$ for Φ'. An integer solution for Φ is then given by the union $\boldsymbol{\nu} \sqcup \boldsymbol{\xi}$ of $\boldsymbol{\nu}$ and $\boldsymbol{\xi}$, i.e., the map defined as $\boldsymbol{\nu}(x)$ for $x \in X_1$ and as $\boldsymbol{\xi}(y)$ for $y \in \bigcup_{j=2}^{r} X_j$. To construct $\boldsymbol{\nu}$ for X_1, we first close Φ under the elimination property following Lemma 2, obtaining an equivalent system Ψ, and extend the solutions $\boldsymbol{b}_p$ to every $p \in \mathbf{P}_+(\Psi)$ thanks to Lemma 3. We then populate $\boldsymbol{\nu}$ following the order $\prec$, starting from the smallest variable. In the proof, this is done with a second induction. Values for the variables in X_1 are found using Theorem 3. When a new value $a_k \in \mathbb{Z}_+$ for a variable $x_k \in X_1$ is found, new primes need to be taken into account, since substituting a_k for x_k yields a complete evaluation of the polynomials in $S(\Delta(\Phi))$ with leading variable x_k, i.e., these polynomials become integers that may be divisible by primes not belonging to $\mathbf{P}_+(\Psi)$. For subsequent variables in X_1, we make sure to pick values that keep the evaluated polynomials as "coprime as possible" with respect to these new primes. This condition is necessary to obtain the new solutions $\boldsymbol{b}_p$ for the formula Φ', modulo every $p \in \mathbb{P}(\Phi')$. The precise system of (non-)congruences considered when computing x_k is

$$\begin{cases} x_k \equiv b_{p,k} & (\text{mod } p^{\mu_p+1}) \quad p \in \mathbf{P}_+(\Psi) \\ g(\boldsymbol{\nu}(\boldsymbol{y}), x_k) \not\equiv 0 & (\text{mod } q) \quad q \in Q \setminus \mathbf{P}_+(\Psi),\ g(\boldsymbol{y}, x_k) \in S(\Delta(\Psi)) \text{ with } \mathrm{LV}_{\prec}(g) = x_k \end{cases}$$

where Q is the set of new primes obtained when fixing the variables $\boldsymbol{y} = (x_1, \ldots, x_{k-1})$, and $\mu_p \coloneqq \max\left\{v_p(f(\boldsymbol{b}_p)) : f(\boldsymbol{x}) \text{ left-hand side of a divisibility in } \Psi\right\}$. Theorem 3 can be applied on the system above because primes in $Q \setminus \mathbf{P}_+(\Psi)$ do not satisfy the properties (P1) and (P2).

To show that Theorem 4 can be applied inductively on Φ', we rely on (P3) and the elimination property of Ψ to show that Φ' has solutions modulo every $p \in \mathbb{P}(\Phi')$, and on properties of S-terms and again on the elimination property of Ψ to show that Φ' is increasing for $X_2 \prec \cdots \prec X_r$.

3.5 Solving an instance of IP-GCD

We now briefly discuss the proof of Theorem 1. In a nutshell, this result is shown by giving an algorithm that reduces an *IP-GCD system* $\Phi(\boldsymbol{x}) \coloneqq A \cdot \boldsymbol{x} \leq \boldsymbol{b} \wedge \bigwedge_{i=1}^{k} \gcd(f_i(\boldsymbol{x}), g_i(\boldsymbol{x})) \sim_i c_i$ into an equi-satisfiable disjunction of several 3-increasing systems of divisibility constraints with coefficients and constants of polynomial bit length. We then study bounds on the solutions of each of these systems modulo the primes required by the local-to-global property, and conclude that IP-GCD has a small witness property over the integers directly from Theorem 4.

Our arguments heavily rely on syntactic properties of the systems of divisibility constraints we obtain when translating an IP-GCD system Φ. These syntactic properties are captured in the complete proof with the notion of *gcd-to-div* triple. The formal definition is rather lengthy, for this overview it suffices to know that a triple $(\Psi, \boldsymbol{u}, E)$ is a gcd-to-div triple if Ψ is a system of divisibility constraints in which all numbers appearing are positive, and $\boldsymbol{u}$ and E are a vector and a matrix that act as a change of variables between the variables in Ψ and the variables in Φ. The following proposition formalizes the role of gcd-to-div triples.

Proposition 1. *Let Φ be an IP-GCD system in d variables. There is a set C of gcd-to-div triples such that the set of integer solutions to Φ is $\{\boldsymbol{u} + E \cdot \boldsymbol{\lambda} : (\Psi, \boldsymbol{u}, E) \in C$ and $\boldsymbol{\lambda} \in \mathbb{N}^m$ solution to $\Psi\}$. Every $(\Psi, \boldsymbol{u}, E) \in C$ has bit length polynomial in $\langle \Phi \rangle$ and is such that Ψ is in 3-increasing form.*

Above, m is the number of free variables in Ψ, which is also the number of columns in E. The algorithm showing this proposition, performs a series of equivalence-preserving syntactic transformations of Φ that are mainly divided into two steps: we first compute from Φ a set of gcd-to-div triples B satisfying $\{\boldsymbol{x} \in \mathbb{Z}^d : \boldsymbol{x}$ solution to $\Phi\} = \{\boldsymbol{u} + E \cdot \boldsymbol{\lambda} : (\Psi, \boldsymbol{u}, E) \in B$ and $\boldsymbol{\lambda} \in \mathbb{N}^m$ solution to $\Psi\}$, and then obtains C by manipulating every system of divisibility constraints in B to make it 3-increasing. Below we give a summary of these two steps.

Step I: from IP-GCD to divisibility constraints. This step is split into three sub-steps:

1. Reduce the input IP-GCD system Φ into an equi-satisfiable disjunction of IP-GCD system having GCD of the form $\gcd(f(\boldsymbol{x}), g(\boldsymbol{x})) = c$ or $\gcd(f(\boldsymbol{x}), g(\boldsymbol{x})) \geq c$, and a system of inequalities $A \cdot \boldsymbol{x} \leq \boldsymbol{b}$ fixing a sign for every polynomial $h(\boldsymbol{x})$ appearing in a GCD constraint, i.e., $A \cdot \boldsymbol{x} \leq \boldsymbol{b}$ has either $h(\boldsymbol{x}) \leq -1$ or $h(\boldsymbol{x}) \geq 1$ as a row.

2. Let G be the set of systems computed at the previous step. The algorithm erases the system of inequalities $A \cdot \boldsymbol{x} \leq \boldsymbol{b}$ from every IP-GCD system $\Psi \in G$ by performing a change of variables. In particular, relying on a well-known result by von zur Gathen and Sieveking [21], the algorithm computes a finite set $\{(\boldsymbol{u}_i, E_i) : i \in I_\Psi\}$ such that $\{\boldsymbol{x} \in \mathbb{Z}^d : A \cdot \boldsymbol{x} \leq \boldsymbol{b}\} = \{\boldsymbol{u}_i + E_i \cdot \boldsymbol{\lambda} : \boldsymbol{\lambda} \in \mathbb{N}^m, i \in I_\Psi\}$. For every $i \in I_\Psi$, the algorithm constructs a system of GCD constraints Ψ_i by replacing $\boldsymbol{x}$ in all GCD constraints of Ψ with $\boldsymbol{u}_i + E_i \cdot \boldsymbol{y}$, where $\boldsymbol{y}$ is a family of fresh variables. The latter transformation also ensures that all numbers in the Ψ_i are positive.

3. The algorithm translates every GCD constraint in every Ψ_i into a divisibility. Each constraint $\gcd(f(\boldsymbol{y}), g(\boldsymbol{y})) = c$ is replaced by $\exists z \in \mathbb{N} : \; c \mid f \wedge c \mid g \wedge f \mid z \wedge g \mid z + c$, following Bézout's identity, whereas $\gcd(f(\boldsymbol{y}), g(\boldsymbol{y})) \geq c$ becomes $\exists z \in \mathbb{N} : \; z + c \mid f \wedge z + c \mid g$. The triple $(\Psi_i, \boldsymbol{u}_i, E_i)$ obtained after these replacements is a gcd-to-div triple.

Step II: enforcing increasingness. The algorithm considers each gcd-to-div triple $(\Psi, \boldsymbol{u}, E)$ computed in the previous step and further manipulates it, producing a set of gcd-to-div triples D having only systems of divisibility constraints in 3-increasing form, and satisfying

$$\{\boldsymbol{u} + E \cdot \boldsymbol{\lambda} : \boldsymbol{\lambda} \in \mathbb{N}^m \text{ solution for } \Psi\} = \{\boldsymbol{u}' + E' \cdot \boldsymbol{\lambda} : (\Psi', \boldsymbol{u}', E') \in D, \boldsymbol{\lambda} \in \mathbb{N}^{m'} \text{ solution for } \Psi'\}. \quad (1)$$

The set D is computed as follows. If Ψ is already 3-increasing, then $D \coloneqq \{(\Psi, \boldsymbol{u}, E)\}$. Otherwise, properties of gcd-to-div triples ensure that there is a non-constant primitive polynomial f with positive coefficients and constant such that $\mathrm{M}_f(\Psi) \cap \mathbb{Z} \neq \{0\}$. The algorithm computes the smallest positive integer c belonging to $\mathrm{M}_f(\Psi)$. We have that Ψ entails $f \mid c$. Let $\lambda_1, \ldots, \lambda_j$ be all the

variables in f. Since the coefficients and constant of f are all positive and variables are now interpreted over the naturals, such a divisibility constraint can only be satisfied by assigning to each variable an integer in $[0, c]$. The algorithm iterates over each assignment $\boldsymbol{\nu}\colon \{\lambda_1, \ldots, \lambda_j\} \to [0, c]$ satisfying $f \mid c$, computing from $(\Psi, \boldsymbol{u}, E)$ the gcd-to-div triple $(\Psi_{\boldsymbol{\nu}}, \boldsymbol{u}_{\boldsymbol{\nu}}, E_{\boldsymbol{\nu}})$ where $\Psi_{\boldsymbol{\nu}} \coloneqq \Psi[\boldsymbol{\nu}(\lambda_i) \,/\, \lambda_i : i \in [1, j]]$, and $\boldsymbol{u}_{\boldsymbol{\nu}}$ and $E_{\boldsymbol{\nu}}$ are obtained from $\boldsymbol{u}$ and E based on $\boldsymbol{\nu}$ too. All such triples are added to D to replace $(\Psi, \boldsymbol{u}, E)$. However, some newly added system $\Psi_{\boldsymbol{\nu}}$ may not be 3-increasing. If that is the case, Step II is iteratively performed on $(\Psi_{\boldsymbol{\nu}}, \boldsymbol{u}_{\boldsymbol{\nu}}, E_{\boldsymbol{\nu}})$. Termination is guaranteed because $\Psi_{\boldsymbol{\nu}}$ has strictly fewer variables than Ψ and the set of computed gcd-to-div triples is the set C from Proposition 1.

Bounds on the solutions modulo primes and proof sketch of Theorem 1. Following Proposition 1, what is left to apply Theorem 4 is to compute the solutions modulo primes in $\mathbb{P}(\Psi)$, for all $(\Psi, \boldsymbol{u}, E) \in C$. We rely on properties of gcd-to-div triples to show the result below.

Lemma 5. *Let $(\Psi, \boldsymbol{u}, E)$ be a gcd-to-div triple in which Ψ has d variables, and consider $p \in \mathbb{P}(\Psi)$. If Ψ has a solution modulo p, then it has a solution $\boldsymbol{b}_p \in \mathbb{Z}^d$ modulo p with $\|\boldsymbol{b}_p\| \leq (d+1) \cdot \|\Psi\|^3 p^2$.*

Proposition 1, and Lemmas 4 and 5 imply the part of Theorem 1 not concerning optimization as a corollary of Theorem 4. For optimization, consider a linear objective $\boldsymbol{c}^\intercal \boldsymbol{x}$ to be minimized (the argument is analogous for maximization) subject to an IP-GCD system $\Phi(\boldsymbol{x})$, and let C be the set of gcd-to-div triples computed from Φ following Proposition 1. We can show the following characterization that implies the optimization part of Theorem 1: an optimal solution exists if and only if (i) there is $(\Psi, \boldsymbol{u}, E) \in C$ such that Ψ satisfiable over $\mathbb{N}$, and (ii) for every $(\Psi, \boldsymbol{u}, E) \in C$ with Ψ satisfiable over $\mathbb{N}$, $\boldsymbol{c}^\intercal(\boldsymbol{u} + E \cdot \boldsymbol{\lambda})$ has no variable with a strictly negative coefficient. Moreover, if there is an optimal solution, then there is one with polynomial bit length with respect to $\langle \Phi \rangle$ and $\langle \boldsymbol{c} \rangle$. Briefly, the double implication comes from the fact that the construction required to establish Theorem 4 also shows that for each variable in $\boldsymbol{\lambda}$ there are infinitely many values that yield a solution to Ψ, both in the positive and negative direction, and therefore the existence of a variable in $\boldsymbol{c}^\intercal(\boldsymbol{u} + E \cdot \boldsymbol{\lambda})$ having a negative coefficient entails the non-existence of an optimum. For the bound, one shows that $\min\{\boldsymbol{c}^\intercal \boldsymbol{u} : (\Psi, \boldsymbol{u}, E) \in C\}$ is a lower bound to every solution of Φ. Then, the polynomial bound follows directly from Proposition 1.

3.6 Conclusion and future work

We have established a polynomial small witness property for integer programming with additional GCD constraints over linear polynomials. Our work also sheds new light on the feasibility problem for systems of divisibility constraints between linear polynomials over the integers, and more broadly on the existential fragment of the first-order theory of the structure $L_{\text{div}} = (\mathbb{Z}, 0, 1, +, \leq, |)$, which is known to be NP-hard and decidable in NEXP [14, 11].

Our work may also enable obtaining improved complexity results for other problems that reduce to the existential theory of L_{div}. For instance, [12] Lin and Majumdar reduce deciding a special class of word equations with length constraints and regular constraints to existential L_{div}, hence obtaining an NEXP for their problem. The formulas resulting from their reduction are of a special shape, and showing them to be r-increasing for some fixed r would directly yield a PSPACE decision procedure for the aforementioned class of word equations.

References

[1] Eric Bach and Jeffrey Shallit. *Algorithmic Number Theory, Vol 1: Efficient Algorithms*. Foundations of Computing. MIT Press, 1996. ISBN 978-0262024051.

[2] A. P. Bel'tyukov. Decidability of the universal theory of natural numbers with addition and divisibility. *J. Sov. Math.*, pages 1436–1444, 1980. doi: 10.1007/BF01693974.

[3] Leonard Berman. The complexity of logical theories. *Theor. Comput. Sci.*, 11(1):71–77, 1980.

[4] Itshak Borosh and Leon Bruce Treybig. Bounds on positive integral solutions of linear diophantine equations. *Proc. Am. Math. Soc.*, 55(2):299–304, 1976. doi: 10.2307/2041711.

[5] Viggo Brun. *Über das Goldbachsche Gesetz und die Anzahl der Primzahlpaare*, volume 34(8) of *Arch. Math. Naturvidenskab.* 1915.

[6] Alina Carmen Cojocaru and M. Ram Murty. *An Introduction to Sieve Methods and Their Applications*. Cambridge University Press, 2005. doi: 10.1017/CBO9780511615993.

[7] Florent Guépin, Christoph Haase, and James Worrell. On the existential theories of Büchi arithmetic and linear p-adic fields. In *Proc. Symposium on Logic in Computer Science, LICS*, pages 1–10, 2019. doi: 10.1109/LICS.2019.8785681.

[8] Hendrik W. Lenstra Jr. Integer programming with a fixed number of variables. *Math. Oper. Res.*, 8(4):538–548, 1983. doi: 10.1287/moor.8.4.538.

[9] Richard M. Karp. Reducibility among combinatorial problems. In *Complexity of Computer Computations*, The IBM Research Symposia Series, pages 85–103, 1972.

[10] Jochen Koenigsmann. *Undecidability in Number Theory*, pages 159–195. Springer Berlin Heidelberg, 2014. doi: 10.1007/978-3-642-54936-6.

[11] Antonia Lechner, Joël Ouaknine, and James Worrell. On the complexity of linear arithmetic with divisibility. In *Proc. Symposium on Logic in Computer Science, LICS*, pages 667–676, 2015. doi: 10.1109/LICS.2015.67.

[12] Anthony W. Lin and Rupak Majumdar. Quadratic word equations with length constraints, counter systems, and Presburger arithmetic with divisibility. *Log. Methods Comput. Sci.*, 17 (4), 2021. doi: 10.46298/lmcs-17(4:4)2021.

[13] Leonard Lipshitz. The Diophantine problem for addition and divisibility. *Trans. Am. Math. Soc*, pages 271–283, 1978. doi: 10.2307/1998219.

[14] Leonard Lipshitz. Some remarks on the Diophantine problem for addition and divisibility. *Bull. Soc. Math. Belg. Sér. B*, 33(1):41–52, 1981.

[15] Yuri Matijasevič. Enumerable sets are diophantine. *J. Sov. Math.*, 11:354–357, 1970. doi: 10.2307/2272763.

[16] Julia Robinson. Definability and decision problems in arithmetic. *J. Symb. Log.*, 14(2):98–114, 1949. doi: 10.2307/2266510.

[17] Mikhail R. Starchak. Positive existential definability with unit, addition and coprimeness. In *Proc. International Symposium on Symbolic and Algebraic Computation, ISSAC*, pages 353–360, 2021. doi: 10.1145/3452143.3465515.

[18] Mikhail R. Starchak. A proof of Bel'tyukov–Lipshitz theorem by quasi-quantifier elimination. I. definitions and GCD-lemma. *Vestnik St. Petersb. Univ. Math.*, 54:264–272, 2021. doi: 10.1134/S1063454121030080.

[19] Mikhail R. Starchak. A proof of Bel'tyukov–Lipshitz theorem by quasi-quantifier elimination. II. the main reduction. *Vestnik St. Petersb. Univ. Math.*, 54:372–380, 2021. doi: 10.1134/S106345412104018X.

[20] Lou van den Dries and Andrew J. Wilkie. The laws of integer divisibility, and solution sets of linear divisibility conditions. *J. Symb. Log.*, 68(2):503–526, 2003. doi: 10.2178/jsl/1052669061.

[21] Joachim von zur Gathen and Malte Sieveking. A bound on solutions of linear integer equalities and inequalities. *Proc. Am. Math. Soc.*, 72(1):155–158, 1978. doi: 10.2307/2042554.