The Double Exponential Runtime is Tight for 2-Stage Stochastic ILPs

Klaus Jansen
Department of Computer Science, Kiel University, Kiel, Germany
kj@informatik.uni-kiel.de

Kim-Manuel Klein
Department of Computer Science, Kiel University, Kiel, Germany
kmk@informatik.uni-kiel.de

Alexandra Lassota
Department of Computer Science, Kiel University, Kiel, Germany
ala@informatik.uni-kiel.de

Abstract

We consider fundamental algorithmic number theoretic problems and their relation to a class of block structured Integer Linear Programs (ILPs) called 2-stage stochastic. A 2-stage stochastic ILP is an integer program of the form

\[ \min \{ w^T x \mid Ax = b, L \leq x \leq U, x \in \mathbb{Z}^{r+s+t} \} \]

where the constraint matrix \( A \in \mathbb{Z}^{r \times (r+s+t)} \) consists of \( n \) repetitions of a block matrix \( A \in \mathbb{Z}^{r \times s} \) on the vertical line and \( n \) repetitions of a matrix \( B \in \mathbb{Z}^{r \times t} \) on the diagonal line aside.

In this paper we show an advanced hardness result for a number theoretic problem called Quadratic Congruences where the objective is to compute a number \( z \leq \gamma \) satisfying

\[ z^2 \equiv \alpha \mod \beta \]

for given \( \alpha, \beta, \gamma \in \mathbb{Z} \). As an implication of our hardness result for the Quadratic Congruences problem we prove a lower bound of

\[ 2^{2^{\delta(r+t)}} |I|^{O(1)} \]

for some \( \delta > 0 \) for the running time of any algorithm solving 2-stage stochastic ILPs where \( |I| \) is the encoding length of the instance. This result even holds if \( s, \|b\|_{\infty} \) and the largest absolute value \( \Delta \) in the constraint matrix \( A \) are constant. This shows that the recent algorithm developed by Klein is nearly optimal. The lower bound is based on the Exponential Time Hypothesis (ETH).

2012 ACM Subject Classification Mathematics of computing → Integer programming

Keywords and phrases 2-stage stochastic ILPs, Lower Bound, ETH

Funding This work was supported by DFG project JA 612/20-1

1 Introduction

One of the most fundamental problems in algorithm theory and optimization is the Integer Linear Programming problem. Many theoretical and practical problems can be modeled as Integer Linear Programs and thus they serve as a very general but powerful framework for tackling various questions. Formally, the Integer Linear Programming problem is defined as

\[ \min \{ w^T x \mid Ax = b, \ell \leq x \leq u, x \in \mathbb{Z}^n \} \]

for some matrix \( A \in \mathbb{Z}^{m \times n} \), a right-hand side \( b \in \mathbb{Z}^m \), an objective function \( w \in \mathbb{Z}^n \) and some lower and upper bounds \( \ell, u \in \mathbb{Z}^n \). Here we aim to find a solution \( x \) such that the value of the objective function \( w^T x \) is minimized. In general, this problem is NP-hard. Thus it is of great interest to find structures to these Integer Linear Programs which make them solvable more efficiently. Here we consider the 2-stage stochastic Integer Linear Programming problem where the constraint matrix admits a specific block structure. Namely, the constraint matrix \( A \) only contains non-zero entries in the first few columns and
The Double Exponential Runtime is Tight for 2-Stage Stochastic ILPs

Thereby $A_1, \ldots, A_n \in \mathbb{Z}^{r \times s}$ and $B_1, \ldots, B_n \in \mathbb{Z}^{r \times t}$ are integer matrices themselves. The complete constraint matrix $A$ has size $((nr) \times (s + nt))$. Let $\Delta$ denote the largest absolute entry in $A$. Further denote by $U, L \in \mathbb{Z}^{s \times nt}$ the upper and lower bounds on the variables.

Such 2-stage stochastic ILPs are a very common tool in stochastic programming [14] and they are often used in practice to model uncertainty of decision making over time. Due to the applicability a lot of research has been done in order to solve those (mixed) ILPs efficiently in practice. Since we focus on the theoretical aspects of 2-stage stochastic ILPs in this paper, we only refer the reader to the surveys [18, 21] regarding the practical methods.

The currently state-of-the-art algorithm to solve 2-stage stochastic ILPs admits a running time of $O(n^{2.387} \cdot |I| \cdot \log^3(nt) \cdot (rs\Delta)^{rs(2r\Delta + 1)^{r^2}})$ where $|I|$ is the encoding length of the Integer Linear Program [15]. Before, very little was known about the parameter dependency in the running time of 2-stage stochastic ILPs. The first result in that respect was by Hemmecke and Schulz [11] who provided an algorithm with a running time of $f(r, s, t, \Delta) \cdot \text{poly}(n)$ for some computable function $f$. However, due to the use of an existential result from commutative algebra, no explicit bound could be stated for the function $f$. Further it was known that if the constraint matrix is the transposed of $A$ (the problem is then called $n$-fold ILP), the problem is solvable in time $f(\Delta, r, s) \cdot \text{poly}(n, t, |I|)$ where $f$ is only single exponential. This naturally rises the questions whether 2-stage stochastic ILPs are intrinsically harder to solve. We answer this question affirmatively by showing a double exponential lower bound in the running time for any algorithm solving the 2-Stage Stochastic Integer Linear Programming problem.

To prove this hardness we reduce from the Quadratic Congruences problem where the objective is to compute a number $z \leq \gamma$ such that $z^2 \equiv \alpha \mod \beta$ or to state correctly that such a number does not exists. In a classical result, this problem was proven to be NP-hard by Manders and Adleman [20] already in 1978, showing a reduction from 3-SAT. This hardness even persists when the prime factorization of $\beta$ is given [20]. By this result, Manders and Adleman proved that it is NP-complete to compute the solutions of diophantine equations of degree 2.

In order to achieve the desired lower bounds on the running time from the reductions we make use of the Exponential Time Hypothesis (ETH) – a widely believed conjecture stating that the 3-SAT problem with $\ell$ variables cannot be solved in subexponential time. Using the ETH, plenty lower bounds for various problems are shown, for an overview on the techniques and results see e.g. [5]. Furthermore, the Sparsification lemma states that we can reformulate a given formula into subexponential many new formulas each with a linear number of clauses regarding $\ell$. Formally this yields:

\[
\textbf{Conjecture 1 (ETH + Sparsification lemma).} \quad \text{The 3-SAT problem cannot be solved in time less than } O(2^{\delta_3(\ell + m)}) \text{ for some constant } \delta_3 > 0 \text{ where } \ell \text{ is the number of variables and } m \text{ is the number of clauses in the instance.}
\]

The current state-of-the-art is an algorithm with running time $O(2^{0.387(\ell + m)})$, i.e., $\delta_3 \leq 0.387$ [5]. Further, we also need the Chinese Remainder theorem in some parts of the proofs, which states the following:
Proposition 1 \((\[12\])\). Let \(n_1, \ldots, n_k\) be pairwise co-prime. Further let \(a_1, \ldots, a_k\) be some integers. Then there exists integers \(x\) satisfying \(x \equiv a_i \mod n_i\) for all \(i\). Further any two solutions \(x_1, x_2\) are congruent modulo \(\prod_{i=1}^{k} n_i\).

Related Work
A closely related class of block structured ILPs are so called \(n\)-fold ILPs. Here, the constraint matrix \(A\) is of the form

\[
\begin{pmatrix}
A_1 & A_2 & \cdots & A_n \\
B_1 & 0 & \cdots & 0 \\
0 & B_2 & \ddots & \vdots \\
\vdots & \ddots & \ddots & 0 \\
0 & \cdots & 0 & B_n
\end{pmatrix}
\]

for some block matrices \(A_i \in \mathbb{Z}^{r \times s}\) and \(B_i \in \mathbb{Z}^{r \times t}\) and hence this matrix is the transposed of a constraint matrix of a 2-stage stochastic ILP.

In recent years there was significant progress in the development of algorithms for \(n\)-fold ILPs and lower bounds on the other hand. The best known algorithm to solve the ILP has a running time of \(\Delta \cdot O^{r^2s} \cdot \text{poly}(|I|) \cdot \text{poly}(n, t)\) [6, 13, 17] while the best known lower bound is \(\Delta n^{(r^2s)^2}\) for some \(\delta_n > 0\) [7].

Despite their similarity it seems that 2-stage stochastic ILPs are significantly harder to solve than \(n\)-fold ILPs. Yet, no super exponential lower bound for the running time of an algorithm solving the 2-STAGE STOCHASTIC INTEGER LINEAR PROGRAMMING problem was shown. There is a lower bound for a more general class of ILPs in [7] that contain 2-stage stochastic ILPs showing that the running time is double-exponential parameterized by the topological height of the treedepth decomposition of the primal or dual graph. However, the topological height of 2-stage stochastic ILPs is constant and thus no strong lower bound can be derived for this case.

If we relax the necessity of an integral solution, the 2-stage stochastic LP problem becomes solvable in time \(2\Delta \cdot n^{O(r^3)} \cdot n \log^3(n) \log(||U - L||_\infty) \log(||w||_\infty)\) [2]. For the case of mixed Integer Linear Programs there exists an algorithm solving 2-stage stochastic MILPs in time \(2\Delta \cdot n^{O(r^3)} \cdot n \log^3(n) \log(||U - L||_\infty) \log(||w||_\infty)\) [2]. Both results rely on the fractionality of a solution, which is of size dependent only on the parameters. This allows us to scale the problem up such that it becomes an ILP (as the solution has to be integral) and thus state-of-the-art algorithms for 2-stage stochastic ILPs can be applied.

There are also studies for a more general case called 4-Block ILPs where the constraint matrix consists of non-zero entries in the first few columns, the first few rows and block-wise along the diagonal. This may be seen as the combination if \(n\)-fold and 2-stage stochastic ILPs. Only little is known about them: 4-Block ILPs are in XP [10]. Further, a lower and upper bound on the Graver Basis elements (inclusion-wise minimal kernel elements) of \(O_{\text{FPT}}(n^{sD})\) were shown recently [3]. Here \(O_{\text{FPT}}\) hides a multiplicative factor that is only dependent on the parameters and \(s_D\) is the number of rows in the submatrix \(D\) appearing repeatedly in the first few rows.

Our Results
One of our main results is to show strong NP-hardness of the following algorithmic number theoretic problem which we call the MODULO REST DECISION problem. Here, we are given
numbers $x_1, \ldots, x_{n_{\text{MRD}}}, y_1, \ldots, y_{n_{\text{MRD}}, \zeta \in \mathbb{N}$ and pairwise coprime numbers $q_1, \ldots, q_{n_{\text{MRD}}}$. The question is to decide whether there exists a number $z \in \mathbb{Z}_{>0}$ with $z \leq \zeta$ satisfying the following congruences:

\[
z \equiv \{x_1, y_1\} \mod q_1 \\
z \equiv \{x_2, y_2\} \mod q_2 \\
\vdots \\
z \equiv \{x_{n_{\text{MRD}}}, y_{n_{\text{MRD}}}\} \mod q_{n_{\text{MRD}}}.
\]

Here $\{x_i, y_i\}$ means that either the residue $x_i$ or $y_i$ should be met. This problem is a natural generalization of the Chinese Remainder theorem where $x_i = y_i$ for all $i$. In this case, the problem can be solved using the Extended Euclidean algorithm. To the best of our knowledge the Modulo Rest Decision problem has not been considered in the literature so far. In this paper, we show hardness of the problem by a reduction from the Quadratic Congruences problem. As the known NP-hardness for the Quadratic Congruences problem by Manders and Adleman \cite{20} only holds for specific numbers $\beta$ with a prime factorization that contains only prime numbers with a high multiplicity, their reduction does not yield the desired properties. Thus we give a new reduction dedicated to the original one where we show a stronger NP-hardness result: The Quadratic Congruences problem remains NP-hard, even if the prime factorization of $\beta$ is given and each prime number greater than 2 occurs at most once and the prime number 2 occurs four times.

Finally, we show that the Modulo Rest Decision problem can be formulated by a 2-stage stochastic ILP. Assuming the ETH, we can then conclude that a doubly exponential lower bound of $2^{2^{r+t+1}|I|^{O(1)}}$ on the running time of any algorithm solving 2-stage stochastic ILPs holds. The double exponential lower bound even holds if $s = 1$ and $\Delta, ||b||_{\infty} \in O(1)$. This proves the suspicion that 2-stage stochastic ILPs are significantly harder to solve than $n$-folds ILPs with respect to the dimensions of the block matrices and $\Delta$. Furthermore it implies that the current state-of-the-art algorithm for solving 2-stage stochastic ILPs is indeed (nearly) optimal.

## 2 Advanced Hardness for Quadratic Congruences

To prove our main result we have to go through two involved reductions. This section is dedicated to the first one: We show that every instance of the 3-SAT problem can be transformed to an instance of the Quadratic Congruences problem in polynomial time. Recall that in the Quadratic Congruences problem the objective is to compute a number $z \leq \gamma$ such that $z^2 \equiv \alpha \mod \beta$ or to state correctly that such a number does not exists. This problem was proven to be NP-hard by Manders and Adleman \cite{20} showing a reduction from 3-SAT. This hardness even persists when the prime factorization of $\beta$ is given \cite{20}. However, we aim for an even stronger statement: The Quadratic Congruences problem remains NP-hard, even if the prime factorization of $\beta$ is given and each prime number greater than 2 occurs at most once and the prime number 2 occurs four times. This does not infer from the original hardness proof. In contrast, if $\ell$ is the number of variables and $m$ the number of clauses in the 3-SAT formula then $\beta$ admits a prime factorization with $O(m + \ell)$ different prime numbers each with a multiplicity of at least $O(m + \ell)$. Even though our new reduction lessens the occurrence of each prime factor greatly, we do not introduce noteworthy more different prime factors as well as their values are of similar dimension. In the following,
we follow the structure of the original proof from [20]. However, the proof requires major adaptations and insights on prime factors in each step.

\textbf{Theorem 2.} The Quadratic Congruences problem is NP-hard, even if the prime factorization of $\beta$ is given and each prime factor greater 2 occurs at most once and the prime factor 2 occurs 4 times.

\textbf{Proof.} We start from the well-known NP-hard problem 3-SAT where we are given a 3-SAT formula $\phi$ with $\ell$ variables and $m$ clauses.

\textit{Transformation:} First, eliminate duplicate clauses from $\phi$ and those where some variable $x_i$ and its negation $\bar{x}_i$ appear together. Call the resulting formula $\phi'$, the number of occurring variables $\ell'$ and denote by $m'$ the number of appearing clauses respectively. Let $\Sigma = (\sigma_1, \ldots, \sigma_{m'})$ be some enumeration of the clauses. Denote by $p_0, \ldots, p_{2m'}$ the first $2m' + 1$ prime numbers. Compute

$$\tau_{\phi'} = -\sum_{i=1}^{m'} \prod_{j=1}^i p_j.$$ 

Further compute for each $i \in 1, 2, \ldots, \ell'$:

$$f^+_i = \sum_{x_i \in \sigma_j} \prod_{k=1}^j p_k \quad \text{and} \quad f^-_i = \sum_{\bar{x}_i \in \sigma_j} \prod_{k=1}^j p_k.$$ 

Set $n = 2m' + \ell'$. Using this we compute the coefficients $c_j$ for all $j = 0, 1, \ldots, n$. Set $c_0 = 0$. For $j = 1, \ldots, 2m'$ compute

$$c_j = -1/2 \prod_{i=1}^j p_i \quad \text{for} \quad j = 2k - 1 \quad \text{and} \quad c_j = -\prod_{i=1}^j p_i \quad \text{for} \quad j = 2k.$$ 

Compute the remaining ones for $j = 1, \ldots, \ell'$ as $c_{2m'+j} = 1/2 \cdot (f^+_j - f^-_j)$. Further compute the number $\tau = \tau_{\phi'} + \sum_{j=0}^n c_j + \sum_{i=1}^{\ell'} f^-_i$.

Denote by $p_{0,0}, p_{0,1}, \ldots, p_{0,n}, p_{1,0}, \ldots, p_{n,n}$ the first $(n+1)^2 = n^2 + 2n + 1$ prime numbers greater than $(4(n+1)2^{n+2}1 \prod_{i=1}^{n+2} q_i^{1/(n^2+2n+1)})$ and greater than $p_{2m'}$, where $q_1, \ldots, q_{n^2+2n+1}$ are the first $n^2 + 2n + 1$ prime numbers. Define $p^*$ as the $(n^2 + 2n + 2m' + 13)$th prime number.

Determine the parameters $\theta_j$ for $j = 0, 1, \ldots, n$ as the least $\theta_j$ satisfying:

$$\theta_j \equiv c_j \mod 2^3 \cdot p^* \prod_{i=1}^{m'} p_i \quad \text{and} \quad \theta_j \equiv 0 \mod \prod_{i=0, i \neq j}^n p_i \quad \text{and} \quad \theta_j \not\equiv 0 \mod p_{j,1}.$$ 

Set the following parameters:

$$H = \sum_{j=0}^n \theta_j \quad \text{and} \quad K = \prod_{i=0}^n \prod_{k=0}^n p_{i,j}.$$
Finally set
\[ \alpha = (2^4 \cdot p^* \prod_{i=1}^{m'} p_i + K)^{-1} \cdot (K^2 + 2^4 \cdot p^* \prod_{i=1}^{m'} p_i \cdot H^2), \]
\[ \beta = 2^4 \cdot p^* \prod_{i=1}^{m'} p_i \cdot K, \]
\[ \gamma = H. \]

where \((2^4 \cdot p^* \prod_{i=1}^{m'} p_i + K)^{-1}\) is the inverse of \((2^4 \cdot p^* \prod_{i=1}^{m'} p_i + K) \text{ mod } 2^4 \cdot p^* \prod_{i=1}^{m'} p_i \cdot K.\]

**Correctness:** The main idea of the transformation is to interpret the 3-SAT formula as a system of equations. Then we use a line of transformations into equivalent systems until we finally reach the QUADRATIC CONGRUENCES problem. To integrate the modulo we introduce a number greater than every possible outcome of some equation thus not influencing the system. For most of the equivalence transformation we prove and use insights on the structure of possible solutions allowing us to reformulate the equations accordingly. Doing so, the prime factors to calculate \(\beta\) are indeed generated uniquely, except 2 which we will need 4 times to prove the equivalence between two systems of equations. Further the prime factors are sufficient small and their number is not noteworthy larger than in the original proof.

Before we start with the transformations of the formula into systems of equations, we will first observe some properties about the generated prime factors. These will come in handy for the estimations later on. In particular we want to show that choosing \(p^*\) as the \((n^2 + 2n + 2m' + 13)\text{th} \) prime factor satisfies \(p^* > p_{n,n}: \) Suppose \(p_{2m'} \leq (4(n+1)2^3 \cdot \prod_{i=1}^{n^2+2n+1} q_i)^{1/(n^2+2n+1)}\). Then \(p_{n,n}\) is the \((n^2 + 2n + 1 + 2m' + 1)\text{th} \) prime number and thus \(p^* > p_{n,n}.\) Otherwise, if \(p_{2m'} < (4(n+1)2^3 \prod_{i=1}^{n^2+2n+1} q_i)^{1/(n^2+2n+1)},\) we bound the function values. It holds that
\[ (4(n+1)2^3 \prod_{i=1}^{n^2+2n+1} q_i)^{1/(n^2+2n+1)} = 4^{1/(n^2+2n+1)}(n + 1)^{1/(n^2+2n+1)}(2^3)^{1/(n^2+2n+1)}( \prod_{i=1}^{n^2+2n+1} q_i)^{1/(n^2+2n+1)} \leq 2 \cdot 2 \cdot 2 \cdot (4^{n^2+2n+1})^{1/(n^2+2n+1)} = 2 \cdot 2 \cdot 4 = 32.\]

The second transformation holds as the product of the first \(k\) prime numbers is bounded by \(4^k.\) There are 11 prime numbers in the interval \([1,32].\) Thus \(p_{n,n}\) is at most the \((11 + n^2 + 2n + 1)\text{th} \) prime number and thus \(p^* > p_{n,n}.\)

Further note that \((4(n+1)2^3 \cdot p^* \prod_{i=1}^{m'} p_i)^{1/(n^2+2n+1)} \leq (4(n+1)2^3 \prod_{i=1}^{n^2+2n+1} q_i)^{1/(n^2+2n+1)}\) holds, i.e., \(p^* \leq \prod_{i=m'+1}^{n^2+2n+1} q_i: \) We can bound the value of the product from beneath as \(\prod_{i=m'+1}^{n^2+2n+1} q_i \geq q_{m'+1}.\) Further it is known that the value of the next prime number after a number \(\rho\) is at most \(2\rho.\) Thus, as there are \(n^2 + 2n + m' + 11\) prime numbers between \(p_{m'+1}\) and \(p^*\), we have that \(p^*\) has a value of at most \(q_{m'+1} \cdot 2^{n^2 + 2n + m' + 11}.\) Setting \(q_{m'+1} = 5\) to the smallest reasonable value for \(m' = 2\) (if we only have one clause in the 3-SAT formula the problem becomes easy) we see that \(5^{n^2 + n} \geq 5 \cdot 2^{n^2 + 2n + m' + 11}\) as \(5^{n^2 + n} \geq 2^{2\ell + 2n} = 2^{n^2 + 2n}\) is greater than \(5 \cdot 2^{n^2 + 2n + m' + 11} \leq 2^3 \cdot 2^{n^2 + 2n + m' + 11} = 2^{n^2 + 2n + m' + 14} \leq 2^{n^2 + 3n + 11}\) for all reasonable values of \(n,\) i.e., \(n \geq 7\) (for the case of \(m' = 2\) clauses and \(\ell' = 3\) variables).
Turning back to the given formula: Obviously the reduced formula $\phi'$ is only satisfiable if and only if $\phi$ is. The formula $\phi'$ is satisfiable if there exists a truth assignment $r: \{x_1, \ldots, x_\nu\} \rightarrow \{0,1\}$ assigning a logical value to each variable $x_1, \ldots, x_\nu$ which satisfies all clauses $\sigma_1, \ldots, \sigma_m$ simultaneously. This can be re-written to the following equation for each clause $\sigma_k \in \phi_k$:

$$0 = R_k = y_k - \sum_{x_i \in \sigma_k} r(x_i) - \sum_{x_i \in \sigma_k} (1 - r(x_i)) + 1, \; y_k \in \{0, 1, 2, 3\}$$

For a clause $\sigma_k$, this equation is only satisfiable if at least one variable $x_i \in \sigma_k$ has value $r(x_i) = 1$ or one variable occurring in its negation $\bar{x}_i \in \sigma_k$ has value $r(x_i) = 0$. Otherwise we have to set $y_k = -1$ which is not allowed. Note that we never have to set $y_k = 3$ to satisfy the formula. However, we allow this value as it will come in handy later on when transforming the equation. Further, set $0 = R_0 = \alpha_0 + 1$ for $\alpha_0 \in \{-1, +1\}$ for later convenience. Obviously the new equation is satisfiable.

We can bound the values of $R_k$ for $k \in \{0, 1, \ldots, m'\}$ by $-2 \leq R_k \leq 4$. For the lower bound the values are given by $y_k = 0$, all $x_i \in \sigma_k$ have value $r(x_i) = 1$ and all $\bar{x}_i \in \sigma_k$ have value $r(x_i) = 0$. For the upper bound we set $y_k = 3$, all $x_i \in \sigma_k$ to $r(x_i) = 0$ and $\bar{x}_i \in \sigma_k$ to $r(x_i) = 1$. For $R_0$ obviously $0 \leq R_k \leq 2$ holds. Thus

$$R_k = 0, \forall k \in \{0, 1, \ldots, m'\} \Rightarrow \sum_{k=0}^{m'} R_k \prod_{i=0}^{k} p_i = 0$$

as the sum is zero if all $R_k = 0$. For the opposite direction, if the sum is zero, then no $R_k \neq 0$ as the other summands can not compensate for it due to the product of the prime numbers growing too fast and as $\prod_{i=0}^{k} p_i \neq 0$. Further we can bound the expression by

$$\sum_{k=0}^{m'} R_k \prod_{i=0}^{k} p_i \leq 4 \sum_{k=0}^{m'} \prod_{i=0}^{k} p_i \leq 4(m' + 1) \prod_{i=0}^{m'} p_i < 2^3 \cdot p^* \prod_{i=1}^{m'} p_i$$

as $p^* > p_{n,n} > p_{m'} > m' + 1$. This yields

$$R_k = 0, \forall k \in \{0, 1, \ldots, m'\} \Rightarrow \sum_{k=0}^{m'} R_k \prod_{i=0}^{k} p_i \equiv 0 \mod 2^3 \cdot p^* \prod_{i=1}^{m'} p_i \quad (I)$$

as the modulo has no impact on the satisfiability of the equation.

Next we aim to re-write $R_k$ by replacing the variables $y_k$ and $r(x_i)$ with new variables with domain $\{-1, 1\}$ as follows:

$$y_k = 1/2 \cdot [(1 - \alpha_{2k-1}) + 2 \cdot (1 - \alpha_{2k})], \; k \in \{1, \ldots, m'\}$$

$$r(x_i) = 1/2 \cdot (1 - \alpha_{2m'+i}), \; i \in \{1, \ldots, \ell'\}.$$

Obviously the value domains of $y_k$ and $r(x_i)$ are preserved. Substituting the variables and re-arranging the equation $(I)$ yields

$$\sum_{j=0}^{n} c_j \alpha_j \equiv \tau \mod 2^3 \cdot p^* \prod_{i=1}^{m'} p_i, \; \alpha_j \in \{-1, +1\}.$$ 

By definition of $\theta_j$ this is equivalent to

$$\sum_{j=0}^{n} \theta_j \alpha_j \equiv \tau \mod 2^3 \cdot p^* \prod_{i=1}^{m'} p_i, \; \alpha_j \in \{-1, +1\}.$$
Let us consider the following system with $H = \sum_{j=0}^{n} \theta_j$ and $K = \prod_{i=0}^{n} \prod_{k=0}^{n} p_{i,j}$ defined as before:

$$0 \leq |x| \leq H, \ x \in \mathbb{Z}$$

$$(H + x)(H - x) \equiv 0 \mod K$$

The unique solutions $x$ to the given system are of form

$$x = \sum_{j=0}^{n} \theta_j \alpha_j, \ \alpha \in \{-1, +1\}, \ j = 0, 1, \ldots, n$$

Let us first verify that an $x$ of such form solves the system. First

$$|x| = |\sum_{j=0}^{n} \theta_j \alpha_j| \leq \sum_{j=0}^{n} \theta_j = H$$

satisfies (1,1). Further we have that each summand in the resolved formula $(H + x)(H - x)$ has to contain all prime factors $p_{i,j}$ for $i = 0, 1, \ldots, n$ and $j = 0, 1, \ldots, n$ in its prime factorization to satisfy (1,2). For $(H + x) = (\sum_{j=0}^{n} \theta_j + \sum_{j=0}^{n} \theta_j \alpha_j)$ it holds that each $\theta_j$ where $\alpha_j = +1$ will occur twice while each $\theta_j$ where $\alpha_j = -1$ will be canceled out by $H$. The other way round holds for $(H - x)$. Thus expanding the brackets will yield that each summand is a product of some $\theta_j$ and $\theta_k$ where $\alpha_j = +1$ and $\alpha_k = -1$. This implies that $j \neq k$. As each $\theta_j$ contains all prime factors of $K$ except $p_{j,0}, \ldots, p_{j,n}$, the product of two different $\theta_j$ and $\theta_k$ will contain each prime factor occurring in $K$ satisfying (1,2).

Regarding the uniqueness, first observe that

$$(H + x)(H - x) \equiv 0 \mod \prod_{j=0}^{n} p_{i,j}, \ \forall i = 0, 1, \ldots, n.$$ 

Assume there exists some number $\tilde{p} = \prod_{j=0}^{n} p_{i,j}$ for some $i \in \{0, 1, \ldots, n\}$ which divides $(H + x)$ and $(H - x)$ without rest. Thus $(H + x) + (H - x) \equiv 0 \mod \tilde{p} \iff 2H \equiv 0 \mod \tilde{p}$. As $\tilde{p}$ is a product of prime numbers greater 2 is follows that $H \equiv 0 \mod \tilde{p} \iff \sum_{j=0}^{n} \theta_j \equiv 0 \mod \tilde{p}$. However, from the definition of $\theta_j$ (third condition) it follows that for each $j$ there exist different prime numbers not present in the prime factorization of $\theta_j$ contradicting the assumption. Thus $\tilde{p}$ divides either $(H + x)$ or $(H - x)$ without rest. Define

$$\alpha_i = \begin{cases} +1 & \text{if } (H - x) \equiv 0 \mod \prod_{j=0}^{n} p_{i,j} \\ -1 & \text{if } (H + x) \equiv 0 \mod \prod_{j=0}^{n} p_{i,j} \end{cases}$$

$$x' = \sum_{i=0}^{n} \alpha_i \theta_i.$$
It holds that

\[ x' \equiv x \mod \prod_{j=0}^{n} p_{i,j} \]

\[ \Leftrightarrow \sum_{i=0}^{n} \alpha_{i} \theta_{i} \equiv x \mod \prod_{j=0}^{n} p_{i,j} \]

\[ \Leftrightarrow \alpha_{i} \theta_{i} \equiv x \mod \prod_{j=0}^{n} p_{i,j} \]

\[ \Leftrightarrow \sum_{k=0}^{n} \alpha_{k} \theta_{k} \equiv x \mod \prod_{j=0}^{n} p_{i,j} \]

\[ \Leftrightarrow \alpha_{i} \sum_{k=0}^{n} \theta_{k} \equiv x \mod \prod_{j=0}^{n} p_{i,j} \]

\[ \Leftrightarrow \alpha_{i} H \equiv x \mod \prod_{j=0}^{n} p_{i,j} \]

for all \( i \in \{0, 1, \ldots, n\} \). Further as \( \alpha_{i} \in \{-1, +1\} \) for all \( j \in \{0, 1, \ldots, n\} \) it holds that \(-H \leq x \leq H\). Since the same holds for \( x \) it follows that \( |x - x'| \leq 2H \). Let us bound the value of \( \lambda_j = \theta_j / (\prod_{i=0, i \neq j}^{n} p_{i,k}) \). It holds that \( \theta_{j} \leq 2^{4} \cdot p^{*} \prod_{i=1}^{m'} p_{i} \prod_{i=0, i \neq j}^{n} p_{i,k} \) [22]. Further, for each \( i \in \{0, 1, \ldots, n\} \) and \( j \in \{0, 1, \ldots, n\} \) it holds that \( p_{i,j} > (4(n + 1)2^{3} \cdot p^{*} \prod_{i=1}^{m'} p_{i})^{1/(n^2 + 2n + 1)} \) as we estimated before. Thus

\[ \lambda_j = \theta_j / (\prod_{i=0, i \neq j}^{n} p_{i,k}) \]

\[ < (2^{4} \cdot p^{*} \prod_{i=1}^{m'} p_{i} \cdot \prod_{i=0, i \neq j}^{n} p_{i,k}) / (\prod_{i=0, i \neq j}^{n} p_{i,k}) \]

\[ = (2^{4} \cdot p^{*} \prod_{i=1}^{m'} p_{i} \cdot k / (\prod_{k=0}^{n} p_{j,k})) / (\prod_{i=0, i \neq j}^{n} p_{i,k}) \]

\[ = (2^{4} \cdot p^{*} \prod_{i=1}^{m'} p_{i} \cdot K / (\prod_{k=0}^{n} p_{j,k})) / (\prod_{i=0, i \neq j}^{n} p_{i,k}) \]

\[ < (2^{4} \cdot p^{*} \prod_{i=1}^{m'} p_{i} \cdot K / (\prod_{i=0, i \neq j}^{n} p_{i,k})) \]

\[ < (2^{4} \cdot p^{*} \prod_{i=1}^{m'} p_{i} \cdot K / (\prod_{i=0, i \neq j}^{n} p_{i,k})) \]

\[ < (2^{4} \cdot p^{*} \prod_{i=1}^{m'} p_{i} \cdot K / (p_{0,0}^{n^2 + 2n + 1})) \]

\[ \leq (2^{4} \cdot p^{*} \prod_{i=1}^{m'} p_{i} \cdot K / ((4(n + 1)2^{3} \cdot p^{*} \prod_{i=1}^{m'} p_{i})^{1/(n^2 + 2n + 1)} n^2 + 2n + 1)) \]

\[ = (2^{4} \cdot p^{*} \prod_{i=1}^{m'} p_{i} \cdot K / (4(n + 1)2^{3} \cdot p^{*} \prod_{i=1}^{m'} p_{i})) \]

\[ = K / (2(n + 1)) \]
As each term of $H$ is bounded by $K/(2(n+1))$ it follows that $2H = 2\sum_{j=0}^{n} \theta_j < 2 \cdot (n+1) \cdot K/(2(n+1)) = K$. Thus $x = x'$ and we conclude that solution of the form $x = \sum_{j=0}^{n} \theta_j \alpha_j$ are the unique solutions to the system (1.1) and (1.2).

Thus we can re-write

$$\sum_{j=0}^{n} \theta_j \alpha_j \equiv \tau \mod 2^3 \cdot p^* \prod_{i=1}^{m'} p_i, \; \alpha_j \in \{-1, +1\}.$$ 

using the system (1.1) and (1.2) to the following one:

1. \[0 \leq |x| \leq H, \; x \in \mathbb{Z} \tag{2.1}\]
2. \[x \equiv \tau \mod 2^3 \cdot p^* \prod_{i=1}^{m'} p_i \tag{2.2}\]
3. \[(H + x)(H - x) \equiv 0 \mod K. \tag{2.3}\]

Next, we re-write the system to:

1. \[0 \leq |x| \leq H, \; x \in \mathbb{Z} \tag{3.1}\]
2. \[(\tau - x)(\tau + x) \equiv 0 \mod 2^4 \cdot p^* \prod_{i=1}^{m'} p_i \tag{3.2}\]
3. \[(H + x)(H - x) \equiv 0 \mod K. \tag{3.3}\]

As only the second conditions differ we focus on their equivalence in the following.

Firstly we prove that if (2.2) holds, i.e., $x \equiv \tau \mod 2^3 \cdot p^* \prod_{i=1}^{m'} p_i$, then (3.2), i.e., $(\tau - x)(\tau + x) \equiv 0 \mod 2^4 \cdot p^* \prod_{i=1}^{m'} p_i$ holds. We can re-write (2.2) to $x = \lambda 2^3 \cdot p^* \prod_{i=1}^{m'} p_i + \tau$ for some $\lambda \in \mathbb{Z}$. Inserting this in (3.2) yields:

\[
(\tau + \lambda 2^3 \cdot p^* \prod_{i=1}^{m'} p_i + \tau)(\tau - \lambda 2^3 \cdot p^* \prod_{i=1}^{m'} p_i - \tau) = (2\tau + \lambda 2^3 \cdot p^* \prod_{i=1}^{m'} p_i)(\lambda 2^3 \cdot p^* \prod_{i=1}^{m'} p_i) \equiv 0 \mod 2^3 \cdot p^* \prod_{i=1}^{m'} p_i
\]
as each factor is multiplied with $\lambda 2^3 \cdot p^* \prod_{i=1}^{m'} p_i$.

Next we prove the opposite direction. First, observe that if $(\tau - x)(\tau + x) \equiv 0 \mod 2^3 \cdot p^* \prod_{i=1}^{m'} p_i$ then either $(\tau - x) \equiv 0 \mod 2^3$ or $(\tau + x) \equiv 0 \mod 2^3$: As (3.2) holds, $(\tau + x) = \lambda_i \cdot 2^i$ and $(\tau - x) = \lambda_j \cdot 2^j$ for some $i, j \in \mathbb{Z}$ and $\lambda_i, \lambda_j \not\equiv 0 \mod 2$. It follows that

\[
(\tau + x) + (\tau - x) = \lambda_i \cdot 2^i + \lambda_j \cdot 2^j
\]

\[
\Rightarrow 2\tau = \lambda_i \cdot 2^i + \lambda_j \cdot 2^j
\]

\[
\Rightarrow \tau = \lambda_i \cdot 2^{i-1} + \lambda_j \cdot 2^{j-1}.
\]

As $\tau$ is odd per definition, either $i$ or $j$ has to be 1 and thus the other parameter has to be 3. Using this we know that if $x$ satisfies (3.2), then $(\tau - x) \equiv 0 \mod 2^3$ or $(\tau + x) \equiv 0 \mod 2^3$. In the first case, $x$ directly corresponds to a solution of (2.2) as $x - \tau$ is a multiple of $2^3$ and thus $x$ is a multiple of $2^3$ with a residue of $\tau$. Otherwise $-x$ satisfies the condition using the same argument. Obviously the other conditions are also satisfied in both systems.
Lastly, we re-write the system one final time to:

\[
0 \leq x \leq H, \ x \in \mathbb{Z}
\]

\[
2^4 \cdot p^* \cdot \prod_{i=1}^{m'} p_i (H^2 - x^2) + K(\tau^2 - x^2) \equiv 0 \mod 2^4 \cdot p^* \cdot \prod_{i=1}^{m'} p_i \cdot K.
\]  

(4.1)

First, as we only consider \( x^2 \), we can suppose \( x \geq 0 \) and thus re-writing (3.1) to (4.1)

is correct. Further (3.2) and (3.3) merge into (4.2). Recall that \( 2^4 \cdot p^* \cdot \prod_{i=1}^{m'} p_i = K \) are co-prime. The first summand obviously always contains the factor \( 2^4 \cdot p^* \cdot \prod_{i=1}^{m'} p_i \), thus we have to find an \( x \) such that \( (H^2 - x^2) \equiv 0 \mod K \) which corresponds to (3.3). The second summand clearly is a multiple of \( K \), thus we have to assure that \( (\tau^2 - x^2) \equiv 0 \mod 2^4 \cdot p^* \cdot \prod_{i=1}^{m'} p_i \). This matches (3.2).

Dissolving the brackets and rearranging the term (4.2) we get

\[
(2^4 \cdot p^* \cdot \prod_{i=1}^{m'} p_i + K)x^2 \equiv K\tau^2 + 2^4 \cdot p^* \cdot \prod_{i=1}^{m'} p_i H^2 \mod 2^4 \cdot p^* \cdot \prod_{i=1}^{m'} p_i \cdot K.
\]

As \( 2^4 \cdot p^* \cdot \prod_{i=1}^{m'} p_i + K \) is relatively prime to \( 2^4 \cdot p^* \cdot \prod_{i=1}^{m'} p_i \cdot K \) it has an inverse modulo \( 2^4 \cdot p^* \cdot \prod_{i=1}^{m'} p_i \cdot K \) [19]. Thus multiplying by the inverse we get the values for \( \alpha, \beta \) and \( \gamma \) as in the transformation above. This proves that satisfying the formula \( \phi \) is equivalent to an instance of the QUADRATIC CONGRUENCES problem admitting a feasible solution.

**Running time:** All steps, numbers and their computation can be bounded in a polynomial dependent of \( \ell \) and \( m \). First, we eliminate unnecessary clauses from the formula. Thus we have to go through all clauses once. The first \( 2m' + 1 \) prime numbers have a value of at most \( O(m' \log(m')) \) and can thus be found in polynomial time via sieving. The function \( (4(n + 1)2^3 \prod_{i=1}^{n^2+2n+1} q_i)^{1/(n^2+2n+1)} \) is at most 32 as shown before. Thus we can also bound the value of the next \( n^2 + 2n + 1 \) prime numbers larger than 32 and \( p_{2m'} \) by a polynomial in \( \ell \) and \( m \) and we can compute them efficiently by sieving. All other numbers calculated in the transformation are a product or sum over these prime numbers (each occurring at most once in the calculation) and thus their values are also in \( \text{poly}(\ell, m) \). We can compute the inverse \( (2^4 \cdot p^* \prod_{i=1}^{m'} p_i + K)^{-1} \) in polynomial time [19].

Having this theorem at hand we still have to bound the size of the generated numbers to apply the ETH. Denote by \( p = p_1, \ldots, p_{n_{QC}} \) the prime factorization of \( \beta \). The above reduction yields the following parameters:

**Theorem 3.** An instance of the 3-SAT problem with \( \ell \) variables and \( m \) clauses is reducible to an instance of the QUADRATIC CONGRUENCES problem in polynomial time with the properties that \( n_{QC} \in \mathcal{O}(e + m^2) \), \( \max_i \{p_i\} \in \mathcal{O}((e + m)^2 \log(e + m)) \), \( \alpha, \beta, \gamma \in \mathcal{O}(e + m)^2 \) and each prime factor occurs at most once except the prime factor 2 which occurs four times.

**Proof.** In Theorem 2 we already showed and proved a reduction from the 3-SAT problem to the QUADRATIC CONGRUENCES problem and argued the running time. It remains to bound the parameters. To do so we will bound the numbers occurring in the reduction above in order of their appearance.
After eliminating the trivial clauses it obviously holds that \( m' \leq m \) and \( \ell' \leq \ell \). Next we calculate \( \tau' \). Its absolute value can be bounded as

\[
|\tau'| = \left| - \sum_{i=1}^{m'} \prod_{j=1}^{i} p_j \right| \leq \prod_{i=1}^{m} p_j \leq m \prod_{j=1}^{m} p_j \leq m4^m \leq 4^O(m)
\]

as it holds that the product of the first \( k \) prime numbers is bounded by \( 4^k \) [9]. Similarly we bound \( \max_i(\{ f^i_+ \}, |f^i_-|) \leq \sum_{x_i \in \sigma_i} \prod_{k=1}^{i} p_k + \sum_{x_i \in \sigma_i} \prod_{k=1}^{i} p_k \leq 2m4^m \leq 4^O(m) \) and \( \max_j(\{ v_j \}) = \max_j(\{ \prod_{k=1}^{i} p_k, f^j_+ + f^j_- \}) \leq 4^O(m) \). Per definition \( n = 2m' + \ell = O(\ell + m) \). The largest prime number we generate in the reduction is \( p^* \), which is the \( (n^2 + 2n + 2m' + 13) \)th prime number. Thus its value is bounded by \( p^* \leq O(n^2 \log(n)) = O((\ell + m)^2 \log(\ell + m)) \) [8]. Due to the modulo we can bound \( \max_j(\{ \Theta_j \}) \) as

\[
\max_j(\{ \Theta_j \}) \leq 2^3 \cdot p^* \prod_{i=1}^{m'} p_i \cdot \prod_{i=0}^{n} \prod_{j \neq k} p_{i,k} \leq 2^4 4^{O((\ell + m)^2)} = 4^{O((\ell + m)^2)}
\]

Thus \( H = \sum_{i=0}^{n} \Theta_i \leq n \cdot 4^{O((\ell + m)^2)} = 4^{O((\ell + m)^2)} \) and \( K = \prod_{i=0}^{n} \prod_{k=0}^{n} p_{i,k} \leq 4^{O((\ell + m)^2)} \). Finally we can bound the main parameters. As \( \alpha \) is bounded by the modulo of \( \beta \) is follows that \( \alpha \leq \beta \). Further \( \beta = 2^4 \cdot p^* \prod_{i=1}^{m'} p_i \cdot K \leq 4^{O((\ell + m)^2)} \). Per definition \( \gamma = H \) and thus \( \gamma \leq 4^{O((\ell + m)^2)} \), which finalizes the estimation of the numbers.

3 Reduction from Quadratic Congruences to 2-stage stochastic ILPs

This sections presents the remaining reductions from the QUADRATIC CONGRUENCES problem to the so called MODULO REST DECISION problem and then finally the interpretation of the MODULO REST DECISION problem as the 2--stage stochastic INTEGER LINEAR PROGRAMMING problem. Recall that in the MODULO REST DECISION problem we are given numbers \( x_1, \ldots, x_{n_{\text{MRD}}}, y_1, \ldots, y_{n_{\text{MRD}}, q_1, \ldots, q_{n_{\text{MRD}}} \in \mathbb{N} \) where the \( q_i \)s are pairwise co-prime. The question is to decide whether the smallest natural number \( z \) greater zero satisfying the following Integer Linear Program is smaller or equal to \( \zeta \):

\[
\begin{align*}
z &\equiv \{ x_1, y_1 \} \mod q_1 \\
&\equiv \{ x_2, y_2 \} \mod q_2 \\
&\vdots \\
&\equiv \{ x_{n_{\text{MRD}}}, y_{n_{\text{MRD}}} \} \mod q_{n_{\text{MRD}}}. \\
\end{align*}
\]

Here \( \{ x_i, y_i \} \) means that either the residue \( x_i \) or \( y_i \) should be met. In other words we can re-write the equation as \( z \equiv x_i \mod q_i \) or \( z \equiv y_i \mod q_i \) for all \( i \). Thus the problems aims to find the smallest number, which is equivalent to either \( x_i \) or \( y_i \) for all equations \( i \) when calculating the modulo of the corresponding number \( q_i \). Indeed this problem becomes easy if \( x_i = y_i \) for all \( i \), i.e., we know the rest we want to satisfy for each equation [22]: First, compute \( s_i \) and \( r_i \) with \( r_i \cdot q_i + s_i \cdot \prod_{j=1,j \neq i}^{n_{\text{MRD}}} q_j = 1 \) for all \( i \) using the Extended Euclidean algorithm. Now it holds that \( s_i \cdot \prod_{j=1,j \neq i}^{n_{\text{MRD}}} q_j \equiv 1 \mod q_i \) as \( q_i \) and \( \prod_{j=1,j \neq i}^{n_{\text{MRD}}} q_j \) are coprime, and \( s_i \cdot \prod_{j=1,j \neq i}^{n_{\text{MRD}}} q_j \equiv 0 \mod q_j \) for \( j \neq i \). Thus the smallest solution corresponds to \( z = \sum_{i=1}^{n} x_i \cdot s_i \cdot \prod_{j=1,j \neq i}^{n_{\text{MRD}}} q_j \) due to the Chinese Remainder theorem [22]. Comparing \( z \) to the bound \( \zeta \) finally yields the answer. Also note that if \( n_{\text{MRD}} \) is constant we can solve the problem by testing all possible vectors \( (v_1, \ldots, v_{n_{\text{MRD}}} \) with \( v_i \in \{ x_i, y_i \} \) and then use the Chinese Remainder theorem as explained above.
\textbf{Theorem 4.} The Quadratic Congruences problem is reducible to the Modulo Rest Decision problem in polynomial time with the properties that \(n_{\text{MRD}} \in O(n_{\text{QC}})\), \(\max_{i \in \{1, \ldots, n_{\text{MRD}}\}} \{q_i, x_i, y_i\} = O(\max_{j \in \{1, \ldots, n_{\text{QC}}\}} \{p_j^{\beta_j}\})\), and \(\zeta \in O(\gamma)\).

\textbf{Proof.} Transformation: Set \(q_1 = p_1^{\beta_1}, \ldots, q_{n_{\text{MRD}}} = p_{n_{\text{MRD}}}^{\beta_{n_{\text{MRD}}}}\) and \(\zeta = \gamma\) where \(\beta_i\) denotes the occurrence of the prime factor in the prime factorization of \(\beta\). Compute \(\alpha_i \equiv \alpha \mod q_i\). Set \(x_i^2 = \alpha_i\) if there exists such an \(x_i \in \mathbb{Z}_{q_i}\). Further, compute \(y_i = -x_i + q_i\). If there is no such number \(x_i\) and thus \(y_i\), produce a trivial no-instance as the instance for the Quadratic Congruences problem has no solution. This can again be traced back to the Chinese Remainder theorem: If and only if there is an \(x\) with \(x^2 \equiv \alpha \mod \beta\) and \(q_1, \ldots, q_{n_{\text{MRD}}}\) is the prime factorization of \(\beta\) then \(x^2 \equiv \alpha_i \mod q_i\), \(\alpha_i \in \mathbb{Z}_{q_i}\) for all \(i\). Denote \(\alpha = (\alpha_1, \ldots, \alpha_{n_{\text{MRD}}})\). Hence if there does not exists a square root of \(\alpha\) in one of the systems then there is no vector \(\alpha\) and thus \(x^2 \equiv \alpha \mod \beta\) has no solution.

However, if we found such \(x_i\) and \(y_i\), both values are indeed in \(\mathbb{Z}_{q_i}\) as \(x_i \leq \alpha_i < q_i\) per definition of \(x_i\) and \(\alpha_i\). Further both values solve the problem \(x_i^2, y_i^2 \equiv \alpha \mod q_i\) as

\[x_i^2 \equiv \alpha_i \mod q_i \equiv \alpha + \rho \cdot q_i \mod q_i \equiv \alpha \mod q_i\]

for some \(\rho \in \mathbb{N}\). Further

\[y_i^2 \equiv (-x_i + q_i)^2 \mod q_i = q_i^2 - 2x_iq_i + x_i^2 \mod q_i \equiv x_i^2 \mod q_i \equiv \alpha \mod q_i.\]

The third equation holds as each summand except the last one is a multiple of \(q_i\). The last transformation is true due to the computation above. Note that for all primes greater 2 and \(\alpha \neq 0\) it holds that \(x_i \neq y_i\). This can easily be seen as we already argued that \(x_i\) and \(y_i\) are in \(\mathbb{Z}_{p_i}\). Let us suppose both values are equal, i.e.,

\[x_i^2 = y_i^2\]

\[\equiv \alpha_i = (-x_i + q_i)^2\]

\[\equiv \alpha_i = q_i^2 - 2q_ix_i + x_i^2\]

\[\equiv \alpha_i = q_i^2 - 2q_ix_i + \alpha_i\]

\[\equiv 2q_ix_i = q_i^2\]

\[\equiv 2x_i = q_i.\]

As \(q_i\) is a product of prime numbers greater than 2, there is no \(x_i\) satisfying the formula above. It follows that \(x_i^2\) and \(y_i^2\) and thus \(x_i\) and \(y_i\) are different numbers.

\textbf{Instance size:} The generated numbers equal the prime numbers of the Quadratic Congruences problem including their occurrence, hence ot holds that \(\max_{i \in \{1, \ldots, n_{\text{MRD}}\}} \{q_i\} = O(\max_{j \in \{1, \ldots, n_{\text{QC}}\}} \{p_j^{\beta_j}\})\). Due to the modulo this value also bounds \(x_i\) and \(y_i\). The upper bound on a solution equal the ones from the Quadratic Congruences instance as well as \(n_{\text{MRD}} = n_{\text{QC}}\).

\textbf{Correctness:} \(\Rightarrow\) Let the instance for Quadratic Congruences be a yes-instance. Then there exists a \(z\) satisfying \(z^2 \equiv \alpha \mod \beta\) with \(0 < z < \gamma\). Indeed this solution directly corresponds to a solution of the Modulo Rest Decision instance. First, \(z \leq \gamma = \zeta\). Second, \(z\) satisfies all equations as it holds that

\[z^2 \equiv \alpha \mod \beta \equiv \alpha \mod \prod_{i=1}^{n_{\text{MRD}}} p_i^{\beta_i} \equiv \alpha \mod p_i^{\beta_i} \text{ for all } i.\]
The first equivalence holds as the $p_i^{\beta_i}$’s are the prime factorization of $\beta$. The second equivalence is true as we can decompose the solution as follows: $z^2 = \pi \cdot \prod_{i=1}^{n_{MRD}} p_i^{\beta_i} + \alpha$ for some $\pi \in \mathbb{N}$. Thus the first summand is not only divided without rest by $\prod_{i=1}^{n_{MRD}} p_i^{\beta_i}$ but also by all primes along with their occurrences alone leaving only the second summand $\alpha$ as the rest.

Further, as $x_i^2, y_i^2 \equiv \alpha \mod q_i$ it holds that $z^2 \equiv x_i^2 \equiv y_i^2 \equiv \alpha \mod p_i^{\beta_i} \equiv \alpha \mod q_i$ for all $i$. Hence this satisfies all equations of the Modulo Rest Decision instance making it a yes-instance.

$\Leftarrow$ Let the instance for Modulo Rest Decision be a yes-instance. Thus we could verify that the minimal solution to the given equations is smaller than $\zeta$. Let this solution be denoted as $z^*$. It holds that $z^* \equiv x_i \mod q_i$ or $z^* \equiv y_i \mod q_i$. Let $v_i$ correspond to the residue that was satisfied, i.e., $v_i = x_i$ or $v_i = y_i$. This solution $z^*$ also solves the Quadratic Congruences problem. First, $z^* \leq \zeta = \gamma$. Further it holds per definition of the numbers, that

$$(z^*)^2 \equiv (v_i)^2 \equiv \alpha \mod q_i$$

As it satisfies all equations simultaneously it follows from the Chinese Remainder theorem that

$$(z^*)^2 \equiv (v_i)^2 \equiv \alpha \mod q_i \text{ for all } i$$

$$(z^*)^2 \equiv \alpha \mod \prod_{i=1}^{n_{MRD}} q_i \equiv \alpha \mod \prod_{i=1}^{n_{QC}} p_i^{\beta_i} \equiv \alpha \mod \beta$$

as the $p_i$’s are the prime factorization of $\beta$.

Running time: Setting the variables accordingly can be done in time polynomial in $n_{QC}$. Further computing each $x_i, y_i$ can be done in poly-logarithmic time regarding the largest absolute number for each $i \in \{1, \ldots, n_{MRD}\}$. 

Finally we reduce the Modulo Rest Decision problem to the 2-stage stochastic Integer Linear Programming problem. To do so we have to reformulate our 2-stage stochastic Integer Linear Programming problem as a decision problem. However, in the following reduction we indeed only seek to determine whether there exists a feasible solution. Thus we neither optimize a solution vector nor are we interested in the solution vector itself. Hence we reduce the Quadratic Congruences problem to the decision variant asking if the given 2-stage stochastic Integer Linear Programming problem instance is feasible implicitly.

\textbf{Theorem 5.} Let an instance of the Modulo Rest Decision problem be defined as above. The Modulo Rest Decision problem is reducible to the decision variant of the 2-stage stochastic Integer Linear Programming problem in polynomial time with the properties that $r, s, t, \max_1\{c_1\}, ||b||_\infty, ||L||_\infty \in O(1), ||U||_\infty \in O(\zeta)$, and $n \in O(n_{MRD})$, $\Delta \in O(\max\{q_i\})$.

\textbf{Proof.} Transformation: Having the instance for Modulo Rest Decision at hand we construct our Integer Linear Program as follows with $n = n_{MRD}$:
\( A \cdot x = \begin{pmatrix} -1 & q_1 & x_1 & y_1 & 0 & \ldots & 0 & 0 & \ldots & 0 \\ 0 & 0 & 1 & 1 & 0 & \ldots & 0 & 0 & \ldots & 0 \\ \vdots & \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots \\ -1 & 0 & \ldots & 0 & 0 & \ldots & 0 & q_n & x_n & y_n \end{pmatrix} \cdot x = b = \begin{pmatrix} 0 \\ 1 \\ \vdots \\ 0 \\ 1 \end{pmatrix} \)

Further, all variables get a lower bound \( L \) of 0 and an upper bound \( U \) of \( \zeta \). We can set the objective function arbitrarily as we are just searching for a feasible solution, hence we set it to \( w = (0, 0, \ldots, 0) \). Thus implicitly we turned this problem into a decision problem where we ask whether the Integer Linear Program admits a feasible solution.

**Instance size:** Due to our construction it holds that \( r = 2, s = 1, t = 3 \) and the number \( n \) of repeated blocks equals the number \( n_{\text{MRD}} \) of equations in the Modulo Rest Decision instance. The largest entry \( \Delta \) can be bounded by \( \max_i \{q_i\} \). The upper bounds \( U \) on the variables are at most \( \|U\|_\infty = O(\zeta) \). For the lower bound \( L \) it holds that \( \|L\|_\infty = O(1) \).

The objective function is set to zero and is thus of constant size. Finally the largest value in the right-hand side is 1.

**Correctness:** \( \Rightarrow \) Let the given instance for Modulo Rest Decision be a yes-instance. Thus there is a minimal solution \( z^* < \zeta \) satisfying all equations. Similar to before, let \( v_i \) correspond to the rest that was satisfied in each equation \( i \), i.e., \( v_i = x_i \) or \( v_i = y_i \). A solution to our Integer Linear Program now looks as follows: Set the first variable to \( z^* \). The columns corresponding to \( x_i \) and \( y_i \) be set as follows for each \( i \): If \( v_i = x_i \) then set this variable occurrence in the solution vector to 1. Set the occurrence to the corresponding variable of \( y_i \) to zero. Otherwise set the variables the other way round. Finally the variable corresponding to the columns of the \( p_i \) are computed as \( (z^* - v_i)/p_i \). It is easy to see that this solution is feasible and satisfies the bounds on the variable sizes.

\( \Leftarrow \) Let the given instance for the 2-stage stochastic Integer Linear Programming problem be a yes-instance. By definition of the constraint matrix we have for every \( 1 \leq i \leq n \) that there exists a multiple \( \lambda \geq 0 \) such that \( z = x_i + \lambda q_i \) or \( z = y_i + \lambda q_i \). Hence \( z \equiv x_i \mod q_i \) or \( z \equiv y_i \mod q_i \) for every \( 1 \leq i \leq n \), which is a solution to the Modulo Rest Decision problem.

**Running time:** Mapping the variables and computing the values for the \( p_i s \) can all be done in polynomial time regarding the largest occurring number and \( n \).

## 4 Runtime Bound for 2-Stage Stochastic ILPs under ETH

This sections presents the proof that the double exponent in the running time of the current state-of-the-art algorithms is nearly tight assuming the Exponential Time Hypothesis (ETH). To do so we make use of the reductions above showing that we can transform an instance of the 3-SAT problem to an instance of the 2-stage stochastic Integer Linear Programming.

\( \blacktriangleleft \textbf{Corollary 6.} \) The 2-stage stochastic Integer Linear Programming problem can not be solved in time less than \( 2^{\delta \sqrt{n}} \) for some \( \delta > 0 \) assuming \( \text{ETH} \).
Suppose the opposite. That is, there is an algorithm solving the 2-stage stochastic Integer Linear Programming problem in time less than $2^{4\sqrt{n}}$. Let an instance of the 3-SAT problem with $\ell$ variables and $m$ clauses be given. Due to the Sparsification lemma we may assume that $m \in O(\ell)$. We can reduce such an instance to an instance of the Quadratic Congruences problem in polynomial time regarding $\ell$ such that $n_{QC} \in O(\ell^2)$, $\max_i \{p_i\} \in O(\ell^2 \log(\ell))$, $\alpha, \beta, \gamma = 4^{O(\ell^2)}$, see Theorem 4.

Next we reduce this instance to an instance of the Modulo Rest Decision problem. By Theorem 4 this yields the parameter sizes $n_{MRD} \in O(\ell^2)$, $\max_i \{q_i, x_i, y_i\} = O(\ell^2 \log(\ell))$, and $\zeta \in 4^{O(\ell^2)}$. Note that all prime numbers greater 2 appear at most once in the prime factorization of $\beta$ and 2 appears 4 times. Thus the largest $q_i$, which corresponds to the largest value of a prime number with its occurrence as an exponent, equals the largest prime number in the Quadratic Congruences problem: The largest prime number is at least the $(n^2 + 2n + 2m + 13) \geq (13)$th prime number by a rough estimation. The $(13)$th prime number is 41 and thus larger than $2^4 = 16$.

Finally we reduce that instance to an instance of the 2-stage stochastic Integer Linear Programming problem can be solved in time less than $2^{4\sqrt{n}}$ this would result in the 3-SAT problem to be solved in time less than $2^{4\sqrt{n}} = 2^{4\sqrt{C_1\ell^2}} = 2^{4(C_2\ell^2)}$ for some constants $C_1, C_2$. Setting $\delta_2 \leq \delta/C_2$, this would violate the ETH.

To prove our main result we still have to reduce the size of the coefficients in the constraint matrix. To do so we encode large coefficients into submatrices. This will reduce their size greatly while just extending the matrix dimensions slightly. A similar approach was used for example in [15] to prove a lower bound for the size of inclusion minimal kern-elements of the 2-stage stochastic Integer Linear Programming problem or in [16] to lessen the value of $\Delta$ in matrices.

**Theorem 7.** The 2-stage stochastic Integer Linear Programming problem can not be solved in time less than $2^{2^{(r+\epsilon)}|I|^{O(1)}}$ for some constant $\delta > 0$, even if $s = 1$, $\Delta, ||b||_{\infty} \in O(1)$, assuming ETH. Here $|I|$ denotes the encoding length of the total input.

**Proof.** First we show that we can alter the resulting Integer Linear program such that we reduce the size of $\Delta$ to $O(1)$. We do so by encoding large coefficients with base 2, which comes at the cost of enlarged dimensions of the constraint matrix. Let $\text{enc}(x)$ be the encoding of a number $x$ with base 2. Further let $\text{enc}_x$ be the $i$th number of $\text{enc}(x)$. Finally $\text{enc}_0(x)$ will denote the last significant number of the encoding. Hence, the encoding of a number $x$ will be $\text{enc}(x) = \text{enc}_0(x)\text{enc}_1(x) \ldots \text{enc}_{|\log(\Delta)|}(x)$ and $x$ can be reconstructed by $x = \sum_{i=0}^{|\log(\Delta)|} \text{enc}_i(x) \cdot 2^i$.

Let a matrix $E$ be defined as follows:

$$
E = \begin{pmatrix}
2 & -1 & 0 & \ldots & 0 \\
0 & 2 & -1 & 0 & \ldots \\
\vdots & \ddots & \ddots & \ddots & \vdots \\
0 & \ldots & 0 & 2 & -1
\end{pmatrix}.
$$

We will rewrite the constraint matrix as follows: For each coefficient $c > 1$ we will insert its encoding $\text{enc}(c)$ and beneath the matrix $E$. Furthermore we have to fix the dimensions for the first row in the constraint matrix, the columns without great coefficients and the
right-hand side $b$ by filling the matrix at the corresponding positions with zeros. In detail, the altered Integer Linear Program $A \cdot x = b$ will look as follows. Note that the ones correspond to $\text{enc}_0(x_i)$ and $\text{enc}_0(y_i)$.

\[
\begin{pmatrix}
-1 & \text{enc}(q_1) & \text{enc}(x_1) & \text{enc}(y_1) & 0 & \ldots & 0 & 0 & \ldots & 0 \\
0 & E & 0 & 0 & 0 & \ldots & 0 & 0 & \ldots & 0 \\
\vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\
0 & 0 & 0 & 0 & E & 0 & \ldots & 0 & \ldots & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \ldots & 0 \\
\vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\
-1 & 0 & \ldots & 0 & 0 & \ldots & \text{enc}(q_n) & \text{enc}(x_n) & \text{enc}(y_1) & 0 \\
0 & 0 & 0 & \ldots & 0 & \ldots & 0 & 0 & \ldots & 0 \\
\vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\
0 & 0 & \ldots & 0 & 0 & \ldots & \ldots & 0 & 0 & \ldots & 0 \\
0 & 0 & \ldots & 0 & 0 & \ldots & 0 & 0 & \ldots & 0 & 0 \\
\vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \vdots \\
\end{pmatrix} \cdot x = \begin{pmatrix}
0 \\
\vdots \\
\vdots \\
0 \\
0 \\
\vdots \\
\end{pmatrix}
\]

The independent block consisting of $\text{enc}(c)$ and the matrix $E$ beneath correctly encodes some number $c$, i.e., it preserves the solution space: Let $x_c$ be the magnitude of the row with entry $c$ in a solution for the original problem. The solution for the altered entry is $(x_c \cdot 2^0, x_c \cdot 2^1, \ldots, x_c \cdot 2^{\lfloor \log(\Delta) \rfloor})$. The additional factor of 2 for each subsequent entry is due to the diagonal of $E$. It is easy to see that $c \cdot x_c = \sum_{i=0}^{\lfloor \log(\Delta) \rfloor} \text{enc}_i(c) \cdot x_c \cdot 2^i$ as we can extract $x_c$ on the right-hand side and solely the decoding of $c$ remains.

Regarding the dimensions, each coefficient $c > 1$ will be replaced by a $(O(\log(\Delta)) \times O(\log(\Delta)))$ matrix. Thus the dimension expands to $r' = r + O(\log(\Delta)) = O(\log(\Delta))$, $t' = t \cdot O(\log(\Delta)) = O(\log(\Delta))$, while $s, n, ||L||_{\infty}, ||U||_{\infty}$ stay the same. Further we get that the largest coefficient is bounded by $\Delta' = O(1)$. The right-hand side $b$ enlarges to a vector $b'$ with $O(n \log(\Delta))$ entries.

Now suppose there is an algorithm solving the 2-stage stochastic Integer Linear Programming problem in time less than $2^{2^{\Theta(r+\epsilon)}} |I|^{O(1)}$. The proof of Theorem 6 shows that we can transform an instance of the 3-SAT problem with $\ell$ variables and $m$ clauses to an instance of the 2-stage stochastic Integer Linear Programming problem with parameters $r, s, t, \max_i \{w_i\}, ||b||_{\infty}, ||L||_{\infty} \in O(1), ||U||_{\infty} \in 4^{O(\ell^2)}, n \in O(\ell^2)$, and $\Delta \in O(\ell^2 \log(\ell))$. Further we explained above that we can transform this matrix to an equivalent one where $r' = O(\log(\Delta)) = O(\log(\ell^2 \log(\ell))) = O(\log(\ell)), t' = t \cdot O(\log(\Delta)) = O(\log(\ell^2 \log(\ell))) = O(\log(\ell))$ and $\Delta' = O(1), b' \in Z^{\ell^2 \log(\ell)}$ while $s, n$ stay the same. The encoding length $|I|$ is then given by

$$|I| = (nr's' + nr't') \log(\Delta') + nr' \log(||L||_{\infty}) + nr' \log(||U||_{\infty}) + nr' \log(||b'||_{\infty}) + (s' + nt') \log(||w||_{\infty}) = 2^{O(\ell^2)}.$$ 

Hence if there is an algorithm solving the 2-stage stochastic Integer Linear Programming problem in time less than $2^{2^{\Theta(r+\epsilon)}} |I|^{O(1)}$ this would result in the 3-SAT problem to be solved in time less than $2^{2^{\Theta(r+\epsilon)}} |I|^{O(1)} = 2^{2^{O(1)}} = 2^{\Theta(C_1 \log(\ell) + C_2 \log(\ell)} 2^{\Theta(1)} = 2^{\Theta(C_1 \log(\ell) + C_2 \log(\ell)}} 2^{\Theta(1)} = 2^{\delta - C_3 \log(\ell)} = 2^{\delta - C_3} 2^{\Theta(1)}$ for some constants $C_1, C_2, C_3, C_4$. Setting $\delta = \delta / C_4$ we get $2^{\delta - C_3} = 2^{|x|}$. As it holds for sufficient large $x$ and $\epsilon < 1$ that $x^\epsilon < c \epsilon$ it follows that $2^{\delta - C_3} < 2^{|x|}$. This violates the ETH. Note that this result even holds if $s = 1, \Delta, ||b||_{\infty} \in O(1)$ as constructed by our reductions.
5 Conclusion and Open Questions

In this work we provided a new, stronger NP-hardness result for the Quadratic Congruences problem. In particular, we showed that the Quadratic Congruences problem remains NP-hard even if the prime factorization of $\beta$ is given and each prime factor greater than 2 occurs at most once and 2 occurs constantly often. Presenting a line of reductions from the 3-SAT problem to the 2-stage stochastic Integer Linear Programming problem utilizing this new hardness result we showed our main result: There is no algorithm solving the 2-stage stochastic Integer Linear Programming problem in time less than $2^{2^{\delta(r+t)}}|I|^{O(1)}$ for some constant $\delta > 0$ assuming ETH where $|I|$ denotes the encoding length of the total input. This result even holds if $s, \Delta, ||b||_\infty \in O(1)$. Thus we also showed that the current state-of-the-art algorithm is nearly tight, i.e., the double exponent is necessary making these ILPs harder to solve than the closely related $n$-fold ILPs — which essentially are the ILPs containing the transposed 2-stage stochastic matrix as the constraint matrix.

An extension of the 2-stage stochastic ILPs are so-called multi-stage stochastic ILPs where the constraint matrix is similar to the 2-stage stochastic ILPs. But the blocks along the diagonal are recursive multi-stage stochastic constraint matrices themselves. The current best algorithm has a running time with a tower of exponents with height equal to the recursion depth of the multi-stage stochastic constraint matrix \cite{7}. This arises the question whether the tower of exponents is indeed necessary or does the complexity collapses at a certain recursion depth?

References

1. Joseph Bertrand. Bertrand’s postulate chapter 2. Proofs from THE BOOK, page 9, 2018.
2. Cornelius Brand, Martin Koutecký, and Sebastian Ordyniak. Parameterized algorithms for milps with small treedepth. CoRR, abs/1912.03501, 2019.
3. Lin Chen, Lei Xu, Weidong Shi, and Martin Koutecký. New bounds on augmenting steps of block-structured integer programs. CoRR, abs/1805.03741, 2019.
4. Richard Crandall and Carl B. Pomerance. Prime numbers: a computational perspective, volume 182. Springer Science & Business Media, 2006.
5. Marek Cygan, Fedor V. Fomin, Łukasz Kowalik, Daniel Lokshtanov, Dániel Marx, Marcin Pilipczuk, Michał Pilipczuk, and Saket Saurabh. Parameterized Algorithms. Springer, 2015.
6. Friedrich Eisenbrand, Christoph Hunkenschröder, and Kim-Manuel Klein. Faster algorithms for integer programs with block structure. In ICALP, pages 49:1–49:13, 2018.
7. Friedrich Eisenbrand, Christoph Hunkenschroder, Kim-Manuel Klein, Martin Koutecký, Asaf Levin, and Shmuel Onn. An algorithmic theory of integer programming. CoRR, abs/1904.01361, 2019.
8. Godfrey H. Hardy, John Edensor Littlewood, et al. Contributions to the theory of the riemann zeta-function and the theory of the distribution of primes. Acta Mathematica, 41:119–196, 1916.
9. Godfrey H. Hardy and Edward M. Wright. An introduction to the theory of numbers (5. ed.). Clarendon Press, 1995.
10. Raymond Hemmecke, Matthias Köppe, and Robert Weismantel. A polynomial-time algorithm for optimizing over $N$-fold 4-block decomposable integer programs. In IPCO, volume 6080 of Lecture Notes in Computer Science, pages 219–229. Springer, 2010.
11. Raymond Hemmecke and Rüdiger Schultz. Decomposition of test sets in stochastic integer programming. Math. Program., 94(2-3):323–341, 2003.
12. Kenneth Ireland and Michael Rosen. A classical introduction to modern number theory, volume 84 of Graduate texts in mathematics. Springer, 1982.
K. Jansen, K. Klein, A. Lassota

13 Klaus Jansen, Alexandra Lassota, and Lars Rohwedder. Near-linear time algorithm for n-fold irlps via color coding. In ICALP, pages 75:1–75:13, 2019.
14 Peter Kall and Stein W. Wallace. Stochastic programming. Springer, 1994.
15 Kim-Manuel Klein. About the complexity of two-stage stochastic irlps. In IPCO, volume 12125 of Lecture Notes in Computer Science, pages 252–265. Springer, 2020.
16 Dusan Knop, Michal Pilipczuk, and Marcin Wrochna. Tight complexity lower bounds for integer linear programming with few constraints. In STACS, volume 126 of LIPIcs, pages 44:1–44:15. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2019.
17 Martin Koutecký, Asaf Levin, and Shmuel Onn. A parameterized strongly polynomial algorithm for block structured integer programs. In ICALP, pages 85:1–85:14, 2018.
18 Simge Küçükyavuz and Suvrajeet Sen. An introduction to two-stage stochastic mixed-integer programming. In Leading Developments from INFORMS Communities, pages 1–27. INFORMS, 2017.
19 Gabriel Lamé. Note sur la limite du nombre des divisions dans la recherche du plus grand commun dividé entre deux nombres entiers. 1844.
20 Kenneth L. Manders and Leonard M. Adleman. Np-complete decision problems for binary quadratics. J. Comput. Syst. Sci., 16(2):168–184, 1978.
21 Rüdiger Schultz, Leen Stougie, and Maarten H. Van Der Vlerk. Two-stage stochastic integer programming: a survey. Statistica Neerlandica, 50(3):404–416, 1996.
22 Stan Wagon. Mathematica in action. Springer Science & Business Media, 1999.