



HAL
open science

Simultaneous Rational Number Codes: Decoding Beyond Half the Minimum Distance with Multiplicities and Bad Primes

Matteo Abbondati, Eleonora Guerrini, Romain Lebreton

► To cite this version:

Matteo Abbondati, Eleonora Guerrini, Romain Lebreton. Simultaneous Rational Number Codes: Decoding Beyond Half the Minimum Distance with Multiplicities and Bad Primes. *Journal of Symbolic Computation*, 2026, 132, pp.102481. <10.1016/j.jsc.2025.102481>. <hal-05027939>

HAL Id: hal-05027939

<https://hal.science/hal-05027939v1>

Submitted on 10 Apr 2025

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire HAL, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



HAL Authorization

Simultaneous Rational Number Codes: Decoding Beyond Half the Minimum Distance with Multiplicities and Bad Primes

Matteo Abbondati, Eleonora Guerrini, Romain Lebreton

^a*LIRMM - University of Montpellier, 161, Rue Ada, Montpellier, 34095, FRANCE*

Abstract

In this paper, we extend the work of [AGL24] on decoding simultaneous rational number codes by addressing two important scenarios: multiplicities and the presence of bad primes (divisors of denominators). First, we generalize previous results to multiplicity rational codes by considering modular reductions with respect to prime power moduli. Then, using hybrid analysis techniques, we extend our approach to vectors of fractions that may present bad primes.

Our contributions include: a decoding algorithm for simultaneous rational number reconstruction with multiplicities, a rigorous analysis of the algorithm's failure probability that generalizes several previous results, an extension to a hybrid model handling situations where not all errors can be assumed random, and a unified approach to handle bad primes within multiplicities. The theoretical results provide a comprehensive probabilistic analysis of reconstruction failure in these more complex scenarios, advancing the state of the art in error correction for rational number codes.

1. Introduction

An efficient approach to solving linear systems in distributed computation involves reconstructing a vector of fractions $\left(\frac{f_1}{g}, \dots, \frac{f_\ell}{g}\right)$, all sharing the same denominator, from its modular reductions with respect to n pairwise coprime elements. In this framework, a network is structured around a central node, which selects a sequence of relatively prime elements $(m_j)_{1 \leq j \leq n}$ and delegates the system solving process to the network. Each node j computes the solution modulo m_j and transmits the reduced solution vector $(f_i/g \bmod m_j)_{1 \leq i \leq \ell}$ back to the central node. The central node then reconstructs the original vector through an interpolation step, formulated as a simultaneous rational reconstruction problem. In the case of polynomial systems, this approach is known as evaluation-interpolation [KPSW17, GLZ19], whereas for integer systems, it corresponds to modular reduction followed by reconstruction via the Chinese Remainder Theorem [Cab71]. In this paper, we focus on the latter case.

Context of this paper. During data reconstruction, the central node may receive incorrect reductions due to computational errors, faulty or untrusted nodes, or network noise. For that reason, it is of great help to look at decoding algorithms in error correcting codes. Viewing the modular reductions as coordinates of an error correcting code enables us to reconstruct the correct solution as long as the number of erroneous reductions is below a certain value, corresponding to the unique decoding radius of a code. In presence of more errors, there exist two possible approaches in coding theory to correct beyond the unique decoding radius of the code; either decoding algorithms which return a list of all codewords within a certain distance of the received word (list decoding) or, by interleaving techniques, obtain positive decoding results under probabilistic assumptions on random errors corrupting ℓ code-words on the same positions. In this paper, we focus on interleaving techniques as they fit in the simultaneous reconstruction problem. Note that, a decoding algorithm working under this latter approach must inevitably fail for some instances, as beyond unique decoding radius there can be many codewords around a given instance. Here the failure probability is intended as the proportion of received words, within a given distance from the codeword \mathbf{f}/g , for which the reconstruction fails.

In this work we consider the simultaneous rational reconstruction problem with $m_j = p_j^{\lambda_j}$ for a sequence of distinct prime numbers p_1, \dots, p_n and relative multiplicities $\lambda_1, \dots, \lambda_n > 0$. One advantage of considering reductions with multiplicities is that solving a linear system modulo p^λ is asymptotically faster than solving it modulo p_1, \dots, p_λ (see [MC79, Dix82, Sto05] or [Leb12, Chapter 3] for a survey).

The prime numbers for which the modular reductions are not defined (divisors of the denominator g) are referred to as *bad primes*. To the best of our knowledge, this work represents the first study of rational number codes in a context with multiplicities and bad primes.

Taking inspiration from [KPR⁺10], we could define the rational number code in terms of modular reductions to a generic sequence of n coprime ideals (not necessarily of the form $(p_j^{\lambda_j})$). From a purely mathematical perspective (thanks to the Chinese Remainder theorem), the approach of [KPR⁺10] where coordinates are defined via modular reductions relative to any sequence of n coprime ideals, is equivalent to ours, where each coprime ideal is generated by the power of a prime element. The advantage of the approach proposed here is that it allows for greater specificity in both the coordinates and the description of the errors affecting them.

Previous results. The approach of this paper generalizes, and matches or even improves several previous results in different ways. In the polynomial case, the codes used for the recovery of a vector of polynomials from partially erroneous evaluations are Interleaved Reed-Solomon codes (IRS), whose best known analysis of the decoding failure probability is provided in [SSB09] and then generalized to the rational function case in [GLZ19].

The integer counterpart of IRS codes are to the so-called Interleaved Chinese remainder codes (ICR), for which a first heuristic analysis of the decoding failure

probability was provided in [LSN13] and made rigorous in [AAGL23].

While there have been various studies on the rational function case [KPSW17, Zap20, GLLZ23], the rational number context had not been investigated until [AGL24].

In any case the extensive literature addressing these problems both in the polynomial [McC77, BK14, GLZ19, KPY20, GLLZ23] and the integer [Cab71, Lip71, AAGL23] contexts rarely shows unified methods, and the techniques used are very specific to the case studied. In [AGL24] the authors analyzing both the rational functions and the rational numbers reconstruction problems (in absence of multiplicities and poles/bad primes), proved it is possible to recover the correct solution vector for almost all instances.

Contributions of this paper. The main results presented are the following:

- A decoding approach to address the simultaneous rational number reconstruction with errors (Problem 1.2) including multiplicities, as well as the relative decoding algorithm (Algorithm 1).
- A detailed analysis of the failure probability of the algorithm, that generalizes several previous results in [AGL24]: see Theorem 2.18 and Theorem 2.19.
- The extension of the analysis to a hybrid model including random and non-random errors, addressing situations where not all errors can be assumed random: see Theorem 3.2, Theorem 3.3.
- The merging of the hybrid model with our decoding approach, to handle bad primes within multiplicities, and relative decoding failure analysis: see Theorem 4.16 and Theorem 4.17.

Our methodology and our results can also be adapted to the rational function case. For the sake of readability, we have chosen to focus on the simultaneous reconstruction of rational numbers in this paper. However, it is worth reporting that adapting our results would improve upon the existing analysis of [GLLZ23] in the sense that the failure probability bound obtained decreases exponentially (not linearly) with respect to the decoding algorithm distance parameter, and the dependency on the choice of the multiplicities can be removed (see multiplicity balancing in [GLLZ23, Theorem 3.4]).

1.1. Notations and preliminary definitions

We will denote vectors with bold letters $\mathbf{f}, \mathbf{r}, \mathbf{c}, \dots$. For $m \in \mathbb{Z}$ with $\mathbb{Z}/m\mathbb{Z}$ we will denote the quotient ring modulo the ideal (m) , while $[x]_m$ will denote the modular element $x \bmod m \in \mathbb{Z}/m\mathbb{Z}$ and $\mathcal{P}(m)$ will denote the set of primes dividing m . Given an indexed family of rings $\{A_j\}_{1 \leq j \leq n}$, we let $\prod_{j=1}^n A_j$ be their Cartesian product.

Given a vector of modular reductions $\mathbf{r} \in \prod_{j=1}^n \mathbb{Z}/p_j^{\lambda_j}$ we use the corresponding capital letter R denote its unique interpolant constructed via the Chinese remainder theorem modulo $N := \prod_{j=1}^n p_j^{\lambda_j}$.

We let $\text{val}_p : \mathbb{Z} \rightarrow \mathbb{N} \cup \{\infty\}$ be the valuation function over \mathbb{Z} with respect to the prime number p , whose output is the highest power of p dividing the input, where we set by convention its value to be ∞ when the input is 0.

Dealing with a fixed sequence of precisions $\lambda_1, \dots, \lambda_n$, we truncate the valuation function considering $\nu_{p_j}(m) := \min\{\text{val}_{p_j}(m), \lambda_j\}$, so that $\nu_{p_j}(a) = \nu_{p_j}(b)$ when $a = b \pmod{p_j^{\lambda_j}}$.

When computing the valuation of a vector we set $\nu(\mathbf{f}) := \min_i\{\nu(f_i)\}$. Given the sequence of multiplicities $\lambda_1, \dots, \lambda_n > 0$, we define the parameter $L := \sum_{j=1}^n \lambda_j$. For us all the vectors of fractions \mathbf{f}/g sharing the same denominator will always be reduced, i.e. they satisfy $\gcd(\gcd(\mathbf{f}), g) = 1$.

Simultaneous rational number reconstruction with errors (SRNRwE). To quantify errors and to establish the correction capacity of the code we are going to use, we need a notion of distance between words. In a context with multiplicities where the coordinates are modular reductions relative to moduli specified by different precisions $\lambda_1, \dots, \lambda_n$, it is classical to consider (see for example [KPY20, GLLZ23]) a minimal error index distance in which each modular reduction is regarded as a truncated development, and the whole tail starting from the first error index in such development is considered erroneous. Furthermore, to take into account that each coordinate depends on a different prime number p_j , it is classical to use a weighted Hamming distance (see for example [AGL24]), thus we are going to consider the following definition:

Definition 1.1 (Distance - Integer case). Let $\mathbf{R}^1, \mathbf{R}^2 \in (\prod_{j=1}^n \mathbb{Z}/p_j^{\lambda_j} \mathbb{Z})^\ell$ be two $\ell \times n$ matrices, where each column $\mathbf{r}_j^1, \mathbf{r}_j^2$ belongs to $(\mathbb{Z}/p_j^{\lambda_j})^\ell$. We define their error support as $\xi_{\mathbf{R}^1, \mathbf{R}^2} := \{j : \mathbf{r}_j^1 \neq \mathbf{r}_j^2\}$ and their error locator as the product $\Lambda_{\mathbf{R}^1, \mathbf{R}^2} := \prod_{j \in \xi_{\mathbf{R}^1, \mathbf{R}^2}} p_j^{\lambda_j - \mu_j}$, where $\mu_j := \nu_{p_j}(\mathbf{r}_j^1 - \mathbf{r}_j^2)$ represents the minimal error index for the development around the prime p_j . The distance between \mathbf{R}^1 and \mathbf{R}^2 is defined as $d(\mathbf{R}^1, \mathbf{R}^2) := \log_2(\Lambda_{\mathbf{R}^1, \mathbf{R}^2})$.

The problem of simultaneous rational number reconstruction with errors is then:

Problem 1.2 (SRNRwE). Given $\ell > 0$, n distinct primes $p_1 < \dots < p_n$ with associated multiplicities $\lambda_1, \dots, \lambda_n$, a received matrix $\mathbf{R} \in (\prod_{j=1}^n \mathbb{Z}/p_j^{\lambda_j} \mathbb{Z})^\ell$, an error parameter d and two bounds F, G such that $FG < N/2$, find a reduced vector of fractions $(f_1/g, \dots, f_\ell/g) \in \mathbb{Q}^\ell$ such that

1. $d\left(\left([f_i/g]_{p_j^{\lambda_j}}\right)_{i,j}, \mathbf{R}\right) \leq d$,
2. for all $1 \leq i \leq \ell$, $|f_i| < F$, $0 < g < G$ and $\gcd(g, N) = 1$.

In the above we have that $\gcd(g, N) = 1$ so that the reductions $[f_i/g]_{p_j^{\lambda_j}}$ are well-defined. We are going to drop this hypothesis in Section 4, when solving a more general version of the SRNRwE problem, allowing for the presence of bad primes.

This problem can be reduced to the simultaneous error correction of ℓ code words (sharing the same denominator) for the multiplicity version of rational number codes. Without multiplicities (i.e. when N is square-free) this code is the natural rational extension of Chinese remainder codes [GRS99], and can be referred to as rational number codes, extensively studied in [AGL24]. It seems these rational codes were part of the folklore; to the best of our knowledge, they were formally introduced in the language of coding theory by Pernet in [Per14, § 2.5.2], whereas [BDFP15] works with redundant residue number systems.

The condition $FG < N/2$ guarantees an injective encoding, whose proof will be given in Proposition 4.2 when introducing the multi-precision encoding (see Definition 4.1) which is a generalization of our current encoding in presence of bad primes.

A long series of papers can be found in the literature where evaluation-interpolation is used for linear systems solving, as [McC77, Vil97, Mon04, OS07, RS16]. Our contributions in this paper concern error correction beyond guaranteed uniqueness. This means that the solution to the problem will not always be unique. In this rare case, our decoding algorithm returns a decoding failure. We analyze the probability of failure in detail.

The paper is structured as follows: In Section 2 we introduce the simultaneous rational number codes whose decoding solves Problem 1.2 as well as the corresponding decoding Algorithm 1. We study the failure probability of our decoding algorithm for error parameters larger than the unique decoding radius of the code. We note that this analysis generalizes the results of [AGL24] to the multiplicity case, it thus follows the same broad lines except for some technical details (see Lemma 2.26).

In Section 3, we adapt our analysis technique to the hybrid distribution model of [GLLZ23] in which not all errors are supposed to be random, but some of them are fixed, either because of specific error patterns introduced by malicious entities or because of specific faults of the network nodes.

Then, in Section 4, by considering the multi-precision encoding of [GLLZ23], we apply the hybrid approach to generalize our analysis to the case of reductions with multiplicities and bad primes, *i.e.* we drop the hypothesis $\gcd(g, N) = 1$.

2. Simultaneous multiplicity rational number codes

We can define an error correcting code associated to Problem 1.2. Code words are the encoding of reduced vectors of rational numbers $(f_1/g, \dots, f_\ell/g)$ sharing the same denominator and such that $0 < g < G$, and $|f_i| < F$ for all $i = 1, \dots, \ell$.

Definition 2.1. Given n distinct primes p_1, \dots, p_n with relative multiplicities $\lambda_1, \dots, \lambda_n$, two positive bounds F, G such that $FG < N/2$ and an integer $\ell > 0$, we define the *simultaneous multiplicity rational number code* as the set of

matrices

$$\text{SRN}_\ell(N; F, G) := \left\{ \left(\left[\frac{f_i}{g} \right]_{p_j}^{\lambda_j} \right)_{\substack{1 \leq i \leq \ell \\ 1 \leq j \leq n}} : \begin{array}{l} |f_i| < F, \quad 0 < g < G, \\ \gcd(f_1, \dots, f_\ell, g) = 1 \\ \gcd(N, g) = 1 \end{array} \right\}.$$

We will refer to SRN codes for short if parameters are not relevant.

Note that when $G = 2$ and $\ell = 1$, we obtain the interleaving of $\text{RN}_\ell(N; F, G)$ codes and if $\ell > 1$, then $\text{RN}_\ell(N; F, G)$ gives the rational codes with multiplicity as described in [Per14, §2.5.2]. When dealing with rational numbers, the denomination *simultaneous* comes from the rational function case, where it is related to simultaneous rational function reconstruction, *i.e.* the variant of Problem 1.2 for rational functions without errors [OS07, RS16, GLZ20].

In the next section, we will see that the common denominator property is necessary to be able to take advantage in the key equations of the fact that the ℓ RN codewords share the same error supports.

The condition $\gcd(f_1, \dots, f_\ell, g) = 1$, which is going to be used in the proof of Lemma 2.23, reflects that the solution vector we seek to reconstruct is a reduced vector of rational numbers.

Remark 2.2. A bounded distance decoding algorithm for the above code which is able to correct errors up to a distance d , can be used to solve Problem 1.2 with error parameter d .

2.1. Minimal distance

The distance $d(\mathcal{C}) := \min_{c_1 \neq c_2 \in \mathcal{C}} d(c_1, c_2)$ of a code \mathcal{C} plays an important role in coding theory to assess the amount of data one can correct. A classic result states that one can correct up to half of the Minimal distance and that there is no guarantee on the decoding success beyond this quantity.

Theorem 2.3. *Let N, F, G as in Definition 2.1. The distance of an RN code satisfies $d(\text{RN}(N; F, G)) > \log\left(\frac{N}{2FG}\right)$.*

This result has the advantage of being independent of the moduli p_j . However, the gap between $d(\text{RN}(N; F, G))$ and $\log(N/(2FG))$ depends on the moduli. Even so, there exists a family of RN codes such that $d(\text{RN}(N; F, G)) \leq \log(N/((F-1)(G-1)))$, *i.e.* the gap is small [Per14, §2.5.2]. We can generalize Theorem 2.3 to SRN codes with multiplicities as follows:

Lemma 2.4. *We have $d(\text{SRN}_\ell(N; F, G)) > \log\left(\frac{N}{2FG}\right)$.*

Proof. Let $\mathbf{C}_1 = \left(\left[\frac{f_i}{g} \right]_{p_j}^{\lambda_j} \right)_{i,j}$ and $\mathbf{C}_2 = \left(\left[\frac{f'_i}{g'} \right]_{p_j}^{\lambda_j} \right)_{i,j}$ be two code words.

Setting $Y := \prod_{j \notin \xi_{\mathbf{C}_1, \mathbf{C}_2}} p_j^{\mu_j}$, with $\mu_j = \nu_{p_j} \left(\left[\frac{f}{g} \right]_{p_j}^{\lambda_j} - \left[\frac{f'}{g'} \right]_{p_j}^{\lambda_j} \right)$. Since $\gcd(Y, g) = \gcd(Y, g') = 1$, we have that $Y | (fg' - f'g)$. Since $\|f\|_\infty, \|f'\|_\infty < F$, and $0 < g, g' < G$ we have $Y < 2FG$. Using the relation $Y = N/\Lambda_{\mathbf{C}_1, \mathbf{C}_2}$, we bound $d(\mathbf{C}_1, \mathbf{C}_2) = \log(\Lambda_{\mathbf{C}_1, \mathbf{C}_2}) = \log(N/Y) > \log(N/2FG)$. \square

2.2. Unique decoding

A unique decoding function D of capacity d is a function from the ambient space to the code such that $D(r) = c$ for all code word c and all r such that $d(r, c) \leq t$. For codes equipped with the Hamming distance, there exists such a decoding function of capacity d if and only if $2d < d(\mathcal{C})$. Pernet gives a polynomial time unique decoding algorithm for RN codes of capacity $\log(\sqrt{N/(2FG)}) = (1/2) \log(N/(2FG))$ for the weighted Hamming distance [Per14, Corollary 2.5.2]. Note that if no such decoding function exists, then no decoding algorithm can exist.

For SRN codes equipped with the weighted Hamming distance, the result is slightly different. If $2d < d(\mathcal{C})$, then there exists such a decoding function of capacity d . However, the converse is false in the strict sense of the term. Indeed, whereas proving that there can not exist a decoding function when $2d = d(\mathcal{C})$, one takes $c_1, c_2 \in \mathcal{C}$ such that $d(\mathcal{C}) = d(c_1, c_2)$, and constructs r as the middle of c_1 and c_2 , *i.e.* with $d(c_1, r) = d(c_2, r) = d(c_1, c_2)/2$. Thanks to that, we obtain the contradiction that a decoding function would have to map r to both c_1 and c_2 . However, it is impossible to construct r as the middle of c_1 and c_2 with the weighted Hamming distance associated to distinct primes. Still, the essence of the result remains correct, and if $2d = d(\mathcal{C}) + \varepsilon$ for a small ε , then we can construct r such that $d(c_1, r), d(c_2, r) \leq (d(c_1, c_2) + \varepsilon)/2 = d$, and no decoding function of capacity d can exist.

Thanks to Lemma 2.4, we know that a unique decoding function of capacity d for SRN codes can exist only if $d < \log(\sqrt{N/2FG})$ (see Proposition 4.14 for a proof in the case of bad primes).

One workaround in coding theory, when no unique decoding function can exist, consists of having decoding functions which can output "decoding failure" when the code word within the decoding capacity is not unique.

The aim of the paper is to properly analyze the decoding failure probability of a decoding algorithm for SRN codes beyond the uniqueness capacity. It is worth of note that our decoding algorithm (Algorithm 1), despite being aimed at correcting errors beyond unique decoding, outputs the unique decoding solution whenever $d < \log(\sqrt{N/2FG})$ (see Remark 2.24).

2.3. Decoding SRN codes

This section presents our first contribution: a decoder of SRN codes of capacity beyond $\frac{d(\mathcal{C})}{2}$. This decoder, is a slight modification of the decoder presented in [AGL24] for SRN codes without multiplicities, and it is based on the interleaved Chinese remainder (ICR) codes decoder of [LSN13, AAGL23], which are a special case of SRN when $g = 1$ and N is square-free. Let $\mathbf{R} := (r_{i,j})_{\substack{1 \leq i \leq \ell \\ 1 \leq j \leq n}}$ be the received matrix.

For any code word $\mathbf{C} \in \text{SRN}_\ell(N; F, G)$, we can write $\mathbf{R} = \mathbf{C} + \mathbf{E}$ for some error matrix \mathbf{E} (which depends on \mathbf{R} and \mathbf{C}). Thanks to the Chinese remainder theorem, we can view each row of the matrix as modular elements in $\mathbb{Z}/N\mathbb{Z}$, and

the ambient space for the code can be viewed as \mathbb{Z}_N^ℓ , thus for every $1 \leq i \leq \ell$ we can write $R_i = C_i + E_i$ with $C_i = [f_i/g]_N$ for some f_i, g .

Letting $\Lambda := \Lambda_{\mathbf{C}, \mathbf{R}} = \prod_{j \in \xi_{\mathbf{C}, \mathbf{R}}} p_j^{\lambda_j - \mu_j}$, with $\mu_j = \nu_{p_j}([f/g]_{p_j^{\lambda_j}} - r_j)$ we conclude that the system of ℓ equations holds:

$$\Lambda f_i = \Lambda g R_i \pmod{N} \text{ for } i = 1, \dots, \ell \quad (1)$$

with unknowns $\Lambda, g, f_1, \dots, f_\ell$.

We linearize it thanks to the substitution $\varphi \leftarrow \Lambda g$ and $\psi_i \leftarrow \Lambda f_i$; the resulting equations

$$\psi_i = \varphi R_i \pmod{N} \text{ for } i = 1, \dots, \ell \quad (2)$$

are called the *key equations*. The solutions $(\varphi, \psi_1, \dots, \psi_\ell)$ are vectors in the lattice $\mathcal{L} \subseteq \mathbb{Z}^{\ell+1}$ spanned by the rows of the integer matrix

$$\mathcal{L} = \text{Span} \begin{pmatrix} 1 & R_1 & \cdots & R_\ell \\ 0 & N & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & N \end{pmatrix}. \quad (3)$$

In particular if $\Lambda \leq 2^d$ for some distance parameter d , the solution vector $v_{\mathbf{C}} := (\Lambda g, \Lambda f_1, \dots, \Lambda f_\ell)$ belongs to the set

$$S_{\mathbf{R}, 2^d} := \{(\varphi, \psi_1, \dots, \psi_\ell) \in \mathcal{L} : 0 < \varphi < 2^d G, |\psi_i| < 2^d F\}.$$

Note that the condition $\Lambda_{\mathbf{C}, \mathbf{R}} \leq 2^d$ means that \mathbf{C} is close to \mathbf{R} for the weighted Hamming distance.

The decoding strategy consists in compute an element of $S_{\mathbf{R}, 2^d}$ and try to recover $v_{\mathbf{C}}$ by dividing all the entries by the first one in order to obtain $(f_1/g, \dots, f_\ell/g)$. There are two main aspects inherent to this procedure. The first one is algorithmic, and it is relative to a choice of how to compute an element in $S_{\mathbf{R}, 2^d}$, the second one is probabilistic, and it is relative to the estimation of the probability that this element is a multiple of the solution vector $v_{\mathbf{C}}$. Concerning the analysis of this second aspect, more will be said in Section 2.8. For the moment we wish to describe the algorithmic aspect at a high level of generality. For this we will assume to have at our disposal an algorithm \mathcal{ASVP}_∞ which solves the following problem:

Problem 2.5 ($\text{SVP}_{\|\cdot\|_\infty}^\beta$). Given a basis $\{v_0, \dots, v_\ell\}$ of a lattice \mathcal{L} and an approximation constant $\beta \geq 1$, find a non-zero vector $w \in \mathcal{L}$ such that $\|w\|_\infty \leq \beta \lambda_\infty(\mathcal{L})$, where $\lambda_\infty(\mathcal{L})$ is the minimum $\|\cdot\|_\infty$ -norm of the non-zero vectors in \mathcal{L} .

We refer the reader to [AM18] for state-of-the-art algorithms solving Problem 2.5. Without loss of generality, we will assume that the output w of the algorithm \mathcal{ASVP}_∞ satisfies $w_0 \geq 0$ (both $\pm w$ are short vectors). We will also assume that w is \mathcal{L} -reduced:

Definition 2.6. Given a lattice \mathcal{L} , a vector $v \in \mathcal{L}$ is said to be \mathcal{L} -reduced if, for $c \in \mathbb{Z} \setminus \{0\}$, $(1/c) \cdot v \in \mathcal{L} \Rightarrow c = \pm 1$.

Because the size constraints in $S_{\mathbf{R}, 2^d}$ do not correspond exactly to conditions on the $\|\cdot\|_\infty$ norm, we need to introduce a scaling operator $\sigma_{F,G} : \mathbb{Q}^{\ell+1} \rightarrow \mathbb{Q}^{\ell+1}$ such that $\sigma_{F,G}((v_0, v_1, \dots, v_\ell)) := (v_0 F, v_1 G, \dots, v_\ell G)$. This scaling will transform \mathcal{L} into the scaled lattice $\bar{\mathcal{L}} := \sigma_{F,G}(\mathcal{L})$, and our solution set $S_{\mathbf{R}, 2^d}$ into

$$S'_{\mathbf{R}, 2^d} := \sigma_{F,G}(S_{\mathbf{R}, 2^d}) = \{(\varphi, \psi_1, \dots, \psi_\ell) \in \bar{\mathcal{L}} : 0 < \varphi < 2^d F G, |\psi_i| < 2^d F G\}.$$

Therefore, a vector $v' \in \bar{\mathcal{L}}$ which satisfies $\|v'\|_\infty < 2^d F G$ must belong to $S'_{\mathbf{R}, 2^d}$. A candidate solution v_s can be obtained by computing a scaled short vector $\bar{v}_s := \mathcal{ASVP}_\infty(\bar{\mathcal{L}})$, and unscaling it $v_s := \sigma_{F,G}^{-1}(\bar{v}_s)$.

We can now prove that, provided that \mathbf{R} is relatively close to the code (see Constraint 2.7 below), since v_s is a β -approximation of the shortest vector, it belongs to a slightly larger solution set.

Constraint 2.7. There exists a code word \mathbf{C} such that $\Lambda_{\mathbf{C}, \mathbf{R}} \leq 2^d$.

Lemma 2.8. Assuming Constraint 2.7, we have that $v_s \in S_{\mathbf{R}} := S_{\mathbf{R}, 2^d \beta}$.

Proof. We know that $\|\bar{v}_s\|_\infty \leq \beta \lambda_\infty(\bar{\mathcal{L}}) \leq \beta \|\sigma_{F,G}(v_{\mathbf{C}})\|_\infty < \beta \Lambda F G \leq \beta 2^d F G$.

Since we assumed that $(\bar{v}_s)_0 \geq 0$, we have $\bar{v}_s \in S'_{\mathbf{R}, 2^d \beta}$ and $v_s \in S_{\mathbf{R}, 2^d \beta}$. \square

We notice that assuming Constraint 2.7 we also have $v_{\mathbf{C}} \in S_{\mathbf{R}}$. Following the error model of SRN codes, one could independently decode each row, which corresponds to an RN code, but the information that the errors share the same support would not be exploited. So, instead, we perform so-called collaborative decoding of ℓ RN codeword together, that is a SRN code word, to take advantage of this common support. We can now state our decoding algorithm for SRN codes.

Algorithm 1: SRN codes decoder.

Input: $\text{SRN}_\ell(N; F, G)$, received word \mathbf{R} , distance bound d

Output: A code word \mathbf{C} s.t. $d(\mathbf{C}, \mathbf{R}) \leq d$ or “decoding failure”

- 1 Let $\bar{\mathcal{L}} := \sigma_{F,G}(\mathcal{L})$ be the scaled lattice of \mathcal{L} defined in Equation (3)
 - 2 Compute a short vector $\bar{v}_s := \mathcal{ASVP}_\infty(\bar{\mathcal{L}})$
 - 3 Unscale the vector: $v_s = (\varphi, \psi_1, \dots, \psi_\ell) := \sigma_{F,G}^{-1}(\bar{v}_s)$
 - 4 Let $\eta := \text{gcd}(\varphi, \psi_1, \dots, \psi_\ell)$, $\varphi' := \varphi/\eta$ and $\forall j, \psi'_j := \psi_j/\eta$
 - 5 **if** $\eta \leq 2^d$, $\text{gcd}(\varphi', N) = 1$, $|\varphi'| < G$ and $\forall j, |\psi'_j| < F$ **then**
 - 6 **return** $(C_1, \dots, C_\ell) := (\psi'_1/\varphi', \dots, \psi'_\ell/\varphi')$
 - 7 **else return** "decoding failure";
-

2.4. A particular sub-routine: LLL

We remark that the complexity of Algorithm 1 is mainly determined by the complexity of the sub-routine \mathcal{ASVP}_∞ . In particular the authors of [AM18] showed that the space and time complexity for the resolution of Problem 2.5 are significantly larger than the relative costs for the resolution of the ℓ_2 -norm version of the same problem.

Problem 2.9 ($\text{SVP}_{\|\cdot\|_2}^\gamma$). Given a basis $\{v_0, \dots, v_\ell\}$ of a lattice \mathcal{L} and an approximation constant $\gamma \geq 1$, find a non-zero vector $w \in \mathcal{L}$ such that $\|w\|_2 \leq \gamma \lambda_2(\mathcal{L})$, where $\lambda_2(\mathcal{L})$ is the minimum $\|\cdot\|_2$ -norm of the non-zero vectors in \mathcal{L} .

Remark 2.10. A γ -approximation SVP for the ℓ_2 -norm yields a $\gamma\sqrt{\ell+1}$ -approximation SVP for the ℓ_∞ -norm: If $w = \mathcal{ASVP}_2(\mathcal{L})$ and s_2 (resp. s_∞) is one of the shortest vector for the ℓ_2 -norm (resp. ℓ_∞ -norm), then $\|w\|_\infty \leq \|w\|_2 \leq \gamma \|s_2\|_2 \leq \gamma \|s_\infty\|_2 \leq \gamma\sqrt{\ell+1} \|s_\infty\|_\infty$.

A well known example of algorithm solving Problem 2.9 is given by LLL [LLL82], which runs in polynomial time for the approximation factor $\gamma = \sqrt{2}^\ell$ (our lattice has dimension $\ell+1$). As Algorithm 1 does not use LLL as subroutine to compute a short vector \bar{v}_s , we are going to assume that the approximation constant β satisfies the following constraint:

Constraint 2.11. The approximation constant β satisfies: $\beta < 3^\ell$.

Thanks to the above remark, Constraint 2.11 is automatically satisfied if using LLL as subroutine, it is enough to notice that $\beta = \gamma\sqrt{\ell+1} = \sqrt{2}^\ell \sqrt{\ell+1} \leq 3^\ell$.

The most efficient $\text{SVP}_{\|\cdot\|_2}^\gamma$ solver is given by the BKZ algorithm [Sch87]. It finds a solution of Problem 2.9 with $\gamma = (1+\epsilon)^{\ell+1}$ in polynomial time of degree increasing as $\epsilon \rightarrow 0$.

Furthermore, since the output of LLL or BKZ is always the first vector of a basis of the lattice, the following Lemma will ensure that it is \mathcal{L} -reduced.

Lemma 2.12. *Let $\{b_1, \dots, b_n\}$ be a basis of a lattice \mathcal{L} , then every vector b_i is \mathcal{L} -reduced.*

Proof. If $\frac{1}{c}b_i \in \mathcal{L}$ for some $c \in \mathbb{Z} \setminus \{0\}$, then we can write $\frac{1}{c}b_i = \sum_{j=1}^n c_j b_j$ for some $c_j \in \mathbb{Z}$. Thus, $b_i = \sum_{j=1}^n cc_j b_j$, which means that $cc_i = 1$, so $c = \pm 1$. \square

2.5. Correctness of Algorithm 1

In this section, we study the correctness of Algorithm 1. We start with Lemma 2.13 which states that the algorithm is correct when it does not fail.

Lemma 2.13. *If Algorithm 1 returns \mathbf{C} on input \mathbf{R} and parameter d , then \mathbf{C} is a code word of $\text{SRN}(N; F, G)$ such that $d(\mathbf{C}, \mathbf{R}) \leq d$.*

Proof. The output vector $\mathbf{C} = (\psi'_1/\varphi', \dots, \psi'_\ell/\varphi')$ is a code word of $\text{SRN}(N; F, G)$ since the algorithm has verified the size conditions $|\varphi'| < G$, $|\psi'_j| < F$ for all j , and that $\gcd(\varphi', N) = 1$. Now, we use that $(\varphi, \psi_1, \dots, \psi_\ell) = (\eta\varphi', \eta\psi'_1, \dots, \eta\psi'_\ell)$ is in the lattice \mathcal{L} , so that $\eta(\varphi'R_i - \psi'_i) = 0 \pmod N$ for all i . Dividing by the invertible φ' modulo N , we obtain $\eta(R_i - C_i) = 0 \pmod N$ for all i , which implies that $\nu_{p_j}(\eta) \geq \lambda_j - \mu_j = \nu_{p_j}(\Lambda_{\mathbf{C}, \mathbf{R}})$. Thus, $\Lambda_{\mathbf{C}, \mathbf{R}}|\eta \leq 2^d$, and we can conclude that $d(\mathbf{C}, \mathbf{R}) = \log \Lambda_{\mathbf{C}, \mathbf{R}} \leq \log \eta \leq d$. \square

Next lemma shows that, when the algorithm fails, the short vector v_s computed by sub-routine \mathcal{ASVP}_∞ is not collinear to $v_{\mathbf{C}}$.

Lemma 2.14. *Assuming Constraint 2.7, if Algorithm 1 fails, then $v_s \notin v_{\mathbf{C}}\mathbb{Z}$.*

Proof. By contraposition, let's prove that if $v_s = rv_{\mathbf{C}}$ for some $r \in \mathbb{Z}$, then the algorithm must succeed. We know that $v_s = rv_{\mathbf{C}}$ is \mathcal{L} -reduced therefore $v_{\mathbf{C}} = \pm v_s$ and $\eta = \Lambda \leq 2^d$ using Constraint 2.7 (see Algorithm 1, Step 4 for η), $\varphi' = \pm g, \psi'_j = \pm f_j$ for every j , thus the algorithm succeeds. \square

Remark 2.15. We emphasize here that the failure of Algorithm 1 is due to the size of the distance parameter d (when larger than the unique decoding capacity), and not to the approximation factor coming from the subroutine \mathcal{ASVP}_∞ . When $d > \log(\sqrt{N}/(2FG))$ the algorithm might sometimes fail even if $\beta = 1$.

The rest of this section is dedicated to the analysis of the decoding failure of Algorithm 1. We will show that if \mathbf{R} is \mathbf{C} plus a random error of weighted Hamming distance up to approximately $\ell/(\ell+1) \log(N/(2FG))$ (see Section 2.6 for precise error models), then this decoder is able to decode most of the time (see Section 2.7 for the statement of the theorem).

2.6. Error models

Algorithm 1 must fail on some instances when the distance parameter d exceeds the maximum distance for which the uniqueness of the solution of Problem 1.2 is guaranteed.

We analyze the failure probability of the algorithm under two different classical error models in Coding Theory, already considered in previous papers [SSB09, AAGL23, AGL24], specifying two possible distributions of the random received word \mathbf{R} .

Error Model 1. In this error model, we fix an error locator Λ among the divisors of N , then we let \mathcal{E}_Λ^1 be the set of error matrices \mathbf{E} whose columns satisfy:

1. $\mathbf{e}_j = \mathbf{0}$ for all j such that $p_j \notin \mathcal{P}(\Lambda)$,
2. $\nu_{p_j}(\mathbf{e}_j) = \lambda_j - \nu_{p_j}(\Lambda)$ for all j such that $p_j \in \mathcal{P}(\Lambda)$.

For any given code word \mathbf{C} and error locator Λ , the distribution $\mathcal{D}_{\mathbf{C}}^{\mathcal{E}_\Lambda^1}$ of random received words \mathbf{R} around the central code word \mathbf{C} is defined as $\mathbf{R} = \mathbf{C} + \mathbf{E}$ for \mathbf{E} uniformly distributed in \mathcal{E}_Λ^1 .

We will need another point of view on the random error matrices \mathbf{E} . For $i \in \{1, \dots, \ell\}$, we denote $E_i \in \mathbb{Z}/N\mathbb{Z}$ the CRT interpolant of the i -th row of \mathbf{E} . By definition of the error valuation μ_j , letting $Y := N/\Lambda = \prod_{j=1}^n p_j^{\mu_j}$, we have that $Y|E_i$ for every index $i = 1, \dots, \ell$. We define the modular integers $E'_i := E_i/Y \in \mathbb{Z}/\Lambda\mathbb{Z}$.

Since $\mu_j = \nu_{p_j}(\mathbf{E}) = \min_i \{\nu_{p_j}(E_i)\}$, we see that $Y = \gcd(E_1, \dots, E_\ell, N)$, and that the random vector $(E'_i)_{1 \leq i \leq \ell}$ is uniformly distributed in the sample space

$$\Omega_\Lambda := \{(F_i)_{1 \leq i \leq \ell} \in (\mathbb{Z}/\Lambda\mathbb{Z})^\ell : \gcd(F_1, \dots, F_\ell, \Lambda) = 1\}. \quad (4)$$

As we will need a more general version of Ω_Λ (for example in the proof of Lemma 2.25), we state the following:

Lemma 2.16. *Given $\Lambda \in \mathbb{Z}$ and $\eta = \prod_{p \in \mathcal{P}(\Lambda)} p_j^{\eta_j}$ be a divisor of Λ , then letting $\bar{\Omega}_{\Lambda, \eta} := \{(F_i)_{1 \leq i \leq \ell} \in (\mathbb{Z}/\Lambda\mathbb{Z})^\ell : \gcd(F_1, \dots, F_\ell, \Lambda) = \eta\}$ we have*

$$\#\bar{\Omega}_{\Lambda, \eta} = \left(\frac{\Lambda}{\eta}\right)^\ell \prod_{p \in \mathcal{P}(\Lambda/\eta)} \left(1 - \frac{1}{p^\ell}\right)$$

Proof. Thanks to the Chinese Remainder Theorem, for every $i = 1, \dots, \ell$, we can factor each of the ℓ copies of the quotient space $\mathbb{Z}/\Lambda\mathbb{Z}$ with respect to the factors of Λ , and obtain that $\bar{\Omega}_{\Lambda, \eta}$ has the same cardinality as

$$\left\{ (\varphi_j) \in \prod_{p_j \in \mathcal{P}(\Lambda)} \left(\mathbb{Z}/p_j^{\nu_{p_j}(\Lambda)}\mathbb{Z}\right)^\ell : \nu_{p_j}(\varphi_j) = \eta_j \right\}.$$

By counting the p_j -adic vectorial expansion coefficients of φ_j , we can compute the cardinality of the above set as

$$\prod_{\eta_j < \nu_{p_j}(\Lambda)} p_j^{\ell(\nu_{p_j}(\Lambda) - \eta_j - 1)} (p_j^\ell - 1) = \left(\frac{\Lambda}{\eta}\right)^\ell \prod_{p \in \mathcal{P}(\Lambda/\eta)} \left(1 - \frac{1}{p^\ell}\right). \quad \square$$

Error Model 2. In this error model we fix a maximal error locator Λ_m among the divisors of N , then we let $\mathcal{E}_{\Lambda_m}^2$ be the set of error matrices \mathbf{E} whose columns satisfy:

1. $\mathbf{e}_j = \mathbf{0}$ for all j such that $p_j \notin \mathcal{P}(\Lambda_m)$,
2. $\nu_{p_j}(\mathbf{e}_j) \geq \lambda_j - \nu_{p_j}(\Lambda_m)$ for all j such that $p_j \in \mathcal{P}(\Lambda_m)$.

We notice that in the error model $\mathcal{E}_{\Lambda_m}^2$, the actual error locator Λ could be a divisor of Λ_m . For a code word \mathbf{C} and a maximal error locator Λ_m , the distribution $\mathcal{D}_{\mathbf{C}}^{\mathcal{E}_{\Lambda_m}^2}$ of random received words \mathbf{R} around the central code word \mathbf{C} is defined as $\mathbf{R} = \mathbf{C} + \mathbf{E}$ for \mathbf{E} uniformly distributed in $\mathcal{E}_{\Lambda_m}^2$.

2.7. Our Results

In this section we present our contributions to the analysis of the decoding failure depending on the given parameters. The error models previously defined will play a role in the latter but not in the choice of parameters. We define a common framework for the algorithm parameters, while in Subsection 2.8 we will adapt the analysis of the failure probability to the two error models specified above. In what follows we set

$$\bar{d} := \frac{\ell}{\ell+1} \left[\log \left(\frac{N}{2FG} \right) - \log(3\beta) \right]. \quad (5)$$

Remark 2.17. Our setting allows decoding up to a distance $d \leq \bar{d}$ that, for $\ell > 1$, can be greater than our estimation $\log \left(\sqrt{\frac{N}{2FG}} \right)$ of the unique decoding capability of $\text{SRN}_\ell(N; F, G)$ codes.

When fixing the decoding bound d close to \bar{d} , we are likely to correct beyond the unique decoding radius, so we must deal with decoding failure for some received word. Note that this remains valid even if $\mathcal{ASVP}_\infty(\mathcal{L})$ gives us the exact short vector (i.e. $\beta = 1$).

Here is our first result (whose proof will be given at the end of Subsection 2.8.1) relative to the failure probability of the decoding algorithm with respect to the error model \mathcal{E}_Λ^1 .

Theorem 2.18. *Decoding Algorithm 1 on input distance parameter $d \leq \bar{d}$ and a random received word \mathbf{R} uniformly distributed in $D_{\mathbf{C}}^{\mathcal{E}_\Lambda^1}$, for some code word $\mathbf{C} \in \text{SRN}_\ell(N; F, G)$ and error locator Λ such that $\log \Lambda \leq d$, outputs the center code word \mathbf{C} of the distribution $\mathcal{D}_{\mathbf{C}}^{\mathcal{E}_\Lambda^1}$, with a probability of failure*

$$\mathbb{P}_{\text{fail}} \leq 2^{-(\ell+1)(\bar{d}-d)} \prod_{p \in \mathcal{P}(\Lambda)} \left(\frac{1 - 1/p^{\ell+\nu_p(\Lambda)}}{1 - 1/p^\ell} \right).$$

Here is our second result (whose proof will be given at the end of Subsection 2.8.2) relative to the failure probability with respect to the error model $\mathcal{E}_{\Lambda_m}^2$.

Theorem 2.19. *Decoding Algorithm 1 on input distance parameter $d \leq \bar{d}$ and a random received word \mathbf{R} uniformly distributed in $\mathcal{D}_{\mathbf{C}}^{\mathcal{E}_{\Lambda_m}^2}$, for some code word $\mathbf{C} \in \text{SRN}_\ell(N; F, G)$ and maximal error locator Λ_m such that $\log \Lambda_m \leq d$, outputs the center code word \mathbf{C} of the distribution $\mathcal{D}_{\mathbf{C}}^{\mathcal{E}_{\Lambda_m}^2}$, with a probability of failure*

$$\mathbb{P}_{\text{fail}} \leq 2^{-(\ell+1)(\bar{d}-d)} \prod_{p \in \mathcal{P}(\Lambda_m)} \left(\frac{1 - 1/p^{\ell+\nu_p(\Lambda_m)}}{1 - 1/p^{\ell+1}} \right).$$

This failure probability bound improves the one of decoding interleaved Chinese remainder codes $\mathbb{P}_{\text{fail}} \leq 2^{-(\ell+1)(\bar{d}-d)} + (\exp(n/p_1^{\ell-1}) - 1)$ which was only

available in the special case of non-negative ($0 \leq f_i$) integer code words ($G = 2$) without multiplicities ($\lambda_j = 1$) [AAGL23, Theorem 3.5]. We remark that both results reduce to [AGL24, Theorem 17 and 18] respectively, when there are no multiplicities in the modular reductions of the code, *i.e.* when N is square-free.

We note that in both theorems the product over the primes dividing the error locator is close to one; indeed we can prove the following lemma.

Lemma 2.20. *Assuming that $p_1 = \min_i \{p_i\}$, given $\eta|N$ divisor of N and $f(\ell)$ any function of the parameter $\ell > 0$, we have that*

$$\prod_{p \in \mathcal{P}(\eta)} \left(\frac{1 - 1/p^{\ell + \nu_p(\eta)}}{1 - 1/p^{f(\ell)}} \right) \leq \frac{1}{1 - n/p_1^{f(\ell)}}.$$

Proof. We start noticing that for each factor in the product we have

$$\frac{1 - 1/p^{\ell + \nu_p(\eta)}}{1 - 1/p^{f(\ell)}} \leq \frac{1}{1 - 1/p_1^{f(\ell)}}$$

Furthermore $\prod_{p \in \mathcal{P}(\eta)} (1 - 1/p^{f(\ell)}) \geq (1 - 1/p_1^{f(\ell)})^n \geq 1 - n/p_1^{f(\ell)}$, from which the statement follows. \square

Remark 2.21. We give a scenario which highlights how Theorem 2.19 can be used in practice. Assume that a code is fixed such that $\log(N/(6FG\beta)) = 20$, so that with an interleaving parameter $\ell = 4$, one has $\bar{d} = 16$. If one wishes to ensure that the failure probability is less than a target probability of 2^{-30} , then Theorem 2.19 states that choosing the distance parameter of the decoder $d = 10$, ensures that for any random error uniformly distributed on a maximal error locator Λ_m such that $\log \Lambda_m \leq d$, the failure probability is less than 2^{-30} .

2.8. Analysis of the decoding failure probability

For any \mathbf{R} uniformly distributed in $\mathcal{D}_{\mathbf{C}}^{\varepsilon_{\Lambda}^1}$ (as in Theorem 2.18), Constraint 2.7 is satisfied. Thus, thanks to Lemma 2.8, we can assume that $v_s \in S_{\mathbf{R}} = S_{\mathbf{R}, 2^d \beta}$.

2.8.1. Decoding failure probability with respect to the first error model

If Algorithm 1 fails, then $v_s \notin v_{\mathbf{C}}\mathbb{Z}$ (see Lemma 2.14). Note that the converse is not necessarily true, for example if there exists another close code word $\mathbf{C}' \neq \mathbf{C}$ with $d(\mathbf{C}', \mathbf{R}) \leq d$ and if the SVP solver outputs $v_s = v_{\mathbf{C}'}$.

Nevertheless, we can upper bound the failure probability of the algorithm as $\mathbb{P}_{fail} \leq \mathbb{P}(S_{\mathbf{R}} \not\subseteq v_{\mathbf{C}}\mathbb{Z})$. We introduce some notations: for $C \in \mathbb{R}_{>0}$ we let $\mathbb{Z}_{m,C} := \{a \in \mathbb{Z}/m\mathbb{Z} : |a \text{ crem } m| \leq C\}$, where $a \text{ crem } m$ is the central remainder of a modulo m , that is the unique representative of a modulo m within the interval $[-\lceil m/2 \rceil + 1, \lfloor m/2 \rfloor]$. Note that this set has cardinality $\#\mathbb{Z}_{m,C} \leq 2\lfloor C \rfloor + 1$. Let $S_{\mathbf{E}}$ be the set $S_{\mathbf{E}} := \{\varphi \in \mathbb{Z}/\Lambda\mathbb{Z} : \forall i, g\varphi E'_i \in \mathbb{Z}_{\Lambda, B\Lambda}\}$ for $B := 2^d \beta \frac{2FG}{N}$.

We need a new constraint to prove the following lemma.

Constraint 2.22. Algorithm 1 parameters satisfy $B < 1$.

Lemma 2.23. If Constraint 2.22 is satisfied, $S_{\mathbf{E}} = \{0\} \Rightarrow S_{\mathbf{R}} \subseteq v_{\mathbf{C}}\mathbb{Z}$.

Proof. Let $(\varphi, \psi_1, \dots, \psi_\ell) \in S_{\mathbf{R}} = S_{\mathbf{R}, 2^d\beta}$. We know that for all $1 \leq i \leq \ell$, $g\varphi E_i = g\varphi \left(R_i - \frac{f_i}{g}\right) = g\psi_i - f_i\varphi \pmod{N}$. Since $Y|E_i$ and $Y|N$, thanks to the above, we have that $Y|(g\psi_i - f_i\varphi)$, and we define the integer $\psi'_i = \frac{g\psi_i - f_i\varphi}{Y}$. Dividing the above modular equation by Y we obtain $g\varphi E'_i = \psi'_i \pmod{\Lambda}$. Therefore,

$$|g\varphi E'_i \pmod{\Lambda}| \leq |\psi'_i| \leq \frac{|g\psi_i| + |f_i\varphi|}{Y} < 2^d\beta \frac{2FG}{N}\Lambda = B\Lambda$$

which means that $\varphi \in S_{\mathbf{E}}$, thus thanks to the hypothesis $S_{\mathbf{E}} = \{0\}$, we get $\Lambda|\varphi$, thus $g\varphi E'_i = \psi'_i = 0 \pmod{\Lambda}$. Thanks to Constraint 2.22 and the above inequality we can conclude that $|\psi'_i| < \Lambda$, therefore $\psi'_i = 0$ in \mathbb{Z} . Which means that

$$\forall i = 1, \dots, \ell, \quad g\psi_i = f_i\varphi. \quad (6)$$

Since $\gcd(f_1, \dots, f_\ell, g) = 1$, Equations (6) imply that $g|\varphi$. We have already seen that $\Lambda|\varphi$, so $g\Lambda|\varphi$ because g and Λ are coprime. Plugging $\varphi = ag\Lambda$ for some $a \in \mathbb{Z}$ into Equations (6), we deduce $g\psi_i = f_i\varphi = f_iag\Lambda$, so $\psi_i = a\Lambda f_i$ for all i . We have shown $(\varphi, \psi_1, \dots, \psi_\ell) \in (\Lambda g, \Lambda f_1, \dots, \Lambda f_\ell)\mathbb{Z}$. \square

Thanks to the above lemma we can upper bound the failure probability of Algorithm 1 with $\mathbb{P}_{fail} \leq \mathbb{P}(S_{\mathbf{E}} \neq \{0\})$.

Remark 2.24. We note that, when the distance parameter d of the decoding algorithm is below the unique decoding capacity of SRN codes, *i.e.* $d < \log(\sqrt{N}/(2FG))$, we must have that $B\Lambda < \beta$ since $\Lambda \leq 2^d$. As pointed out in Remark 2.15, it is not because of the approximation factor that Algorithm 1 might fail, thus, at the cost of using an exact SVP solver, *i.e.* a subroutine \mathcal{ASVP}_∞ returning the shortest vector of $\tilde{\mathcal{L}}$, we can assume $\beta = 1$. Note that polynomial time exact SVP solver exist for constant dimension ℓ . Under such circumstance we therefore have $\mathbb{Z}_{\Lambda, B\Lambda} = \mathbb{Z}_{\Lambda, 0} = \{0\}$, thus estimating the failure probability of Algorithm 1 by studying $\mathbb{P}(S_{\mathbf{E}} \neq \{0\})$ yields the expected unique decoding result when $d < \log(\sqrt{N}/(2FG))$.

In order to estimate $\mathbb{P}(S_{\mathbf{E}} \neq \{0\})$, we need the following preliminary result:

Lemma 2.25. If $\varphi \in \mathbb{Z}$ is such that $\gcd(\varphi, \Lambda) = \eta = \prod_{j \in \xi} p_j^{\eta_j}$, then for the probability distribution of error model \mathcal{E}_Λ^1 , we have

$$\mathbb{P}(\forall i, g\varphi E'_i \in \mathbb{Z}_{\Lambda, B\Lambda}) \leq \frac{(\#\mathbb{Z}_{\Lambda/\eta, B\Lambda/\eta})^\ell}{\left(\frac{\Lambda}{\eta}\right)^\ell \prod_{p \in \mathcal{P}(\frac{\Lambda}{\eta})} (1 - 1/p^\ell)}$$

If we also suppose $B < \eta/\Lambda < 1$, then $\mathbb{P}(\forall i, g\varphi E'_i \in \mathbb{Z}_{\Lambda, B\Lambda}) = 0$.

Proof. Since $\gcd(g, N) = 1$, the distributions of the vectors $(\varphi E'_1, \dots, \varphi E'_\ell)$ and $(g\varphi E'_1, \dots, g\varphi E'_\ell)$ over the sample space

$$\Omega_\Lambda := \{(F_i)_{1 \leq i \leq \ell} \in (\mathbb{Z}/\Lambda\mathbb{Z})^\ell : \gcd(F_1, \dots, F_\ell, \Lambda) = 1\},$$

are identical. Thus, we have $\mathbb{P}(\forall i, g\varphi E'_i \in \mathbb{Z}_{\Lambda, B\Lambda}) = \mathbb{P}(\forall i, \varphi E'_i \in \mathbb{Z}_{\Lambda, B\Lambda})$.

Let us now show that $\varphi E'_i \in \mathbb{Z}_{\Lambda, B\Lambda} \Leftrightarrow (\varphi/\eta)E'_i \in \mathbb{Z}_{\Lambda/\eta, B\Lambda/\eta}$: The first condition can be rephrased as $\varphi E'_i = a_i\Lambda + c_i$ with $a_i, c_i \in \mathbb{Z}$ and $|c_i| \leq B\Lambda$. But then we must have that $\eta|c_i$. Thus, we can divide the above by η and obtain $(\varphi/\eta)E'_i = a_i\Lambda/\eta + c_i/\eta$ with $|c_i/\eta| \leq B\Lambda/\eta$, which is equivalent to $(\varphi/\eta)E'_i \in \mathbb{Z}_{\Lambda/\eta, B\Lambda/\eta}$.

When $B\Lambda < \eta$, the previous condition implies that $(\varphi/\eta)E'_i = 0 \pmod{\Lambda/\eta}$ for all i . Since φ/η is coprime with Λ/η , we have $E'_i = 0 \pmod{\Lambda/\eta}$ for all i . If $\eta < \Lambda$, this is in contradiction with $\gcd(E'_1, \dots, E'_\ell, \Lambda) = 1$ for all random matrix \mathbf{E} . Therefore, the associated probability $\mathbb{P}(\forall i, g\varphi E'_i \in \mathbb{Z}_{\Lambda, B\Lambda})$ is zero.

We have seen that

$$\mathbb{P}(\forall i, g\varphi E'_i \in \mathbb{Z}_{\Lambda, B\Lambda}) = \mathbb{P}(\{\mathbf{E} = (\mathbf{e}_j)_{1 \leq j \leq n} : \forall i, (\varphi/\eta)E'_i \in \mathbb{Z}_{\Lambda/\eta, B\Lambda/\eta}\}),$$

and since $\gcd(\Lambda/\eta, \varphi/\eta) = 1$, the above reduces to

$$\mathbb{P}(\{\mathbf{E} = (\mathbf{e}_j)_{1 \leq j \leq n} : \forall i, E'_i \in \mathbb{Z}_{\Lambda/\eta, B\Lambda/\eta}\}).$$

Now, the condition $E'_i \in \mathbb{Z}_{\Lambda/\eta, B\Lambda/\eta}$ only depends on the columns (\mathbf{e}'_j) of the reduced random matrix for $j \in \xi_{\Lambda/\eta} := \{j : \eta_j < \nu_{p_j}(\Lambda)\}$. These columns are uniformly distributed in the sample space $\bar{\Omega}_{\Lambda, \eta}$.

Therefore, letting $\Upsilon := \{\mathbf{E} = (\mathbf{e}_j)_{1 \leq j \leq n} : \forall i, E'_i \in \mathbb{Z}_{\Lambda/\eta, B\Lambda/\eta}\}$, we note that $\#\Upsilon = (\#\mathbb{Z}_{\Lambda/\eta, B\Lambda/\eta})^\ell$, and we can deduce that our probability equals

$$\mathbb{P}(\Upsilon) = \frac{\#(\bar{\Omega}_{\Lambda, \eta} \cap \Upsilon)}{\#\bar{\Omega}_{\Lambda, \eta}} \leq \frac{\#\Upsilon}{\#\bar{\Omega}_{\Lambda, \eta}}.$$

Finally, Lemma 2.16 tells us that $\#\bar{\Omega}_{\Lambda, \eta} = \left(\frac{\Lambda}{\eta}\right)^\ell \prod_{p \in \mathcal{P}(\frac{\Lambda}{\eta})} (1 - 1/p^\ell)$. \square

Before proving our results we still need the following technical lemma.

Lemma 2.26. *Given $\Lambda \in \mathbb{Z}$ and $f(x, y)$ an arbitrary real-valued function of two variables. Then*

$$\sum_{\eta|\Lambda} \prod_{p \in \mathcal{P}(\eta)} f(p, \nu_p(\eta)) = \prod_{p \in \mathcal{P}(\Lambda)} \left[1 + \sum_{k=1}^{\nu_p(\Lambda)} f(p, k) \right]$$

Proof. By expanding the product on the right-hand side we obtain

$$\prod_{p \in \mathcal{P}(\Lambda)} \left[1 + \sum_{k=1}^{\nu_p(\Lambda)} f(p, k) \right] = \sum_{S \subseteq \mathcal{P}(\Lambda)} \sum_{\substack{(\eta_p)_{p \in S} \\ 1 \leq \eta_p \leq \nu_p(\Lambda)}} \prod_{p \in S} f(p, \eta_p).$$

The double sum above corresponds exactly to a single sum over the divisors η of Λ with $S = \mathcal{P}(\eta)$ and $\eta_p = \nu_p(\eta)$. \square

Rewriting $\{\mathbf{E} : S_{\mathbf{E}} \neq \{0\}\}$ as $\cup_{\varphi=1}^{\Lambda-1} \{\mathbf{E} : \varphi \in S_{\mathbf{E}}\}$, we get

$$\mathbb{P}(S_{\mathbf{E}} \neq \{0\}) \leq \sum_{\varphi=1}^{\Lambda-1} \mathbb{P}(\forall i, g\varphi E'_i \in \mathbb{Z}_{\Lambda, B\Lambda}) = \sum_{\varphi=1}^{\Lambda-1} \mathbb{P}(\forall i, \varphi E'_i \in \mathbb{Z}_{\Lambda, B\Lambda}) \quad (7)$$

where the last equality comes from the proof of Lemma 2.25. We analyze the latter quantity in the following lemma.

Lemma 2.27. *Given a random vector (E'_1, \dots, E'_ℓ) uniformly distributed in Ω_{Λ} , we have that*

$$\sum_{\varphi=1}^{\Lambda-1} \mathbb{P}(\forall i, \varphi E'_i \in \mathbb{Z}_{\Lambda, B\Lambda}) \leq (3B)^\ell \Lambda \prod_{p \in \mathcal{P}(\Lambda)} \left(\frac{1 - 1/p^{\ell + \nu_p(\Lambda)}}{1 - 1/p^\ell} \right).$$

Proof. We can use Lemma 2.25 and upper bound the terms in the sum with

$$\mathbb{P}(\forall i, \varphi E'_i \in \mathbb{Z}_{\Lambda, B\Lambda}) \leq \frac{(\#\mathbb{Z}_{\Lambda/\eta, B\Lambda/\eta})^\ell}{\left(\frac{\Lambda}{\eta}\right)^\ell \prod_{p \in \mathcal{P}(\frac{\Lambda}{\eta})} \left(1 - \frac{1}{p^\ell}\right)}$$

where $\eta = \gcd(\varphi, \Lambda)$. Thanks to the second point in Lemma 2.25, we can restrict the sum only to the elements φ such that $\eta \leq B\Lambda$, which in turn allows us to deduce that $\#\mathbb{Z}_{\Lambda/\eta, B\Lambda/\eta} \leq 2\lfloor B\Lambda/\eta \rfloor + 1 \leq 3B\Lambda/\eta$. Since this expression depends only on η , we regroup the φ in the sum by their gcd with Λ . Note that the number of elements $\varphi \in \mathbb{Z}_{\Lambda}$ such that $\gcd(\varphi, \Lambda) = \eta$, is equal to $\phi\left(\frac{\Lambda}{\eta}\right)$ with ϕ being the Euler's totient function. Therefore,

$$\sum_{\substack{\varphi=1 \\ \eta = \gcd(\varphi, \Lambda) \leq B\Lambda}}^{\Lambda-1} \frac{(\#\mathbb{Z}_{\Lambda/\eta, B\Lambda/\eta})^\ell}{\left(\frac{\Lambda}{\eta}\right)^\ell \prod_{p \in \mathcal{P}(\frac{\Lambda}{\eta})} (1 - 1/p^\ell)} \leq \sum_{\substack{\eta|\Lambda \\ \eta \leq B\Lambda}} \frac{\phi\left(\frac{\Lambda}{\eta}\right) \left(\frac{3B\Lambda}{\eta}\right)^\ell}{\left(\frac{\Lambda}{\eta}\right)^\ell \prod_{p \in \mathcal{P}(\frac{\Lambda}{\eta})} (1 - 1/p^\ell)}.$$

Extending the sum over all the divisors η , we can upper bound the quotient $\mathbb{P}(S_{\mathbf{E}} \neq \{0\}) / (3B)^\ell$ with

$$\sum_{\eta|\Lambda} \frac{\phi\left(\frac{\Lambda}{\eta}\right)}{\prod_{p \in \mathcal{P}(\frac{\Lambda}{\eta})} \left(1 - \frac{1}{p^\ell}\right)} = \sum_{\eta|\Lambda} \prod_{p \in \mathcal{P}(\eta)} \frac{1 - \frac{1}{p}}{1 - \frac{1}{p^\ell}} p^{\nu_p(\eta)} = \prod_{p \in \mathcal{P}(\Lambda)} \left(1 + \frac{1 - \frac{1}{p}}{1 - \frac{1}{p^\ell}} \sum_{k=1}^{\nu_p(\Lambda)} p^k\right)$$

where in the last equality we used Lemma 2.26 with

$$f(x, y) = \frac{1 - \frac{1}{x}}{1 - \frac{1}{x^\ell}} x^y.$$

To conclude we notice that

$$\prod_{p \in \mathcal{P}(\Lambda)} \left(1 + \frac{1 - \frac{1}{p}}{1 - \frac{1}{p^\ell}} \sum_{k=1}^{\nu_p(\Lambda)} p^k\right) = \prod_{p \in \mathcal{P}(\Lambda)} \frac{p^{\nu_p(\Lambda)} - 1/p^\ell}{1 - 1/p^\ell} = \Lambda \prod_{p \in \mathcal{P}(\Lambda)} \frac{1 - 1/p^{\ell + \nu_p(\Lambda)}}{1 - 1/p^\ell}. \quad \square$$

Proof of Theorem 2.18. We start by proving that any choice of the input parameter $d \leq \bar{d}$ satisfies Constraint 2.22, thus we can apply all the previous lemmas and upper bound the failure probability of Algorithm 1 with the quantity given by Lemma 2.27. Remark that

$$2\beta \frac{2^d FG}{N} \leq 2\beta \frac{2^{\bar{d}} FG}{N} = \frac{2\beta FG}{N} \left(\frac{N}{6FG\beta} \right)^{\frac{\ell}{\ell+1}} = \left(\frac{2FG}{N} \frac{\beta}{3^\ell} \right)^{\frac{1}{\ell+1}}.$$

We already noticed when defining the $\text{SRN}_\ell(N; F, G)$ code that $2FG < N$. Thanks to Constraint 2.11, we know that $\beta < 3^\ell$, thus the above quantity is smaller than 1 and Constraint 2.22 is satisfied.

As noticed in Equation (7), $\mathbb{P}(S_{\mathbf{E}} \neq \{0\}) \leq \sum_{\varphi=1}^{\Lambda-1} \mathbb{P}(\forall i, \varphi E'_i \in \mathbb{Z}_{\Lambda, B\Lambda})$, which we can upper bound using Lemma 2.27. Thanks to the hypothesis of Theorem 2.18 we know that $\Lambda \leq 2^d$, and using $(3B)^\ell 2^d = 2^{-(\ell+1)(\bar{d}-d)}$, we have proved Theorem 2.18. \square

2.8.2. Decoding failure probability with respect to the second error model

In the second error model, we need to make a distinction between the maximal error locator Λ_m (over which there are uniform random errors) and the actual error locator Λ which is in general a divisor of Λ_m . We will denote $\mathbb{P}_{\mathcal{E}_{\Lambda_m}^2}$ (resp. $\mathbb{P}_{\mathcal{E}_\Lambda^1}$) the probability function under the error model 2 (resp. the error model 1). Let \mathcal{F} be the event of decoding failure with algorithm parameter $d \geq \log(\Lambda_m)$ i.e. the set of random matrices \mathbf{E} such that Algorithm 1 returns "decoding failure". Using the law of total probability, we have

$$\mathbb{P}_{\mathcal{E}_{\Lambda_m}^2}(\mathcal{F}) = \sum_{\Lambda|\Lambda_m} \mathbb{P}_{\mathcal{E}_{\Lambda_m}^2}(\mathcal{F} | \Lambda_{\mathbf{E}} = \Lambda) \mathbb{P}_{\mathcal{E}_{\Lambda_m}^2}(\Lambda_{\mathbf{E}} = \Lambda) \quad (8)$$

where $\Lambda_{\mathbf{E}} = \Lambda_{C, R}$ (see Definition 1.1). The conditional probabilities $\mathbb{P}_{\mathcal{E}_{\Lambda_m}^2}(\mathcal{F} | \Lambda_{\mathbf{E}} = \Lambda)$ in the sum are equal to $\mathbb{P}_{\mathcal{E}_\Lambda^1}(\mathcal{F})$, which are upper bounded within the proof of Lemma 2.27 by

$$\mathbb{P}_{\mathcal{E}_\Lambda^1}(\mathcal{F}) \leq (3B)^\ell \Lambda \prod_{p \in \mathcal{P}(\Lambda)} \left(\frac{1 - 1/p^{\ell + \nu_p(\Lambda)}}{1 - 1/p^\ell} \right). \quad (9)$$

Moreover, using again Lemma 2.16, we have

$$\mathbb{P}_{\mathcal{E}_{\Lambda_m}^2}(\Lambda_{\mathbf{E}} = \Lambda) = \frac{\#\Omega_\Lambda}{\Lambda_m^\ell} = \left(\frac{\Lambda}{\Lambda_m} \right)^\ell \prod_{p \in \mathcal{P}(\Lambda)} \left(1 - \frac{1}{p^\ell} \right). \quad (10)$$

Using these facts we can prove Theorem 2.19.

Proof of Theorem 2.19. Plug Equations (10) and (9) in Equation (8) to obtain that $\mathbb{P}_{\mathcal{E}_{\Lambda_m}^2}(\mathcal{F}) / (\frac{3B}{\Lambda_m})^\ell$ is less than or equal to

$$\sum_{\Lambda|\Lambda_m} \Lambda^{\ell+1} \prod_{p \in \mathcal{P}(\Lambda)} \left(1 - \frac{1}{p^{\ell + \nu_p(\Lambda)}} \right) = \sum_{\Lambda|\Lambda_m} \prod_{p \in \mathcal{P}(\Lambda)} p^{\nu_p(\Lambda)(\ell+1)} \left(1 - \frac{1}{p^{\ell + \nu_p(\Lambda)}} \right)$$

$$\begin{aligned}
&= \prod_{p \in \mathcal{P}(\Lambda_m)} \left[1 + \sum_{k=1}^{\nu_p(\Lambda_m)} p^{k(\ell+1)} \left(1 - \frac{1}{p^{\ell+k}} \right) \right] \\
&\leq \prod_{p \in \mathcal{P}(\Lambda_m)} \left[1 + \left(1 - \frac{1}{p^{\ell+\nu_p(\Lambda_m)}} \right) \sum_{k=1}^{\nu_p(\Lambda_m)} p^{k(\ell+1)} \right],
\end{aligned}$$

where we used again Lemma 2.26 with $f(x, y) = x^{y(\ell+1)} \left(1 - \frac{1}{x^{\ell+y}} \right)$, and in the last inequality we used that $1 - 1/p^{\ell+k} \leq 1 - 1/p^{\ell+\nu_p(\Lambda_m)}$ for every $k = 1, \dots, \nu_p(\Lambda_m)$. By computing the geometric sum inside the last product, the above is equal to

$$\begin{aligned}
&\prod_{p \in \mathcal{P}(\Lambda_m)} \left[1 + \left(1 - \frac{1}{p^{\ell+\nu_p(\Lambda_m)}} \right) \left(\frac{p^{(\ell+1)(\nu_p(\Lambda_m)+1)} - 1}{p^{\ell+1} - 1} - 1 \right) \right] \\
&= \prod_{p \in \mathcal{P}(\Lambda_m)} \left[1 + \frac{1 - 1/p^{\ell+\nu_p(\Lambda_m)}}{1 - 1/p^{\ell+1}} \left(p^{\nu_p(\Lambda_m)(\ell+1)} - 1 \right) \right].
\end{aligned}$$

Since $\nu_p(\Lambda_m) \geq 1$ we have that $1 \leq (1 - 1/p^{\ell+\nu_p(\Lambda_m)})/(1 - 1/p^{\ell+1})$ and the above product is upper bounded as:

$$\prod_{p \in \mathcal{P}(\Lambda_m)} \left[1 + \frac{1 - 1/p^{\ell+\nu_p(\Lambda_m)}}{1 - 1/p^{\ell+1}} \left(p^{\nu_p(\Lambda_m)(\ell+1)} - 1 \right) \right] \leq \Lambda_m^{\ell+1} \prod_{p \in \mathcal{P}(\Lambda_m)} \frac{1 - 1/p^{\ell+\nu_p(\Lambda_m)}}{1 - 1/p^{\ell+1}}$$

Now, thanks to the hypothesis of the theorem we know that $\Lambda_m \leq 2^d$, thus we can write

$$\begin{aligned}
\mathbb{P}_{\xi_r}^{\mathcal{E}_{\Lambda_m}^2}(\mathcal{F}) &\leq (3B)^\ell \Lambda_m \prod_{p \in \mathcal{P}(\Lambda_m)} \frac{1 - 1/p^{\ell+\nu_p(\Lambda_m)}}{1 - 1/p^{\ell+1}} \\
&\leq (3B)^\ell 2^d \prod_{p \in \mathcal{P}(\Lambda_m)} \frac{1 - 1/p^{\ell+\nu_p(\Lambda_m)}}{1 - 1/p^{\ell+1}}.
\end{aligned}$$

Using $2^{-(\ell+1)(\bar{d}-d)} = (3B)^\ell 2^d$, we have proved Theorem 2.19. \square

3. Analysis of the decoder for a hybrid error model

In this section we consider a hybrid approach to the failure probability analysis for the multiplicity rational codes studied above. The approach is hybrid in the sense that it lies in between unique decoding and interleaving.

More specifically, in the algorithm, the parameter d is chosen, and it is strictly related to the failure probability. In the analysis, d splits into two components: d_i and d_u . Essentially, d_u is bounded for fitting the unique decoding, whereas d_i can be larger as it is related to the interleaving decoding and its bound \bar{d}_i (Equation (12)) is directly proportional to the parameter ℓ . Notably,

if $d_i = 0$, the algorithm never fails. Therefore, the probability of decoding failure is strictly related to d_i and is analyzed under probabilistic assumptions, particularly considering a random error distribution.

The motivation for splitting d is that not all errors can be assumed to be purely random. For instance, in the context of distributed computation, some errors might be introduced by malicious entities that deliberately choose specific error patterns to force the algorithm to fail. In such cases, the errors captured by d_u remain independent of the error distribution and can still be corrected.

Since we are above the unique decoding radius, not all errors are decodable. Interleaving techniques can provide positive decoding results by considering error sets where most errors are decodable using probabilistic arguments. These techniques focus on fixed error positions and consider all possible errors at each position. In contrast, in a hybrid setting one can handle more general sets of errors, analyzing the set of all possible errors across certain subsets of the error positions. This approach may be of broader interest in coding theory. We first introduced this hybrid analysis in [GLLZ23]. However, we have a more specific motivation in this paper; in the forthcoming case of codes allowing bad primes (See Section 4), the only result we are able to get is when we only interleave a subset of all errors (namely evaluation errors). We remark that, as in [GLLZ23] for the rational function case and a different analysis, with the hybrid technique we are only able to interleave a specific type of errors. This suggests us that there could be a deeper obstacle preventing us to interleave the other type of errors (namely valuation errors).

On a technical level this hybrid analysis consists in studying the failure probability with respect to a specific portion of the error's distribution; allowing the errors to vary only over a subset $\xi_i \subseteq \xi$ of the error support, while the errors in the complementary set $\xi_u := \xi \setminus \xi_i$ are held fixed. Note that, in this section the above partition might seem arbitrary but, as we will see in the next Section 4 on bad primes, it is clearly described by some property of the error itself (see Definition 4.8). Here we generalize the analysis of the previous section relative to the decoding of SRN codes (Definition 2.1) by means of Algorithm 1. In this setting we decompose the distance parameter d of the algorithm as

$$d = d_i + d_u, \tag{11}$$

for some $d_i, d_u \geq 0$ bounds on the sizes of random and fixed errors respectively.

Error models. With the given distance parameter d as in Equation (11), we perform the hybrid analysis with respect to a distribution specified by a factorization of $\Lambda = \Lambda_u \Lambda_i$ with $\gcd(\Lambda_u, \Lambda_i) = 1$, Λ divides N . To specify the error model, we fix a sequence of nonzero error vectors $\epsilon_j \in \left(\mathbb{Z}/p_j^{\lambda_j}\mathbb{Z}\right)^\ell$ for every j such that $p_j \in \mathcal{P}(\Lambda_u)$, with $\nu_{p_j}(\epsilon_j) = \lambda_j - \nu_{p_j}(\Lambda)$. Then the random distribution for the hybrid error model is determined by the set of error matrices $\mathbf{E} \in \prod_{j=1}^n \left(\mathbb{Z}/p_j^{\lambda_j}\mathbb{Z}\right)^\ell$ such that the columns \mathbf{e}_j of \mathbf{E} satisfy

1. $\mathbf{e}_j = \mathbf{0}$ for all j such that $p_j \notin \mathcal{P}(\Lambda)$,

2. $\mathbf{e}_j = \boldsymbol{\epsilon}_j$ for all j such that $p_j \in \mathcal{P}(\Lambda_u)$,
3. $\nu_{p_j}(\mathbf{e}_j) = \lambda_j - \nu_{p_j}(\Lambda)$ for all j such that $p_j \in \mathcal{P}(\Lambda_i)$.

We let $\mathcal{H}_{\Lambda_i \Lambda_u, \boldsymbol{\epsilon}}^1$ be the set of error matrices specified as above.

Lemma 3.1. *If \mathbf{E} is uniformly distributed in $\mathcal{H}_{\Lambda_i \Lambda_u, \boldsymbol{\epsilon}}^1$, then the random vector $(E'_1 \bmod \Lambda_i, \dots, E'_\ell \bmod \Lambda_i)$ is uniformly distributed in the sample space Ω_{Λ_i} .*

Proof. For the duration of this proof, we will only consider indices j such that $p_j \in \mathcal{P}(\Lambda_i)$. Recall that \mathbf{e}_j is a random vector of $(\mathbb{Z}/p_j^{\lambda_j} \mathbb{Z})^\ell$ of valuation $\lambda_j - \nu_{p_j}(\Lambda)$ for all those particular j . Since $Y = p_j^{\lambda_j - \nu_{p_j}(\Lambda)} \bmod p_j^{\lambda_j}$, we get that $E'_i = E_i/Y = e_{i,j}/Y \bmod p_j^{\nu_{p_j}(\Lambda)}$. By definition of $\mathcal{H}_{\Lambda_i \Lambda_u, \boldsymbol{\epsilon}}^1$, the vector $\mathbf{e}_j/Y \in (\mathbb{Z}/p_j^{\nu_{p_j}(\Lambda)} \mathbb{Z})^\ell$ is random of valuation 0. As a consequence, we obtain that $(E'_1 \bmod \Lambda_i, \dots, E'_\ell \bmod \Lambda_i)$ is random among the vectors of $(\mathbb{Z}/\Lambda_i \mathbb{Z})^\ell$ such that $\gcd(E'_1, \dots, E'_\ell, \Lambda_i) = 1$. \square

Whereas for the hybrid version of the error model $\mathcal{E}_{\Lambda_m}^2$, we fix a maximal error locator Λ_m factorized as $\Lambda_m = \Lambda_{m,i} \Lambda_u$ with $\gcd(\Lambda_{m,i}, \Lambda_u) = 1$. We fix a sequence of nonzero error vectors $\boldsymbol{\epsilon}_j \in (\mathbb{Z}/p_j^{\lambda_j} \mathbb{Z})^\ell$ for every j such that $p_j \in \mathcal{P}(\Lambda_u)$, with $\nu_{p_j}(\boldsymbol{\epsilon}_j) = \lambda_j - \nu_{p_j}(\Lambda_m)$. Then we consider the set of error matrices $\mathbf{E} \in \prod_{j=1}^n (\mathbb{Z}/p_j^{\lambda_j} \mathbb{Z})^\ell$ such that

1. $\mathbf{e}_j = \mathbf{0}$ for all j such that $p_j \notin \mathcal{P}(\Lambda_m)$,
2. $\mathbf{e}_j = \boldsymbol{\epsilon}_j$ for all j such that $p_j \in \mathcal{P}(\Lambda_u)$,
3. $\nu_{p_j}(\mathbf{e}_j) \geq \lambda_j - \nu_{p_j}(\Lambda_m)$ for all j such that $p_j \in \mathcal{P}(\Lambda_{m,i})$.

We let $\mathcal{H}_{\Lambda_{m,i} \Lambda_u, \boldsymbol{\epsilon}}^2$ be the set of error matrices specified as above.

We notice that for a given error matrix in the distribution $\mathcal{H}_{\Lambda_{m,i} \Lambda_u, \boldsymbol{\epsilon}}^2$ the associated error locator has the form $\Lambda = \Lambda_i \Lambda_u$ for some divisor $\Lambda_i | \Lambda_{m,i}$.

Our results. We can now state our results concerning the analysis of the correctness of the decoder *w.r.t.* to a hybrid error model. Define

$$\bar{d}_i := \frac{\ell}{\ell + 1} [\log(N/2FG) - \log(3\beta) - 2d_u]. \quad (12)$$

Note that we must have $2d_u \leq \log(N/(6FG\beta))$ in order to ensure $\bar{d}_i \geq 0$.

Theorem 3.2. *Decoding Algorithm 1 on input*

1. *distance parameter $d = d_u + d_i$ for $d_u \leq \log\left(\sqrt{N/(6FG\beta)}\right)$ and $d_i \leq \bar{d}_i$,*
2. *a random received word \mathbf{R} uniformly distributed in $[\mathbf{f}/g]_N + \mathcal{H}_{\Lambda_i \Lambda_u, \boldsymbol{\epsilon}}^1$ for some code word $[\mathbf{f}/g]_N \in \text{SRN}_\ell(N; F, G)$ and error locator $\Lambda = \Lambda_i \Lambda_u$ such that $\log(\Lambda_u) \leq d_u$ and $\log(\Lambda_i) \leq d_i$,*

outputs the center code word $[\mathbf{f}/g]_N$ of the distribution with a probability of failure

$$\mathbb{P}_{fail} \leq 2^{-(\ell+1)(\bar{d}_i-d_i)} \prod_{p \in \mathcal{P}(\Lambda_i)} \left(\frac{1 - 1/p^{\ell+\nu_p(\Lambda_i)}}{1 - 1/p^\ell} \right).$$

Theorem 3.3. *Decoding Algorithm 1 on input*

1. distance parameter $d = d_u + d_i$ for $d_u \leq \log\left(\sqrt{N/(6FG\beta)}\right)$ and $d_i \leq \bar{d}_i$,
2. a random received word \mathbf{R} uniformly distributed in $[\mathbf{f}/g]_N + \mathcal{H}_{\Lambda_{m,i}\Lambda_u, \epsilon}^2$ for some code word $[\mathbf{f}/g]_N \in \text{SRN}_\ell(N; F, G)$ and error locator $\Lambda_m = \Lambda_{m,i}\Lambda_u$ such that $\log(\Lambda_u) \leq d_u$ and $\log(\Lambda_{m,i}) \leq d_i$,

outputs the center code word $[\mathbf{f}/g]_N$ of the distribution with a probability of failure

$$\mathbb{P}_{fail} \leq 2^{-(\ell+1)(\bar{d}_i-d_i)} \prod_{p \in \mathcal{P}(\Lambda_{m,i})} \left(\frac{1 - 1/p^{\ell+\nu_p(\Lambda_{m,i})}}{1 - 1/p^{\ell+1}} \right).$$

Example 3.4. Let's give a scenario that would highlight how Theorem 3.3 can be used in practice. Assume that a code is fixed such that $\log(N/(6FG\beta)) = 200$, so that $\bar{d} = 160$ when one interleaves for $\ell = 4$. Assume one wanted to make sure that the failure probability is less than a target probability of 2^{-30} , and also that 50 weighted errors can always be corrected ($d_u = 50$), for instance for protecting against a malicious entity. Then $\bar{d}_i = 80$ and one would have to choose the parameter $d = 134$ (thus $d_i = 74$) for the decoder (where we approximate the failure probability by $2^{-(\ell+1)(\bar{d}_i-d_i)}$). Then Theorem 3.3 would ensure that for any error with locator Λ_u such that $\log \Lambda_u \leq 50$ and for any random error distributed uniformly on an error locator $\Lambda_{m,i}$ such that $\log \Lambda_{m,i} \leq 74$ (with $\Lambda_{m,i}$ and Λ_u coprime), the failure probability is less than 2^{-30} .

We introduce a modified version of the set $S_{\mathbf{E}}$ defined as

$$S_{\mathbf{E}}^h := \{\varphi \in \mathbb{Z}/\Lambda_i\mathbb{Z} : \forall i, g\varphi E'_i \in \mathbb{Z}_{\Lambda_i, B\Lambda}\}$$

with $B := 2^d \beta \frac{2FG}{N} = 2^{d_i+d_u} \beta \frac{2FG}{N}$. The hybrid versions of Constraint 2.22 and Lemma 2.23 are as follows:

Constraint 3.5. The parameters of Algorithm 1 satisfy $2^{d_u} B < 1$.

Lemma 3.6. *If Constraint 3.5 is satisfied then $S_{\mathbf{E}}^h = \{0\} \Rightarrow S_{\mathbf{R}} \subseteq v_{\mathbf{C}}\mathbb{Z}$.*

Proof. Let $(\varphi, \psi_1, \dots, \psi_\ell) \in S_{\mathbf{R}}$. The proof of Lemma 2.23 shows that $g\varphi E'_i$ is equal to $\psi'_i := \frac{g\psi_i - f_i\varphi}{Y}$ modulo Λ , hence also modulo Λ_i . The same proof gives $|\psi'_i| \leq B\Lambda$. This means that $\varphi \in S_{\mathbf{E}}^h$, thus thanks to the hypothesis $S_{\mathbf{E}}^h = \{0\}$, we get $\Lambda_i|\varphi$, thus $g\varphi E'_i = \psi'_i = 0 \pmod{\Lambda_i}$. Since $\Lambda_u \leq 2^{d_u}$, this implies $|\psi'_i| \leq B\Lambda < \Lambda_i$, therefore $\psi'_i = 0$ in \mathbb{Z} . The end of the proof is identical to the one of Lemma 2.23. \square

As in Equation (7), we have $\mathbb{P}(S_{\mathbf{E}}^h \neq \{0\}) \leq \sum_{\varphi=1}^{\Lambda_i-1} \mathbb{P}(\forall i, \varphi E'_i \in \mathbb{Z}_{\Lambda_i, B\Lambda})$, which we now bound.

Lemma 3.7. *Given a random vector (E'_1, \dots, E'_ℓ) uniformly distributed in Ω_{Λ_i} , we have that*

$$\sum_{\varphi=1}^{\Lambda_i-1} \mathbb{P}(\forall i, \varphi E'_i \in \mathbb{Z}_{\Lambda_i, B\Lambda}) \leq (3B2^{d_u})^\ell \Lambda_i \prod_{p \in \mathcal{P}(\Lambda_i)} \left(\frac{1 - 1/p^{\ell + \nu_p(\Lambda_i)}}{1 - 1/p^\ell} \right).$$

Proof. As in the proof of Lemma 2.27, we can upper bound the probability $\sum_{\varphi=1}^{\Lambda_i-1} \mathbb{P}(\forall i, \varphi E'_i \in \mathbb{Z}_{\Lambda_i, B\Lambda})$ with

$$\sum_{\substack{\eta | \Lambda_i \\ \eta \leq B\Lambda}} \frac{\phi\left(\frac{\Lambda_i}{\eta}\right) \left(\frac{3B\Lambda}{\eta}\right)^\ell}{\left(\frac{\Lambda_i}{\eta}\right)^\ell \prod_{p \in \mathcal{P}\left(\frac{\Lambda_i}{\eta}\right)} (1 - 1/p^\ell)} \leq (3B\Lambda_u)^\ell \Lambda_i \prod_{p \in \mathcal{P}(\Lambda_i)} \left(\frac{1 - 1/p^{\ell + \nu_p(\Lambda_i)}}{1 - 1/p^\ell} \right).$$

Using that $\Lambda_u \leq 2^{d_u}$ we obtain our statement. \square

Proof of Theorem 3.2. As in the proof of Theorem 2.18, we start by noticing that our choice of parameters satisfy Constraint 3.5. We first notice that $d_i \leq \bar{d}_i = \ell/(\ell+1)(\log(N/2FG) - \log(3\beta) - 2d_u)$, thus

$$\begin{aligned} 2^{2d_u + d_i} \frac{2\beta FG}{N} &\leq 2^{2d_u + \bar{d}_i} \frac{2\beta FG}{N} = \left(\frac{N}{6\beta FG 2^{2d_u}} \right)^{\frac{\ell}{\ell+1}} \frac{2\beta FG 2^{2d_u}}{N} \\ &= \left(\frac{2FG 2^{2d_u}}{N} \frac{\beta}{3^\ell} \right)^{\frac{1}{\ell+1}}. \end{aligned}$$

Since $d_u \leq \log\left(\sqrt{N/2FG}\right)$, the fraction $2FG 2^{2d_u}/N$ is less than or equal to 1. Thanks to Constraint 2.11, we know that $\beta < 3^\ell$, thus the above quantity is less than 1 and Constraint 3.5 is satisfied. Thanks to Lemma 3.6 and Lemma 3.7, we can upper bound the failure probability by

$$\begin{aligned} \mathbb{P}_{fail} &\leq \mathbb{P}(S_{\mathbf{E}}^h \neq \{0\}) \leq \sum_{\varphi=1}^{\Lambda_i-1} \mathbb{P}(\forall i, \varphi E'_i \in \mathbb{Z}_{\Lambda_i, B\Lambda}) \\ &\leq (3B2^{d_u})^\ell \Lambda_i \prod_{p \in \mathcal{P}(\Lambda_i)} \left(\frac{1 - 1/p^{\ell + \nu_p(\Lambda_i)}}{1 - 1/p^\ell} \right). \end{aligned}$$

Since $\Lambda_i \leq 2^{d_i}$, we have $(3B2^{d_u})^\ell \Lambda_i \leq (3B)^\ell 2^{\ell d_u} 2^{d_i} = 2^{-(\ell+1)(\bar{d}_i - d_i)}$. \square

Proof of Theorem 3.3. Let \mathcal{F} be the event of decoding failure, i.e. the set of random matrices \mathbf{E} such that Algorithm 1 returns "decoding failure" with input parameter $d = d_i + d_u$ as in the statement of Theorem 3.3. We will denote $\mathbb{P}_{\mathcal{H}_{\Lambda_m, i\Lambda_u, \epsilon}^2}$ (resp. $\mathbb{P}_{\mathcal{H}_{\Lambda_i, \Lambda_u, \epsilon}^1}$) the probability function under the hybrid error

model 2 (resp. model 1) specified by a given factorization of the error locator, and by a sequence of fixed error vectors ϵ_j for every j such that $p_j \in \mathcal{P}(\Lambda_u)$.

Using the law of total probability, we have that $\mathbb{P}_{\mathcal{H}_{\Lambda_{m,i}\Lambda_u,\epsilon}^2}(\mathcal{F})$ can be decomposed as the sum

$$\mathbb{P}_{\mathcal{H}_{\Lambda_{m,i}\Lambda_u,\epsilon}^2}(\mathcal{F}) = \sum_{\Lambda_i|\Lambda_{m,i}} \mathbb{P}_{\mathcal{H}_{\Lambda_{m,i}\Lambda_u,\epsilon}^2}(\mathcal{F} \mid \Lambda_{\mathbf{E}} = \Lambda_i\Lambda_u) \mathbb{P}_{\mathcal{H}_{\Lambda_{m,i}\Lambda_u,\epsilon}^2}(\Lambda_{\mathbf{E}} = \Lambda_i\Lambda_u),$$

where $\mathbb{P}_{\mathcal{H}_{\Lambda_{m,i}\Lambda_u,\epsilon}^2}(\mathcal{F} \mid \Lambda_{\mathbf{E}} = \Lambda_i\Lambda_u) = \mathbb{P}_{\mathcal{H}_{\Lambda_i\Lambda_u,\epsilon}^1}(\mathcal{F})$, whereas

$$\mathbb{P}_{\mathcal{H}_{\Lambda_{m,i}\Lambda_u,\epsilon}^2}(\Lambda_{\mathbf{E}} = \Lambda_i\Lambda_u) = \left(\frac{\Lambda_i}{\Lambda_{m,i}}\right)^\ell \prod_{p \in \mathcal{P}(\Lambda_i)} \left(1 - \frac{1}{p^\ell}\right)$$

as in Equation (10).

Plugging the above two expressions in the decomposition from the law of total probability, similarly as done in the proof of Theorem 2.19, we can upper bound $\mathbb{P}_{\mathcal{H}_{\Lambda_{m,i}\Lambda_u,\epsilon}^2}(\mathcal{F}) / \left(\frac{2^{d_u}3B}{\Lambda_{m,i}}\right)^\ell$ by

$$\sum_{\Lambda_i|\Lambda_{m,i}} \Lambda_i^{\ell+1} \prod_{p \in \mathcal{P}(\Lambda_i)} \left(1 - \frac{1}{p^{\ell+\nu_p(\Lambda_i)}}\right) \leq \Lambda_{m,i}^{\ell+1} \prod_{p \in \mathcal{P}(\Lambda_{m,i})} \frac{1 - 1/p^{\ell+\nu_p(\Lambda_{m,i})}}{1 - 1/p^{\ell+1}}.$$

Thus,

$$\begin{aligned} \mathbb{P}_{\mathcal{H}_{\Lambda_{m,i}\Lambda_u,\epsilon}^2}(\mathcal{F}) &\leq (2^{d_u}3B)^\ell \Lambda_{m,i} \prod_{p \in \mathcal{P}(\Lambda_{m,i})} \frac{1 - 1/p^{\ell+\nu_p(\Lambda_{m,i})}}{1 - 1/p^{\ell+1}} \\ &\leq (2^{d_u}3B)^\ell 2^{d_i} \prod_{p \in \mathcal{P}(\Lambda_{m,i})} \frac{1 - 1/p^{\ell+\nu_p(\Lambda_{m,i})}}{1 - 1/p^{\ell+1}} \end{aligned}$$

and we conclude by using that $(2^{d_u}3B)^\ell 2^{d_i} = 2^{-(\ell+1)(\bar{d}_i - d_i)}$. \square

4. The case of bad primes

In this section we use the hybrid analysis technique presented above to extend our study of the decoding failure in a context where the hypothesis $\gcd(g, N) = 1$ of Definition 2.1 does not hold, thus some reductions in the encoding of \mathbf{f}/g may not be defined. Primes relative to undefined reductions are called *bad primes*. Our theorems (4.16 and 4.17) are the first results (in the bad primes' scenario) relative to the decoding beyond uniqueness for rational number codes.

In the case of simultaneous rational function reconstruction (over $\mathbb{F}_q[x]$), instead of the primes p_1, \dots, p_n , distinct evaluation points $\alpha_1, \dots, \alpha_n \in \mathbb{F}_q$ are chosen, so in the polynomial context the notion of bad primes correspond to

poles of the vector of rational functions \mathbf{f}/g , i.e. roots of the denominator g . We find two approaches in the literature to deal with poles: in [KPY20] an extra symbol ∞ is used, while in [GLLZ23] coordinates are given by shifted Laurent series representations of the fractions.

In particular, considering the rational function case, the authors of [GLLZ23] introduced the following multi-precision encoding composed of a valuation part and a reduction part, which we give in the rational number case, where the shifted Laurent series is replaced by a shifted p -adic expansion of the vector of fractions \mathbf{f}/g :

Definition 4.1 (Multi-precision encoding). Given a sequence of multiplicities $\lambda_1, \dots, \lambda_n$ associated to the primes $p_1, \dots, p_n \in \mathbb{Z}$, and a reduced vector of fractions $\mathbf{f}/g \in \mathbb{Q}^\ell$, we define its multi-precision encoding to be the sequence of couples $\text{Ev}^\infty(\mathbf{f}/g) := (\nu_{p_j}(g), \mathcal{S}_j(\mathbf{f}/g))_{1 \leq j \leq n}$ such that

$$\mathcal{S}_j(\mathbf{f}/g) := \mathbf{f} / \left(g/p_j^{\nu_{p_j}(g)} \right) \bmod p_j^{\lambda_j - \nu_{p_j}(g)}.$$

By convention, we set $\mathcal{S}_j(\mathbf{f}/g) = \mathbf{1}$ when $\nu_{p_j}(g) = \lambda_j$.

Here we prove, under the hypothesis $N \geq 2FG$, the injectivity of the above encoding.

Proposition 4.2. *Let $\mathbf{f}/g, \mathbf{f}'/g' \in \mathbb{Q}^\ell$ with $\|\mathbf{f}\|_\infty < F$, $0 < g < G$ such that $\text{Ev}^\infty(\mathbf{f}/g) = \text{Ev}^\infty(\mathbf{f}'/g')$. If we assume that $N \geq 2FG$, the equality $\mathbf{f}/g = \mathbf{f}'/g'$ holds.*

Proof. For every $j = 1, \dots, n$ we let $v_j := \nu_{p_j}(g) = \nu_{p_j}(g')$. By hypothesis $\mathcal{S}_j(\mathbf{f}/g) = \mathcal{S}_j(\mathbf{f}'/g')$, i.e.

$$\mathbf{f} / (g/p_j^{v_j}) = \mathbf{f}' / (g'/p_j^{v_j}) \bmod p_j^{\lambda_j - v_j} \iff \mathbf{f}g' / p_j^{v_j} = \mathbf{f}'g / p_j^{v_j} \bmod p_j^{\lambda_j - v_j}.$$

In other words $\mathbf{f}g' = \mathbf{f}'g \bmod p_j^{\lambda_j}$, which implies that $\mathbf{f}g' = \mathbf{f}'g \bmod N$. Since by hypothesis $\|\mathbf{f}g' - \mathbf{f}'g\|_\infty < 2FG \leq N$, we conclude that $\mathbf{f}g' = \mathbf{f}'g$ in \mathbb{Q}^ℓ . \square

Under the hypothesis $2FG \leq N$, we can then introduce the *simultaneous rational number code with bad primes* as the set

$$\text{SRN}_\ell^\infty(N; F, G) := \left\{ \text{Ev}^\infty \left(\frac{\mathbf{f}}{g} \right) : \begin{array}{l} \|\mathbf{f}\|_\infty < F, \quad 0 < g < G, \\ \gcd(f_1, \dots, f_\ell, g) = 1 \end{array} \right\}.$$

We will refer to it as the SRN code with bad primes.

Being composed of two parts, codewords $\text{Ev}^\infty(\mathbf{f}/g)$ can be affected by two kinds of errors (valuation and evaluation errors). Here we adapt the hybrid analysis of Section 3, with the factorization of the error locator $\Lambda = \Lambda_i \Lambda_u$ reflecting these two types of errors (see Definition 4.8).

Definition 4.3. Let the *ambient space of received words* be the quotient

$$\mathbb{S}_\lambda^\ell := \left(\prod_{j=1}^n [0, \lambda_j] \times \left(\mathbb{Z}/p_j^{\lambda_j} \mathbb{Z} \right)^\ell \right) / \sim$$

where \sim is the equivalence relation for which $(v_j, \mathbf{r}_j)_{1 \leq j \leq n} \sim (v'_j, \mathbf{r}'_j)_{1 \leq j \leq n}$ if and only if for every $j = 1, \dots, n$, $p_j^{v'_j} \mathbf{r}'_j = p_j^{v_j} \mathbf{r}_j \pmod{p_j^{\lambda_j}}$. We say that a representative $(v_j, \mathbf{r}_j)_{1 \leq j \leq n}$ is *reduced* if $\gcd(\mathbf{r}_j, p_j^{v_j}) = 1$ for every $j = 1, \dots, n$. Define $R_i := \text{CRT}_N(r_{i,1}, \dots, r_{i,n})$ for every $i = 1, \dots, \ell$.

In what follows we can always assume that the received word $(v_j, \mathbf{r}_j)_{1 \leq j \leq n}$ is reduced, thanks to the following proposition:

Proposition 4.4. *Any equivalence class contains a reduced representative.*

Proof. Given any received word $(v_j, \mathbf{r}_j)_{1 \leq j \leq n}$, for every $j = 1, \dots, n$ we let $p_j^{\eta_j} := \gcd(\mathbf{r}_j, p_j^{v_j})$. Then $(v_j, \mathbf{r}_j)_{1 \leq j \leq n} \sim \left(v_j - \eta_j, \mathbf{r}_j^{\lambda_j} / p_j^{\eta_j} \pmod{p_j^{\lambda_j}} \right)_{1 \leq j \leq n}$, with the representative on the right-hand side clearly reduced by the definition of $p_j^{\eta_j}$. \square

In the ambient space \mathbb{S}_λ^ℓ we identify received words which represent the same reduced vector of fractions in the sense that, by definition

- $(v_j, \mathbf{r}_j)_{1 \leq j \leq n} \sim (v_j, \mathbf{r}'_j)_{1 \leq j \leq n} \Leftrightarrow \mathbf{r}_j = \mathbf{r}'_j \pmod{p_j^{\lambda_j - v_j}}$.
- Given a received valuation $0 \leq v_j \leq \lambda_j$ then for every $1 \leq \delta_j \leq \lambda_j - v_j$

$$(v_j, \mathbf{r}_j)_{1 \leq j \leq n} \sim (v_j + \delta_j, p_j^{\delta_j} \mathbf{r}_j)_{1 \leq j \leq n}.$$

Remark 4.5. Thanks to the first of the above two points we can map the evaluation of a reduced vector of rationals $\text{Ev}^\infty(\mathbf{f}/g)$ into the space of received words.

Definition 4.6. Given two elements $\mathbf{R}_1 := (v_j, \mathbf{r}_j)_{1 \leq j \leq n}$, $\mathbf{R}_2 := (v'_j, \mathbf{r}'_j)_{1 \leq j \leq n}$ in \mathbb{S}_λ^ℓ , we define the columns \mathbf{e}_j of the relative error matrix $\mathbf{E}_{\mathbf{R}_1, \mathbf{R}_2}$ as

$$\mathbf{e}_j := p_j^{v_j} \mathbf{r}'_j - p_j^{v'_j} \mathbf{r}_j \pmod{p_j^{\lambda_j}}.$$

We let the relative error and truth locator be

$$\Lambda_{\mathbf{R}_1, \mathbf{R}_2} := \prod_{j=1}^n p_j^{\lambda_j - \nu_{p_j}(\mathbf{e}_j)}, \quad Y_{\mathbf{R}_1, \mathbf{R}_2} := \prod_{j=1}^n p_j^{\nu_{p_j}(\mathbf{e}_j)}$$

respectively, and the relative distance $d(\mathbf{R}_1, \mathbf{R}_2) := \log(\Lambda_{\mathbf{R}_1, \mathbf{R}_2})$.

Remark 4.7. Unlike the errors considered in Sections 2 and 3, in this case the usual relation $\mathbf{R}_1 = \mathbf{R}_2 + \mathbf{E}$ does not hold. For this reason the error models (see Subsection 4.4) will be defined directly by distributions in the space of received words \mathbb{S}_λ^ℓ .

In spite of the above remark, we note the consistency of the error \mathbf{e}_j with the equivalence relation \sim , indeed by definition

$$\mathbf{e}_j = \mathbf{0} \bmod p_j^{\lambda_j} \quad \forall j = 1, \dots, n \Leftrightarrow (\mathbf{v}_j, \mathbf{r}_j)_{1 \leq j \leq n} \sim (\mathbf{v}'_j, \mathbf{r}'_j)_{1 \leq j \leq n}.$$

Due to the properties of \sim , we can partition the set of error positions into valuation and evaluation errors.

Definition 4.8. Given two evaluations $(\mathbf{v}_j, \mathbf{r}_j)_{1 \leq j \leq n}, (\mathbf{v}'_j, \mathbf{r}'_j)_{1 \leq j \leq n} \in \mathbb{S}_\lambda^\ell$ satisfying $\gcd(p_j^{\mathbf{v}_j}, \mathbf{r}_j) = 1$, we divide the error support

$$\xi = \{j \mid p_j^{\mathbf{v}_j} \mathbf{r}'_j \neq p_j^{\mathbf{v}'_j} \mathbf{r}_j \bmod p_j^{\lambda_j}\} = \{j \mid (\mathbf{v}_j, \mathbf{r}_j) \not\sim (\mathbf{v}'_j, \mathbf{r}'_j)\}$$

into the *valuation errors*

$$\xi_v := \{j \mid \mathbf{v}_j \neq \mathbf{v}'_j\}$$

and the remaining *evaluation errors*

$$\xi_e = \{j \mid (\mathbf{v}_j = \mathbf{v}'_j) \text{ and } (\mathbf{r}_j \neq \mathbf{r}'_j \bmod p_j^{\lambda_j - \mathbf{v}_j})\}.$$

We provide an equivalent, yet more practical, representation of the errors.

Remark 4.9. Given a codeword $(\nu_{p_j}(g), \mathcal{S}_j(\mathbf{f}/g))_{1 \leq j \leq n}$ (as in Definition 4.1) and a received word $(\mathbf{v}_j, \mathbf{r}_j)_{1 \leq j \leq n} \in \mathbb{S}_\lambda^\ell$, the sequence of error vectors $(\mathbf{e}_j)_{1 \leq j \leq n}$ is given by

$$\mathbf{e}_j = p_j^{\mathbf{v}_j} \mathcal{S}_j(\mathbf{f}/g) - p_j^{\nu_{p_j}(g)} \mathbf{r}_j \bmod p_j^{\lambda_j}.$$

Multiplying the above by the invertible element $g/p_j^{\nu_{p_j}(g)}$, we obtain that up to invertible transformations of the error sequence components (leaving the distance unchanged), we can equivalently view the sequence of error vectors as given by

$$\tilde{\mathbf{e}}_j := \frac{g}{p_j^{\nu_{p_j}(g)}} \mathbf{e}_j = p_j^{\mathbf{v}_j} \mathbf{f} - g \mathbf{r}_j \bmod p_j^{\lambda_j}.$$

Study of potential errors and received words around a fixed codeword. Due to Remark 4.7, we need to study what kind of errors and received words we can obtain around a fixed vector of fractions \mathbf{f}/g , in particular with respect to the distinction between valuation and evaluation errors. Regarding the error positions as long as $\xi_e, \xi_v \subset \{1, \dots, n\}$ and $\xi_e \cap \xi_v = \emptyset$ we have no constraints: all valuation (resp. evaluation) error supports ξ_v (resp. ξ_e) are attained. Once the error positions have been fixed and partitioned as $\xi_v \cup \xi_e$, the valuations of the error vectors need to satisfy $\mu_j = \nu_{p_j}(\mathbf{e}_j) = \lambda_j$ for every position j which is not erroneous, *i.e.* $\forall j \notin \xi_e \cup \xi_v$. Let us examine what can happen in the evaluation and valuation error cases respectively:

- If $j \in \xi_e$, we have an evaluation error, thus any received word \mathbf{R} must satisfy $\mathbf{v}_j = \nu_{p_j}(g)$, furthermore we must have that the valuation of any error vector \mathbf{e}_j must satisfy $\mu_j = \nu_{p_j}(p_j^{\mathbf{v}_j} \mathbf{f} - g \mathbf{r}_j) \geq \nu_{p_j}(g)$ thus, dividing by $p_j^{\mathbf{v}_j}$, we have that $\mathcal{S}_j(\mathbf{f}/g) - \mathbf{r}_j$ can be any element of valuation $\mu_j - \nu_{p_j}(g)$.

- If $j \in \xi_v$, we have a valuation error, thus for every received word we have either
 1. $v_j < \nu_{p_j}(g)$: in this case the valuation of the error vector and the received word must coincide, *i.e.* $\mu_j = v_j$, and from the definition of $\tilde{\mathbf{e}}_j$ we must have that $\tilde{\mathbf{e}}_j = p_j^{\mu_j} \mathbf{f} \pmod{p_j^{\nu_{p_j}(g)}}$, regardless of the reduction part \mathbf{r}_j . Thus, in this case we do not have any constraints on \mathbf{r}_j .
 2. $v_j > \nu_{p_j}(g)$: in this case the valuation of the error vector must coincide with the valuation of g , *i.e.* $\mu_j = \nu_{p_j}(g)$. Besides this valuation constraint, the error vectors can take any value, as well as the received reductions \mathbf{r}_j .

Minimal distance. Similarly to Lemma 2.4, we can prove that the Minimal distance of SRN codes with bad primes satisfies the following

Lemma 4.10. *We have $d(\text{SRN}_\ell^\infty(N; F, G)) > \log_2\left(\frac{N}{2FG}\right)$.*

Proof. Let $\mathbf{C}_1 = (\nu_{p_j}(g), \mathcal{S}_j(\mathbf{f}/g))_{1 \leq j \leq n}$, $\mathbf{C}_2 = (\nu_{p_j}(g'), \mathcal{S}_j(\mathbf{f}'/g'))_{1 \leq j \leq n}$ be two distinct codewords. From

$$\mathbf{e}_j = p_j^{\nu_{p_j}(g)} \left(\frac{\mathbf{f}'}{g'/p_j^{\nu_{p_j}(g')}} \right) - p_j^{\nu_{p_j}(g')} \left(\frac{\mathbf{f}}{g/p_j^{\nu_{p_j}(g)}} \right) \pmod{p_j^{\lambda_j}},$$

we see that

$$\frac{g}{p_j^{\nu_{p_j}(g)}} \frac{g'}{p_j^{\nu_{p_j}(g')}} \mathbf{e}_j = \mathbf{f}'g - \mathbf{f}g' \pmod{p_j^{\lambda_j}}.$$

Using $\Lambda \mathbf{e}_j = 0 \pmod{p_j^{\lambda_j}}$ for all j , we obtain

$$\forall 1 \leq j \leq n, \quad 0 = \Lambda \frac{g}{p_j^{\nu_{p_j}(g)}} \frac{g'}{p_j^{\nu_{p_j}(g')}} \mathbf{e}_j = \Lambda(\mathbf{f}'g - \mathbf{f}g') \pmod{p_j^{\lambda_j}}.$$

Therefore, N divides $\Lambda(\mathbf{f}'g - \mathbf{f}g')$, so $Y = N/\Lambda$ divides $(\mathbf{f}'g - \mathbf{f}g')$. By the injectivity of the evaluation, $\mathbf{C}_1 \neq \mathbf{C}_2$ involves $\mathbf{f}'g - \mathbf{f}g' \neq \mathbf{0}$, which implies that $Y \leq \|\mathbf{f}'g - \mathbf{f}g'\|_\infty < 2FG$. Hence, for all codewords $\mathbf{C}_1 \neq \mathbf{C}_2$, we bound $d(\mathbf{C}_1, \mathbf{C}_2) = \log(\Lambda) = \log(N/Y) > \log(N/2FG)$. \square

4.1. Key equations

The decoding of SRN codes with bad primes, as in Section 2, is based on a basis reduction over a lattice describing the solution set to some key equations. Thanks to Remark 4.9 and the definition of Λ , we have that $\Lambda \mathbf{e}_j = 0 \pmod{p_j^{\lambda_j}}$, and so $0 = \Lambda \tilde{\mathbf{e}}_j = p_j^{\nu_j} \Lambda \mathbf{f} - \Lambda g \mathbf{r}_j \pmod{p_j^{\lambda_j}}$. We observe that for every couple of received word $(v_j, \mathbf{r}_j)_{1 \leq j \leq n}$ and reduced vector of fractions \mathbf{f}/g we have that the equation $\text{CRT}_N(p_j^{\nu_j}) \Lambda f_i = \Lambda g R_i \pmod{N}$ holds for every $i = 1, \dots, \ell$. By

defining the new variables $\varphi := \Lambda g$, $\boldsymbol{\psi} = \Lambda \mathbf{f}$ we get the *key equations* in presence of bad primes:

$$\forall i = 1, \dots, \ell, \quad \text{CRT}_N(p_j^{v_j}) \psi_i = \varphi R_i \pmod{N}. \quad (13)$$

For some distance parameter d , we let the set of solutions be

$$S_{\mathbf{R}, 2^d} := \left\{ (\varphi, \boldsymbol{\psi}) \in \mathbb{Z}^{\ell+1} : \begin{array}{l} \text{CRT}_N(p_j^{v_j}) \psi_i = \varphi R_i \pmod{N}, \quad \forall i \\ 0 < \varphi < 2^d G, \quad \|\boldsymbol{\psi}\|_\infty < 2^d F \end{array} \right\}.$$

If $\Lambda \leq 2^d$ we see that $v_C := (\Lambda g, \Lambda \mathbf{f}) \in S_{\mathbf{R}, 2^d}$.

Reduced key equations. It is possible to give an equivalent description of the solutions in $S_{\mathbf{R}, 2^d}$, whose size constraints are smaller. Letting $N_\infty := \prod_{j=1}^n p_j^{v_j}$ we note that, thanks to Equation (13), $N_\infty | \varphi$ since $N_\infty | \text{CRT}_N(p_j^{v_j})$, $N_\infty | N$ and by hypothesis $\gcd(N_\infty, R_i) = 1$ as received words are assumed to be reduced. Thus, we can rewrite Equation (13) in the following form, which we call *reduced key equations*

$$\forall i = 1, \dots, \ell, \quad \begin{array}{l} \psi_i = \varphi' R'_i \pmod{\frac{N}{N_\infty}} \\ 0 < \varphi' < 2^d G / N_\infty \text{ and } \|\boldsymbol{\psi}\|_\infty < 2^d F \end{array} \quad (14)$$

where $\varphi' := \varphi / N_\infty$ and $R'_i := R_i \text{CRT}_{N/N_\infty} \left(\frac{N}{p_j^{v_j}} \right)$.

4.2. Decoding SRN_ℓ^∞ codes

In this section we give our decoding algorithm for SRN codes with bad primes, which is a modification of Algorithm 1.

As in Section 2, the decoding is based on the computation of a short vector $v_s = (\varphi, \boldsymbol{\psi})$ solution of Equations (13). Given the lattice \mathcal{L}_∞ spanned by the rows of the matrix

$$\mathcal{L}_\infty = \text{Span} \left(\begin{array}{cccc} N_\infty & R'_1 & \cdots & R'_\ell \\ 0 & N/N_\infty & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & N/N_\infty \end{array} \right), \quad (15)$$

we note that all solutions of Equation (13) are spanned by the rows of \mathcal{L}_∞ . Indeed, given a solution $(\varphi, \boldsymbol{\psi})$, thanks to Equation (14), we know that $\varphi' = \varphi / N_\infty$ is an integer, and that for $i = 1, \dots, n$ the $(i+1)$ -th entry $\psi_i - \varphi' R'_i$ of the difference $(\varphi, \boldsymbol{\psi}) - \varphi' (N_\infty, R'_1, \dots, R'_\ell)$ is zero modulo $\frac{N}{N_\infty}$.

Also in this case we introduce a scaling operator $\sigma_{F,G} : \mathbb{Q}^{\ell+1} \rightarrow \mathbb{Q}^{\ell+1}$ such that $\sigma_{F,G}((v_0, v_1, \dots, v_\ell)) := (v_0 F, v_1 G, \dots, v_\ell G)$, to match the size constraints of $S_{\mathbf{R}, 2^d}$ with the $\|\cdot\|_\infty$ -norm of its elements. This scaling will transform \mathcal{L}_∞ into the scaled lattice $\bar{\mathcal{L}}_\infty := \sigma_{F,G}(\mathcal{L}_\infty)$, and our solution set $S_{\mathbf{R}, 2^d}$ into

$$S'_{\mathbf{R}, 2^d} := \sigma_{F,G}(S_{\mathbf{R}, 2^d}) = \{(\varphi, \psi_1, \dots, \psi_\ell) \in \bar{\mathcal{L}}_\infty : 0 < \varphi < 2^d F G, \|\boldsymbol{\psi}\|_\infty < 2^d F G\}.$$

In the decoding algorithm we compute an element $v_s \in S_{\mathbf{R}, 2^d}$ by computing a scaled short vector $\bar{v}_s := \mathcal{ASVP}_\infty(\bar{\mathcal{L}}_\infty)$, and unscaling it $v_s := \sigma_{F,G}^{-1}(\bar{v}_s)$.

Due to the approximation factor β of the sub-routine \mathcal{ASVP}_∞ , assuming Constraint 4.11, the solution v_s belongs to a larger set.

Constraint 4.11. Given the received word $(v_j, \mathbf{r}_j)_{1 \leq j \leq n}$, there exists a code word $\text{Ev}^\infty(\mathbf{f}/g)$ such that the corresponding error locator satisfies $\Lambda \leq 2^d$.

Lemma 4.12. *Assuming Constraint 4.11, we have that $v_s \in S_{\mathbf{R}} := S_{\mathbf{R}, 2^d \beta}$.*

Proof. We know that $\|\bar{v}_s\|_\infty \leq \beta \lambda_\infty(\bar{\mathcal{L}}_\infty) \leq \beta \|\sigma_{F,G}(v_C)\|_\infty < \beta \Lambda F G \leq \beta 2^d F G$. Since we assumed that $(\bar{v}_s)_0 \geq 0$, we have $\bar{v}_s \in S'_{\mathbf{R}, 2^d \beta}$ and $v_s \in S_{\mathbf{R}, 2^d \beta}$. \square

We notice that assuming Constraint 4.11 we also have $v_C \in S_{\mathbf{R}}$. We are ready to introduce our decoding algorithm for SRN codes with bad primes.

Algorithm 2: SRN_ℓ^∞ codes decoder.

Input: $\text{SRN}_\ell^\infty(N; F, G)$, received word $\mathbf{R} := (v_j, \mathbf{r}_j)_{1 \leq j \leq n}$, distance bound d

Output: A reduced vector of fractions $\boldsymbol{\psi}'/\varphi'$ s.t. $d(\text{Ev}^\infty(\boldsymbol{\psi}'/\varphi'), \mathbf{R}) \leq d$ or “decoding failure”

- 1 Let $\bar{\mathcal{L}}_p := \sigma_{F,G}(\mathcal{L}_\infty)$ be the scaled lattice of \mathcal{L}_∞ defined in Equation (15)
 - 2 Compute a short vector $\bar{v}_s := \mathcal{ASVP}_\infty(\bar{\mathcal{L}}_\infty)$
 - 3 Unscale the vector: $v_s = (\varphi, \psi_1, \dots, \psi_\ell) := \sigma_{F,G}^{-1}(\bar{v}_s)$
 - 4 Let $\eta := \gcd(\varphi, \psi_1, \dots, \psi_\ell)$, $\varphi' := \varphi/\eta$ and $\forall i, \psi'_i := \psi_i/\eta$
 - 5 **if** $\eta \leq 2^d$, $|\varphi'| < G$ and $\forall i, |\psi'_i| < F$ **then**
 - 6 **return** $(\psi'_1/\varphi', \dots, \psi'_\ell/\varphi')$
 - 7 **else return** "decoding failure";
-

Lemma 4.13. *If Algorithm 2 returns $\boldsymbol{\psi}'/\varphi'$ on input \mathbf{R} and parameter d , then $\boldsymbol{\psi}'/\varphi'$ is associated to a code word of $\text{SRN}_\ell^\infty(N; F, G)$ close to \mathbf{R} , i.e. it is a reduced vector of fractions with $\|\boldsymbol{\psi}'\|_\infty < F$, $0 < \varphi' < G$ and $d(\text{Ev}^\infty(\boldsymbol{\psi}'/\varphi'), \mathbf{R}) \leq d$.*

Proof. The output vector $\boldsymbol{\psi}/\varphi$ is associated to a code word of $\text{SRN}_\ell^\infty(N; F, G)$ since the algorithm has verified the size conditions $|\varphi'| < G$, $|\psi'_i| < F$ for all i . Now, we use that $(\varphi, \boldsymbol{\psi}) = (\eta\varphi', \eta\boldsymbol{\psi}')$ is in the lattice \mathcal{L}_∞ , so that $\eta(\text{CRT}_N(p_j^{v_j})\boldsymbol{\psi}' - \varphi' R_i) = 0 \pmod N$ for all i , which implies that $\nu_{p_j}(\eta) \geq \lambda_j - \mu_j = \nu_{p_j}(\Lambda)$ with Λ being the error locator between $\text{Ev}^\infty(\boldsymbol{\psi}/\varphi)$ and the input \mathbf{R} . Thus, $\Lambda|\eta| \leq 2^d$, and we can conclude that $d(\text{Ev}^\infty(\boldsymbol{\psi}'/\varphi'), \mathbf{R}) = \log \Lambda \leq \log \eta \leq d$. \square

4.3. Unique decoding

As pointed out in Remark 2.15, it is not because of the approximation factor β that Algorithm 2 might fail, but because we are decoding with a distance

parameter $d > \log(\sqrt{N/2FG})$. Thus, at the cost of using an exact SVP solver, *i.e.* a subroutine \mathcal{ASVP}_∞ returning the shortest vector of $\tilde{\mathcal{L}}_\infty$, we can assume $\beta = 1$. The drawback of exact SVP solvers is that their complexity is exponential in the dimension of the lattice, nevertheless in our context can be reasonable to employ an exact SVP solver to compute v_s , as the dimension $\ell + 1$ is fixed and can be assumed to be relatively small. For this reason we prove the unique decoding of SRN codes with bad primes by means of Algorithm 2 with $\beta = 1$ (thus computing an element in $S_{\mathbf{R}, 2^d}$) and when the distance parameter d is below unique decoding capacity.

Proposition 4.14. *If $d(\text{Ev}^\infty(\mathbf{f}/g), (v_j, \mathbf{r}_j)_{1 \leq j \leq n}) \leq d \leq \log_2(\sqrt{N/2FG})$, then $S_{\mathbf{R}, 2^d} \subset v_{\mathbf{C}}\mathbb{Z}$.*

Proof. By hypothesis $d(\text{Ev}^\infty(\mathbf{f}/g), (v_j, \mathbf{r}_j)_{1 \leq j \leq n}) \leq d$, we have $v_{\mathbf{C}} = (\Lambda g, \Lambda \mathbf{f}) \in S_{\mathbf{R}, 2^d}$. Let $(\varphi, \psi) \in S_{\mathbf{R}, 2^d}$ be another solution of the key equations. We have that

$$\begin{cases} p_j^{v_j} \Lambda \mathbf{f} = \mathbf{r}_j \Lambda g & \text{mod } p_j^{\lambda_j} \\ p_j^{v_j} \psi = \mathbf{r}_j \varphi & \text{mod } p_j^{\lambda_j} \end{cases}$$

for some $\Lambda \in \mathbb{Z}$ with $\log_2(\Lambda) \leq d \leq \log_2(\sqrt{N/2FG})$. Since the received word $(v_j, \mathbf{r}_j)_{1 \leq j \leq n}$ is assumed to be reduced, *i.e.* $\gcd(p_j^{v_j}, \mathbf{r}_j) = 1$, from the above we get that $p_j^{v_j} | \Lambda g$ and $p_j^{v_j} | \varphi$ for every $j = 1, \dots, n$. Thus, when multiplying the first equation by φ and the second one by Λg we get

$$\begin{cases} p_j^{v_j} \Lambda \varphi \mathbf{f} = \mathbf{r}_j \Lambda g \varphi & \text{mod } p_j^{\lambda_j + v_j} \\ p_j^{v_j} \Lambda g \psi = \mathbf{r}_j \Lambda g \varphi & \text{mod } p_j^{\lambda_j + v_j} \end{cases}$$

and subtracting one another, and dividing by $p_j^{v_j}$, we obtain $\Lambda(\varphi \mathbf{f} - g\psi) = 0 \text{ mod } N$. By hypothesis, we have that $\Lambda \|\varphi \mathbf{f} - g\psi\|_\infty < 2^d(2FG2^d) \leq N$ which implies that $\Lambda(\varphi \mathbf{f} - g\psi) = 0$ thus $\varphi \mathbf{f} = g\psi$ in \mathbb{Z}^ℓ .

Since \mathbf{f}/g is a reduced vector of fractions, there exists $a \in \mathbb{Z}$ such that $(\varphi, \psi) = a(g, \mathbf{f})$. Substituting in the key equations for (φ, ψ) , we get $a(p_j^{v_j} \mathbf{f} - \mathbf{r}_j g) = 0 \text{ mod } p_j^{\lambda_j}$. However, Λ divides a by definition of Λ , so $(\varphi, \psi) \in v_{\mathbf{C}}\mathbb{Z}$. \square

4.4. Hybrid Error Models for Bad Primes

In this subsection we adapt the hybrid error models of the previous section to the case with bad primes. Recall that the hybrid error model is composed of both fixed errors and random errors. As done in [GLLZ23], here we consider a hybrid error model where valuation errors are fixed, while evaluation errors are random. In previous Sections 2 and 3, the error models were defined on the error matrices \mathbf{E} , then the theorems applied to received words \mathbf{R} such that $\mathbf{R} = \mathbf{C} + \mathbf{E}$. In this section, as pointed out in Remark 4.7, we have a more complicated relation between \mathbf{R} , \mathbf{C} and \mathbf{E} . So we are going to define the error model directly on \mathbf{R} .

Our error model needs to fix the following parameters:

- a reduced vector of rationals $\mathbf{f}/g \in \mathbb{Q}^\ell$ such that $\|\mathbf{f}\|_\infty < F$, $0 < g < G$,
- valuation ξ_v and evaluation ξ_e error supports such that $\xi_e, \xi_v \subset \{1, \dots, n\}$ and $\xi_e \cap \xi_v = \emptyset$,
- error valuations $(\mu_j)_{1 \leq j \leq n}$ such that
 - $\mu_j = \lambda_j$ for $j \notin \xi_e \cup \xi_v$,
 - $\mu_j \geq \nu_{p_j}(g)$ and $\mu_j < \lambda_j$ for $j \in \xi_e$,
 - $\mu_j \leq \nu_{p_j}(g)$ and $\mu_j < \lambda_j$ for $j \in \xi_v$,
- a partial received word $\mathfrak{R}_j = (v_j, \mathbf{r}_j)$ for all $j \in \xi_v$ such that
 - $v_j = \mu_j$ when $\mu_j < \nu_{p_j}(g)$,
 - $v_j > \nu_{p_j}(g)$ when $\mu_j = \nu_{p_j}(g)$.

Denote $\Lambda_e := \prod_{j \in \xi_e} p_j^{\lambda_j - \mu_j}$, $\Lambda_v := \prod_{j \in \xi_v} p_j^{\lambda_j - \mu_j}$ and $\Lambda = \Lambda_e \Lambda_v$. Remark that $\Lambda_e, \Lambda_v, \Lambda$ contain all the information of ξ_v, ξ_e and μ_j since $\xi_v = \mathcal{P}(\Lambda_v)$, $\xi_e = \mathcal{P}(\Lambda_e)$ and $\mu_j = \lambda_j - \nu_{p_j}(\Lambda)$.

We are ready to define our error models. The random received words $\mathbf{R} = (v_j, \mathbf{r}_j)_j$ are uniformly distributed in the following set $\mathcal{B}_{\Lambda_e \Lambda_v, \mathfrak{R}}^1$

1. $\mathbf{R}_j = \text{Ev}^\infty(\mathbf{f}/g)_j$ for all j such that $p_j \notin \mathcal{P}(\Lambda)$,
2. $\mathbf{R}_j = \mathfrak{R}_j$ for all j such that $p_j \in \mathcal{P}(\Lambda_v)$,
3. $\mathbf{R}_j = (\nu_{p_j}(g), \mathbf{r}_j)$ with $\nu_{p_j}(\mathbf{r}_j - \mathcal{S}_j(\mathbf{f}/g)) = \mu_j - \nu_{p_j}(g)$ for all j such that $p_j \in \mathcal{P}(\Lambda_e)$.

As before, we will determine the distribution of the error matrices $\mathbf{E}_{\mathbf{R}, \text{Ev}^\infty(\mathbf{f}/g)}$ when \mathbf{f}/g is fixed and \mathbf{R} is random.

For $i \in \{1, \dots, \ell\}$, we still denote $E_i \in \mathbb{Z}/N\mathbb{Z}$ the CRT interpolant of the i -th row of \mathbf{E} , and we obtain that $Y|E_i$ for every index $i = 1, \dots, \ell$ as in Subsection 2.6. We define the modular integers $E'_i := E_i/Y \in \mathbb{Z}/\Lambda\mathbb{Z}$, which verify $\gcd(E'_1, \dots, E'_\ell, \Lambda) = 1$.

Because of our hybrid error model where the randomness only appears on the columns $j \in \mathcal{P}(\Lambda_e)$, we need to study the random vector $(E'_1 \bmod \Lambda_e, \dots, E'_\ell \bmod \Lambda_e)$.

Lemma 4.15. *If \mathbf{R} is uniformly distributed in $\mathcal{B}_{\Lambda_e \Lambda_v, \mathfrak{R}}^1$, then the random vector $(E'_1 \bmod \Lambda_e, \dots, E'_\ell \bmod \Lambda_e)$ is uniformly distributed in the sample space Ω_{Λ_e} .*

Proof. For the duration of this proof, we will only consider indices j such that $p_j \in \mathcal{P}(\Lambda_e)$. Recall that $\mathbf{e}_j = p_j^{\nu_{p_j}(g)} (\mathbf{r}_j - \mathcal{S}_j(\mathbf{f}/g)) \bmod p_j^{\lambda_j}$ for all those particular j . Since $Y = p_j^{\mu_j} \bmod p_j^{\lambda_j}$, we get that $E'_i = E_i/Y = e_{i,j}/Y \bmod p_j^{\lambda_j - \mu_j}$ and

$$\mathbf{e}_j/Y = (\mathbf{r}_j - \mathcal{S}_j(\mathbf{f}/g))/p_j^{\mu_j - \nu_{p_j}(g)} \bmod p_j^{\lambda_j - \mu_j}.$$

Therefore, by definition of $\mathcal{B}_{\Lambda_e \Lambda_v, \mathfrak{R}}^1$, the vector $\mathbf{e}_j/Y \in (\mathbb{Z}/p_j^{\lambda_j - \mu_j}\mathbb{Z})^\ell$ is random of valuation 0. As a consequence, we obtain that $(E'_1 \bmod \Lambda_e, \dots, E'_\ell \bmod \Lambda_e)$ is random among the vectors of $(\mathbb{Z}/\Lambda_e\mathbb{Z})^\ell$ such that $\gcd(E'_1, \dots, E'_\ell, \Lambda_e) = 1$. \square

Second error model. Similarly, we need to fix a reduced vector of rationals $\mathbf{f}/g \in \mathbb{Q}^\ell$, valuation ξ_v and evaluation $\xi_{m,e}$ error supports, error valuations $(\mu_j)_{1 \leq j \leq n}$ and a partial received word $\mathfrak{R}_j = (v_j, \mathbf{r}_j)$ for all $j \in \xi_v$. All these parameters must satisfy the same conditions as the first error model.

The set $\xi_{m,e}$ is now called the maximal error support because actual errors could result in an evaluation error support $\xi_e \subset \xi_{m,e}$.

Denote $\Lambda_{m,e} := \prod_{j \in \xi_{m,e}} p_j^{\lambda_j - \mu_j}$, $\Lambda_v := \prod_{j \in \xi_v} p_j^{\lambda_j - \mu_j}$ and $\Lambda_m = \Lambda_{m,e} \Lambda_v$.

In the second error model, the random received words $\mathbf{R} = (v_j, \mathbf{r}_j)_j$ are uniformly distributed in the following set $\mathcal{B}_{\Lambda_{m,e} \Lambda_v, \mathfrak{R}}^2$

1. $\mathbf{R}_j = \text{Ev}^\infty(\mathbf{f}/g)_j$ for all j such that $p_j \notin \mathcal{P}(\Lambda_m)$,
2. $\mathbf{R}_j = \mathfrak{R}_j$ for all j such that $p_j \in \mathcal{P}(\Lambda_v)$,
3. $\mathbf{R}_j = (\nu_{p_j}(g), \mathbf{r}_j)$ with $\nu_{p_j}(\mathbf{r}_j - \mathcal{S}_j(\mathbf{f}/g)) \geq \mu_j - \nu_{p_j}(g)$ for all j such that $p_j \in \mathcal{P}(\Lambda_{m,e})$.

Notice that for a given received word in the set $\mathcal{B}_{\Lambda_{m,e} \Lambda_v, \mathfrak{R}}^2$, the associated error locator has the form $\Lambda = \Lambda_e \Lambda_v$ for some divisor $\Lambda_e | \Lambda_{m,e}$.

4.5. Our results on bad primes

We are ready to state our results regarding the failure probability of the decoding algorithm in presence of bad primes. We let \bar{d}_e be the maximal distance on the evaluation errors

$$\bar{d}_e := \frac{\ell}{\ell+1} \lceil \log(N/2FG) - \log(3\beta) - 2d_v \rceil \quad (16)$$

Theorem 4.16. *Decoding Algorithm 2 on input*

1. distance parameter $d = d_v + d_e$ for $d_v \leq \log\left(\sqrt{N/(6FG\beta)}\right)$ and $d_e \leq \bar{d}_e$,
2. a random received word $\mathbf{R} = (v_j, \mathbf{r}_j)_{1 \leq j \leq n}$ uniformly distributed in $\mathcal{B}_{\Lambda_e \Lambda_v, \mathfrak{R}}^1$, for some reduced vector of fractions $\mathbf{f}/g \in \mathbb{Q}^\ell$ with $\|\mathbf{f}\|_\infty < F$, $0 < g < G$, and $\log(\Lambda_v) \leq d_v$ and $\log(\Lambda_e) \leq d_e$,

outputs the center vector \mathbf{f}/g of the distribution with a probability of failure

$$\mathbb{P}_{\text{fail}} \leq 2^{-(\ell+1)(\bar{d}_e - d_e)} \prod_{p \in \mathcal{P}(\Lambda_e)} \left(\frac{1 - 1/p^{\ell + \nu_p(\Lambda_e)}}{1 - 1/p^\ell} \right).$$

Theorem 4.17. *Decoding Algorithm 2 on input*

1. distance parameter $d = d_v + d_e$ for $d_v \leq \log\left(\sqrt{N/(6FG\beta)}\right)$ and $d_e \leq \bar{d}_e$,
2. a random received word $\mathbf{R} = (v_j, \mathbf{r}_j)_{1 \leq j \leq n}$ uniformly distributed in $\mathcal{B}_{\Lambda_{m,e} \Lambda_v, \mathfrak{R}}^2$, for some reduced vector of fractions $\mathbf{f}/g \in \mathbb{Q}^\ell$ with $\|\mathbf{f}\|_\infty < F$, $0 < g < G$, and $\log(\Lambda_v) \leq d_v$ and $\log(\Lambda_{m,e}) \leq d_e$,

outputs the center vector \mathbf{f}/g of the distribution with a probability of failure

$$\mathbb{P}_{\text{fail}} \leq 2^{-(\ell+1)(\bar{d}_e - d_e)} \prod_{p \in \mathcal{P}(\Lambda_{m,e})} \left(\frac{1 - 1/p^{\ell + \nu_p(\Lambda_{m,e})}}{1 - 1/p^{\ell+1}} \right).$$

Remark 4.18. The results presented here have a polynomial counterpart in the context of rational function codes with multiplicities and poles as studied in [GLLZ23].

We remark that the results given in this paper provide several improvements on the state of the art of the polynomial counterpart (see [GLLZ23, Theorem 3.4]). For instance, the failure probability bound decreases exponentially when the actual error distance is less than the maximal error distance in this paper, whereas the failure probability bound in [GLLZ23] is a linear function of the distance parameter. Furthermore, our bound removes the technical dependency of the multiplicity balancing, making the results independent of how the multiplicities are distributed.

In this work, we establish our results only in the setting of rational numbers. Following similar lines of reasoning, we have also obtained analogous theorems in the case of rational functions; however, to keep the paper concise, enhance readability, and focus on the more original contributions, we have opted not to include these statements.

4.6. Decoding failure probability with respect to the first error model

We let

$$S_{\mathbf{E}}^{\infty} := \left\{ \omega \in \mathbb{Z}/\Lambda_e \mathbb{Z} : \forall i, \omega \tilde{E}'_i \in \mathbb{Z}_{\Lambda_e, B\Lambda} \right\}$$

with $B := 2^d \beta \frac{2FG}{N} = 2^{d_e+d_v} \beta \frac{2FG}{N}$ and $\tilde{E}'_i := \text{CRT}_N \left(g/p_j^{\nu_{p_j}(g)} \right) E_i \pmod{N}$. We can now prove the version of Lemma 2.23 with bad primes.

Constraint 4.19. The parameters of Algorithm 1 satisfy $2^{d_v} B < 1$.

Lemma 4.20. *If Constraint 4.19 is satisfied then $S_{\mathbf{E}}^{\infty} = \{0\} \Rightarrow S_{\mathbf{R}} \subseteq v_C \mathbb{Z}$.*

Proof. Let $(\varphi, \psi_1, \dots, \psi_\ell) \in S_{\mathbf{R}} = S_{\mathbf{R}, 2^d \beta}$. From (13) we know that $\prod_{j=1}^n p_j^{\nu_j} | \varphi$ and that for every i, j there exists $h_{i,j} \in \mathbb{Z}$ such that $\varphi r_{i,j} = p_j^{\nu_j} \psi_i + h_{i,j} p_j^{\lambda_j}$. Furthermore,

$$\varphi \Lambda_v \tilde{e}_{i,j} = p_j^{\nu_j} \Lambda_v (\varphi f_i - g \psi_i) - \Lambda_v g h_{i,j} p_j^{\lambda_j} \pmod{p_j^{\lambda_j + \nu_j}}. \quad (17)$$

From

$$\nu_{p_j}(\Lambda_v g) = \begin{cases} \lambda_j - \min\{\nu_j, \nu_{p_j}(g)\} + \nu_{p_j}(g) & \text{if } \nu_j \neq \nu_{p_j}(g) \\ \nu_{p_j}(g) & \text{if } \nu_j = \nu_{p_j}(g) \end{cases},$$

as $\lambda_j \geq \nu_j$, we conclude that $\nu_{p_j}(\Lambda_v g) \geq \nu_j$ for every $j = 1, \dots, n$. Taking the CRT interpolant modulo N on both sides of (17) after dividing by $p_j^{\nu_j}$, we conclude that

$$\text{CRT}_N(\varphi/p_j^{\nu_j}) \Lambda_v \tilde{E}'_i = \Lambda_v (\varphi f_i - g \psi_i) \pmod{N}$$

with $\tilde{E}_i := \text{CRT}_N \left(g/p_j^{\nu_{p_j}(g)} \right) E_i \pmod N$. The integer $Y\Lambda_v$ divides both $\Lambda_v \tilde{E}_i$ and N , so it divides $\Lambda_v(\varphi f_i - g\psi_i)$. Dividing by $Y\Lambda_v$, we obtain

$$\text{CRT}_{\Lambda_e} \left(\frac{\varphi}{p_j^{\nu_j}} \right) \tilde{E}'_i = \frac{\varphi f_i - g\psi_i}{Y} \pmod{\Lambda_e},$$

with $\tilde{E}'_i := \text{CRT}_{\Lambda_e} \left(g/p_j^{\nu_{p_j}(g)} \right) E'_i \pmod{\Lambda_e}$. Thus, $\omega := \text{CRT}_{\Lambda_e} (\varphi/p_j^{\nu_j}) \in S_{\mathbf{E}}^\infty$ and, thanks to the hypothesis $S_{\mathbf{E}}^\infty = \{0\}$, $(\varphi f_i - g\psi_i)/Y = 0 \pmod{\Lambda_e}$. Thanks to Constraint 4.19 and since $\Lambda_v \leq 2^{d_v}$, we have $\frac{2FG}{N} 2^d \beta \Lambda_v < 1$ which implies $\frac{|g\psi_i - f_i\varphi|}{Y} \leq \frac{2FG}{N} 2^d \beta \Lambda < \Lambda_e$. As a result, $g\psi_i = f_i\varphi$ for all $i = 1, \dots, \ell$. Since $\text{gcd}(f_1, \dots, f_\ell, g) = 1$, we must have that $g|\varphi$, i.e. $\varphi = sg$ for some $s \in \mathbb{Z}$ and, from the above conclusion, as well that $\boldsymbol{\psi} = s\mathbf{f}$. Let us note

$$s\tilde{e}_j = p_j^{\nu_j} \boldsymbol{\psi} - \varphi \mathbf{r}_j = \mathbf{0} \pmod{p_j^{\lambda_j}}.$$

As $\nu_{p_j}(\tilde{e}_j) = \nu_{p_j}(e_j) = \lambda_j - \text{val}_j(\Lambda)$, we obtain $\nu_j(s) \geq \lambda_j - (\lambda_j - \text{val}_j(\Lambda))$ for every j , i.e. Λ divides s . \square

Remark 4.21. Along the same lines of Remark 2.24 relative to the analysis of Section 2, also in this context we see that, at the cost assuming of $\beta = 1$, our technique yields the unique decoding when the distance parameter d of Algorithm 2 is below unique decoding capacity. Indeed, when $d < \log(\sqrt{N/(2FG)})$, we must have that $B\Lambda < \beta$ since $\Lambda \leq 2^d$. Under such circumstance we therefore have $\mathbb{Z}_{\Lambda_e, B\Lambda} = \mathbb{Z}_{\Lambda_e, 0} = \{0\}$. Thus, also in this case, estimating the failure probability of Algorithm 2 by studying $\mathbb{P}(S_{\mathbf{E}}^\infty \neq \{0\})$ yields the expected unique decoding result when $d < \log(\sqrt{N/(2FG)})$.

Proof of Theorem 4.16. Since Constraint 4.11 is verified for all received words in our random distribution, Lemma 4.12 and an adaptation of Lemma 2.14 shows that $\mathbb{P}_{\text{fail}} \leq \mathbb{P}(S_{\mathbf{R}} \not\subseteq v_{\mathbf{C}}\mathbb{Z})$.

We can prove that our choice of parameters satisfy Constraint 4.19 in the same fashion as the proof of Theorem 3.2. So we can apply Lemma 4.20 to obtain $\mathbb{P}(S_{\mathbf{R}} \not\subseteq v_{\mathbf{C}}\mathbb{Z}) \leq \mathbb{P}(S_{\mathbf{E}}^\infty \neq \{0\})$.

As in Equation (7), we have $\mathbb{P}(S_{\mathbf{E}}^\infty \neq \{0\}) \leq \sum_{\omega=1}^{\Lambda_e-1} \mathbb{P}(\forall i, \omega \tilde{E}'_i \in \mathbb{Z}_{\Lambda_e, B\Lambda})$. Since $\tilde{E}_i = \text{CRT}_N \left(g/p_j^{\nu_{p_j}(g)} \right) E_i \pmod N$, and since $\text{CRT}_N \left(g/p_j^{\nu_{p_j}(g)} \right)$ is an invertible element of $\mathbb{Z}/N\mathbb{Z}$, we have that for every $1 \leq \omega \leq \Lambda_e - 1$, $\mathbb{P}(\forall i, \omega \tilde{E}'_i \in \mathbb{Z}_{\Lambda_e, B\Lambda}) = \mathbb{P}(\forall i, \omega E'_i \in \mathbb{Z}_{\Lambda_e, B\Lambda})$. Now, since we know the distribution of $(E'_i)_{1 \leq i \leq n}$ thanks to Lemma 4.15, we use Lemma 3.7 with Λ_i and d_u being replaced by Λ_e and d_v to get

$$\sum_{\omega=1}^{\Lambda_e-1} \mathbb{P}(\forall i, \omega E'_i \in \mathbb{Z}_{\Lambda_e, B\Lambda}) \leq (3B2^{d_v})^\ell \Lambda_e \prod_{p \in \mathcal{P}(\Lambda_e)} \left(\frac{1 - 1/p^{\ell + \nu_p(\Lambda_e)}}{1 - 1/p^\ell} \right).$$

Since $\Lambda_e \leq 2^{d_e}$, we have $(3B2^{d_v})^\ell \Lambda_e \leq (3B)^\ell 2^{\ell d_v} 2^{d_e} = 2^{-(\ell+1)(\bar{d}_e - d_e)}$, we have proven Theorem 4.16. \square

4.7. Decoding failure probability with respect to the second error model

We will denote $\mathbb{P}_{\mathcal{B}_{\Lambda_m, e, \Lambda_v, \mathfrak{R}}^2}$ (resp. $\mathbb{P}_{\mathcal{B}_{\Lambda_e \Lambda_v, \mathfrak{R}}^1}$) the probability function under the second (resp. first) error model specified by a given factorization Λ_m, e, Λ_v of the error locator, and by a partial received word $(\mathfrak{R}_j)_{j \in \mathcal{P}(\Lambda_v)}$.

Proof of Theorem 4.17. As done in the proof of Theorem 3.3, letting \mathcal{F} be the event of decoding failure. We will denote $\mathbb{P}_{\mathcal{B}_{\Lambda_m, e, \Lambda_v, \mathfrak{R}}^2}$ (resp. $\mathbb{P}_{\mathcal{B}_{\Lambda_e \Lambda_v, \mathfrak{R}}^1}$) the probability function under the second (resp. first) error model specified by a given factorization Λ_m, e, Λ_v of the error locator, and by a partial received word $(\mathfrak{R}_j)_{j \in \mathcal{P}(\Lambda_v)}$. Using the law of total probability, we have that $\mathbb{P}_{\mathcal{B}_{\Lambda_m, e, \Lambda_v, \mathfrak{R}}^2}(\mathcal{F})$ can be decomposed as the sum

$$\sum_{\Lambda_e | \Lambda_m, e} \mathbb{P}_{\mathcal{B}_{\Lambda_m, e, \Lambda_v, \mathfrak{R}}^2}(\mathcal{F} | \Lambda_E = \Lambda_e \Lambda_v) \mathbb{P}_{\mathcal{B}_{\Lambda_m, e, \Lambda_v, \mathfrak{R}}^2}(\Lambda_E = \Lambda_e \Lambda_v),$$

where

$$\mathbb{P}_{\mathcal{B}_{\Lambda_m, e, \Lambda_v, \mathfrak{R}}^2}(\mathcal{F} | \Lambda_E = \Lambda_e \Lambda_v) = \mathbb{P}_{\mathcal{B}_{\Lambda_e \Lambda_v, \mathfrak{R}}^1}(\mathcal{F})$$

is upper bounded by

$$\mathbb{P}_{\mathcal{B}_{\Lambda_e \Lambda_v, \mathfrak{R}}^1}(\mathcal{F}) \leq (3B2^{d_v})^\ell \Lambda_e \prod_{p \in \mathcal{P}(\Lambda_e)} \left(\frac{1 - 1/p^{\ell + \nu_p(\Lambda_e)}}{1 - 1/p^\ell} \right)$$

Whereas

$$\begin{aligned} \mathbb{P}_{\mathcal{B}_{\Lambda_m, e, \Lambda_v, \mathfrak{R}}^2}(\mathcal{F} | \Lambda_E = \Lambda_e \Lambda_v) &= \frac{\prod_{p \in \mathcal{P}(\Lambda_e)} (p^\ell - 1) p^{\ell(\nu_p(\Lambda_e) - 1)}}{\prod_{p \in \mathcal{P}(\Lambda_m, e)} p^{\ell \nu_p(\Lambda_m, e)}} \\ &= \left(\frac{\Lambda_e}{\Lambda_m, e} \right)^\ell \prod_{p \in \mathcal{P}(\Lambda_e)} \left(1 - \frac{1}{p^\ell} \right). \end{aligned}$$

Plugging the above in the decomposition of $\mathbb{P}_{\mathcal{B}_{\Lambda_m, e, \Lambda_v, \mathfrak{R}}^2}(\mathcal{F})$ and following the proof of Theorem 3.3 with $\Lambda_m, i, \Lambda_i, \Lambda_u$ being replaced by $\Lambda_m, e, \Lambda_e, \Lambda_v$ respectively, we conclude the proof of Theorem 4.17. \square

References

- [AAGL23] Matteo Abbondati, Antoine Afflatet, Eleonora Guerrini, and Romain Lebreton. Probabilistic analysis of LLL-based decoder of interleaved Chinese remainder codes. In *ITW 2023-IEEE Information Theory Workshop*, 2023.
- [AGL24] Matteo Abbondati, Eleonora Guerrini, and Romain Lebreton. Decoding simultaneous rational evaluation codes. In *Proceedings of the 2024 International Symposium on Symbolic and Algebraic Computation*, pages 153–161, 2024.

- [AM18] Divesh Aggarwal and Priyanka Mukhopadhyay. Improved algorithms for the shortest vector problem and the closest vector problem in the infinity norm. In *29th International Symposium on Algorithms and Computation*, 2018.
- [BDFP15] Janko Böhm, Wolfram Decker, Claus Fieker, and Gerhard Pfister. The use of bad primes in rational reconstruction. *Mathematics of Computation*, 84(296):3013–3027, 2015.
- [BK14] Brice Boyer and Erich L Kaltofen. Numerical linear system solving with parametric entries by error correction. In *Proceedings of the 2014 Symposium on Symbolic-Numeric Computation*, pages 33–38, 2014.
- [Cab71] Stanley Cabay. Exact solution of linear equations. In *Proceedings of the second ACM symposium on Symbolic and algebraic manipulation*, pages 392–398, 1971.
- [Dix82] John D. Dixon. Exact solution of linear equations using P-adic expansions. *Numerische Mathematik*, 40(1):137–141, February 1982.
- [GLLZ23] Eleonora Guerrini, Kamel Lairedj, Romain Lebreton, and Ilaria Zappatore. Simultaneous rational function reconstruction with errors: Handling multiplicities and poles. *Journal of Symbolic Computation*, 116:345–364, 2023.
- [GLZ19] Eleonora Guerrini, Romain Lebreton, and Ilaria Zappatore. Polynomial linear system solving with errors by simultaneous polynomial reconstruction of interleaved Reed-Solomon codes. In *2019 IEEE International Symposium on Information Theory (ISIT)*, pages 1542–1546. IEEE, 2019.
- [GLZ20] Eleonora Guerrini, Romain Lebreton, and Ilaria Zappatore. On the uniqueness of simultaneous rational function reconstruction. In *Proceedings of the 45th International Symposium on Symbolic and Algebraic Computation, ISSAC '20*, pages 226–233, New York, NY, USA, July 2020. Association for Computing Machinery.
- [GRS99] Oded Goldreich, Dana Ron, and Madhu Sudan. Chinese remaindering with errors. In *Proceedings of the thirty-first annual ACM symposium on Theory of computing*, pages 225–234, 1999.
- [KPR⁺10] Majid Khonji, Clément Pernet, Jean-Louis Roch, Thomas Roche, and Thomas Stalinski. Output-sensitive decoding for redundant residue systems. In *Proceedings of the 2010 International Symposium on Symbolic and Algebraic Computation*, pages 265–272, 2010.
- [KPSW17] Erich L Kaltofen, Clément Pernet, Arne Storjohann, and Cleveland Waddell. Early termination in parametric linear system solving and

- rational function vector recovery with error correction. In *Proceedings of the 2017 ACM on International Symposium on Symbolic and Algebraic Computation*, pages 237–244, 2017.
- [KPY20] Erich L Kaltofen, Clément Pernet, and Zhi-Hong Yang. Hermite rational function interpolation with error correction. In *Computer Algebra in Scientific Computing: 22nd International Workshop, CASC 2020, Linz, Austria, September 14–18, 2020, Proceedings 22*, pages 335–357. Springer, 2020.
- [Leb12] Romain Lebreton. *Contributions à l’algorithmique détendue et à la résolution des systèmes polynomiaux*. These de doctorat, Palaiseau, Ecole polytechnique, January 2012.
- [Lip71] John D Lipson. Chinese remainder and interpolation algorithms. In *Proceedings of the second ACM symposium on Symbolic and algebraic manipulation*, pages 372–391, 1971.
- [LLL82] Arjen K Lenstra, Hendrik Willem Lenstra, and László Lovász. Factoring polynomials with rational coefficients. *Mathematische annalen*, 261(ARTICLE):515–534, 1982.
- [LSN13] Wenhui Li, Vladimir Sidorenko, and Johan SR Nielsen. On decoding interleaved Chinese remainder codes. In *2013 IEEE International Symposium on Information Theory*, pages 1052–1056. IEEE, 2013.
- [MC79] R. T. Moenck and J. H. Carter. Approximate algorithms to derive exact solutions to systems of linear equations. In *EUROSAM ’79*, volume 72, pages 65–73. Springer, 1979.
- [McC77] Michael T McClellan. The exact solution of linear equations with rational function coefficients. *ACM Transactions on Mathematical Software (TOMS)*, 3(1):1–25, 1977.
- [Mon04] Michael Monagan. Maximal quotient rational reconstruction: an almost optimal algorithm for rational reconstruction. In *Proceedings of the 2004 international symposium on Symbolic and algebraic computation*, pages 243–249, 2004.
- [OS07] Zach Olesh and Arne Storjohann. The vector rational function reconstruction problem. In *Computer Algebra 2006: Latest Advances in Symbolic Algorithms*, pages 137–149. World Scientific, 2007.
- [Per14] Clément Pernet. *High performance and reliable algebraic computing*. PhD thesis, Université Joseph Fourier, Grenoble 1, 2014.
- [RS16] Johan Rosenkilde né Nielsen and Arne Storjohann. Algorithms for simultaneous Padé approximations. In *Proceedings of the ACM on International Symposium on Symbolic and Algebraic Computation*, pages 405–412, 2016.

- [Sch87] Claus-Peter Schnorr. A hierarchy of polynomial time lattice basis reduction algorithms. *Theoretical computer science*, 53(2-3):201–224, 1987.
- [SSB09] Georg Schmidt, Vladimir R Sidorenko, and Martin Bossert. Collaborative decoding of interleaved Reed–Solomon codes and concatenated code designs. *IEEE Transactions on Information Theory*, 55(7):2991–3012, 2009.
- [Sto05] Arne Storjohann. The shifted number system for fast linear algebra on integer matrices. *Journal of Complexity*, 21(4):609–650, August 2005.
- [Vil97] Gilles Villard. A study of Coppersmith’s block Wiedemann algorithm using matrix polynomials. Technical report, IMAG, 1997.
- [Zap20] Ilaria Zappatore. *Simultaneous Rational Function Reconstruction and applications to Algebraic Coding Theory*. PhD thesis, Université Montpellier, 2020.