Decoding Simultaneous Rational Evaluation Codes

Matteo Abbondati, Eleonora Guerrini, Romain Lebreton

Decoding Simultaneous Rational Evaluation Codes

Matteo Abbondati
LIRMM - University of Montpellier
Montpellier, France
matteo.abbondati@lirmm.fr

Eleonora Guerrini
LIRMM - University of Montpellier
Montpellier, France
eleonora.guerrini@lirmm.fr

Romain Lebreton
LIRMM - University of Montpellier
Montpellier, France
romain.lebreton@lirmm.fr

ABSTRACT

In this paper, we deal with the problem of simultaneous reconstruction of a vector of rational numbers, given modular reductions containing errors (SRNRwE). Our methods apply as well to the simultaneous reconstruction of rational functions given evaluations containing errors (SRFRwE), improving known results [7, 9]. In the latter case, one can take advantage of techniques from coding theory [4, 10] and provide an algorithm that extends classical Reed-Solomon decoding. In recent works [7, 9], interleaved Reed-Solomon codes [3, 19] are used to correct beyond the unique decoding capability in the case of random errors at the price of positive but small failure probability. Our first contribution is to extend these works to the simultaneous reconstruction with errors of rational numbers instead of functions. Thus considering rational number codes [16], we provide an algorithm decoding beyond the unique decoding capability and, as a central result of this paper, we analyze in detail its failure probability. Our analysis generalizes for the first time the best known analysis for interleaved Reed-Solomon codes [19] to SRFRwE, improving on the existing bound [8], to interleaved Chinese remainder codes, also improving the known bound [1], and finally for the first time to SRNRwE.

CCS CONCEPTS

• **Computing methodologies** → **Algebraic algorithms**; **Linear algebra algorithms**; • **Mathematics of computing** → **Probabilistic algorithms**; **Coding theory**; **Interpolation**.

KEYWORDS

Simultaneous rational number reconstruction, Simultaneous rational function reconstruction, Fault tolerant algorithm, Reed Solomon codes, Chinese remainder codes, Interleaved codes, Decoding failure probability analysis.

ACM Reference Format:
Matteo Abbondati, Eleonora Guerrini, and Romain Lebreton. 2024. Decoding Simultaneous Rational Evaluation Codes. In *International Symposium on Symbolic and Algebraic Computation (ISSAC '24), July 16–19, 2024, Raleigh, NC, USA.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3666000.3669686

1 INTRODUCTION

The solution of a linear system $A\vec{x} = \vec{b}$ with ℓ unknowns and with coefficients in an integral domain R, can be written as a vector $\vec{x} = \left(\frac{f_1}{g}, \dots, \frac{f_\ell}{g}\right)$ of elements in the field of fractions of R sharing the same denominator (the largest invariant factor of the matrix A). In this paper we deal with both the cases $R = \mathbb{Z}$ and $R = \mathbb{F}_q[x]$ for some finite field $\mathbb{F}_q$. Following the evaluation-interpolation technique of [5], we consider the resolution of the system in the framework of a distributed network in which, given n distinct evaluation points $\alpha_1, \dots, \alpha_n \in \mathbb{F}_q$ (in the case $R = \mathbb{F}_q$) or n distinct prime numbers $p_1 < \dots < p_n$ (in the case $R = \mathbb{Z}$), a central node delegates the resolution of the reduced systems modulo $(x - \alpha_j)$ (or modulo p_j if $R = \mathbb{Z}$) for $j = 1, \dots, n$ to n computing nodes. These nodes send the n reductions of the solution to the central node, which can therefore reconstruct the vector $\vec{x}$ with an interpolation algorithm in the form of a simultaneous rational reconstruction [4, 7–10]. It thus needs to solve an instance of a simultaneous rational number reconstruction with errors (SRNRwE, see Problem 2 below) in the case $R = \mathbb{Z}$ or an instance of a simultaneous rational function reconstruction with errors (SRFRwE, see Problem 3 below) in the case $R = \mathbb{F}_q[x]$. Both can be seen respectively as generalizations of decoding interleaved Chinese remainder codes [1, 12] or interleaved Reed-Solomon codes [3]. In coding theory, the correction capacity is expressed in terms of the minimum distance of the code (minimum of the relative distances between code words). It is classical to use the Hamming distance for mono-alphabetic codes and a weighted Hamming distance in the poly-alphabetic scenario. For the integer case, in order to express that the coordinates depend on the associated moduli, we define the weighted Hamming distance (see Definition 1 below). In what follows $\mathbb{Z}_p$ will denote the quotient ring modulo the ideal (p), $\prod_{j=1}^{n} \mathbb{Z}_{p_j}$ will denote the Cartesian product $\mathbb{Z}_{p_1} \times \dots \times \mathbb{Z}_{p_n}$ while $[x]_p$ will denote the modular element $x \bmod p \in \mathbb{Z}_p$.

Definition 1 (Weighted Hamming distance). Let $\boldsymbol{R}^1, \boldsymbol{R}^2 \in (\prod_{j=1}^{n} \mathbb{Z}_{p_j})^\ell$ be two $\ell \times n$ matrices, where each row belongs to $\prod_{j=1}^{n} \mathbb{Z}_{p_j}$. We define their error support as $\xi_{\boldsymbol{R}^1,\boldsymbol{R}^2} := \bigcup_{i=1}^{\ell}\{j : \boldsymbol{R}^1_{i,j} \neq \boldsymbol{R}^2_{i,j}\}$ and their error locator as the product of the primes in the error support $\Lambda_{\boldsymbol{R}^1,\boldsymbol{R}^2} := \prod_{j \in \xi_{\boldsymbol{R}^1,\boldsymbol{R}^2}} p_j$. The weighted Hamming distance between $\boldsymbol{R}^1$ and $\boldsymbol{R}^2$ is defined as $d(\boldsymbol{R}^1, \boldsymbol{R}^2) := \log(\Lambda_{\boldsymbol{R}^1,\boldsymbol{R}^2})$.

Set $N := \prod_{j=1}^{n} p_j$. Thanks to the Chinese remainder theorem, each row of the matrices can be viewed as a modular element in $\mathbb{Z}_N$, we call this its CRT interpolant.

Problem 2 (SRNRwE). *Given $\ell > 0$, n distinct primes $p_1 < \dots < p_n$, a received matrix $\boldsymbol{R} \in (\prod_{j=1}^{n} \mathbb{Z}_{p_j})^\ell$, 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) $d\left(\left([f_i/g]_{p_j}\right)_{i,j}, \boldsymbol{R}\right) \leq d$,
(2) *for all* $1 \leq i \leq \ell$, $|f_i| < F$, $0 < g < G$ *and* $\gcd(g, N) = 1$.

In the above problem we have that $\gcd(g, N) = 1$, so that the reduction $[f_i/g]_{p_j}$ is well-defined. For the rational functions case, we will use the notation $\partial(p)$ to denote the degree of a polynomial $p \in \mathbb{F}_q[x]$. We use the Hamming distance $d(\boldsymbol{R}^1, \boldsymbol{R}^2) := \partial(\Lambda_{\boldsymbol{R}^1,\boldsymbol{R}^2}) = \#\xi_{\boldsymbol{R}^1,\boldsymbol{R}^2}$ for any pair of received matrices $\boldsymbol{R}^1, \boldsymbol{R}^2 \in \mathbb{F}_q^{\ell\times n}$, where $\Lambda_{\boldsymbol{R}^1,\boldsymbol{R}^2} := \prod_{j\in\xi_{\boldsymbol{R}^1,\boldsymbol{R}^2}}(x - \alpha_j)$. Set also $M(x) := \prod_{j=1}^{n}(x - \alpha_j)$.

Problem 3 (SRFRwE). *Given $\ell > 0$, n distinct evaluation points $\alpha_1, \ldots, \alpha_n \in \mathbb{F}_q$, a received matrix $\boldsymbol{R} \in \mathbb{F}_q^{\ell\times n}$, an error parameter t and two degree bounds d_f, d_g such that $d_f + d_g \leq n+1$, find a reduced vector of fractions $(f_1/g, \ldots, f_\ell/g) \in \mathbb{F}_q(x)^\ell$ such that*

(1) $d\left(\left(f_i(\alpha_j)/g(\alpha_j)\right)_{i,j}, \boldsymbol{R}\right) \leq t$,
(2) *for all* $1 \leq i \leq \ell$, $\partial(f_i) < d_f$, $\partial(g) < d_g$ *and* $\gcd(g, M) = 1$.

In analogy with the integer case, we will always assume that $g(\alpha_j) \neq 0$ for every $j = 1, \ldots, n$. Both these problems can be reduced to the simultaneous error correction of ℓ code words (sharing the same denominator) for the rational number code and the rational function code respectively. These are rational extensions of Chinese remainder codes [6] and Reed-Solomon codes [17], and can be referred to as rational evaluation codes. It seems these rational codes were part of the folklore; to the best of our knowledge, they were first introduced by Pernet in [16, § 2.5.2].

The two conditions $FG < N/2$ and $d_f + d_g \leq n + 1$ guarantee an injective encoding. A long series of papers can be found in the literature where evaluation-interpolation is used for linear systems solving, as [13–15, 18, 21]. Our contributions in this paper concern error correction beyond guaranteed uniqueness. This means that, in order to correct a large number of errors, we compromise the uniqueness of the solution to the problem. The idea is therefore to exceed the number of errors that guarantee uniqueness and to analyze the probability of failure in detail. A failure here will be expressed by a case where the solution of the rational reconstruction with errors problem is not unique. It turns out that our analysis follows and generalizes to rational case the best known analysis of the decoding of interleaved Reed-Solomon codes [19].

This paper is structured as follows: In Section 2 we deal with the definition of rational number codes and their decoding algorithm. In Section 3 we present our main results about the failure analysis of the algorithm and in Section 4 we present the adaptation of previous results to the rational functions case. This last result improves the analysis of [9], giving a generalization of the best known analysis of the decoding failure in the polynomial case [19].

2 SIMULTANEOUS RATIONAL NUMBER CODES

We can define an error correcting code from the SRNRwE problem. Code words are the encoding of reduced vectors of rational numbers $(f_1/g, \ldots, f_\ell/g)$ sharing the same denominator and such that $0 < g < G$, and $|f_i| < F$ for all $i = 1, \ldots, \ell$. The adjective "simultaneous" is kept for referring to the rational reconstruction problem and has no reference to any action (as is the case in "simultaneous decoding").

Definition 4. *Given n distinct primes $p_1 < \ldots < p_n$, two positive bounds F, G such that $FG < N/2$ and an integer $\ell > 0$, we define the simultaneous rational number code as the set of matrices*

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

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

The condition $\gcd(f_1, \ldots, f_\ell, g) = 1$, which is going to be used in the proof of Lemma 20, reflects that the solution vector we seek to reconstruct is a reduced vector of rational numbers.

Remark 5. *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 2 with error parameter d.*

2.1 Unique decoding and minimal distance

The distance $d(C) := \min_{c_1\neq c_2\in C} d(c_1, c_2)$ of a code C plays an important role in coding theory to assess the amount of data one can correct. In the special case $\ell = 1$, $SRN_\ell(N; F, G)$ codes correspond to rational number codes $RN(N; F, G)$ [16, §2.5.2] whose weighted Hamming distance is given in [16, Theorem 2.5.1].

Theorem 6. *Let N, F, G as in Definition 4. The distance of an RN code satisfies $d(RN(N; F, G)) > \log\left(\frac{N}{2FG}\right)$.*

This result has the advantage of being independent of the moduli p_j. However, the gap between $d(RN(N; F, G))$ and $\log(N/(2FG))$ depends on the moduli. Even so, there exists a family of RN codes such that $d(RN(N; F, G)) \leq \log(N/((F-1)(G-1)))$, *i.e.* the gap is small [16, §2.5.2]. We can generalize Theorem 6 to SRN codes:

Lemma 7. *We have $d(SRN_\ell(N; F, G)) > \log\left(\frac{N}{2FG}\right)$.*

Proof. Let $C_1 = \left([f_i/g]_{p_j}\right)_{i,j}$ and $C_2 = \left([f_i'/g']_{p_j}\right)_{i,j}$ be two code words. For $j \notin \xi_{C_1,C_2}$, $f_i/g = f_i'/g' \bmod p_j$ for all i. We set $Y := \prod_{j\notin\xi_{C_1,C_2}} p_j$, so that $Y|(f_i g' - f_i' g)$ for all i. Since $|f_i|, |f_i'| < F$, and $0 < g, g' < G$ we have $Y < 2FG$. Using the relation $Y = N/\Lambda_{C_1,C_2}$, we bound $d(C_1, C_2) = \log(\Lambda_{C_1,C_2}) = \log(N/Y) > \log(N/2FG)$. □

Note that the family of RN codes such that the distance inequality is tight extends to SRN codes.

Unique decoding. A unique decoding function D of capacity t is a function from the ambient space to the code such that $D(r) = c$ for all code word c and r such that $d(r, c) \leq t$. 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 [16, Corollary 2.5.2]. A classical result in coding theory states that, for codes equipped with the Hamming distance, there exists such a decoding function of capacity t if and only if $2t < d(C)$. 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 $2t < d(C)$, then there exists such a decoding function of capacity t. However, the converse is false in the strict sense of the term. Indeed, in the proof that there can not exist a decoding function when $2t = d(C)$, one takes $c_1, c_2 \in C$ such that $d(C) = d(c_1, c_2)$, and constructs r as the middle of c_1 and c_2, *i.e.* with $d(c_1, r) = d(c_2, r) = d(c_1, c_2)/2$, to 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 $2t = d(C) + \varepsilon$ for a small ε, then we can construct r such that $d(c_1, r), d(c_2, r) \leq (d(c_1, c_2) + \varepsilon)/2 = t$, and no decoding function of capacity t can exist. One workaround in coding theory 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.

2.2 Decoding SRN codes

This section presents our first contribution: a decoder of SRN codes of capacity beyond $\frac{d(C)}{2}$. This decoder is based on the interleaved Chinese remainder (ICR) codes decoder of [1, 12], which are a special case of SRN when $g = 1$. Let $\boldsymbol{R} := (r_{i,j})_{\substack{1 \leq i \leq \ell \\ 1 \leq j \leq n}}$ be the received matrix. For any code word $\boldsymbol{C} \in SRN_\ell(N; F, G)$, we can write $\boldsymbol{R} = \boldsymbol{C} + \boldsymbol{E}$ for some error matrix $\boldsymbol{E}$ (which depends on $\boldsymbol{R}$ and $\boldsymbol{C}$). Thanks to the Chinese remainder theorem, we can view each row of the matrix as modular elements in $\mathbb{Z}_N$, and the ambient space for the code can be viewed as $\mathbb{Z}_N^\ell$, 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 $\Lambda := \Lambda_{\boldsymbol{C},\boldsymbol{R}}$, we know [16] that the system of ℓ equations holds:

$$\Lambda f_i = \Lambda g R_i \bmod N \text{ for } i = 1, \dots, \ell \tag{1}$$

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

$$\psi_i = \varphi R_i \bmod N \text{ for } i = 1, \dots, \ell \tag{2}$$

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

$$\mathcal{L} = \operatorname{Span} \begin{pmatrix} 1 & R_1 & \cdots & R_\ell \\ 0 & N & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & N \end{pmatrix}. \tag{3}$$

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

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

Note that the condition $\Lambda_{\boldsymbol{C},\boldsymbol{R}} \leq 2^d$ means that $\boldsymbol{C}$ is close to $\boldsymbol{R}$ for the weighted Hamming distance. The idea of the decoding is now to compute an element of $S_{\boldsymbol{R},d}$ and "hope" that it will be a multiple of v_C; if this is the case, by dividing all the entries by the first one we can retrieve the correct vector of fractions $(f_1/g, \dots, f_\ell/g)$. 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 $S_{\boldsymbol{R},d}$, 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_C. Concerning the analysis of this second aspect, more will be said in Section 3.3. 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 $\mathcal{ASVP}_\infty$ which solves the following problem:

Problem 8 ($\beta-$Approx-SVP$_{\|\cdot\|_\infty}$). *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 [2] for state-of-the-art algorithms solving Problem 8. Without loss of generality, we will assume that the output w of the algorithm $\mathcal{ASVP}_\infty$ satisfies $w_0 \geq 0$ (both $\pm w$ are short vectors). We will also assume that w is $\mathcal{L}$-reduced:

Definition 9. *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 $S_{\boldsymbol{R},d}$ do not correspond exactly to conditions on the $\|\cdot\|_\infty$ norm, we need to introduce a scaling operator $\sigma_{F,G} : \mathbb{Q}^{\ell+1} \to \mathbb{Q}^{\ell+1}$ 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 $S_{\boldsymbol{R},d}$ into

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

Therefore, a vector $v' \in \bar{\mathcal{L}}$ which satisfies $\|v'\|_\infty < 2^d FG$ must belong to $S'_{\boldsymbol{R},d}$. We can obtain a candidate solution v_s 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 $\boldsymbol{R}$ is relatively close to the code (see Constraint 10 below), since v_s is a β-approximation of the shortest vector, it belongs to a slightly larger solution set.

Constraint 10. *There exists a code word $\boldsymbol{C}$ such that $\Lambda_{\boldsymbol{C},\boldsymbol{R}} \leq 2^d$.*

Lemma 11. *Assuming Constraint 10, we have that $v_s \in S_{\boldsymbol{R}} := S_{\boldsymbol{R},\tau}$ with $\tau := d + \log(\beta)$.*

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

We notice that assuming Constraint 10 we also have $v_C \in S_{\boldsymbol{R}}$. We can now state our decoding algorithm for SRN codes.

Algorithm 1: SRN codes decoder.

Input: $SRN_\ell(N; F, G)$, received word $\boldsymbol{R}$, distance bound d
Output: A code word $\boldsymbol{C}$ s.t. $d(\boldsymbol{C}, \boldsymbol{R}) \leq d$ or "decoding failure"

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

2.3 A particular sub-routine: LLL

We remark that the complexity of Algorithm 1 is mainly determined by the complexity of the sub-routine $\mathcal{ASVP}_\infty$. In particular the authors of [2] showed that the space and time complexity for the resolution of Problem 8 are significantly larger than the relative costs for the resolution of the ℓ_2-norm version of the same problem.

Problem 12 (γ − Approx-SVP$_{\|\cdot\|_2}$). *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 $\mathcal{L}$.*

Nevertheless, a γ-approximation SVP for the ℓ_2-norm yields a $\gamma\sqrt{\ell+1}$-approximation SVP for the $\ell_\infty-$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. $\ell_\infty-$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 12 is given by LLL [11], which runs in polynomial time for the approximation factor $\gamma = \sqrt{2}^\ell$ (our lattice has dimension $\ell+1$). For this reason, we can always assume to employ a sub-routine $\mathcal{ASVP}_\infty$ which solves Problem 8 with $\beta \leq \sqrt{2}^\ell\sqrt{\ell+1}$.

The most efficient $\gamma-$Approx-SVP$_{\|\cdot\|_2}$ solver is given by the BKZ algorithm [20]. It finds a solution of Problem 12 with $\gamma = (1+\epsilon)^{\ell+1}$ in polynomial time of degree increasing as $\epsilon \to 0$. 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 13. *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 $c = \pm 1$. □

3 CORRECTNESS OF THE DECODER

In this section, we study the correctness of Algorithm 1. We start with Lemma 14 which states that the algorithm is correct when it does not fail.

Lemma 14. *If Algorithm 1 returns $\boldsymbol{C}$ on input $\boldsymbol{R}$ and parameter d, then $\boldsymbol{C}$ is a code word of $SRN(N; F, G)$ such that $d(\boldsymbol{C}, \boldsymbol{R}) \leq d$.*

Proof. The output vector $\boldsymbol{C} = (\psi'_1/\varphi', \dots, \psi'_\ell/\varphi')$ is a code word of $SRN(N; F, G)$ 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 $(\varphi, \psi_1, \dots, \psi_\ell) = (\lambda\varphi', \lambda\psi'_1, \dots, \lambda\psi'_\ell)$ is in the lattice $\mathcal{L}$, so that $\lambda(\varphi' R_i - \psi'_i) = 0 \bmod N$ for all i. Dividing by the invertible φ' modulo N, we obtain $\lambda(R_i - C_i) = 0 \bmod N$ for all i. For all $j \in \xi_{\boldsymbol{C},\boldsymbol{R}}$, there exists i such that $p_j \nmid (R_i - C_i)$, which implies that $p_j | \lambda$. As a consequence, $\Lambda_{\boldsymbol{C},\boldsymbol{R}} | \lambda$. Considering that $\lambda \leq 2^d$, we can conclude that $d(\boldsymbol{C}, \boldsymbol{R}) = \log \Lambda_{\boldsymbol{C},\boldsymbol{R}} \leq \log\lambda \leq d$. □

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

Lemma 15. *Assuming Constraint 10, if Algorithm 1 fails, then $v_s \notin v_C\mathbb{Z}$.*

Proof. We will prove this by contraposition, thus we show that if $v_s = rv_C$, for some $r \in \mathbb{Z}$ then the algorithm must succeed. We know that $v_s = rv_C$ is $\mathcal{L}$-reduced therefore $v_C = \pm v_s$ and $\lambda = \Lambda \leq 2^d$ using Constraint 10 (see Algorithm 1, Step 4 for λ), $\varphi' = \pm g, \psi'_j = \pm f_j$ for every j, thus the algorithm succeeds. □

The rest of this section is dedicated to the analysis of the decoding failure of Algorithm 1. We will show that if $\boldsymbol{R}$ is $\boldsymbol{C}$ plus a random error of weighted Hamming distance up to approximately $\ell/(\ell+1)\log(N/(2FG))$ (see Section 3.1 for precise error models), then this decoder is able to decode most of the time (see Section 3.2 for the statement of the theorem).

3.1 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 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 [1, 19], specifying two possible distributions of the random received word $\boldsymbol{R}$.

Error Model 1. In this error model we fix an error support ξ (see Definition 1), then the columns of the error matrix $\boldsymbol{E}$ are distributed independently as follows

$$\vec{e}_j = \vec{0} \text{ if } j \notin \xi, \quad \vec{e}_j \sim \mathcal{U}\left(\mathbb{Z}_{p_j}^\ell \setminus \{\vec{0}\}\right) \text{ if } j \in \xi \tag{4}$$

where $\mathcal{U}(\mathcal{S})$ denotes the uniform distribution on any finite set $\mathcal{S}$. For any given code word $\boldsymbol{C}$ and error support ξ, we obtain the distribution $\mathcal{D}^{ERR1}_{C,\xi} := \{\boldsymbol{R} = \boldsymbol{C} + \boldsymbol{E} : \boldsymbol{E} \text{ as in Eq.(4)}\}$ of the random received words $\boldsymbol{R}$ around the central code word $\boldsymbol{C}$. We will need another point of view on the random error matrices $\boldsymbol{E}$. Let $i \in \{1, \dots, \ell\}$, and denote $E_i \in \mathbb{Z}_N$ the CRT interpolant of the i-th row of $\boldsymbol{E}$. Since $p_j | E_i$ for all i and $j \notin \xi$, we have that $Y | E_i$ for all i, where $Y := N/\Lambda$. We define the modular integers $E'_i := E_i/Y \in \mathbb{Z}_\Lambda$. The random variables $(E'_i)_{1\leq i\leq\ell}$ are uniformly distributed in $\{(F_i)_{1\leq i\leq\ell} \in (\mathbb{Z}_\Lambda)^\ell : \gcd(F_1, \dots, F_\ell, \Lambda) = 1\}$, because if $p|\Lambda$, then there is an error modulo p, so $\exists i$ s.t. $E_i \neq 0 \bmod p$ and therefore $\gcd(E_1, \dots, E_\ell, \Lambda) = 1$.

Error Model 2. In this error model we fix a maximal error support ξ_r and the columns of the error matrix $\boldsymbol{E}$ are distributed as follows

$$\vec{e}_j = \vec{0} \text{ if } j \notin \xi_r, \quad \vec{e}_j \sim \mathcal{U}\left(\mathbb{Z}_{p_j}^\ell\right) \text{ if } j \in \xi_r \tag{5}$$

We notice that in the error model ERR2, the actual error support ξ could be contained in ξ_r. For a code word $\boldsymbol{C}$ and a maximal error support ξ_r, we have the distribution $\mathcal{D}^{ERR2}_{C,\xi_r} := \{\boldsymbol{R} = \boldsymbol{C} + \boldsymbol{E} : \boldsymbol{E} \text{ as in Eq. (5)}\}$ of the random received words $\boldsymbol{R}$ around the central code word $\boldsymbol{C}$.

3.2 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, and we will adapt the analysis of the failure probability to the two error models in 3.3. In what follows we set

$$d_{\max} := \frac{\ell}{\ell+1}\left[\log(N/2FG) - \log(3\beta)\right]. \tag{6}$$

Remark 16. *Our setting allows decoding up to a distance* $d \le d_{\max}$ *that, for* $\ell > 1$*, can be greater than our estimation* $\log\left(\sqrt{\frac{N}{2FG}}\right)$ *of the unique decoding capability of* $SRN_\ell(N;F,G)$ *codes.*

When fixing the decoding bound d close to $d_{\max}$, 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 true even if $\mathcal{ASVP}_\infty(\bar{\mathcal{L}})$ 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 3.3.1) relative to the failure probability of the decoding algorithm with respect to the error model ERR1.

Theorem 17. *Decoding Algorithm 1 on input a random received word* $\boldsymbol{R} \in \mathcal{D}^{ERR1}_{C,\xi}$*, for some code word* $\boldsymbol{C} \in SRN_\ell(N;F,G)$ *and error support* ξ *such that* $\log\Lambda \le d \le d_{\max}$*, and distance parameter* d*, outputs the center code word* $\boldsymbol{C}$ *of the distribution* $\mathcal{D}^{ERR1}_{C,\xi}$*, with a probability of failure* $\mathbb{P}_{fail} \le 2^{-(\ell+1)(d_{\max}-d)}\exp(n/p_1^\ell)$.

Here is our second result (whose proof will be given at the end of Subsection 3.3.3) relative to the failure probability with respect to the error model ERR2.

Theorem 18. *Decoding Algorithm 1 on input a random received word* $\boldsymbol{R} \in \mathcal{D}^{ERR2}_{C,\xi_r}$*, for some code word* $\boldsymbol{C} \in SRN_\ell(N;F,G)$ *and error support* ξ_r *such that* $\log\Lambda_r \le d \le d_{\max}$*, and distance parameter* d*, outputs the center code word* $\boldsymbol{C}$ *of the distribution* $\mathcal{D}^{ERR2}_{C,\xi_r}$*, with a probability of failure* $\mathbb{P}_{fail} \le 2^{-(\ell+1)(d_{\max}-d)}$.

This failure probability bound improves the one of decoding interleaved Chinese remainder codes $\mathbb{P}_{fail} \le 2^{-(\ell+1)(d_{\max}-d)} + (\exp(n/p_1^{\ell-1}) - 1)$ which was only available in the special case of non-negative ($0 \le f_i$) integer code words ($G = 2$) [1, Theorem 3.5].

3.3 Analysis of the decoding failure probability

For any $\boldsymbol{R} \in \mathcal{D}^{ERR1}_{C,\xi}$ (as in Theorem 17), Constraint 10 is satisfied. Thus, thanks to Lemma 11, we can assume that $v_s \in S_{\boldsymbol{R}} = S_{\boldsymbol{R},\tau}$ where $\tau = d + \log(\beta)$.

3.3.1 $\mathbb{P}_{fail}$ under ERR1.

If Algorithm 1 fails, then $v_s \notin v_{\boldsymbol{C}}\mathbb{Z}$ (see Lemma 15). Note that the converse is not necessarily true, for example if there exists another close code word $\boldsymbol{C}' \ne \boldsymbol{C}$ with $d(\boldsymbol{C}',\boldsymbol{R}) \le d$ and if the SVP solver outputs $v_s = v_{\boldsymbol{C}'}$. Nevertheless, we can upper bound the failure probability of the algorithm as $\mathbb{P}_{fail} \le \mathbb{P}(S_{\boldsymbol{R}} \not\subseteq v_{\boldsymbol{C}}\mathbb{Z})$. We introduce some notations: for $C \in \mathbb{R}_{>0}$ we let $\mathbb{Z}_{m,C} := \{a \in \mathbb{Z}_m : |a \text{ crem } m| \le C\}$, where a 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 $\#\mathbb{Z}_{m,C} \le 2\lfloor C\rfloor + 1$. Let $S_{\boldsymbol{E}}$ be the set $S_{\boldsymbol{E}} := \left\{\varphi \in \mathbb{Z}_\Lambda : \forall i,\ g\varphi E'_i \in \mathbb{Z}_{\Lambda,B}\right\}$ for $B := \frac{2^{\tau+1}FG}{N}\Lambda$. We need a new constraint to prove the following lemma.

Constraint 19. *Algorithm 1 parameters satisfy* $\frac{2^{\tau+1}FG}{N} < 1$.

Lemma 20. *If Constraint 19 is satisfied,* $S_{\boldsymbol{E}} = \{0\} \Rightarrow S_{\boldsymbol{R}} \subseteq v_{\boldsymbol{C}}\mathbb{Z}$.

Proof. Let $(\varphi,\psi_1,\ldots,\psi_\ell) \in S_{\boldsymbol{R}}$. We know that for all $1 \le i \le \ell$, $g\varphi E_i = g\varphi\left(R_i - \frac{f_i}{g}\right) = g\psi_i - f_i\varphi \bmod N$. 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 $g\varphi E'_i = \psi'_i \bmod \Lambda$. Therefore

$$|g\varphi E'_i \text{ crem } \Lambda| \le |\psi'_i| \le \frac{|g\psi_i| + |f_i\varphi|}{Y} < \frac{2^{\tau+1}FG}{N}\Lambda$$

which means that $\varphi \in S_{\boldsymbol{E}}$, thus thanks to the hypothesis $S_{\boldsymbol{E}} = \{0\}$, we get $\Lambda|\varphi$, thus $\psi'_i = 0 \bmod \Lambda$. Thanks to Constraint 19 and the above inequality we can conclude that $|\psi'_i| < \Lambda$, therefore $\psi'_i = 0$ in $\mathbb{Z}$. Which means that

$$\forall i = 1,\ldots,\ell,\ g\psi_i = f_i\varphi. \tag{7}$$

Since $\gcd(f_1,\ldots,f_\ell,g) = 1$, Equations (7) 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 (7), we deduce $g\psi_i = f_i\varphi = f_iag\Lambda$, so $\psi_i = a\Lambda f_i$ for all i. We have shown $(\varphi,\psi_1,\ldots,\psi_\ell) \in (\Lambda g, \Lambda f_1,\ldots,\Lambda f_\ell)\mathbb{Z}$. □

Thanks to the above lemma we can upper bound the failure probability of Algorithm 1 with $\mathbb{P}_{fail} \le \mathbb{P}(S_{\boldsymbol{E}} \ne \{0\})$. In order to estimate the above, we need the following preliminary result:

Lemma 21. *If* $\varphi \in \mathbb{Z}$ *is such that* $\gcd(\varphi,\Lambda) = \nu$*, then for the probability distribution of error model ERR1, we have*

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

where $\mathcal{P}(n)$ *is the set of primes dividing* n.

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

Proof. Since $\gcd(g,N) = 1$, the distributions of the vectors $(\varphi E'_1,\ldots,\varphi E'_\ell)$ and $(g\varphi E'_1,\ldots,g\varphi E'_\ell)$ are identical. Thus, we have $\mathbb{P}(\forall i,\ g\varphi E'_i \in \mathbb{Z}_{\Lambda,B}) = \mathbb{P}(\forall i,\ \varphi E'_i \in \mathbb{Z}_{\Lambda,B})$. Let us now show that $\varphi E'_i \in \mathbb{Z}_{\Lambda,B} \Leftrightarrow (\varphi/\nu)E'_i \in \mathbb{Z}_{\Lambda/\nu,B/\nu}$: The first condition can be rephrased as $\varphi E'_i = a_i\Lambda + c_i$ with $a_i, c_i \in \mathbb{Z}$ and $|c_i| \le B$. But then we must have that $\nu|c_i$. Thus we can divide the above by ν and obtain $(\varphi/\nu)E'_i = a_i\Lambda/\nu + c_i/\nu$ with $|c_i/\nu| \le B/\nu$, which is equivalent to $(\varphi/\nu)E'_i \in \mathbb{Z}_{\Lambda/\nu,B/\nu}$. When $B < \nu$, the previous condition implies that $(\varphi/\nu)E'_i = 0 \bmod \Lambda/\nu$ for all i. Since (φ/ν) is coprime with Λ/ν, we have $E'_i = 0 \bmod \Lambda/\nu$ for all i. If $\nu < \Lambda$, this is in contradiction with $\gcd(E'_1,\ldots,E'_\ell,\Lambda) = 1$ for all random matrix $\boldsymbol{E}$. Therefore, the associated probability $\mathbb{P}(\forall i,\ g\varphi E'_i \in \mathbb{Z}_{\Lambda,B})$ is zero. We have seen that our probability $\mathbb{P}(\forall i,\ g\varphi E'_i \in \mathbb{Z}_{\Lambda,B})$ is equal to $\mathbb{P}(\{\boldsymbol{E} = (\vec{e}_j)_{1\le j\le n} : \forall i,\ (\varphi/\nu)E'_i \in \mathbb{Z}_{\Lambda/\nu,B/\nu}\})$.

Now, the condition $(\varphi/\nu)E'_i \in \mathbb{Z}_{\Lambda/\nu,B/\nu}$ only depends on the columns $(\vec{e}_j)$ of the random matrix for $j \in \xi_{\Lambda/\nu} := \{j : p_j \in \mathcal{P}(\Lambda/\nu)\}$. These columns are uniformly distributed in the sample space $\Omega := \{(\vec{e}_j)_{j\in\xi_{\Lambda/\nu}} : \forall j \in \xi_{\Lambda/\nu},\ \vec{e}_j \ne \vec{0} \in (\mathbb{Z}_{p_j})^\ell\}$. Therefore, if we write the condition as

$$\mathcal{E} := \left\{(\vec{e}_j)_{j\in\xi_{\Lambda/\nu}} \in \prod_{j\in\xi_{\Lambda/\nu}}(\mathbb{Z}_{p_j})^\ell : \forall i,\ g\varphi E'_i \in \mathbb{Z}_{\Lambda,B}\right\},$$

we can deduce that our probability equals

$$\mathbb{P}(\{(\vec{e}_j)_{j\in\xi_{\Lambda/\nu}} : \forall i,\ (\varphi/\nu)E'_i \in \mathbb{Z}_{\Lambda/\nu,B/\nu}\}) = \frac{\#(\Omega\cap\mathcal{E})}{\#\Omega} \le \frac{\#\mathcal{E}}{\#\Omega}.$$

Note that $\#\Omega = \prod_{p\in\mathcal{P}(\frac{\Lambda}{\nu})}(p^\ell - 1)$. When the $(\vec{e}_j)_{j\in\xi_{\Lambda/\nu}}$ are independent and uniformly distributed in $\prod_{j\in\xi_{\Lambda/\nu}}(\mathbb{Z}_{p_j})^\ell$, as it is the case in $\mathcal{E}$, the random variables E'_i are uniformly distributed in

$\mathbb{Z}_{\Lambda/\nu}$. Moreover, since φ/ν is coprime to Λ/ν, the multiplication by φ/ν is a bijection of $\mathbb{Z}_{\Lambda/\nu}$. Therefore, the cardinality of $\mathcal{E}$ is $\#\mathcal{E} = (\#\mathbb{Z}_{\Lambda/\nu,B/\nu})^\ell$. □

We now have the ingredients to prove our upper bound on the failure probability.

LEMMA 22. *Given $E'_1, \dots, E'_\ell$, distributed according to the error model ERR1, we have that*

$$\mathbb{P}(S_{\boldsymbol{E}} \neq \{0\}) \leq \left(6\frac{2^\tau FG}{N}\right)^\ell \Lambda \exp\left(\frac{n}{p_1^\ell}\right).$$

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

$$\mathbb{P}(S_{\boldsymbol{E}} \neq \{0\}) \leq \sum_{\varphi=1}^{\Lambda-1} \mathbb{P}\left(\forall i,\ g\varphi E'_i \in \mathbb{Z}_{\Lambda,B}\right) \tag{8}$$

We can use Lemma 21 and upper bound the terms in the sum with

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

where $\nu = \gcd(\varphi, \Lambda)$. Thanks to the second point in Lemma 21, we can restrict the sum only to the elements φ such that $\nu \leq B$, which in turn allows us to deduce that $\#\mathbb{Z}_{\Lambda/\nu,B/\nu} \leq 2\lfloor B/\nu \rfloor + 1 \leq 3B/\nu$. Since this expression depends only on ν, 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 $\gcd(\varphi, \Lambda) = \nu$, is equal to $\phi\left(\frac{\Lambda}{\nu}\right)$ with ϕ being the Euler totient function. Therefore

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

Plugging in the definition of B we can collect a common term $\left(6\frac{2^\tau FG}{N}\right)^\ell$. Thus, extending the sum over all the divisors ν, we can upper bound the quotient $\mathbb{P}(S_{\boldsymbol{E}} \neq \{0\})/\left(6\frac{2^\tau FG}{N}\right)^\ell$ with

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

where in the last equality we used the distributive property to express a product of the form $\prod_{p\in\mathcal{P}(\Lambda)}[f(p)+1]$, with f an arbitrary function, as a sum $\sum_{\nu|\Lambda}\prod_{p\in\mathcal{P}\left(\frac{\Lambda}{\nu}\right)} f(p)$. Bringing each term to its common denominator, the above product can be rewritten as

$$\prod_{p\in\mathcal{P}(\Lambda)} \left(\frac{p^{\ell+1} - 1}{p^\ell - 1}\right) = \Lambda \prod_{p\in\mathcal{P}(\Lambda)} \left(\frac{p^\ell - \frac{1}{p}}{p^\ell - 1}\right) = \Lambda \prod_{p\in\mathcal{P}(\Lambda)} \left(\frac{1 - \frac{1}{p^{\ell+1}}}{1 - \frac{1}{p^\ell}}\right)$$
$$\leq \Lambda \prod_{p\in\mathcal{P}(\Lambda)} \left(1 + \frac{1}{p^\ell}\right) \leq \Lambda \exp\left(\frac{n}{p_1^\ell}\right).$$

Where the first inequality above is true since

$$\frac{1 - \frac{1}{p^{\ell+1}}}{1 - \frac{1}{p^\ell}} \leq \left(1 + \frac{1}{p^\ell}\right) \Leftrightarrow 1 - \frac{1}{p^{\ell+1}} \leq 1 - \frac{1}{p^{2\ell}} \Leftrightarrow 2\ell \geq \ell + 1,$$

while for the second one we used that $p_1 = \min p_j$ to get

$$\prod_{p\in\mathcal{P}(\Lambda)} \left(1 + \frac{1}{p^\ell}\right) \leq \left(1 + \frac{1}{p_1^\ell}\right)^n \leq \exp\left(\frac{n}{p_1^\ell}\right)$$ □

3.3.2 Proof of Theorem 17. We start by proving that with $\tau = d + \log(\beta)$ and with the hypothesis of Theorem 17, Constraint 19 holds, thus we can apply all the previous lemmas and upper bound the failure probability of Algorithm 1 with the quantity given by Lemma 22. Let us start by verifying that our choice of parameters satisfy Constraint 19:

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

We already noticed when defining the $SRN_\ell(N; F, G)$ code that $2FG < N$. We said in Section 2.3 that we can assume $\beta \leq \sqrt{2}^\ell\sqrt{\ell+1}$. Since $\sqrt{2}^\ell\sqrt{\ell+1} \leq 3^\ell$ for every $\ell \in \mathbb{Z}_{>0}$, the above quantity is smaller than 1 and Constraint 19 is satisfied. Hence, we can upper bound the failure probability using Lemma 22. Thanks to the hypothesis of Theorem 17 we know that $\Lambda \leq 2^d$, and since $2^\tau = \beta 2^d$, and using $2^{d_{\max}(\ell+1)} = (N/(6FG\beta))^\ell$, we have proved Theorem 17. □

3.3.3 $\mathbb{P}_{fail}$ under ERR2. In the second error model, we need to make a distinction between the maximal error support ξ_r (over which there are uniform random errors) and the actual error support ξ, which is included in ξ_r but may be different if a zero column is drawn. We will denote $\mathbb{P}^{ERR2}_{\xi_r}$ (resp. $\mathbb{P}^{ERR1}_{\xi}$) the probability function under the error model 2 with error support ξ_r (resp. the error model 1 with ξ). Let $\mathcal{F}$ be the event of decoding failure, *i.e.* the set of random matrices $\boldsymbol{E}$ such that Algorithm 1 returns "decoding failure". Using the law of total probability, we have

$$\mathbb{P}^{ERR2}_{\xi_r}(\mathcal{F}) = \sum_{\xi\subseteq\xi_r} \mathbb{P}^{ERR2}_{\xi_r}(\mathcal{F} \mid \xi_{\boldsymbol{E}} = \xi)\, \mathbb{P}^{ERR2}_{\xi_r}(\xi_{\boldsymbol{E}} = \xi). \tag{9}$$

where $\xi_{\boldsymbol{E}} := \xi_{\boldsymbol{R},\boldsymbol{C}}$ (see Definition 1). The conditional probabilities $\mathbb{P}^{ERR2}_{\xi_r}(\mathcal{F} \mid \xi_{\boldsymbol{E}} = \xi)$ in the sum are equal to $\mathbb{P}^{ERR1}_{\xi}(\mathcal{F})$, which are upper bounded within the proof of Lemma 22 by

$$\mathbb{P}^{ERR1}_{\xi}(\mathcal{F}) \leq \left(6\frac{2^\tau FG}{N}\right)^\ell \Lambda \prod_{p\in\mathcal{P}(\Lambda)} \left(\frac{p^\ell - \frac{1}{p}}{p^\ell - 1}\right) \tag{10}$$

where $\Lambda = \prod_{j\in\xi} p_j$. Moreover,

$$\mathbb{P}^{ERR2}_{\xi_r}(\xi_{\boldsymbol{E}} = \xi) = \frac{\prod_{j\in\xi}(p_j^\ell - 1)}{\prod_{j\in\xi_r} p_j^\ell} = \frac{\prod_{p\in\mathcal{P}(\Lambda)}(p^\ell - 1)}{\Lambda_r^\ell} \tag{11}$$

where $\Lambda_r = \prod_{j\in\xi_r} p_j$. Using these facts we can prove Theorem 18.

3.3.4 Proof of Theorem 18. Plug Equations (11) and (10) in Equation (9) to obtain that $\mathbb{P}^{ERR2}_{\xi_r}(\mathcal{F})/(6\frac{2^\tau FG}{N})^\ell$ is less than or equal to

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

Now, thanks to the hypothesis of the theorem, we know that $\Lambda_r \leq 2^d$, and since $2^\tau = 2^d\beta$, we can write

$$\mathbb{P}^{ERR2}_{\xi_r}(\mathcal{F}) \leq \left(6\frac{2^\tau FG}{N}\right)^\ell \Lambda_r \leq \left(6\frac{2^d\beta FG}{N}\right)^\ell 2^d = 2^{d(\ell+1)}\left(6\frac{\beta FG}{N}\right)^\ell.$$

Using $2^{d_{\max}(\ell+1)} = (N/(6FG\beta))^\ell$, we have proved Theorem 18. □

4 THE RATIONAL FUNCTION CASE

In this section we show how the previous analysis fits the decoding of simultaneous rational functions codes, for the resolution of Problem 3. This improves the result of [19], generalizing the best known analysis of the decoding failure for interleaved Reed-Solomon codes to the rational function case [7]. We relate Problem 3 with the decoding problem for the rational extension of Reed-Solomon codes, defined as follows

Definition 23. *Given n distinct evaluation points $\alpha_1, \ldots, \alpha_n \in \mathbb{F}_q$, let $M(x) := \prod_{i=1}^n (x - \alpha_i) \in \mathbb{F}_q[x]$, two degree bounds $d_f, d_g \in \mathbb{Z}_{>0}$ such that $d_f + d_g \leq n + 1$ and a parameter $\ell > 0$, we define the simultaneous rational function code as the set of matrices*

$$SRF_\ell(M; d_f, d_g) := \left\{ \left(\frac{f_i}{g}(\alpha_j)\right)_{\substack{1\leq i\leq \ell\\ 1\leq j\leq n}} : \begin{array}{c}\partial(f_i) < d_f, \partial(g) < d_g,\\ \gcd(f_1, \ldots, f_\ell, g) = 1\\ \gcd(M, g) = 1\end{array}\right\}.$$

Recall that we denote $\partial(p)$ the degree of a polynomial $p \in \mathbb{F}_q[x]$. Let $\boldsymbol{R} := (r_{i,j})_{\substack{1\leq i\leq \ell\\ 1\leq j\leq n}}$ be the received matrix. For any code word $\boldsymbol{C} \in SRF_\ell(M; d_f, d_g)$, we can write $\boldsymbol{R} = \boldsymbol{C}+\boldsymbol{E}$ for some error matrix $\boldsymbol{E}$. We can associate an interpolation polynomial to every row, which we write $R_i = C_i+E_i$. We set $\xi_{\boldsymbol{R},\boldsymbol{C}} := \{j : \exists i,\ r_{ij} \neq c_{ij}\}$, $d(\boldsymbol{R}, \boldsymbol{C}) = \#\xi_{\boldsymbol{R},\boldsymbol{C}}$ and $\Lambda_{\boldsymbol{R},\boldsymbol{C}} = \prod_{j\in\xi_{\boldsymbol{R},\boldsymbol{C}}}(x - \alpha_j)$. We refer to Λ and ξ instead of $\Lambda_{\boldsymbol{R},\boldsymbol{C}}$ and $\xi_{\boldsymbol{R},\boldsymbol{C}}$ for short. We know that $\Lambda f_i = \Lambda g R_i \bmod M(x)$ holds for any $1 \leq i \leq \ell$ [16]. Making the substitutions $\varphi \leftarrow \Lambda g, \psi_i \leftarrow \Lambda f_i$ we linearize the previous equations, obtaining the *key equations*

$$\psi_i = \varphi R_i \bmod M(x) \text{ for } i = 1, \ldots, \ell$$

which are $\mathbb{F}_q$-linear. In particular if $\partial(\Lambda) \leq t$ for some distance parameter t, the solution vector $v_{\boldsymbol{C}} := (\Lambda g, \Lambda f_1, \ldots, \Lambda f_\ell)$ belongs to the $\mathbb{F}_q$-linear subspace

$$S_{\boldsymbol{R}} := \left\{(\varphi, \psi_1, \ldots, \psi_\ell) \in \mathbb{F}_q[x]^{\ell+1} : \begin{array}{c}\psi_i = \varphi R_i \bmod M(x)\\ \partial(\varphi) < d_g + t, \partial(\psi_i) < d_f + t\end{array}\right\}.$$

In this context the decoding is a linear problem. Indeed we know that $v_{\boldsymbol{C}} \in S_{\boldsymbol{R}}$, and we can compute an element $(\varphi, \psi_1, \ldots, \psi_\ell) \in S_{\boldsymbol{R}}$ by solving the linear equations $\psi_i = \varphi R_i \bmod M(x)$ for the $\ell(d_f + t) + d_g + t$ coefficients of the polynomials $\varphi, \psi_1, \ldots, \psi_\ell$. The idea of the algorithm is to find a non-zero element $(\varphi, \psi_1, \ldots, \psi_\ell)$ in $S_{\boldsymbol{R}}$, then compute $\lambda = \gcd(\varphi, \psi_1, \ldots, \psi_\ell)$, $\varphi' := \varphi/\lambda$, $\psi_i' := \psi_i/\lambda$ and check if $\partial(\lambda) \leq t$, $\partial(\varphi') < d_g$ and $\partial(\psi_i') < d_f$ (see for example [10, Algorithm 2.1]). If this holds, then we can state an equivalent of Lemma 14 that guarantees the correctness of the algorithm. Regarding the distributions of the received word $\boldsymbol{R}$, in analogy with Subsection 3.1, we define the two distributions as follows.

Error Model 1. Let ξ be a fixed error support, and let the columns of the error matrix $\boldsymbol{E}$ be distributed as follows

$$\vec{e}_j = \vec{0} \text{ if } j \notin \xi, \quad \vec{e}_j \sim \mathcal{U}\left(\mathbb{F}_q^\ell \setminus \{\vec{0}\}\right) \text{ if } j \in \xi. \tag{12}$$

For any given code word $\boldsymbol{C}$ and error support ξ, we obtain the distribution $\mathcal{D}^{ERR1}_{\boldsymbol{C},\xi} := \{\boldsymbol{R} = \boldsymbol{C}+\boldsymbol{E} : \boldsymbol{E} \text{ as in Eq. (12)}\}$ of the random received words $\boldsymbol{R}$ around the central code word $\boldsymbol{C}$.

Error Model 2. Let ξ_r be a fixed maximal error support, and let the columns of the error matrix $\boldsymbol{E}$ be distributed as follows

$$\vec{e}_j = \vec{0} \text{ if } j \notin \xi_r, \quad \vec{e}_j \sim \mathcal{U}\left(\mathbb{F}_q^\ell\right) \text{ if } j \in \xi_r. \tag{13}$$

For any given code word $\boldsymbol{C}$ and maximal error support ξ_r, we obtain the distribution $\mathcal{D}^{ERR2}_{\boldsymbol{C},\xi_r} := \{\boldsymbol{R} = \boldsymbol{C} + \boldsymbol{E} : \boldsymbol{E} \text{ as in Eq. (13)}\}$ of the random received words $\boldsymbol{R}$ around the central code word $\boldsymbol{C}$.

4.1 Our Results

Under these two error models, we can correct up to

$$t_{\max} := \frac{\ell}{\ell+1}(n - d_f - d_g + 1) \tag{14}$$

errors, with the probability of failure of the decoding algorithm given respectively by the following theorems.

Theorem 24. *Given a random received word $\boldsymbol{R} \in \mathcal{D}^{ERR1}_{\boldsymbol{C},\xi}$ for some code word $\boldsymbol{C} \in SRF_\ell(M; d_f, d_g)$ and error support ξ such that $t := \#\xi \leq t_{\max}$, the decoding algorithm returns the center $\boldsymbol{C}$ of the distribution with a probability of failure $\mathbb{P}_{fail}$ upper-bounded by*

$$\mathbb{P}_{fail} \leq \left(\frac{q^\ell - \frac{1}{q}}{q^\ell - 1}\right)^t \frac{q^{-(\ell+1)(t_{\max}-t)}}{q-1}.$$

This result generalizes the best known bound [19, Theorem 7] for Interleaved Reed-Solomon codes to rational functions.

Theorem 25. *Given a random received word $\boldsymbol{R} \in \mathcal{D}^{ERR2}_{\boldsymbol{C},\xi_r}$ for some code word $\boldsymbol{C} \in SRF_\ell(M; d_f, d_g)$ and maximal error support ξ_r such that $t := \#\xi_r \leq t_{\max}$, the decoding algorithm returns the center $\boldsymbol{C}$ of the distribution with a probability of failure $\mathbb{P}_{fail}$ upper-bounded by $\mathbb{P}_{fail} \leq q^{-(\ell+1)(t_{\max}-t)}/(q-1)$.*

This failure probability bound improves the previous known bound for SRFRwE: $\mathbb{P}_{fail} \leq (d_g + t)/q$ (see [8, Theorem 4]). We stress out that this result also applies to Interleaved Reed-Solomon codes ($d_g = 1$), and this gives a new best bound in the context of error model 2, to the best of our knowledge.

4.2 Analysis of the failure probability

As we did in Section 3.3, we start with the proof of the result in the error model ERR1, and then we use the total law of probability to reduce this result to an estimate for the probability in the error model ERR2 (see Subsection 4.2.2). Given the random received word $\boldsymbol{R} \in \mathcal{D}^{ERR1}_{\boldsymbol{C},\xi}$ as in Theorem 24, we know that we can write $\boldsymbol{R} = \boldsymbol{C} + \boldsymbol{E}$ with $\xi_{\boldsymbol{R},\boldsymbol{C}} = \xi$, $\#\xi = t$ and $\boldsymbol{C} = (f_i(\alpha_j)/g(\alpha_j))_{\substack{i=1,\ldots,\ell\\ j=1,\ldots,n}}$, thus the solution vector $v_{\boldsymbol{C}} := (\Lambda g, \Lambda f_1, \ldots, \Lambda f_\ell) \in S_{\boldsymbol{R}}$. The linear algebra decoding approach of [4, 9, 10] allows computing an element $(\varphi, \psi_1, \ldots, \psi_\ell) \in S_{\boldsymbol{R}}$, thus if $S_{\boldsymbol{R}} \subseteq v_{\boldsymbol{C}}\mathbb{F}_q[x]$ then the decoding succeeds; which means that $\mathbb{P}_{fail} \leq \mathbb{P}(S_{\boldsymbol{R}} \not\subseteq v_{\boldsymbol{C}}\mathbb{F}_q[x])$. Letting $E_i \in \mathbb{F}_q[x]$ be the interpolation polynomial associated to the i-th row of the error matrix $\boldsymbol{E}$, we still have the equations $\Lambda E_i = 0 \bmod M(x)$, thus defining $Y := M/\Lambda \in \mathbb{F}_q[x]$ we have that $E_i = 0 \bmod Y$ and we can define $E_i' := E_i/Y \in \mathbb{F}_q[x]$.

We let $B := d_f + d_g + 2t - n - 2$ and for $m \in \mathbb{F}_q[x]$ and $C \in \mathbb{Z}_{>0}$ such that $C \leq \partial(m)$, we introduce the sets

$$\mathbb{F}_q[x]_{m,C} := \left\{ p(x) \in {}^{\mathbb{F}_q[x]}\!/_{m} : \partial(p \text{ rem } m) \leq C \right\},\ \#\mathbb{F}_q[x]_{m,C} = q^{C+1}$$

$$S_E := \left\{ \varphi \in {}^{\mathbb{F}_q[x]}\!/_{\Lambda} : g\varphi E'_i \in \mathbb{F}_q[x]_{\Lambda,B} \right\}$$

where p rem m is the usual remainder of the Euclidean division between p and m. We can now state the polynomial version of Lemma 20, which requires the

Constraint 26. *The parameters satisfy $t < n - d_f - d_g + 2$.*

Lemma 27. *Assuming Constraint 26, $S_E = \{0\} \Rightarrow S_R \subseteq v_C \mathbb{F}_q[x]$.*

Proof. Let $(\varphi, \psi_1, \ldots, \psi_\ell) \in S_R$. We know that $g\varphi E_i = g\varphi R_i - f_i\varphi = g\psi_i - f_i\varphi \bmod M$ and since Y divides M and E_i, we can define the polynomial $\psi'_i = (g\psi_i - f_i\varphi)/Y$ such that $g\varphi E'_i = \psi'_i \bmod \Lambda$ and whose degree is bounded by $\partial(\psi'_i) = \partial(g\psi_i - f_i\varphi) - \partial(Y) \leq (d_f + d_g + t - 2) - (n - t)$ which is less than t thanks to Constraint 26. Thus $\varphi \bmod \Lambda \in S_E$ therefore by hypothesis $\Lambda | \varphi$ in $\mathbb{F}_q[x]$, and thanks to the above $\Lambda | \psi'_i$. Since we proved that $\partial(\psi'_i) < t = \partial(\Lambda)$ we can conclude that $g\psi_i = f_i\varphi$ for all i. Since $\gcd(f_1, \ldots, f_\ell, g) = 1$, as in the proof of Lemma 20 we conclude that $(\varphi, \psi_1, \ldots, \psi_\ell) \in (\Lambda g, \Lambda f_1, \ldots, \Lambda f_\ell)\mathbb{F}_q[x]$. □

Thanks to the above lemma we can upper bound the failure probability of the decoding algorithm as $\mathbb{P}_{fail} \leq \mathbb{P}(S_E \neq \{0\})$. A standard argument of probability shows that $\mathbb{E}[\#S_E] = \sum_{m \geq 1} \mathbb{P}(\#S_E \geq m) \geq 1 + (q-1)\mathbb{P}(S_E \neq \{0\})$, because $\mathbb{P}(\#S_E \geq 1) = 1$ and, for $2 \leq m \leq q$, $\mathbb{P}(\#S_E \geq m) = \mathbb{P}(S_E \neq \{0\})$ since the cardinality of the $\mathbb{F}_q$-vector space S_E is a power of q. Therefore, we have $\mathbb{P}(S_E \neq \{0\}) \leq (\mathbb{E}[\#S_E] - 1)/(q-1)$. Using the expression $\mathbb{E}[\#S_E] = \sum_{\varphi \in \mathbb{F}_q[x]/\Lambda} \mathbb{P}(\varphi \in S_E)$ and $\mathbb{P}(0 \in S_E) = 1$, we can write

$$\begin{aligned} \mathbb{P}(S_E \neq \{0\}) &\leq \frac{1}{q-1} \sum_{\varphi \in (\mathbb{F}_q[x]/\Lambda)\setminus\{0\}} \mathbb{P}(\varphi \in S_E) \\ &= \frac{1}{q-1} \sum_{\varphi \in (\mathbb{F}_q[x]/\Lambda)\setminus\{0\}} \mathbb{P}\left(\forall i,\ g\varphi E'_i \in \mathbb{F}_q[x]_{\Lambda,B}\right). \quad (15) \end{aligned}$$

We can now prove the polynomial version of Lemma 21.

Lemma 28. *Given $\varphi \in \mathbb{F}_q[x]/\Lambda$, let $v := \gcd(\varphi, \Lambda)$, then*

$$\mathbb{P}\left(\forall i,\ g\varphi E'_i \in \mathbb{F}_q[x]_{\Lambda,B}\right) \leq \frac{\left(\#\mathbb{F}_q[x]_{\Lambda/v, B-\partial(v)}\right)^\ell}{\left(q^\ell - 1\right)^{\partial(\Lambda/v)}}.$$

If also $B < \partial(v) < \partial(\Lambda)$, then $\mathbb{P}\left(\forall i,\ g\varphi E'_i \in \mathbb{F}_q[x]_{\Lambda,B}\right) = 0$.

Proof. Since $\gcd(g, M) = 1$, the distributions of the vectors $(g\varphi E'_1, \ldots, g\varphi E'_\ell)$ and $(\varphi E'_1, \ldots, \varphi E'_\ell)$ are identical. Thus, we have $\mathbb{P}\left(\forall i,\ g\varphi E'_i \in \mathbb{F}_q[x]_{\Lambda,B}\right) = \mathbb{P}\left(\forall i,\ \varphi E'_i \in \mathbb{F}_q[x]_{\Lambda,B}\right)$. We now show that $\varphi E'_i \in \mathbb{F}_q[x]_{\Lambda,B} \Leftrightarrow (\varphi/v)E'_i \in \mathbb{F}_q[x]_{\frac{\Lambda}{v}, B-\partial(v)}$: The first condition can be rephrased as $\varphi E'_i = a_i\Lambda + c_i$ with $a_i, c_i \in \mathbb{F}_q[x]$ and $\partial(c_i) \leq B$, but then we must have that $v | c_i$. Thus we can divide the above by v and obtain $(\varphi/v)E'_i = a_i\Lambda/v + c_i/v$ with $\partial(c_i/v) \leq B - \partial(v)$ which is equivalent to $(\varphi/v)E'_i \in \mathbb{F}_q[x]_{\frac{\Lambda}{v}, B-\partial(v)}$. When $B < \partial(v) < \partial(\Lambda)$, the previous condition implies that $(\varphi/v)E'_i = 0 \bmod \Lambda/v$ for all i. Since (φ/v) is coprime with Λ/v, we have $E'_i = 0 \bmod \Lambda/v$ for all i. If $\partial(v) < \partial(\Lambda)$, this is in contradiction with $\gcd(E'_1, \ldots, E'_\ell, \Lambda) = 1$ for all random matrix $\boldsymbol{E}$. Therefore, the associated probability $\mathbb{P}(\forall i,\ g\varphi E'_i \in \mathbb{F}_q[x]_{\Lambda,B})$ is zero. To conclude, thanks to the coprimality between Λ/v and φ/v the distributions of the vectors $\left((\varphi/v)E'_i\right)_{i=1}^\ell$ and $\left(E'_i\right)_{i=1}^\ell$ are the same. So, we need to compute the probability $\mathbb{P}\left(\forall i,\ E'_i \in \mathbb{F}_q[x]_{\Lambda/v, B-\partial(v)}\right)$. The condition $E'_i \in \mathbb{F}_q[x]_{\frac{\Lambda}{v}, B-\partial(v)}$ only depends on the columns $(\vec{e}_j)$ of the random matrix $\boldsymbol{E}$ for $j \in \xi_{\Lambda/v} := \{j : (\Lambda/v)(\alpha_j) = 0\}$ with $\#\xi_{\Lambda/v} = \partial(\Lambda/v)$. These columns are uniformly distributed in the sample space $\Omega := (\mathbb{F}_q^\ell \setminus \{\vec{0}\})^{\partial(\Lambda/v)}$. Therefore, if we write the condition as $\mathcal{E} := \left\{ (\vec{e}_j)_{j \in \xi_{\Lambda/v}} \in \mathbb{F}_q^{\ell \times \partial(\Lambda/v)} : \forall i,\ g\varphi E'_i \in \mathbb{F}_q[x]_{\Lambda,B} \right\}$, our probability is

$$\frac{\#(\Omega \cap \mathcal{E})}{\#\Omega} \leq \frac{\#\mathcal{E}}{\#\Omega} = \frac{\left(\#\mathbb{F}_q[x]_{\Lambda/v, B-\partial(v)}\right)^\ell}{\left(q^\ell - 1\right)^{\partial(\Lambda/v)}}. \quad \square$$

4.2.1 Proof of Theorem 24. Using the bound of Eq. (15) and the previous Lemma 28 we have

$$\mathbb{P}(S_E \neq \{0\}) \leq \frac{1}{q-1} \sum_{\substack{\varphi \in (\mathbb{F}_q[x]/\Lambda)\setminus\{0\} \\ \partial(\gcd(\varphi,\Lambda)) \leq B}} \frac{\left(q^{B - \partial(\gcd(\varphi,\Lambda)) + 1}\right)^\ell}{\left(q^\ell - 1\right)^{\partial(\Lambda/\gcd(\varphi,\Lambda))}}.$$

Since the generic term in the sum depends only on the degree of $\gcd(\varphi, \Lambda)$, we can regroup the sum by φ such that $\partial(\gcd(\varphi, \Lambda)) = r$. There are $\binom{t}{t-r}(q-1)^{t-r}$ such φ. Thus we can write

$$\mathbb{P}(S_E \neq \{0\}) \leq \frac{1}{q-1} \sum_{r=0}^{t} \binom{t}{t-r} (q-1)^{t-r} \frac{\left(q^{B-r+1}\right)^\ell}{\left(q^\ell - 1\right)^{t-r}}.$$

Using the binomial identity, we obtain

$$\mathbb{P}(S_E \neq \{0\}) \leq \frac{q^{\ell(B+1-t)}}{q-1} q^t \left(\frac{q^\ell - \frac{1}{q}}{q^\ell - 1} \right)^t.$$

Using the relation $\ell(B + 1 - t) + t = -(\ell + 1)(t_{\max} - t)$ which results from the definitions of B before Constraint 26, and of $t_{\max}$ in Eq. (14), we conclude the proof:

$$\mathbb{P}_{fail} \leq \mathbb{P}(S_E \neq \{0\}) \leq \left(\frac{q^\ell - \frac{1}{q}}{q^\ell - 1} \right)^t \frac{q^{-(\ell+1)(t_{\max}-t)}}{q-1}.$$

4.2.2 Proof of Theorem 25. Along the same lines of what we have done in Subsection 3.3.3, we fix a maximal error support ξ_r with $t := \#\xi_r$, and we use the law of total probability (see Eq. (9)) to express the failure probability over the error support ξ_r as a weighted sum of the failure probabilities over all the sub error supports $\xi \subseteq \xi_r$ according to the error model ERR1. Since $\mathbb{P}^{ERR2}_{\xi_r}\left(\xi_{R,C} = \xi\right) = \frac{(q^\ell - 1)^{\#\xi}}{q^{\ell t}}$, using the law of total probability of Eq. (9), we obtain

$$\mathbb{P}^{ERR2}_{\xi_r}(\mathcal{F}) \leq \sum_{\xi \subseteq \xi_r} \left(\frac{q^\ell - \frac{1}{q}}{q^\ell - 1} \right)^{\#\xi} \frac{q^{-(\ell+1)(t_{\max} - \#\xi)}}{q-1} \frac{(q^\ell - 1)^{\#\xi}}{q^{\ell t}}.$$

Regrouping all the $\binom{t}{s}$ supports $\xi \subseteq \xi_r$ such that $\#\xi = s$, and using the binomial identity, we conclude the proof:

$$\mathbb{P}^{ERR2}_{\xi_r}(\mathcal{F}) \leq \frac{\left(q^{2\ell+1} - q^\ell + 1\right)^t}{q^{\ell t} q^{(\ell+1)t_{\max}}(q-1)} \leq \frac{q^{-(\ell+1)(t_{\max}-t)}}{q-1}. \quad \square$$

REFERENCES

[1] Matteo Abbondati, Antoine Afflatet, Eleonora Guerrini, and Romain Lebreton. 2023. Probabilistic Analysis of LLL-based Decoder of Interleaved Chinese Remainder Codes. In *ITW 2023-IEEE Information Theory Workshop*.

[2] Divesh Aggarwal and Priyanka Mukhopadhyay. 2018. Improved Algorithms for the Shortest Vector Problem and the Closest Vector Problem in the Infinity Norm. In *29th International Symposium on Algorithms and Computation*.

[3] Daniel Bleichenbacher, Aggelos Kiayias, and Moti Yung. 2003. Decoding of interleaved Reed Solomon codes over noisy data. In *Automata, Languages and Programming: 30th International Colloquium, ICALP 2003 Eindhoven, The Netherlands, June 30–July 4, 2003 Proceedings 30*. Springer, 97–108.

[4] Brice Boyer and Erich L Kaltofen. 2014. Numerical linear system solving with parametric entries by error correction. In *Proceedings of the 2014 Symposium on Symbolic-Numeric Computation*. 33–38.

[5] Stanley Cabay. 1971. Exact solution of linear equations. In *Proceedings of the second ACM symposium on Symbolic and algebraic manipulation*. 392–398.

[6] Oded Goldreich, Dana Ron, and Madhu Sudan. 1999. Chinese remaindering with errors. In *Proceedings of the thirty-first annual ACM symposium on Theory of computing*. 225–234.

[7] Eleonora Guerrini, Kamel Lairedj, Romain Lebreton, and Ilaria Zappatore. 2023. Simultaneous Rational Function Reconstruction with errors: Handling multiplicities and poles. *Journal of Symbolic Computation* 116 (2023), 345–364.

[8] Eleonora Guerrini, Romain Lebreton, and Ilaria Zappatore. 2019. Polynomial linear system solving with errors by simultaneous polynomial reconstruction of interleaved Reed-Solomon codes. In *2019 IEEE International Symposium on Information Theory (ISIT)*. IEEE, 1542–1546.

[9] Eleonora Guerrini, Romain Lebreton, and Ilaria Zappatore. 2021. Polynomial linear system solving with random errors: New bounds and early termination technique. In *Proceedings of the 2021 on International Symposium on Symbolic and Algebraic Computation*. 171–178.

[10] Erich L Kaltofen, Clément Pernet, Arne Storjohann, and Cleveland Waddell. 2017. 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*. 237–244.

[11] Arjen K Lenstra, Hendrik Willem Lenstra, and László Lovász. 1982. Factoring polynomials with rational coefficients. *Mathematische annalen* 261, ARTICLE (1982), 515–534.

[12] Wenhui Li, Vladimir Sidorenko, and Johan SR Nielsen. 2013. On decoding interleaved Chinese remainder codes. In *2013 IEEE International Symposium on Information Theory*. IEEE, 1052–1056.

[13] Michael T McClellan. 1977. The exact solution of linear equations with rational function coefficients. *ACM Transactions on Mathematical Software (TOMS)* 3, 1 (1977), 1–25.

[14] Michael Monagan. 2004. Maximal quotient rational reconstruction: an almost optimal algorithm for rational reconstruction. In *Proceedings of the 2004 international symposium on Symbolic and algebraic computation*. 243–249.

[15] Zach Olesh and Arne Storjohann. 2007. The vector rational function reconstruction problem. In *Computer Algebra 2006: Latest Advances in Symbolic Algorithms*. World Scientific, 137–149.

[16] Clément Pernet. 2014. *High performance and reliable algebraic computing*. Ph. D. Dissertation. Université Joseph Fourier, Grenoble 1.

[17] Irving S Reed and Gustave Solomon. 1960. Polynomial codes over certain finite fields. *Journal of the society for industrial and applied mathematics* 8, 2 (1960), 300–304.

[18] Johan Rosenkilde né Nielsen and Arne Storjohann. 2016. Algorithms for simultaneous Padé approximations. In *Proceedings of the ACM on International Symposium on Symbolic and Algebraic Computation*. 405–412.

[19] Georg Schmidt, Vladimir R Sidorenko, and Martin Bossert. 2009. Collaborative decoding of interleaved Reed–Solomon codes and concatenated code designs. *IEEE Transactions on Information Theory* 55, 7 (2009), 2991–3012.

[20] Claus-Peter Schnorr. 1987. A hierarchy of polynomial time lattice basis reduction algorithms. *Theoretical computer science* 53, 2-3 (1987), 201–224.

[21] Gilles Villard. 1997. *A study of Coppersmith's block Wiedemann algorithm using matrix polynomials*. Technical Report.