Decoding Simultaneous Rational Evaluation Codes
Matteo Abbondati, Eleonora Guerrini, Romain Lebreton

To cite this version:
lirmm-04628184

HAL Id: lirmm-04628184
https://hal-lirmm.ccsd.cnrs.fr/lirmm-04628184
Submitted on 28 Jun 2024

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.
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:

1 INTRODUCTION
The solution of a linear system $A\tilde{x} = \tilde{b}$ with $\ell$ unknowns and with coefficients in an integral domain $R$, can be written as a vector $\tilde{x} = \left(\frac{b_1}{g}, \ldots, \frac{b_\ell}{g}\right)$ 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 $a_1, \ldots, a_n \in \mathbb{F}_q$ (in the case $R = \mathbb{F}_q$) or $n$ distinct prime numbers $p_1 < \ldots < p_n$ (in the case $R = \mathbb{Z}$), a central node delegates the resolution of the reduced systems modulo $(x - a_j)$ (or modulo $p_j$ if $R = \mathbb{Z}$) for $j = 1, \ldots, n$ to $n$ computing nodes. These nodes send the $n$ reductions of the solution to the central node, which can therefore reconstruct the vector $\tilde{x}$ 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 the modular ideal $(p)$. $\prod_{j=1}^n \mathbb{Z}_{p_j}$ will denote the Cartesian product $\mathbb{Z}_{p_1} \times \ldots \times \mathbb{Z}_{p_n}$ while $[x]_p$ will denote the modular element $x \mod p \in \mathbb{Z}_p$.

Definition 1 (Weighted Hamming distance). Let $R^1, R^2 \in (\prod_{j=1}^n \mathbb{Z}_{p_j})^\ell$ 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 $\xi_{R^1, R^2} := \bigcup_{j=1}^n (j : R^1_j \neq R^2_j)$ and their error locator as the product of the primes in the error support $\Lambda_{R^1, R^2} := \prod_{j \in \xi_{R^1, R^2}} p_j$. The weighted Hamming distance between $R^1$ and $R^2$ is defined as $d(R^1, R^2) := \log(\Lambda_{R^1, R^2})$.

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 $p_1 < \ldots < p_n$, a received matrix $R \in (\prod_{j=1}^n \mathbb{Z}_{p_j})^\ell$, an error parameter $d$ and two bounds $F, G$ such that $FG < N/2$, find a reduced vector of fractions $(f_1/g, \ldots, f_\ell/g) \in \mathbb{Q}^\ell$ such that
We can define an error correcting code from the SRNRwE problem. With errors problem is not unique. It turns out that our analysis were first introduced by Pernet in [16, §2.5.2]. We can generalize Theorem 6 to SRN codes: SRNψ(N,F;G) := \left\{ \left[ \begin{array}{c} \ell \\ \end{array} \right]_{1\leq i\leq \ell} : \gcd(f_1, \ldots, f_\ell, g) = 1 \right\}.

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

2.1 Unique decoding and minimal distance

The distance d(C) := \min_{c_1, 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 \( \ell = 1 \), SRNψ(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 \( d(RN(N,F,G)) > \log \left( \frac{N}{2F} \right) \).

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

Lemma 7. We have \( d(SRN\psi(N,F;G)) > \log \left( \frac{N}{2F} \right) \).

Proof. Let \( C_1 = \left[ \begin{array}{c} f_1/g \\ \end{array} \right]_{i,j} \) and \( C_2 = \left[ \begin{array}{c} f'_1/g' \\ \end{array} \right]_{i,j} \) be two code words. For \( j \notin \xi(C_1,C_2) \), \( f_1/g = f'_1/g' \mod p_j \) for all i. We set \( Y := \min \{ |f | g'-g |\} \), and \( 0 < g, g' < G \) we have \( Y < 2F \). Using the relation \( Y = N/\lambda \xi(C_1,C_2) \), we bound \( d(C_1,C_2) = \log(\lambda \xi(C_1,C_2)) = \log(N/Y) > \log(N/2F) \).

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 \( \log(\sqrt{(N/2F)}) = (1/2) \log(N/2F) \) 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 RN 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., $d(c_1, r) = d(c_2, r) = d(c_1, c_2)/2$, 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 $2t = d(C) + \epsilon$ for a small $\epsilon$, then we can construct $r$ such that $d(c_1, r), d(c_2, r) \leq (d(c_1, c_2) + \epsilon)/2 = t$, 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 $L := \{r_j\}_{1 \leq j \leq l}$ be the received matrix. Any word code $C \in SRN_l(N; F, G)$, we can write $R = C + E$ for some error matrix $E$ (which depends on $R$ and $C$). Thanks to the Chinese remainder theorem, we can view each row of the matrix as modular elements in $Z_N$, and the ambient space for the code can be viewed as $Z_N^l$, thus for every $1 \leq i \leq l$ we can write $R_i = C_i + E_i$ with $C_i = \{f_i/g\}_N$ for some $f_i/g$. Letting $\Lambda := \Lambda_{C, R}$, we know [16] that the system of $l$ equations holds:

$$A_i f_i \equiv A g_i \mod N \text{ for } i = 1, \ldots, l$$

(1)

with unknowns $\Lambda, g_i, f_i, \ldots, f_i$. We linearize it thanks to the substitution $\varphi \leftarrow A g_i$ and $\psi_i \leftarrow A f_i$; the resulting equations

$$\psi_i = \varphi R_i \mod N \text{ for } i = 1, \ldots, l$$

(2)

are called the key equations. The solutions $(\varphi, \psi_1, \ldots, \psi_l)$ are vectors in the lattice $L \subseteq Z^{l+1}$ spanned by the rows of the integer matrix

$$L = \text{Span}(1 \quad R_1 \quad \ldots \quad R_l \quad 0 \quad N \quad \ldots \quad 0 \quad \vdots \quad \vdots \quad \vdots \quad \vdots \quad 0 \quad 0 \quad \ldots \quad N).$$

(3)

In particular if $\Lambda \leq 2d^4$ for some distance parameter $d$, the solution vector $v_C := (A g_1 f_1, \ldots, A g_l f_l)$ belongs to the set

$$S_{R,d} := \{(\varphi, \psi_1, \ldots, \psi_l) \in L : 0 < \varphi < 2d^4 G, |\psi_i| < 2d^4 F\}.$$

Note that the condition $\Lambda_{C, R} \leq 2d^4$ means that $C$ is close to $R$ for the weighted Hamming distance. The idea of the decoding is now to compute an element of $S_{R,d}$ and "hope" that it will be a multiple of $v_C$: if this is by divided by all the entries by the first one we can retrieve the correct vector of fractions $(f_1/g, \ldots, f_l/g)$. 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 $S_{R,d}$, the second one is probabilistic, and it is relative to 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 that at our disposal an algorithm $ASV P$ which solves the following problem:

**Problem 8 (β–Approx-SVP)**. Given a basis $\{v_0, \ldots, v_N\}$ of a lattice $L$ and an approximation constant $\beta \geq 1$, find a non-zero vector $w \in L$ such that $|w|_\infty \leq \beta \lambda_\infty(L)$, where $\lambda_\infty(L)$ is the minimum $\|\cdot\|_\infty$-norm of the non-zero vectors in $L$.

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 $ASV P \infty$ satisfies $w_0 \geq 0$ (both $x \omega$ are short vectors). We will also assume that $w$ is $L$-reduced.

**Definition 9**. Given a lattice $L$, a vector $v \in L$ is said to be $L$-reduced if, for $c \in Z \setminus \{0\}$, $(1/c) \cdot v \in L \Rightarrow c = \pm 1$.

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

$$S_{R,d} := \sigma_{F,G}(S_{R,d}) = \{(\varphi, \psi_1, \ldots, \psi_l) \in L : 0 < \varphi < 2d^4 FG, |\psi_i| < 2d^4 FG\}.$$

Therefore, a vector $\psi' \in \tilde{L}$ which satisfies $|\psi'|_\infty < 2d^4 FG$ must belong to $S'_{R,d}$. We can obtain a candidate solution $v_{\tilde{R}}$ by computing a scaled short vector $\tilde{v}_s := ASV P(\tilde{L})$, and unscaling it $v_s := \sigma_{F,G}^{-1}(\tilde{v}_s)$. We can now prove that, provided that $R$ is relatively close to the code (see Constraint 10 below), since $v_s$ is a $\beta$-approximation of the shortest vector, it belongs to a slightly larger solution set.

**Constraint 10**. There exists a code word $C$ such that $\Lambda_{C, R} \leq 2d^4$.

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

**Proof**. We know that $\|\tilde{v}_s\|_\infty \leq \beta \lambda_\infty(\tilde{L}) \leq \beta \|\sigma_{F,G}(v_C)\|_\infty \leq \beta \lambda_{\infty}(C)$. Since we assumed that $(\tilde{v}_s)_0 \geq 0$, we have $\tilde{v}_s \in S'_{\tilde{R}, \tilde{R}}$ and $v_s \in S_{\tilde{R}, \tilde{R}}$. 

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

**Algorithm 1**: SRN codes decoder.

**Input**: $SRN_l(N; F, G)$, received word $R$, distance bound $d$.

**Output**: A code word $C$ s.t. $d(C, R) \leq d$ or "decoding failure".

1. Let $\tilde{L} := \sigma_{F,G}(L)$ be the scaled lattice of $L$ defined in Eq. (3).
2. Compute a short vector $\tilde{v}_s := ASV P(\tilde{L})$.
3. Unscale the vector: $v_s := (\varphi, \psi_1, \ldots, \psi_l) := \sigma_{F,G}^{-1}(\tilde{v}_s)$.
4. Let $\lambda := \gcd(\varphi, \psi_1, \ldots, \psi_l, \psi)' = \varphi/\lambda$ and $\forall j \in \mathbb{Z}$, $\psi_j' := \psi_j/\lambda$.
5. If $\lambda \leq 2d^4$, gcd$(\varphi', N) = 1$, $|\varphi'| < G$ and $\forall j \in \mathbb{Z}$, $|\psi_j'| < F$ then return $(C_1, \ldots, C_l) := (\psi_1'/\varphi', \ldots, \psi_l'/\varphi')$.
6. 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 $ASVP_{\infty}$. 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 $\ell_2$-norm version of the same problem.

Problem 12 ($y = \text{APPROX-SVP}_{\ell_2}(L)$). Given a basis $\{v_0, \ldots, v_\ell\}$ of a lattice $L$ and an approximation constant $\gamma \geq 1$, find a non-zero vector $w \in L$ such that $\|w\|_2 \leq \gamma \lambda_2(L)$, where $\lambda_2(L)$ is the minimum $\|\cdot\|_2$-norm of the non-zero vectors in $L$.

Nevertheless, a $\gamma$-approximation SVP for the $\ell_2$-norm yields a $\gamma \sqrt{\ell + 1}$-approximation SVP for the $\ell_\infty$-norm: If $w = ASVP_{\ell_\infty}(L)$ and $s_2$ (resp. $s_\infty$) is one of the shortest vector for the $\ell_2$-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 $\gamma = \sqrt{2}$ (our lattice has dimension $t + 1$). For this reason, we can always assume to employ a sub-routine $ASVP_{\ell_\infty}$ which solves Problem 8 with $\beta \leq \sqrt{2}$ instead.

The most efficient $y$-Approx-SVP$_{\ell_\infty}$ solver is given by the BKKZ algorithm [20]. It finds a solution of Problem 12 with $y = (1 + \epsilon)\ell + 1$ in polynomial time of degree increasing as $\epsilon \rightarrow 0$. Furthermore, since the output of LLL or BKKZ is always the first vector of a basis of the lattice, the following Lemma will ensure that it is $L$-reduced.

Lemma 13. Let $\{b_1, \ldots, b_n\}$ be a basis of a lattice $L$, then every vector $b_i$ is $L$-reduced.

Proof. If $\frac{1}{\ell}b_j \in L$ for some $c \in \mathbb{Z} \setminus \{0\}$, then we can write $\frac{1}{\ell}b_j = \sum_{j=1}^n a_j b_j$ for some $a_j \in \mathbb{Z}$. Thus, $b_j = \sum_{j=1}^n c_j b_j$, which means that $c_j \ell = 1$, 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 $C$ on input $R$ and parameter $d$, then $C$ is a code word of $SNR(N; F, G)$ such that $d(C, R) \leq d$.

Proof. The output vector $C = (\psi_1' / \psi_0', \ldots, \psi_r' / \psi_0')$ is a code word of $SNR(N; F, G)$ since the algorithm has verified the size conditions $|\psi_0'| < G$, $|\psi_0'| < F$ for all $j$, and that gcd($\psi_0', N) = 1$. Now, we use that $(\phi, \psi_1, \ldots, \psi_r) = (\lambda \psi_0', \lambda \psi_1', \ldots, \lambda \psi_r')$ is in the lattice $L$, so that $\lambda (\psi_0' R_i - \psi_j') = 0 \mod N$ for all $i$. Dividing by the invertible $\psi_0'$ modulo $N$, we obtain $\lambda (R_i - C_j) = 0 \mod N$ for all $i$. For all $j \in \xi \cup \mathcal{C}$, there exists $i$ such that $\phi_j \in (R_i - C_j)$, which implies that $\phi_j \|\lambda$. As a consequence, $A_{C, R} \lambda$. Considering that $\lambda \leq 2^d$, we can conclude that $d(C, R) = \log A_{C, R} \leq \log \lambda \leq d$.

Next lemma shows that, when the algorithm fails, the short vector $v_0$ computed by sub-routine $ASVP_{\ell_\infty}$ is not collinear to $v_C$.

Lemma 15. Assuming Constraint 10, if Algorithm 1 fails, then $v_0 \not\in v_C \mathbb{Z}$.

Proof. We will prove this by contraposition, thus we show that if $v_0 = r v_C$, for some $r \in \mathbb{Z}$ then the algorithm must succeed.

We know that $v_0 = r v_C$ is $L$-reduced therefore $v_C = r v_0$ and $\lambda = \Lambda \leq 2^d$ using Constraint 10 (see Algorithm 1, Step 4 for $\lambda$), $\phi'_0 = r \phi$ $\phi'_j = r \phi_j$ 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 $R$ is $C$ 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 $R$.

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

$$\tilde{e}_j = 0 \text{ if } j \not\in \xi, \quad \tilde{e}_j \sim \mathcal{U} \left(2F \mathbb{Z}_{\xi} \setminus \{0\} \right) \text{ if } j \in \xi$$

(4)

where $\mathcal{U}(S)$ denotes the uniform distribution on any finite set $S$. For any given code word $C$ and error support $\xi$, we obtain the distribution $D_{C, \xi}^{ERR1} := \{R = C + E : E \text{ as in Eq.}(4)\}$ of the random received words $R$ around the central code word $C$. We will need another point of view on the random error matrices $E$. Let $i \in \{1, \ldots, \ell\}$, and denote $E_i \in \mathbb{Z}_N$ the CRT interpolant of the $i$-th row of $E$. Since $p_j | E_i$ for all $i$ and $j$, 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 $\{(F_1, \ldots, F_\ell, A) : \gcd(F_1, \ldots, F_\ell, A) = 1\}$, because if $p | A$, then there is an error modulo $p$, so $\exists i \text{ s.t. } E_i \not\equiv 0 \mod p$ and therefore gcd$(E_1, \ldots, E_\ell, A) = 1$.

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

$$\tilde{e}_j = 0 \text{ if } j \not\in \xi, \quad \tilde{e}_j \sim \mathcal{U} \left(2F \mathbb{Z}_{\xi} \setminus \{0\} \right) \text{ if } j \in \xi, \quad \tilde{e}_j \sim \mathcal{U} \left(2F \mathbb{Z}_{\xi} \setminus \{0\} \right) \text{ if } j \in \xi$$

(5)

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

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

$$d_{\max} := \frac{\ell}{\ell + 1} \log(N/(2FG)) - \log(3\beta).$$

(6)
Remark 16. Our setting allows decoding up to a distance $d \leq d_{\text{max}}$ that, for $t > 1$, can be greater than our estimation $\log \left( \frac{N}{2PF} \right)$ of the unique decoding capability of $\text{SRN}(N; F, G)$ codes.

When fixing the decoding bound $d$ close to $d_{\text{max}}$, 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 $\mathcal{N} \mathcal{S} \mathcal{N} \mathcal{P} \mathcal{F}_\infty (L)$ 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 $R \in \mathcal{D}_{\text{ERR1}}$, for some code word $C \in \text{SRN}(N; F, G)$ and error support $\xi$ such that $\log \Lambda \leq d \leq d_{\text{max}}$, and distance parameter $d$, outputs the center code word $C$ of the distribution $\mathcal{D}_{\text{ERR1}} C_d$, with a probability of failure $P_{\text{fail}} \leq 2^{-d/2\log \Lambda} \exp(n/p_1^2)$

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 $R \in \mathcal{D}_{\text{ERR2}}$, for some code word $C \in \text{SRN}(N; F, G)$ and error support $\xi$ such that $\log \Lambda \leq d \leq d_{\text{max}}$, and distance parameter $d$, outputs the center code word $C$ of the distribution $\mathcal{D}_{\text{ERR2}} C_d$, with a probability of failure $P_{\text{fail}} \leq 2^{-d/2\log \Lambda} \exp(n/p_1^2)$.

This failure probability bound improves the one of decoding interleaved Chinese remainder codes $P_{\text{fail}} \leq 2^{-d/2\log \Lambda} \exp(n/p_1^2)$ which was only available in the special case of non-negative (0 ≤ $f_i$) integer code words ($G = 2$) [1, Theorem 3.5].

3.3 Analysis of the decoding error probability

For any $R \in \mathcal{D}_{\text{ERR1}}$ (as in Theorem 17), Constraint 10 is satisfied. Thus, thanks to Lemma 11, we can assume that $v_i \in S_G = S_{R, t}$ where $t = d + \log(\beta)$.

3.3.1 $P_{\text{fail}}$ under ERR1. If Algorithm 1 fails, then $v_i \notin S_C \subset \mathbb{Z}$ (see Lemma 15). Note that the converse is not necessary true, for example if there exists another close code word $C' \neq C$ with $d(C', R) \leq d$ and if the SVF solver outputs $v_i = C'$. Nevertheless, we can upper bound the failure probability of the algorithm as $P_{\text{fail}} \leq P(S_R \notin S_C)$. We introduce some notations: for $C \in \mathbb{Z}_{\geq 0}$, we let $\mathbb{Z}_{m, C} := \{ a \in \mathbb{Z} : |a \text{ mod } m| \leq C \}$, where a mod m is the central remainder of a modulo m, that is the unique representative of a modulo m within the interval $[-|m/2| + 1, |m/2|]$. Note that this set has cardinality $|\mathbb{Z}_{m, C}| = |\mathbb{Z}| + 1$. Let $S_R$ be the set $S_R := \{ v_i \in S_C : \text{gcd}(v_i, C) = 1 \}$, and $S_G$ be the set $S_G := \{ v_i \in S_G : (\varphi/v_i) \in \mathbb{Z} \}$. As we need a new constraint to prove the following lemma.

Constraint 19. Algorithm 1 parameters satisfy $\frac{2^{n+t+1}}{N} < 1$.

Lemma 20. If Constraint 19 is satisfied, $S_C = \{ 0 \} \Rightarrow S_R \subset \subset C \subset \mathbb{Z}$.

Proof. Let $\varphi, \psi_1, \ldots, \psi_t \in S_R$. We know that for all $1 \leq i \leq t$, $g\varphi E_i = \text{gcd} \left( R_i - \frac{\varphi}{\psi_i} \right) = g\psi_i - f_i \varphi$ mod N. Since $Y|E_i$ and $Y|N$, thanks to the above, we have that $Y|(g\varphi_i - f_i \varphi)$, and we define the integer $\varphi_i = g\psi_i - f_i \varphi$. Dividing the above modular equation by $Y$ we obtain $g\varphi E'_i = \varphi_i \mod \Lambda$. Therefore $|g\varphi E'_i | \leq |\varphi_i | \leq \frac{|\varphi_i | + |f_i \varphi |}{Y} < \frac{2^{n+t+1}FG}{N} \Lambda$ which means that $\varphi \in S_E$, thus thanks to the hypothesis $S_E = \{ 0 \}$, we get $\Lambda | \varphi$, thus $\varphi'_i = 0 \mod \Lambda$. Thanks to Constraint 19 and the above inequality we can conclude that $|\varphi'_i | < \Lambda$, therefore $\varphi'_i = 0$ in $\mathbb{Z}$. Which means that

$$\forall i = 1, \ldots, t, g\psi_i = f_i \varphi.$$ (7)

Since $\text{gcd}(f_1, \ldots, f_t, g) = 1$, Equations (7) imply that $g|\varphi$. We have already seen that $\Lambda \varphi$, so $g\varphi \varphi$ because $\varphi$ and $\Lambda$ are coprime. Plugging $\varphi = g\lambda \Lambda$ for some $\alpha \in \mathbb{Z}$ into Equations (7), we deduce $g\psi_i = f_i \varphi = f_i g\lambda \Lambda$, so $g\psi_i = a_i f_i$ for all $i$. We have shown $(g, \psi_1, \ldots, \psi_t) = (a_i g, a_i f_i, \ldots, a_i f_t) \in \mathbb{Z}$.

Thanks to the above lemma we can upper bound the failure probability of Algorithm 1 with $P_{\text{fail}} = P(S_C \neq \{ 0 \})$. In order to estimate the above, we need the following preliminary result:

Lemma 21. If $\varphi \in \mathbb{Z}$ is such that $\text{gcd}(\varphi, \Lambda) = 1$, then for the probability distribution of error model ERR1, we have

$$P(\forall i, g\varphi E'_i \in \mathbb{Z} \setminus \Lambda) \leq \left( \prod_{p \in \mathbb{P}(n)} \left( \frac{1}{\varphi_i} \right) \right)^{t}$$

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

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

Proof. Since $\text{gcd}(g, N) = 1$, the distributions of the vectors $(g\varphi E'_i, \ldots, g\varphi E_t)$ and $(g\varphi E'_1, \ldots, g\varphi E'_t)$ are identical. Thus, we have

$$P(\forall i, g\varphi E'_i \in \mathbb{Z} \setminus \Lambda) = P(\forall i, g\varphi E'_i \in \mathbb{Z} \setminus \Lambda) \text{ is zero.}$$

We have seen that our probability $P(\forall i, g\varphi E'_i \in \mathbb{Z} \setminus \Lambda)$ is equal to $P((E_1, \ldots, E_t) = (0, \ldots, 0))$. Now, the condition $(\varphi/v) \in \mathbb{Z} \setminus \Lambda/v$ only depends on the columns $(E_j)$ of the random matrix for $j \in \mathbb{Z} \setminus \Lambda/v := \{ j : p_j \in \mathbb{P}(\Lambda/v) \}$. These columns are uniformly distributed in the sample space $\Omega = \{(E_j)_{j \in \mathbb{Z} \setminus \Lambda/v} : \forall j \in \mathbb{Z} \setminus \Lambda/v, E_j \neq 0 \in \mathbb{Z} \}$. Therefore, if we write the condition as

$$\mathcal{E} := \{(E_j)_{j \in \mathbb{Z} \setminus \Lambda/v} : \forall i \in \mathbb{Z} \setminus \Lambda/v, g\varphi E'_i \in \mathbb{Z} \}$$

we can deduce that our probability equals

$$P(\forall i \in \mathbb{Z} \setminus \Lambda/v, g\varphi E'_i \in \mathbb{Z} \setminus \Lambda/v) = \frac{\#(\Omega \cap \mathcal{E})}{\#(\Omega)} \leq \frac{\# \mathcal{E}}{\#(\Omega)}.$$ Note that $\# \mathcal{E} = \prod_{p \in \mathbb{P}(n)} (p - 1)$. When the $(E_j)_{j \in \mathbb{Z} \setminus \Lambda/v}$ are independent and uniformly distributed in $\prod_{j \in \mathbb{Z} \setminus \Lambda/v} (\mathbb{Z} \setminus \Lambda/v)$, as it is the case in $\mathcal{E}$, the random variables $E'_i$ are uniformly distributed in...
where the first inequality above is true since \( \phi/v \) is coprime to \( \Lambda/v \), the multiplication by \( \phi/v \) is a bijection of \( \mathbb{Z}_{\Lambda/v} \). Therefore, the cardinality of \( \mathcal{E} \) is 
\[ |\mathcal{E}| = (\# \mathbb{Z}_{\Lambda/v,B/v})^f. \]

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

**Lemma 22.** Given \( E'_1, \ldots, E'_v \), distributed according to the error model ERR1, we have that
\[ \Pr(S_E \neq \emptyset) \leq \left( \frac{6^{2FG}}{N} \right)^f \Lambda \exp \left( \frac{n}{p_1^f} \right). \]

**Proof.** Rewriting \( \{ E : S_E \neq \emptyset \} \) as \( \bigcup_{\nu=1}^{\Lambda-1} \{ E : \nu \in S_E \} \), we get
\[ \Pr(S_E \neq \emptyset) \leq \sum_{\nu=1}^{\Lambda-1} \Pr(\nu, \nu E'_\nu \in \mathbb{Z}_{\nu,B}) \]
where \( v = \gcd(\phi, \Lambda) \). Thanks to the second point in Lemma 21, we can restrict the sum only to the elements \( \nu \) such that \( v \leq B \), which in turn allows us to deduce that \( \# \mathbb{Z}_{\nu,B} \leq 2[B/v] + 1 \leq 3B/v \). Since this expression depends only on \( v \), we regroup the \( \nu \) in the sum by \( \nu = \gcd(\phi, \Lambda) \), such that \( \gcd(\phi, \Lambda) = v \), is equal to \( \phi \left( \frac{\Lambda}{v} \right) \) with \( \phi \) being the Euler totient function. Therefore
\[ \sum_{\nu=1}^{\Lambda-1} \frac{\phi \left( \frac{\Lambda}{v} \right) \left( \frac{B}{v} \right) f^f}{\prod_{p \in P(\frac{\Lambda}{v})} (p^f - 1)} \leq \sum_{v=1}^{\Lambda} \frac{\phi \left( \frac{\Lambda}{v} \right) \left( \frac{B}{v} \right) f^f}{\prod_{p \in P(\frac{\Lambda}{v})} (p^f - 1)}. \]

Plugging in the definition of \( B \) we can collect a common term \( \left( \frac{6^{2FG}}{N} \right)^f \). Thus, extending the sum over all the divisors \( v \), we can upper bound the quotient \( \Pr(S_E \neq \emptyset) / \left( \frac{6^{2FG}}{N} \right)^f \) with
\[ \sum_{v=1}^{\Lambda} \frac{\phi \left( \frac{\Lambda}{v} \right)^f}{\prod_{p \in P(\frac{\Lambda}{v})} (p^f - 1)} \sum_{v=1}^{\Lambda} \frac{\phi \left( \frac{\Lambda}{v} \right)^f}{\prod_{p \in P(\frac{\Lambda}{v})} (p^f - 1)} \]
where in the last equality we used the distributive property to express a product of the form \( \prod_{p \in P(\Lambda)} f(p) + 1 \), with \( f \) an arbitrary function, as a sum \( \sum_{v=1}^{\Lambda} \prod_{p \in P(\frac{\Lambda}{v})} f(p) \). Bringing each term to its common denominator, the above product can be rewritten as
\[ \prod_{p \in P(\Lambda)} \frac{p^f + 1}{p^f - 1} = \Lambda \prod_{p \in P(\Lambda)} \frac{p^f + 1}{p^f - 1} = \Lambda \prod_{p \in P(\Lambda)} \frac{1 - \frac{1}{p^f}}{1 - \frac{1}{p^f}} \]
\[ \leq \Lambda \prod_{p \in P(\Lambda)} \left( 1 + \frac{1}{p^f} \right) \leq \Lambda \exp \left( \frac{n}{p_1^f} \right). \]

Where the first inequality above is true since
\[ \frac{1 - \frac{1}{p^f}}{1 - \frac{1}{p^f}} \leq \left( 1 + \frac{1}{p^f} \right) \Leftrightarrow 1 - \frac{1}{p^f} \leq 1 - \frac{1}{p^f} \Rightarrow 2^f \geq f + 1, \]
while for the second one we used that \( p_1 = \min p_j \) to get
\[ \prod_{p \in P(\Lambda)} \left( 1 + \frac{1}{p^f} \right) \leq \left( 1 + \frac{1}{p_1^f} \right)^n \leq \exp \left( \frac{n}{p_1^f} \right). \]

### 3.3.2 Proof of Theorem 17
We start by proving that with \( r = d + \log(\beta) \) 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:
\[ 2^{2FG} N = 2^d \sqrt{\frac{d^2}{\beta^2}} \leq \frac{2 \beta d \max FG}{N} = \frac{2 \beta \max FG}{N} \left( \frac{N}{6FG\beta} \right)^\frac{1}{2} = \left( \frac{2 \beta \max FG}{3N} \right)^\frac{1}{2} \]
We already noticed when defining the SRN\(_t\)\((N; F, G)\) code that \( 2FG < N \). We said in Section 2.3 that we can assume \( \beta \leq \sqrt{2} \sqrt{\frac{1}{2} + 1} \). Since \( \sqrt{2} \sqrt{1/2 + 1} \leq \frac{3}{2} \) for every \( \ell \in \mathbb{Z}_{\geq 0} \), 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 \( 2^d \geq \beta d^2 \), and using \( \max(\ell + 1) = (N/(6FG\beta))^f \), we have proved Theorem 17.

### 3.3.3 Proof of Theorem 18
In the second error model, we need to make a distinction between the maximal error support \( \xi \) (over which there are uniform random errors) and the actual error support \( \xi \), which is included in \( \xi \), but may be different if a zero column is drawn. We will denote \( \mathbb{P}_{ERR2}(\xi) \) (resp. \( \mathbb{P}_{ERR2}(\xi) \)) the probability function under the error model 2 with error support \( \xi \) (resp. the error model 1 with \( \xi \)). Let \( F \) be the event of decoding failure, i.e. the set of random matrices \( E \) such that Algorithm 1 returns "decoding failure". Using the law of total probability, we have
\[ \mathbb{P}_{ERR2}(\xi) = \sum_{\xi \in \xi} \mathbb{P}_{ERR2}(\xi | \xi_E = \xi) \mathbb{P}_{ERR2}(\xi_E = \xi). \]
where \( \xi_E := \xi_{RC} \) (see Definition 1). The conditional probabilities \( \mathbb{P}_{ERR2}(\xi | \xi_E = \xi) \) are equal to \( \mathbb{P}_{ERR}(\xi) \), which are upper bounded within the proof of Lemma 22 by
\[ \mathbb{P}_{ERR}(\xi) \leq \left( \frac{6^{2FG}}{N} \right)^f \Lambda \prod_{p \in P(\Lambda)} \left( \frac{p^f - \frac{1}{p}}{p^f - 1} \right) \]
where \( \Lambda = \prod_{j \in \xi} p_j \). Moreover,\n\[ \mathbb{P}_{ERR2}(\xi_E = \xi) = \prod_{j \in \xi} \left( p_j^f - 1 \right) \mathbb{P}_{ERR2}(\xi_E = \xi) = \frac{\prod_{j \in \xi} p_j^f - 1}{N^f} \Lambda \]
where \( \Lambda = \prod_{j \in \xi} p_j \). 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 \( \mathbb{P}_{ERR2}(\xi_E = \xi) \leq \mathbb{P}_{ERR2}(\xi_E = \xi) \leq \mathbb{P}_{ERR2}(\xi_E = \xi) \leq \frac{\prod_{j \in \xi} p_j^f - 1}{N^f} \Lambda \)
\[ = \frac{1}{N \Lambda} \sum_{\xi \in \Xi} \prod_{j \in \xi} \left( p_j^f - 1 \right) = \frac{1}{N \Lambda} \prod_{j \in \xi} \left( p_j^f - 1 \right) + 1 = \Lambda. \]
Now, thanks to the hypothesis of the theorem, we know that \( \Lambda_r \leq 2^d \), and since \( 2^d = 2^d \beta \), we can write
\[
\xi_{ERR2}(F) \leq \left( \frac{2^d F G}{N} \right)^\ell \Lambda_r \leq \left( \frac{2^d \beta F G}{N} \right)^\ell 2^d = 2^{d(\ell+1)} \left( \frac{\beta F G}{N} \right)^\ell.
\]
Using \( 2^{d_{\max}(\ell+1)} = (N/(6 F G \beta))^\ell \), we have proved Theorem 18. \( \square \)

4 THE RATIONAL FUNCTION CASE

In this section we show how the previous analysis fits the decoding of simultaneous rational functions coding, 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 \( M(x) := \prod_{i=1}^n \left( x - \alpha_i \right) \in \mathbb{F}_q[x] \), two degree bounds \( d, d' \in \mathbb{Z}_{\geq 0} \) such that \( d + d' \leq n + 1 \) and a parameter \( \ell > 0 \), we define the simultaneous rational function code as the set of matrices
\[
S_{RF}(M; d, d') := \left\{ \left( \frac{\partial(f_i)}{\partial(g)} \right)_{1 \leq i \leq n} : \gcd(f_i, \ldots, f_i, g) = 1 \right\}.
\]

Recall that we denote \( \partial(p) \) the degree of the polynomial \( p \in \mathbb{F}_q[x] \). Let \( R := (r_{ij})_{1 \leq i \leq \ell} \) be the received matrix. For any code word \( C \in S_{RF}(M; d, d') \), we can write \( R = C + E \) for some error matrix \( E \). We can associate an interpolation polynomial to every row, which we write \( R_i = C_i + E_i \). We set \( \xi_{RC} := \{ i : \max(r_{ij}) \neq \max(c_{ij}) \}, d(R, C) = \#_{\xi_{RC}} \) and \( \Lambda_{RC} = \prod_{i \in \xi_{RC}} (x - \alpha_i) \). We refer to \( \Lambda \) and \( \xi \) instead of \( \Lambda_{RC} \) and \( \xi_{RC} \) for short. We know that \( \Lambda_f = \Lambda Q_i \mod M(x) \) holds for any \( 1 \leq i \leq \ell \) [16]. Making the substitutions \( \psi = \Lambda Q, \psi_i = \Lambda f_i \) we linearize the previous equations, obtaining the key equations
\[
\psi_i = \psi_{R_i} \mod M(x) \quad \text{for} \quad i = 1, \ldots, \ell
\]
which are \( \mathbb{F}_q \)-linear. In particular if \( \partial(\Lambda) \leq t \) for some parameter \( t \), the solution vector \( v_C := (\Lambda Q, \Lambda f_1, \ldots, \Lambda f_\ell) \) belongs to the \( \mathbb{F}_q \)-linear subspace
\[
S_R := \left\{ (\psi, \psi_1, \ldots, \psi_\ell) \in \mathbb{F}_q[x]^{\ell+1} : \psi_i = \psi_{R_i} \mod M(x), \partial(\psi) < d_f + t, \partial(\psi_i) < d_f + t \right\}.
\]
In this context the decoding is a linear problem. Indeed we know that \( v_C \in S_R \), and we can compute an element \( (\psi, \psi_1, \ldots, \psi_\ell) \in S_R \) by solving the linear equations \( \psi_i = \psi_{R_i} \mod M(x) \) for the \( t(d_f + t) + d_g + t \) coefficients of the polynomials \( \psi, \psi_1, \ldots, \psi_\ell \). The idea of the algorithm is to find a non-zero element of \( (\psi, \psi_1, \ldots, \psi_\ell) \in S_R \) by solving the linear equations \( \psi_i = \psi_{R_i} \mod M(x) \) for the \( t(d_f + t) + d_g + t \) coefficients of the polynomials \( \psi, \psi_1, \ldots, \psi_\ell \). The idea of the algorithm is to find a non-zero element of \( (\psi, \psi_1, \ldots, \psi_\ell) \in S_R \) by solving the linear equations \( \psi_i = \psi_{R_i} \mod M(x) \) for the \( t(d_f + t) + d_g + t \) coefficients of the polynomials \( \psi, \psi_1, \ldots, \psi_\ell \). The idea of the algorithm is to find a non-zero element of \( (\psi, \psi_1, \ldots, \psi_\ell) \in S_R \) by solving the linear equations \( \psi_i = \psi_{R_i} \mod M(x) \) for the \( t(d_f + t) + d_g + t \) coefficients of the polynomials \( \psi, \psi_1, \ldots, \psi_\ell \). The idea of the algorithm is to find a non-zero element of \( (\psi, \psi_1, \ldots, \psi_\ell) \in S_R \) by solving the linear equations \( \psi_i = \psi_{R_i} \mod M(x) \) for the \( t(d_f + t) + d_g + t \) coefficients of the polynomials \( \psi, \psi_1, \ldots, \psi_\ell \). The idea of the algorithm is to find a non-zero element of \( (\psi, \psi_1, \ldots, \psi_\ell) \in S_R \) by solving the linear equations \( \psi_i = \psi_{R_i} \mod M(x) \) for the \( t(d_f + t) + d_g + t \) coefficients of the polynomials \( \psi, \psi_1, \ldots, \psi_\ell \). The idea of the algorithm is to find a non-zero element of \( (\psi, \psi_1, \ldots, \psi_\ell) \in S_R \) by solving the linear equations \( \psi_i = \psi_{R_i} \mod M(x) \) for the \( t(d_f + t) + d_g + t \) coefficients of the polynomials \( \psi, \psi_1, \ldots, \psi_\ell \).

Error Model 1. Let \( \xi \) be a fixed error support, and let the columns of the error matrix \( E \) be distributed as follows
\[
\tilde{e}_j = \tilde{0} \text{ if } j \notin \xi, \quad \tilde{e}_j \sim \mathcal{U}\left( \mathbb{F}_q \setminus \{ \tilde{0} \} \right) \text{ if } j \in \xi.
\]
We let $B = df + d_n + 2t - n - 2$ and for $m \in \mathbb{Z}_q[x]$ and $C \in \mathbb{Z}_{>0}$ such that $C \leq \partial(m)$, we introduce the sets

$$F_q[x]_{m,C} := \left\{ p(x) \in F_q[x]_m : \partial(p \mod m) \leq C \right\}, \quad \#F_q[x]_{m,C} = q^{C+1}$$

and whose degree is bounded by $\partial(p \mod m) \leq (df + d_n + 2t - n - 2) - (n - t)$ which is less than $t$ thanks to Constraint 26. Thus $\partial(p \mod m) \leq t = \partial(\Lambda)$ and we can conclude that $q^i \equiv f_i \mod p$ for all $i$. Since gcd$\left(f_1, \ldots, f_t\right) = 1$, as in the proof of Lemma 20 we conclude that $(\varphi, \psi_1, \ldots, \psi_t) \in (\Lambda, \Lambda_f_1, \ldots, \Lambda_f_t) F_q[x]$.

Thanks to the above lemma we can upper bound the failure probability of the decoding algorithm as $P_{\text{fail}} \leq P(S_E \neq 0)$. A standard argument of probability shows that $\#F_q[x]_{\Lambda} = \sum_{m \geq 1} \#F_q[x]_m \geq 1 + (q - 1)^2 P(S_E \neq 0)$, because $\#F_q[x]_m = 1$ and, for $2 \leq m \leq p$, $\#F_q[x]_m \geq 1$. Using the expression $\#F_q[x]_m = \prod_{\varphi \in \mathbb{Z}_q[x], \# \varphi \leq \partial(\varphi), \partial(\varphi) \leq \partial(\Lambda)} \varphi$ we can write

$$P(S_E \neq 0) \leq \frac{1}{q - 1} \sum_{\varphi \in \mathbb{Z}_q[x], \# \varphi \leq \partial(\Lambda)} \varphi$$

where $\#F_q[x]_m \geq 1$ for all $m$. Therefore, if we write the probability

$$P(S_E \neq 0) \leq \frac{1}{q - 1} \sum_{\varphi \in \mathbb{Z}_q[x], \# \varphi \leq \partial(\varphi), \partial(\varphi) \leq \partial(\Lambda)} \varphi$$

and $\partial(\varphi) \leq \partial(\Lambda)$, we can conclude that $q^i \equiv f_i \mod p$ for all $i$. Since gcd$\left(f_1, \ldots, f_t\right) = 1$, as in the proof of Lemma 20 we conclude that $(\varphi, \psi_1, \ldots, \psi_t) \in (\Lambda, \Lambda_f_1, \ldots, \Lambda_f_t) F_q[x]$.

Thanks to the above lemma we can upper bound the failure probability of the decoding algorithm as $P_{\text{fail}} \leq P(S_E \neq 0)$. A standard argument of probability shows that $\#F_q[x]_{\Lambda} = \sum_{m \geq 1} \#F_q[x]_m \geq 1 + (q - 1)^2 P(S_E \neq 0)$, because $\#F_q[x]_m = 1$ and, for $2 \leq m \leq p$, $\#F_q[x]_m \geq 1$. Using the expression $\#F_q[x]_m = \prod_{\varphi \in \mathbb{Z}_q[x], \# \varphi \leq \partial(\varphi), \partial(\varphi) \leq \partial(\Lambda)} \varphi$ we can write

$$P(S_E \neq 0) \leq \frac{1}{q - 1} \sum_{\varphi \in \mathbb{Z}_q[x], \# \varphi \leq \partial(\Lambda)} \varphi$$

and $\partial(\varphi) \leq \partial(\Lambda)$, we can conclude that $q^i \equiv f_i \mod p$ for all $i$. Since gcd$\left(f_1, \ldots, f_t\right) = 1$, as in the proof of Lemma 20 we conclude that $(\varphi, \psi_1, \ldots, \psi_t) \in (\Lambda, \Lambda_f_1, \ldots, \Lambda_f_t) F_q[x]$.

Thanks to the above lemma we can upper bound the failure probability of the decoding algorithm as $P_{\text{fail}} \leq P(S_E \neq 0)$. A standard argument of probability shows that $\#F_q[x]_{\Lambda} = \sum_{m \geq 1} \#F_q[x]_m \geq 1 + (q - 1)^2 P(S_E \neq 0)$, because $\#F_q[x]_m = 1$ and, for $2 \leq m \leq p$, $\#F_q[x]_m \geq 1$. Using the expression $\#F_q[x]_m = \prod_{\varphi \in \mathbb{Z}_q[x], \# \varphi \leq \partial(\varphi), \partial(\varphi) \leq \partial(\Lambda)} \varphi$ we can write

$$P(S_E \neq 0) \leq \frac{1}{q - 1} \sum_{\varphi \in \mathbb{Z}_q[x], \# \varphi \leq \partial(\Lambda)} \varphi$$

and $\partial(\varphi) \leq \partial(\Lambda)$, we can conclude that $q^i \equiv f_i \mod p$ for all $i$. Since gcd$\left(f_1, \ldots, f_t\right) = 1$, as in the proof of Lemma 20 we conclude that $(\varphi, \psi_1, \ldots, \psi_t) \in (\Lambda, \Lambda_f_1, \ldots, \Lambda_f_t) F_q[x]$.

Thanks to the above lemma we can upper bound the failure probability of the decoding algorithm as $P_{\text{fail}} \leq P(S_E \neq 0)$. A standard argument of probability shows that $\#F_q[x]_{\Lambda} = \sum_{m \geq 1} \#F_q[x]_m \geq 1 + (q - 1)^2 P(S_E \neq 0)$, because $\#F_q[x]_m = 1$ and, for $2 \leq m \leq p$, $\#F_q[x]_m \geq 1$. Using the expression $\#F_q[x]_m = \prod_{\varphi \in \mathbb{Z}_q[x], \# \varphi \leq \partial(\varphi), \partial(\varphi) \leq \partial(\Lambda)} \varphi$ we can write

$$P(S_E \neq 0) \leq \frac{1}{q - 1} \sum_{\varphi \in \mathbb{Z}_q[x], \# \varphi \leq \partial(\Lambda)} \varphi$$

and $\partial(\varphi) \leq \partial(\Lambda)$, we can conclude that $q^i \equiv f_i \mod p$ for all $i$. Since gcd$\left(f_1, \ldots, f_t\right) = 1$, as in the proof of Lemma 20 we conclude that $(\varphi, \psi_1, \ldots, \psi_t) \in (\Lambda, \Lambda_f_1, \ldots, \Lambda_f_t) F_q[x]$.

Thanks to the above lemma we can upper bound the failure probability of the decoding algorithm as $P_{\text{fail}} \leq P(S_E \neq 0)$. A standard argument of probability shows that $\#F_q[x]_{\Lambda} = \sum_{m \geq 1} \#F_q[x]_m \geq 1 + (q - 1)^2 P(S_E \neq 0)$, because $\#F_q[x]_m = 1$ and, for $2 \leq m \leq p$, $\#F_q[x]_m \geq 1$. Using the expression $\#F_q[x]_m = \prod_{\varphi \in \mathbb{Z}_q[x], \# \varphi \leq \partial(\varphi), \partial(\varphi) \leq \partial(\Lambda)} \varphi$ we can write

$$P(S_E \neq 0) \leq \frac{1}{q - 1} \sum_{\varphi \in \mathbb{Z}_q[x], \# \varphi \leq \partial(\Lambda)} \varphi$$

and $\partial(\varphi) \leq \partial(\Lambda)$, we can conclude that $q^i \equiv f_i \mod p$ for all $i$. Since gcd$\left(f_1, \ldots, f_t\right) = 1$, as in the proof of Lemma 20 we conclude that $(\varphi, \psi_1, \ldots, \psi_t) \in (\Lambda, \Lambda_f_1, \ldots, \Lambda_f_t) F_q[x]$.
REFERENCES


