List decoding of number field codes

Nicholas Coxon

Designs, Codes and Cryptography manuscript No.
(will be inserted by the editor)

List decoding of number field codes

Nicholas Coxon

Received: date / Accepted: date

Abstract This paper presents a list decoding algorithm for the number field codes of Guruswami [12]. The algorithm is an implementation of the unified framework for list decoding of algebraic codes of Guruswami, Sahai and Sudan [14], specialised for number field codes. The computational complexity of the algorithm is evaluated in terms of the size of the inputs and field invariants.

Keywords Number field codes · Chinese remainder codes · List decoding

Mathematics Subject Classification (2000) 11T71 · 11H71 · 11Y40

1 Introduction

Error-correcting codes derived from algebraic number fields were first considered by Lenstra [28], and more recently by Guruswami [12]. These codes generalise the construction of Chinese remainder codes (or simply, CRT codes), the number-theoretic analogues of Reed-Solomon codes. Although the two are similar, Guruswami's construction, called *number field codes* (or NF-codes), is less general than the construction given by Lenstra. However, the generalisation of CRT codes to NF-codes provides a clearer analogue of the generalisation to algebraic geometry codes of Reed-Solomon codes.

Currently, all known algorithms for decoding of codes derived from algebraic number fields are limited to CRT codes. For CRT codes, decoding reduces to the following problem: given n relatively prime integers $p_1 < \ldots < p_n$, a vector $(r_1, \ldots, r_n) \in \mathbb{Z}^n$ and an integer $k < n$, find all $m \in \mathbb{Z}$ with $0 \leq m < \prod_{i=1}^{k} p_i$ such that $m \equiv r_i \pmod{p_i}$ for t values of i, $1 \leq i \leq n$. For $t \geq (n+k)/2$, if such a value of m exists, then it is unique and can be found with Mandelbaum's [30] decoding algorithm. For smaller values of t, uniqueness is no longer guaranteed, and the problem

Nicholas Coxon
Department of Mathematics, The University of Queensland, Brisbane, Queensland, 4072
E-mail: ncoxon@maths.uq.edu.au

is referred to as the *list decoding* problem for CRT codes. The first efficient algorithm for list decoding of CRT codes was provided by Goldreich, Ron and Sudan [11]. Their algorithm runs in polynomial time and solves the problem whenever

$$t \geq \left(1 + \frac{2}{k}\right) \cdot \sqrt{2kn \frac{\log p_n}{\log p_1}} + \frac{k+6}{2}.$$

Subsequently, Boneh [3] provided a polynomial time algorithm that solves the problem whenever $t \geq \sqrt{kn \log p_n / \log p_1}$.

A problem common to these algorithms is the decline in their decoding performance whenever $p_1 \ll p_n$. In particular, Mandelbaum's algorithm may cease to run in polynomial time for such parameters (see [11]). Roughly speaking, the problem occurs since the amount of information provided by an integer's residue modulo p_i is determined by the size of p_i. This problem was overcome for unique and list decoding by Guruswami, Sahai and Sudan [14]. For unique decoding, they used the algorithm of Goldreich, Ron and Sudan in a generalised minimum distance style algorithm similar to Forney [9] to correct the natural bias in the contributions of the residues. Consequently, they obtained the first polynomial time algorithm for unique decoding. For list decoding, Guruswami et al. gave an efficient algorithm for solving the more general problem of weighted list decoding of CRT codes. Given positive weights $\beta_1, \ldots, \beta_n$, the *weighted list decoding* problem for CRT codes asks to find all $m \in \mathbb{Z}$ with $0 \leq m < \prod_{i=1}^{k} p_i$ such that $\sum_i \beta_i \geq t$, where the sum is over all i such that $m \equiv r_i \pmod{p_i}$. The (uniform) list decoding problem for CRT codes then corresponds to the case where $\beta_i = 1$, for $1 \leq i \leq n$. By carefully selecting weights for their weighted list decoding algorithm, Guruswami et al. are able to solve the list decoding problem for CRT codes whenever $t \geq \sqrt{k(n+\varepsilon)}$, for all $\varepsilon > 0$, in time polynomial in n, $\log p_n$ and $1/\varepsilon$. This decoding performance essentially matches that of the celebrated list decoding algorithms for Reed-Solomon and algebraic geometry codes [42, 15].

It is natural to ask whether decoding algorithms exist for codes constructed from arbitrary number fields. The construction of NF-codes lies within the general framework of "ideal-based" codes [14, 43]. Thus, the construction lends itself to decoding by an algorithm based on the framework for list decoding of algebraic error-correcting described by Guruswami et al. [14, Appendix A]. This observation is used in this paper to generalise the results of Guruswami et al. on list decoding of CRT codes to number fields.

The paper is organised as follows. In Section 2, relevant background on NF-codes is provided, and notation established. In Section 3, a generic coding bound is used to establish a condition under which decoding of NF-codes is combinatorially feasible. Necessary algorithmic preliminaries are provided in Section 4. In Section 5 and Section 6, an algorithm for decoding of NF-codes is proposed. The algorithm's performance is analysed in Section 7. In Section 8, parameter selection for the decoding algorithm is considered, with attention given to the weighted and uniform list decoding problems. Finally, conclusions given in Section 9 end the paper.

While this paper was in preparation, another paper [6] containing an algorithm for solving polynomial equations over number fields came to the author's attention.

The algorithm presented there shares much in common with ours and leads to an alternative approach to list decoding of NF-codes (see Remark 7.1). In this application, both algorithms yield similar results.

2 Review of number field codes

Introduced by Guruswami [12], NF-codes serve as a natural generalisation of CRT codes to number fields. In this section, their construction is briefly reviewed. For further background and motivation behind the construction of NF-codes, the reader is referred to Guruswami's original description. First, some notation and generalities on number fields are introduced.

Throughout, K denotes an (algebraic) number field of degree d and signature (r_1, r_2). The ring of algebraic integers in K is denoted by $\mathcal{O}_K$ and the discriminant of K by D_K. The field embeddings of K in the field $\mathbb{C}$ are denoted by $\sigma_1, \ldots, \sigma_d$ and assumed to be ordered such that $\sigma_1, \ldots, \sigma_{r_1}$ are the real embeddings of K. The remaining $2r_2 = d - r_1$ complex embeddings are assumed to be ordered such that $\sigma_{r_1+i} = \overline{\sigma_{r_1+r_2+i}}$, for $1 \leq i \leq r_2$. For all $x \in K$, the *(field) norm* of x, denoted $N_K(x)$, is defined as the product $N_K(x) = \prod_{i=1}^{d} \sigma_i(x)$. For any nonzero ideal $\mathfrak{a} \subseteq \mathcal{O}_K$, the quotient $\mathcal{O}_K/\mathfrak{a}$ is finite. The *norm* of a nonzero integral ideal $\mathfrak{a} \subseteq \mathcal{O}_K$, denoted $\mathfrak{Na}$, is defined as $\mathfrak{Na} = |\mathcal{O}_K/\mathfrak{a}|$. For all $x \in \mathcal{O}_K$, the relationship $|N_K(x)| = \mathfrak{N}(x)$ holds, where (x) denotes the principal ideal generated by x in $\mathcal{O}_K$. Given a nonzero integral ideal $\mathfrak{a} \subseteq \mathcal{O}_K$, it follows that $\mathfrak{Na}$ divides $N_K(x)$, for all $x \in \mathfrak{a}$. The reader is referred to the texts of Marcus [31] and Narkiewicz [34] for further background on algebraic number theory.

Given a vector $\mathbf{s} = (s_1, \ldots, s_{r_1+r_2}) \in \mathbb{R}^{r_1} \times \mathbb{C}^{r_2}$, the $\mathbf{s}$-*shifted size* of an element $x \in K$ is defined to be

$$\mathrm{size}_{\mathbf{s}}(x) = \sum_{i=1}^{r_1} |\sigma_i(x) - s_i| + 2 \sum_{i=1}^{r_2} |\sigma_{r_1+i}(x) - s_{r_1+i}|.$$

The $\mathbf{0}$-shifted size of an element $x \in K$ is simply denoted $\mathrm{size}(x)$. With this notion of size, NF-codes are defined as follows:

Definition 2.1 (NF-codes) For pairwise comaximal nonzero ideals $\mathfrak{a}_1, \ldots, \mathfrak{a}_n \subset \mathcal{O}_K$, a positive real number M and a vector $\mathbf{s} \in \mathbb{R}^{r_1} \times \mathbb{C}^{r_2}$, the *NF-code* $\mathcal{C} = \mathcal{C}_K$ based on the number field K with parameters $(n, \mathfrak{a}_1, \ldots, \mathfrak{a}_n; M, \mathbf{s})$ is defined as follows: the message set of $\mathcal{C}$ is $\mathcal{M}_{\mathcal{C}} = \{m \in \mathcal{O}_K \mid \mathrm{size}_{\mathbf{s}}(m) \leq M\}$ and a message $m \in \mathcal{M}_{\mathcal{C}}$ is encoded as the vector $(m + \mathfrak{a}_1, \ldots, m + \mathfrak{a}_n) \in \mathcal{O}_K/\mathfrak{a}_1 \times \ldots \times \mathcal{O}_K/\mathfrak{a}_n$.

It is assumed throughout, without loss of generality, that the ideals $\mathfrak{a}_1, \ldots, \mathfrak{a}_n$ of an NF-code are ordered so that $\mathfrak{Na}_1 \leq \mathfrak{Na}_2 \leq \ldots \leq \mathfrak{Na}_n$.

Let $\mathcal{C} = \mathcal{C}_K$ be an NF-code with parameters $(n, \mathfrak{a}_1, \ldots, \mathfrak{a}_n; M, \mathbf{s})$. To each ideal $\mathfrak{a}_i$, $1 \leq i \leq n$, an indicator function $\chi_i : \mathcal{O}_K \to \{0, 1\}$ is assigned such that $\chi_i(x) = 1$ if and only if $x \in \mathfrak{a}_i$. Then the minimum (Hamming) distance of $\mathcal{C}$, denoted $d(\mathcal{C})$, is defined to be the minimum of the sum $\sum_{i=1}^{n}(1 - \chi_i(x - y))$ over

all pairs of distinct elements $x, y \in \mathcal{M}_{\mathcal{C}}$. To obtain a lower bound on $d(\mathcal{C})$, consider distinct elements $x, y \in \mathcal{M}_{\mathcal{C}}$. Then, on one hand,

$$\prod_{i=1}^{n} \mathfrak{Na}_i^{\chi_i(x-y)} \leq |N_K(x-y)| \leq \frac{1}{d^d} \mathrm{size}(x-y)^d,$$

where the final inequality follows from the observation that $|N_K(z)| \leq (\mathrm{size}(z)/d)^d$ for all $z \in K$, which is obtained by applying the inequality of arithmetic and geometric means (AM–GM). On the other hand, it is readily verified that

$$\mathrm{size}(x-y) \leq \mathrm{size}_{\mathbf{s}}(x) + \mathrm{size}_{\mathbf{s}}(y) \leq 2M.$$

On combining the inequalities it follows that

$$\prod_{i=1}^{n} \mathfrak{Na}_i^{\chi_i(x-y)} \leq (2M/d)^d \,. \tag{2.1}$$

Consequently, as x and y were arbitrary distinct elements of $\mathcal{M}_{\mathcal{C}}$, the lower bound $d(\mathcal{C}) \geq n - k$ holds for any value of $k \leq n$ such that $(2M/d)^d \leq \prod_{i=1}^{k} \mathfrak{Na}_i$.

The rate of an NF-code $\mathcal{C} = \mathcal{C}_K$ with parameters $(n, \mathfrak{a}_1, \ldots, \mathfrak{a}_n; M, \mathbf{s})$ is defined to be the quotient $R(\mathcal{C}) = \log |\mathcal{M}_{\mathcal{C}}| / \sum_{i=1}^{n} \log \mathfrak{Na}_i$ (see [39, Section II]). Guruswami [12, Section E] notes that a standard argument from the geometry of numbers suggests that $|\mathcal{M}_{\mathcal{C}}| \approx (2^{r_1} \pi^{r_2} M^d)/(d! \sqrt{|D_K|})$. However, this estimate cannot be used in general, since the error term may dominate. Instead, Guruswami [12, Proposition 19] uses an averaging argument due to Lenstra [28] to prove the existence of a shift $\mathbf{s} \in \mathbb{R}^{r_1} \times \mathbb{C}^{r_2}$ such that

$$\Big| \{x \in \mathcal{O}_K \mid \mathrm{size}_{\mathbf{s}}(x) \leq M\} \Big| \geq \frac{2^{r_1} \pi^{r_2}}{\sqrt{|D_K|}} \frac{M^d}{d!}. \tag{2.2}$$

For any such shift $\mathbf{s}$, an NF-code $\mathcal{C} = \mathcal{C}_K$ with parameters $(n, \mathfrak{a}_1, \ldots, \mathfrak{a}_n; M; \mathbf{s})$ has rate $R(\mathcal{C})$ satisfying

$$R(\mathcal{C}) \geq \frac{\log(2^{r_1} \pi^{r_2} M^d) - \log d! - \log \sqrt{|D_K|}}{n \log \mathfrak{Na}_n}.$$

The existence proof is nonconstructive, and it remains an open problem as to how to find a vector $\mathbf{s}$ satisfying (2.2).

3 A combinatorial result on list decoding

For a list decoding algorithm to run in polynomial time, it is necessary that the algorithm only returns polynomially many codewords. The classical Johnson bound [20, 21] provides an upper bound on the number of codewords in a binary code at Hamming distance *exactly* e from an arbitrary word. This bound was later generalised by Guruswami and Sudan [16] who derived a "Johnson-type" bound for the number of codewords at distance *at most* e from an arbitrary word in a q-ary code. In addition, Guruswami and Sudan extended their result to provide bounds on the number of

codewords with sufficiently large weighted agreement. The following combinatorial result due to Guruswami [13, Theorem 7.10] provides an analogous bound, which may be applied to polyalphabetic codes such as NF-codes:

Theorem 3.1 *Let $\Sigma_1, \ldots, \Sigma_n$ be finite nonempty sets and $\Sigma \subseteq \Sigma_1 \times \ldots \times \Sigma_n$ be nonempty. Let vectors $(\alpha_1, \ldots, \alpha_n)$ and $(\beta_1, \ldots, \beta_n)$ contain positive real entries. Define $d(\Sigma)_{\boldsymbol{\alpha}}$ to be the minimum of $\sum_{i:x_i \neq y_i} \alpha_i$ over all pairs of distinct vectors $(x_1, \ldots, x_n), (y_1, \ldots, y_n) \in \Sigma$. Then, for any vector $(t_1, \ldots, t_n) \in \Sigma_1 \times \ldots \times \Sigma_n$ and $\ell > 0$, there exist at most ℓ vectors $(x_1, \ldots, x_n) \in \Sigma$ such that*

$$\sum_{i:x_i=t_i} \beta_i \geq \sqrt{\left(\sum_{i=1}^{n} \alpha_i - \left(1 - \frac{1}{\ell}\right) d(\Sigma)_{\boldsymbol{\alpha}}\right) \sum_{i=1}^{n} \frac{\beta_i^2}{\alpha_i}}.$$

Theorem 3.1 is now used to derived a condition under which decoding of NF-codes is combinatorially feasible. In Section 8, the condition is used to evaluate the performance of the algorithm for decoding of NF-codes developed in Section 5.

Corollary 3.1 *Let $\mathcal{C}$ be an NF-code based on a number field K with parameters $(n, \mathfrak{a}_1, \ldots, \mathfrak{a}_n; M, \mathbf{s})$ such that $M > d/2$, and $\beta_1, \ldots, \beta_n$ be positive real numbers. Then given a vector $(r_1 + \mathfrak{a}_1, \ldots, r_n + \mathfrak{a}_n) \in \mathcal{O}_K/\mathfrak{a}_1 \times \cdots \times \mathcal{O}_K/\mathfrak{a}_n$ and any tolerance parameter $\varepsilon > 0$, there are at most polynomially many (in $1/\varepsilon$ and $\sum_{i=1}^{n} \log \mathfrak{Na}_i$) elements $m \in \mathcal{M}_{\mathcal{C}}$ such that*

$$\sum_{i=1}^{n} \chi_i(m - r_i)\beta_i \geq \sqrt{d \log(2M/d) \left(\sum_{i=1}^{n} \frac{\beta_i^2}{\log \mathfrak{Na}_i} + \varepsilon \max_{1 \leq i \leq n} \frac{\beta_i^2}{\log^2 \mathfrak{Na}_i}\right)}. \quad (3.1)$$

Proof Define

$$\ell = \frac{1}{\varepsilon} \left(\frac{\sum_{i=1}^{n} \log \mathfrak{Na}_i}{d \log(2M/d)} - 1\right) \sum_{i=1}^{n} \log \mathfrak{Na}_i.$$

If $\ell < 1$, then an application of the Cauchy-Schwarz inequality shows that the right hand side of (3.1) is greater than $\sum_{i=1}^{n} \beta_i$ and thus (3.1) is never satisfied. Therefore, assume that $\ell \geq 1$. Then it is sufficient to show that there exist at most ℓ elements $m \in \mathcal{M}_{\mathcal{C}}$ such that (3.1) holds.

Define $\Sigma_i = \mathcal{O}_K/\mathfrak{a}_i$, $\alpha_i = \log \mathfrak{Na}_i$ and $t_i = r_i + \mathfrak{a}_i$, for $1 \leq i \leq n$. Then applying Theorem 3.1 with $\Sigma = \mathcal{C}$ implies that there exist at most ℓ elements $m \in \mathcal{M}_{\mathcal{C}}$ such that

$$\sum_{i=1}^{n} \chi_i(m - r_i)\beta_i \geq \sqrt{\left(\sum_{i=1}^{n} \log \mathfrak{Na}_i - \left(1 - \frac{1}{\ell}\right) d(\mathcal{C})_{\boldsymbol{\alpha}}\right) \sum_{i=1}^{n} \frac{\beta_i^2}{\log \mathfrak{Na}_i}}, \quad (3.2)$$

where $d(\mathcal{C})_{\boldsymbol{\alpha}}$ is the minimum value of the sum $\sum_{i=1}^{n} (1 - \chi_i(x - y))\, \alpha_i$ over all pairs of distinct elements $x, y \in \mathcal{M}_{\mathcal{C}}$. As the inequality (2.1) holds for any pair of distinct elements $x, y \in \mathcal{M}_{\mathcal{C}}$, it follows that

$$d(\mathcal{C})_{\boldsymbol{\alpha}} \geq \sum_{i=1}^{n} \log \mathfrak{Na}_i - d \log(2M/d).$$

This lower bound and the assumption that $\ell \geq 1$ imply that the right had side of (3.2) is less than or equal to the right hand side of (3.1). Therefore, any element $m \in \mathcal{M}_{\mathcal{C}}$ that satisfies (3.1) must also satisfy (3.2). Hence, there are at most ℓ elements $m \in \mathcal{M}_{\mathcal{C}}$ such that (3.1) holds. □

4 Algorithmic preliminaries

In this section, mathematical concepts and procedures which are used in the development of the weighted list decoding algorithm are summarised. These include lattices and reduced bases (Section 4.1), and methods for computing products of algebraic integers and ideals in a number field (Section 4.2). First, some notation is introduced.

For a matrix or vector $(a_{i,j}) \in \mathbb{C}^{n \times m}$, define norms $\|(a_{i,j})\|_2 = (\sum_{i,j} |a_{i,j}|^2)^{1/2}$ and $\|(a_{i,j})\|_\infty = \max_{i,j} |a_{i,j}|$. For $f \in \mathbb{C}[x]$, define $\|f\|_p$, for $p = 2, \infty$, to be the respective vector norms of the corresponding $(1 + \deg f)$-dimensional vector having the coefficients of the polynomial as coordinates. Denote by $\lfloor x \rceil := \lfloor x + 1/2 \rfloor$ the integer nearest to $x \in \mathbb{R}$.

4.1 Generalities on lattices

A *lattice* is a pair (L, q), where L is a free $\mathbb{Z}$-module of finite rank and q is a positive definite quadratic form on $L \otimes \mathbb{R}$. If q is clear from the context, then L is referred to as a lattice. In the following, if (L, q) is a lattice, then b denotes the symmetric bilinear form on $L \times L$ defined by $b(x, y) = (q(x + y) - q(x) - q(y))/2$. A lattice is called *integral* if $b(x, y) \in \mathbb{Z}$, for all $x, y \in L$.

There are two important numerical invariants associated with a lattice (L, q): its rank and determinant. The *rank* (or *dimension*) of (L, q) is simply the rank of L as a free $\mathbb{Z}$-module. If $(b_i)_{1 \leq i \leq n}$ is an integral basis of L, then the *determinant* of (L, q) is defined to be $\det(L, q) = (\det Q)^{1/2}$, where $Q = (b(b_i, b_j))_{1 \leq i,j \leq n}$ is a symmetric positive definite matrix called the *Gram matrix* of the basis. If $(b'_i)_{1 \leq i \leq n}$ is another basis of L, then there exists a unimodular matrix $(u_{i,j})_{1 \leq i,j \leq n} \in \mathbb{Z}^{n \times n}$ such that $b'_i = \sum_{j=1}^n u_{i,j} b_j$. The Gram matrix of $(b'_i)_{1 \leq i \leq n}$ is then UQU^t. It follows that $\det(L, q)$ is independent of the choice of basis. If $(b^*_i)_{1 \leq i \leq n}$ is the result of applying Gram-Schmidt orthogonalisation to $(b_i)_{1 \leq i \leq n}$, then $\det(L, q) = \prod_{i=1}^n q(b^*_i)^{1/2}$.

A *sublattice* of (L, q) is a lattice (M, q) such that M is a subgroup of L. If M is a finite index subgroup of L, then the determinant of (M, q) is

$$\det(M, q) = [L : M] \cdot \det(L, q).$$

A nonzero element $y \in L$ with $q(y)$ approximating the minimum of $q(x)$ over all nonzero $x \in L$, denoted $\lambda(L, q)$, may be found by computing an LLL-reduced basis:

Theorem 4.1 *Let $(b_i)_{1 \leq i \leq n}$ be an LLL-reduced basis of a rank n lattice (L, q) and $(b^*_i)_{1 \leq i \leq n}$ be the associated orthogonalised Gram–Schmidt basis. Then*

1. $q(b_1) \leq 2^{n-1} \lambda(L, q)$;
2. $q(b_1) \leq 2^{(n-1)/2} \det(L, q)^{2/n}$;

3. $q(b_i) \leq 2^{i-1} q(b_i^*)$, *for* $1 \leq i \leq n$; *and*
4. $\mu_{i,j} := b(b_i, b_j^*)/b(b_j^*, b_j^*)$ *satisfies* $|\mu_{i,j}| \leq 1/2$, *for* $1 \leq j < i \leq n$.

Proofs of properties 1–3 of Theorem 4.1 are provided by Cohen [4, Theorem 2.6.2]. Property 4 of the theorem is required to be satisfied by the definition of an LLL-reduced basis (see [4, Definition 2.6.1]). Given an integral basis $(b_i)_{1 \leq i \leq n}$ of a rank n subgroup $L \subseteq \mathbb{Z}^n$, the L^2 algorithm [36,35] returns an LLL-reduced basis of the lattice $(L, \|.\|_2^2)$ in $O(n^5(n+\log B)\log B)$ bit operations, where $B = \max_{1 \leq i \leq n} \|b_i\|_2$.

4.2 Multiplication in $\mathcal{O}_K$

In this section, algorithms for multiplying elements and ideals in $\mathcal{O}_K$ are discussed. What follows is described in greater detail by Belabas [2] and the reader is referred there and to references [4, 29] for further details on arithmetic in number fields. In this section and the remainder of the paper, $\mathsf{M}(n)$ denotes a function such that $O(\mathsf{M}(n))$ bit operations are sufficient to multiply two n-bit integers: $\mathsf{M}(n) = n^2$ with classical algorithms and $\mathsf{M}(n) = n \log n \log\log n$ with the Schönhage–Strassen algorithm. A matrix is in Hermite normal form (HNF) if it satisfies the definition given by Cohen [4, Definition 2.4.2]. In particular, a nonsingular matrix $A = (a_{i,j}) \in \mathbb{Z}^{n \times n}$ is in HNF if A is upper triangular; $a_{i,i} > 0$, for $1 \leq i \leq n$; and $0 \leq a_{i,j} < a_{i,i}$, for $1 \leq i < j \leq n$. In the following, it is assumed that the HNF computation of a matrix $A \in \mathbb{Z}^{m \times n}$ is performed with the algorithm of Storjohann [41, Chapter 6] in

$$\tilde{O}\left(mnr^2 \log \|A\|_\infty + mn\mathsf{M}\left(r \log \|A\|_\infty\right)\right)$$

bit operations, where r is the rank of A and $\tilde{O}$ is the "soft-O" notation, which ignores polylogarithmic factors.

Let $\omega_1, \ldots, \omega_d$ be an integral basis of $\mathcal{O}_K$. Then computing the coefficients of the product xy, for $x, y \in \mathcal{O}_K$, with respect to $\omega_1, \ldots, \omega_d$ reduces by linearity to computing the coefficients of each of the products $\omega_i \omega_j$. For $1 \leq i, j, k \leq d$, define $m_{i,j,k} \in \mathbb{Z}$ such that $\omega_i \omega_j = \sum_{k=1}^d m_{i,j,k} \omega_k$. Then $(m_{i,j,k})_{1 \leq i,j,k \leq d}$ is called the *multiplication table* of $\mathcal{O}_K$ with respect to the basis $\omega_1, \ldots, \omega_d$. Given elements $x = \sum_{k=1}^d x_k \omega_k$ and $y = \sum_{k=1}^d y_k \omega_k$ such that the coefficients x_k and y_k are B-bit integers, the product xy is computed using the multiplication table in $O\left(d^3 \mathsf{M}\left(B + \log d + \log \|(m_{i,j,k})\|_\infty\right)\right)$ bit operations, where $\|(m_{i,j,k})\|_\infty := \max_{1 \leq i,j,k \leq d} |m_{i,j,k}|$.

In situations where it is necessary to compute several products involving an element $x \in \mathcal{O}_K$, it may be more efficient to compute the matrix M_x representing multiplication by x: if $x = \sum_{k=1}^d x_k \omega_k$ for integers $x_1, \ldots, x_d$, then $M_x = (M_{k,j})_{1 \leq k,j \leq d}$, where $M_{k,j} = \sum_{i=1}^d x_i m_{i,j,k}$, for $1 \leq k, j \leq d$. If $y = \sum_{k=1}^d y_k \omega_k$ and $xy = \sum_{k=1}^d z_k \omega_k$ for integers $y_1, \ldots, y_d$ and $z_1, \ldots, z_d$, then the column vectors $\mathbf{y} = (y_1, \ldots, y_d)$ and $\mathbf{z} = (z_1, \ldots, z_d)$ satisfy $\mathbf{z} = M_x \mathbf{y}$. Therefore, if the coefficients x_k and y_k are B-bit integers for $1 \leq k \leq d$, then the matrix M_x is computed in $O\left(d^3 \mathsf{M}\left(B + \log d + \log \|(m_{i,j,k})\|_\infty\right)\right)$ bit operations and the product xy (i.e., the vector $M_x \mathbf{y}$) in $O\left(d^2 \mathsf{M}\left(B + \log d + \log \|(m_{i,j,k})\|_\infty\right)\right)$ bit operations.

There are two commonly used representations of an integral ideal $\mathfrak{a} \subseteq \mathcal{O}_K$. The first, is by a matrix $A = (a_{i,j}) \in \mathbb{Z}^{d \times d}$ such that the elements $\sum_{i=1}^{d} a_{i,j}\omega_i \in \mathcal{O}_K$, for $1 \leq j \leq n$, form an integral basis for $\mathfrak{a}$. If A is in HNF, then A is unique and called the HNF of $\mathfrak{a}$ with respect to the basis $\omega_1, \ldots, \omega_d$. The second representation, is by generators $\alpha, \beta \in \mathcal{O}_K$ such that $\mathfrak{a} = (\alpha, \beta)$, which always exists [4, Proposition 4.7.7] and is called a *two-element representation*. Given a two element representation $\mathfrak{a} = (\alpha, \beta)$, the HNF of $\mathfrak{a}$ is computed as the HNF of the matrix $(M_\alpha \mid M_\beta) \in \mathbb{Z}^{d \times 2d}$. Using a matrix representation to compute a two-element representation is more difficult, and is not considered here (instead, see [2, Section 6.3]).

If nonzero integral ideals $\mathfrak{a}, \mathfrak{a}' \subseteq \mathcal{O}_K$ are given by HNF matrices $A, A' \in \mathbb{Z}^{d \times d}$ respectively, then their product $\mathfrak{a}\mathfrak{a}'$ is computed as follows: let $\alpha_1, \ldots, \alpha_d \in \mathcal{O}_K$ (resp. $\alpha'_1, \ldots, \alpha'_d \in \mathcal{O}_K$) be the integral basis described by A (resp. A'), then compute the HNF of the $d \times d^2$ matrix whose column vectors are the coordinate vectors of the products $\alpha_i\alpha'_j$, for $1 \leq i, j, \leq d$. If $\|A\|_\infty$ and $\|A'\|_\infty$ are B-bit integers, then the d^2 products $\alpha_i\alpha'_j$ are computed in $O\left(d^5\mathsf{M}\left(B + \log d + \log \|(m_{i,j,k})\|_\infty\right)\right)$ bit operations and the HNF computation performs

$$\tilde{O}\left(d^5\left(2B + \log \|(m_{i,j,k})\|_\infty\right) + d^3\mathsf{M}\left(d\left(2B + \log \|(m_{i,j,k})\|_\infty\right)\right)\right)$$

bit operations. As the matrix A is in HNF, the product of its diagonal elements (i.e., the determinant of A) is equal to $[\mathcal{O}_K : \mathfrak{a}] = \mathfrak{N}\mathfrak{a}$ (see [4, Proposition 4.7.4]). It follows that $\|A\|_\infty \leq \mathfrak{N}\mathfrak{a}$ and, similarly, that $\|A'\|_\infty \leq \mathfrak{N}\mathfrak{a}'$. Hence, the HNF of the product $\mathfrak{a}\mathfrak{a}'$ is computed in $O(\mathrm{poly}(d, \log \mathfrak{N}\mathfrak{a}, \log \mathfrak{N}\mathfrak{a}', \log \|(m_{i,j,k})\|_\infty))$ bit operations.

5 Weighted list decoding of NF-codes

In this section, a weighted list decoding algorithm for NF-codes is developed. For an NF-code $\mathcal{C} = \mathcal{C}_K$ with parameters $(n, \mathfrak{a}_1, \ldots, \mathfrak{a}_n; M, \mathbf{s})$, weighted list decoding reduces to the following problem: given a vector $(r_1 + \mathfrak{a}_1, \ldots, r_n + \mathfrak{a}_n) \in \mathcal{O}_K/\mathfrak{a}_1 \times \ldots \times \mathcal{O}_K/\mathfrak{a}_n$, called a *received word*, positive real weights $\beta_1, \ldots, \beta_n$ and a real number $t \geq 0$, find all $m \in \mathcal{M}_\mathcal{C}$ such that $\sum_{i=1}^{n} \chi_i(m - r_i)\beta_i \geq t$. List decoding is captured by the special case in which the weights are equal. Currently, no method is known for determining shift parameters $\mathbf{s} \in \mathbb{R}^{r_1} \times \mathbb{C}^{r_2}$ such that (2.2) holds. Consequently, attention is limited in this section to decoding with the shift $\mathbf{s} = \mathbf{0}$.

The decoding algorithm's development is based on implementing the unified framework for list decoding of algebraic error-correcting codes of Guruswami, Sahai and Sudan [14, Appendix A], specialised for NF-codes. A full description and analysis of the framework for list decoding of algebraic error-correcting codes is provided by Guruswami [13, Section 7]. At a high level, the framework suggests the following approach to decoding when given an NF-code $\mathcal{C} = \mathcal{C}_K$ with parameters $(n, \mathfrak{a}_1, \ldots, \mathfrak{a}_n; M, \mathbf{0})$, weights $\beta_1, \ldots, \beta_n$ and a received word $(r_1+\mathfrak{a}_1, \ldots, r_n+\mathfrak{a}_n)$:

1. Define ideals $\mathfrak{A}_1, \ldots, \mathfrak{A}_n \subseteq \mathcal{O}_K[x]$ as follows:

$$\mathfrak{A}_i = (x - r_i)\mathcal{O}_K[x] + \mathfrak{a}_i\mathcal{O}_K[x], \quad \text{for } 1 \leq i \leq n.$$

To each ideal $\mathfrak{A}_i$, assign a positive integer parameter z_i, for $1 \leq i \leq n$.

2. Find a nonzero polynomial $h \in \bigcap_{i=1}^{n} \mathfrak{A}_i^{z_i}$ of degree at most ℓ and with *small coefficients*.
3. Find the roots of h over K, and return all roots $m \in \mathcal{M}_{\mathcal{C}}$ such that the sum $\sum_{i=1}^{n} \chi_i(m - r_i)\beta_i$ is sufficiently large.

The definition of the ideals $\mathfrak{A}_1, \ldots, \mathfrak{A}_n$ and the Chinese remainder theorem imply that any polynomial $h \in \bigcap_{i=1}^{n} \mathfrak{A}_i^{z_i}$ satisfies $h(m) \in \prod_{i=1}^{n} \mathfrak{a}_i^{\chi_i(m-r_i)z_i}$, for all $m \in \mathcal{O}_K$. The requirement that h has "small coefficients" is used to guarantee that all $m \in \mathcal{M}_{\mathcal{C}}$ for which $\prod_{i=1}^{n} \mathfrak{N}\mathfrak{a}_i^{\chi_i(m-r_i)z_i}$ is sufficiently large are not only modular roots of h but are also roots of h over K. Thus, all such $m \in \mathcal{M}_{\mathcal{C}}$ are found in the root finding step. For $K = \mathbb{Q}$, repeating arguments of Howgrave-Graham [19, Section 2] provides the following sufficient condition for the modular root to additionally be an integer root: if $m \in \mathcal{M}_{\mathcal{C}}$ and a polynomial $h \in \bigcap_{i=1}^{n} \mathfrak{A}_i^{z_i}$ of degree at most ℓ satisfy

$$\prod_{i=1}^{n} \mathfrak{N}\mathfrak{a}_i^{\chi_i(m-r_i)z_i} > \sqrt{\ell+1} \cdot \|h(xM)\|_2 \,, \tag{5.1}$$

then $h(m) = 0$. Thus, in the special case where $K = \mathbb{Q}$, the result of Howgrave-Graham provides motivation for the following notion of a polynomial $h \in \bigcap_{i=1}^{n} \mathfrak{A}_i^{z_i}$ with small coefficients: h has small coefficients whenever $\|h(xM)\|_2$ is small. This notion of size is used by Guruswami et al. in their weighted list decoding algorithm for CRT codes, where a lattice-based approach is used to find a small polynomial. To obtain a decoding algorithm for NF-codes based on fields other than $\mathbb{Q}$, the result of Howgrave-Graham and the lattice-based approach to finding a small polynomial are generalised to arbitrary number fields.

The ring of integer $\mathcal{O}_K$ carries a natural lattice structure defined by the positive definite quadratic form $T_2(x) := \mathrm{Tr}_{K_{\mathbb{R}}/\mathbb{R}}(x\bar{x})$ on the $\mathbb{R}$-algebra $K_{\mathbb{R}} := K \otimes_{\mathbb{Q}} \mathbb{R}$, where $\mathrm{Tr}_{K_{\mathbb{R}}/\mathbb{R}}$ denotes the trace of $K_{\mathbb{R}}$ and $\bar{} : K_{\mathbb{R}} \to K_{\mathbb{R}}$ is the involution induced by complex conjugation. For $x \in K$, the quadratic form is given explicitly by the sum $T_2(x) = \sum_{i=1}^{d} |\sigma_i(x)|^2$. Consequently, the determinant $\det(\mathcal{O}_K, T_2) = \sqrt{|D_K|}$. Define the norm associated with T_2 by $\|x\| = \sqrt{T_2(x)}$, for all $x \in K_{\mathbb{R}}$. For polynomials in $\mathcal{O}_K[x]$, the following lemma provides a sufficient condition for a modular root to also be a root over K, which, in addition, provides a notion of a polynomial with "small coefficients" which is compatible with a lattice-based approach:

Lemma 5.1 *Let K be a degree d number field, $h = \sum_{i=0}^{\ell} h_i x^i \in \mathcal{O}_K[x]$ be a polynomial of degree at most ℓ, $\mathfrak{a}$ be a nonzero ideal of $\mathcal{O}_K$ and M be a positive real number. Suppose that*

1. *$h(m) \in \mathfrak{a}$, for some $m \in \mathcal{O}_K$ with $\mathrm{size}(m) \leq M$; and*
2. *$\left(\sum_{i=0}^{\ell} T_2(h_i) M^{2i}\right)^{1/2} < d\,(\mathfrak{N}\mathfrak{a})^{1/d}/\sqrt{\ell+1}$.*

Then $h(m) = 0$ over K.

Proof Applying the AM–GM inequality yields

$$|N_K(h(m))|^{1/d} \leq \frac{1}{d}\mathrm{size}\left(h(m)\right) \leq \frac{1}{d}\sum_{i=0}^{\ell} \mathrm{size}\left(h_i m^i\right).$$

Furthermore, applying the Cauchy–Schwarz inequality shows that

$$\text{size}(h_i m^i) \leq \sqrt{T_2(h_i) \cdot T_2(m^i)} \leq \sqrt{T_2(h_i) \cdot T_2(m)^i} \leq \sqrt{T_2(h_i)} \cdot \text{size}(m)^i,$$

for $0 \leq i \leq \ell$. Thus,

$$|N_K(h(m))| \leq \frac{1}{d^d} \left((\ell + 1) \cdot \sum_{i=0}^{\ell} T_2(h_i) M^{2i} \right)^{\frac{d}{2}} < \mathfrak{N}\mathfrak{a}.$$

However, $\mathfrak{N}\mathfrak{a}$ divides $N_K(h(m))$, since $h(m) \in \mathfrak{a}$. Therefore, if $h(m) \neq 0$, then $\mathfrak{N}\mathfrak{a} \leq |N_K(h(m))| < \mathfrak{N}\mathfrak{a}$, which is absurd. Hence, $h(m) = 0$ over K. □

Lemma 5.1 provides a natural generalisation of condition (5.1): the lemma implies that a polynomial $h \in \bigcap_{i=1}^{n} \mathfrak{A}_i^{z_i}$ of degree at most ℓ will have amongst its roots all $m \in \mathcal{M}_{\mathcal{C}}$ such that

$$\prod_{i=1}^{n} \mathfrak{N}\mathfrak{a}_i^{\chi_i (m - r_i) z_i} > \left(\frac{1}{d} \cdot \sqrt{\ell + 1} \cdot \|h\|_{2,M} \right)^d, \tag{5.2}$$

where $\left\| \sum_i a_i x^i \right\|_{2,M} := \left(\sum_i T_2(a_i) M^{2i} \right)^{1/2}$, for all $\sum_i a_i x^i \in K[x]$ and any positive real number M. Based on this observation, a polynomial $h \in \mathcal{O}_K$ is said to have small coefficients whenever $\|h\|_{2,M}$ is small. With this notion of size, there are three main tasks that arise from the approach to decoding of NF-codes suggested by the general framework: the first, is the selection of the parameters $z_1, \ldots, z_n$ and ℓ; the second, is to find a nonzero polynomial $h \in \bigcap_{i=1}^{n} \mathfrak{A}_i^{z_i}$ of degree at most ℓ such that $\|h\|_{2,M}$ is small; and the third, is to determine the roots of h in $\mathcal{M}_{\mathcal{C}}$. Selection of the parameters $z_1, \ldots, z_n$ and ℓ is considered in Section 8, with the parameters considered as free in the meantime. For the third task, one of several efficient algorithms for factoring polynomials over number fields may be applied (see [26,25] and references therein). However, a specialised algorithm is developed in Section 6, which is used to establish the complexity of the decoding algorithm. For the second task, as suggested above, a lattice-based approach is employed.

For nonnegative $\ell \in \mathbb{Z}$, let $\mathcal{O}_K[x]_\ell \simeq \mathcal{O}_K^{\ell+1}$ (resp. $\mathfrak{A}_\ell$) denote the free $\mathbb{Z}$-module of polynomials of degree at most ℓ in $\mathcal{O}_K[x]$ (resp. $\bigcap_{i=1}^{n} \mathfrak{A}_i^{z_i}$). Then $\mathcal{O}_K[x]_\ell$ together with the quadratic form $\|.\|_{2,M}^2$ forms a lattice. Thus, a polynomial $h \in \mathfrak{A}_\ell$ with small coefficients is found by computing an LLL-reduced basis of the sublattice $(\mathfrak{A}_\ell, \|.\|_{2,M}^2)$. However, the Gram matrix of the sublattice does not contain integral entries in general. As more is known about the complexity of the LLL-algorithm for integral lattices, an approach of Belabas [2, Section 4.2], which uses integral reduction to produce a (not necessarily LLL-reduced) basis containing a short vector, is used in the decoding algorithm.

Let $\omega_1, \ldots, \omega_d$ be an integral basis of $\mathcal{O}_K$ and $R^t R$ be the Cholesky decomposition of the corresponding Gram matrix of the lattice $(\mathcal{O}_K, T_2)$. Then

$$R = \text{diag}(\|\omega_1^*\|, \ldots, \|\omega_d^*\|) \cdot (\mu_{j,i})_{1 \leq i,j \leq d},$$

where $\mu_{i,j}$ is defined as in Theorem 4.1, for $1 \leq j < i \leq d$; $\mu_{i,i} := 1$, for $1 \leq i \leq d$; and $\mu_{i,j} := 0$, for $1 \leq i < j \leq d$. If $x = \sum_{i=1}^{d} x_i \omega_i \in K[x]$ is represented by the column vector $\boldsymbol{\delta}(x) := (x_1, \ldots, x_d) \in \mathbb{Q}^d$, then $T_2(x) = \|R\boldsymbol{\delta}(x)\|_2^2$. Belabas [2, Section 4.2] observes that for $e \in \mathbb{Z}$ such that

$$2^e \cdot \min_{1 \leq i \leq d} \|\omega_i^*\| > 1/2, \tag{5.3}$$

the matrix $R_e := \lfloor 2^e R \rceil$ has maximal rank and $\|R_e \boldsymbol{\delta}(x)\|_2^2$ provides a "convenient" integral approximation to $2^{2e} T_2(x)$, which may be substituted for T_2 whenever reduction is performed on a sublattice of $(\mathcal{O}_K, T_2)$. If the integral basis is LLL-reduced, then property 3 of Theorem 4.1 and the observation that $\lambda(\mathcal{O}_K, T_2) = d$ imply that

$$\rho := \min_{1 \leq i \leq d} \|\omega_i^*\| \geq \min_{1 \leq i \leq d} 2^{(1-i)/2} \|\omega_i\| \geq 2^{(1-d)/2} \sqrt{d}, \tag{5.4}$$

so that e need only satisfy $e > (d - 3 - \log d)/2$. This idea extended for use in the decoding algorithm by defining the block diagonal matrix

$$R_{e,\ell,M} := \operatorname{diag}\left(\lfloor 2^e R \rceil, \lfloor 2^e M R \rceil, \ldots, \lfloor 2^e M^\ell R \rceil\right) \in \mathbb{Z}^{d(\ell+1) \times d(\ell+1)},$$

for nonnegative $\ell \in \mathbb{Z}$, nonnegative $e \in \mathbb{Z}$ such that (5.3) holds, and real $M \geq 1$. To each polynomial $a = \sum_{i=0}^{\ell} a_i x^i \in \mathcal{O}_K[x]_\ell$, assign the column vector $\boldsymbol{\delta}_\ell(a) \in \mathbb{Z}^{d(\ell+1)}$, which is viewed as having $\ell + 1$ blocks, with the ith block equal to $\boldsymbol{\delta}(a_{i-1})$. Then $\left\|R_{e,\ell,M} \boldsymbol{\delta}_\ell(a)\right\|_2^2$ provides an integral approximation of $2^{2e} \|a\|_{2,M}^2$, for all $a \in \mathcal{O}_K[x]_\ell$, which is substituted for $\|.\|_{2,M}^2$ in the decoding algorithm.

In order to perform lattice reduction, an initial basis of $\mathfrak{A}_\ell$ is required. To address this problem for CRT codes (i.e., for $K = \mathbb{Q}$), Guruswami et al. begin by computing a basis (provided by [14, Lemma 2]) for the free $\mathbb{Z}$-module of polynomials of degree at most ℓ in $\mathfrak{A}_i^{z_i}$, for $1 \leq i \leq n$. Then a general method for computing the intersection of two real full-rank lattices [14, Appendix B] is used to compute a basis of the lattice $\boldsymbol{\delta}_\ell(\mathfrak{A}_\ell)$. Applying this approach to general number fields requires performing operations on potentially large matrices: computing the intersection requires $2(n-1)$ inversions of $d(\ell+1) \times d(\ell+1)$ upper triangular matrices and $n-1$ Hermite normal form computations of $d(\ell+1) \times 2d(\ell+1)$ matrices. The following lemma provides an explicit basis of $\mathfrak{A}_\ell$ and a means to circumvent operations on large matrices:

Lemma 5.2 *With notation as above, let $\alpha_{j,1}, \ldots, \alpha_{j,d} \in \mathcal{O}_K$ be an integral basis of the ideal $\prod_{i=1}^{n} \mathfrak{a}_i^{\max\{z_i - j, 0\}}$, for $0 \leq j \leq \ell$. Suppose that $r \in \mathcal{O}_K$ satisfies $r - r_i \in \mathfrak{a}_i$, for $1 \leq i \leq n$. (Such an element exists by the Chinese remainder theorem.) Then the polynomials*

$$u_{dj+k}(x) = x^{\max\{0, j - z_{\max}\}} \alpha_{j,k} (x - r)^{\min\{j, z_{\max}\}}, \quad \text{for } 0 \leq j \leq \ell,\ 1 \leq k \leq d,$$

where $z_{\max} = \max_{1 \leq i \leq n} z_i$, form an integral basis of $\mathfrak{A}_\ell$.

Proof Each polynomial $h \in \mathcal{O}_K[x]_\ell$ may be expressed in the form

$$h(x) = h_{z_{\max}}(x) \cdot (x-r)^{z_{\max}} + \sum_{j=0}^{z_{\max}-1} h_j (x-r)^j, \tag{5.5}$$

such that $h_0, \ldots, h_{z_{\max}-1} \in \mathcal{O}_K$ and $h_{z_{\max}} \in \mathcal{O}_K[x]$. The coefficients $h_0, \ldots, h_{z_{\max}}$ are then uniquely determined. From the definition of r, it follows that

$$\mathfrak{A}_i = ((x-r) + (r-r_i))\,\mathcal{O}_K[x] + \mathfrak{a}_i \mathcal{O}_K[x] = (x-r)\mathcal{O}_K[x] + \mathfrak{a}_i \mathcal{O}_K[x],$$

for $1 \leq i \leq n$. Therefore, if $h \in \mathcal{O}_K[x]_\ell$, when expressed in the form (5.5), has coefficients satisfying $h_j \in \prod_{i=1}^n \mathfrak{a}_i^{\max\{z_i - j, 0\}}$, for $0 \leq j < z_{\max}$, then $h \in \mathfrak{A}_\ell$ since the ideals $\mathfrak{a}_1, \ldots, \mathfrak{a}_n$ are pairwise comaximal. In particular, the polynomials $u_1, \ldots, u_{d(\ell+1)} \in \mathfrak{A}_\ell$. It is shown that the converse also holds. Then it is clear that any polynomial $h \in \mathfrak{A}_\ell$ may be expressed as a $\mathbb{Z}$-linear combination of the polynomials $u_1, \ldots, u_{d(\ell+1)}$. Linear independence of the polynomials follows from that of the bases $(\alpha_{j,k})_{1 \leq k \leq d}$, for $0 \leq j \leq \ell$. Therefore, if the converse holds, then the polynomials $u_1, \ldots, u_{d(\ell+1)}$ form an integral basis of $\mathfrak{A}_\ell$.

To prove the converse, assume now that $h \in \mathfrak{A}_\ell$. Then h may be expressed in the form (5.5) and, since $h \in \mathfrak{A}_1^{z_1}$, may also be expressed in the form

$$h(x) = \lambda_{z_{\max}}(x) \cdot (x-r)^{z_{\max}} + \sum_{j=0}^{z_{\max}-1} \lambda_j(x) \cdot a_j (x-r)^j,$$

where $a_j \in \mathfrak{a}_1^{\max\{z_1 - j, 0\}}$, for $0 \leq j < z_{\max}$, and $\lambda_0, \ldots, \lambda_{z_{\max}} \in \mathcal{O}_K[x]$. Moreover, since

$$(x-r) \cdot a_j (x-r)^j = a_j \cdot (x-r)^{j+1} \quad \text{and} \quad \mathfrak{a}_1^{\max\{z_1 - j, 0\}} \subseteq \mathfrak{a}_1^{\max\{z_1 - j - 1, 0\}},$$

for $0 \leq j < z_{\max}$, it is assumed that $\lambda_0, \ldots, \lambda_{z_{\max}-1} \in \mathcal{O}_K$. However, uniqueness of the coefficients $h_0, \ldots, h_{z_{\max}}$ then implies that $h_j = \lambda_j a_j \in \mathfrak{a}_1^{\max\{z_1 - j, 0\}}$, for $0 \leq j < z_{\max}$. Repeating this argument for $i = 2, \ldots, n$, shows that

$$h_j \in \bigcap_{i=1}^n \mathfrak{a}_i^{\max\{z_i - j, 0\}} = \prod_{i=1}^n \mathfrak{a}_i^{\max\{z_i - j, 0\}}, \quad \text{for } 0 \leq j < z_{\max}.$$

Hence, the converse holds. □

Lemma 5.2 permits an integral basis for $\mathfrak{A}_\ell$ to be computed with operations performed on matrices which have dimensions dependent on d and not on ℓ. The resulting CRT problem may be solved by means of the polynomial time extended Euclidean algorithm for number fields of Cohen [5, Algorithm 1.3.2, Proposition 1.3.7], or the improved variant of Belabas [2, Algorithm 5.4]. In particular, the CRT problem may be solved by the method described by Belabas [2, Algorithm 6.10], which is used with minor modifications in the decoding algorithm.

Based on the above discussion, the following algorithm is proposed for weighted list decoding of NF-codes:

Algorithm 5.1 (Weighted List Decoding of NF-codes)
INPUT: A code $\mathcal{C}$ based on a number field K with parameters $(n, \mathfrak{a}_1, \ldots, \mathfrak{a}_n; M, \mathbf{0})$ such that $M \geq 1$ and the ideal $\mathfrak{a}_i$ is provided by its HNF $A_i \in \mathbb{Z}^{d\times d}$, for $1 \leq i \leq n$; a vector $(r_1, \ldots, r_n) \in \mathcal{O}_K^n$ such that $\|\boldsymbol{\delta}(r_i)\|_\infty \leq \mathfrak{N}\mathfrak{a}_i$, for $1 \leq i \leq n$; an LLL-reduced basis $\omega_1, \ldots, \omega_d$ of the lattice $(\mathcal{O}_K, T_2)$ such that $\omega_1 = 1$, the corresponding multiplication table $(m_{i,j,k})_{1\leq i,j,k\leq d}$ and Cholesky factor R; a nonnegative integer e that satisfies (5.3); and positive integers t, $z_1, \ldots, z_n$ and ℓ.
OUTPUT: A set containing elements $m \in \mathcal{O}_K$ such that $\mathrm{size}(m) \leq M + 2^{-t}$.

1. [Solve the CRT problem] Set $r = 0$. For $j = 1, \ldots, n$, construct an element $r \in \mathcal{O}_K$ such that $r - r_i \in \mathfrak{a}_i$, for $1 \leq i \leq j$, by performing the following steps:
   (a) Use the methods discussed in Section 4.2 to compute the HNF $B_j \in \mathbb{Z}^{d\times d}$ of the ideal $\mathfrak{b}_j := \prod_{i\neq j} \mathfrak{a}_i$.
   (b) Use the extended gcd algorithm for number fields [2, Algorithm 5.4] to find an element $\beta_j \in \mathfrak{b}_j$ such that $1 - \beta_j \in \mathfrak{a}_j$.
   (c) Set $r \leftarrow r + r_j\beta_j \pmod a$ such that $\|\boldsymbol{\delta}(r)\|_\infty < a$, where a is the positive generator of $(\prod_{i=1}^n \mathfrak{a}_i) \cap \mathbb{Z}$. (Note that a is computed as the lcm of the upper left most entries of A_1 and B_1, since $\omega_1 = 1$.)
2. Define $z_{\max} = \max_{1\leq i\leq n} z_i$. For $j = \min\{z_{\max}, \ell\}, \ldots, 0$, use the methods discussed in Section 4.2 to compute the HNF $V_j \in \mathbb{Z}^{d\times d}$ of $\prod_{i=1}^n \mathfrak{a}_i^{\max\{z_i-j,0\}}$. (Note that $V_{z_{\max}} = I_d$.)
3. [Compute an integral basis of $\mathfrak{A}_\ell$] Let $M_r \in \mathbb{Z}^{d\times d}$ be the matrix representing multiplication by r and a_z be the upper left most entry of V_0. Compute the $d(\ell+1) \times d(\ell+1)$ block matrix $U := (U_{s,t})_{1\leq s,t\leq \ell+1}$, with blocks $U_{s,t} \in \mathbb{Z}^{d\times d}$ such that $\|U_{s,t}\|_\infty \leq \prod_{i=1}^n \mathfrak{N}\mathfrak{a}_i^{z_i}$, defined as follows:

$$U_{m+1,j+1} = \begin{cases} (-1)^{j-m}\binom{j}{m} M_r^{j-m} V_j \pmod{a_z} & \text{if } m < j, \\ V_j & \text{if } m = j, \\ 0 & \text{if } m > j, \end{cases}$$

   for $0 \leq m \leq \ell$, $0 \leq j \leq \min\{z_{\max}, \ell\}$; and, if $\ell > z_{\max}$,

$$U_{m+1,j+1} = \begin{cases} 0 & \text{if } m < j - z_{\max} \text{ or } m > j, \\ U_{m-j+z_{\max}+1, z_{\max}+1} & \text{if } j - z_{\max} \leq m \leq j, \end{cases}$$

   for $0 \leq m \leq \ell$, $z_{\max} < j \leq \ell$.
4. [Perform lattice reduction] Compute $Q := R_{e,\ell,M} U$. Use the L^2 algorithm [36] to find an LLL-reduced basis $\mathbf{b}_1, \ldots, \mathbf{b}_{d(\ell+1)}$ of the lattice $(\Lambda, \|.\|_2^2)$ with basis given by the column vectors of Q.
5. [Compute roots of h] Compute $h := \boldsymbol{\delta}_\ell^{-1}((R_{e,\ell,M})^{-1}\mathbf{b}_1)$. If h is constant, then return the empty set and stop. Otherwise, perform Sub-Algorithm 6.1.
6. [Remove spurious roots] Return the set of all $m \in \mathcal{O}_K$ such that $\boldsymbol{\delta}(m)$ belongs to the set $\mathcal{R}$ returned by Sub-Algorithm 6.1 and (5.2) holds.

Remarks 5.1
1. There is little loss of generality resulting from the input requirement on M: if $0 \leq M < d$, then $\mathcal{M}_\mathcal{C} = \{0\}$ and decoding is a greatly simplified problem.

2. A suitable input basis $\omega_1, \ldots, \omega_d$ (i.e., LLL-reduced, with $\omega_1 = 1$) is found with the method described by Belabas [2, Section 4.3]. As noted by Belabas [1, Section 4.2], the entries of R are obtained as by-products of the reduction used to obtain the input basis. It is assumed above (and in the analysis of the decoding algorithm) that the integral basis and R are provided with the precision required by Algorithm 5.1 and Sub-Algorithm 6.1.
3. The computation of the ideal products in Step 1 and Step 2 may be greatly aided by the knowledge of two-element representations for some or all of the ideals $\mathfrak{a}_1, \ldots, \mathfrak{a}_n$ (see [4, Section 4.7.1] or [1, Section 5.3.2]).
4. For an NF-code $\mathcal{C}$ with parameters $(n, \mathfrak{a}_1, \ldots, \mathfrak{a}_n; M, \mathbf{s})$ such that $\mathbf{s} \neq \mathbf{0}$, decoding may be performed as follows: use an algorithm for finding an approximate closest vector in a lattice [17], applied to $(\mathcal{O}_K, T_2)$, to find an element $y \in \mathcal{O}_K$ such that $\mathrm{size}_{\mathbf{s}}(y)$ is small; apply Algorithm 5.1 to the code $\mathcal{C}'$ with parameters $(n, \mathfrak{a}_1, \ldots, \mathfrak{a}_n; M + \mathrm{size}_{\mathbf{s}}(y), \mathbf{0})$ and received word $((r_1 - y) + \mathfrak{a}_1, \ldots, (r_n - y) + \mathfrak{a}_n)$; for each output m' of Algorithm 5.1, return $m = m' + y$ if $m \in \mathcal{M}_{\mathcal{C}}$.

6 The root finding sub-algorithm

In this section, the sub-algorithm used in Step 5 of Algorithm 5.1 to determine the roots of the polynomial h is described. As noted in Section 5, the computation of the roots of h can be performed by applying existing algorithms for factoring over number fields. However, the full factorisation of h is not required in the context of Algorithm 5.1 and only those roots of h in $\mathcal{M}_{\mathcal{C}}$ are of interest. By specialising existing algorithms, namely those based on the norm-based approach proposed by Kronecker [24] (see also [25, 8] and references therein), this problem is addressed more efficiently. Moreover, the specialised algorithm is designed to exploit the fact that h belongs to the special subring $\mathcal{O}_K[x]$ of $K[x]$ and that an integral basis of $\mathcal{O}_K$ is known.

For a polynomial $f \in K[x]$, define $M_f = (m_{i,j})_{1 \leq i,j \leq d}$ by $f\omega_j = \sum_{i=1}^{d} m_{i,j}\omega_i$, for $1 \leq j \leq d$. The *norm* of $f \in K[x]$ is then defined to be the determinant $N_K(f) := \det M_f$. When $f \in \mathcal{O}_K$, this definition is consistent with the previous definition from Section 4.2. For all $f, g \in K[x]$, the definition of matrix multiplication implies that $M_{fg} = M_f M_g$. Consequently, the norm map is multiplicative. It follows immediately that if $a \in K$ is a root of $f \in K[x]$, then $N_K(x - a)$ divides $N_K(f)$ in $\mathbb{Q}[x]$. If $\phi_a(x) \in \mathbb{Q}[x]$ denotes the minimal polynomial of $a \in K$, then

$$N_K(x - a) = \det(xI_d - M_a) = \phi_a(x)^{[K:\mathbb{Q}(a)]}.$$

Therefore, the minimal polynomial of any root of $f \in K[x]$ is an irreducible factor of $N_K(f)$ in $\mathbb{Q}[x]$.

To determine the roots of the polynomial h in Step 5 of Algorithm 5.1, the norm $N_K(h) \in \mathbb{Z}[x]$ is factored into irreducible factors in $\mathbb{Z}[x]$, then the roots (in $\mathbb{C}$) of each monic irreducible factor of degree at most d are computed. Amongst the roots are all roots of h in $\mathcal{M}_{\mathcal{C}}$. The norm $N_K(h)$ can be computed using the modular evaluation-interpolation approach of McClellan [32] (see also [18]). The following

lemma provides an alternative method for computing the norm, which is used to establish the polynomial bit complexity of the root finding sub-algorithm:

Lemma 6.1 *Suppose that the polynomial $h \in \mathcal{O}_K[x]$ from Step 5 of Algorithm 5.1 is non-constant. Then $N_K(h)$ is computed in $O(\mathrm{poly}(d, \log|D_K|, \ell, \log\|h\|_{2,M}))$ bit operations. Furthermore, $\log\|N_K(h)\|_\infty$ is $O(\mathrm{poly}(d, \log|D_K|, \ell, \log\|h\|_{2,M}))$.*

Proof Suppose that $h = \sum_{k=0}^{\lambda} h_k x^k$ is non-constant, with $h_0, \ldots, h_\lambda \in \mathcal{O}_K$ and $h_\lambda \neq 0$. Then the matrix $M_h = \sum_{k=0}^{\lambda} M_{h_k} x^k$, where $\det M_{h_\lambda} = N_K(h_\lambda)$ is nonzero. Thus, $N_K(h)$ and the determinant of the matrix $M_{h_\lambda}^{-1} M_h$ have precisely the same roots. It follows from a result of Gohberg, Lancaster and Rodman [10, Theorem 1.1] that the roots of $\det(M_{h_\lambda}^{-1} M_h)$, and thus $N_K(h)$, are precisely the eigenvalues of the matrix $N_K(h_\lambda)^{-1} C$, where C is the $d\lambda \times d\lambda$ block matrix

$$C = N_K(h_\lambda) \cdot \begin{pmatrix} 0 & I_d & 0 & \ldots & 0 \\ 0 & 0 & I_d & \ldots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \ldots & I_d \\ -M_{h_\lambda}^{-1} M_{h_0} & -M_{h_\lambda}^{-1} M_{h_1} & -M_{h_\lambda}^{-1} M_{h_2} & \ldots & -M_{h_\lambda}^{-1} M_{h_{\lambda-1}} \end{pmatrix}.$$

Let $\chi_C(x)$ denote the characteristic polynomial of C. Then the characteristic polynomial of the matrix $N_K(h_\lambda)^{-1} C$ is $\chi_C(N_K(h_\lambda)x) \cdot N_K(h_\lambda)^{-d\lambda}$. Therefore, the roots of $N_K(h)$ coincide with the roots of the polynomial $\chi_C(N_K(h_\lambda)x) \cdot N_K(h_\lambda)^{1-d\lambda}$. However, the leading coefficient of the latter polynomial is equal to $N_K(h_\lambda)$ and

$$N_K(h)(x) = N_K(h_\lambda) x^{d\lambda} + \text{terms of degree less than } d\lambda.$$

Hence, the two polynomials are in fact equal.

The matrix C has integer entries, thus its characteristic polynomial χ_C is computed in $O(\mathrm{poly}(d, \lambda, \log\|C\|_\infty))$ bit operations (see [7] and references therein). Applying the AM–GM inequality provides the bound

$$|N_K(h_\lambda)| \leq d^{-d/2} T_2(h_\lambda)^{d/2} \leq d^{-d/2} \|h\|_{2,M}^d . \tag{6.1}$$

The Cauchy-Schwarz inequality and the bound on the norm of the adjoint matrix obtained by Richter [38] imply that

$$\left\|M_{h_\lambda}^{\mathrm{adj}} M_{h_k}\right\|_\infty \leq \left\|M_{h_\lambda}^{\mathrm{adj}}\right\|_2 \left\|M_{h_k}\right\|_2 \leq d^{-(d-2)/2} \left\|M_{h_\lambda}\right\|_2^{d-1} \left\|M_{h_k}\right\|_2, \tag{6.2}$$

for $0 \leq k < \lambda$, where $M_{h_\lambda}^{\mathrm{adj}} = N_K(h_\lambda) M_{h_\lambda}^{-1}$ is the adjoint matrix of M_{h_λ}. Finally, since $\omega_1, \ldots, \omega_d$ is an LLL-reduced basis for $(\mathcal{O}_K, T_2)$, Belabas [2, Proposition 5.1] provides the following "pessimistic" bound on the entires of the multiplication table $(m_{i,j,k})_{1 \leq i,j,k \leq d}$:

$$|m_{i,j,k}| \leq 2^{3d(d-1)/4} d^{(1-2d)/2} |D_K| , \quad \text{for } 1 \leq i, j, k \leq d. \tag{6.3}$$

Therefore, combining (6.1), (6.2) and (6.3) implies that the matrix C, its characteristic polynomial χ_C and $N_K(h)$ are each computed in

$$O\left(\text{poly}\left(d, \log|D_K|, \ell, \log\|h\|_{2,M}, \log\max_{0\leq k\leq\lambda}\|\boldsymbol{\delta}(h_k)\|_\infty\right)\right)$$

bit operations. The coefficients of $N_K(h)$ satisfy

$$\|N_K(h)\|_\infty \leq |N_K(h_\lambda)| \cdot \|\chi_C\|_\infty \leq |N_K(h_\lambda)| \cdot \max_{0\leq k\leq d\lambda}\binom{d\lambda}{k} k^{k/2}\|C\|_\infty^k,$$

where the final inequality is obtained by applying a bound of Cohen [4, Proposition 2.2.10]. It follows that $\log\|N_K(h)\|_\infty$ is

$$O\left(\text{poly}\left(d, \log|D_K|, \ell, \log\|h\|_{2,M}, \log\max_{0\leq k\leq\lambda}\|\boldsymbol{\delta}(h_k)\|_\infty\right)\right)$$

Arguments due to Belabas [2, p. 28] are now used to establish the following claim:

$$\|\boldsymbol{\delta}(a)\|_2 \leq 2^{(1-d)/2} d^{d+1/2}\|a\|, \quad \text{for all } a \in \mathcal{O}_K. \tag{6.4}$$

Given the claim, it follows that

$$\|\boldsymbol{\delta}(h_k)\|_\infty \leq \|\boldsymbol{\delta}(h_k)\|_2 \leq 2^{(1-d)/2} d^{d+1/2}\|h_k\| \leq 2^{(1-d)/2} d^{d+1/2}\|h\|_{2,M}, \tag{6.5}$$

for $0 \leq k \leq \lambda$, thus completing the proof of the lemma.

An application of the Cauchy-Schwarz inequality shows that

$$\|\boldsymbol{\delta}(a)\|_2 \leq \left\|R^{-1}\right\|_2 \|R\boldsymbol{\delta}(a)\|_2 = \left\|R^{-1}\right\|_2 \|a\|, \quad \text{for all } a \in \mathcal{O}_K. \tag{6.6}$$

Write $R = D + N$, where $D = \text{diag}(\|\omega_1^*\|, \ldots, \|\omega_d^*\|)$ and N is upper triangular nilpotent. Setting $Z = D^{-1}N$, it follows that

$$R^{-1} = (I_d + Z)^{-1}D^{-1} = \left(\sum_{i=0}^{d-1}(-1)^{i-1}Z^i\right)D^{-1}.$$

A nonzero entry of Z is a Gram-Schmidt coefficient $\mu_{i,j}$, such that $j < i$, of the reduced basis $\omega_1, \ldots, \omega_d$. Therefore, property 4 of Theorem 4.1 implies that $\|Z\|_\infty \leq 1/2$. Consequently,

$$\left\|R^{-1}\right\|_2 \leq d\left\|R^{-1}\right\|_\infty \leq \frac{d}{\rho}\sum_{i=0}^{d-1}\left\|Z^i\right\|_\infty \leq \frac{d}{\rho}\sum_{i=0}^{d-1} d^i\|Z\|_\infty^i \leq \frac{d^{d+1}}{2^{d-1}\rho}, \tag{6.7}$$

where $\rho = \min_{1\leq i\leq d}\|\omega_i^*\|$. Combining (6.6) and (6.7) with the lower bound (5.4) then completes the proof of (6.4). □

The norm $N_K(h) \in \mathbb{Z}[x]$ is factored into irreducible factors in $\mathbb{Z}[x]$ by the algorithm of Lenstra, Lenstra and Lovász [27] in $O(\text{poly}(\deg N_K(h), \log\|N_K(h)\|_\infty))$ bit operations. The roots of the irreducible factors are computed, with guaranteed error terms, by the polynomial time algorithm of Pan [37]:

Theorem 6.1 *Let $\phi \in \mathbb{Z}[x]$ be a degree d polynomial with roots $\alpha_1, \ldots, \alpha_d \in \mathbb{C}$ and t be a positive integer. Then $O(\mathrm{poly}(d, t, \log \|\phi\|_\infty))$ bit operations are sufficient to find numbers $\bar{\alpha}_1, \ldots, \bar{\alpha}_d \in \mathbb{Q} + i\mathbb{Q}$ such that $\max_{1 \le i \le d} |\alpha_i - \bar{\alpha}_i| \le 2^{-t}$.*

If $\phi \in \mathbb{Z}[x]$ is a monic irreducible factor of $N_K(h)$ such that $\deg \phi \le d$, and $a \in \mathbb{C}$ is a root of ϕ, then there is no guarantee that a belongs to $\mathcal{M}_\mathcal{C}$, much less that a is a root of h over K. If $a \in \mathcal{M}_\mathcal{C}$ and $\phi_a \in \mathbb{Z}[x]$ is its minimal polynomial, then $\prod_{i=1}^{d}(x - \sigma_i(a)) = \phi_a(x)^{d/\deg \phi_a}$, where the conjugates $\sigma_1(a), \ldots, \sigma_d(a)$ satisfy $|\sigma_i(a)| \le \mathrm{size}(a)$, for $1 \le i \le d$. By expressing the coefficients of ϕ_a as symmetric polynomials in its roots, it follows that

$$\|\phi_a\|_\infty \le \max_{0 \le j \le \deg \phi_a} \left\lfloor \binom{\deg \phi_a}{j} M^{\deg \phi_a - j} \right\rfloor.$$

Moreover, if $\phi_a = \prod_{j=1}^{\deg \phi_a}(x - \sigma_{i_j}(a))$, then

$$\mathrm{size}(a) = \frac{d}{\deg \phi_a} \sum_{j=1}^{\deg \phi_a} |\sigma_{i_j}(a)| \le M.$$

To determine if $a \in \mathbb{C} \setminus \mathbb{Z}$ belongs to $\mathcal{O}_K$, the algorithm of Just [22,23] is used to find a nonzero vector $(y_1, \ldots, y_d, q) \in \mathbb{Z}^{d+1}$ such that $y_1\omega_1 + \ldots + y_d\omega_d = qa$, or show that no such vector exists. If such a vector is found, then the linear independence of the integral basis implies that $a \in \mathcal{O}_K$ if and only if $(y_1/q, \ldots, y_d/q) \in \mathbb{Z}^d$. Moreover, if $a \in \mathcal{O}_K$, then $\boldsymbol{\delta}(a) = (y_1/q, \ldots, y_d/q)$. The algorithm of Just for determining an integer relation, or proving that one does not exist, has complexity summarised by the following theorem:

Theorem 6.2 *Let algebraic numbers $\alpha_1, \ldots, \alpha_n \in \mathbb{C}$ be given by approximations $\bar{\alpha}_1, \ldots, \bar{\alpha}_n \in \mathbb{Q} + i\mathbb{Q}$ such that $\max_{1 \le i \le n} |\alpha_i - \bar{\alpha}_i| \le 2^{-t}$, for*

$$t = \left\lceil \log \left(4n \left(\sqrt{2} \cdot Y^{1/n} \cdot \max_{1 \le i \le n} M(\alpha_i)^{1/[\mathbb{Q}(\alpha_i):\mathbb{Q}]} \right)^{n[\mathbb{Q}(\alpha_1,\ldots,\alpha_n):\mathbb{Q}]} \right) \right\rceil,$$

where $Y > 1$ and $M(\alpha_i)$ is the Mahler measure of α_i, for $1 \le i \le n$. Then

$$O\left(\mathrm{poly}\left(n, \log Y, \max_{1 \le i \le n} \log M(\alpha_i), [\mathbb{Q}(\alpha_1, \ldots, \alpha_n) : \mathbb{Q}] \right) \right)$$

bit operations are sufficient to either find a nonzero integer vector $\mathbf{y} = (y_1, \ldots, y_n)$ such that $\|\mathbf{y}\|_2 \le 2^{n/2}Y$ and $\sum_{i=1}^{n} a_i\alpha_i = 0$, or prove that there is no nonzero integer vector $\mathbf{x} = (x_1, \ldots, x_n)$ such that $\|\mathbf{x}\|_2 \le Y$ and $\sum_{i=1}^{n} x_i\alpha_i = 0$.

The *Mahler measure* of an algebraic number $\alpha \in \mathbb{C}$, with minimal polynomial $\phi_\alpha(x) = \varphi \cdot (x - \alpha_1) \cdots (x - \alpha_n)$ over $\mathbb{Z}$, is defined to be the product $M(\alpha) = |\varphi| \cdot \prod_{i=1}^{n} \max\{1, |\alpha_i|\}$. Mignotte [33, Theorem 1] showed that the Mahler measure of α satisfies $M(\alpha) \le \|\phi_\alpha\|_2$. For $a \in K$,

$$M(a)^{1/[\mathbb{Q}(a):\mathbb{Q}]} = \left(\prod_{i=1}^{d} \max\{1, |\sigma_i(a)|\} \right)^{1/d} \le \frac{1}{d} \sum_{i=1}^{n} \max\{1, |\alpha_i|\} \le \frac{\|a\|}{\sqrt{d}} + 1,$$

where the first inequality is obtained by applying the AM–GM inequality.

Based on the above discussion, the following sub-algorithm is proposed for determining the roots of h in Step 5 of Algorithm 5.1:

Sub-Algorithm 6.1 (Root finding step)

1. Use the method described in the proof of Lemma 6.1 to compute $N_K(h)$.
2. Use the algorithm of Lenstra, Lenstra and Lovász [27] to compute the set Φ of monic non-constant irreducible factors of $N_K(h)$ in $\mathbb{Z}[x]$.
3. Set $\mathcal{R} \leftarrow \{\boldsymbol{\delta}(a) \mid (x-a) \in \Phi, \operatorname{size}(a) \leq M \text{ and } h(a) = 0 \text{ over } K\}$.
4. For each polynomial $\phi \in \Phi$ that satisfies $2 \leq \deg \phi \leq d$ and

$$\|\phi\|_\infty \leq \max_{0 \leq j \leq \deg \phi} \left\lfloor \binom{\deg \phi}{j} M^{\deg \phi - j} \right\rfloor, \tag{6.8}$$

perform the following steps:

(a) Set

$$\tau \leftarrow \left\lceil \log \left(4(d+1) \left(\sqrt{2} \cdot Y^{1/(d+1)} \cdot \max \left\{ X, \|\phi\|_2^{1/\deg \phi} \right\} \right)^{d^2(d+1)} \right) \right\rceil,$$

where

$$Y = \left(2^{1-d} d^{2d+1} \left(M + 2^{-t} \right)^2 + 1 \right)^{1/2} \quad \text{and} \quad X = 1 + \frac{1}{\sqrt{d}} \max_{1 \leq i \leq d} \|\omega_i\|.$$

(b) Let $a_1, \ldots, a_{\deg \phi} \in \mathbb{C}$ denote the roots of ϕ. Use the algorithm of Pan [37] to compute approximations $\bar{a}_1, \ldots, \bar{a}_{\deg \phi} \in \mathbb{Q} + i\mathbb{Q}$ such that

$$\max_{1 \leq j \leq \deg \phi} |a_j - \bar{a}_j| \leq \min \left\{ 2^{-\tau}, (2^{-(t+1)} \cdot \deg \phi)/d^2 \right\}.$$

(c) If the inequality

$$(d/\deg \phi) \cdot \sum_{j=1}^{\deg \phi} |\bar{a}_j| \leq M + 2^{-(t+1)} \tag{6.9}$$

is satisfied, then perform the following steps for $k = 1, \ldots, \deg \phi$:

i. Use the algorithm of Just [22] to search for a nonzero integer vector $\mathbf{y} = (y_1, \ldots, y_d, q)$ such that $\|\mathbf{y}\|_2 \leq 2^{(d+1)/2} Y$ and $\sum_{i=1}^d y_i \omega_i = q a_k$.

ii. If such a vector $(y_1, \ldots, y_d, q) \in \mathbb{Z}^{d+1}$ is found and $(y_1/q, \ldots, y_d/q)$ has integral entries, then compute $h(a_k) \in \mathcal{O}_K$. If $h(a_k) = 0$ over K, then set $\mathcal{R} \leftarrow \mathcal{R} \cup \{(y_1/q, \ldots, y_d/q)\}$.

5. Return $\mathcal{R}$.

Lemma 6.2 *The set $\mathcal{R}$ returned by Sub-Algorithm 6.1 contains the vector $\boldsymbol{\delta}(m)$ for all roots $m \in \mathcal{M}_\mathcal{C}$ of h over K, plus finitely many vectors $\boldsymbol{\delta}(a)$ such that $a \in \mathcal{O}_K$ is a root of h (over K) and $M < \operatorname{size}(a) \leq M + 2^{-t}$. Moreover, the algorithm performs $O(\operatorname{poly}(d, \log |D_K|, \ell, \log M, t, \log \|h\|_{2,M}))$ bit operations.*

Proof The minimal polynomial of any root of h in $\mathcal{O}_K$ is a monic irreducible factor of $N_K(h)$ in $\mathbb{Z}[x]$. Therefore, if $a \in \mathbb{Z}$ is a root of h over K, then it is clear from the construction of $\mathcal{R}$ that $\boldsymbol{\delta}(a) \in \mathcal{R}$ if and only if $\mathrm{size}(a) \leq M$. Moreover, if $a \in \mathcal{O}_K \setminus \mathbb{Z}$ is a root of h over K, then the minimal polynomial of a belongs to the set Φ.

Let $\phi \in \Phi$ and $a_1, \ldots, a_{\deg \phi}$ be the roots of ϕ in $\mathbb{C}$. Fix an index k such that $1 \leq k \leq \deg \phi$. If $a_k \in \mathcal{M}_{\mathcal{C}}$, then the approximate roots $\bar{a}_1, \ldots, \bar{a}_{\deg \phi}$ satisfy (6.9):

$$\frac{d}{\deg \phi} \sum_{j=1}^{\deg \phi} |\bar{a}_j| \leq \mathrm{size}(a_k) + \frac{d}{\deg \phi} \sum_{j=1}^{\deg \phi} |a_j - \bar{a}_j| \leq M + 2^{-(t+1)}.$$

The converse may not hold. However, if $a_k \in \mathcal{O}_K$ and (6.9) is satisfied, then

$$\mathrm{size}(a_k) \leq \left(M + 2^{-(t+1)}\right) + \frac{d}{\deg \phi} \sum_{j=1}^{\deg \phi} |a_j - \bar{a}_j| \leq M + 2^{-t}.$$

Consequently, (6.4) implies that

$$\|\boldsymbol{\delta}(a_k)\|_2 \leq 2^{(1-d)/2} d^{d+1/2} \|a_k\| \leq 2^{(1-d)/2} d^{d+1/2} \left(M + 2^{-t}\right).$$

Therefore, if the approximate roots $\bar{a}_1, \ldots, \bar{a}_{\deg \phi}$ satisfy (6.9), then $a_k \in \mathcal{O}_K$ if and only if there exists a nonzero vector $\mathbf{x} = (x_1, \ldots, x_d, -1) \in \mathbb{Z}^{d+1}$ such that $\|\mathbf{x}\|_2 \leq Y$ and $\sum_{i=1}^{d} x_i \omega_i = a_k$. As $[\mathbb{Q}(\omega_1, \ldots, \omega_d, a_k) : \mathbb{Q}] \leq d^2$,

$$\max_{1 \leq i \leq d} M(\omega_i)^{1/[\mathbb{Q}(\omega_i):\mathbb{Q}]} \leq X \quad \text{and} \quad M(a_k)^{1/[\mathbb{Q}(a_k):\mathbb{Q}]} \leq \|\phi\|_2^{1/\deg \phi},$$

it follows from Theorem 6.2 that the algorithm of Just either proves such a vector $\mathbf{x}$ does not exist, or finds a nonzero vector $(y_1, \ldots, y_d, q) \in \mathbb{Z}^{d+1}$ such that $\sum_{i=1}^{d} y_i \omega_i = q a_k$. In the latter case, $a_k \in \mathcal{O}_K$ if and only if $(y_1/q, \ldots, y_d/q)$ has integral entries. Moreover, if $a_k \in \mathcal{O}_K$, then $\boldsymbol{\delta}(a_k) = (y_1/q, \ldots, y_d/q)$. Therefore, if the approximate roots $\bar{a}_1, \ldots, \bar{a}_{\deg \phi}$ satisfy (6.9) and $a_k \in \mathcal{O}_K$, then $\boldsymbol{\delta}(a_k)$ is added to $\mathcal{R}$ if $h(a_k) = 0$ over K. Hence, if $a_k \in \mathcal{M}_{\mathcal{C}}$ and $h(a_k) = 0$ over K, then $\boldsymbol{\delta}(a_k) \in \mathcal{R}$. Furthermore, if $a_k \in \mathcal{O}_K$ and $\boldsymbol{\delta}(a_k) \in \mathcal{R}$, then a_k is a root of h over K, and $\mathrm{size}(a_k) \leq M + 2^{-t}$. The proof of the first assertion of the lemma is completed by noting that the cardinality of $\mathcal{R}$ is bounded by $\deg N_K(h) \leq d\ell$.

The factorisation of $N_K(h)$ is performed in $O(\mathrm{poly}(d, \ell, \log \|N_K(h)\|_\infty))$ bit operations. Consequently, it follows from Lemma 6.1 that Step 1 and Step 2 perform $O(\mathrm{poly}(d, \log |D_K|, \ell, \log \|h\|_{2,M}))$ bit operations. The inequalities (6.3), (6.4) and (6.5) imply that the evaluation of $h(a)$, for an element $a \in \mathcal{O}_K$ such that $\mathrm{size}(a) \leq M + 2^{-t}$, performs $O(\mathrm{poly}(d, \log |D_K|, \ell, \log M, \log \|h\|_{2,M}))$ bit operations, using the methods described in Section 4.2. As the number of linear factors of $N_K(h)$ is bounded by $\deg N_K(h) \leq d\ell$, it follows that Step 3 performs $O(\mathrm{poly}(d, \log |D_K|, \ell, \log M, \log \|h\|_{2,M}))$ bit operations. Similarly, the number of iterations of the outer loop in Step 4 is bounded by $d\ell/2$. For each iteration, the condition (6.8) and Theorem 6.1 imply that $O(\mathrm{poly}(d, \log |D_K|, \ell, \log M, t, \log \|h\|_{2,M}))$ bit operations are performed in Step 4b. For each iteration of the outer loop, there are

at most d iterations of the inner loop in Step 4c. As $\omega_1, \dots, \omega_d$ is a reduced basis for $(\mathcal{O}_K, T_2)$, property 3 of Theorem 4.1 implies that

$$\max_{1 \leq i \leq d} \|\omega_i\| \leq \frac{\prod_{i=1}^{d} 2^{(i-1)/2} \|\omega_i^*\|}{(\min_{x \in \mathcal{O}_k \setminus \{0\}} \|x\|)^{d-1}} = \frac{2^{d(d-1)/4} \sqrt{|D_K|}}{d^{(d-1)/2}}.$$

since $\det(\mathcal{O}_K, T_2) = \prod_{i=1}^{d} \|\omega_i^*\|$ (see [4, Proposition 2.5.4]). Using this inequality to bound the constant X and applying Theorem 6.2, shows that Step 4(c)i performs $O(\mathrm{poly}(d, \log |D_K|, \ell, \log M, \log \|h\|_{2,M}))$ bit operations for each iteration of the inner loop. In Step 4(c)ii, $O(d(d + \log Y)^2)$ bit operations are performed to determine if the vector $(y_1/q, \dots, y_d/q)$ has integral entries. Thus, for each iteration of the inner loop, Step 4(c)ii performs $O(\mathrm{poly}(d, \log |D_K|, \ell, \log M, \log \|h\|_{2,M}))$ bit operations. Hence, Step 4 performs $O(\mathrm{poly}(d, \log |D_K|, \ell, \log M, t, \log \|h\|_{2,M}))$ bit operations. □

7 Analysis of the decoding algorithm

In this section, the decoding performance and complexity of Algorithm 5.1 is evaluated. The following two lemmas are used to establish a bound on $\|h\|_{2,M}$, which, when combined with Lemma 5.1, determines the decoding performance of the algorithm:

Lemma 7.1 *Let $h \in \mathcal{O}_K[x]$ be the polynomial found in Step 5 of Algorithm 5.1. Then*

$$\|h\|_{2,M} \leq 2^{-e} \cdot \left(1 + \frac{d}{2} \sqrt{\frac{d(d+1)}{2}} \left(2^e \rho - \frac{1}{2}\right)^{-1}\right) \cdot \|R_{e,l,M} \boldsymbol{\delta}_\ell(h)\|_2 \,,$$

where $\rho = \min_{1 \leq i \leq d} \|\omega_i^*\|$.

Proof Write $h = \sum_{j=0}^{\ell} h_j x^j$, where $h_0, \dots, h_\ell \in \mathcal{O}_K$. Define matrices $E_j = \lfloor 2^e M^j R \rceil - 2^e M^j R$, for $0 \leq j \leq \ell$. Then

$$\begin{aligned} 2^{2e} \|h\|_{2,M}^2 &= \sum_{j=0}^{\ell} 2^{2e} M^{2j} \|R\boldsymbol{\delta}(h_j)\|_2^2 = \sum_{j=0}^{\ell} \left\| \left(\lfloor 2^e M^j R \rceil - E_j \right) \boldsymbol{\delta}(h_j) \right\|_2^2 \\ &\leq \sum_{j=0}^{\ell} \left(1 + \left\| E_j \left\lfloor 2^e M^j R \right\rceil^{-1} \right\|_2 \right)^2 \left\| \lfloor 2^e M^j R \rceil \, \boldsymbol{\delta}(h_j) \right\|_2^2 , \end{aligned} \tag{7.1}$$

where the inequality is obtained by applying the Cauchy–Schwarz inequality to each term of the sum and $\left\lceil 2^e M^j R \right\rfloor^{-1}$ exists, for $0 \leq j \leq \ell$, as a result of the inequalities $M \geq 1$ and (5.3). The matrix $E_j \left\lfloor 2^e M^j R \right\rceil^{-1}$ is upper triangular and $\|E_j\|_\infty \leq 1/2$, for $0 \leq j \leq \ell$. Thus,

$$\left\| E_j \left\lfloor 2^e M^j R \right\rceil^{-1} \right\|_2 \leq \frac{d}{2} \sqrt{\frac{d(d+1)}{2}} \left\| \left\lfloor 2^e M^j R \right\rceil^{-1} \right\|_\infty , \quad \text{for } 0 \leq j \leq \ell. \tag{7.2}$$

For each value of j, $0 \leq j \leq \ell$, if $E_j = \left(\varepsilon_{s,t}^{(j)}\right)_{1\leq s,t\leq d}$, then

$$\left\lfloor 2^e M^j R \right\rceil = 2^e M^j R + E_j = 2^{e+1} M^j \cdot \mathrm{diag}\left(\|\omega_1^*\|, \ldots, \|\omega_d^*\|\right) \cdot (I_d + F_j),$$

where

$$F_j := \left(\frac{\mu_{t,s}}{2} + \frac{\varepsilon_{s,t}^{(j)}}{2^{e+1} M^j \|\omega_s^*\|} \right)_{1\leq s,t\leq d} - I_d.$$

For $0 \leq j \leq \ell$, the inequalities $M \geq 1$ and (5.3) imply that

$$1 - \|F_j\|_\infty \geq 1 - \left(\frac{1}{2} + \frac{1}{2^{e+2} M^j \rho} \right) \geq \frac{1}{2} \left(1 - \frac{1}{2^{e+1}\rho} \right) > 0.$$

Therefore, $\left\|(I_d + F_j)^{-1}\right\|_\infty \leq (1 - \|F_j\|_\infty)^{-1}$ (see [40, Lemma 4.4.14]) and

$$\left\| \left\lfloor 2^e M^j R \right\rceil^{-1} \right\|_\infty \leq \frac{1}{2^{e+1}\rho \left(1 - \|F_j\|_\infty\right)} \leq \frac{1}{2^e \rho - 1/2}, \quad \text{for } 0 \leq j \leq \ell. \tag{7.3}$$

Hence, combining inequalities (7.1), (7.2) and (7.3) shows that

$$2^e \|h\|_{2,M} \leq \left(1 + \frac{d}{2} \sqrt{\frac{d(d+1)}{2}} \left(2^e \rho - \frac{1}{2} \right)^{-1} \right) \left(\sum_{j=0}^{\ell} \left\| \left\lfloor 2^e M^j R \right\rceil \boldsymbol{\delta}(h_j) \right\|_2^2 \right)^{\frac{1}{2}},$$

where the second factor on the right-hand side is equal to $\|R_{e,\ell,M} \boldsymbol{\delta}_\ell(h)\|_2$. □

Lemma 7.2 *The lattice $(\Lambda, \|.\|_2^2)$ from Step 4 of Algorithm 5.1 has rank $d(\ell+1)$ and*

$$\det(\Lambda, \|.\|_2^2) \leq \left(\prod_{i=1}^{n} \mathfrak{N}\mathfrak{a}_i^{\binom{z_i+1}{2}} \right) \left(2^e M^{\frac{\ell}{2}} |D_K|^{\frac{1}{2d}} \left(1 + \frac{1}{2^{e+1}\rho} \right) \right)^{d(\ell+1)}. \tag{7.4}$$

Proof The rank of $(\Lambda, \|.\|_2^2)$ is equal to the rank of the matrix Q and its determinant $\det(\Lambda, \|.\|_2^2) = \det(Q^t Q)^{1/2}$. The inequalities $M \geq 1$ and (5.3) imply that the matrix $\left\lfloor 2^e M^j R \right\rceil$ has maximal rank, for $0 \leq j \leq \ell$. Thus, $R_{e,\ell,M}$ has maximal rank and

$$\begin{aligned} \det R_{e,\ell,M} &= \prod_{j=0}^{\ell} \prod_{i=1}^{d} \left\lfloor 2^e M^j \|\omega_i^*\| \right\rceil \leq \prod_{j=0}^{\ell} \prod_{i=1}^{d} \left(2^e M^j \|\omega_i^*\| + \frac{1}{2} \right) \\ &\leq \left(2^e M^{\frac{\ell}{2}} |D_K|^{\frac{1}{2d}} \left(1 + \frac{1}{2^{e+1}\rho} \right) \right)^{d(\ell+1)}. \end{aligned} \tag{7.5}$$

The matrix $U = (U_{s,t})_{1\leq s,t\leq \ell+1}$ is block upper triangular, with $U_{j+1,j+1} = V_j$ for $0 \leq j \leq \min\{z_{\max}, \ell\}$, and all remaining blocks along the leading diagonal (if any) equal to the $d \times d$ identity matrix. By construction, the matrix V_j is the HNF of the ideal $\prod_{i=1}^{n} \mathfrak{a}_i^{\max\{z_i - j, 0\}}$, for $0 \leq j \leq \min\{z_{\max}, \ell\}$. It follows that

$$|\det V_j| = \left[\mathcal{O}_K : \prod_{i=1}^{n} \mathfrak{a}_i^{\max\{z_i - j, 0\}} \right] = \prod_{i=1}^{n} \mathfrak{N}\mathfrak{a}_i^{\max\{z_i - j, 0\}} \neq 0,$$

for $0 \leq j \leq \min\{z_{\max}, \ell\}$ (see [4, Section 4.7.1]). Hence, U has maximal rank and

$$|\det U| = \prod_{j=0}^{\min\{z_{\max},\ell\}} |\det V_j| = \prod_{i=1}^{n} \prod_{j=0}^{\min\{z_i,\ell\}} \mathfrak{N}\mathfrak{a}_i^{z_i-j} \leq \prod_{i=1}^{n} \mathfrak{N}\mathfrak{a}_i^{\binom{z_i+1}{2}}. \tag{7.6}$$

Consequently, $Q = R_{e,\ell,M}U$ has maximal rank and

$$\det(\Lambda, \|.\|_2^2) = \sqrt{\det(Q^t Q)} = |\det R_{e,\ell,M}| \cdot |\det U|. \tag{7.7}$$

Combining (7.5), (7.6) and (7.7) then yields (7.4). □

The following theorem, the main algorithmic result of the paper, describes the decoding performance and computational complexity of Algorithm 5.1:

Theorem 7.1 *Algorithm 5.1 returns all $m \in \mathcal{M}_{\mathcal{C}}$ such that*

$$\prod_{i=1}^{n} \mathfrak{N}\mathfrak{a}_i^{\chi_i(m-r_i)z_i} > \left(2^{\frac{d\ell}{4}} \Delta_K^{(\rho,e)} M^{\frac{\ell}{2}} \sqrt{\ell+1}\right)^d \left(\prod_{i=1}^{n} \mathfrak{N}\mathfrak{a}_i^{\binom{z_i+1}{2}}\right)^{\frac{1}{\ell+1}}, \tag{7.8}$$

where

$$\Delta_K^{(\rho,e)} := \frac{2^{\frac{d-1}{4}} |D_K|^{\frac{1}{2d}}}{d} \left(1 + \frac{d}{2}\sqrt{\frac{d(d+1)}{2}} \left(2^e \rho - \frac{1}{2}\right)^{-1}\right) \left(1 + \frac{1}{2^{e+1}\rho}\right).$$

Moreover, the algorithm performs

$$O\left(\text{poly}\left(d, \log|D_K|, n, z_{\max}, \ell, \log M, \sum_{i=1}^{n} \log \mathfrak{N}\mathfrak{a}_i, e, t\right)\right)$$

bit operations.

Proof Let $\alpha_{j,k} \in \prod_{i=1}^{n} \mathfrak{a}_i^{\max\{z_i-j,0\}}$ be the element such that $\boldsymbol{\delta}(\alpha_{j,k})$ is the kth column vector of V_j, for $0 \leq j \leq \min\{z_{\max}, \ell\}$, $1 \leq k \leq d$. Let $u_j \in \mathcal{O}_K[x]$ be the polynomial such that $\boldsymbol{\delta}_\ell(u_j)$ is the jth column vector of U, for $1 \leq j \leq d(\ell+1)$. Then there exist polynomials $u_1^*, \ldots, u_{d(\ell+1)}^* \in \mathcal{O}_K[x]$ such that $\deg u_{dj+k}^* < j$ and

$$u_{dj+k}(x) = u_{dj+k}^*(x) \cdot a_z + \begin{cases} \alpha_{j,k}(x-r)^j & \text{if } j \leq z_{\max}, \\ x^{j-z_{\max}} \omega_k (x-r)^{z_{\max}} & \text{if } j > z_{\max}, \end{cases}$$

for $0 \leq j \leq \ell$, $1 \leq k \leq d$. By construction $r - r_i \in \mathfrak{a}_i$, for $1 \leq i \leq n$, and a_z is the positive generator of $(\prod_{i=1}^{n} \mathfrak{a}_i^{z_i}) \cap \mathbb{Z}$. It follows that $u_1, \ldots, u_{d(\ell+1)} \in \mathfrak{A}_\ell$ (in fact, the polynomials form an integral basis of $\mathfrak{A}_\ell$ as a consequence of Lemma 5.2). Therefore, the polynomial h, which by construction is a $\mathbb{Z}$-linear combination of the polynomials $u_1, \ldots, u_{d(\ell+1)}$, also belongs to $\mathfrak{A}_\ell$.

From Lemma 7.2, it is known that $(\Lambda, \|.\|_2^2)$ is a rank $d(\ell+1)$ lattice. Therefore, property 2 of Theorem 4.1 implies that

$$\|\mathbf{b}_1\|_2 \leq 2^{\frac{d(\ell+1)-1}{4}} \det(\Lambda, \|.\|_2^2)^{\frac{1}{d(\ell+1)}}.$$

Combining this inequality with Lemma 7.1 and Lemma 7.2 implies that

$$\|h\|_{2,M} \leq 2^{\frac{d\ell}{4}} d\, \Delta_K^{(\rho,e)} M^{\frac{\ell}{2}} \left(\prod_{i=1}^{n} \mathfrak{N}\mathfrak{a}_i^{\binom{z_i+1}{2}} \right)^{\frac{1}{d(\ell+1)}}.$$

Therefore, given $m \in \mathcal{M}_{\mathcal{C}}$ such that (7.8) holds, (5.2) is satisfied and applying Lemma 5.1 with $\mathfrak{a} = \prod_{i=1}^{n} \mathfrak{a}_i^{\chi_i(m-r_i)z_i}$ shows that $h(m) = 0$ over K. Hence, it follows from Lemma 6.2 that for all such $m \in \mathcal{M}_{\mathcal{C}}$, the vector $\boldsymbol{\delta}(m)$ is found by Sub-Algorithm 6.1 and, thus, m is returned by Algorithm 5.1.

If computations from previous iterations are reused on each pass of the loop in Step 1, then the matrices $B_1, \ldots, B_n$ are computed by performing $O(n)$ multiplications of ideals, given by their HNF, with norms bounded by $\prod_{i=1}^{n} \mathfrak{N}\mathfrak{a}_i$. For each iteration, the extended gcd computation in Step 1b performs $O(\mathrm{poly}(d, \sum_{i=1}^{n} \log \mathfrak{N}\mathfrak{a}_i))$ bit operations, with each β_i satisfying $\|\boldsymbol{\delta}(\beta_i)\|_\infty \leq \prod_{i=1}^{n} \mathfrak{N}\mathfrak{a}_i$. By definition, the generator a satisfies $a \leq \prod_{i=1}^{n} \mathfrak{N}\mathfrak{a}_i$. Furthermore, $\|\boldsymbol{\delta}(r_i)\|_\infty \leq \mathfrak{N}\mathfrak{a}_i$, for $1 \leq i \leq n$. As a result, for each iteration of the loop, the sum in Step 1c is computed by performing $O(\mathrm{poly}(d, \sum_{i=1}^{n} \log \mathfrak{N}\mathfrak{a}_i, \log \|(m_{i,j,k})\|_\infty))$ bit operations, using the methods described in Section 4.2. Therefore, using (6.3) to bound the entries of the multiplication table shows that $O(\mathrm{poly}(d, \log |D_K|, n, \sum_{i=1}^{n} \log \mathfrak{N}\mathfrak{a}_i))$ bit operations are performed in Step 1.

If computations from previous iterations are reused on each pass of the loop in Step 2, then the matrices $V_0, \ldots, V_{\min\{z_{\max}, \ell\}}$ are computed by performing $O(z_{\max} + n)$ multiplications of ideals, given by their HNF, with norms bounded by $\prod_{i=1}^{n} \mathfrak{N}\mathfrak{a}_i^{z_i}$. Therefore, the bound (6.3) on the entries of the multiplication table implies that Step 2 performs $O(\mathrm{poly}(d, \log |D_K|, n, z_{\max}, \sum_{i=1}^{n} \log \mathfrak{N}\mathfrak{a}_i))$ bit operations, using the methods described in Section 4.2.

Computing the matrix U in Step 3 reduces to computing the blocks $U_{m+1,j+1}$, for $0 \leq m < j \leq \min\{z_{\max}, \ell\}$: all remaining blocks are either equal to one of these blocks, equal to the $d \times d$ zero matrix, or equal to one of the matrices V_j. The bound (6.3) on the entries of the multiplication table implies that M_r is computed modulo $a_z \leq \prod_{i=1}^{n} \mathfrak{N}\mathfrak{a}_i^{z_i}$ in $O(\mathrm{poly}(d, \log |D_K|, z_{\max}, \sum_{i=1}^{n} \log \mathfrak{N}\mathfrak{a}_i))$ bit operations. The matrix V_j is the HNF of the ideal $\prod_{i=1}^{n} \mathfrak{a}_i^{\max\{z_i - j, 0\}}$, thus $\|V_j\|_\infty \leq \prod_{i=1}^{n} \mathfrak{N}\mathfrak{a}_i^{z_i}$, for $0 \leq j \leq \min\{z_{\max}, \ell\}$. Therefore, by performing arithmetic modulo a_z when computing the blocks $U_{m+1,j+1}$, for $0 \leq m < j \leq \min\{z_{\max}, \ell\}$, Step 3 performs $O(\mathrm{poly}(d, \log |D_K|, z_{\max}, \sum_{i=1}^{n} \log \mathfrak{N}\mathfrak{a}_i))$ bit operations.

The matrix R satisfies

$$\|R\|_\infty \leq \frac{1}{2} \cdot \max_{1 \leq i \leq d} \|\omega_i^*\| \leq \frac{\prod_{i=1}^{d} \|\omega_i^*\|}{2 \left(\min_{1 \leq i \leq d} \|\omega_i^*\|\right)^{d-1}} \leq \frac{2^{(d-1)^2/2}}{2d^{(d-1)/2}} \sqrt{|D_K|},$$

where the final inequality follows from (5.4). Consequently, the matrix $R_{e,\ell,M}$ is computed in $O(\mathrm{poly}(d, \log |D_K|, \ell, \log M, e))$ bit operations. The matrix $R_{e,\ell,M}$ is block diagonal and U is block upper triangular, with each matrix having $d \times d$ blocks. Therefore, $O(d^3 \ell^2 \mathsf{M}(\log \|R_{e,\ell,M}\|_\infty + \log \|U\|_\infty))$ bit operations are performed when computing their product $Q = R_{e,\ell,M} U$. The L^2 algorithm performs

$$O\left(d^5 (\ell+1)^5 \left(d(\ell+1) + \log \|Q\|_\infty\right) \log \|Q\|_\infty\right)$$

bit operations. As the matrix $R_{e,\ell,M}$ is block diagonal, it follows that $\|Q\|_\infty \leq d\,\|R_{e,\ell,M}\|_\infty\,\|U\|_\infty$, where $\|U\|_\infty \leq \prod_{i=1}^{n} \mathfrak{N}\mathfrak{a}_i^{z_i}$ and

$$\|R_{e,\ell,M}\|_\infty \leq 2^e M^\ell\,\|R\|_\infty + \frac{1}{2} \leq \frac{2^{(d-1)^2/2}}{2d^{(d-1)/2}} 2^e M^\ell \sqrt{|D_K|} + \frac{1}{2}. \tag{7.9}$$

Therefore, Step 4 performs $O(\mathrm{poly}(d, \log|D_K|, z_{\max}, \ell, \log M, \sum_{i=1}^{n} \log \mathfrak{N}\mathfrak{a}_i, e))$ bit operations.

The matrix $R_{e,\ell,M}$ is block diagonal, with upper triangular blocks on the diagonal. Therefore, using backward substitution to compute each coefficient of h, the polynomial is computed in $O(d^2\ell\mathsf{M}(\log\|R_{e,\ell,M}\|_\infty + \log\|\mathbf{b}_1\|_\infty))$ bit operations. Sub-Algorithm 6.1 performs $O(\mathrm{poly}(d, \log|D_K|, \ell, \log M, t, \log\|h\|_{2,M}))$ bit operations (see Lemma 6.2). Therefore, Step 5 performs

$$O\left(\mathrm{poly}\left(d, \log|D_K|, z_{\max}, \ell, \log M, \sum_{i=1}^{n} \log \mathfrak{N}\mathfrak{a}_i, e, t\right)\right)$$

bit operations.

For $a \in \mathcal{O}_K$ such that $\boldsymbol{\delta}(a) \in \mathcal{R}$, the value of $\chi_i(a - r_i)$, $1 \leq i \leq n$, is determined by using backward substitution to successively compute the entries of the vector $A_i^{-1}\boldsymbol{\delta}(a - r_i)$ and check that they are integers. Lemma 6.2 and (6.4) imply that $\|\boldsymbol{\delta}(a)\|_\infty \leq 2^{(1-d)/2} d^{d+1/2}(M + 2^{-t})$, for all $\boldsymbol{\delta}(a) \in \mathcal{R}$. Moreover, the set $\mathcal{R}$ contains at most $d\ell$ vectors. Therefore, computing the left hand side of (5.2), for all $a \in \mathcal{O}_K$ such that $\boldsymbol{\delta}(a) \in \mathcal{R}$, is performed in

$$O\left(\mathrm{poly}\left(d, n, z_{\max}, \ell, \log M, \sum_{i=1}^{n} \log \mathfrak{N}\mathfrak{a}_i\right)\right)$$

bit operations. □

Remark 7.1 Given a monic polynomial $f \in \mathcal{O}_K[x]$, a nonzero ideal $\mathfrak{a} \subset \mathcal{O}_K$, a positive real number M and positive integers k and ℓ such that $\ell \geq k \deg f$, the polynomial time algorithm described by Cohn and Heninger [6] in the proof of their Theorem 1.3, when applied with parameters f, $I = \mathfrak{a}$, $\lambda_1 = \ldots = \lambda_d = M$, k and $t = \ell + 1 - k\deg f$, returns all $m \in \mathcal{O}_K$ such that $\mathrm{size}(m) \leq M$ and

$$\mathfrak{N}\gcd\left((f(m)), \mathfrak{a}\right)^k > \left(2^{\frac{d(\ell+1)-1}{4}} \sqrt{\frac{\ell+1}{d}} |D_K|^{\frac{1}{2d}} M^{\frac{\ell}{2}}\right)^d \left(\mathfrak{N}\mathfrak{a}^{\binom{k+1}{2}}\right)^{\frac{\deg f}{\ell+1}}.$$

Given an NF-code $\mathcal{C} = \mathcal{C}_K$ with parameters $(n, \mathfrak{a}_1, \ldots, \mathfrak{a}_n; M, \mathbf{0})$, positive integer weights $z_1, \ldots, z_n$ and a received word $(r_1 + \mathfrak{a}_1, \ldots, r_n + \mathfrak{a}_n)$, weighted list decoding is performed with their algorithm by additionally setting $\mathfrak{a} = \prod_{i=1}^{n} \mathfrak{a}_i$, $k = \max_{1\leq i\leq n} z_i$ and $f = x - r$, where $r \in \mathcal{O}_K$ satisfies $r - r_i \in \mathfrak{a}_i$, for $1 \leq i \leq n$. For list decoding, where the weights $z_1, \ldots, z_n$ are equal, this approach performs almost identically to Algorithm 5.1, with both methods providing a generalisation of the list decoding algorithm for CRT codes of Boneh [3].

8 Parameter selection

Theorem 7.8 shows that Algorithm 5.1 returns all elements of the message space that satisfy a certain "weighted" condition. In this section, parameter selection for Algorithm 5.1 is used to evaluate the algorithm's performance for arbitrarily chosen weights and for the uniform weighting associated with traditional list decoding. The results of this section are straightforward generalisations of those obtained by Guruswami, Sahai and Sudan [14, Section 3.4] for decoding of CRT codes (see also [13, Section 7.6.3]). Consequently, their proofs are abridged, with details left to the reader.

Theorem 8.1 *Let K be a degree d number field and $\mathcal{C} = \mathcal{C}_K$ be an NF-code with parameters $(n, \mathfrak{a}_1, \ldots, \mathfrak{a}_n; M, \mathbf{0})$ such that $M \geq 1$. For positive reals $\beta_1, \ldots, \beta_n$ and any tolerance parameter $\varepsilon > 0$, there exists a choice of parameters $z_1, \ldots, z_n, \ell, e$ such that Algorithm 5.1, when given any input vector $(r_1, \ldots, r_n) \in \mathcal{O}_K^n$, performs*

$$O\left(\operatorname{poly}\left(d, \log|D_K|, n, \log M, \sum_{i=1}^{n} \log \mathfrak{N}\mathfrak{a}_i, t, 1/\varepsilon\right)\right)$$

bit operations and returns all $m \in \mathcal{M}_\mathcal{C}$ such that

$$\sum_{i=1}^{n} \chi_i(m - r_i)\beta_i \geq \sqrt{d \log\left(2^{\frac{d}{2}} M\right)\left(\sum_{i=1}^{n} \frac{\beta_i^2}{\log \mathfrak{N}\mathfrak{a}_i} + \varepsilon \max_{1 \leq i \leq n} \frac{\beta_i^2}{\log^2 \mathfrak{N}\mathfrak{a}_i}\right)}. \quad (8.1)$$

Proof The condition (8.1) is invariant under scaling of the parameters $\beta_1, \ldots, \beta_n$. Therefore, assume without loss of generality that $\max_{1 \leq i \leq n}(\beta_i / \log \mathfrak{N}\mathfrak{a}_i) = 1$. Let A be an integer parameter to be determined later and set $z_i = \lceil A\beta_i / \log \mathfrak{N}\mathfrak{a}_i \rceil$, for $1 \leq i \leq n$. Then $A\beta_i / \log \mathfrak{N}\mathfrak{a}_i \leq z_i < (A\beta_i / \log \mathfrak{N}\mathfrak{a}_i) + 1$, for $1 \leq i \leq n$. Therefore, Theorem 7.1 implies that Algorithm 5.1, for parameters $z_1, \ldots, z_n$, finds all $m \in \mathcal{M}_\mathcal{C}$ such that

$$\sum_{i=1}^{n} \chi_i(m - r_i)\beta_i \geq \frac{1}{2A}\left(d\ell \log(2^{\frac{d}{2}} M) + d \log(\ell + 1)\right) + \frac{d}{A} \log \Delta_K^{(\rho,e)} + \frac{A}{2(\ell+1)} \sum_{i=1}^{n} \left(\frac{\beta_i^2}{\log \mathfrak{N}\mathfrak{a}_i} + \frac{3}{A}\beta_i + \frac{2}{A^2} \log \mathfrak{N}\mathfrak{a}_i\right). \quad (8.2)$$

To complete the proof, the remaining parameters ℓ, A and e are chosen such that this inequality holds for all $m \in \mathcal{M}_\mathcal{C}$ that satisfy condition (8.1).

Define $Z_i = \frac{\beta_i^2}{\log \mathfrak{N}\mathfrak{a}_i} + \frac{3}{A}\beta_i + \frac{2}{A^2} \log \mathfrak{N}\mathfrak{a}_i$, for $1 \leq i \leq n$. Set

$$\ell = \left\lceil A\sqrt{\frac{\sum_{i=1}^{n} Z_i}{d \log(2^{\frac{d}{2}} M)}} \right\rceil - 1 \quad \text{and} \quad e = \left\lceil \frac{1}{2}(d - 3 - \log d) \right\rceil + 1.$$

For this choice of ℓ and e, the right hand side of (8.2) is bounded by

$$\frac{d}{2A} \log\left(A\sqrt{\frac{\sum_{i=1}^{n} Z_i}{d \log(2^{\frac{d}{2}} M)}} + 1\right) + \frac{d}{A} \log \Delta_K^{(\rho,e)} + \sqrt{d \log\left(2^{\frac{d}{2}} M\right) \sum_{i=1}^{n} Z_i},$$

and (5.4) implies that $\Delta_K^{(\rho,e)} < 2^{(d+3)/4}|D_K|^{1/2d}(d + 1/d)$. Consequently, there exists a positive constant $A_0 = \mathrm{poly}(d, \log|D_K|, \sum_{i=1}^n \log \mathfrak{Na}_i, 1/\varepsilon)$ such that, for all $A \geq A_0$, ℓ is positive and the right hand side of (8.2) is less than or equal to

$$\sqrt{d \log\left(2^{\frac{d}{2}} M\right) \left(\sum_{i=1}^{n} \frac{\beta_i^2}{\log \mathfrak{Na}_i} + \varepsilon\right)}.$$

Therefore, set $A = A_0$. For this choice of A, the parameters $z_1, \ldots, z_n$ and ℓ chosen above are all $O(\mathrm{poly}(d, \log|D_K|, \sum_{i=1}^n \log \mathfrak{Na}_i, 1/\varepsilon))$. Thus, the bit complexity follows directly from Theorem 7.1. □

A bound on the performance of Algorithm 5.1 as a traditional (uniform) list decoding algorithm is obtained directly from Theorem 8.1 by setting $\beta_i = 1$, for $1 \leq i \leq n$. A second and generally incomparable bound, is provided by the following corollary to Theorem 8.1:

Corollary 8.1 *Let K be a degree d number field and $\mathcal{C} = \mathcal{C}_K$ be an NF-code with parameters $(n, \mathfrak{a}_1, \ldots, \mathfrak{a}_n; M, \mathbf{0})$ such that $M \geq 1$. Suppose there exists an integer $k \leq n$ such that $\prod_{i=1}^k \mathfrak{Na}_i \geq 2^{d^2/2} M^d$. Then, for any tolerance parameter $\varepsilon > 0$, there exist parameters for Algorithm 5.1 such that, given any input vector $(r_1, \ldots, r_n) \in \mathcal{O}_K^n$, the algorithm returns all $m \in \mathcal{M}_\mathcal{C}$ such that*

$$\sum_{i=1}^{n} \chi_i(m - r_i) \geq \sqrt{k(n + 1 + \varepsilon)}.$$

If $\prod_{i=1}^k \mathfrak{Na}_i = 2^{d^2/2} M^d$, the bound becomes $\sum_{i=1}^n \chi_i(m - r_i) \geq \sqrt{k(n + \varepsilon)}$.

Proof By applying Theorem 8.1 with $\beta_i = \log \mathfrak{Na}_i / \log \mathfrak{Na}_k$, for $1 \leq i < k$, $\beta_i = 1$, for $k \leq i \leq n$, and tolerance parameter $\varepsilon' = \varepsilon \log \mathfrak{Na}_k$, it follows that there exists a choice of parameters for Algorithm 5.1 with the property that, given any input vector $(r_1, \ldots, r_n) \in \mathcal{O}_K^n$, the algorithm returns all $m \in \mathcal{M}_\mathcal{C}$ such that $\sum_{i=1}^n \chi_i(m - r_i)$ is at least

$$k - \frac{d \log\left(2^{d/2} M\right)}{\log \mathfrak{Na}_k} + \sqrt{\frac{d \log\left(2^{d/2} M\right)}{\log \mathfrak{Na}_k} \left(\frac{d \log\left(2^{d/2} M\right)}{\log \mathfrak{Na}_k} + \sum_{i=k}^{n} \frac{\log \mathfrak{Na}_k}{\log \mathfrak{Na}_i} + \varepsilon\right)}.$$

If $\prod_{i=1}^k \mathfrak{Na}_i = 2^{d^2/2} M^d$, then the lower bound of summation in this expression may be increased from k to $k + 1$. In either case, the remainder of the proof follows arguments of Guruswami et al. [14, Theorem 5]. □

9 Conclusion

A polynomial time algorithm (Algorithm 5.1) for solving the weighted list decoding problem for number field codes was presented. The algorithm extends known results for Chinese remainder codes to number fields and reduces the gap to analogous results

for Reed-Solomon and algebraic geometry codes [42, 15]. Comparing Theorem 8.1 with the combinatorial result Corollary 3.1 suggests that Algorithm 5.1 performs sub-optimally for number fields other than $\mathbb{Q}$. The difference between conditions (8.1) and (3.1), which depends on the degree of the number field only, is a consequence of the loss of structure that results from treating the $\mathcal{O}_K$-module $\mathfrak{A}_\ell$ as a $\mathbb{Z}$-module. Progress toward an algorithm that attains the combinatorial bound may result from considering the additional structure.

Acknowledgements The author would like to thank Dr Victor Scharaschkin for many helpful discussions and suggestions throughout the preparation of this paper.

References

1. Belabas, K.: A relative van Hoeij algorithm over number fields. J. Symb. Comput. **37**(5), 641–668 (2004)
2. Belabas, K.: Topics in computational algebraic number theory. J. Théor. Nr. Bordx. **16**(1), 19–63 (2004)
3. Boneh, D.: Finding smooth integers in short intervals using CRT decoding. J. Comput. Syst. Sci. **64**(4), 768–784 (2002).
4. Cohen, H.: A course in computational algebraic number theory. Graduate Texts in Math., vol. 138. Springer-Verlag, Berlin (1993)
5. Cohen, H.: Advanced topics in computational number theory. Graduate Texts in Math., vol. 193. Springer-Verlag, New York (2000)
6. Cohn, H., Heninger, N.: Ideal forms of Coppersmith's theorem and Guruswami-Sudan list decoding. ArXiv:1008.1284v1 (2010).
7. Dumas, J.G., Pernet, C., Wan, Z.: Efficient computation of the characteristic polynomial. In: ISSAC'05: Proceedings of the 2005 international symposium on symbolic and algebraic computation, pp. 140–147. ACM, New York (2005).
8. Edwards, H.M.: Galois theory. Graduate Texts in Math., vol. 101. Springer-Verlag, New York (1984)
9. Forney Jr., G.D.: Generalized minimum distance decoding. IEEE Trans. Inform. Theory **IT-12**(2), 125–131 (1966)
10. Gohberg, I., Lancaster, P., Rodman, L.: Matrix polynomials. Academic Press, New York (1982).
11. Goldreich, O., Ron, D., Sudan, M.: Chinese remaindering with errors. IEEE Trans. Inform. Theory **46**(4), 1330–1338 (2000)
12. Guruswami, V.: Constructions of codes from number fields. IEEE Trans. Inform. Theory **49**(3), 594–603 (2003)
13. Guruswami, V.: List decoding of error-correcting codes. Lecture Notes in Comput. Sci., vol. 3282. Springer, New York (2004)
14. Guruswami, V., Sahai, A., Sudan, M.: "Soft-decision" decoding of Chinese remainder codes. In: Proceedings of the 41st Annual Symposium on Foundations of Computer Science, pp. 159–168. IEEE Comput. Soc. Press, Los Alamitos, CA (2000)
15. Guruswami, V., Sudan, M.: Improved decoding of Reed-Solomon and algebraic-geometry codes. IEEE Trans. Inform. Theory **45**(6), 1757–1767 (1999)
16. Guruswami, V., Sudan, M.: Extensions to the Johnson bound. Manuscript (2001)
17. Hanrot, G., Pujol, X., Stehlé, D.: Algorithms for the shortest and closest lattice vector problems. In: Coding and cryptology, Lecture Notes in Comput. Sci., vol. 6639, pp. 159–190. Springer, Heidelberg (2011)
18. Horowitz, E., Sahni, S.: On computing the exact determinant of matrices with polynomial entries. J. Assoc. Comput. Mach. **22**, 38–50 (1975)
19. Howgrave-Graham, N.: Finding small roots of univariate modular equations revisit In: Darnell, M.J. (ed.) Cryptography and Coding 1997, Lecture Notes in Comput. Sci., vol. 1355, pp. 131–142. Springer, Heidelberg (1997)
20. Johnson, S.M.: A new upper bound for error-correcting codes. IRE Trans. **IT-8**(3), 203–207 (1962)

21. Johnson, S.M.: Improved asymptotic bounds for error-correcting codes. IEEE Trans. Inform. Theory **IT-9**(3), 198–205 (1963)
22. Just, B.: Integer relations among algebraic numbers. In: Kreczmar, A., Mirkowska, G. (eds.) Mathematical Foundations of Computer Science 1989, Lecture Notes in Comput. Sci., vol. 379, pp. 314–320. Springer, Berlin (1989)
23. Just, B.: Integer relations among algebraic numbers. Math. Comp. **54**(189), 467–477 (1990)
24. Kronecker, L.: Grundzüge einer arithmetischen Theorie der algebraischen Grössen. J. Reine. Angew. Math. **92**, 1–122 (1882)
25. Landau, S.: Factoring polynomials over algebraic number fields. SIAM J. Comput. **14**(1), 184–195 (1985)
26. Lenstra, A.K.: Factoring polynomials over algebraic number fields. In: van Hulzen, J.A. (ed.) Computer algebra, Lecture Notes in Comput. Sci., vol. 162, pp. 245–254. Springer, Berlin (1983)
27. Lenstra, A.K., Lenstra Jr., H.W., Lovász, L.: Factoring polynomials with rational coefficients. Math. Ann. **261**(4), 515–534 (1982)
28. Lenstra Jr., H.W.: Codes from algebraic number fields. Hazewinkel, M., Lenstra, J.K., Meertens, L.G.L.T. (eds.) Mathematics and computer science II, Fundamental contributions in the Netherlands since 1945, CWI Monogr., vol. 4, pp. 95–104, North-Holland, Amsterdam (1986).
29. Lenstra Jr., H.W.: Algorithms in algebraic number theory. Bull. Amer. Math. Soc. **26**(2), 211–244 (1992).
30. Mandelbaum, D.M.: On a class of arithmetic codes and a decoding algorithm. IEEE Trans. Inform. Theory **IT-22**(1), 85–88 (1976)
31. Marcus, D.A.: Number fields. Springer-Verlag, New York (1977).
32. McClellan, M.T.: The exact solution of systems of linear equations with polynomial coefficients. J. Assoc. Comput. Mach. **20**, 563–588 (1973)
33. Mignotte, M.: An inequality about factors of polynomials. Math. Comp. **28**, 1153–1157 (1974)
34. Narkiewicz, W.: Elementary and analytic theory of algebraic numbers, 3rd ed. Springer-Verlag, Berlin (2004)
35. Nguyen, P.Q., Stehlé, D.: Floating-point LLL revisited. In: Cramer, R. (ed.) EUROCRYPT 2005, Lecture Notes in Comput. Sci., vol. 3494, pp. 215–233. Springer, Berlin (2005)
36. Nguyen, P.Q., Stehlé, D.: An LLL algorithm with quadratic complexity. SIAM J. Comput. **39**(3), 874–903 (2009).
37. Pan, V.: Sequential and parallel complexity of approximate evaluation of polynomial zeros. Comput. Math. Appl. **14**(8), 591–622 (1987)
38. Richter, H.: Bemerkung zur Norm der Inversen einer Matrix. Arch. Math. **5**, 447–448 (1954)
39. Sidorenko, V., Schmidt, G., Gabidulin, E., Bossert, M., Afanassiev, V.: On polyalphabetic block codes. In: Dinneen, M.J. (ed.) Proc. IEEE ISOC Information Theory Workshop 2005 on Coding and Complexity, pp. 207–210 (2005).
40. Stoer, J., Bulirsch, R.: Introduction to numerical analysis, 2nd ed. Texts in Appl. Math., vol. 12. Springer-Verlag, New York (1993).
41. Storjohann, A.: Algorithms for matrix canonical forms. Ph.D. thesis, ETH Zürich (2000)
42. Sudan, M.: Decoding of Reed-Solomon codes beyond the error-correction bound. J. Complex. **13**(1), 180–193 (1997)
43. Sudan, M.: Ideal error-correcting codes: Unifying algebraic and number-theoretic algorithms. In: Boztas, S., Shparlinski, I.E., (eds.) Proc. Int. 14th Symp. Applied Algebra, Algebraic Algorithms, and Error-Correcting Codes, Lecture Notes in Comput. Sci., vol. 2227, pp. 36–45. Springer, Berlin (2001)