Covering Radius and First Minima Bound on Diagonally Dominant Lattices in the l ∞ -norm

Andrea Lesavourey, Kazuhide Fukushima, Thomas Plantard, Arnaud Sipasseuth

Covering Radius and First Minima Bound on Diagonally Dominant Lattices in the l_∞-norm

Andrea Lesavourey*[1], Kazuhide Fukushima†[2], Thomas Plantard‡ , and Arnaud Sipasseuth§[2]

[1]Institut de Recherche en Informatique et Systèmes Aléatoires, Rennes, France
[2]Information Security Laboratory of KDDI Research, Inc, Japan

Abstract

Diagonally dominant matrices have been a prolific object for mathematical studies for over a century; to this day, it is still an active topic of research. In this paper, we study the lattices generated by diagonal dominant matrices. First, we provide a novel upper bound on their shortest vectors in the maximum norm, as well as a novel bound on the covering radius. Furthermore, we provide a new lattice vector reduction algorithm that gives a better proven worst-case reduction in maximum norm than Babai's Round-Off algorithm within a polynomial amount of operations. Those results expand the understanding of this particular mathematical object, and have some potential applications in modular arithmetic, cryptography and cryptanalysis.

Keywords Diagonally Dominant Lattice, First Minima, Covering Radius, Max norm, Bounded Distance Decoding, Guarantee Distance Decoding, Lattice-based cryptography

1 Introduction

Diagonally dominant matrices have been an interesting object of study for over a century, starting at least from the Lévy-Desplanques theorem (1881)[1], with several links to general matrix theory with research spanning up to today [16, 33, 40, 23, 5, 6, 20, 4, 36]. Numerous applications of diagonal dominance can be found in various fields from numerical linear algebra [22], Markov chains, graphs Laplacians, perturbation theory, etc[2]. On the other hand, the study of lattices generated by diagonally dominant matrices fitting the Lévy-Desplanques theorem were not really studied. Such lattices seemed to have found some application in cryptography on few specific instances [34, 38] where in both papers the focus was more in the matrix generation than a study of the resulting lattice. On the other hand, when strict dominance is not required (i.e not fitting the Lévy-Desplanques theorem), "large diagonals" saw some uses in cryptography [17, 24, 35] and also in modular arithmetic [3].

Note that the study of computational problems on lattices in general is also an old and very studied topic [29, 8, 1] and most recent results including the study of the covering radius for lattices in general are often done by researchers involved in cryptography [25, 18, 19, 21, 9]. This is not surprising: the covering radius of a lattice, such as it is the case of error-correcting codes, directly impacts the decryption capacity of the lattice. Since the heuristic security of lattice-based cryptography is often based on computing a short vector of the lattice of close to it, research on the covering radius can impact both the feasibility and the security of

*andrea.lesavourey@irisa.fr
†ka-fukushima@kddi-research.jp
‡thomas.plantard@gmail.com
§ar-sipasseuth@kddi-research.jp

[1]a history of this theorem through the ages can be seen in [39]
[2]`https://gauss.uc3m.es/fdopico/talks/2014-manch-nasc.pdf` lists some applications

lattice-based constructions. Lattices currently seem very popular in cryptography, as 26 out of 64 accepted submissions to the first round of the NIST competition for Post-Quantum Cryptography Standardisation were based on lattices[30].

Complexity aside, computations using lattices (and ideal lattices) as number systems to provide faster arithmetic or protections against side-channel attacks also have been an important topic of research [3, 15, 11, 10, 7]. In particular, [3] uses a diagonally dominant matrix. The problematic when using such objects is very different from building cryptosystems: the goal is to create structures where vector reduction over a lattice is as easy as possible: if it takes a *few seconds* to "break" a cryptosystem on a laptop, it is bad news for the said cryptosystem, but is also bad news for a number system expecting operations within milliseconds.

In this paper, we study the lattices generated by diagonally dominant matrices and give novel results for the infinity norm which consequently impacts our knowledge on the covering radius. First, we give a lower bound on the size of the shortest vector in infinity norm. Guessing the size of the shortest vector, or even an approximation is known to be NP-hard [12], thus we believe providing a tighter upper bound for any specific family of lattices is welcome to help the comprehension of those objects. Second, we give an improved study of the reduction algorithm of [35] for diagonally dominant matrices and prove a stronger reduction capability than previously proven for such lattices [38]. As the former algorithms only operate with row diagonally dominant matrices, we also give novel algorithms for doing so in the case of column diagonally dominant matrices. We also prove that our aforementioned algorithms, both in the row and column domination cases, operate at most a polynomial (in the dimension and the size of its entries) amount of vector additions or multiplications by a scalar. Furthermore, we show that our upper bound on the size of our algorithm output is asymptotically lower than Babai's Nearest Plane algorithm's upper bound. Note, that we do not claim a *better best case*. Consequently, both results give novel upper and lower bounds on the size of the covering radius for such lattices *for the max norm*, which is also known to be hard to approximate in general [21]. As mentioned previously, theoretical results on lattices often link themselves to applications in cryptography, and we also showcase a potential application of our results for lattice-based encryption and modular arithmetic.

Organization of the paper We first reintroduce the relevant background and some notations in section 2, then present our results on diagonally dominant matrices, first column dominant and then row dominant in section 3. We then exhibit a potential application of our theoretical results to lattice-based encryption scheme in section 4.

2 Background and notations

2.1 Background on lattices

We assume the readers know what is the set of integers $\mathbb{Z}$, the set of integral matrices with n rows and m columns $\mathrm{M}_{n,m}(\mathbb{Z})$, the determinant, norms and other basics of linear algebra. We refer readers to [26, 27] for a more complete background of lattice theory.

Definition 1. *We define an integral lattice $\mathcal{L}$ as a subgroup of $\mathbb{Z}^n$. A basis B of an integral lattice $\mathcal{L}$ is a basis of $\mathcal{L}$ as a $\mathbb{Z}$-module, and denote by $\mathcal{L}(B)$ the lattice generated by the rows of a basis B. We write the volume (or determinant) of the lattice and compute it as* $\det(\mathcal{L}) = \sqrt{\det(BB^t)}$.

Definition 2 (Minimas of a lattice)**.**
We denote by $\lambda_k^{(l)}(\mathcal{L})$ the smallest value r such that a ball centered in zero and of radius r in norm l contains k linearly independent vectors of $\mathcal{L}$.

While many computational problems on lattices exist, we define only the lattice problems that we deem useful for the comprehension of the paper.

Definition 3 (Closest Vector Problem (**CVP**))**.**
Given a basis B of a lattice $\mathcal{L}$ of dimension n and $t \in \mathbb{R}^n$, find $y \in \mathcal{L}$ such that $\forall y_2 \in \mathcal{L}, \|t - y\| \leq \|t - y_2\|$.

An approximate variation of this problem, more in line with the concrete impact of the covering radius we later define, is the following:

Definition 4 (γ-Guaranteed Distance Decoding ($\mathbf{GDD}_\gamma$))**.**
Given a lattice $\mathcal{L}$, and a bound $\gamma > \lambda_1(\mathcal{L})$, for any point x find a lattice vector $v \in \mathcal{L}$ such that $\|x - v\| < \gamma$.

This problem is usually tackled with the combination of a "good" basis (HKZ reduced) with an appropriate algorithm (Babai [1]). Our paper aims to tackle the problem in a slightly different way. To determine to what extent the above problem is solvable, we need to define the *covering radius*.

Definition 5 (Covering radius)**.**
Given a lattice $\mathcal{L}$, we define the covering radius $\mu^{(l)}$ as the smallest value such that for any point $x \in \mathbb{R}^n$, there exists $v \in \mathcal{L}$ such that $\|x - v\|_l < \mu^{(l)}$.

Thus, the larger μ is, the larger is the set where $\mathbf{GDD}_\gamma$ admits a solution. It is known that for any lattice $\frac{1}{2}\lambda_1(\mathcal{L}) \leq \mu(\mathcal{L}) \leq \frac{\sqrt{n}}{2}\lambda_n(\mathcal{L})$ [26]. In this paper, we use a specific family of "good" lattice bases to deal with the above problems:

Definition 6 (Diagonally Dominant Matrix)**.** *Let us consider the matrix $B = (D \times \mathrm{Id}_n) + N$ and the following definitions:*

- $\mathrm{CN}(B, j) = \sum_{\substack{i=1 \\ i \neq j}}^{n} |b_{i,j}|$ *i.e* $\mathrm{CN}(B, j)$ *is the sum of the non-diagonal absolute values of the column j of B.*
  Similarly we write $\mathrm{CN}(B) = \max_{j \in [\![1,n]\!]} \mathrm{CN}(B, j)$.
- $\mathrm{RN}(B, i) = \sum_{\substack{j=1 \\ i \neq j}}^{n} |b_{i,j}|$ *i.e* $\mathrm{RN}(B, i)$ *is the sum of the non-diagonal absolute values of the row i of B.*
  Similarly we write $\mathrm{RN}(B) = \max_{i \in [\![1,n]\!]} \mathrm{RN}(B, i)$.
- *B is row diagonally dominant (i.e r.d.d) iff* $\forall i, \mathrm{RN}(B, i) < B_{i,i}$
- *B is column diagonally dominant (i.e c.d.d) iff* $\forall i, \mathrm{CN}(B, i) < B_{i,i}$
- *We say B has diagonal D if* $\forall i, B_{i,i} = D$.

It follows from the Lévy-Desplanques theorem that a diagonally dominant matrix is always full-rank. Using strictly diagonally dominant matrices of diagonal D, we can deduce from preexisting results that $\mu(\mathcal{L}) \leq \frac{\sqrt{n}}{2}\lambda_n(\mathcal{L}) \leq \frac{\sqrt{n}}{2}D$ when $\mathcal{L}$ admits such a matrix as a basis[3].

2.2 Specific notations

Let us consider the diagonally dominant matrix $B = (D \times \mathrm{Id}_n) + N$ (regardless of whether it is c.d.d or a r.d.d). We will use the following objects and notations.

- $D \in \mathbb{N}^*$ is called the *diagonal coefficient* of the basis B.
- N is called the *noise matrix* of B and its elements *noise values*.
- For $I \subset [\![1, n]\!]$, we denote by $B_I \in \mathrm{M}_{|I|,|I|}(\mathbb{Z})$ the submatrix of B composed of the rows and columns of indexes in I. Naturally, if B is a r.d.d/c.d.d matrix, so is B_I.
- $S_\infty(l)$ is the set of positions i given $l \in \mathbb{Z}^n$ such that $|l_i| = \|l\|_\infty$
- $\mathcal{B}(I, B) = \min\left\{\max_{j \in I}\{|(lB)_j| \mid \|l\|_\infty = 1, S_\infty(l) = I\}\right\}$ given any set of indexes I.
  It is simply $\min\{\|lB_I\|_\infty \mid l \in \{-1, 1\}^{|I|}\}$.
  We denote $\mathcal{B}(I, B)$ by $\mathcal{B}_I$ when B is implied, and stress that $\mathcal{B}_I \neq \lambda_1(B)$.

[3] A lecture of Micciancio explains the bounds on the covering radius `https://cseweb.ucsd.edu/classes/sp14/cse206A-a/lec2.pdf`, and we know that λ_n is at most D in max norm

2.3 Background on number systems

We assume the readers are familiar with integer and polynomial rings, and basic modular arithmetic. While there are many divergent branches on the research over number systems, we only focus here on the parts we believe are related to lattices.

Definition 7 (Modular Number System)**.** *A Modular Number System (**MNS**) $\mathcal{B}$ is defined by a tuple (p, n, γ, ρ), such that for every integer $0 \leq x < p$, there exists a polynomial $U = (u_0, ..., u_{n-1})$ such that: $x = \sum_{i=0}^{n-1} u_i \gamma^i \mod p$ (i.e $U(\gamma) = x \mod p$), with $|u_i| < \rho$ and $0 < \rho, \gamma < p$. We say that U is a representation of x in $\mathcal{B}$ and we denote $U \equiv x_{\mathcal{B}}$*

Note, that the number system holds in the sense that modulo p the equivalencies are conserved by arithmetic operations. There is however a problem with the multiplication of polynomials. Let $U \equiv x_{\mathcal{B}}$ and $V \equiv y_{\mathcal{B}}$, we indeed have $U(\gamma)V(\gamma) = xy \mod p$ however UV is a polynomial of degree higher than n. The following tackle this issue.

Definition 8 (Polynomial Modular Number System)**.** *A Polynomial Modular Number System (PMNS) $\mathcal{B}$ is defined by a tuple (p, n, γ, ρ, E) where (p, n, γ, ρ) is a **MNS** and the monic polynomial $E \in \mathbb{Z}[X]$ of degree n verifies $E(\gamma) = 0 \mod p$. $\|E\|_\infty$ must be "small".*

In a PMNS the computations are done modulo E (called *external reduction polynomial*), which keeps the degree within the **MNS** conditions. The common problem however in both definitions above is the methodology on how to keep coefficients of the resulting polynomials lower than ρ in absolute value. To tackle this issue, an operation called *internal reduction* is used. Several methods exists to achieve this internal reductions, and one of them is to consider the lattice generated by E and apply one of Babai's reduction algorithm to the polynomials we wish to reduce: internal reductions can be seen as some form of Bounded Distance Decoding (**BDD**) solving. When considering the quotient ring constructing using E, one can directly link reduction modulo E by a reduction modulo the lattice (basis) generated by E where E can be considered a "short" vector of the lattice. The link between the covering radius of diagonally dominant matrices and the number systems stem from [3] where E has a particular shape, exhibited in the definition below.

Definition 9 (Adapted Modular Number System)**.** *A Adapted Modular Number System (AMNS) is defined by a tuple (p, n, γ, ρ, E) where (p, n, γ, ρ, E) is a PMNS and $E(X) = X^n - c$ where c is small.*

Note that for most of the paper, we will consider any diagonally dominant matrix which do not necessarily admits a polynomial structure to keep our work generic and open to other applications such as the construction of cryptographic primitives. We might discuss the implication of supplementary structures at some points in the paper.

3 Covering Radius, Shortest vector and reduction algorithms

One of our contributions is to improve our current knowledge by providing known values α, β such that $\alpha \leq \mu^\infty(\mathcal{L}) \leq \beta$ where $\mathcal{L}$ admits a diagonally dominant basis of the form $B = D \times \mathrm{Id}_n + N$. We show that:

- $\alpha = \frac{D - \mathrm{RN}(B)}{2}$ and $\beta = \frac{D + \mathrm{RN}(B)}{2}$ whenever B is a r.d.d.
- $\alpha = \frac{D - \mathrm{CN}(B)}{2}$ and $\beta = \frac{D + \mathrm{CN}(B)}{2}$ whenever B is a c.d.d.

The proofs for those values are similar. The value α is not a direct improvement of the known generic bound, but is a consequence of a new proof on the value of λ_1^∞. Then β is the consequence of novel proofs on our reduction algorithms: we show that for any $v \in \mathbb{Z}^n$ we can reduce it by a deterministic algorithm to $v' \equiv v \mod \mathcal{L}$ such that $\|v'\|_\infty \leq \beta$ which terminates within a polynomial amount of arithmetic operations. Note that for a lattice basis constructed by cycling a vector/polynomial (multiplying by X to obtain the next coefficient) with quotient $X^n - 1$ or $X^n + 1$, being a r.d.d basis also implies being a c.d.d basis.

3.1 Short vector and reduction algorithm on r.d.d

First let us consider r.d.d. matrices. The results proven in this subsection will prove the following theorem.

Theorem 1. *Consider $B \in \mathrm{M}_n(\mathbb{Z})$ a r.d.d. matrix and $\mathcal{L} = \mathcal{L}(B)$. Then $\lambda_1(\mathcal{L}) \geqslant D - \mathrm{RN}(B)$ and there is an algorithm `PSW` (Alg. 1) running in polynomial arithmetic complexity such that*

$$\forall v \in \mathrm{span}(\mathcal{L}), \mathtt{PSW}(v) \equiv v \bmod \mathcal{L}, \|\mathtt{PSW}(v)\|_\infty \leqslant \frac{D + \mathrm{RN}(B)}{2}.$$

Consequently one has $\mu^{(\infty)}(\mathcal{L}) \leqslant \frac{D+\mathrm{RN}(B)}{2}$.

The proof of this theorem is done in two steps: first by proving a lower bound on the size of the shortest vector, second by proving an upper bound on the convergence radius of a polynomial-time reduction algorithm which which we will prove to terminate within a polynomial number of arithmetic operations. The `PSW` acronym stands for `Plantard-Susilo-Win`, which were the original authors of a reduction algorithm [35] we only slightly modify below.

3.1.1 Short vectors

Exposing a simple relationship between $\mathrm{RN}(B)$ and λ_1 does not seem simple, and does not seem to have been studied in details. We here prove an upper bound of λ_1 based on $\mathrm{RN}(B)$.

Lemma 1. *Let $B \in \mathrm{M}_n(\mathbb{Z})$ be r.d.d. of diagonal D. Then $\lambda_1^{(\infty)}(\mathcal{L}(B)) \geqslant D - \mathrm{RN}(B)$.*

Proof. Consider $l \in \mathbb{Z}^n$, and write $v = lB$. Then write $l' = (|l_i|)_{i \in [\![1,n]\!]}$. Clearly there exists $B' \in \mathrm{M}_n(\mathbb{Z})$ a matrix such that $|B'_{i,j}| = |B_{i,j}|$ for any pair $(i,j) \in [\![1,n]\!]^2$, and for all $i \in [\![1,n]\!], B'_{i,i} = D$ and $v_i = \pm(l'B')_i$. Thus B' is a r.d.d. matrix such that $\mathrm{RN}(B', i) = \mathrm{RN}(B, i)$ for all $i \in [\![1,n]\!]$. Now let us show that $\|v\|_\infty \geqslant D - \mathrm{RN}(B)$. We will first bound the taxicab norm, and then use the classic norm inequality

$$\|v\|_\infty \leqslant \|v\|_1 \leqslant n\|v\|_\infty. \tag{1}$$

First remark that we have the following:

$$\|v\|_1 = \sum_{j=1}^{n} |(l'B')_j| \geqslant \left| \sum_{j=1}^{n} \sum_{i=1}^{n} l_i B'_{i,j} \right|.$$

Moreover for any $i \in [\![1,n]\!]$, $l'_i \geqslant 0$ and $D > \mathrm{RN}(B,i)$, so we have

$$\left| \sum_{j=1}^{n} \sum_{i=1}^{n} l_i B'_{i,j} \right| = \sum_{j=1}^{n} \sum_{i=1}^{n} l_i B'_{i,j} \geqslant \sum_{i=1}^{n} l'_i (D - \mathrm{RN}(B,i)).$$

Therefore, if $k = |\{i \in [\![1,n]\!] \mid l_i \neq 0\}|$ we obtain $\|v\|_1 \geqslant k(D - \mathrm{RN}(B))$.
If $k = n$ then Equation (1) gives

$$\|v\|_\infty \geqslant D - \mathrm{RN}(B).$$

Now consider the case with $k < n$. Without any loss of generality, assume $\forall i \in [\![1,k]\!], l_i \neq 0$. Denote by l'' the tuple $(l'_1, \ldots, l'_k)$ and B'' the top left $k \times k$ submatrix of B'. Then B'' is r.d.d. and $\forall i \in [\![1,k]\!], \mathrm{RN}(B'', i) \leqslant \mathrm{RN}(B', i) = \mathrm{RN}(B, i)$. We have

$$\forall \in [\![1,k]\!], (lB)_i = (l'B')_i = (l''B'')_i.$$

Then, since $|\{i \in [\![1,k]\!] \mid l''_i \neq 0\}| = k$, we can apply the previous result to l'' and B'', therefore $\|l''B''\|_\infty \geqslant D - \mathrm{RN}(B'')$ and $\exists i_0 \in [\![1,k]\!], |(l''B'')_{i_0}| = \|l''B''\|_\infty$. Finally we get

$$|(lB)_{i_0}| = |(l'B')_{i_0}| = |(l''B'')_{i_0}| \geqslant D - \mathrm{RN}(B'') \geqslant D - \mathrm{RN}(B') = D - \mathrm{RN}(B). \qquad \square$$

3.1.2 r.d.d-specific reduction algorithm

The PSW reduction algorithm was first introduced in [35], and is a known approximation of Babaï's Round-off algorithm [1] in the case of matrices of the form $D - M$ where MD^{-1} have a spectral radius lower than 1. It was then used a second time in cryptography [34] in the case of r.d.d. matrices. The algorithm was proven to finish for $\delta = D$ in [34], but did not take account of the gap between $\mathrm{RN}(B)$ and D. A slight modification of the reduction proof given in [38] gives us a tighter bound by changing the loop condition in line 2 of the algorithm to a comparison with a value $R_i = \frac{D+\mathrm{RN}(B,i)}{2}$ for every index i. This gives us the modified version, described in Algorithm 1.

Algorithm 1 PSW reduction

Require: $v \in Z^n$, B a r.d.d matrix, a bound vector $R \in \mathbb{N}^n$
Ensure: $w \equiv v \bmod \mathcal{L}(B)$ and $\|w\|_\infty < \max(R_i)$.

1: $w \leftarrow v$
2: **while** $\bigvee_{j=1}^{n}(|w_j| > R_j)$ **do**
3: $\quad i \leftarrow$ any index such that $|w_i| > R_i$
4: $\quad w \leftarrow w - \lfloor \frac{w_i}{D} \rceil B_i$ {Reduce $|w_i|$}
5: **end while**
6: **return** w

The following lemma states that for a given R, the algorithm terminates given that values R_i are above a certain bound which varies for each index.

Lemma 2 (Tighther bound in PSW-reduction algorithm)**.** *For any $v \in \mathbb{Z}^n$ and a r.d.d. matrix B, the PSW reduction (algorithm 1) can output $w \equiv v \bmod \mathcal{L}(B)$ where $\forall i,\ |w_i| \leq \frac{D+\mathrm{RN}(B,i)}{2}$.*

Proof. Let f be the function defined on $\mathbb{Z}^n \times [\![1, n]\!]$ by $f : (w, i) \mapsto w - \lfloor \frac{w_i}{D} \rceil B_i$. In order to show that Algorithm 1 ends and outputs a correct vector, we will prove the following:

$$\bigvee_{j=1}^{n}(|w_j| > R_j) \implies \forall i \in S(w, R), \|f(w, i)\|_1 < \|w\|_1. \tag{2}$$

First let us show if the left side of (2) is verified, then f modifies w. Remark that for all $i \in [\![1, n]\!]$, $f(w, i) = w$ if, and only if, $\lfloor \frac{w_i}{D} \rceil = 0$, which is clearly equivalent to $|w_i| \in [\![-\frac{D}{2}, \frac{D}{2}]\!]$. This condition is clearly verified for any $i \in [\![1, n]\!]$ such that $|w_i| > R_i$. Now let us show that (2) is true. First assume that there is $i \in S(w, R)$ such that $|w_i| > D$. Then $f(w, i)_i$ has the same sign than w_i, therefore $|f(w, i)| = |w_i| - \lfloor \frac{w_i}{D} \rceil D$. Moreover we have

$$\forall j \in [\![1, n]\!] \setminus \{i\}, |w_j| \leqslant |w_j| + \left\lfloor \frac{w_i}{D} \right\rceil |B_{i,j}|,$$

which gives

$$\|f(w, i)\|_1 \leqslant |f(w, i)_i| + \sum_{\substack{j=1 \\ j\neq j}}^{n} |f(w, i)_j| \leqslant |w_i| - \left\lfloor \frac{w_i}{D} \right\rceil D + \sum_{\substack{j=1 \\ j\neq i}}^{n} |w_j| + \left\lfloor \frac{w_i}{D} \right\rceil |B_{i,j}|.$$

This leads to

$$\|f(w, i)\|_1 \leqslant \|w\|_1 + \left\lfloor \frac{w_i}{D} \right\rceil (\mathrm{RN}(B, i) - D) \leqslant \|w\|_1 - \left\lfloor \frac{w_i}{D} \right\rceil < \|w\|_1.$$

Now consider $i \in S(w, R)$ such that $|w_i| < D$. Then $\lfloor \frac{w_i}{D} \rceil = 1$, and the signs of w_i and $f(w, i)_i$ are different. Moreover if we write $|w_i| = R_i + t$ with $t \in [\![1, \frac{D-\mathrm{RN}(B)}{2}]\!]$, we obtain $|f(w, i)_i| = |R_i - D + t| = \frac{D-\mathrm{RN}(B,i)}{2} - t$. Therefore we have

$$|f(w, i)_i| = \frac{D + \mathrm{RN}(B, i)}{2} - t - \mathrm{RN}(B, i) = |w_i| - \mathrm{RN}(B, i) - 2t.$$

Following the same reasoning as before to bound $\|f(w, i)\|_1$ we obtain

$$\|f(w, i)\|_1 \leqslant \|w\|_1 - \mathrm{RN}(B, i) - 2t + \mathrm{RN}(B, i) < \|w\|_1. \qquad \square$$

As stated previously, there is no general polynomial-time algorithm that will give strictly better bounds on l_∞ in every case: by setting $\mathrm{RN}(B) = 0$ we do obtain a covering radius that is half the size of the shortest vector in approximately n vector operations but this is indeed an extreme case. Algorithm 1 use a linear memory and does not need to store much more than the size of the target and the matrix. This is an advantage compared to Babaï's nearest plane algorithm which needs the GSO or Babaï's rounding-off algorithm which requires a matrix inverse. The average-case time-complexity of algorithm 1 was briefly experimented in [35], however a proper worst-case analysis was not provided as in `RSR` and does not seem to have been done in the literature.

Proposition 1. *Let $B \in M_n(\mathbb{Z})$ be a r.d.d. matrix and $v \in \mathbb{Z}^n$, and denote by b the value $\frac{nD}{nD-(D-\mathrm{RN}(B))}$. An upper bound on the complexity of vector operations done by `PSW` with upper bound set to D to reach $\|w\|_1 \leq nD$ is*

$$O\left(\log_b\left(\frac{\|v\|_1}{nD}\right)\right)$$

Proof. Let us consider the reduction of $\|w\|_1$ to count the number of reduction steps, using the results and the reasoning of the above lemma. As the guarantee of the effective reduction was proven using the taxicab norm, we will use it again and consider the case $\|w\|_1 > nD$.
Assuming $\|w\|_1 > nD$ guarantees $\|w\|_\infty > D$ thus $q = \left\lceil \frac{\|w\|_\infty}{D} \right\rfloor \geq 1$. Denote by w' the value of the vector after the update in step 4 of Algorithm 1. Then $\|w\|_1$ is updated as

$$\|w'\|_1 = \|w\|_1 - qD + q\mathrm{RN}(B) = \|w\|_1 - q(D - \mathrm{RN}(B))$$

From $\|w\|_\infty \leqslant \|w\|_1 \leqslant n\|w\|_\infty$ we obtain $q \geqslant \frac{\|v\|_1}{nD}$. Thus we get

$$\|w'\|_1 \leq \|w\|_1 - \frac{\|w\|_1}{nD}(D - \mathrm{RN}(B)) = \|w\|_1 \left(\frac{nD - (D - \mathrm{RN}(B))}{nD}\right)$$

If we use this inequality and we write k the number of steps necessary to reach the condition $\|w\|_1 \leqslant nD$, i.e to reach the second case, using the worst assumptions we obtain:

$$\|w\|_1 = \left(\frac{nD - (D - \mathrm{RN}(B))}{nD}\right)^k \|v\|_1 \leqslant nD.$$

This gives a $O\left(\log_{\frac{nD}{nD-(D-\mathrm{RN}(B))}}\left(\frac{\|v\|_1}{nD}\right)\right)$ number of vector operations to reach $\|w\|_1 \leqslant nD$. □

Recall that in our presentation of the `PSW` algorithm (see algorithm 1), we did not mention in which order the iterations must occur. We are now making the distinction, between naively reducing by index order (`ogPSW` (see in appendix Algorithm 5) which was used in [35, 34]) and choosing the index with the highest coefficient (see in appendix `maxPSW` Algorithm 6). Obviously, the larger is the value $|w_i|$ at the index i we apply the reduction, the larger is the minimal decrease of $\|w\|_1$. The intuition is the following: if we always pick the optimal choice for the reduction, then picking successively based on index order will require at most n times more operations to achieve the same level of efficiency. We then suppose that we have reached $\|w\|_1 \leq nD$ per the previous algorithm, and use an analysis combining both the worst case on $\|w\|_\infty$ and $\|w\|_1$. In the worst-case scenario where we consistently target the largest coefficient of w until every coefficient is lower than D, the worst possibility is where reductions have to continue until $\|w\|_1 < D$ as it is possible that $\|w\|_1 = \|w\|_\infty$. Which reusing the previous reasoning leads to Proposition 2.

Proposition 2. *Let $B \in M_n(\mathbb{Z})$ be a r.d.d. matrix and $v \in \mathbb{Z}^n$, and denote by b the value $\frac{nD}{nD-(D-\mathrm{RN}(B))}$. An upper bound on the complexity of vector operations done by `maxPSW` to reach $\|w\|_\infty \leq D$ is*

$$O\left(\log_b\left(\frac{\|v\|_1}{D}\right)\right)$$

Proof. Reusing the last proof, replace nD by D in the last equation, as we assume in this (very improbable) worst-case where $\|w\|_1 = \|w\|_\infty$ at every iteration we will consistently have $q = \left\lceil \frac{\|w\|_\infty}{D} \right\rfloor \geq 1$ with no coefficient flipping, thus the reasoning still holds past $\|w\|_1 < nD$. □

We stress that proposition 1 differs from 2: proposition 1 do not assume $\|w\|_1 = \|w\|_\infty$, thus yield different results. This overestimated complexity does not reflect at all the significantly faster experimental results reported in [35, 38, 34], which is understandable: the probability to trigger a *single* least-impactful iteration is $2^{-(n-1)}$, i.e as probable as solving a $\{0,1\}$-knapsack problem with $n-1$ entries randomly. However, our result still proves polynomial operation complexity and constant memory (besides input memory) as far as vector operations (i.e fixed dimension) are concerned.

Note that we proved convergence of our algorithms until a bound where $\|w\|_1 < \frac{D+\mathrm{RN}(B)}{2}$ but we still did not upper-bound the complexity of our algorithms to reach $\|w\|_1 < \frac{D+\mathrm{RN}(B)}{2}$. We cannot reuse exactly the same reasoning for further steps beyond b because of coefficient flipping as they change signs: we will indeed have $q = 1$, but we will not have an update of the form $|w_i| \leftarrow |w_i| - D$ but instead an update of the form $|w_i| \leftarrow D - |w_i|$. It makes the analysis less trivial as we cannot guarantee that the max norm stays stable as we iterate, and it is unclear how far reduction should go as terminating to $\|w\|_1 = 0$ means the last step decreases $\|w\|_1$ by exactly $D + \mathrm{RN}(B)$ which is pretty much an optimal reduction and not a worst-case one. Nevertheless we can still obtain a worst-case approach.

Property 1. *We need strictly less than $\frac{nD}{2}$ iterations to reduce $\|v\|_1 = nD$ to $\|w\|_1 = 0$.*

Proof. Assuming we work over $\mathbb{Z}$, every reduction over $\mathbb{Z}$ decreases by 2. If not over $\mathbb{Z}$ and t is the smallest value possible, replace this by $2t$. □

This leads to the grossly overestimated worst-case complexity:

Proposition 3. *Let $B \in M_n(\mathbb{Z})$ be a r.d.d. matrix and $v \in \mathbb{Z}^n$, and denote by b the value $\frac{nD}{nD-(D-\mathrm{RN}(B))}$. An upper bound on the complexity of vector operations done by `PSW` is*

$$O\left(\log_b\left(\frac{\|v\|_1}{nD}\right) + \frac{nD}{2}.\right)$$

Note, that by denoting $\Delta = D - \mathrm{RN}(B)$ the gap between the diagonal and non-diagonal values, approximating $\log(b) = -\log(1 - \frac{\Delta}{nD}) \approx \frac{\Delta}{nD}$ ($\frac{\Delta}{nD}$ is close to 0 so the approximation holds) and setting $\|v\|_1 = nD^n$ (i.e each coefficient to an approximate of the determinant), we can obtain the simpler formula ignoring constants:

$$O\left(n^2 D \frac{\log(D)}{\Delta} + nD\right)$$

Comparison with Babaï's Nearest Plane It is well known that Babaï's nearest plane algorithm [1] gives an upper bound of

$$\mu^{(2)} \leq \frac{\sqrt{n}}{2}\sqrt{\sum \|b_i^*\|_2^2} \leq \frac{\sqrt{n}}{2}\max_i \|b_i^*\|_2$$

where b_i^* are the vectors of the GSO. We also know from norm inequalities

- $\forall v \in \mathbb{R}^n$, $\|v\|_\infty \leqslant \|v\|_2$ which directly implies $\mu^{(\infty)} \leqslant \mu^{(2)}$
- $\|b_i^*\|_2 \leqslant \|b_i\|_2 \leqslant \|b_i\|_1 \leqslant D + \mathrm{RN}(B)$ by r.d.d definition

In the worst-case those inequalities gives $\mu^{(\infty)} \leqslant \frac{\sqrt{n}}{2}(D + \mathrm{RN}(B))$. Thus our worst-case bound seem to be better than Babaï's Nearest Plane worst-case naive bound for *the maximum norm in those lattices*, but this does **not** states the average case is better. However as we will see later, we do have the same output *in the best case* (fully orthogonal lattices). In appendix A.3, we showcase an example where Babaï's Nearest Plane algorithm outputs a larger vector than our algorithms *in maximum norm*, effectively demonstrating that `PSW` can provide better results than Babai's Nearest Plane algorithm in some cases.

3.2 Short vectors and reduction algorithms for c.d.d. matrices

Now let us consider c.d.d. matrices. The overall methodology used in this subsection is very similar to the previous one. Again, the results proven in this subsection can be grouped in the following theorem.

Theorem 2. *Consider $B \in \mathbb{Z}^n$ a c.d.d. matrix and $\mathcal{L} = \mathcal{L}(B)$. Then $\lambda_1(\mathcal{L}) \geqslant D - \mathrm{CN}(B)$ and there is an algorithm, `RSR` (Alg. 3), running within a polynomial amount of arithmetic operations such that*

$$\forall v \in \mathrm{span}(\mathcal{L}), \mathtt{RSR}(v) \equiv v \bmod \mathcal{L}, \|\mathtt{RSR}(v)\|_\infty \leqslant \frac{D + \mathrm{CN}(B)}{2}.$$

Consequently one has $\mu^{(\infty)}(\mathcal{L}) \leqslant \frac{D+\mathrm{CN}(B)}{2}$.

As done previously, the proof of this theorem will be done in two steps: bounding the minimal size of the shortest vector, then bounding the maximal convergence radius of a reduction algorithm. Note that the acronym `RSR` stands for `RepeatedSingleReduce`.

3.2.1 Short vectors

First let us study the norm of a shortest vector.

Lemma 3 (Minimal largest value of non-zero combinations)**.** *Consider $k \in \mathbb{Z}^n \setminus \{0\}$, $j \in [\![1, n]\!]$ such that $|k_j| = \|k\|_\infty$, B be a c.d.d matrix, and $v = kB$. Then one has $|v_j| \geqslant \|k\|_\infty \times (D - \mathrm{CN}(B, j))$.*

Proof. Without any loss of generality we can assume $v_i \geq 0$ and $k_j > 0$. Then

$$|v_i| = \left|\sum_{i=1}^{n} k_i b_{i,j}\right| \geqslant k_j D - \sum_{\substack{i=1 \\ i\neq j}}^{n} |k_i b_{i,j}| \geqslant k_j (D - \sum_{\substack{i=1 \\ i\neq j}}^{n} |b_{i,j}|) = k_j (D - \mathrm{CN}(B, j)). \qquad \square$$

This directly implies that $\lambda_1^{(\infty)}(\mathcal{L}(B)) \geqslant D - \mathrm{CN}(B)$. Let us show some additional results on c.d.d. matrices.

Lemma 4 (Submatrix bound on non-zero combinations)**.** *Consider B a c.d.d. matrix, $k \in \mathbb{Z}^n$, $I = S_\infty(k)$ and $v = kB$. Then there is $j \in I$ such that $|v_j| \geqslant \mathcal{B}(I, B)$.*

Proof. If $\forall j, |k_j| \in \{0, \|k\|_\infty\}$, then there is $j \in S_\infty(k)$ such that $|v_j| \geqslant \|k\|_\infty \times \mathcal{B}(S_\infty(k), B)$. If $\exists j_1, |k_{j_1}| \notin \{0, \|k\|_\infty\}$ with $k_{j_1} \neq 0$, one can pick j_1 such that $|k_{j_1}| \geqslant |k_j|$ for all $j \notin S_\infty(k)$. Consider the vectors k' and k'' such that $k = k' + k''$ and

$$k'_j = \begin{cases} \mathrm{sign}(k_j)(|k|_\infty - |k_{j_1}|), & \text{if } j \in I \\ 0, & \text{otherwise.} \end{cases}$$

Therefore we also have

$$k''_j = \begin{cases} \mathrm{sign}(k_j)(|k_s|), & \text{if } j \in I \\ k_j, & \text{otherwise.} \end{cases}$$

Remark that for all $j \in S_\infty(k)$ we have $\mathrm{sign}(k''_j) = \mathrm{sign}(k'_j) = \mathrm{sign}(k_i)$ and $|k''_j| = |k''|_\infty$. From what precedes we know that there is $j \in S_\infty(k)$ such that $|(k'B)_j| \geqslant \mathcal{B}(S_\infty(k), B)$. Moreover $S_\infty(k) \subset S_\infty(k'')$ and the signs are the same so $\mathrm{sign}((k''B)_j) = \mathrm{sign}((k'B)_j)$. Thus we obtain $|(kM)_j| \geqslant \mathcal{B}(S_\infty(k), B)$. $\square$

This gives us the following theorem.

Theorem 3 (Bound by the minimal submatrix)**.** *Let B be a c.d.d. matrix. Then $\lambda_1^{(\infty)}(\mathcal{L}(B)) \geqslant \min_{I \subseteq [\![1,n]\!]} \mathcal{B}_I$.*

Algorithm 2 `SingleReduce`

Require: $v \in \mathbb{Z}^n$, B a c.d.d matrix, $R_i \geqslant \frac{D+\mathrm{CN}(B,i)}{2}$.
Ensure: $w \equiv v \bmod \mathcal{L}(B)$ and $\|w\|_\infty \leqslant \max(R_i, \|v\|_\infty - (D - \mathrm{CN}(B)))$.

$w \leftarrow v$, $i \leftarrow 1$, $s \leftarrow [0, ..., 0] \in \{0,1\}^n$ {initialization vector, index, reduction status}
while $\bigvee_{j=1}^{n}((|w_j| > R_j) \wedge (s_j = 0))$ **do**
  if $|w_i| > R_i$ and $s_i = 0$ **then**
    $w \leftarrow w - \frac{w_i}{|w_i|} B_i$ {Reduce $|w_i|$}
    $s_i \leftarrow 1$ {"Update" the reduction status of index i}
  end if
  $i \leftarrow (i \bmod n) + 1$ {Enforces i to be within $[1, n]$ and not $[0, n-1]$}
end while
return w

3.2.2 Reduction algorithms for c.d.d. matrices

The previous reduction algorithm only concerned r.d.d matrices and are not guaranteed to terminate on c.d.d matrices. We will propose here a different algorithm relying on the c.d.d structure. Before we present the full algorithm, we first introduce the core part that we denote by `SingleReduce`. It is described in Algorithm 2.

Lemma 5. *Given a vector $v \in \mathbb{Z}^n$, a c.d.d. matrix B with diagonal coefficient D. Moreover let $R \in \mathbb{Z}^n$ be such that $R_i \geqslant \frac{D+\mathrm{CN}(B,i)}{2}$. Then* `SingleReduce` *(Alg. 2) transforms v into $w \in \mathbb{Z}^n$ verifying the following properties.*

1. $v \equiv w \bmod \mathcal{L}(B)$.
2. $\forall i \in [\![1, n]\!], |v_i| > R_i \implies |v_i| > |w_i|$.
3. $\forall i \in [\![1, n]\!], |v_i| \leqslant R_i \implies |w_i| \leqslant R_i$.

Moreover the algorithm performs at most n additions on vectors.

Proof. First remark that we add or remove at most one time each row vector to the variable w. This is ensured by the flag vector s. Therefore we add at most n vectors to w. Write $v = w^{(0)}, w^{(1)}, \ldots, w^{(r)} = w$ the two by two distinct values of the variable w with $r \leqslant n$. Similarly write $s^{(0)}, \ldots, s^{(r)}$ the different values taken by s. Fix some index $i \in [\![1, n]\!]$. First assume $s_i^{(r)} = 0$. Then we know that $|w_i^{(r)}| \leqslant R_i$ and w_i satisfies the claimed properties. Now assume $s_i^{(r)} = 1$. Let us denote by k_0 the integer such that $w_i^{(k_0)} = w_i^{(k_0-1)} \pm D$. Without loss of generality we can assume $v_i \geqslant 0$. First we consider the case where $w_i^{(0)} > R_i$. Then for some $J \subset [\![1, n]\!] \setminus \{i\}$ we have

$$w_i^{(k_0-1)} = w_i^{(0)} + \sum_{j \in J} \pm b_{j,i} \geqslant w_i^{(0)} - \mathrm{CN}(B,i) > R_i - \mathrm{CN}(B,i) \geqslant \frac{D - \mathrm{CN}(B,i)}{2} > 0$$

therefore $w_i^{k_0} = w_i^{(k_0-1)} - D$. We can write

$$w_i^{(0)} > w_i^{(n)} = w_i^{(0)} - D + \sum_{\substack{j \in [\![1,n]\!] \\ j \neq i}} \pm b_{j,i} > R_i - D - \mathrm{CN}(B,i) \geqslant -\frac{D + \mathrm{CN}(B,i)}{2}$$

which ensures $|w_i^{(n)}| < |w_i^{(0)}|$. Now consider the case where $w_i^{(0)} \leqslant R_i$. From $\frac{D+\mathrm{CN}(B,i)}{2} > \mathrm{CN}(B,i)$ we deduce that $w_i^{(k_0-1)} > 0$ and $w_i^{(k_0)} = w_i^{(k_0-1)} - D$. With the same reasoning as before we can conclude $w_i^n < w_i^0$ and $w_i^{(n)} > w_i^{(k_0)} - D - \mathrm{CN}(B,i) > -\frac{D+\mathrm{CN}(B,i)}{2}$ which ensures $|w_i^{(n)}| \leqslant R_i$. Finally we remark that the results obtained are independent of the choice of i. □

Algorithm 3 `RSR`

Require: $v \in \mathbb{Z}^n$, B a c.d.d matrix, $R_i \geqslant \frac{D+\mathrm{CN}(B,i)}{2}$.
Ensure: $w \equiv v \bmod \mathcal{L}(B)$ and $|w_i| \leq R_i$.
1: $w \leftarrow v$
2: **while** $\bigvee_{j=1}^{n}(|w_j| > R_j)$ **do**
3: $\quad w \leftarrow$`SingleReduce`(w,B,R).
4: **end while**
5: **return** w

This building block naturally gives us the `RSR` reduction algorithm, which is guaranteed to finish given a c.d.d. lattice basis. Theoretically, there is no algorithm that can provide strictly better bounds on l_∞ for every single column diagonal dominant lattice: the covering radius cannot be lower than half the size of the shortest vector, and for $\mathrm{CN}(B) = 0$ we do reach this extremity.

Proposition 4. *Given a vector $v \in \mathbb{Z}^n$, $R \in \mathbb{Z}^n$ such that $R_i \geqslant \frac{D+\mathrm{CN}(B,i)}{2}$ where $D, \mathrm{CN}(B,i)$ are associated to a c.d.d. matrix B, `RSR` (Alg. 3) transforms v into $w \in \mathbb{Z}^n$ verifying the following properties.*

1. $v \equiv w \bmod \mathcal{L}(B)$.
2. $\forall i,\ |w_i| < R_i$

Moreover the algorithm performs at most $n\|v\|_\infty$ additions on vectors.

We want to stress this does not show the algorithm is practically efficient: `SingleReduce` might run a *quadratic* amount of absolute value comparisons on scalars in a single call. However, the reduction still runs a polynomial amount of vector operations in the dimension and in the entry size. Despite its apparent inefficiency, the algorithm only requires an extra amount of booleans that is linear in the dimension: this is a significant advantage compared to some alternatives that could require at least quadratic amount of elements which size could be larger than the scalar entries themselves (typically, requiring to inverse a matrix or computing a GSO).

Comparison with Babai's Nearest Plane Unlike the r.d.d case, we do not have a measure of $\|b_i\|_1$. However, we estimate that it is possible in the case of c.d.d to have rows with very large noise, which might give $\|b_i\|_1 > 2D$ and thus a larger worst-case bound than a r.d.d for Babaï's nearest plane algorithm.

3.3 A looser algorithm for reducing both r.d.d, c.d.d, and matrices in between

In the previous sections, the main argumentation for both r.d.d and c.d.d was based on the diagonal dominant structure and the reduction of $\|v\|_1$ per iteration for an entry v. In particular, we can observe that in either case, for a matrix of the form $B = D \times Id + N$ we have $\sum |B_{i,i}| > \sum_{i \neq j} |B_{i,j}|$ for both r.d.d and c.d.d. In this subsection we present an algorithm that reduces vectors in the very generic case of $\sum |B_{i,i}| > \sum_{i \neq j} |B_{i,j}|$. For the sake of simplicity, we assume here that $B_{i,i} \neq 0$. This algorithm, `BalancedReduction` (algorithm 4), can also prove to be a heuristically faster alterative to `SingleReduce` when the initial entries are very large. We present this algorithm last, as it cannot reduce as tightly as any of the previously presented algorithms, thus cannot give the "finishing touches" to reach the minimal bounds for the max norm.

Proposition 5. *Given a vector $v \in \mathbb{R}^n$, a matrix $B \in \mathrm{M}_n(\mathbb{R})$ such that $\sum |B_{i,i}| > \sum_{i \neq j} |B_{i,j}|$ with $B_{i,i} \neq 0$, `BalancedReduction` completes and outputs $w \in \mathbb{R}^n$ such that*

1. $v \equiv w \bmod \mathcal{L}(B)$.
2. $\exists i,\ |w_i| < |B_{i,i}|$
3. $\|w\|_1 \leq \|v\|_1$

Algorithm 4 `BalancedReduction`

Require: $v \in \mathbb{R}^n$, B a matrix with $\sum |B_{i,i}| > q \sum_{i,nej} |B_{i,j}|$
Ensure: $w \equiv v \bmod \mathcal{L}(B)$ and $\|w\|_1 \leqslant \|v\|_1$.
1: $w \leftarrow v$, $q \leftarrow \min\{q_i = \lfloor \| \frac{w_{i,i}}{B_{i,i}} \| \rfloor\}$ {initialization}
2: **while** $q > 0$ **do**
3: **for** $i \in [1, n]$ **do**
4: $w \leftarrow w - q \frac{w_i}{|w_i|} \frac{B_{i,i}}{|B_{i,i}|} B_i$ {reduces by qB_i}
5: **end for**
6: $q \leftarrow \min\{q_i = \lfloor \| \frac{w_{i,i}}{B_{i,i}} \| \rfloor\}$ {updates q}
7: **end while**
8: **return** w

Proof. The first two points items are trivial given the termination: thus termination and reduction of the taxicab norm have to be proven. The choice of q in `BalancedReduction` ensures that if the matrix is actually diagonal, no coefficients flips and thus the reduction of $\|w\|_1$ is at best $q \sum |B_{i,i}|$ per update of q. However if the matrix is not diagonal, there is also at worst an increase of $\|w\|_1$ by $q \sum_{i \neq j} |B_{i,j}|$. Since $\sum |B_{i,i}| > \sum_{i \neq j} |B_{i,j}|$, $\|w\|_1$ is then guaranteed to be reduced. Since $\|w\|_1$ is finite and cannot decrease by less than $q(\sum |B_{i,i}| - \sum_{i \neq j} |B_{i,j}|)$ per update of q, `BalancedReduction` will terminate. □

Unlike the previous algorithms where the reduction operations are decided coefficient per coefficient, in this algorithm the reduction process checks the whole vector and the whole matrix per iteration. This is reminiscent of the reduction algorithm of [3] where each loop iteration takes the whole vector into account, although in their case polynomial structures were used.

4 Potential applications for further work

4.1 Application to lattice-based encryption

Having an efficient reduction algorithm and a lower bound on the shortest vector of a lattice naturally allow for the birth of mathematical encryption primitives. We stress that this section is not the main point of this paper, but merely a presentation of a relevant application. Diagonal dominance itself is a property used in several other fields, and we currently do not fully grasp the overall impact of our contribution besides cryptography. But as we mentioned earlier, cryptography seems a very popular field of application as far as computational problems on lattices are concerned.

Let us denote $\mathcal{L}$ the lattice generated by a diagonal dominant matrix $B = D \times Id + N$. Let R be the radius in which we can find for any $c \in \mathbb{Z}^n$ a vector $m \equiv c \in \mathcal{L}$ s.t. $\|m\|_\infty < R$. Algorithms 1 and 3 offers us parametrisable radii R directly from a parametrisable B. Evidently, B is kept as a secret trapdoor as it allows decryption. Let M be the upper bound of the max norm of the vector messages we wish to recover, such that if the vectors associated to the valid messages belong to a set $\mathcal{M}$, then $\mathcal{M} \subseteq [-M, M]^n$. Here, we consider that each message is associated to a vector $m \in \mathbb{Z}^n$ we wish to recover, and that the encryption of m is associated to a ciphertext vector $c = m + v$ where $v \in \mathcal{L}(B)$. With a similar approach to [17], we first show how one can use our results to guarantee correctness in a decryption. Second, we discuss potential security concerns, which is mostly relevant to cryptographers if they wish to instantiate a cryptosystem.

4.1.1 Guaranteeing decryption of valid messages (i.e correctness)

A sufficient condition to ensure that from any valid ciphertext of the form $c = v + m$ where $v \in \mathcal{L}(B)$ and $m \in [-M, M]^n$, we can recover exactly m, is the following:

$$2M < \lambda_1^{(\infty)}(\mathcal{L}) \text{ and } M < R \tag{3}$$

the first part ensures that if we find $m' \equiv m \mod \mathcal{L}$ then their difference must be a lattice vector of size 0 since $2M < \lambda_1^{(\infty)}(\mathcal{L})$, i.e $m' = m$. The second part enforces that a vector $m' \equiv m \mod \mathcal{L}$ with m' can

always be found. In particular, with our results, equations 3 can be simply verified by fixing

$$\mathrm{CN}(B) < \frac{D-2M}{3} \text{ for c.d.d and } \mathrm{RN}(B) < \frac{D-2M}{3} \text{ for r.d.d} \tag{4}$$

which is straightforward to construct.

4.1.2 Security concerns

There are several security concerns that one needs to address if planning to build a cryptosystem. One of them is to ensure that deciphering c into m is not trivial without the secret key. Heuristically, if c is large enough, the problem of recovering m from c can be seen as a specific instance of **CVP**, which is known to be hard. It is possible to prove that under certain conditions (which are strictly dependent on B), recovering the message is provably as hard as recovering the secret key B within at most a linear factor n (see appendix C).

With that in mind, what is left is the security of the public key. Since [24], it makes sense to provide a basis of $\mathcal{L}(B)$ as a Hermite Normal Form for the public key, however other choices might be possible: it might not even be necessary to provide a basis of $\mathcal{L}(B)$ in the first place. Let us assume the public key is chosen as another basis of the same lattice: in the last decades, it seemed that pure key recovery attacks on diagonal dominant matrices [34, 37] or close structures [17, 28] are rather unsuccessful. The weaknesses were mostly on signature scheme instances [32, 13, 14] which do not concern this section. Note that [32] also consider that the *encryption* approach of [17] is still secure, and to the extent of our knowledge this claim has not been challenged yet.

4.2 Application to modular arithmetic

Without entering the details of the number systems concerned, namely AMNS and PMNS, one of the research directions we could pursue is to propose novel bases for number representations and corresponding algorithms. In those systems, the numbers are represented by vectors modulo a lattice, and thus various numbers have different and redundant information. Depending on the purpose, we might want to either reduce or increase the amount of representations a number have. Computations in such systems usually increase the size of the coefficients in the resulting vectors, thus to avoid overflow and to keep an efficient arithmetic a process call "internal reduction" is proposed, which can be seen as specific instances of a **BDD** problem.

In [11], two ways of producing that internal reduction are exhibited: one is a Montgomery-like reduction, inherited from [31], and the other one is using Babai's algorithms to achieve this. With our work, it might be possible to produce other set of lattices in which those so-called "internal reductions" are processed differently, maybe giving different trade-offs between redundancy and efficacy: we exhibited stronger guarantees over the maximum norm than Babai's algorithms, which is exactly what is more in tune with bounding the coefficients when using **MNS** which actually requires no condition over the euclidean norm l_2.

Note, that **MNS** and its improved variants were created with simplifying operations over $\mathbb{Z}/p\mathbb{Z}$ in mind: thus, lies an open question: how do we create a diagonally dominant lattice with *guaranteed* determinant p? [37] do exhibit a way to create "somewhat diagonally dominant" matrix with a guaranteed determinant with some probability, but it is not tailored for matrices with a polynomial structure nor does it guarantee the exact norm of *every* vector.

5 Conclusion

We gave some improved analysis of the interval where the covering radius lies in the case of diagonally dominant matrices, by analysing the tools given by `PSW` and previous work. We also presented an application for lattice-based encryption as a direct consequence of our results. There is however several avenues to improve our work.

1. The novel algorithms we proposed for the c.d.d case and the "somewhere betweem r.d.d and c.d.d" are inefficient. In particular, we always considered worst-case matrices but it is possible that simple reductions on worst-case matrices could reduce the bounds on the worst-case (a c.d.d matrix with all its noise concentrated on the first row might have its noise reduced by lower rows, leading to overall smaller coefficients).

2. Constructing a worst-case basis where the lower bound of the shortest vector is reached is easy (see appendix A.1), but it is possible this worst-case only exists in such extreme matrices forms. Studying the distribution of all possible minima could also be a future direction.

3. It would be also interesting to study specific results concerning other families of lattices besides those generated by a diagonal dominant matrix, whether or not they have an application to cryptography or other fields such as number systems using lattice-based algorithms for subroutines [2].

References

[1] László Babai. On lovász' lattice reduction and the nearest lattice point problem. *Combinatorica*, 6(1):1–13, 1986.

[2] J-C Bajard, Laurent Imbert, and Thomas Plantard. Arithmetic operations in the polynomial modular number system. In *17th IEEE Symposium on Computer Arithmetic (ARITH'05)*, pages 206–213. IEEE, 2005.

[3] Jean-Claude Bajard, Laurent Imbert, and Thomas Plantard. Modular number systems: Beyond the mersenne family. In *International Workshop on Selected Areas in Cryptography*, pages 159–169. Springer, 2004.

[4] Richard P Brent, H Osborn Judy-anne, and Warren D Smith. Note on best possible bounds for determinants of matrices close to the identity matrix. *Linear Algebra and its Applications*, 466:21–26, 2015.

[5] Richard A Brualdi. Matrices eigenvalues, and directed graphs. *Linear and Multilinear Algebra*, 11(2):143–165, 1982.

[6] Richard A Brualdi and Herbert J Ryser. *Combinatorial matrix theory*, volume 39. Cambridge University Press, 1991.

[7] Titouan Coladon, Philippe Elbaz-Vincent, and Cyril Hugounenq. Mphell: A fast and robust library with unified and versatile arithmetics for elliptic curves cryptography. In *2021 IEEE 28th Symposium on Computer Arithmetic (ARITH)*, pages 78–85, 2021.

[8] JH Conway, RA Parker, and NJA Sloane. The covering radius of the leech lattice. *Proceedings of the Royal Society of London. Series A, Mathematical and Physical Sciences*, pages 261–290, 1982.

[9] Daniel Dadush. On approximating the covering radius and finding dense lattice subspaces. In *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing*, pages 1021–1026, 2019.

[10] Laurent-Stéphane Didier, Fangan-Yssouf Dosso, and Pascal Véron. Efficient modular operations using the adapted modular number system. *Journal of Cryptographic Engineering*, 10(2):111–133, 2020.

[11] Laurent-Stéphane Didier, Fangan-Yssouf Dosso, Nadia El Mrabet, Jérémy Marrez, and Pascal Véron. Randomization of arithmetic over polynomial modular number system. In *2019 IEEE 26th Symposium on Computer Arithmetic (ARITH)*, pages 199–206, 2019.

[12] Irit Dinur. Approximating SVP ∞ to within almost-polynomial factors is NP-hard. In Giancarlo Bongiovanni, Rossella Petreschi, and Giorgio Gambosi, editors, *Algorithms and Complexity*, pages 263–276, Berlin, Heidelberg, 2000. Springer Berlin Heidelberg.

[13] Léo Ducas and Phong Q Nguyen. Learning a zonotope and more: Cryptanalysis of NTRUsign countermeasures. In *International Conference on the Theory and Application of Cryptology and Information Security*, pages 433–450. Springer, 2012.

[14] Léo Ducas and Yang Yu. Learning strikes again: The case of the drs signature scheme. *Journal of Cryptology*, 34(1):1–24, 2021.

[15] Nadia El Mrabet and Nicolas Gama. Efficient multiplication over extension fields. In Ferruh Özbudak and Francisco Rodríguez-Henríquez, editors, *Arithmetic of Finite Fields*, pages 136–151, Berlin, Heidelberg, 2012. Springer Berlin Heidelberg.

[16] S. Gersgorin. Uber die abgrenzung der eigenwerte einer matrix. *Bulletin de l'Academie des Sciences de l'URSS. Classe des Sciences Mathematiques et na*, 6:749–754, 1931.

[17] Oded Goldreich, Shafi Goldwasser, and Shai Halevi. Public-key cryptosystems from lattice reduction problems. In *CRYPTO'97*, pages 112–131. Springer, 1997.

[18] Venkatesan Guruswami, Daniele Micciancio, and Oded Regev. The complexity of the covering radius problem on lattices and codes. In *Proceedings. 19th IEEE Annual Conference on Computational Complexity, 2004.*, pages 161–173. IEEE, 2004.

[19] Venkatesan Guruswami, Daniele Micciancio, and Oded Regev. The complexity of the covering radius problem. *computational complexity*, 14(2):90–121, 2005.

[20] Masunori Harada, Mastaka Usui, and Hiroshi Niki. An extension of the criteria for generalized diagonally dominant matrices. *International journal of computer mathematics*, 60(1-2):115–119, 1996.

[21] Ishay Haviv and Oded Regev. Hardness of the covering radius problem on lattices. In *21st Annual IEEE Conference on Computational Complexity (CCC'06)*, pages 14–pp. IEEE, 2006.

[22] David JN Limebeer. The application of generalized diagonal dominance to linear system stability theory. *International Journal of Control*, 36(2):185–212, 1982.

[23] AI Mees. Achieving diagonal dominance. *Systems & Control Letters*, 1(3):155–158, 1981.

[24] Daniele Micciancio. Improving lattice based cryptosystems using the hermite normal form. In *International Cryptography and Lattices Conference*, pages 126–145. Springer, 2001.

[25] Daniele Micciancio. Almost perfect lattices, the covering radius problem, and applications to ajtai's connection factor. *SIAM Journal on Computing*, 34(1):118–169, 2004.

[26] Daniele Micciancio and Shafi Goldwasser. *Complexity of lattice problems: a cryptographic perspective*, volume 671. Springer Science & Business Media, 2012.

[27] Daniele Micciancio and Oded Regev. Lattice-based cryptography. In *Post-quantum cryptography*, pages 147–191. Springer, 2009.

[28] Daniele Micciancio and Bogdan Warinschi. A linear space algorithm for computing the hermite normal form. In *Proceedings of the 2001 international symposium on Symbolic and algebraic computation*, pages 231–236. ACM, 2001.

[29] Hermann Minkowski. *Geometrie der Zahlen*. B.G. Teubner, Leipzig, 1896.

[30] Dustin Moody. Let's get ready to rumble. the nist pqc "competition". In *Proc. of First PQC Standardization Conference*, pages 11–13, 2018.

[31] Christophe Negre and Thomas Plantard. Efficient modular arithmetic in adapted modular number system using lagrange representation. In *Australasian Conference on Information Security and Privacy*, pages 463–477. Springer, 2008.

[32] Phong Q Nguyen and Oded Regev. Learning a parallelepiped: Cryptanalysis of GGH and NTRU signatures. *Journal of Cryptology*, 22(2):139–160, 2009.

[33] Alexander Ostrowski. Über die determinanten mit überwiegender hauptdiagonale. *Commentarii Mathematici Helvetici*, 10(1):69–96, 1937.

[34] Thomas Plantard, Arnaud Sipasseuth, Cédric Dumondelle, and Willy Susilo. DRS : Diagonal dominant reduction for lattice-based signature. PQC Standardization Conference, Round 1 submissions, 2018. URL: `https://csrc.nist.gov/CSRC/media/Projects/Post-Quantum-Cryptography/documents/round-1/submissions/DRS.zip`.

[35] Thomas Plantard, Willy Susilo, and Khin Than Win. A digital signature scheme based on CVP max. In *PKC 2008*, pages 288–307. Springer, 2008.

[36] Siegfried M Rump. Estimates of the determinant of a perturbed identity matrix. *Linear algebra and its applications*, 558:101–107, 2018.

[37] Arnaud Sipasseuth, Thomas Plantard, and Willy Susilo. Enhancing Goldreich, Goldwasser and Halevi's scheme with intersecting lattices. *Journal of Mathematical Cryptology*, 13(3-4):169–196, 2019.

[38] Arnaud Sipasseuth, Thomas Plantard, and Willy Susilo. Improving the security of the DRS scheme with uniformly chosen random noise. In Julian Jang-Jaccard and Fuchun Guo, editors, *Information Security and Privacy*, pages 119–137, Cham, 2019. Springer International Publishing.

[39] Olga Taussky. A recurring theorem on determinants. *The American Mathematical Monthly*, 56(10P1):672–676, 1949.

[40] Richard S Varga. On diagonal dominance arguments for bounding $|a^{-1}|_\infty$. *Linear Algebra and its applications*, 14(3):211–217, 1976.

APPENDIX

A Specific families of diagonally dominant matrices

As we mentioned earlier, there is no better general results that the one we already provided as the bounds are reached in practice. However additional structures influence the possible covering radii and the behavior of reduction algorithms. This appendix section explore some possibilities.

A.1 All positive, all negative

This subsection considers the case where every $m_{i,j}$ is positive or negative.

A.1.1 Negative case

The negative case offers simple properties that are useful for experimental measure, for the general understanding of diagonally dominant matrices or potential other applications.

Lemma 6 (Shortest vector of the negative case)**.** *Let B be a c.d.d. matrix where $b_{i,j} \leqslant 0$ for all $i \neq j$. Then $v = \sum_{i=1}^{n} b_{i,j}$ is a shortest non-zero vector of $\mathcal{L}(B)$.*

Proof. $v_i = D - \mathrm{CN}(B, i)$, thus reaching the minimal bound non-zero value in every i. □

The advantage of this lemma is to be able to use our bound as the general case when experimenting on a wide array of possibilities regarding the distribution of the non-diagonal coefficients.

A.1.2 Positive case

The positive case gives an interesting intuition for reduction algorithms: they give a very attractive graphical intuition as every vector operation moves every coefficient in the same "direction" (go up or down), i.e the vector's coefficient interval range is guaranteed to be shrinking in each iteration until convergence.

As far as the length of the shortest vector is concerned, there is no guarantee it will be higher than the minimal bound. In fact, the example below show we can reach the general bound:

Example 1. *The matrix*

$$\begin{bmatrix} D & D-1 & 0 & 0 & 0 & 0 \\ 0 & D & D-1 & 0 & 0 & 0 \\ 0 & 0 & D & D-1 & 0 & 0 \\ 0 & 0 & 0 & D & D-1 & 0 \\ 0 & 0 & 0 & 0 & D & D-1 \\ D-1 & 0 & 0 & 0 & 0 & D \end{bmatrix}$$

generates the vector $[1, -1, 1, -1, 1, -1]$

Some constructions with bounded noise coefficients and specific distributions can force limitations on how small the shortest vector can be, however those are very specific cases and expanding on it should be done in another work.

A.2 Polarity-circular blocks

This section deal with matrices that have specific distribution on positive and negative noise coefficients.

A.2.1 2×2 blocks

Here we consider the case where the noise matrix M takes the following form:

$$\begin{bmatrix} 0 & A \\ B & 0 \end{bmatrix}$$

where every coefficient of A is strictly positive and B strictly negative. (A and B can be reversed and are square). In that case, $D > \mathrm{CN}(B) \geqslant n/2$ and the shortest vector is large. In dimension 2, it is clear that the shortest vector is a vector of the basis. In larger dimension, it is not that simple.

Proposition 6 (Shortest vector of 2×2 sign-blocks)**.** *Let $B \in \mathrm{M}_n(\mathbb{Z})$ be c.d.d as described above. Then $\lambda_1(\mathcal{L}(B)) \geqslant D - \mathrm{CN}(B) + \frac{n}{4} + 1$.*

Proof. We mentioned above that we can assume that the sign matrix of M is as follows:

$$\begin{bmatrix} 0 & + \\ - & 0 \end{bmatrix}.$$

As given by Theorem 3, one can concentrate on lB with $\|l\|_\infty = 1$. First consider l such that $\forall i \in [\![1, n]\!], l_i \geqslant 0$ or $\forall i \in [\![1, n]\!], l_i \leqslant 0$. Then clearly $\|lB\|_\infty \geqslant D$. We will now consider l which have at least two distinct coefficients with opposite signs. Let us fix $I_1 = [\![1, \frac{n}{2}]\!]$ and $I_2 = [\![\frac{n}{2} + 1, n]\!]$. First let us consider l such that l is all positive or all negative over I_1, i.e. without loss of generality

$$\forall i \in I_1, l_i \geqslant 0.$$

We will distinguish the two cases where there is $i_2 \in I_2$ with $l_{i_2} > 0$, or $\forall i \in I_2, l_i \leqslant 0$. In the first case we have $|(lB)_{i_1}| \geqslant D + 1$, whereas in the second $|(lB)_{i_1}| \geqslant D + 1$ for $i_1 \in I_1$ is such that $l_1 > 0$. Now let us consider l such that

$$\forall k \in \{1, 2\}, \exists (i_k, j_k) \in I_k^2, (l_{i_k} > 0) \wedge (l_{j_k} < 0).$$

Fix $A = \{i \in I_2 \mid l_i \geqslant 0$. If $|A| \geqslant \frac{n}{4}$ then one has

$$(lM)_{j_1} = -D + \sum_{i \in I_2 \setminus A} |l_i m_{i,j_1}| - \sum_{i \in A} |l_i m_{i,i_1}| \leqslant -D + (\mathrm{CN}(B) - n/4) - 1.$$

If $|A| \leqslant \frac{n}{4}$ then one can do the same reasoning with $I_2 \setminus A$ and i_1. □

A.2.2 3×3 blocks

Now consider the case where the noise matrix M takes the following form:

$$\begin{bmatrix} 0 & A_{12} & B_{13} \\ B_{21} & 0 & A_{23} \\ A_{31} & B_{32} & 0 \end{bmatrix}$$

where every coefficient of A_{ij} is strictly negative and B_{ij} strictly positive (signs of A_{ij} and B_{ij} can be reversed and are square). We assume further the following:

$$\forall j \in [\![1, n]\!], \sum_{i=1}^{n} m_{i,j} = 0.$$

Let us fix some notation. We will write:

- $I = [\![1, n]\!]$;
- $I_k = [\![\frac{(k-1)n}{3} + 1, \frac{kn}{3}]\!]$ for $k \in \{1, 2, 3\}$.

Lemma 7. *Let $M = [m_{i,j}]_{\substack{i \in [\![1,n]\!] \\ j \in [\![1,n]\!]}} \in \mathrm{M}_n(\mathbb{Z})$ a c.d.d. matrix with a structure such as defined above and $n \in 3\mathbb{N}$, and three different values $k_1, k_2, k_3 \in \{1, 2, 3\}$. Consider $l \in \{-1, 0, 1\}^n \setminus \{0\}$ such that $l_i \geqslant 0$ for all $i \in I_{k_1}$ or $l_i \leqslant 0$ for all $i \in I_{k_1}$. Then the following statements are true.*

1. $(\forall i \in I_{k_1} \cup I_{k_2}, l_i = 0) \implies \|lM\|_\infty \geqslant D\|l\|_\infty$*; (same for $I_{k_1} \cup I_{k_3}$ and $I_{k_2} \cup I_{k_3}$).*
2. $\exists k \in \{k_2, k_3\} \mid \forall j \in I_k, l_j = 0 \implies \|lM\|_\infty \geqslant D$.
3. $\forall k \in \{k_2, k_3\}, \exists i_k \in I_k \mid l_{i_k} \neq 0 \implies \|lM\|_\infty \geqslant D - \frac{\mathrm{CN}(B)}{2} + 1$.

Proof. Without any loss of generality, we can assume that $l_i \geqslant 0$ for all $i \in I_1$ and $m_{i_j} > 0$ for all $(i, j) \in I_2 \times I_1$. The sign matrix of M is as follows:

$$\begin{bmatrix} 0 & - & + \\ + & 0 & - \\ - & + & 0 \end{bmatrix}.$$

The first statement is clear. Now let us prove statement (ii). It corresponds to proving Proposition 6. Without loss of generality assume $l_j = 0$ for all $j \in I_3$ (i.e $k = 3$). If there is $j \in I_2$ such that $l_j < 0$, then since $l_i \geqslant 0$ and $m_{i,j} \leqslant 0$ for all $i \in I_1$, we have

$$(lM)_j = -|l_j| D - \sum_{i=1}^{n/3} l_i |m_{i,j}| \leqslant -D - 1,$$

thus $\|lM\|_\infty > D$. If $l_j \geqslant 0$ for all $j \in I_2$ then $\|(lM)_i\|_\infty \geqslant D$ for all $i \in I_1$.
Let us now prove (iii). Following the same reasoning as before, one can see that if $l_{i_2} < 0$ then

$$(lM)_{i_2} = -|l_j| D - \sum_{i \in I_1} l_i |m_{i,j}| + \sum_{i \in I_3} l_i m_{i,j} \leqslant -D - 1 + \sum_{i \in I_3} l_i m_{i,j} < 0$$

thus $|(lM)_{i_2}| \geqslant D + 1 - \frac{\mathrm{CN}(B)}{2}$. Similarly if $l_{i_3} > 0$ then $|(lM)_{i_3}| \geqslant D + 1 - \frac{\mathrm{CN}(B)}{2}$. Finally if $l_i \geqslant 0$ for all $i \in I_2$ and $l_i \leqslant 0$ for all $i \in I_3$ then $\|lM\|_\infty > D$ and (iii) is true.
Since all of the above can be adapted to the cases where $l_i \leqslant 0$ for all $i \in I_1$, or where we replace I_1 by I_2 or I_3 we proved that if there is $k \in \{1, 2, 3\}$ such that all of the coefficients l_i with $i \in I_k$ have the same sign, then $\|lM\|_\infty > D - \frac{\mathrm{CN}(B)}{2}$. □

Lemma 8. *Let $M \in \mathrm{M}_n(\mathbb{Z})$ with a structure such as defined above and $n \in 3\mathbb{N}$. Then for $l \in \{-1,0,1\}^n$, $v = lM$ has $\|v\|_\infty \geq \min\{D - \frac{\mathrm{CN}(B)}{2}, D - \mathrm{CN}(B) + \frac{n}{3} + 2\}$.*

Proof. The previous lemma dealt with the case where $\exists k \in \{1,2,3\}$ such that $\forall i \in I_k, l_i \geqslant 0$ or $\forall i \in I_k, l_i \leqslant 0$. Now assume the following:

$$\forall k \in \{1,2,3\}, \exists (i_k, j_k) \in I_k^2, (l_{i_k} > 0) \wedge (l_{j_k} < 0).$$

Remark that it implies $n \geqslant 6$. With no loss of generality, let us fix $k = 1$ and define

$$A = \{i \in I_2 \mid l_i \geqslant 0\} \text{ and } B = \{i \in I_3 \mid l_i \leqslant 0\}$$

First assume that $|A| \geqslant \frac{n}{6}$ and $|B| \geqslant \frac{n}{6}$. Then we have

$$(lM)_{i_1} = D + \sum_{i \in A \cup B} |l_i m_{i,i_1}| - \sum_{i \in I_2 \cup I_3 \setminus A \cup B} |l_i m_{i,i_1}| \geqslant D + 2 - 2(\frac{\mathrm{CN}(B)}{2} - \frac{n}{6}) \geqslant D - \mathrm{CN}(B) + \frac{n}{3} + 2.$$

Now assume $|A| \leqslant \frac{n}{6}$ and $|B| \leqslant \frac{n}{6}$. Then similarly as before we obtain

$$(lM)_{j_1} = -D + \sum_{i \in A \cup B} |l_i m_{i,j_1}| - \sum_{i \in I_2 \cup I_3 \setminus A \cup B} |l_i m_{i,j_1}| \leqslant -D + \mathrm{CN}(B) - \frac{n}{3} - 2.$$

Finally, assume $|A| \geqslant \frac{n}{6}$ and $|B| \leqslant \frac{n}{6}$. This means that $|I_3 \setminus B| \geqslant \frac{n}{6}$ so we obtain

$$\begin{aligned}(lM)_{j_1} &= -D + \sum_{i \in A \cup B} |l_i m_{i,j_1}| - \sum_{i \in I_2 \cup I_3 \setminus A \cup B} |l_i m_{i,j_1}| \\ &\leqslant -D - 1 - \frac{n}{6} + (\frac{\mathrm{CN}(B)}{2} - 1) + (\frac{\mathrm{CN}(B)}{2} - \frac{n}{6}) \leq -D + \mathrm{CN}(B) - \frac{n}{3} - 2.\end{aligned}$$

The case $\#A \geqslant \frac{n}{6}$ and $\#B < \frac{n}{6}$ follows a similar reasoning. □

Finally, using Theorem 3 one can deduce from the results over $l \in \mathbb{Z}^n$ with $\|l\|_\infty = 1$ a lower bound for λ_1.

Corollary 1. *Consider B a c.d.d. matrix by blocks. Then it verifies $\lambda_1^{(\infty)}(\mathcal{L}(B)) \geqslant \min\{D - \mathrm{CN}(B) + \frac{n}{3} + 2, D - \frac{\mathrm{CN}(B)}{2}\}$.*

Note that those bounds are reached in the very worst case, and we present below an example that was built to reach the bound.

Example 2. *Set $D = 19, \mathrm{CN}(B) = 18, n = 6$. This gives $\lambda_1^{(\infty)} \geqslant 5$. Consider the matrix*

$$M = \begin{bmatrix} D & 0 & -1 & 1 - \frac{\mathrm{CN}(B)}{2} & 1 & \frac{\mathrm{CN}(B)}{2} - 1 \\ 0 & D & 1 - \frac{\mathrm{CN}(B)}{2} & -1 & \frac{\mathrm{CN}(B)}{2} - 1 & 1 \\ \frac{\mathrm{CN}(B)}{2} - 1 & 1 & D & 0 & -1 & 1 - \frac{\mathrm{CN}(B)}{2} \\ 1 & \frac{\mathrm{CN}(B)}{2} - 1 & 0 & D & 1 - \frac{\mathrm{CN}(B)}{2} & -1 \\ 1 - \frac{\mathrm{CN}(B)}{2} & -1 & \frac{\mathrm{CN}(B)}{2} - 1 & 1 & D & 0 \\ -1 & 1 - \frac{\mathrm{CN}(B)}{2} & 1 & \frac{\mathrm{CN}(B)}{2} - 1 & 0 & D \end{bmatrix}$$

$$= \begin{bmatrix} 19 & 0 & -1 & -8 & 1 & 8 \\ 0 & 19 & -8 & -1 & 8 & 1 \\ 8 & 1 & 19 & 0 & -1 & -8 \\ 1 & 8 & 0 & 19 & -8 & -1 \\ -8 & -1 & 8 & 1 & 19 & 0 \\ -1 & -8 & 1 & 8 & 0 & 19 \end{bmatrix}$$

and $l = \begin{bmatrix} -1 & 1 & 1 & -1 & -1 & 1 \end{bmatrix}$. This gives out $v = lM = \begin{bmatrix} -5 & 5 & 5 & -5 & -5 & 5 \end{bmatrix}$ which has a norm of 5.

Note that unlike the example above, for large dimensions (and large diagonal value D) it is very unlikely that the maximum noise with absolute value $(\frac{\mathrm{CN}(B)}{2} - \frac{n}{3} + 1)$ is picked for uniform distributions. Bounding the maximum noise coefficient will further increase the minimum possible length of the shortest vector.

A.3 Example where Babai's output is worse than PSW's output

When taking random examples *especially* in cases where *unicity of the solution* is guaranteed, it is common to see Babai's output being equal to PSW's output. It is however possible to create instances where Babai's output has a larger maximum norm than our algorithms' output. For r.d.d matrices, shifting a lot of the non-zero coefficients on the last column increase the probability that Babai's nearest plane algorithm outputs a larger vector in the maximum norm than the `PSW` algorithm. We present below a code example which can be easily tested on `http://magma.maths.usyd.edu.au/calc/`.

```
M:=Matrix(Integers(),[
[22,  0,  0, 0, 0, 0, 0, 1,-1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,-1],
[ 1, 22,  0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,-1, 1],
[ 0,  0, 22, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 0, 0,-1],
[ 0,  0,  0,22, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0,-1],
[ 0,  0,  0, 0,22, 0, 1, 0, 0, 0, 0,-1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1],
[ 0,  0,  1, 0, 0,22, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1,-1],
[ 1,  0,  0, 0, 0, 0,22, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0,-1],
[ 0,  0,  0, 0, 0, 0, 0,22, 0, 0, 0, 0, 1, 0,-1, 0, 0, 0, 0, 0, 0, 1],
[ 0,  0,  0, 0, 0,-1, 0, 0,22, 0, 0, 0, 0, 0, 0, 0,-1, 0, 0, 0, 0,-1],
[ 0,  1,  0, 0, 0, 0, 0, 0, 0,22,-1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,-1],
[ 0,  0,  0, 0, 0, 0, 0, 0, 0, 0,22, 0, 0, 0, 0, 0, 1, 0, 0,-1, 0, 1],
[ 1,  0, -1, 0, 0, 0, 0, 0, 0, 0, 0,22, 0, 0, 0, 0, 0, 0, 0, 0, 0,-1],
[ 0,  0,  0, 0, 0, 0,-1, 0, 0, 0, 0, 0,22, 0, 0,-1, 0, 0, 0, 0, 0, 1],
[ 0,  1,  0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,22, 0, 1, 0, 0, 0, 0, 0, 1],
[ 0,  0,  1, 0,-1, 0, 0, 0, 0, 0, 0, 0, 0, 0,22, 0, 0, 0, 0, 0, 0, 1],
[ 0,  0,  0, 0,-1, 0, 0, 0, 0, 0, 0, 0, 0,-1, 0,22, 0, 0, 0, 0, 0, 1],
[ 0,  0,  0, 0, 0, 1, 0, 0, 0, 0,-1, 0, 0, 0, 0, 0,22, 0, 0, 0, 0, 1],
[ 0, -1,  0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0,22, 0, 0, 0,-1],
[ 0,  0,  0, 0, 0, 0, 0,-1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0,22, 0, 0, 1],
[ 0,  0,  0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,-1, 0, 0, 0, 0, 0, 0,23, 0,-1],
[ 0, -1,  0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,-1,22, 1],
[ 0,  0,  0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0,-1, 0, 0, 0, 0, 0, 0,23]
]);

Alpha:=Ceiling((22+3)/2);
print "Wanted max norm <", Alpha;

ExpectedPsw:=Vector(
[ 6,-6, 6, 6,-6, 6, 6,-6, 6, 6,-6, 6,-6,-6,-6,-6,-6, 6,-6, 6,-6, 6]
);
ExpectedBabai:=Vector(
[ 6,-6, 6, 6,-6, 5, 6,-6, 6, 6,-6, 6,-6,-6,-5,-6,-6, 6,-6, 6,-6,-17]
);

Det:=Determinant(M);
c:=Vector(ExpectedPsw);
for i:=1 to 22 do
  c+:=Random(-Round(Sqrt(Det)),Round(Sqrt(Det)))*M[i];
end for;
print "large vector to reduce";c;

vext := Matrix(Integers(),[[23] cat [c[i] : i in [1..22]]]);
Bext:=VerticalJoin(
        HorizontalJoin(ZeroMatrix(Integers(),22,1),M),
          Matrix(Integers(),vext));
```

```
BabaiVec:=Vector([LLL(Bext)[23][i] : i in [2..23]]);
print "Recovered Vector by Babai"; BabaiVec;

i:=1;
while Max([Abs(c[i]) : i in [1..22]]) gt Alpha do
  c-:=Round(c[i]/(M[i][i]))*M[i];
  i:=(i mod 22)+1;
end while;
print "Recovered Vector by PSW"; c;

print "Results as expected:",
(c eq ExpectedPsw) and (BabaiVec eq ExpectedBabai);
```

In the above code snippet, the `PSW`'s output can be given to Babai's algorithm as an input which *will increase* the maximum norm. We could not find a **BaseReduce** method nor a **BabaiNearestPlane** method, thus to save some lines we used LLL on an extended basis in which we recover an output that is equal to Babai's output.

B Complexity analysis of r.d.d reduction

In this part we give a further study of the complexity of the `PSW` algorithm. In this paper, we have proven an upper-bound for the worst-case complexity of the `PSW` reduction algorithm and in [35, 34] experimental results were given for some specific examples. Recall that in our presentation of the `PSW` algorithm (see algorithm 1), we did not mention in which order the iterations must occur. We are now making the distinction, between `ogPSW` (Algorithm 5) which were used in [35, 34] and `maxPSW` (Algorithm 6).

Algorithm 5 Original `PSW` reduction

Require: $v \in Z^n$, B a r.d.d matrix, a vector $R \in \mathbb{N}^n$
Ensure: $w \equiv v \bmod \mathcal{L}(B)$ and $\|w\|_\infty < \max(R_i)$.

$w \leftarrow v$, $i \leftarrow 1$, $s \leftarrow 0$ {initial vector, index, skipped iterations}
while $s < n$ **do**
  $i \leftarrow (i \mod n) + 1$
  if $|w_i| < R_i$ **then**
    $s \leftarrow s + 1$ {skip iteration}
  else
    $w \leftarrow w - \lfloor \frac{w_i}{D} \rceil B_i$, $s \leftarrow 1$ {Reduce $|w_i|$, reset iteration count}
  end if
end while
return w

Algorithm 6 Max-choice `PSW` reduction

Require: $v \in Z^n$, B a r.d.d matrix, a vector $R \in \mathbb{N}^n$
Ensure: $w \equiv v \bmod \mathcal{L}(B)$ and $\|w\|_\infty < \max(R_i)$.

$w \leftarrow v$
while $\bigvee_{j=1}^{n}(|w_j| > R_j)$ **do**
  $i \leftarrow$ any index such that $|w_i| = \|w\|_\infty$
  $w \leftarrow w - \lfloor \frac{w_i}{D} \rceil B_i$ {Reduce $|w_i|$}
end while
return w

In `ogPSW`, the next index to reduce was chosen using the natural incremental order of the indexes over $\mathbb{N}$. This choice was made by default, chosen for its simplicity. However in `maxPSW`, we choose the index i which

maximizes the ratio $|w_i|/D$, i.e $|w_i| = \|w\|_\infty$. This does not guarantee the fastest run time possible: our reduction proofs so far relied on the decrease of $\|w\|_1$ which consequently decreased $\|w\|_\infty$, and maximizing $|w_i|/D$ do not automatically ensure $\|w\|_1$ is reduced maximally for this choice of i. Furthermore, the search for $|w_i| = \|w\|_\infty$ induces further computations that could be seen as equivalent to an extra vector operation per iteration. Note that `ogPSW` do not even compute $\|w\|_\infty$, but has to measure the number of skipped iterations where a count of n indicates a sufficiently reduced vector.

However, there is both a practical and theoretical advantage of `maxPSW` over `ogPSW`, which is why we introduce `maxPSW`: `maxPSW` performs noticeably less iterations in our experiments compared to `ogPSW`, and also provides a worst-case approach *based on* l_∞ that is specific to `maxPSW`, and provides us with some arguments to construct heuristic average-case complexities that match more closely the iterations count observed in practice. Let us first start with a worst-case study of `maxPSW` after *exactly* n *iterations*.

B.1 Worst-case first n iterations

The initial worst-case as input would be a vector w where for $\forall i, |w_i| = \|w\|_\infty$: this, given $\|w\|_\infty$, maximizes $\|w\|_1$ thus maximizes the number of iterations necessary to reach our bounds according to our previous proofs. It also creates the first problematic phenomena that we have to deal with:

For each iteration, reducing $\|w\|_1$ *can* increase $\|w\|_\infty$.

In particular, suppose that in case of equality between the absolute values of each coefficient, we choose to reuse the incremental order of `ogPSW`: then, after the first iteration, we have at worst

$$|w_1| \leftarrow D/2,\ |w_2| \leftarrow |w_2| + \frac{|w_1|}{D}\mathrm{RN}(B) \ \text{ i.e } \ \|w\|_\infty \leftarrow \|w\|_\infty + \frac{|w_1|}{D}\mathrm{RN}(B)$$

The next maximum value is then at position 2, thus using `maxPSW` we reduce w_2 which in turn increases $\|w\|_\infty$ again and make the next position 3 (or another), and so on. Note, however, that this incremental pattern on $\|w\|_\infty$ can only repeat $n-1$ times: w_1 is at most $D/2$ after the first iteration, so on the n-th iteration we obtain

$$|w_1| \leftarrow D/2 + \frac{|w_n|}{D}\mathrm{RN}(B)$$

which gives us our reasoning: we will analyse the worst-case complexity by n *consecutive* iterations, as opposed as *per* iteration. Thus, we consider in our worst-case approach of `maxPSW` that in every n iterations we obtain

$$\|w\|_\infty \leftarrow D/2 + \delta \times \mathrm{RN}(B)$$

and we need to determine δ. This can be determined by remarking that the previous pattern is an arithmetico-geometric sequence up to $n-1$ terms, in particular it is the series

$$u_{i+1} = \frac{\mathrm{RN}(B)}{D}u_i + u_0,\ r = \frac{u_0}{1-\mathrm{RN}(B)/D} = \frac{u_0 D}{D-\mathrm{RN}(B)},\ u_k = \left(\frac{\mathrm{RN}(B)}{D}\right)^k (u_0 - r) + r$$

thus giving $\delta = u_{n-1} = \left(\frac{\mathrm{RN}(B)}{D}\right)^{n-1}(\|w\|_\infty - r) + r$ which gives after n iterations on $\|w\|_\infty$

$$\|w\|_\infty \leftarrow \frac{D}{2} + \mathrm{RN}(B) \times \left[\left(\frac{\mathrm{RN}(B)}{D}\right)^{n-1}\left(\|w\|_\infty - \frac{\|w\|_\infty D}{D-\mathrm{RN}(B)}\right) + \frac{\|w\|_\infty D}{D-\mathrm{RN}(B)}\right]$$

$$\|w\|_\infty \leftarrow \frac{D}{2} + \mathrm{RN}(B) \times \left[\left(\frac{\mathrm{RN}(B)}{D}\right)^{n-1}\left(1 - \frac{D}{D-\mathrm{RN}(B)}\right) + \frac{D}{D-\mathrm{RN}(B)}\right]\|w\|_\infty$$

$$\|w\|_\infty \leftarrow \frac{D}{2} + \mathrm{RN}(B) \times \left[\left(\frac{\mathrm{RN}(B)}{D}\right)^{n-1}\frac{-\mathrm{RN}(B)}{D-\mathrm{RN}(B)} + \frac{D}{D-\mathrm{RN}(B)}\right]\|w\|_\infty$$

$$\|w\|_\infty \leftarrow \frac{D}{2} + \mathrm{RN}(B) \times \left[\frac{D - \left(\frac{\mathrm{RN}(B)}{D}\right)^{n-1} \mathrm{RN}(B)}{D - \mathrm{RN}(B)} \right] \|w\|_\infty$$

This shows that after n iterations the new maximum norm can be higher than initially, while decreasing its l_1 norm by the largest possible factor. In this specific example, the next iterations will not repeat that pattern however: the update $\|w\|_\infty \leftarrow \|w\|_\infty + \frac{|w_1|}{D}\mathrm{RN}(B)$ will no longer hold as the values of w_i will no longer be equal and in particular $w_i \leq D/2$ for most i. This observation, however, gives us a framework for an analysis of an average case complexity.

B.2 Average-case complexity

In our previous approach, we expected the max norm to increase after n iterations, using in each iteration the worst-case update $\|w\|_\infty \leftarrow \|w\|_\infty + \frac{|w_1|}{D}\mathrm{RN}(B)$. However, for each position, it is not possible to receive exactly $\frac{\|w\|_\infty}{D}\mathrm{RN}(B)$ and it is much more likely to receive instead

$$|w_i| \leftarrow |w_i| \pm \left(\frac{\|w\|_\infty}{D} \times \frac{\mathrm{RN}(B)}{n} \right)$$

per iteration of `maxPSW`, supposing the noise is somewhat equally distributed in every position. However, it is also possible that in some positions the noise make the value decrease, thus it is hard to determine the exact value with which we can approximate our reductions. In particular, let us denote by f the *noise update* such that our updates in n consecutive iterations goes as

$$\|w\|_\infty \leftarrow \|w\|_\infty + \left(\frac{\|w\|_\infty}{D} \times f \right)$$

Which, when we plug into our previous update after n iterations give us

$$\|w\|_\infty \leftarrow \frac{D}{2} + f \times \left[\frac{D - \left(\frac{f}{D}\right)^{n-1} f}{D - f} \right] \|w\|_\infty$$

and more exactly

$$|w_i| \leftarrow \left\lfloor \frac{w_i}{D} \right\rceil \pm f \times \left[\frac{D - \left(\frac{f}{D}\right)^{n-1} f}{D - f} \right] \|w\|_\infty$$

which is also an arithmetico-geometric sequence, except each term represents the result after n consecutive iterations. This computed approximation on l_∞ show much closer results to the average case than the ones we used in the worst-case approach on l_1. The question is on how to determine the value f: it seems natural that the value must be lower than $\max_{i \neq j}(B_{(i,j)})$ ans it seems to also show experimentally. The correct value for f also seem to be higher than $\min_{i \neq j}(B_{(i,j)})$, and could be dependent on the noise distribution of our samples. We leave the establishment of f as an open question, as it is also not guaranteed that our approach does not have major flaws. At the very least, $f = 0$ do perfectly represent the diagonal matrix with no noise. This approach, which do not rely on the strict diagonal dominance, can also be useful when testing average-case reductions using a matrix that have large diagonal coefficients but do not conform to a strict diagonal dominance fitting the Lévy-Desplanques theorem.

C Reducing the security of the key to the security of the message

To achieve this we define two folklore problems in which our keys and messages are just particular instance of.

Definition 10 (Problem 1: Basis Completion Problem)**.**
Let $D \in \mathrm{M}_n(\mathbb{Z})$, a set $\mathcal{S}_n \subset \mathrm{M}_n(\mathbb{Z})$ and $\mathcal{L}$ a lattice.
Given D and $\mathcal{L}$, find $N \in \mathcal{S}_n$ such that $\mathcal{L} = \mathcal{L}(D - N)$.

Definition 11 (Problem 2: Finding an unique representative)**.**
Let $\mathcal{L}$ a lattice and a non-empty set $\mathcal{M} \subset \mathbb{Z}^n$.
Given $x \in \mathbb{Z}^n$ such that $Y = (\mathcal{L} + x) \cap \mathcal{M}$ and $|Y| = 1$, find $y \in Y$. (i.e $x \equiv y \mod \mathcal{L}$)

Depending on the parameters $\mathcal{S}_n, \mathcal{M}$, above definitions can relate to well-known minimization problems but it is out of this paper's scope. Rather, our main point is the following lemma:

Lemma 9 (Solving problem 2 can solve problem 1)**.**
Let D, $\mathcal{L}$, $\mathcal{M}$ and $\mathcal{S}_n$ as defined above. If $\forall N \in \mathcal{S}_n, \forall i \in [1, n], N_i \in \mathcal{M}$ then being able to solve problem 2 for all instances $x = D_i$ solves problem 1.

Proof. We aim to find N such that $\mathcal{L} = \mathcal{L}(D - N)$. D is known, and $\forall i \in [1, n]$ we have $D_i \equiv N_i \mod \mathcal{L}$. Our conditions state that $N_i \in \mathcal{M}$. Thus, we can recover N_i by solving problem 2 on $x = D_i$: the solution exists, and unicity being enforced by the prerequisite of problem 2 does the rest. □

The definitions and the lemma we gave here are generic, but simple enough to fit some parameter sets for encryption using diagonally dominant lattices (the notations are not coincidental). In crypto-language, it just means being able to recover certain messages allow full recovery of the secret key. As usual, not fitting the above properties does not mean a system is insecure itself, but this is a widely different problematic.