Pseudorandom sequences prediction and Cayley graphs description by means of integer lattices

---

Ph.D. Thesis

Álvar Ibeas Martín

Universidad de Cantabria

July 2008

Advisor: **Jaime Gutiérrez**

Contents

Agradecimientos

En el trabajo de estudio e investigación que se recoge en esta memoria he recibido la ayuda de muchas personas sin la cual no lo habría podido llevar a cabo.

En primer lugar debo nombrar a Jaime Gutiérrez, mi director de tesis, que desde un primer momento me ha *encaminado*, con la mirada puesta siempre hacia adelante, proponiéndome temas y compartiendo reflexiones. Jaime me ha animado durante todo este tiempo, con su permanente disposición e inagotable capacidad para el trabajo.

Quiero mencionar a los compañeros con los que he coincidido en la Universidad de Cantabria, que han favorecido siempre un ambiente amistoso. A Domingo y Pilar, que también trabajan en los temas que recoge esta memoria, y con los que he tenido la suerte de colaborar; a Germán y Luis Felipe, a los que he podido recurrir siempre con problemas informáticos de todo tipo; y a todos los demás, especialmente a Fernando, Jose y Luis.

También quiero reconocer la acogida que he recibido en las estancias breves que he realizado en Linz y Waterloo. Con Arne Winterhof, Josef Schicho y Jose Manuel G. Vallinas pude conocer el original y acogedor espacio de trabajo del insituto RICAM y hacer muchas visitas por los alrededores de Linz. En Waterloo, puede disfrutar gracias a Mark Giesbrecht de una visita a una Universidad puntera en Matemáticas y Computación. Agradezco la hospitalidad de Clément Pernet y de Arne Storjohann, que me alojó en su casa.

Por supuesto, tengo que agradecer a mi familia su apoyo incondicional e insustituible. A mis padres, Domingo y Conchita y a mi hermana Elvira, que seguramente sean los que más me han soportado los malos ratos; a mi abuela Edita, que siempre me ha conocido muy bien; a mis tíos Mª Carmen, Isaac y Olga; y en el recuerdo, a mis abuelos Primo y Carmen, de los que resulta muy sencillo acordarse sólo de lo bueno.

Muchas gracias a todos.

Preamble

Geometry of Numbers is a suggestive name for a theory initiated by H. Minkowski in 1896. One main question which that theory aims to answer is the following. Given a convex body symmetric about the origin, does it contain any point with integer coordinates? The set of points with integer coordinates in the space $\mathbb{R}^m$, and more generally, the images of that set by linear transformations, are the *full-dimensional lattices* of $\mathbb{R}^m$. Lattices arise as a beautiful mathematical object which permits a very intuitive realization of the abstract concept of modulus. Indeed, a lattice is the set of intersection points of a regular grid. Restricting ourselves to the affine plane $\mathbb{R}^2$, a lattice is the set of intersection points of two groups of parallel and equispaced lines. Leaving the case of $\mathbb{R}^2$, for which the common problems concerning lattices are completely solved, that innocent object leads to deep questions about computational complexity.

We have presented lattices as the set of points of a grid, but different grids may define the same lattice, as displayed in Figure 1. Selecting the most convenient grid (the

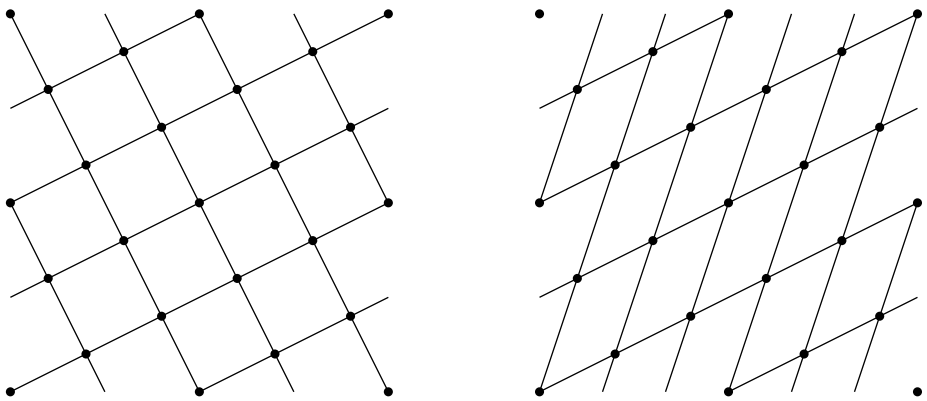


Figure 1: Two grids with the same intersection point set

basis with smaller orthogonality defect, as we will define) is related with the question expressed above. A less general problem is finding a nonzero vector with minimum norm in a lattice, the so-called shortest vector problem (SVP). The celebrated LLL algorithm

described by Lenstra, Lenstra, and Lovász [1982] returns in polynomial time a nonzero vector whose norm is an exponential approximation to that of the shortest one. That algorithm has been a major thrust for the research in Algorithmic Mathematics and Cryptology.

Among the consequences of that algorithm we must mention the factorization of univariate polynomials with rational coefficients and the break of the cryptosystems based on the knapsack problem. Lattice basis reduction techniques seem inherently linear. The general idea is to relate a nonlinear problem to a lattice problem and translate the original problem to finding a vector with minimum Euclidean norm in the associated lattice: the so-called shortest vector problem.

This thesis dissertation explores two separated problems that enrol in the field of Theory of Communication. The first studies the prediction of pseudorandom sequences in Cryptology, and the second the minimum distance diagrams of Cayley digraphs in network architectures. Despite these two problems seem quite different, both use the same mathematical tool, namely lattice theory.

The fundamental objective of Cryptography is to enable two parties, usually referred to as Alice and Bob, to communicate over an insecure channel in such a way that a potential eavesdropper, Eve, cannot understand what is being said. Alice encrypts the plaintext, using a predetermined key, and sends the resulting ciphertext over the channel. Eve, upon seeing the ciphertext in the channel cannot determine what the plaintext was; but Bob, who knows the encryption key, can decrypt the ciphertext and reconstruct the plaintext.

A pseudorandom number generator (PRNG) is an algorithm that from a limited amount of entropy, in the form of a number called seed, generates a stream of bits that may be used as if they were "random". A simple application of pseudorandom sequences in Cryptology is the so-called *stream cipher*, inspired in the *one-time pad* cryptosystem. The latter consists of two parties sharing a secret key with the same size as the message to be sent. The i-th bit of the ciphertext is directly obtained by combining the i-th bits of the plaintext and the secret key. This famous system is proved to have the *perfect secrecy*, as defined by C.E. Shannon [1949]. This means that in order to guess the clear message, an eavesdropper with access to the ciphertext has no advantage over another who has not intercepted the channel's traffic. The necessity of sharing a so long secret key reduces the practical applicability of this cryptosystem. The stream cipher replace the secret key of the one-time pad by a keystream produced by a PRNG.

We consider an eavesdropper performing a "known plaintext" type attack. More precisely, we assume that the cryptanalyst has access to a certain piece of the original message and that the channel is insecure and therefore, he has also access to the whole ciphertext. It is quite realistic to consider this scenario, for it may be easy for the eavesdropper to guess some common parts of the plaintext, as a heading composed by an address, a date, customary greetings, etc. In a stream cipher, a known plaintext attack reveals a certain part of the keystream. Therefore, the used pseudorandom number generator should not be reproducible from a sample of its output.

Most of the results and applications of lattice theory are related to the Euclidean norm ℓ_2. But other norms are also important. The norm ℓ_∞ is the natural norm for problems in integer programing. A remarkable paper of C.P. Schnorr [1993] reduces

the problem of factoring integers to the problem of finding a vector of minimum norm ℓ_1. In fact, Lovász and Scarf [1992] proposed a LLL-type reduction algorithm for an arbitrary norm. The special case of two dimensional lattices has been widely studied. We are obliged to remark the article [Kaib and Schnorr, 1996], which generalizes the Gauss algorithm for two-dimensional lattices to an arbitrary norm in polynomial time.

The theoretical foundations of computer network modeling are based on graph theory, where processors are represented by the nodes of the graph and communication links by its edges. The Cayley digraphs of cyclic groups, also called circulant graphs or multi-loop networks, have a wide variety of applications on telecommunication networks, VLSI design and distributed computing. One of the fundamental problems in networks is finding effective ways for directing a message between each pair of processors in an optimal way, which in graph theory is equivalent to computing the minimum path between two nodes.

There is a very rich literature with focuses on directed and undirected circulant graphs, addressing computational and theoretical problems. The concept of "L"-shape, introduced by Wong and Coppersmith [1974] for digraphs of degree two facilitates the computation of two graph's parameters: the diagram and the average distance. In this thesis we propose the use of monomial ideals in order to generalize that object to digraphs of arbitrary degree. We mainly use two computational tools, namely Gröbner bases and lattice theory with ℓ_1 norm to solve the routing problem and build minimum distance diagrams.

Here is a brief synopsis of the contents of this thesis.

Chapter I is a relatively small introduction to lattices theory, including the most interesting computational problems: shortest vector problem and closest vector problem, and the LLL basis reduction. The last section is dedicated to introduce some results about lattice ideals.

Chapter II deals with several results on predicting pseudorandom number generators from partial information. It contains an introduction to the problem, then Section 2.1 studies the quadratic generator and, in particular, the famous Pollard generator. Section 2.2 analyses the linear congruential generator over elliptic curves.

Chapter III is devoted to the study of minimum distance diagrams of Cayley digraphs. It is divided into nine sections. After presenting notations, definitions and preliminary results, Section 3.2 provides several results on minimum distance diagrams. Then, Section 3.3 connects lattices ideals and those diagrams. In Section 3.4, we show an algorithm for the routing problem. Section 3.5 contains an algorithm for computing a MDD for triple-loop computer networks. Section 3.6 provides formulae for computing the diameter and the average distance. Finally, we propose a family of circulant graphs with a degenerated MDD and a relatively small diameter.

Chapter IV comments the implementations in C++ and some numerical results of the algorithms presented in Chapter II and present the software Circule, which draws figures of circulant digraphs and associated diagrams.

Finally, The **Further work** section is dedicated to questions that are still open and future lines of investigation.

Chapter I

Introduction to lattices

In this chapter we review the definition and some basic facts about lattice theory, which is used as a main tool in the following two chapters. We explain the LLL reduction algorithm and the concept of binomial ideal associated with an integer lattice.

We are going to gather some well-known results about point lattices that we will use in the following. The literature about this topic is rich, some enlightening references are [Cassels, 1997; Grötschel et al., 1993; Gruber and Lekkerkerker, 1987]. A point lattice consists of vectors in the space $\mathbb{R}^m$. We employ bold Latin letters to denote vectors and the notation $\langle \mathbf{u}, \mathbf{v} \rangle$ for the standard inner product. We also use the notation $\mathrm{B}_{\|\cdot\|}(\varepsilon)$ for the ball of the norm $\|\cdot\|$ with radius ε centered at the origin, omitting the norm $\|\cdot\|$ when referring to the Euclidean one:

$$\mathrm{B}(\varepsilon) := \{\mathbf{u} \in \mathbb{R}^m \mid \langle u, u \rangle < \varepsilon^2\}.$$

Vaguely, a lattice can be defined as the set of intersection points of an infinite and regular grid. In a precise way, we say that a lattice is a discrete subgroup of $(\mathbb{R}^m, +)$, i.e.:

Definition 1.1 *A nonempty set $\Lambda \subset \mathbb{R}^m$ is called lattice if:*

- $\mathbf{a}, \mathbf{b} \in \Lambda \Rightarrow \mathbf{a} - \mathbf{b} \in \Lambda$.
- $\exists \varepsilon > 0 \ \mid \ \mathrm{B}(\varepsilon) \cap \Lambda = \{0\}$.

In particular, a lattice is a free $\mathbb{Z}$-module with finite rank, inheriting then the concept of basis, that is redefined below. Throughout the thesis, we often write matrices as a concatenation of its column vectors: $A = [\mathbf{a}_1 | \cdots | \mathbf{a}_n]$.

Definition 1.2 *For matrix $A = [\mathbf{a}_1 | \cdots | \mathbf{a}_n] \in \mathbb{R}^{m \times n}$, we let*

$$\mathcal{L}(A) := \{A\mathbf{x} \mid \mathbf{x} \in \mathbb{Z}^n\} \subset \mathbb{R}^m$$

denote the set of integral linear combinations of its column vectors. Matrix $B \in \mathbb{R}^{m \times n}$ is a basis for lattice Λ if $\operatorname{rank}(B) = n$ and $\Lambda = \mathcal{L}(B)$. We also say that the set of column vectors of B is a basis for Λ.

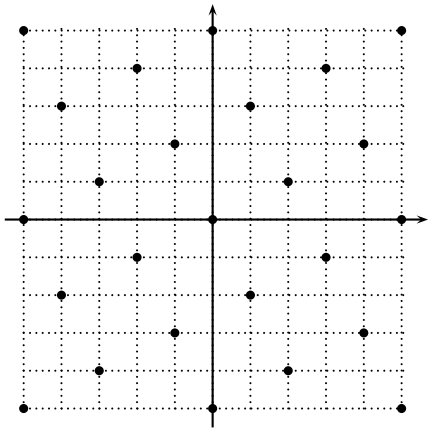


Figure 1.1: $\{(x, y) \in \mathbb{Z}^2 \mid 2x + y \equiv_5 0\}$

A basis is a natural way for representing a lattice, for instance if one needs to use it as the input of an algorithm. This can be done because every lattice admits a basis. On the other hand, every set of linearly independent vectors $\{\mathbf{b}_1, \ldots, \mathbf{b}_n\}$ is a basis for the lattice $\mathcal{L}([\mathbf{b}_1|\cdots|\mathbf{b}_n])$. Moreover, for any rational matrix A, $\mathcal{L}(A)$ is a lattice, although it need not be so for arbitrary real matrix A. There are several bases for any fixed lattice Λ, all of them with the same number of elements (columns). This number is called the *dimension* or *rank* of the lattice, and denoted by $\dim(\Lambda)$. As an example, the columns of matrix

$$A = \left[\begin{array}{c|c|c} 4 & 0 & 5 \\ 2 & 5 & 0 \end{array}\right]$$

span the two-dimensional lattice Λ depicted in Figure 1.1, and $\{(2, 1), (1, 3)\}$ and $\{(2, 1), (-1, 2)\}$ are bases for it, whereas no pair of column vectors from A form a basis.

For a lattice Λ with basis B_1, matrix B_2 is another basis for Λ if and only if the matrix of change P such that $B_1 P = B_2$ is square, has integer coefficients, and its determinant is ± 1, i.e., it is a *unimodular matrix*. In this case, we say that B_1 and B_2 are *equivalent* bases.

The volume is a lattice invariant with a simple interpretation related to the informal definition of lattices as grids. Let Λ be a lattice with basis $\{\mathbf{b}_1, \ldots, \mathbf{b}_n\}$. The lines with direction $\mathbf{b}_i$ through the points of Λ form a grid whose set of intersection points is precisely Λ. Changing the basis leads to a different grid, whose cells' measure is the same. That invariant measure is called the *volume* of a lattice (see Figure 1.2). This definition can be formalized with the following notation:

Definition 1.3 *Let $B \in \mathbb{R}^{m \times n}$ be an n-rank matrix. Its associated fundamental parallelepiped is*

$$\mathcal{P}(B) := \left\{ \sum_{i=1}^{n} \alpha_i \mathbf{b}_i \mid 0 \leq \alpha_i < 1 \right\}.$$

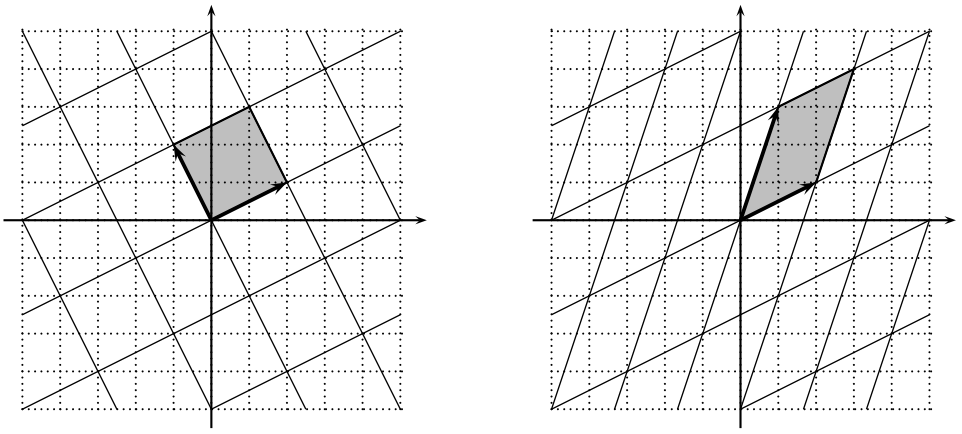


Figure 1.2: The volume is an invariant of a lattice.

Let μ denote the Lebesgue measure in an n-dimensional space. A basis B determines a positive-definite quadratic form, whose associated matrix is B^tB, called Grammian. The measure of the fundamental parallelepiped associated to B is the square root of the determinant of its associated Grammian form:

$$\mu(\mathcal{P}(B)) = \sqrt{\det(B^tB)}.$$

As a unimodular transformation on B preserves the determinant of B^tB, we can define the *volume* or *determinant* of a lattice as the measure of the fundamental parallelepiped associated with any basis. We employ the notation $\mathrm{vol}(\Lambda)$ for this concept. If the bases of a lattice $\mathcal{L}(B)$ are square matrices ($n = m$, with our previous notation), the lattice is called *full dimensional* and its volume equals $|\det(B)|$. The volume of a lattice is inversely proportional to its density in the space $\mathbb{R}\langle\Lambda\rangle = \mathbb{R}\langle\mathbf{b}_1, \ldots, \mathbf{b}_n\rangle$, i.e.:

$$\mathrm{vol}(\Lambda) = \lim_{k\to\infty} \frac{\mu(\mathrm{B}(k) \cap \mathbb{R}\langle\Lambda\rangle)}{\#(\Lambda \cap \mathrm{B}(k))}.$$

The history of this theory goes back to the problem of classifying quadratic forms [Gauss, 1966]. In short, the problem of finding the smallest image of integer vectors through a (positive-definite) quadratic form $q(X_1, \ldots, X_n) = q(\mathbf{x})$ is reduced to obtaining the shortest nonzero vector in a lattice by rewriting $q(\mathbf{x}) = \mathbf{x}^tB^tB\mathbf{x} = \|B\mathbf{x}\|_2^2$, where $\|\mathbf{u}\|_2$ is the Euclidean norm $\langle\mathbf{u}, \mathbf{u}\rangle^{1/2}$.

Lattices are very interesting objects from the complexity point of view, particularly since the discovering of the LLL reduction [Lenstra et al., 1982]. For algorithmic issues, we consider exclusively integer lattices (i.e., those contained in $\mathbb{Z}^m$).

Definition 1.4 (Shortest Vector Problem, SVP) *Given a basis $B \in \mathbb{Z}^{m\times n}$ for lattice Λ and a norm $\|\cdot\|$ in $\mathbb{R}^m$, find $\mathbf{u} \in \Lambda\backslash\{0\}$ such that $\|\mathbf{u}\| \leq \|\mathbf{v}\|$, for all $\mathbf{v} \in \Lambda\backslash\{0\}$.*

This problem is easily proved to be NP-hard when referred to the ℓ_∞ norm, and believed to be so for any ℓ_p norm with $p \geq 1$. Although the latter has not been proved for deterministic (Karp) reductions, that problem is NP-hard under randomized reductions [Ajtai, 1998]. A comprehensive reference for complexity problems on lattices is [Micciancio and Goldwasser, 2002].

The first approach for solving this problem takes as input a lattice, expressed by a basis, and transforms this basis into an equivalent one, containing a shortest nonzero element of the lattice as a basis vector. In a more general way, one in interested in "reducing" a basis, or finding an equivalent one with more convenient properties, mainly, with shorter vectors.

Definition 1.5 *For lattice Λ, the i-th minimum of Minkowski is the smallest real number $\lambda = \lambda_i(\Lambda)$ such that there exist i linearly independent elements in $\Lambda \cap B(\lambda)$.*

H. Minkowski introduced a new geometric point of view, by regarding lattice bases as convex bodies. If a lattice is generated by basis $B \in \mathbb{R}^{m\times n}$, the inverse image of the unit closed ball $\bar{\mathrm{B}}(1) \subset \mathbb{R}^m$ through mapping B is a convex and symmetric about the origin body K (see Figure 1.3). Vector $\mathbf{x} \in \mathbb{R}^n$ lies in the interior of K if and only if $\|B\mathbf{x}\|_2 < 1$. In this way, K defines a norm in $\mathbb{R}^n$ by:

$$\|\mathbf{x}\|_K := \min\{\alpha \geq 0 \ \mid \ \mathbf{x} \in \alpha K\},$$

where $\alpha K := \{\alpha\mathbf{x} \mid \mathbf{x} \in K\}$ is the dilatation of K with factor α.

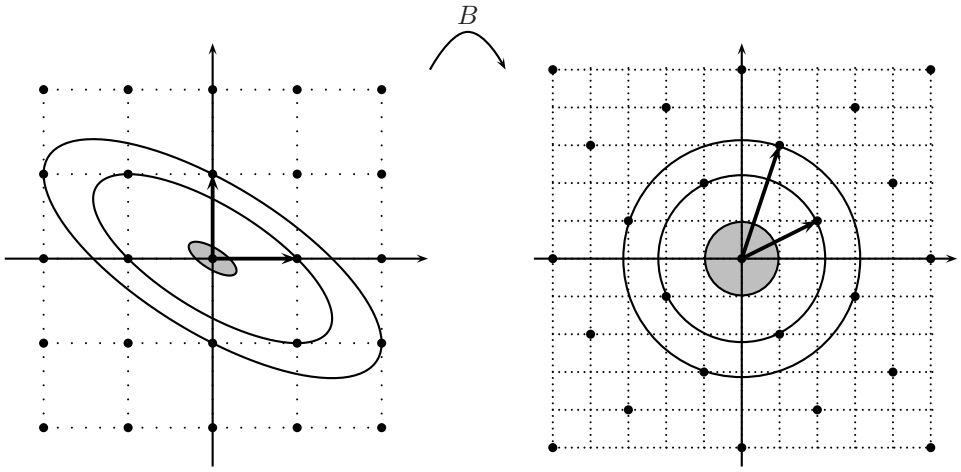


Figure 1.3: The convex body associated to basis $\{(2,1),(1,3)\}$

The bijection from points with integer coordinates in $\mathbb{Z}^n$ into elements of $\mathcal{L}(B)$, transforms $\|\cdot\|_K$ norm in $\mathbb{R}^n$ into ℓ_2 norm in $\mathbb{R}^m$. This idea links the search of short vectors in a lattice with the theory called Geometry of Numbers, in which convex and symmetric about the origin bodies are studied. We state now the two basic results in this theory.

Theorem 1.6 (Blichfeldt) *Let Λ be a lattice with $\dim(\Lambda) = n$ and $\mathbb{R}\langle\Lambda\rangle$ the $\mathbb{R}$-linear space generated by Λ. If $S \subseteq \mathbb{R}\langle\Lambda\rangle$ is a measurable set with $\mu(S) > \mathrm{vol}(\Lambda)$, then there are two distinct points in S such that their difference lies in Λ.*

This result appeared in [Blichfeldt, 1914] has a simple proof (sketched in Figure 1.4) due to G.D. Birkhoff that uses the so-called "pigeonhole principle" on the intersections of the set S with the cells of a grid associated to Λ. It yields the following theorem:

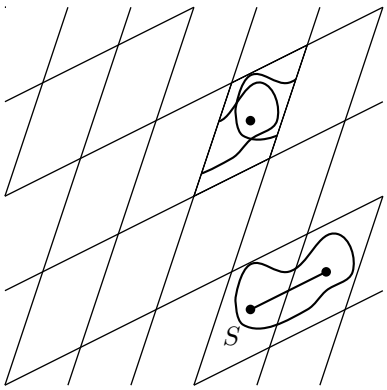


Figure 1.4: There must be two points in S with the same image modulo Λ.

Theorem 1.7 (Minkowski) *Let Λ be a lattice with $\dim(\Lambda) = n$ and $\mathbb{R}\langle\Lambda\rangle$ the linear space it generates. If $S \subseteq \mathbb{R}\langle\Lambda\rangle$ is convex, symmetric about the origin, and measurable with $\mu(S) > 2^n \mathrm{vol}(\Lambda)$, then $S \cap \Lambda \neq \{0\}$, and therefore there are at least three points in that intersection.*

The bound provided in Theorem 1.7 is sharp, as can be seen with the convex body $S = \{\sum_{i=1}^{n} \alpha_i \mathbf{b}_i \;\mid\; \alpha_i \in (-1, 1)\}$. It implies the following estimation for the shortest nonzero element in an n-dimensional lattice Λ, that is valid for every norm:

$$\lambda_1(\Lambda) \leq 2 \left(\frac{\mathrm{vol}(\Lambda)}{\mu(\mathrm{B}_{\|\cdot\|}(1))} \right)^{1/n}.$$

When particularized to the Euclidean norm, we get:

$$\lambda_1(\Lambda) \leq \frac{2\left(\Gamma(1 + n/2)\right)^{1/n}}{\sqrt{\pi}} (\mathrm{vol}\,\Lambda)^{1/n},$$

which by Stirling's formula gives

$$\lambda_1(\Lambda) \leq \left(\sqrt{\frac{2n}{e\pi}} + O(1) \right) (\mathrm{vol}\,\Lambda)^{1/n}.$$

Shortly, we get $\lambda_1(\Lambda) = O(n^{1/2})(\mathrm{vol}\,\Lambda)^{1/n})$. The Gaussian heuristic suggests that we often cannot find substantially shorter nonzero vectors in a lattice. Next problem is the inhomogeneous version of SVP.

Definition 1.8 (Closest Vector Problem, CVP) *Given a basis $B \in \mathbb{Z}^{m\times n}$ for lattice Λ, a vector $\mathbf{s} \in \mathbb{R}^m$, and a norm $\|\cdot\|$ in $\mathbb{R}^m$, find $\mathbf{u} \in \mathbf{s}+\Lambda$ such that $\|\mathbf{u}\| \leq \|\mathbf{v}\|$, for all $\mathbf{v} \in \mathbf{s}+\Lambda$.*

This problem is usually formulated in the following (and equivalent) form: finding the closest vector to a target vector $-\mathbf{s}$ among all the elements of an input lattice. It is NP-hard for any ℓ_p norm, with $p \geq 1$, including $p = \infty$ [Kannan, 1987; van Emde Boas, 1981].

Assuming that we want to solve some problem like SVP, we can be given a "bad-quality" basis B for which some short norm vector $\mathbf{v}$ lies in $\mathcal{L}(B)$ although every vector $\mathbf{b}_i$ in B has a higher norm. Then, the usual approach is to perform some operations on the vectors $\mathbf{b}_i$ until reaching a suitable basis that includes $\mathbf{v}$. For a given lattice, bases with smaller vectors are those nearer to orthogonal, as can be seen analyzing the following concept:

Definition 1.9 *For basis $B = [\mathbf{b}_1|\cdots|\mathbf{b}_n] \in \mathbb{R}^{m\times n}$, its orthogonality defect is:*

$$\mathrm{od}(B) = \frac{\prod_{i=1}^{n}\|\mathbf{b}_i\|}{\mathrm{vol}(\mathcal{L}(B))} = \prod_{i=2}^{n}\sin^{-1}(\alpha_i),$$

where α_i is the angle between $\mathbf{b}_i$ and the space $\mathbb{R}\langle\mathbf{b}_1,\ldots,\mathbf{b}_{i-1}\rangle$.

The orthogonality defect of a basis is greater or equal than 1, and the equality holds for orthogonal bases. The process of finding a basis with smaller orthogonality defect than the given one is called *reduction* and has an optimal solution for two-rank lattices.

Lattices are intimately related to Cryptology. In a first stage, lattice basis reduction was used to break several cryptosystems based on knapsacks, a well-known NP-hard problem. See for [Odlyzko, 1990] a review on those attacks and [Joux and Stern, 1998; Nguyen and Stern, 2001] for a general perspective on the role played by lattices in Cryptology. Since the publication of [Coppersmith, 1996, 1997], lattice basis reduction has extended its influence in Cryptanalysis to nonlinear problems, like for instance, some attacks on RSA cryptosystem. A very interesting complexity result set the base of lattices' jump to the "positive" side: Cryptography. M. Ajtai proved in [Ajtai, 1996] a transference result which relates the complexity of SVP in the worst and average cases. Summing up, one can build a probability distribution of lattice bases for which SVP is essentially as hard on the average as the worst case. This kind of result is precisely what is needed to provide a security proof based on stated complexity conjectures for a cryptosystem.

1.1 Two-dimensional lattices

Let Λ be a two-dimensional lattice, i.e., $\Lambda = \mathcal{L}(B)$, for $B \in \mathbb{R}^{m\times 2}$ a two-rank matrix. And let $\|\cdot\|$ be a norm in $\mathbb{R}^m$. Next definition gives a simple characterization of bases of our interest:

Definition 1.10 *A basis $B = [\mathbf{b}_1|\mathbf{b}_2] \in \mathbb{R}^{m\times 2}$ is called reduced if the following holds:*

$$\|\mathbf{b}_1\| \leq \|\mathbf{b}_2\| \leq \|\mathbf{b}_1 - \mathbf{b}_2\|, \|\mathbf{b}_1 + \mathbf{b}_2\|.$$

Originally conceived for the Euclidean norm, this definition can be referred to any norm in $\mathbb{R}^m$. As is shown below, any two-dimensional lattice admits a reduced basis. Moreover, we can compute it efficiently. The importance of this concept arises from next result, that sheds a light on the accuracy of the name "reduced" for Definition 1.10.

Proposition 1.11 *For $B = [\mathbf{b}_1|\mathbf{b}_2] \in \mathbb{R}^{m\times 2}$ and norm $\|\cdot\|$, the following are equivalent:*

- *B is reduced.*
- *$\|\mathbf{b}_i\| = \lambda_i(\mathcal{L}(B))$, for $i = 1, 2$.*

Using Algorithm 1.1, we can solve in polynomial time SVP for the Euclidean norm in any lattice of rank two. That algorithm can be considered as an extension of the centralized Euclidean algorithm, substituting the search of the smallest element in a "discrete line" $\mathbf{b}_1+\mathbb{Z}\mathbf{b}_2$ for the remainder operation. That search, represented in Figure 1.5, is a discrete variant of the Gram-Schmidt orthogonalization in which the bigger vector $\mathbf{b}_2$ is reduced by the smaller one $\mathbf{b}_1$, selecting a new vector $\mathbf{b}'_2$ in the stripe $\{\mathbf{u} \in \mathbb{R}^m \mid 2|\langle \mathbf{u}, \mathbf{b}_1\rangle| \leq \langle \mathbf{b}_1, \mathbf{b}_1\rangle\}$. After this step, we obtain a new basis $\{\mathbf{b}_1, \mathbf{b}'_2\}$, for the performed operation is unimodular. It satisfies:

$$\|\mathbf{b}'_2\|_2 \leq \|\mathbf{b}_1 + \mathbf{b}'_2\|_2, \|\mathbf{b}_1 - \mathbf{b}'_2\|_2.$$

So if the obtained vector $\mathbf{b}'_2$ remains bigger than $\mathbf{b}_1$, we have reached a reduced basis. In other case, we can swap these vectors and iterate.

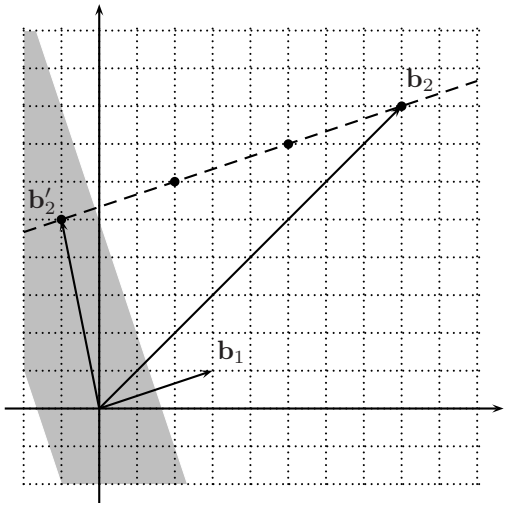


Figure 1.5: A step in Gauss' algorithm.

The number of steps performed by Algorithm 1.1 can be roughly estimated as smaller than $2 + \log_2(\|\mathbf{b}_1\|_2 + \|\mathbf{b}_2\|_2)$. An almost optimal bound is provided in [Vallée, 1991], characterizing as well the pairs of input vectors that lead to a given number of iterations. Algorithm 1.1 obtains a reduced basis with respect to the ℓ_2 norm. It was modified in [Kaib and Schnorr, 1996] to deal with any computable norm.

Next results present further properties of a basis satisfying Definition 1.10. We use them in Chapter II.

Algorithm 1.1: Gauss reduction algorithm

Input: $\mathbf{b}_1, \mathbf{b}_2 \in \mathbb{Z}^m$, linearly independent.
Output: A reduced basis for $\mathcal{L}([\mathbf{b}_1|\mathbf{b}_2])$ with respect to ℓ_2 norm.

if $\|\mathbf{b}_2\|_2 > \|\mathbf{b}_1\|_2$ **then**
  swap$(\mathbf{b}_1, \mathbf{b}_2)$.
repeat
  swap$(\mathbf{b}_1, \mathbf{b}_2)$
  $\alpha \leftarrow \left\lfloor \frac{\langle \mathbf{b}_2, \mathbf{b}_1 \rangle}{\langle \mathbf{b}_1, \mathbf{b}_1 \rangle} \right\rceil$
  $\mathbf{b}_2 \leftarrow \mathbf{b}_2 - \alpha \mathbf{b}_1$
until $\|\mathbf{b}_1\|_2 \leq \|\mathbf{b}_2\|_2$
return $[\mathbf{b}_1|\mathbf{b}_2]$

Algorithm 1.2: Kaib-Schnorr extension for Gauss algorithm

Input: $\mathbf{b}_1, \mathbf{b}_2 \in \mathbb{Z}^m$, linearly independent.
Output: A reduced basis for $\mathcal{L}([\mathbf{b}_1|\mathbf{b}_2])$.

if $\|\mathbf{b}_1\| > \|\mathbf{b}_2\|$ **then**
  swap$(\mathbf{b}_1, \mathbf{b}_2)$.
if $\|\mathbf{b}_1 - \mathbf{b}_2\| > \|\mathbf{b}_1 + \mathbf{b}_2\|$ **then**
  $\mathbf{b}_2 \leftarrow -\mathbf{b}_2$
if $\|\mathbf{b}_2\| \leq \|\mathbf{b}_1 - \mathbf{b}_2\|$ **then**
  return $[\mathbf{b}_1|\mathbf{b}_2]$
if $\|\mathbf{b}_1\| \leq \|\mathbf{b}_1 - \mathbf{b}_2\|$ **then**
  goto 12
if $\|\mathbf{b}_1\| = \|\mathbf{b}_2\|$ **then**
  return $[\mathbf{b}_1|\mathbf{b}_1 - \mathbf{b}_2]$
$[\mathbf{b}_1|\mathbf{b}_2] \leftarrow [\mathbf{b}_2 - \mathbf{b}_1|\mathbf{b}_2]$
while $\|\mathbf{b}_2\| \geq \|\mathbf{b}_1 - \mathbf{b}_2\|$ **do**
  Find α such that $\|\mathbf{b}_2 - \alpha \mathbf{b}_1\|$ is minimal
  $\mathbf{b}_2 \leftarrow \mathbf{b}_2 - \alpha \mathbf{b}_1$
  if $\|\mathbf{b}_1 - \mathbf{b}_2\| > \|\mathbf{b}_1 + \mathbf{b}_2\|$ **then**
    $\mathbf{b}_2 \leftarrow -\mathbf{b}_2$
  swap$(\mathbf{b}_1, \mathbf{b}_2)$
end
return $[\mathbf{b}_1|\mathbf{b}_2]$

Lemma 1.12 *Let $\{\mathbf{b}_1, \mathbf{b}_2\} \subset \mathbb{R}^m$ be a reduced basis of a two-rank lattice Λ and $\mathbf{x} \in \Lambda$. The unique integers $\alpha, \beta \in \mathbb{Z}$ such that $\mathbf{x} = \alpha\mathbf{b}_1 + \beta\mathbf{b}_2$ also satisfy:*

$$\|\alpha\mathbf{b}_1\|, \|\beta\mathbf{b}_2\| \leq \frac{2}{\sqrt{3}}\|\mathbf{x}\|.$$

We include the simple proof of this lemma, for we have not found it in the literature.
Proof. Firstly, as the basis is reduced, we have:

$$\langle \mathbf{b}_1, \mathbf{b}_1\rangle, \langle \mathbf{b}_2, \mathbf{b}_2\rangle \leq \begin{cases} \langle \mathbf{b}_1, \mathbf{b}_1\rangle + \langle \mathbf{b}_2, \mathbf{b}_2\rangle + 2\langle \mathbf{b}_1, \mathbf{b}_2\rangle \\ \langle \mathbf{b}_1, \mathbf{b}_1\rangle + \langle \mathbf{b}_2, \mathbf{b}_2\rangle - 2\langle \mathbf{b}_1, \mathbf{b}_2\rangle, \end{cases}$$

and so, $2|\langle \mathbf{b}_1, \mathbf{b}_2\rangle| \leq \langle \mathbf{b}_1, \mathbf{b}_1\rangle, \langle \mathbf{b}_2, \mathbf{b}_2\rangle$. This implies

$$\frac{|\langle \mathbf{b}_1, \mathbf{b}_2\rangle|}{\|\mathbf{b}_1\|\|\mathbf{b}_2\|} \leq \frac{1}{2},$$

i.e., the angle between $\mathbf{b}_1$ and $\mathbf{b}_2$ is greater than $\pi/3$. Once this fact is seen,

$$\frac{\langle \alpha\mathbf{b}_1, \alpha\mathbf{b}_1\rangle}{\langle \alpha\mathbf{b}_1 + \beta\mathbf{b}_2, \alpha\mathbf{b}_1 + \beta\mathbf{b}_2\rangle} = \frac{\alpha^2\langle \mathbf{b}_1, \mathbf{b}_1\rangle}{\alpha^2\langle \mathbf{b}_1, \mathbf{b}_1\rangle + \beta^2\langle \mathbf{b}_2, \mathbf{b}_2\rangle + 2\alpha\beta\langle \mathbf{b}_1, \mathbf{b}_2\rangle}.$$

The minimum of the denominator function in variable β is attained at

$$\beta = \frac{-\alpha\langle \mathbf{b}_1, \mathbf{b}_2\rangle}{\langle \mathbf{b}_2, \mathbf{b}_2\rangle}.$$

Thus,

$$\begin{aligned} \frac{\langle \alpha\mathbf{b}_1, \alpha\mathbf{b}_1\rangle}{\langle \alpha\mathbf{b}_1 + \beta\mathbf{b}_2, \alpha\mathbf{b}_1 + \beta\mathbf{b}_2\rangle} &\leq \frac{\langle \mathbf{b}_1, \mathbf{b}_1\rangle\langle \mathbf{b}_2, \mathbf{b}_2\rangle}{\langle \mathbf{b}_1, \mathbf{b}_1\rangle\langle \mathbf{b}_2, \mathbf{b}_2\rangle - \langle \mathbf{b}_1, \mathbf{b}_2\rangle^2} = \\ &= \left(1 - \frac{\langle \mathbf{b}_1, \mathbf{b}_2\rangle^2}{\|\mathbf{b}_1\|^2\|\mathbf{b}_2\|^2}\right)^{-1} \leq \frac{4}{3}, \end{aligned}$$

and the same result holds for $\|\beta\mathbf{b}_2\|$. ■

The provided bound $2/\sqrt{3}$ is tight, as illustrated with the following example. Consider the lattice generated by the reduced basis

$$\{\mathbf{b}_1 = (1, 0), \mathbf{b}_2 = (1/2, \sqrt{3}/2)\}.$$

We have:

$$2\mathbf{b}_1 - \mathbf{b}_2 = (3/2, -\sqrt{3}/2),\ \|2\mathbf{b}_1 - \mathbf{b}_2\| = \sqrt{3}.$$

$$\|-\mathbf{b}_2\| = 1 < \|2\mathbf{b}_1\| = 2 = \frac{2}{\sqrt{3}}\sqrt{3}.$$

Lemma 1.13 *Let $\{\mathbf{b}_1, \mathbf{b}_2\} \subset \mathbb{R}^m$ be a reduced basis of a two-rank lattice Λ. Then we have:*

$$\mathrm{vol}(\Lambda) \leq \|\mathbf{b}_1\|\|\mathbf{b}_2\| \leq \frac{2}{\sqrt{3}}\mathrm{vol}(\Lambda).$$

Proof. The first inequality is immediate. For the second one, we must simply note that $\mathrm{vol}(\Lambda) = \|\mathbf{b}_1\|\|\mathbf{b}_2\||\sin(\widehat{\mathbf{b}_1, \mathbf{b}_2})|$ and by the proof given in Lemma 1.12, $|\sin(\widehat{\mathbf{b}_1, \mathbf{b}_2})| \geq \sqrt{3}/2$. ■

1.2 LLL reduction

The neat situation of the two-rank case does not extend to higher dimensions, as there is not a canonical definition for reduced basis as Definition 1.10. That sort of bases should present a small orthogonality defect, or at least provide a certain approximation for the shortest nonzero vector in the lattice. Moreover, there should exist an efficient algorithm for computing a basis of this kind for every lattice.

For low-dimensional lattices, [Nguyen and Stehlé, 2004] studies a greedy reduction algorithm that generalizes the Gauss reduction. In the general case, several definitions have been proposed, being Minkowski reduced, Korkine-Zolotarev reduced [Korkine and Zolotareff, 1873] and LLL reduced the more relevant.

The invention of the latter in [Lenstra et al., 1982] led to the first polynomial time algorithm for factoring polynomials with rational coefficients. We can obtain efficiently a basis that is reduced in this sense, but the obtained approximation of the shortest nonzero element is exponential in the lattice dimension (see Proposition 1.16). However, this is enough to solve theoretically many questions, and moreover, the empirical behaviour is usually better than expected.

For linearly independent vectors $\mathbf{b}_1, \ldots, \mathbf{b}_n$, we use the notation π_i for the orthogonal projection from the linear space they span onto the orthogonal complement of $\mathbb{R}\langle \mathbf{b}_1, \ldots, \mathbf{b}_{i-1} \rangle$.

Definition 1.14 (Gram-Schmidt) *For linearly independent vectors $\mathbf{b}_1, \ldots, \mathbf{b}_n$, the i-th orthogonalized vector is defined as $\pi_i(\mathbf{b}_i)$. These vectors can be recursively obtained by the formula:*

$$\mathbf{b}_i^* = \mathbf{b}_i - \sum_{j=1}^{i-1} \mu_{i,j} \mathbf{b}_j^*, \text{ where } \mu_{i,j} := \frac{\langle \mathbf{b}_i, \mathbf{b}_j^* \rangle}{\langle \mathbf{b}_j^*, \mathbf{b}_j^* \rangle}.$$

Obviously, from a basis B for lattice $\mathcal{L}(B)$, the orthogonalized family B^* need not span the same lattice, although both span the same linear space. Nevertheless, we have that $\mathrm{vol}(\mathcal{L}(B)) = \prod_{i=1}^{n} \|\mathbf{b}_i^*\|$.

We give below the definition of LLL-reduced bases. It depends on a parameter $1/4 < \delta < 1$, that was originally selected to be $3/4$.

Definition 1.15 (LLL) *A basis $B \in \mathbb{R}^{m \times n}$ is called LLL-reduced with parameter δ if*

- *The Gram-Schmidt coefficients satisfy $|\mu_{i,j}| \leq 1/2$.*
- *$\delta \|\pi_i(\mathbf{b}_i)\|^2 \leq \|\pi_i(\mathbf{b}_{i+1})\|^2$, for $i = 1, \ldots, n-1$.*

Last definition particularizes to Definition 1.10 when $n = 2, \delta = 1$. In the general case, the bigger δ is selected, the closest to orthogonal the output basis is guaranteed to be. However, $\delta < 1$ is required to prove the computational efficiency. Algorithm 1.3 computes a reduced basis for any lattice in polynomial time in the dimension and the bit size of the basis' coefficients.

The algorithm keeps track of a parameter l, indicating that after each step, vectors $\mathbf{b}_1, \ldots, \mathbf{b}_{l-1}$ form an LLL-reduced basis. This is trivial for $l = 2$, from where the algorithm starts. Steps 4-7 perform a weak form of reduction to guarantee $|\mu_{i,j}| < 1/2$,

Algorithm 1.3: LLL reduction algorithm

Input: $\mathbf{b}_1, \ldots, \mathbf{b}_n \in \mathbb{Z}^m$, $1/4 < \delta < 1$.
Output: A LLL reduced basis for $\mathcal{L}([\mathbf{b}_1|\cdots|\mathbf{b}_n])$ with parameter δ.

1 $l \leftarrow 2$
2 Compute the Gram-Schmidt coefficients $\mu_{i,j}$ and the orthogonalized vectors $\mathbf{b}_i^*$
3 **while** $l \leq n$ **do**
4 **for** $i = l-1, \ldots, 1$ **do**
5 $\mathbf{b}_l \leftarrow \lfloor \mu_{l,i} \rceil \, \mathbf{b}_i$
6 Actualize coefficients $\mu_{l,j}$ (for $j = 1, \ldots, i-1$)
7 **end**
8 **if** $\delta \langle \mathbf{b}_{l-1}^*, \mathbf{b}_{l-1}^* \rangle > \mu_{l,l-1}^2 \langle \mathbf{b}_{l-1}^*, \mathbf{b}_{l-1}^* \rangle + \langle \mathbf{b}_l^*, \mathbf{b}_l^* \rangle$ **then**
9 swap$(\mathbf{b}_{l-1}, \mathbf{b}_l)$
10 Actualize coefficients $\mu_{i,j}$ and vectors $\mathbf{b}_i^*$
11 **if** $l > 2$ **then**
12 $l \leftarrow l - 1$
13 **end**
14 **else**
15 $l \leftarrow l + 1$
16 **end**
17 **end**
18 **return** $[\mathbf{b}_1|\cdots|\mathbf{b}_n]$

for $i < l$. Then, Step 8 checks whether the other condition from Definition 1.15 is satisfied or not. If it is not, the last two vectors are exchanged and the algorithm goes back to a smaller dimension. The basic point in the complexity analysis, that guarantees the polynomial behaviour, relies on the quantity:

$$D := \prod_{i=1}^{n} \mathrm{vol}\left(\mathcal{L}(\mathbf{b}_1, \ldots, \mathbf{b}_i)\right)^2 = |B_i^t B_i| \in \mathbb{Z},$$

where $B_i := [\mathbf{b}_1|\cdots|\mathbf{b}_i]$. Note that D is a positive integer. The reduction steps leave unchanged all these "partial" lattices, so D remains the same. It is easy to analyse the effect of Step 9 in that number:

$$\begin{aligned} D' = \; & \left(\textstyle\prod_{i \neq l-1} |B_i^t B_i|\right) \det\left([\mathbf{b}_1|\cdots|\mathbf{b}_{l-2}|\mathbf{b}_l]^t [\mathbf{b}_1|\cdots|\mathbf{b}_{l-2}|\mathbf{b}_l]\right) = \\ & D \frac{\mathrm{vol}\left(\mathcal{L}(\mathbf{b}_1, \ldots, \mathbf{b}_{l-2}, \mathbf{b}_l)\right)^2}{\mathrm{vol}\left(\mathcal{L}(\mathbf{b}_1, \ldots, \mathbf{b}_{l-1})\right)^2} = D \frac{\|\mathbf{b}_1^*\|^2 \cdots \|\mathbf{b}_{l-2}^*\|^2 \|\pi_{l-1}(\mathbf{b}_l)\|^2}{\|\mathbf{b}_1^*\|^2 \cdots \|\mathbf{b}_{l-1}^*\|^2} = \\ & D \frac{\|\pi_{l-1}(\mathbf{b}_l)\|^2}{\|\mathbf{b}_{l-1}^*\|^2} < \delta D. \end{aligned}$$

Vectors in a basis satisfying Definition 1.15 are an (exponential) approximation to the Minkowski minima.

Proposition 1.16 *Let the list* $\mathbf{b}_1, \ldots, \mathbf{b}_n$ *be LLL reduced with parameter* δ*, and let* $\mathbf{b}_1^*, \ldots, \mathbf{b}_n^*$ *the corresponding orthogonalized vectors. It is satisfied:*

- $\|\mathbf{b}_i\|_2 \leq (\delta - 1/4)^{(1-n)/2}\, \lambda_i(\Lambda)$.
- $\|\mathbf{b}_1\|_2 \leq (\delta - 1/4)^{(1-n)/4}\, (\mathrm{vol}\,\Lambda)^{1/n}$.
- $\prod_{i=1}^{n} \|\mathbf{b}_i\|_2 \leq (\delta - 1/4)^{n(1-n)/4}\, \mathrm{vol}\,\Lambda$.

This result shows that the LLL algorithm provides an approximation for the shortest vector problem in polynomial time. Despite the approximation factor is exponential in the lattice dimension (originally, $2^{(n-1)/2}$), it is enough to efficiently solve the factorization problem for rational polynomials. The LLL reduction has been the source for many lattice algorithms. R. Kannan [1983, 1987] provides exact solutions for both SVP and CVP, with a super-exponential algorithm. In [Babai, 1986], an approximated solution (within a factor $2^{n/2}$) for the closest vector problem is obtained by a polynomial time algorithm, the so-called "nearest-plane" method. In [Lovász and Scarf, 1992], a generalization of the LLL algorithm to an arbitrary norm is provided, obtaining as well a polynomial complexity. The Korkine-Zolotarev reduction is also studied under that generalization.

1.3 Lattice ideals

We collect in this section some basic results concerning the binomial ideal associated to an integer lattice, that was defined in [Eisenbud and Sturmfels, 1996; Sturmfels et al., 1995]. In the latter article, Gröbner bases for this binomial ideal are studied and a connection with Integer Programming is shown. We will use in Chapter III the results exposed in this section.

We denote by $\mathbb{N}$ the set of nonnegative integers. For integral vector $\mathbf{a} \in \mathbb{Z}^m$, we use the notation $\mathbf{a}^+, \mathbf{a}^-$ for the *positive* and *negative parts* of $\mathbf{a}$, i.e., the unique vectors with nonnegative components satisfying:

$$\mathbf{a} = \mathbf{a}^+ - \mathbf{a}^-.$$

Let $\mathbb{K}$ be an arbitrary field, and $\mathbb{K}[X_1, \ldots, X_m] = \mathbb{K}[\mathbf{x}]$ the polynomial ring in the variables $X_1, \ldots, X_m$. We write

$$(A) := \left\{ \sum_{i=1}^{l} f_i g_i \mid f_i \in \mathbb{K}[\mathbf{x}],\ g_i \in A \right\}$$

for the ideal generated by the set of polynomials $A \subseteq \mathbb{K}[\mathbf{x}]$. A *monomial* in $\mathbb{K}[\mathbf{x}]$ is an element of this ring with the form $M = X_1^{a_1} \cdots X_m^{a_m}$, with $a_i \in \mathbb{N}$. We use the notation $M = \mathbf{x}^{\mathbf{a}}$, where $\mathbf{a} = (a_1, \ldots, a_m)$ is the *exponent* of M. A *binomial* in $\mathbb{K}[\mathbf{x}]$ is a polynomial with at most two terms, i.e, an element of the form $\alpha_1 \mathbf{x}^{\mathbf{a}_1} + \alpha_2 \mathbf{x}^{\mathbf{a}_2}$, where $\alpha_1, \alpha_2 \in \mathbb{K}$ and $\mathbf{a}_1, \mathbf{a}_2 \in \mathbb{N}^m$. A *binomial ideal* is an ideal in $\mathbb{K}[\mathbf{x}]$ generated by a set of binomials.

Definition 1.17 *Let $\Lambda \subseteq \mathbb{Z}^m$ be an integer lattice. The lattice ideal associated with Λ is the ideal $I_\Lambda \subset \mathbb{K}[X_1, \ldots, X_m]$ generated by the following binomials:*

$$\{\mathbf{x}^{\mathbf{a}^+} - \mathbf{x}^{\mathbf{a}^-} \mid \mathbf{a} \in \Lambda\}.$$

With the same argument used in [Eisenbud and Sturmfels, 1996, Proposition 1.1], we can prove that the reduced Gröbner basis for a binomial ideal I with respect to any monomial ordering consists of binomials of the form $\mathbf{x}^{\mathbf{a}} - \mathbf{x}^{\mathbf{b}}$, where $\mathbf{a}, \mathbf{b} \in \mathbb{N}^m$. As a consequence, the coefficients of the two terms of any binomial in I must be opposite. This is,

$$f \text{ binomial}, \ f \in I \Rightarrow \exists \alpha \in \mathbb{K} \ \exists \mathbf{a}, \mathbf{b} \in \mathbb{N}^m \text{ s.t. } f = \alpha \mathbf{x}^{\mathbf{a}} - \alpha \mathbf{x}^{\mathbf{b}}.$$

The elements of a given integer lattice Λ determine the binomials in I_Λ:

Proposition 1.18 *Let Λ be an integer lattice and $\mathbf{a}, \mathbf{b} \in \mathbb{N}^m$. We have:*

$$\mathbf{x}^{\mathbf{a}} - \mathbf{x}^{\mathbf{b}} \in I_\Lambda \iff \mathbf{a} - \mathbf{b} \in \Lambda.$$

Therefore, we can obtain a generating set of vectors for Λ from any generating set of binomials for I_Λ, although the converse is not true. For instance, $\{(-1, 2), (-3, 1)\}$ is a basis for the lattice Λ represented in Figure 1.1, but

$$x^2y - 1 \notin (y^2 - x, y - x^3) \subsetneq I_\Lambda.$$

The binomial ideal associated with this lattice can be proved to be $I_\Lambda = (x^2y - 1, x - y^2)$ by the following result:

Lemma 1.19 *Let $A = [\mathbf{a}_1 | \cdots | \mathbf{a}_n] \in \mathbb{Z}^{m \times n}$. If the sum of the column vectors of A with no negative component has all its coordinates positive, then*

$$I_{\mathcal{L}(A)} = (\mathbf{a}^{\mathbf{a}_i^+} - \mathbf{x}^{\mathbf{a}_i^-} \mid 1 \leq i \leq n).$$

Chapter II

Prediction of pseudorandom sequences

In this chapter we present algorithms for predicting a pseduran-dom sequence with knowledge of partial information, following the linearizing technique developed by D. Coppersmith to find small roots of polynomials. We consider a quadratic congruential gener-ator over a finite prime field and a linear generator over an elliptic curve.

A source of random numbers with a certain probability distribution is useful in many situations, as testing an algorithm with some samples of input data, evaluating a definite integral by Monte Carlo integration, or even making an unbiased decision. Processes as tossing a coin or rolling a die might be called *purely random*, although a determinist may prefer to describe them as processes for which we cannot (at least for now) predict its behaviour. Anyway, it is not practical to resort to a method like that, especially if one needs to have access to a quick and long stream of random numbers, which is the case of cryptographic applications.

Pseudorandom sequences are sequences produced by a deterministic method, but which apparently look as obtained by a random process. As this sloppy definition suggests, deciding whether a sequence is pseudorandom, or how random it is, is not a simple question. In general, one can describe some properties that a psedurandom sequence is expected to fullfil and develop several tests in order to measure how well these properties are satisfied. For instance, if a pseudorandom sequence of elements in a set X is supposed to obtain each element at random with a certain probability distribution in X, one can use methods as the Kolmogorov-Smirnov or the χ^2 tests to discard failing sequences and increase the degree of confidence in passing ones. See [Niederreiter, 2001, 1995] for further references on pseudorandom number generators.

A simple application of pseudorandom sequences in Cryptology is the so-called *stream cipher*, inspired in the *one-time pad* cryptosystem (see Figure 2.1). The latter consists of two parties sharing a secret key at least as long as the message to be sent. The i-th bit of the clear message is just added to the i-th bit of the key (using the XOR operator) to obtain the i-th bit of the ciphertext, which is transmitted by a potentially insecure channel. This famous system is proved to have the *perfect secrecy*, as defined by C.E. Shannon [1949]. This means that in order to guess the clear message, an

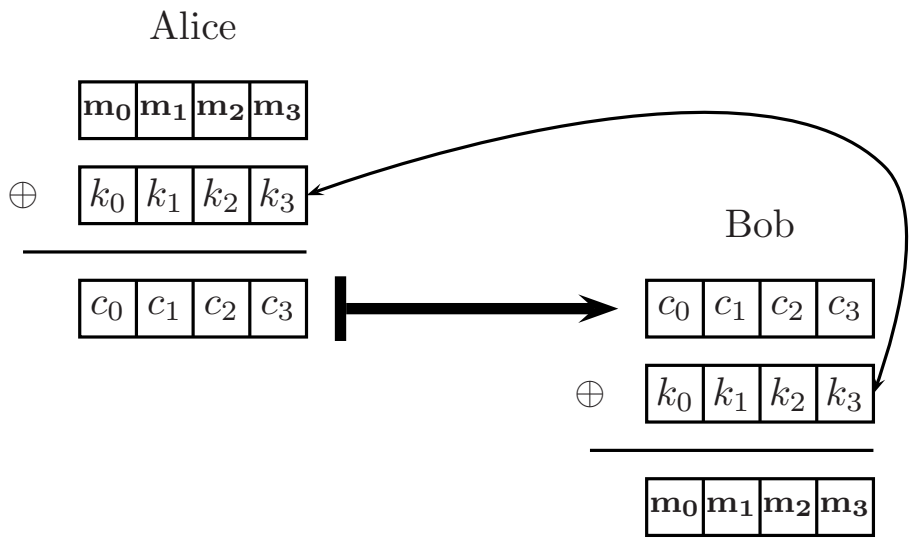


Figure 2.1: One-time pad cryptosystem

eavesdropper with access to the ciphertext has no advantage over another who has not intercepted the channel's traffic. The necessity of sharing a so long secret key reduces the practical applicability of this cryptosystem. A stream cipher overcomes this drawback by replacing the shared secret key with a pseudorandom sequence (called *keystream*) which can be reproduced independently by both speakers. Those must have previously agreed (by a secure channel) on some parameters (the secret key) that generate the sequence, as depicted in Figure 2.2. The perfect secrecy is lost, for the set of keys is (much) shorter than the set of possible messages.

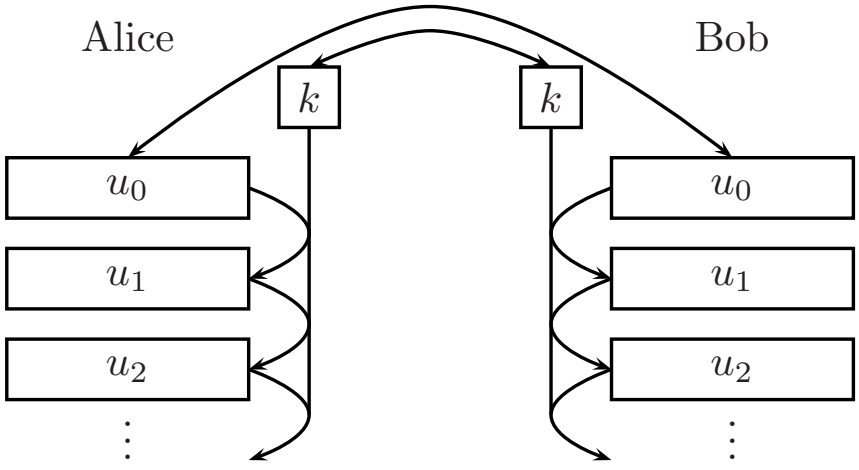


Figure 2.2: Keystream generation in a (synchronous) stream cipher

We consider an eavesdropper performing a "known plaintext" type attack. In other words, we assume that the cryptanalyst has access to a certain piece of the original

message and that the channel is insecure and therefore, he also has access to the whole ciphertext. It is quite realistic to consider this scenario, for it may be easy for the eavesdropper to guess some common parts of the plaintext, as a heading composed by an address, a date, customary greetings, etc. In a stream cipher, a known plaintext attack reveals a certain part of the keystream. Therefore, the used pseudorandom number generator should not be reproducible from a sample of its output. This unpredictability feature is required for pseudorandom number generators used in Cryptology, although it is unnecessary for applications in other contexts.

The *linear congruential generator (LCG)*, introduced in [Lehmer, 1951], is a simple method for generating numbers in a residue ring $\mathbb{Z}/m\mathbb{Z}$. Each element in the sequence is recursively obtained applying an affine transformation on the previous one:

$$u_{n+1} := au_n + b, \ \forall n \geq 0.$$

As any recursive sequence over a finite set, the output of LCG is periodic. A sequence with a short period does not look random and can be easily reproduced. Therefore, the period should be as long as possible. In [Knuth, 1981], the period of LCG is studied in detail.

As the sequence (u_n) can be reproduced from the parameters a, b and the *seed* u_0, these three numbers form the private key in the associated stream cipher cryptosystem. Note that if the modulus m is a prime number, revealing three consecutive elements in the sequence is enough to discover the parameters a, b and the whole pseudorandom stream. Thus, one should not use the elements just like they are output by this generator. A possible approach is using only a certain amount of the bits of each, in such a way that in a known plaintext attack, the eavesdropper has only access to approximations to the numbers output by the congruential generator. For instance, in the Figure 2.3 sketch, a certain amount of the less significant bits of each value is discarded, using the approximations $u_i - \varepsilon_i$ to compose the keystream. On the one

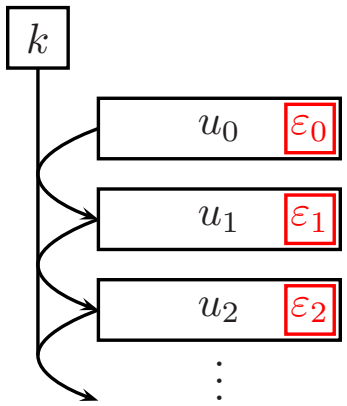


Figure 2.3: Keystream generation by approximating a pseudorandom sequence

hand, it is desirable to employ as much bits from each sequence element as possible, for efficiency reasons. On the other one, the more bits are revealed, the more vulnerable the cryptosystem might be, for a lattice basis reduction attack may discover the hidden bits and reproduce the keystream.

This kind of methods was introduced in [Knuth, 1985] and then considered in [Boyar, 1989a,b; Frieze et al., 1988; Joux and Stern, 1998; Krawczyk, 1992], see also the surveys [Brickell and Odlyzko, 1992; Lagarias, 1990]. One can apply similar algorithms to predict nonlinear pseudorandom number generators as well. This extension is inspired in the seminal paper [Coppersmith, 1997], which presents algorithms for computing the small roots (up to a certain bound) of a univariate modular polynomial and a bivariate integer polynomial.

In short, the former method builds an integer lattice Λ and computes an LLL-reduced basis $\mathbf{b}_1, \ldots, \mathbf{b}_n$ for it. A vector in Λ is associated with each sufficiently small root of the considered polynomial, and the Euclidean norm of that vector is small enough to guarantee that its expression as an integral combination of $\mathbf{b}_1, \ldots, \mathbf{b}_n$ does not involve the last element. This leads to a polynomial over the integers (not over a residue ring) that vanishes at the considered root. The integer roots of this last polynomial are easily computed, including all the desired solutions to the initial problem.

This linearizing technique has been applied in many situations, especially in Cryptology [Howgrave-Graham, 1997; May, 2003]. Indeed, the original paper presented already a cryptanalytic application: the factorization of an RSA modulus $N = pq$ from the high-order $(\log_2 N)/4$ bits of prime p.

We show in this chapter several algorithms for predicting pseudorandom sequences from partial information that are inspired in the Coppersmith method. These algorithms follow the approach initiated in [Blackburn et al., 2003, 2005], where the quadratic and inversive congruential generators are studied and improve some of the results therein. In [Blackburn et al., 2006], a similar method is developed for dealing with a congruential generator produced by an arbitrary polynomial. Nevertheless, one usually obtains better performances using specifically tailored methods for each particular case. These previous results are compiled in [Gómez, 2006]. Their common structure can be sketched as follows:

A pseudorandom number generator over a finite prime field $\mathbb{F}_p$ is proposed. Sometimes we treat elements of that set as integer numbers and use the notation $u \equiv_p v$ in stead of $u \equiv v \bmod p$. The cryptanalyst has access to approximations w_i to several consecutive values $u_0, \ldots, u_l$ such that the approximation errors $\varepsilon_i := u_i - w_i$ are not bigger than a certain *tolerance* Δ, i.e., each approximation error ε_i is the residue class modulo p of an integer of absolute value not bigger than Δ. Next procedure tries to obtain the original values u_i. Once these numbers are recovered, it is usually easy to reproduce the whole sequence.

- The generator's equation leads to a polynomial which vanishes on the approximation errors $(\varepsilon_0, \ldots, \varepsilon_l)$.
- That polynomial is transformed into a linear equation by regarding as a simple variable each monomial in $\mathbb{K}[\varepsilon_0, \ldots, \varepsilon_l]$, or a sum of several with the same coefficient.
- It is derived a homogeneous system of congruences satisfied by an integer vector $\mathbf{e}$ whose components contain enough information to reproduce the pseudorandom sequence. Moreover, there is a uniform bound for the absolute value of each component, of the order of a certain power of Δ.

- It is computed a nonzero solution $\mathbf{f}$ for that system with minimum Euclidean norm. Note that the set of solutions is an integer lattice. The exact SVP can be solved, for the dimension of the employed lattices is small in the cases we deal with.

- If the obtained vector $\mathbf{f}$ is parallel to the expected one $\mathbf{e}$, the approximation errors can be recovered.

- It is proved that in failure cases, the first value u_0 we try to recover must satisfy a polynomial equation from a certain bounded set. That limits the failure cases to instances whose initial value lies in a "bad" set $\mathcal{U}$ of cardinality $\#\mathcal{U} = O(\Delta^t)$.

Following this theoretical behaviour, the bigger the tolerance Δ for the approximation errors is, the more failure possibilities appear. If the tolerance $\Delta = p^\delta$ is expressed as a power of p, we can compare the size of the bad set $\mathcal{U}$ with the set of all possible values $\mathbb{F}_p$. It follows that when $p^\delta > p^{1/t}$, the bad set covers the whole finite field and the algorithm may fail always. On the other hand, if the fraction δ of unknown bits is smaller than $1/t$, the algorithm is expected to return the approximation errors with high probability, assuming that the first value u_0 is uniformly distributed in $\mathbb{F}_p$. Experimental tests are useful to confirm that threshold.

We illustrate this general procedure with a result for the inversive generator obtained in [Blackburn et al., 2005]. In the *inversive congruential generator*, two parameters $a, b \in \mathbb{F}_p^*$ are fixed and each element is obtained from the previous one following the recurrence relation:

$$u_{n+1} = \begin{cases} au_n^{-1} + b, & \text{if } u_n \neq 0 \\ b, & \text{if } u_n = 0. \end{cases}$$

In the assumed scenario, the cryptanalyst knows the multiplier and shift parameters (a and b) together with two approximations $w_0, w_1 \in \mathbb{Z}$ to two consecutive values u_0, u_1. Assuming $u_0 \neq 0$, we get

$$(w_0 + \varepsilon_0)(w_1 + \varepsilon_1 - b) \equiv_p a.$$

Then, integer vector $\tilde{\mathbf{e}} = (1, \varepsilon_0, \varepsilon_1, \varepsilon_0\varepsilon_1)$ is a root of the linear polynomial

$$p(X_0, X_1, X_2, X_3) = (w_0w_1 - w_0b - a)X_0 + (w_1 - b)X_1 + w_0X_2 + X_3$$

modulo p. In stead of searching for vector $\tilde{\mathbf{e}}$, we substitute $\mathbf{e} := (\Delta^2, \Delta\varepsilon_0, \Delta\varepsilon_1, \varepsilon_0\varepsilon_1)$, whose components' absolute values are balanced, for it. It solves the following homogeneous system of congruences:

$$\begin{cases} (w_0w_1 - w_0b - a)X_0 + \Delta(w_1 - b)X_1 + \Delta w_0X_2 + \Delta^2 \equiv 0 \bmod p \\ X_0 \equiv 0 \bmod \Delta^2 \\ X_1 \equiv 0 \bmod \Delta \\ X_2 \equiv 0 \bmod \Delta. \end{cases}$$

The smallest nonzero solution $\mathbf{f}$ of that system satisfies $\|\mathbf{f}\| \leq \|\mathbf{e}\| \leq 2\Delta^2$. Writing $\mathbf{f} = (\Delta^2 f_0, \Delta f_1, \Delta f_2, f_3)$, we have:

$$|f_0| \leq 2, \quad |f_1|, |f_2| \leq 2\Delta, \quad |f_3| \leq 2\Delta^2.$$

If vectors $\mathbf{e}$ and $\mathbf{f}$ are parallel, it is easy to recover the approximations errors $\varepsilon_i = f_{i+1}/f_0$, $i = 0, 1$. Let us consider the bad (for the cryptanalyst) case in which those vectors are not parallel. Then, vector $\mathbf{d} := f_0\mathbf{e} - \mathbf{f} = (0, \Delta d_1, \Delta d_2, d_3)$ is a nonzero solution of the system of congruences above. Therefore,

$$w_1 d_1 - b d_1 + w_0 d_2 + d_3 \equiv_p 0. \tag{2.1}$$

The given bounds for the components of $\mathbf{e}$ and $\mathbf{f}$ imply:

$$|d_1|, |d_2| \leq 4\Delta, \quad |d_3| \leq 4\Delta^2. \tag{2.2}$$

After the substitutions $w_i = u_i - \varepsilon_i$ in Equation (2.1), we get:

$$a u_0^{-1} d_1 + u_0 d_2 \equiv_p \varepsilon_1 d_1 + \varepsilon_0 d_2 - d_3 =: E.$$

We recall that $\mathbf{d}$ is a nonzero vector. Let us assume that the accuracy of the approximations reveals 3/4 of the bits of each element. In other words, $\Delta < p^{1/4}$. Then, bounds in (2.2) imply that the integers d_1 and d_2 cannot be both zero. The bad set $\mathcal{U}(\Delta; a)$ consists of 0 together with the set of elements $u \in \mathbb{F}_p$ satisfying an equation $a u^{-1} d_1 + u d_2 \equiv_p E$, for integers d_1, d_2, E such that $d_1^2 + d_2^2 \neq 0$, $|d_1|, |d_2| \leq 4\Delta$, and $|E| \leq 12\Delta^2$. It is simple to check that $\#\mathcal{U}(\Delta; a) = O(\Delta^4)$ and by the explained construction, this method returns the expected result when the input instance satisfies $u_0 \notin \mathcal{U}(\Delta; a)$. Note that, a fortiori, the assumption $\Delta < p^{1/4}$ does not limit the consequences of the algorithm, for the size of the "bad" set would exceed p when $\Delta \geq p^{1/4}$.

In this chapter we apply lattice basis reduction to the cryptanalysis of several pseudorandom sequences. As a minor detail, in all of them we solve a CVP to find a solution of an inhomogeneous system of congruences, in stead of a SVP to solve a homogeneous one, as is done in the previous example. The first section deals with a sequence of pseudorandom numbers recursively obtained by a quadratic polynomial without linear term: $ax^2 + c$. We analyse three cases: when both parameters are known, when the multiplier a is known and the shift c is unknown, and finally, the case $a = 1$, corresponding with the so-called Pollard generator. In this last scenario, we apply lattice basis reduction twice, obtaining a two-round algorithm. The second section applies lattice reduction to the linear generator of points in an elliptic curve over a finite prime field. These results have been published in [Gómez et al., 2005b, 2006; Gutierrez and Ibeas, 2007]. Throughout this chapter, "polynomial time" means polynomial in $\log p$.

2.1 Quadratic Generator

We consider in this section a *quadratic generator* of elements of $\mathbb{F}_p$, given by the recurrence relation

$$u_{n+1} = a u_n^2 + c, \forall n \geq 0. \tag{2.3}$$

The initial value u_0 is called *seed*. We refer to the parameters a and c as the *multiplier* and *shift*, respectively.

In the cryptographic setting, the initial value u_0 and the constants a, c are assumed to be the secret key, and the stream cipher is obtained from the output of the generator. If three consecutive values (u_n, u_{n+1}, u_{n+2}) such that $u_n \neq \pm u_{n+1}$ are revealed, it is easy to find a and c and reproduce the sequence. So in this setting, we only use the most significant bits of each u_n in the hope that this makes the resulting keystream difficult to predict.

In this section, we apply lattice basis reduction in three different scenarios. In the first one, we assume that the cryptanalyst has access to both the multiplier and the shift. In the second one, we only assume that the multiplier is known. Finally, the third one studies the particular case $a = 1$, known as *Pollard generator*.

Let Δ be a positive integer, and $w, u \in \mathbb{F}_p$. We say that w is a Δ-*approximation* to u if $u - w \in \{-\Delta, 1 - \Delta, \ldots, \Delta\} \subseteq \mathbb{F}_p$. So, the case where Δ grows like a fixed power p^δ, where $0 < \delta < 1$, corresponds to the situation where a positive fraction δ of the least significant bits of terms of the output sequence remains hidden.

2.1.1 Known multiplier and shift

Theorem 2.1 *Let p be a prime number and Δ a positive integer. For any $a \in \mathbb{F}_p^*$ and $c \in \mathbb{F}_p$, there exists a set $\mathcal{U}(\Delta; a) \subseteq \mathbb{F}_p$ of cardinality $\#\mathcal{U}(\Delta; a) = O(\Delta^4)$ with the following property. There exists an algorithm which, when given a, c, and Δ-approximations w_0, w_1 to two consecutive values u_0, u_1 produced by the quadratic generator (2.3) such that $u_0 \notin \mathcal{U}(\Delta; a)$, returns the value u_0 in polynomial time.*

Proof. The result is trivial when $\Delta^4 < p$, so we can assume the opposite. Using the notation $\varepsilon_i := u_i - w_j$ for the approximation errors, we get

$$w_1 + \varepsilon_1 - a(w_0 + \varepsilon_0)^2 - c \equiv_p 0.$$

Then, vector

$$\mathbf{e} := (\Delta\varepsilon_0, \Delta\varepsilon_1, \varepsilon_0^2),$$

which contains the unknown information, is a solution of the system of congruences below and its norm is bounded by $\sqrt{3}\Delta^2$.

$$\begin{cases} 2aw_0\Delta X_1 - \Delta X_2 + a\Delta^2 X_2 & \equiv & \Delta^2(w_1 - aw_0^2 - c) \bmod p \\ X_1 & \equiv & 0 \bmod \Delta \\ X_2 & \equiv & 0 \bmod \Delta. \end{cases} \tag{2.4}$$

Solving the CVP for this system leads to a vector $\mathbf{f} = (\Delta f_1, \Delta f_2, f_3)$ with $\|\mathbf{f}\| \leq \sqrt{3}\Delta^2$. Therefore,

$$|f_1|, |f_2| \leq \sqrt{3}\Delta, \quad f_3 \leq \sqrt{3}\Delta^2. \tag{2.5}$$

We might hope that this output vector contains the approximations errors. In order to analyse the failure case, we build vector $\mathbf{d} := \mathbf{e} - \mathbf{f} = (\Delta d_1, \Delta d_2, d_3)$, which is a solution of the homogeneous system of congruences associated with (2.4), and therefore,

$$2aw_0 d_1 - d_2 + ad_3 \equiv_p 0. \tag{2.6}$$

Using the bounds from (2.5), we get:

$$|d_1|, |d_2| \leq 3\Delta, \quad |d_3| \leq 3\Delta^2.$$

Undoing the change $u_0 = w_0 + \varepsilon_0$ on Equation (2.6) above,

$$2ad_1u_0 \equiv_p E, \quad \text{where } E = a(2d_1\varepsilon_0 - d_3) + d_2. \tag{2.7}$$

We define $\mathcal{U}(\Delta; a)$ as the set of values u_0 that satisfy some congruence of the form (2.7) for integers $d_1, d_2, d_3, \varepsilon_0$ such that $d_1 \not\equiv_p 0$, $|d_1|, |d_2| \leq \sqrt{3}\Delta$, $|d_3| \leq \sqrt{3}\Delta$, and $|\varepsilon_0| \leq \Delta$. These bounds restrict the possible values of d_1 to $O(\Delta)$. Moreover, E can take $O(\Delta^3)$ distinct values, for $2d_1\varepsilon_0 - d_3 = O(\Delta^2)$ and $d_2 = O(\Delta)$. That gives the bound $\#\mathcal{U}(\Delta; a) = O(\Delta^4)$.

To finish the proof, we just note that when $u_0 \notin \mathcal{U}(\Delta; a)$, by computing vector $\mathbf{f}$ we can recover the first approximation error ε_0, and reproduce the pseudorandom sequence. ■

2.1.2 Known multiplier and unknown shift

Theorem 2.2 *Let p be a prime number and Δ a positive integer. For any $a \in \mathbb{F}_p^*$ and $c \in \mathbb{F}_p$, there exists a set $\mathcal{U}(\Delta; a, c) \subseteq \mathbb{F}_p$ of cardinality $\#\mathcal{U}(\Delta; a, c) = O(\Delta^5)$ with the following property. There exists an algorithm which, when given a and Δ-approximations w_0, w_1, w_2 to three consecutive values u_0, u_1, u_2 produced by the quadratic generator (2.3) such that $u_0 \notin \mathcal{U}(\Delta; a, c)$, recovers u_0 and c in polynomial time.*

Proof. Let us assume $\Delta^5 < p$. Using the notation $\varepsilon_i = u_i - w_i$, for $i = 0, 1, 2$, we obtain:

$$a(w_0 + \varepsilon_0)^2 - (w_1 + \varepsilon_1) \equiv_p a(w_1 + \varepsilon_1)^2 - (w_2 + \varepsilon_2),$$

and therefore,

$$aw_0^2 - w_1 - aw_1^2 + w_2 + 2aw_0 \underbrace{\varepsilon_0} - (1 + 2aw_1) \underbrace{\varepsilon_1} + \underbrace{\varepsilon_2} + a \underbrace{(\varepsilon_0^2 - \varepsilon_1^2)} \equiv_p 0.$$

We consider the system of congruences:

$$\begin{cases} 2aw_0\Delta X_1 - (1 + 2aw_1)\Delta X_2 + \Delta X_3 + a\Delta_4^X & \equiv & w_1 - w_2 + a(w_1^2 - w_0^2) \bmod p \\ X_1 & \equiv & 0 \bmod \Delta \\ X_2 & \equiv & 0 \bmod \Delta \\ X_3 & \equiv & 0 \bmod \Delta, \end{cases} \tag{2.8}$$

that is solved by vector $\mathbf{e} = (\Delta\varepsilon_0, \Delta\varepsilon_1, \Delta\varepsilon_2, \varepsilon_0^2 - \varepsilon_1^2)$, whose norm is bounded by $\sqrt{7}\Delta^2$. We compute a solution of (2.8):

$$\mathbf{f} = (\Delta f_1, \Delta f_2, \Delta f_3, f_4), \quad \text{with } \|\mathbf{f}\| \leq \sqrt{7}\Delta^2. \tag{2.9}$$

That leads to the following bounds:

$$|f_1|, |f_2|, |f_3| \leq 2\Delta, \quad |f_4| \leq 2\Delta^2. \tag{2.10}$$

We denote by $\mathbf{d} := \mathbf{e} - \mathbf{f} = (\Delta d_1, \Delta d_2, \Delta d_3, d_4)$ the difference vector of the desired and obtained ones. It satisfies:

$$2aw_0d_1 - (1 + 2aw_1)d_2 + d_3 + ad_4 \equiv_p 0, \tag{2.11}$$

$$|d_1|, |d_2|, |d_3| \leq 3\Delta, \quad |d_4| \leq 3\Delta^2. \tag{2.12}$$

Equation (2.11) above implies $P(u_0) \equiv_p 0$, where:

$$P(T) := -2a^2 d_2 T^2 + 2ad_1 T - 2ad_1\varepsilon_0 - d_2 - 2ad_2 c + 2ad_2\varepsilon_1 + d_3 + ad_4. \tag{2.13}$$

We define $\mathcal{U}(\Delta; a, c)$ as the set of elements $u_0 \in \mathbb{F}_p$ such that there exist integers $d_1, d_2, d_3, d_4, \varepsilon_0, \varepsilon_1$ satisfying:

$$(d_1 \not\equiv_p 0 \ \vee \ d_2 \not\equiv_p 0), \ P(u_0) \equiv_p 0.$$

Bounds in (2.12) imply $\#\mathcal{U}(\Delta; a, c) = O(\Delta^5)$. Moreover, when $u_0 \notin \mathcal{U}(\Delta; a, c)$, we can recover the approximation errors, for $d_1 = d_2 = 0$. ■

2.1.3 Pollard generator

We consider now the case of a generator of numbers in $\mathbb{F}_p$ produced by a quadratic and monic polynomial without linear term:

$$u_n = u_{n-1}^2 + c. \tag{2.14}$$

We assume that the sequence (u_n) is hidden but approximations w_j to two consecutive values u_j, $j = 0, 1$, are given. Then, Blackburn et al. [2005] show that the values u_j can be recovered from this information in polynomial time if the approximations w_j are sufficiently good and if a certain small set of initial values u_0 is excluded.

Theorem 2.3 ([Blackburn et al., 2005], Thm. 4) *Let p be a prime number and Δ an integer such that $p > \Delta \geq 1$. For any $c \in \mathbb{F}_p$, there exists a set $\mathcal{U}(\Delta, c) \subseteq \mathbb{F}_p$ of cardinality $\#\mathcal{U}(\Delta, c) = O(\Delta^3)$ with the following property. There exists an algorithm which, when given Δ-approximations w_j, $j = 0, 1$, to two consecutive values u_0, u_1 produced by the Pollard generator (2.14), returns the value of u_0 in deterministic polynomial time if $u_0 \notin \mathcal{U}(\Delta, c)$.*

In other words, the Pollard generator is likely polynomial time predictable if 2/3 of the bits of two consecutive elements are revealed. We improve the above result showing that one only needs to know 9/14 of the most significant bits in order to (almost always) predict the Pollard generator:

Theorem 2.4 *Let p be a prime number and Δ a positive integer. For any $c \in \mathbb{F}_p$, there exists a set $\mathcal{V}(\Delta, c) \subseteq \mathbb{F}_p \times [-\Delta, \Delta]$ of cardinality $\#\mathcal{V}(\Delta, c) = O\left(\max\{\Delta^{15}p^{-4}, \Delta^{19/5}\}\right)$ with the following property. There exists an algorithm which, when given Δ-approximations w_j, $j = 0, 1$, to two consecutive values u_0, u_1 produced by the Pollard generator (2.14), returns the value of u_0 in deterministic polynomial time if $(u_0, u_0 - w_0) \notin \mathcal{V}(\Delta, c)$.*

In order to prove this result, we introduce some modifications and additions to the method of [Blackburn et al., 2005]. We demonstrate our technique in the special case when c is public. Of course, this assumption reduces the relevance of the problem in

Cryptology. This approach can be extended heuristically to the case when c is secret, as is pointed out at the end of this section.

Proof. In a similar way to the proof of the previous results, we build a "bad" set for which we cannot guarantee the succes of the algorithm. However, this time the set does not consist of values for the first pseudorandom number u_0. In this case, the exceptional set consists of pairs including both the first value u_0 and the first approximation error ε_0. We will prove that when the input instance does not match a pair in the set, the algorithm works correctly. Moreover, as the size of that set is in $O\left(\max\{\Delta^{15}p^{-4}, \Delta^{19/5}\}\right)$, if this quantity remains lower than $p\Delta$ (the total number of pair choices), we can probably obtain successful guessing. So, the threshold for the maximum error provided by this proof is $\Delta < p^{5/14}$. Let us start to describe the algorithm.

Let $\varepsilon_j = u_j - w_j$ be the approximation errors, for $j = 0, 1$. From

$$u_1 \equiv u_0^2 + c \bmod p,$$

we obtain

$$2w_0\varepsilon_0 + \varepsilon_0^2 - \varepsilon_1 + w_0 + c - w_1 \equiv_p 0, \tag{2.15}$$

which can be looked on as a linear equation over a vector containing enough information to discover the goal u_0. In fact, vector

$$\mathbf{e} = \left(\Delta\varepsilon_0, \varepsilon_0^2 - \varepsilon_1\right)$$

is a solution to the following linear system of congruences:

$$\begin{cases} 2w_0X_1 + \Delta X_2 & \equiv \Delta(w_1 - c - w_0^2) \bmod p \\ X_1 & \equiv 0 \bmod \Delta. \end{cases} \tag{2.16}$$

Moreover, $\mathbf{e}$ is a relatively short vector. Indeed, $\|\mathbf{e}\| \leq \sqrt{5}\Delta^2$.

Let Λ be the lattice consisting of integer solutions $\mathbf{x} = (x_1, x_2) \in \mathbb{Z}^2$ of the system of congruences:

$$\begin{cases} 2w_0X_1 + \Delta X_2 & \equiv 0 \bmod p \\ X_1 & \equiv 0 \bmod \Delta. \end{cases} \tag{2.17}$$

It is easily checked that Λ is a two-dimensional lattice with volume $p\Delta$. Vector $\mathbf{t} = (\Delta, w_1 - c - w_0^2 - 2w_0)$ is a particular solution of the linear system (2.16). Now, we can apply an algorithm solving the CVP for the shift vector $\mathbf{t}$ and the lattice Λ to obtain a vector $\mathbf{f} = (\Delta f_1, f_2)$ satisfying Equations (2.16) and

$$\|\mathbf{f}\| \leq \sqrt{5}\Delta^2.$$

Note that we can compute $\mathbf{f}$ in polynomial time from the information we are given. We might hope that $\mathbf{e}$ and $\mathbf{f}$ coincide. In other case, it can be shown (as in [Blackburn et al., 2005]) that u_0 belongs to a subset $\mathcal{U} \subseteq \mathbb{F}_p$ of cardinality $\#\mathcal{U} = O(\Delta^3)$. Our approach here is more involved. We compute in polynomial time (see Section 1.1) a reduced basis

$$\{\mathbf{g} = (\Delta g_1, g_2), \mathbf{h} = (\Delta h_1, h_2)\}$$

for lattice Λ, with $\|\mathbf{g}\| \leq \|\mathbf{h}\|$. Since $\mathbf{e}-\mathbf{f} \in \Lambda$, there exist integers γ_1 and γ_2 satisfying:

$$\mathbf{e} - \mathbf{f} = \gamma_1 \mathbf{g} + \gamma_2 \mathbf{h}. \tag{2.18}$$

By the bounds on $\|\mathbf{e}\|$ and $\|\mathbf{f}\|$ we have $\|\mathbf{e} - \mathbf{f}\| \leq 2\sqrt{5}\Delta^2$, and Lemma 1.12 provides the following:

$$\begin{array}{rl} |\gamma_1| \leq & \max\left\{1, \left\lfloor \dfrac{4\sqrt{5}\Delta^2}{\sqrt{3}\|\mathbf{g}\|} \right\rfloor\right\} =: \Delta_1, \\ |\gamma_2| \leq & \max\left\{1, \left\lfloor \dfrac{4\sqrt{5}\Delta^2}{\sqrt{3}\|\mathbf{h}\|} \right\rfloor\right\} =: \Delta_2. \end{array} \tag{2.19}$$

So, the missing information is now γ_1 and γ_2. From (2.18) we derive two new equations:

$$\begin{array}{rl} \varepsilon_0 = & f_1 + \gamma_1 g_1 + \gamma_2 h_1, \\ \varepsilon_0^2 - \varepsilon_1 = & f_2 + \gamma_1 g_2 + \gamma_2 h_2. \end{array} \tag{2.20}$$

Eliminating ε_0, we obtain:

$$(f_1 + \gamma_1 g_1 + \gamma_2 h_1)^2 - \varepsilon_1 = f_2 + \gamma_1 g_2 + \gamma_2 h_2.$$

Operating, we reach an equation involving six variables related to γ_1, γ_2 and ε_1:

$$f_1^2 + 2f_1\gamma_1 g_1 + 2f_1\gamma_2 h_1 + g_1^2\gamma_1^2 + 2g_1 h_1 \gamma_1\gamma_2 + h_1^2\gamma_2^2 = f_2 + \gamma_1 g_2 + \gamma_2 h_2 + \varepsilon_1. \tag{2.21}$$

Linearizing this new equation, we obtain that the following system of congruences

$$\left\{ \begin{array}{rcl} (2f_1g_1 - g_2)\Delta_1 X_1 + (2f_1h_1 - h_2)\Delta_2 X_2 + g_1^2\Delta_1^2 X_3 + & & \\ 2g_1h_1\Delta_1\Delta_2 X_4 + h_1^2\Delta_2^2 X_5 - \Delta X_6 & = & (f_2 - f_1^2)\Delta_1^2\Delta_2^2\Delta \\ X_1 & \equiv & 0 \bmod \Delta_1\Delta_2^2\Delta \\ X_2 & \equiv & 0 \bmod \Delta_1^2\Delta_2\Delta \\ X_3 & \equiv & 0 \bmod \Delta_2^2\Delta \\ X_4 & \equiv & 0 \bmod \Delta_1\Delta_2\Delta \\ X_5 & \equiv & 0 \bmod \Delta_1^2\Delta \\ X_6 & \equiv & 0 \bmod \Delta_1^2\Delta_2^2 \end{array} \right. \tag{2.22}$$

is solved by vector

$$\mathbf{e}' = (\Delta_1\Delta_2^2\Delta\gamma_1, \Delta_1^2\Delta_2\Delta\gamma_2, \Delta_2^2\Delta\gamma_1^2, \Delta_1\Delta_2\Delta\gamma_1\gamma_2, \Delta_1^2\Delta\gamma_2^2, \Delta_1^2\Delta_2^2\varepsilon_1).$$

By the bounds in (2.19), the Euclidean norm of $\mathbf{e}'$ satisfies the inequality:

$$\|\mathbf{e}'\| \leq \sqrt{6}\Delta_1^2\Delta_2^2\Delta. \tag{2.23}$$

Let Λ' be the lattice consisting of integer solutions $\mathbf{x} = (x_1, \ldots, x_6) \in \mathbb{Z}^6$ of the homogeneus system obtained from (2.22):

$$\left\{ \begin{array}{rcl} (2f_1g_1 - g_2)\Delta_1 X_1 + (2f_1h_1 - h_2)\Delta_2 X_2 + & & \\ g_1^2\Delta_1^2 X_3 + 2g_1h_1\Delta_1\Delta_2 X_4 + h_1^2\Delta_2^2 X_5 - \Delta X_6 & = & 0 \\ X_1 & \equiv & 0 \bmod \Delta_1\Delta_2^2\Delta \\ X_2 & \equiv & 0 \bmod \Delta_1^2\Delta_2\Delta \\ X_3 & \equiv & 0 \bmod \Delta_2^2\Delta \\ X_4 & \equiv & 0 \bmod \Delta_1\Delta_2\Delta \\ X_5 & \equiv & 0 \bmod \Delta_1^2\Delta \\ X_6 & \equiv & 0 \bmod \Delta_1^2\Delta_2^2. \end{array} \right. \tag{2.24}$$

Applying an algorithm solving the CVP for the shift vector $(0,0,0,0,0,(f_1^2-f_2)\Delta_1^2\Delta_2^2)$ and the lattice Λ', we obtain a vector

$$\mathbf{f}' = (\Delta_1\Delta_2^2\Delta f_1', \Delta_1^2\Delta_2\Delta f_2', \Delta_2^2\Delta f_3', \Delta_1\Delta_2\Delta f_4', \Delta_1^2\Delta f_5', \Delta_1^2\Delta_2^2 f_6'),$$

satisfying equations (2.22) and:

$$\|\mathbf{f}'\| \leq \sqrt{6}\Delta_1^2\Delta_2^2\Delta. \tag{2.25}$$

Again, we note that we may compute $\mathbf{f}'$ in polynomial time from the information we are given. We might hope that

$$\varepsilon_0 = f_1 + f_1' g_1 + f_2' h_1.$$

Let us bound the "bad" possibilities for which this identity may be wrong. For that purpose, we define vector $\mathbf{d} := \mathbf{f}' - \mathbf{e}'$, which lies in Λ'.

$$\mathbf{d} = (\Delta_1\Delta_2^2\Delta d_1, \Delta_1^2\Delta_2\Delta d_2, \Delta_2^2\Delta d_3, \Delta_1\Delta_2\Delta d_4, \Delta_1^2\Delta d_5, \Delta_1^2\Delta_2^2 d_6).$$

Bounds (2.23) and (2.25) imply $\|\mathbf{d}\| \leq 2\sqrt{6}\Delta_1^2\Delta_2^2\Delta$ and

$$\begin{array}{ll} |d_1| & \leq 2\sqrt{6}\Delta_1, \\ |d_2| & \leq 2\sqrt{6}\Delta_2, \\ |d_3| & \leq 2\sqrt{6}\Delta_1^2, \\ |d_4| & \leq 2\sqrt{6}\Delta_1\Delta_2, \\ |d_5| & \leq 2\sqrt{6}\Delta_2^2, \\ |d_6| & \leq 2\sqrt{6}\Delta. \end{array} \tag{2.26}$$

Then, as an integer linear combination of $\mathbf{g}$ and $\mathbf{h}$, vector $\mathbf{q} := d_1\mathbf{g} + d_2\mathbf{h}$ lies in Λ. Now, writing $\mathbf{q} = (\Delta q_1, q_2)$ and using Equations (2.24),

$$\begin{array}{ll} q_1 & = d_1 g_1 + d_2 h_1, \\ q_2 & = 2f_1 g_1 d_1 + 2f_1 h_1 d_2 + g_1^2 d_3 + 2g_1 h_1 d_4 + h_1^2 d_5 - d_6. \end{array}$$

So, we have:

$$2q_1 w_0 + q_2 \equiv_p 0. \tag{2.27}$$

After substituting $w_0 = u_0 - \varepsilon_0 = u_0 - (f_1 + \gamma_1 g_1 + \gamma_2 h_1)$ in Equation (2.27) above, we find

$$2q_1 u_0 \equiv_p E, \tag{2.28}$$

where $E = 2q_1(f_1 + \gamma_1 g_1 + \gamma_2 h_1) - q_2$. And substituting the expressions for q_1 and q_2 and operating, we obtain:

$$E = g_1^2(2\gamma_1 d_1 - d_3) + 2g_1 h_1(-d_4 + \gamma_1 d_2 + \gamma_2 d_1) + h_1^2(-d_5 + 2\gamma_2 d_2) + d_6. \tag{2.29}$$

Since we are assuming that $\varepsilon_0 \neq f_1 + f_1' g_1 + f_2' h_1$, then $q_1 = d_1 g_1 + d_2 h_1 \neq 0$. So, for each value q_1 and E there exists one (and only one) value u_0 satisfying congruence (2.28).

Then, we define $\mathcal{V}$ as the set of pairs (u_0, ε_0) for which, setting $\{\mathbf{g} = (\Delta g_1, g_2), \mathbf{h} = (\Delta h_1, h_2)\}$ as the output of the algorithm used to obtain reduced basis when given the

lattice (2.17) with $w_0 := u_0 - \varepsilon_0$ as input, there exist integers d_1,d_2,d_3,d_4,d_5,d_6,γ_1,γ_2,f_1 such that

$$\begin{array}{ll} |d_1| \leq 2\sqrt{6}\Delta_1, & |d_2| \leq 2\sqrt{6}\Delta_2, \\ |d_3| \leq 2\sqrt{6}\Delta_1^2, & |d_4| \leq 2\sqrt{6}\Delta_1\Delta_2, \\ |d_5| \leq 2\sqrt{6}\Delta_2^2, & |d_6| \leq 2\sqrt{6}\Delta, \\ |\gamma_1| \leq \Delta_1, & |\gamma_2| \leq \Delta_2, \\ |f_1| \leq \sqrt{5}\Delta & \end{array}$$

satisfying the following two identities:

$$\begin{aligned} 2(d_1g_1 + d_2h_1)u_0 \equiv_p g_1^2(2\gamma_1 d_1 - d_3) + 2g_1h_1(-d_4 + \gamma_1 d_2 + \gamma_2 d_1) + \\ h_1^2(-d_5 + 2\gamma_2 d_2) + d_6, \end{aligned} \tag{2.30}$$

$$(2f_1g_1 - g_2)d_1 + (2f_1h_1 - h_2)d_2 + g_1^2 d_3 + 2g_1h_1d_4 + h_1^2 d_5 - d_6 = 0. \tag{2.31}$$

It is clear that for any pair outside $\mathcal{V}$, the algorithm proposed must work properly, because we have seen how, if the algorithm fails, we reach a situation as defined for pairs in $\mathcal{V}$. But, in order to properly bound the size of this set, we introduce another set $\mathcal{T}$ consisting of pairs $(u_0, \varepsilon_0) \in \mathbb{F}_p \times [-\Delta, \ldots, \Delta]$ for which there are integers q_1, E such that

$$|q_1| \leq 84\Delta^{4/5}, \qquad |E| \leq 20\sqrt{6}\Delta^2$$

satisfying:

$$2u_0q_1 \equiv_p E,\ q_1 \neq 0.$$

As the pair's second component ε_0 is meaningless in this definition, we can state $\#\mathcal{T} = O(\Delta^{19/5})$. Let us now measure the difference set $\mathcal{V}\backslash\mathcal{T}$: if (u_0, ε_0) and $(\tilde{u}_0, \tilde{\varepsilon}_0)$ are two elements in this set, by the definition of $\mathcal{V}$ there are nine integers for each one satisfying Equations (2.30) and (2.31). Let us suppose the first seven integers $(d_1, d_2, \gamma_1, \gamma_2, d_3, d_4, d_5)$ are the same for both, as well as the differences $u_0 - \varepsilon_0 = \tilde{u}_0 - \tilde{\varepsilon}_0$. Then, by (2.31),

$$2(f_1 - \tilde{f}_1)(d_1g_1 + d_2h_1) = d_6 - \tilde{d}_6.$$

By the bounds provided by $\mathcal{V}$ definition and by being these two pairs outside $\mathcal{T}$, it must be $d_1g_1 + d_2h_2 = \Omega\left(\Delta^{4/5}\right)$; and then, $|f_1 - f_1'|$ is upper bounded by $O(\Delta^{1/6})$. Now, using equations (2.17), we reach:

$$2u_0(f_1 - \tilde{f}_1) \equiv_p \varepsilon_0(f_1 - \tilde{f}_1) + (f_2 - \tilde{f}_2).$$

However, using once again the fact that the first pair selected is outside $\mathcal{T}$, it must be that $f_1 - \tilde{f}_1 = 0$. As a consequence, fixing $u_0 - \varepsilon_0$, $d_1, d_2, \gamma_1, \gamma_2, d_3, d_4$ and d_5 is enough to determine one unique element in $\mathcal{V}\backslash\mathcal{T}$. With this, after the difference $u_0 - \varepsilon_0$ is fixed, the number of choices is bounded by $O\left((\Delta_1\Delta_2)^5\right)$. Using Lemma 1.13, this is the same as $O\left((\Delta^3 p^{-1})^5\right)$. Then, $\#\left(\mathcal{V}\backslash\mathcal{T}\right) = O(\Delta^{15}p^{-4})$. Finally, $\#\mathcal{V} \leq \#\left(\mathcal{V}\backslash\mathcal{T}\right) + \#\mathcal{T}$. ■

We deal now with the case where the parameter (known as shift) is supposed to be known. If this information is not previously given, paper [Blackburn et al., 2005] requires three (instead of two) approximations to consecutive sequence elements to perform a lattice attack. This is not a restrictive feature, for it is usually easy to have access to a signifacant amount of approximations. However, the algorithm in

[Blackburn et al., 2005] requires better-quality approximations: one may recover the seed when $\Delta < p^{1/4}$.

One can develop a similar algorithm to the one presented in the proof of Theorem 2.4, but it is not immediately clear how to bound the failure possibilities as we do there. Some tests have been performed (see Section 4.1) in order to obtain empirically the threshold for the allowed tolerance.

The admitted fraction of unknown bits, as observed in the tests, grows from $\delta = 0.25$ to $\delta \simeq 0.261$. Here is an example of the test results, comparing the proportion of unknown bits with the success percentage:

δ	0.26	0.2613	0.2616	0.2618	0.262
	100%	100%	48%	16%	0

So, the improving factor with this two-round technique is only about 4%. Moreover, in [Blackburn et al., 2005] a heuristic method that reached $\delta = 1/3$ as maximum tolerance was presented.

2.2 Linear Generator over Elliptic Curves

The Theory of Numbers, in which G.H. Hardy saw the beauty of uselessness, is the mathematical source employed by many cryptographic protocols and cryptanalytic procedures. Elliptic curves have provided several applications to Cryptology, starting with an improvement of the existing methods for factoring integers in [Lenstra, 1987]. There have been developed public key encryption schemes based on elliptic curves. An interesting feature of these schemes is the low size of the keys needed to provide the same security than other common protocols (as RSA). This makes elliptic curve cryptography a convenient choice for processors with reduced computing capability, as those implemented in mobile devices. In this section we apply the lattice basis reduction technique to attack the linear generator over elliptic curves.

2.2.1 The group of an elliptic curve

Let $\mathbb{K}$ be a field with characteristic distinct to 2 and $f \in \mathbb{K}[X]$ a cubic polynomial with three distinct roots in a certain extension $\mathbb{K}'$ of $\mathbb{K}$. The set of points $(x, y) \in \mathbb{K}^2$ such that

$$y^2 = f(x) = f_0 + f_1 x + f_2 x^2 + f_3 x^3$$

is an *affine elliptic curve*. It is useful to consider the corresponding projective curve, i.e., the set of points $(x : y : z) \in \mathbb{P}_2(\mathbb{K})$ such that

$$y^2 z = f_0 z^3 + f_1 x z^2 + f_2 x^2 z + f_3 x^3.$$

It is easy to see that this curve has only one point at infinity, which is denoted by $\mathcal{O} := (0 : 1 : 0)$. Moreover, the line at infinity $z = 0$ is the tangent line of the curve at $\mathcal{O}$, and the order of tangency is 3.

Any line intersecting with an elliptic curve has three points P_1, P_2, P_3 (counting their multiplicity) in common with it. This permits the definition of a commutative

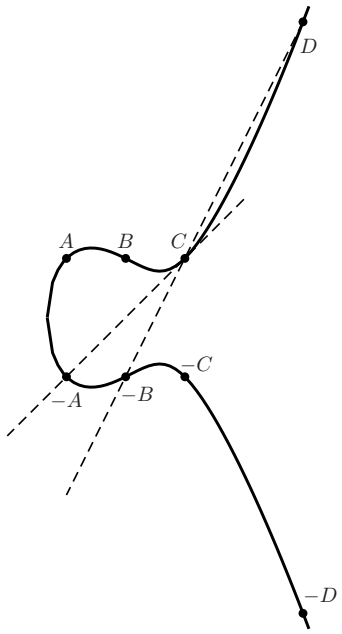


Figure 2.4: Curve $Y^2 = X^3 - X + 1$ over the reals.

group law, for which $\mathcal{O}$ is the neutral element, satisfying:

$$P_1 \oplus P_2 \oplus P_3 = \mathcal{O}.$$

Figure 2.4 depicts an affine elliptic curve E over the field of real numbers. Note that $(x, y) \in E \iff (x, -y) \in E$. These (symmetric about the abscissae axis) points are opposite, as the projective line $X = xZ$ containing them also intersects the curve at $\mathcal{O}$.

Let E be an elliptic curve defined over $\mathbb{F}_p$ by an *affine Weierstrass equation*, which for $\gcd(p, 6) = 1$ takes the form

$$Y^2 = X^3 + aX + b, \tag{2.32}$$

for some $a, b \in \mathbb{F}_p$ with $4a^3 + 27b^2 \neq 0$. Equations for the group law $\oplus$ acting over affine points $P = (x_P, y_P), Q = (x_Q, y_Q) \in E$ are easily derived, resulting:

$$P \oplus Q = R = (x_R, y_R), \text{ where :}$$

- If $x_P \neq x_Q$, then

$$\begin{cases} x_R = m^2 - x_P - x_Q \\ y_R = m(x_P - x_R) - y_P, \quad \text{where } m = \dfrac{y_Q - y_P}{x_Q - x_P}. \end{cases} \tag{2.33}$$

- If $x_P = x_Q$ but $y_P \neq y_Q$, then $P \oplus Q = \mathcal{O}$.

- If $P = Q$ and $y_P \neq 0$, then

$$\begin{cases} x_R = m^2 - 2x_P \\ y_R = m(x_P - x_R) - y_P, \quad \text{where } m = \dfrac{3x_P^2 + a}{2y_P}. \end{cases} \tag{2.34}$$

- If $P = Q$ and $y_P = 0$, then $P \oplus Q = \mathcal{O}$.

The size of an elliptic curve over a finite prime field is bounded by the Hasse-Weil inequality:

$$|\#E - p - 1| \leq 2\sqrt{p}. \tag{2.35}$$

It is well-known that the group E is of the form

$$E \cong \mathbb{Z}/L\mathbb{Z} \times \mathbb{Z}/M\mathbb{Z},$$

for two unique integers L, M satisfying $L \mid M$, see [Blake et al., 2000; Cohen et al., 2005; Silverman, 1992] for these and other general properties of elliptic curves.

2.2.2 Linear congruential generator over elliptic curves

Our context is a pseudorandom number generator which produces affine points in an elliptic curve E. One obtains recursively them by operating a fixed element G with the previous value. So, almost always, Equation (2.33) determine the process.

For a given point $G \in E$, the *linear congruential generator over elliptic curves (EC-LCG)* is a sequence (U_n) of pseudorandom points defined by the relation

$$U_n = U_{n-1} \oplus G = nG \oplus U_0, \quad n = 0, 1, \ldots, \tag{2.36}$$

where $U_0 \in E$ is the *initial value* or *seed*. We refer to parameter G as the *composer* of EC-LCG.

It is clear that the period of the sequence defined in (2.36) is equal to the order of G. The EC-LCG, introduced in [Hallgren, 1994], provides a very attractive alternative to linear and non-linear congruential generators with many applications to cryptography and has been extensively studied in the literature, see the articles by Beelen and Doumen [2002]; El Mahassni and Shparlinski [2002]; Gong and Lam [2002]; Gong et al. [2000]; Hess and Shparlinski [2005]; Shparlinski [2003, 2005]. A very recent survey of related problems is [Shparlisnki, 2008].

In the cryptographic setting, the initial value $U_0 = (x_0, y_0)$ and the constants G, a, and b are assumed to be the secret key, and we want to use the output of the generator as a stream cipher. Of course, if two consecutive values U_n are revealed, it is almost always easy to find U_0 and G. So, we output only the most significant bits of each U_n in the hope that this makes the resulting output sequence difficult to predict. But not too many bits can be output at each stage: the linear congruential generator on elliptic curves is polynomial time predictable if sufficiently many bits of its consecutive elements are revealed. We rigorously demonstrate our approach in the special case when the composer G is public. We show that if G and sufficiently many of the most significant bits of two consecutive values U_n, U_{n+1} of the EC-LCG are given, one can

recover the seed U_0 (even in the case where the elliptic curve is private) provided that the first coordinate x_0 of the former value $U_n = (x_n, y_n)$ does not lie in a certain small set. Of course, the assumption that G is public reduces the relevance of the problem in Cryptography, but we believe that the strength of the result we obtain makes this situation of interest in its own right. We also believe that this approach can be extended to the case where G is secret and we present a heuristic approach for this case. Concretely, we show that if sufficiently many of the most significants bits of three values U_n, U_{n+1}, U_{n+2} of the EC-LCG are given, one can recover the seed U_0 and the composer G provided that the first value U_n for which an approximation is used does not lie in a certain small set of exceptional values. This suggests that for cryptographic applications EC-LCG should be used with great care.

More precisely, we say that $W = (x_W, y_W) \in \mathbb{F}_p^2$ is a Δ-*approximation* to $U = (x_U, y_U) \in \mathbb{F}_p^2$ if

$$x_U - x_W, y_U - y_W \in \{-\Delta, 1-\Delta, \ldots, \Delta\} \subseteq \mathbb{F}_p.$$

As in previous scenarios, the case where Δ grows like a fixed power p^δ, with $0 < \delta < 1$, corresponds to the situation where a positive fraction δ of the least significant bits of each number remains hidden.

2.2.3 Predicting Result for Known Composer

Let us formulate and prove our result on predicting the linear pseudorandom number generator on elliptic curves, when the composer G is public. Assume that a, b are unknown, but $G = (x_G, y_G) \in E(\mathbb{F}_p)$ is given to us. We show that when we are given Δ-approximations W_n, W_{n+1} to (respectively) two consecutive affine values U_n, U_{n+1} produced by the EC-LCG; we can recover the exact values, provided that the first component x_n of $U_n = (x_n, y_n)$ does not lie in a certain set, whose size is bounded by $O(\Delta^6)$. Note that once two affine points in a curve as (2.32) are given, such that their first component is different, the curve (the parameters a and b) are determined. Then, after discovering the values U_n and U_{n+1}, we can reproduce (backwards and forwards) the whole sequence. To simplify the notation, we assume that $n = 0$ from now on.

We write $W_j = (\alpha_j, \beta_j)$, $U_j = (x_j, y_j)$, for $j = 0, 1$; and so there exist integers ε_j, η_j with:

$$\begin{aligned} x_j = \alpha_j + \varepsilon_j, \quad y_j = \beta_j + \eta_j \\ |\varepsilon_j|, |\eta_j| \leq \Delta, \quad j = 0, 1. \end{aligned} \tag{2.37}$$

Theorem 2.5 *With the notations above, there exists a set $\mathcal{U}(\Delta; a, x_G, y_G) \subseteq \mathbb{F}_p$ of cardinality $\#\mathcal{U}(\Delta; a, x_G, y_G) = O(\Delta^6)$ with the following property. Whenever $x_0 \notin \mathcal{U}(\Delta; a, x_G, y_G)$, given $\Delta-$approximations W_0, W_1 to two consecutive affine values U_0, U_1 produced by the linear congruential generator on elliptic curves* (2.36)*, and given the value of $G = (x_G, y_G)$, one can recover the seed U_0 in polynomial time.*

Proof. We assume that $x_0 \in \mathbb{F}_p$ is chosen so as not to lie in a certain subset $\mathcal{U}(\Delta; a, x_G, y_G)$ of $\mathbb{F}_p$. The cardinality of this set is bounded by $O(\Delta^6)$. It consists of the solutions of a certain polynomial together with two other values. Its construction is explained throughout the proof.

We place the value $x_G \in \mathcal{U}(\Delta; a, x_G, y_G)$, so that U_0 is not G or $-G$. Then, clearing denominators in Equations (2.33), we can translate

$$U_1 = U_0 \oplus G \tag{2.38}$$

into the following identities in the field $\mathbb{F}_p$:

$$\begin{cases} L_1 = L_1(x_0, y_0, x_1) \equiv 0 \bmod p \\ L_2 = L_2(x_0, y_0, x_1, y_1) \equiv 0 \bmod p, \end{cases}$$

where

$$\begin{aligned} L_1 = {x_G}^3 + x_1 {x_G}^2 - x_0 {x_G}^2 - 2\, x_1 x_G x_0 - x_G {x_0}^2 + {x_0}^3 + 2\, y_G y_0 + x_1 {x_0}^2 - {y_G}^2 - {y_0}^2, \\ L_2 = y_1 x_G - y_1 x_0 - y_G x_0 + y_G x_1 - y_0 x_1 + y_0 x_G. \end{aligned} \tag{2.39}$$

Using the identities $x_j = \alpha_j + \varepsilon_j$ and $y_j = \beta_j + \eta_j$ for $j = 0, 1$, these equations become:

$$\begin{aligned} (-2x_G\alpha_0 - 2x_G\alpha_1 + 3\alpha_0^2 + 2\alpha_1\alpha_0 - x_G^2)\,\varepsilon_0 + (\alpha_0^2 - 2x_G\alpha_0 + x_G^2)\,\varepsilon_1 + (2y_G - 2\beta_0)\,\eta_0 + \\ (3\alpha_0 + \alpha_1 - x_G)\,\varepsilon_0^2 + (2\alpha_0 - 2x_G)\,\varepsilon_0\varepsilon_1 + [\varepsilon_0^3 + \varepsilon_0^2\varepsilon_1 - \eta_0^2] = \\ x_G^2\alpha_0 - x_G^2\alpha_1 + x_G\alpha_0^2 - \alpha_1\alpha_0^2 + 2x_G\alpha_0\alpha_1 - \alpha_0^3 - x_G^3 + \beta_0^2 + y_G^2 - 2y_G\beta_0, \end{aligned}$$

$$\begin{aligned} (-\beta_1 - y_G)\,\varepsilon_0 + (y_G - \beta_0)\,\varepsilon_1 + (x_G - \alpha_1)\,\eta_0 + (x_G - \alpha_0)\,\eta_1 - [\varepsilon_0\eta_1 + \varepsilon_1\eta_0] = \\ \beta_1\alpha_0 - x_G\beta_1 + y_G\alpha_0 - y_G\alpha_1 + \beta_0\alpha_1 - x_G\beta_0. \end{aligned}$$

Now, we linearize this polynomial system. Writing

$$\begin{aligned} &A_0 = x_G^2\alpha_0 - x_G^2\alpha_1 + x_G\alpha_0^2 - \alpha_1\alpha_0^2 + 2x_G\alpha_0\alpha_1 - \alpha_0^3 - x_G^3 + \beta_0^2 + y_G^2 - 2y_G\beta_0, \\ &A_1 = -2\, x_G\alpha_1 - 2x_G\alpha_0 + 3{\alpha_0}^2 + 2\alpha_1\alpha_0 - {x_G}^2, \quad A_2 = {\alpha_0}^2 + {x_G}^2 - 2x_G\alpha_0, \\ &A_3 = 2y_G - 2\beta_0, \quad A_4 = 0, \quad A_5 = \alpha_1 + 3\alpha_0 - x_G, \\ &A_6 = -2x_G + 2\alpha_0, \quad A_7 = 0, \quad A_8 = 1, \\ &B_0 = \beta_1\alpha_0 - x_G\beta_1 + y_G\alpha_0 - y_G\alpha_1 + \beta_0\alpha_1 - x_G\beta_0, \quad B_1 = -\beta_1 - y_G, \\ &B_2 = y_G - \beta_0, \quad B_3 = x_G - \alpha_1, \quad B_4 = x_G - \alpha_0, \\ &B_5 = 0, \quad B_6 = 0, \quad B_7 = -1, \quad B_8 = 0, \end{aligned} \tag{2.40}$$

we obtain that vector

$$\begin{aligned} \mathbf{e} = \ & (\Delta^2\varepsilon_0, \Delta^2\varepsilon_1, \Delta^2\eta_0, \Delta^2\eta_1, \Delta\varepsilon_0^2, \Delta\varepsilon_0\varepsilon_1, \Delta(\varepsilon_1\eta_0 + \varepsilon_0\eta_1), \varepsilon_0^3 + \varepsilon_0^2\varepsilon_1 - \eta_0^2) = \\ & (\Delta^2 e_1, \Delta^2 e_2, \Delta^2 e_3, \Delta^2 e_4, \Delta e_5, \Delta e_6, \Delta e_7, e_8) \end{aligned}$$

is a solution of the following linear system of congruences:

$$\begin{cases} \sum_{i=1}^4 A_i X_i + \sum_{i=5}^7 \Delta A_i X_i + \Delta^2 A_8 X_8 \equiv \Delta^2 A_0 \bmod p \\ \sum_{i=1}^4 B_i X_i + \sum_{i=5}^7 \Delta B_i X_i + \Delta^2 B_8 X_8 \equiv \Delta^2 B_0 \bmod p \\ X_1 \equiv X_2 \equiv X_3 \equiv X_4 \equiv 0 \bmod \Delta^2 \\ X_5 \equiv X_6 \equiv X_7 \equiv 0 \bmod \Delta. \end{cases} \tag{2.41}$$

Moreover, $\mathbf{e}$ is a relatively short vector. We have:

$$|e_i| \le \Delta, i = 1, 2, 3, 4,\ |e_i| \le \Delta^2, i = 5, 6,\ |e_7| \le 2\Delta^2,\ |e_8| \le 3\Delta^3;\ \|\mathbf{e}\| \le \sqrt{19}\Delta^3. \tag{2.42}$$

Let Λ be the lattice consisting of integer solutions $\mathbf{x} = (x_1, x_2, \ldots, x_8) \in \mathbb{Z}^8$ of the system of congruences:

$$\begin{cases} \sum_{i=1}^{4} A_i X_i + \sum_{i=5}^{7} \Delta A_i X_i + \Delta^2 A_8 X_8 & \equiv & 0 \bmod p \\ \sum_{i=1}^{4} B_i X_i + \sum_{i=5}^{7} \Delta B_i X_i + \Delta^2 B_8 X_8 & \equiv & 0 \bmod p \\ X_1 \equiv X_2 \equiv X_3 \equiv X_4 & \equiv & 0 \bmod \Delta^2 \\ X_5 \equiv X_6 \equiv X_7 & \equiv & 0 \bmod \Delta. \end{cases} \tag{2.43}$$

We compute a solution $\mathbf{T}$ of the system of congruences (2.41), using linear diophantine equations methods. Applying an algorithm solving the CVP for shift vector $\mathbf{T}$ and lattice Λ, we obtain a vector

$$\mathbf{f} = (\Delta^2 f_1, \Delta^2 f_2, \Delta^2 f_3, \Delta^2 f_4, \Delta f_5, \Delta f_6, \Delta f_7, f_8).$$

We have that $\mathbf{f} = \mathbf{v} + \mathbf{T}$ (where $\mathbf{v}$ is the lattice vector returned by the CVP algorithm) is the vector of minimal norm satisfying equations (2.41), hence $\mathbf{f}$ must have norm at most equal to the norm of the solution $\mathbf{e}$. Using the bounds (2.42), we get:

$$\begin{aligned} &|f_i| \leq \sqrt{19}\Delta, i = 1, 2, 3, 4, \quad |f_i| \leq \sqrt{19}\Delta^2, i = 5, 6, 7, \quad |f_8| \leq \sqrt{19}\Delta^3, \\ &\|\mathbf{f}\| \leq \sqrt{19}\Delta^3. \end{aligned} \tag{2.44}$$

Note that we can compute $\mathbf{f}$ in polynomial time from the information we are given. We might hope that $\mathbf{e}$ and $\mathbf{f}$ are the same, or at least, that we can recover the approximations errors from $\mathbf{f}$. If not, we show that x_0 belongs to a subset $\mathcal{U}(\Delta; a, x_G, y_G) \subseteq \mathbb{F}_p$ of cardinality $\#\mathcal{U}(\Delta; a, x_G, y_G) = O(\Delta^6)$. Vector $\mathbf{d} = \mathbf{e} - \mathbf{f}$ lies in Λ:

$$\mathbf{d} = (\Delta^2 d_1, \Delta^2 d_2, \Delta^2 d_3, \Delta^2 d_4, \Delta d_5, \Delta d_6, \Delta d_7, d_8), \quad d_i = e_i - f_i, i = 1, \ldots, 8.$$

Bounds (2.42) and (2.44) imply $\|\mathbf{d}\| \leq 2\sqrt{19}\Delta^3$ and

$$|d_i| \leq 2\sqrt{19}\Delta, i = 1, 2, 3, 4, \quad |d_i| \leq 2\sqrt{19}\Delta^2, i = 5, 6, 7, \quad |d_8| \leq 2\sqrt{19}\Delta^3. \tag{2.45}$$

If $d_1 \equiv 0 \bmod p$ and $d_3 \equiv 0 \bmod p$, then $U_0 = (x_0, y_0) = (\alpha_0 + f_1, \beta_0 + f_3) \in \mathbb{F}_p^2$ and we can recover the original values U_0 and U_1.

By the same argument, if $d_2 \equiv 0 \bmod p$ and $d_4 \equiv 0 \bmod p$, we have $U_1 = (x_1, y_1) = (\alpha_1 + f_2, \beta_1 + f_4)$. In order to recover $U_0 = U_1 \oplus (-G)$, we need $U_1 \neq -G$. So, let us include the first component of $-2G$, namely $\left(\frac{3x_G^2 + a}{2y_G}\right)^2 - 2x_G$ (see Equation (2.34)), in the set $\mathcal{U}(\Delta; a, x_G, y_G)$.

So, we can assume ($d_1 \neq 0$ or $d_3 \neq 0$), and ($d_2 \neq 0$ or $d_4 \neq 0$). Substituting $\mathbf{d}$ in Equations (2.43) defining lattice Λ we obtain:

$$\sum_{i=1}^{8} A_i d_i \equiv_p 0, \quad \sum_{i=1}^{8} B_i d_i \equiv_p 0. \tag{2.46}$$

Using the definition of A_i, B_i, $i = 1, \ldots, 9$ and after the substitutions $\alpha_i = x_i - \varepsilon_i$ and $\beta_i = y_i - \eta_i$, $i = 0, 1$ in the second congruence of (2.46), we find

$$N(x_0, y_0, x_1, y_1) = N_0 - d_4 x_0 - d_2 y_0 - d_3 x_1 - d_1 y_1 \equiv_p 0, \tag{2.47}$$

where

$$N_0 = -d_7 + d_4\,\varepsilon_0 + d_2\,\eta_0 + d_2\,y_G + d_1\,\eta_1 - d_1\,y_G + d_4\,x_G + d_3\,\varepsilon_1 + d_3\,x_G.$$

We claim that

$$F(x_0) \equiv 0 \bmod p \tag{2.48}$$

for some nonconstant polynomial of degree at most 18 of the form:

$$F(X) = \sum_{i=0}^{18} C_i X^i,$$

where the coefficients $C_i \in \mathbb{F}_p[N_0, d_1, d_2, d_3, d_4]$, $i = 0, \ldots, 18$. Then, for every choice of $d_1, d_2, d_3, d_4, d_7, \varepsilon_0, \varepsilon_1, \eta_0$, and η_1 only a constant number of values x_0 are possible. In order to prove this last claim, we distinguish two cases: $d_1 \not\equiv_p 0$ and $d_1 \equiv_p 0$.

- Case $d_1 \not\equiv_p 0$.

  From (2.47) we obtain that

  $$y_1 = \frac{N_0 - d_2\,y_0 - d_3\,x_1 - d_4\,x_0}{d_1}.$$

  Substituting this expression in the following equation:

  $$E_1(x_1, y_1) = y_1^2 - x_1^3 - a x_1 - b$$

  and clearing denominators, we obtain a polynomial $E_1'(x_0, y_0, x_1)$ in the variables x_0, y_0 and y_1:

  $$E_1'(x_0, y_0, x_1) = E_1\left(x_1, \frac{N_0 - d_2\,y_0 - d_3\,x_1 - d_4\,x_0}{d_1}\right) d_1^2 = 0.$$

  Solving x_1 from equation (2.39), substituting it in $E_1'(x_0, y_0, x_1)$ and clearing denominators (we note that $x_0 = x_G$ belongs to the bad set $\mathcal{U}(\Delta; a, x_G, y_G)$), we obtain a polynomial $A(x_0, y_0)$ of degree 6 with respect the variable y_0:

  $$A(x_0, y_0) = E_1'(x_0, y_0, \left(\frac{y_G - y_0}{x_G - x_0}\right)^2 - x_0 - x_G)(x_G - x_0)^6 = -d_1^2 y_0^6 + \cdots$$

  Let $F(x_0)$ be the resultant of $A(x_0, y_0)$ and the polynomial

  $$E_0(x_0, y_0) = y_0^2 - x_0^3 - a x_0 - b$$

  with respect to the variable y_0:

  $$F(x_0) = \text{resultant}_{y_0}(A(x_0, y_0), E_0(x_0, y_0)) = \sum_{i=0}^{18} C_i x_0^i.$$

  Using MAPLE we have computed the coefficients C_i explicitly, and we present some of these expressions below:

$$
\begin{aligned}
C_{18} &= d_2^4, \\
C_{17} &= -2\, d_2^2 \left(6\, x_G\, d_2^2 + d_4^2\right), \\
C_{16} &= 2\, d_2^2 \left(33\, d_2^2 x_G^2 + d_2^2 a - 4\, d_3\, y_G\, d_2 + 2\, d_4\, N_0 + 12\, x_G\, d_4^2 - 2\, x_G\, d_3\, d_4\right) + d_4^4.
\end{aligned}
\tag{2.49}
$$

Now, we need to prove that $F(x_0)$ is a nonconstant polynomial for every choice of d_1, d_2, d_3, d_4 and N_0. Clearly, if $d_2 \not\equiv 0 \bmod p$, then degree of $F(x_0)$ is 18. Otherwise, we obtain from bounds in (2.45) and Equation (2.49) that

$$C_{18} = 0, C_{17} = 0, C_{16} = d_4^4.$$

Since $d_2 = 0$, then $d_4 \neq 0$ and the degree of $F(x_0)$ is 16.

- Case: $d_1 \equiv 0 \bmod p$.

  From (2.47) we obtain that

  $$x_1 = \frac{N_0' - d_2\, y_0 - d_4\, x_0}{d_3},$$

  where $N_0' = -d_7 + d_4\, \varepsilon_0 + d_2\, \eta_0 + d_2\, y_G + d_4\, x_G + d_3\, \varepsilon_1 + d_3\, x_G$. Substituting this expression in Equation (2.39): $L_1(x_0, y_0, x_1)$, we derive a polynomial $B(x_0, y_0)$ of degree 2 with respect the variable y_0:

  $$B(x_0, y_0) = L_1\left(x_0, y_0, \frac{N_0' - d_2\, y_0 - d_4\, x_0}{d_3}\right) d_3 = -d_3 y_0^2 + \cdots$$

  Let $F(x_0)$ be the resultant of $B(x_0, y_0)$ and $E_0(x_0, y_0) = y_0^2 - x_0^3 - a x_0 - b$ with respect the variable y_0:

  $$F(x_0) = \text{resultant}_{y_0}(B(x_0, y_0), E_0(x_0, y_0)) = -d_2^2 x_0^7 + (4\, x_G\, {d_2}^2 + {d_4}^2) x_0^6 + \cdots$$

  Again, we need to prove that $F(x_0)$ is a non constant polynomial for every choice of d_1, d_2, d_3, d_4 and N_0. Firstly, we note that the degree of $B(x_0, y_0)$ and $E_0(x_0, y_0)$ does not drop when we plug in concrete values d_1, d_2, d_3, d_4 and N_0. specialize well, because the leadings coefficients of $B(x_0, y_0)$ and $e_0(x_0, y_0)$ with respect y_0 are non zero. Secondly, if $d_2 \not\equiv 0 \bmod p$, then degree of $F(x_0)$ is 7. Otherwise, we have that $F(x_0)$ is a polynomial of degree 6 because its leading coefficient is $d_4^2 \neq 0$.

Since F is a non-constant polynomial in x_0 of degree at most 18, the congruence (2.48) can be satisfied for at most 18 values of x_0 once d_i, $i = 1, \ldots, 4$, and N_0 have been chosen. By (2.45) the total number of possible choices for d_1, d_2, d_3, d_4 is $O(\Delta^4)$. On the other hand, N_0 can take $O(\Delta^2)$ distinct values, because writing N_0 as:

$$N_0 = d_7 + d_4\, \varepsilon_0 + d_2\, \eta_0 + d_1\, \eta_1 + d_3\, \varepsilon_1 + (d_2 - d_1)\, y_G + (d_3 + d_4)\, x_G.$$

From bounds in (2.37) and (2.45) we obtain that $d_7 + d_4\, \varepsilon_0 + d_2\, \eta_0 + d_1\, \eta_1 + d_3\, \varepsilon_1 = O(\Delta^2)$. And fixed d_1, d_2, d_3 and d_4 then is fixed $d_2 - d_1$ and $d_3 + d_4$. Hence there are only $O(\Delta^6)$

values of x_0 that satisfy some congruence (2.48). We place these $O(\Delta^6)$ values of x_0 in $\mathcal{U}(\Delta; a, x_G, y_G)$. So all short vectors satisfying (2.41) lead to discover the approximation errors whenever $x_0 \notin \mathcal{U}(\Delta; a, x_G, y_G)$. Finally, if that is not the case, we can trivially calculate $\varepsilon_0, \varepsilon_1$ and then U_n for $n = 0, 1, \ldots$, which finishes the proof. ■

2.2.4 Unknown Composer

In the previous section we have provided an lower bound (namely, 1/6) for the fraction of bits one should hide from each value obtained with EC-LCG in order to avoid lattice attacks which could reveal the sequence. However, it has been assumed that the cryptanalyst has access to the composer G, which places his task in a quite optimistic frame. So, let us suppose now that the parameter G is also private. In this case we require three approximations, instead of two.

We assume that the sequence (U_n) is not known, but for some n, approximations W_j of 3 consecutive values U_{n+j}, $j = 0, 1, 2$ are given. We show that the value $U_n = (x_n, y_n)$ can be recovered from this information if the approximations W_j are sufficiently good. We can suppose that n=0.

We write $W_j = (\alpha_j, \beta_j)$, where $\varepsilon_j = x_j - \alpha_j$, $\eta_j = y_j - \beta_j$ for $j = 0, 1, 2$ verifying

$$|\varepsilon_j|, |\eta_j| \leq \Delta, \quad j = 0, 1, 2 \tag{2.50}$$

So, our input of this new algorithm consists of $\alpha_0, \alpha_1, \alpha_2, \beta_0, \beta_1, \beta_2 \in \mathbb{F}_p$ and the positive integer Δ.

The first attempt to design a such procedure would be, as in the previous section, to suppose that $U_0, U_1 \notin \{G, -G\}$ and use the addition formulae (2.33) to derive a closest vector problem instance whose solution may lead to recover the three values, and the secret parameter G.

In that case, the polynomial equations obtained grow significantly in degree and number of monomials involved; so in order to keep dealing with low-dimensional lattices, we have followed a different approach.

We just consider the information given as approximations to arbitrary points in the same elliptic curve, in such a way that we are not taking advantage from the knowledge of the procedure which has generated them. In other words, we give a method to recover three points lying in an elliptic curve in the form (2.32), given corresponding approximations. And we use that method in the frame of an EC-LCG and three values partially revealed.

The starting point is:

$$\begin{aligned} y_0^2 &= x_0^3 + ax_0 + b, \\ y_1^2 &= x_1^3 + ax_1 + b, \\ y_2^2 &= x_2^3 + ax_2 + b. \end{aligned}$$

Eliminating the curve parameters a, b and assuming that $U_0 \notin \{U_1, -U_1\}$ (that is, $x_0 \neq x_1$), we obtain the following equation:

$$-y_2^2x_1 + y_2^2x_0 + x_2^3x_1 - x_2^3x_0 - x_2y_0^2 + x_2x_0^3 + x_2y_1^2 - x_2x_1^3 - y_1^2x_0 + x_1^3x_0 + x_1y_0^2 - x_1x_0^3 = 0. \tag{2.51}$$

Following the same process as in Section 2.2.3, we substitute $x_i = \alpha_i + \varepsilon_i$, $y_i = \beta_i + \eta_i$ for $i = 0, 1, 2$ in the above equation and obtain a linear system of congruence equations:

$$\begin{cases} \sum_{i=1}^{6} A_i X_i + \sum_{i=7}^{15} A_i \Delta X_i + \sum_{i=16}^{21} A_i \Delta^2 X_i + A_{22}\Delta^3 X_{22} \equiv_p A_0\Delta^3, \\ X_i \equiv_{\Delta^3} 0, \quad i = 1, \ldots, 6 \\ X_i \equiv_{\Delta^2} 0, \quad i = 7, \ldots, 15 \\ X_i \equiv_{\Delta} 0, \quad i = 16, \ldots, 21. \end{cases} \tag{2.52}$$

with at least a solution bounded by $\sqrt{42}\Delta^4$:

$$\mathbf{e} = (\Delta^3 e_1, \ldots, \Delta^3 e_6, \Delta^2 e_7, \ldots, \Delta^2 e_{15}, \Delta e_{16}, \ldots, \Delta e_{21}, e_{22}), \tag{2.53}$$

which first six components contain the approximation errors, and the other ones are polynomial expressions on those:

$$\begin{array}{lllllllll} e_1 = \varepsilon_0, & & e_2 = \varepsilon_1, & & e_3 = \varepsilon_2, \\ e_4 = \eta_0, & & e_5 = \eta_1, & & e_6 = \eta_2, \\ e_7 = \varepsilon_0^2, & & e_8 = \varepsilon_0\varepsilon_1, & & e_9 = \varepsilon_0\varepsilon_2, \\ e_{10} = \varepsilon_1^2, & & e_{11} = \varepsilon_1\varepsilon_2, & & e_{12} = \varepsilon_2^2, \\ e_{13} = \eta_0(\varepsilon_1 - \varepsilon_2), & & e_{14} = \eta_1(\varepsilon_0 - \varepsilon_1), & & e_{15} = \eta_2(\varepsilon_0 - \varepsilon_1), \\ e_{16} = \eta_0^2 - \varepsilon_0^3, & & e_{17} = \eta_1^2 - \varepsilon_1^3, & & e_{18} = \eta_2^2 - \varepsilon_2^3, \\ e_{19} = \varepsilon_0^2(\varepsilon_1 - \varepsilon_2), & & e_{20} = \varepsilon_1^2(\varepsilon_0 - \varepsilon_2), & & e_{21} = \varepsilon_2^2(\varepsilon_1 - \varepsilon_2), \\ e_{22} = \varepsilon_0(\eta_2^2 - \eta_1^2 + \varepsilon_1^3 - \varepsilon_2^3) + \varepsilon_1(\eta_0^2 - \eta_2^2 + \varepsilon_2^3 - \varepsilon_0^3) + \varepsilon_2(\eta_1^2 - \eta_0^2 + \varepsilon_0^3 - \varepsilon_1^3). \end{array}$$

The coefficients $A_i, i = 1, \ldots, 22$ describing the system are easily obtained from the known information $\alpha_i, \beta_i, i = 0, 1, 2$ and Δ. Now, we can find a particular solution $\mathbf{T}$ to the system (2.52) and then apply the CVP algorithm for the shift vector $\mathbf{T}$ and the homogenization lattice obtained from system (2.52):

$$\begin{cases} \sum_{i=1}^{6} A_i X_i + \sum_{i=7}^{15} A_i \Delta X_i + \sum_{i=16}^{21} A_i \Delta^2 X_i + A_{22}\Delta^3 X_{22} \equiv 0 \bmod p, \\ X_i \equiv 0 \bmod \Delta^3, \quad i = 1, \ldots, 6 \\ X_i \equiv 0 \bmod \Delta^2, \quad i = 7, \ldots, 15 \\ X_i \equiv 0 \bmod \Delta, \quad i = 16, \ldots, 21. \end{cases}$$

Then, we obtain an smaller vector $\mathbf{f}$ in polynomial time from the given information. We might hope that $\mathbf{e}$ and $\mathbf{f}$ are the same. This time, we are not giving a rigorous proof to bound the number of possibilities for which this method could fail.

The so-called "Gaussian heuristic" suggests that and s-dimensional lattice Λ with volume $\mathrm{vol}(\Lambda)$ is unlikely to have a nonzero vector which is substantially shorter than $\mathrm{vol}(\Lambda)^{1/s}$. Moreover, if it is known that such a very short vector does exist, then up to a scalar factor it is likely to be the only vector with this property.

Then, vector $\mathbf{e}$ is likely to be the one founded whenever $\Delta^4 < p^{1/22}\Delta^{42/22}$, this is,

$$\Delta < p^{1/46} = p^{0,0217\ldots}.$$

Chapter III

Minimum distance diagrams of Cayley digraphs

In this chapter we show how monomial ideals and lattice theory play a natural role for solving problems related to circulant digraphs, and more generally, Cayley digraphs associated with finite abelian groups. We give routing algorithms and a method of computing the diameter and average distance. Finally, we present a family of circulants of diameter d and degree r connecting $O(d^r)$ vertices.

In [Stone, 1970], an organization of a multimodule memory is described. Its purpose is to provide an efficient way of moving blocks of data within the memory, not necessarily within the same module. It uses a device called *circulator*, consisting of as many registers as the number of modules in which the memory is divided. Each register is directly connected by a link to r other registers, forming a cyclical symmetric structure. In short, r numbers or *jumps* $j_1, \ldots, j_r$ are selected, and each register is connected, for every $i = 1, \ldots, r$, to the one located j_i positions forwards, considering (as in a ring) that the last and first registers are consecutive. Figure 3.1 depicts a circulator with 8 registers, $r = 2$, $j_1 = 1$, and $j_2 = 3$.

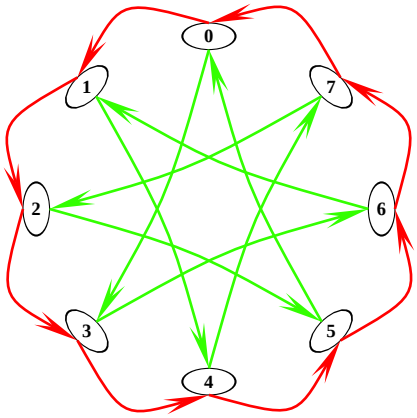


Figure 3.1: $C_8(1, 3)$

The circulator is aimed to provide a connecting path between every pair of registers. In other words, the corresponding graph is expected to be *strongly connected*. A simple way to do so is just building a link from each register to the next one (i.e., $r = 1$, $j_1 = 1$). The drawback of this choice is the high number of steps required to send a datum between some pairs of registers. For instance, if N is the number of memory modules, a path connecting a register with its predecessor needs $N-1$ chords or steps. In the opposite side, one could design a "complete" circulator with a link between any pair of registers. This is achieved with the parameters $r = N-1$, $j_i = i$, for $i = 1, \ldots, N-1$. In this structure the transmission delay is minimal (only one step for any pair of registers), but $N^2 - N$ (unidirectional) links are required. One should look for a tradeoff solution with not too high *degree* r that permits a path between every pair of registers in a number of steps as small as possible. In the Figure 3.1 example, 8 registers are interconnected with 16 links, and the maximum transmission delay between two registers is 3 steps.

A natural problem is then to determine which choice of the jumps $j_1, \ldots, j_r$ leads to an optimal network, provided that the number of registers N and the out-degree r of each are fixed. As those two parameters grow, an exhaustive search becomes impracticable. In [Wong and Coppersmith, 1974], the quality of the circulator is measured by two parameters: the *diameter* and the *average distance*. The former is the maximum over every pair (i, j) of registers of the minimum number of steps required to send a datum from i to j. The latter is the average of those distances. That article provides lower bounds for those parameters and proposes a family of networks which attains a reasonable good approximation to the optimal. Those lower bounds are obtained from the analysis of some properties of a diagram (see Figure 3.2) representing a shortest way for joining any pair of nodes.

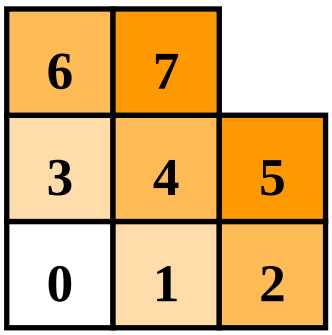


Figure 3.2: Minimum distance diagram for $C_8(1, 3)$.

The (directed) graph defined by the circulator proposed in [Stone, 1970] is the Cayley digraph of a cyclic group and is called *circulant digraph*. This structure, as well as its undirected version (*circulant graph*) in which any link is bidirectional, has been the subject of intense investigation due to its applications in Computer Science such as block transfers in a multimodule memory as we have already pointed out, communication networks, and VLSI design. Bermond et al. [1995] review several results for both directed and undirected cases, paying special attention to graphs with degree

two. Other references on this subject are [Boesch and Tindell, 1984; Cai et al., 1999; Cheng and Hwang, 1988; Cheng et al., 1992; Erdős and Hsu, 1992; Espona and Serra, 1998; Guan, 1998; Hsu and Jia, 1994; Hwang, 2001, 2003].

In this chapter we expose the results published in [Gómez et al., 2005a, 2007a,b]. We use monomial ideals to represent the minimum distance diagram of a circulant digraph of arbitrary degree. This permits the usage of Gröbner bases to give algorithms for computing the diameter and average distance. We also show a specifically tailored algorithm for computing a shortest path in circulants with two jumps, both in the directed and undirected cases.

3.1 Definitions

A *directed graph* or *digraph* is a pair (V, A), where V is a set and A is a subset of the cartesian product V^2. The elements of V are called *vertices* or *nodes* and those of A are called *arcs* of the graph. If $\alpha = (g, h) \in A$, g and h are called the *origin* and *destination* of α, respectively. A *path* in a digraph (V, A) is a list of arcs $(\alpha_1, \ldots, \alpha_l)$ such that the origin of α_i coincides with the destination of α_{i-1}, for $1 < i \leq l$. The origin and destination of this path are those of α_1 and α_l, respectively. The number of arcs a path consists of (l, according to previous notation) is called the *length* of the path.

The *out-degree* of a vertex g in a digraph is the number of arcs with origin g. Accordingly, the *in-degree* of a vertex is defined as the number of arcs with that vertex as destination. A digraph is called *regular* of *degree* r if the out-degree and the in-degree of every vertex is r.

A directed graph is *strongly connected* if for every pair of nodes $(g, h) \in V^2$ there is a path connecting them, i.e., with origin g and destination h. It is *connected* if the associated undirected graph

$$(V, A \cup A^t) = (V, \{(g, h), (h, g) \mid (g, h) \in A\})$$

is strongly connected.

Let Γ be a group and $S \subseteq \Gamma$ a subset. The *Cayley digraph* $C(\Gamma, S)$ is a directed graph (V, A) whose vertex set is $V = \Gamma$ and whose arc set is $A = \{(g, h) \in \Gamma^2 \mid g^{-1}h \in S\}$. In other words, arc (g, gs) is in the graph if and only $g \in \Gamma$ and $s \in S$. For instance, Figure 3.3 represents the way a permutation and a three element cycle generate the symmetric group of three elements.

Cayley digraphs are *vertex-transitive* (or *vertex-symmetric*), i.e., for every pair of nodes (g, h) there exists a permutation $\sigma \in \Sigma_\Gamma$ such that $\sigma(g) = h$ and $(i, j) \in A \iff (\sigma(i), \sigma(j)) \in A$. In other words, the automorphism group of the graph is transitive, or less formally, every node plays the same role in the graph. As a consequence, any Cayley digraph $C(\Gamma, S)$ is regular, and its degree equals $\#S$.

These graphs are connected if and only if the set S generates the group. When dealing with Cayley digraphs of a finite group, this condition is also equivalent to being strongly connected, for every element $s \in S$ has finite order o and $s^{-1} \in \langle S \rangle$, in other words, a "reversed" arc (gs, g) can be replaced by a path $(gs, gs^2, \ldots, gs^o = g)$. Then, we need not distinguish between those two connection definitions. We will consider

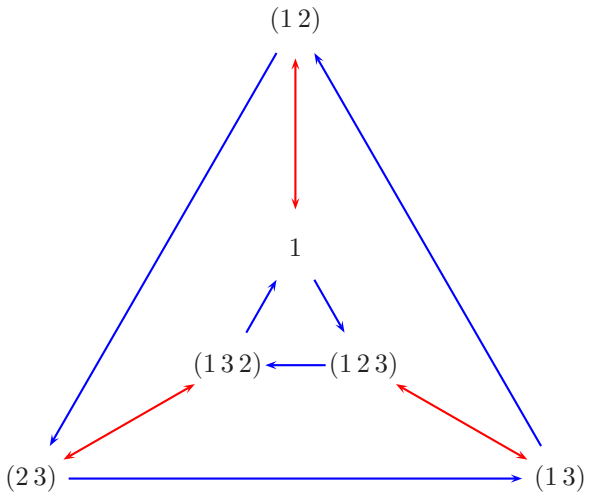


Figure 3.3: The Cayley digraph $C(\Sigma_3, \{(1\,2), (1\,2\,3)\})$.

digraphs associated to finite abelian groups, but we are mainly interested in those associated to cyclic groups. Let N be a positive integer and $\mathbb{Z}_N$ the additive group of integers modulo N. For $S = \{j_1, \ldots, j_r\}$, we denote by $C_N(S) = C_N(j_1, \ldots, j_r)$ the corresponding Cayley digraph (see Figure 3.1), which is called the *circulant digraph* of jumps $j_1, \ldots, j_r$. It is connected if and only if $\gcd(j_1, \ldots, j_r, N) = 1$. If $-j \in S$, for every $j \in S$, then $C_N(S)$ is an undirected graph called *circulant graph*.

We finish this section defining the concepts of diameter and average distance in a digraph (V, A). For nodes $g, h \in V$, the *distance* $d(V, A; g, h)$ from g to h is defined as the minimum length of a path with origin g and destination h. We use the convention that a vertex can be joined with itself by means of an empty path, and therefore, $d(V, A; g, g) = 0$. The *diameter* of the digraph is the maximum over every pair of nodes of their distance:

$$d(V, A) := \max\{d(V, A; g, h) \mid g, h \in V\}.$$

Similarly, the *average distance* of a digraph with a finite number of nodes is the average of the distances between pairs of nodes:

$$\bar{d}(V, A) := \frac{1}{\#V^2} \sum_{g,h \in V} d(V, A; g, h). \tag{3.1}$$

Some authors (see [Bermond et al., 1995], for instance) exclude the trivial distances from a node to itself from this definition:

$$\frac{1}{\#V^2 - \#V} \sum_{\substack{g,h \in V \\ g \neq h}} d(V, A; g, h).$$

but the use of Equation (3.1) leads to a simpler computation (see Section 3.6.2).

3.1.1 Monomial Ideals

Monomial ideals form an important link between Commutative Algebra and Combinatorics. Here we review several basic related results and definitions concerning monomial ideals. Further explanations can be found in [Becker and Weispfenning, 1993; Sturmfels, 1996].

Let $\mathbb{K}$ be an arbitrary field and $\mathbb{K}[X_1, \ldots, X_r]$ the polynomial ring in the variables $X_1, \ldots, X_r$. Throughout this chapter, we very often identify a monomial of $\mathbb{K}[X_1, \ldots, X_r]$ wit its exponent, as a vector of $\mathbb{N}^r$, with the following notation:

$$\mathbf{x}^{\mathbf{a}} = X_1^{a_1} \cdots X_r^{a_r} \longleftrightarrow \mathbf{a} = (a_1, \ldots, a_r).$$

For vectors $\mathbf{a} = (a_1, \ldots, a_r), \mathbf{b} = (b_1, \ldots, b_r) \in \mathbb{N}^r$, we write

- $\mathbf{a} < \mathbf{b}$ if $a_i < b_i$, $\forall i = 1, \ldots, r$,
- $\mathbf{a} \leq \mathbf{b}$ if $a_i \leq b_i$, $\forall i = 1, \ldots, r$.

Note that $\mathbf{x}^{\mathbf{a}} \mid \mathbf{x}^{\mathbf{b}} \iff \mathbf{a} \leq \mathbf{b}$. We also employ the notation

$$\mathbf{e}_i = (0, \ldots, \overset{i}{\smile}{1}, \ldots, 0), \ \mathbf{1} := (1, \ldots, 1).$$

For the sake of simplicity in the characterization given in Equation (3.3), we introduce the following symbol:

$$\mathbf{a} = (a_1, \ldots, a_r) \sqsubset \mathbf{b} = (b_1, \ldots, b_r) \overset{\text{def}}{\iff} (b_i > 0 \Rightarrow a_i < b_i)\,.$$

In the case of our interest, the expression $\mathbf{a} \sqsubset \mathbf{b}$ reduces to $\mathbf{a} < \mathbf{b}$, for every variable occurs in monomial $\mathbf{x}^{\mathbf{b}}$, i.e., $\mathbf{1} \leq \mathbf{b}$.

A *monomial ideal* is an ideal generated by monomials, i.e., $I \subset \mathbb{K}[X_1, \ldots, X_r]$ is a monomial ideal if there is a subset $A \subseteq \mathbb{N}^r$ such that:

$$I = (\mathbf{x}^{\mathbf{a}} \mid \mathbf{a} \in A)\,.$$

An ideal that cannot be decomposed as the proper intersection of two ideals is called *irreducible*. A monomial ideal I is irreducible if and only if there exists $\mathbf{a} \in \mathbb{N}^r$ such that $I = (X_i^{a_i} \mid a_i > 0)$. We write $I = \mathfrak{m}^{\mathbf{a}}$.

There are two standard ways of describing an arbitrary monomial ideal distinct from the zero ideal (0) and the entire ring $\mathbb{K}[X_1, \ldots, X_r]$:

- via the (unique) minimal system of monomial generators, $I = (\mathbf{x}^{\mathbf{a}_1}, \ldots, \mathbf{x}^{\mathbf{a}_s})$, we have:
$$\mathbf{x}^{\mathbf{u}} \in I \iff \exists\, i \in \{1, \ldots, s\} \mid \mathbf{a}_i \leq \mathbf{u}. \tag{3.2}$$
- via the (unique) irredundant decomposition by irreducible monomial ideals, $I = \mathfrak{m}^{\mathbf{b}_1} \cap \cdots \cap \mathfrak{m}^{\mathbf{b}_f}$, we have:
$$\mathbf{x}^{\mathbf{u}} \notin I \iff \exists\, i \in \{1, \ldots, f\} \mid \mathbf{u} \sqsubset \mathbf{b}_i. \tag{3.3}$$

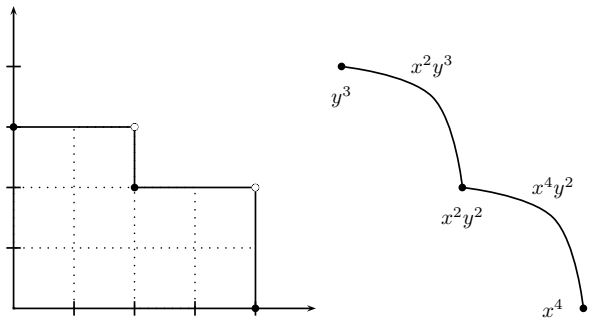


Figure 3.4: Staircase diagram and Buchberger's graph.

The so-called staircase diagram is a useful graphical representation of monomial ideals. For example, the monomial ideal $I_1 := (x^4, x^2y^2, y^3) = (x^2, y^3) \cap (x^4, y^2)$ is represented on the left in Figure 3.4.

Both ways of representing a monomial ideal are interchangeable. One can obtain the irredundant decomposition from the system of generators, and vice versa, using for instance Alexander duality, as explained in [Miller, 2000]. An irreducible component $\mathfrak{m}^{\mathbf{a}}$ can be associated to a "corner" in the corresponding graphic, namely, $\operatorname{lcm}(X_1^{a_1}, \ldots, X_r^{a_r}) = \mathbf{x}^{\mathbf{a}}$. On the other hand, if $\mathbb{K}[X_1, \ldots, X_r]/I$ is an artinian ring then the monomial $\mathbf{x}^{\mathbf{a}}$ associated to the irreducible component $\mathfrak{m}^{\mathbf{a}}$ must coincide with the least common multiple of a subset of the minimal generators of I. In the example of Figure 3.4 we have:

$$x^2y^3 = \operatorname{lcm}(x^2y^2, y^3),\ x^4y^2 = \operatorname{lcm}(x^4, x^2y^2).$$

The diagram on the right in Figure 3.4 is called *Buchberger's graph* of the monomial ideal I_1, see [Miller and Sturmfels, 1999]. At any stage in Buchberger's algorithm for computing Gröbner bases, one considers the S-pairs among the current polynomials and removes those which are redundant; the minimal S-pairs define a graph on the generators of any monomial ideal.

Theorem 3.1 *Let I be a nontrivial monomial ideal given by a minimal system of generators $I = (\mathbf{x}^{\mathbf{a}_1}, \ldots, \mathbf{x}^{\mathbf{a}_s})$ and by the irredundant irreducible decomposition $I = \mathfrak{m}^{\mathbf{b}_1} \cap \cdots \cap \mathfrak{m}^{\mathbf{b}_f}$. The following are equivalent:*

1. *$\mathbb{K}[X_1, \ldots, X_r]/I$ is an artinian ring.*
2. *$\forall i = 1, \ldots, r$, $\exists j \in \{1, \ldots, s\}$, $\exists \alpha_i \in \mathbb{N}$ s.t. $\mathbf{a}_j = \alpha_i \mathbf{e}_i$.*
   In other words, there is a pure power of every variable in the minimal system of generators.
3. *$\forall i = 1, \ldots, f,\ \forall j = 1, \ldots, r,\ b_{i,j} > 0$.*

Proof. We need to prove that the number of monomials outside I is finite if and only if either of the two last items is satisfied. We do that using the characterizations in (3.2) and (3.3).

If the second item is true, then the number of monomials which do not lie in the ideal is bounded by the product $\prod \alpha_i$. Conversely, if that item is false, there exists an index $i \in \{1, \ldots, r\}$ such that $X_i^\alpha \notin I,\ \forall \alpha \in \mathbb{N}$.

For the third item, $\mathbb{N}^r \backslash I$ is the union of the complements of $\mathfrak{m}^{\mathbf{b}_1}, \ldots, \mathfrak{m}^{\mathbf{b}_f}$. For $i = 1, \ldots, f$, $\mathbb{N}^r \backslash \mathfrak{m}^{\mathbf{b}_i}$ is finite if and only if $b_{i,j} > 0$, $\forall j = 1, \ldots, r$. ■

We conclude this section illustrating those facts by an example:

Example 3.2 *In [Miller and Sturmfels, 1999], a planar graph is associated to every monomial ideal in three variables satisfying Theorem 3.1 conditions. The monomial* $\mathbf{x}^{\mathbf{b}}$ *associated to an irreducible component* $\mathfrak{m}^{\mathbf{b}}$ *is identified with a connected component in the graph's complement and can be obtained as the least common multiple of the generators in its boundary. In Figure 3.5 we show this construction for the following ideal:*

$$\begin{aligned} I_2 := \quad & (x^8, x^4y^2, y^5, y^3z, z^5, x^3z^4, x^7z, x^3y^2z^2) = \\ & (x^8, y^2, z) \cap (x^7, y^2, z^4) \cap (x^4, y^3, z^2) \cap (x^4, y^5, z) \cap (x^3, y^3, z^5). \end{aligned} \tag{3.4}$$

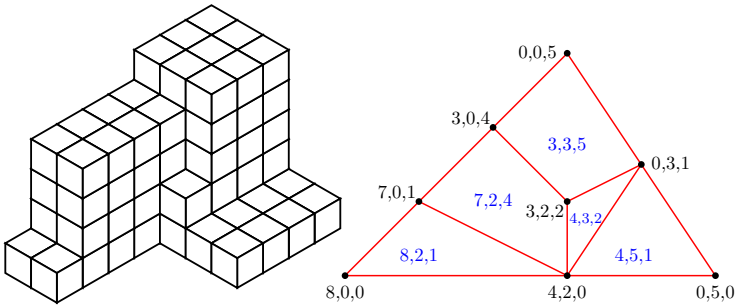


Figure 3.5: Planar graph associated to I_2.

The description of those relations permits the simplification of some computations on Cayley digraphs, as pointed out in Section 3.6.

3.2 Minimum Distance Diagrams

There are different ways to relate monomial ideals with graphs (see, for instance, [Sturmfels, 1996]). In this section we propose a method for studying Cayley digraphs in which we associate a graph with a monomial ideal. We consider the routing problem in a Cayley digraph $C = C(\Gamma, \{s_1, \ldots, s_r\})$, where Γ is finite and abelian. From now on, we use the additive notation for the operation in Γ. As the graph is vertex-symmetric, the problem is reduced to paths originating at node 0_Γ, for a path with origin g and destination h is easily identified with a path with origin 0_Γ and destination $h - g$.

We have defined a path as a list of arcs $(\alpha_1, \ldots, \alpha_l)$ where the origin of one of them coincides with the destination of the former. By the definition of a Cayley digraph,

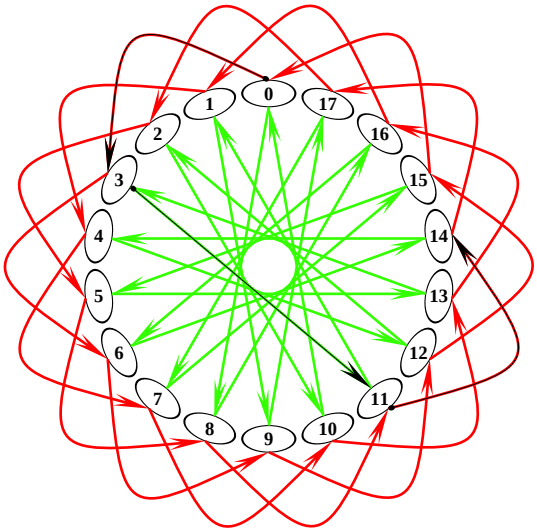


Figure 3.6: Path $(2,1)$ in $C_{18}(3,8)$.

each arc in that list is of the form $(g_i, g_i + s_{j_i})$, where $1 \leq j_i \leq r$. Then, each arc is determined by its origin and its "type" or "color" j_i.

For instance, Figure 3.6 depicts a path in $C_{18}(3,8)$ with origin 0 and destination 14. It includes two chords of the first color and one chord of the second. Considering only paths with the same origin, as long as we remain only interested in the destination, and length (number of chords) of a path, we can identify two paths when the number of chords of each color is the same for both. Then, path $((0,3)(3,11)(11,14))$ represented in Figure 3.6 lies in the same class as $((0,3)(3,6)(6,14))$ and $((0,8)(8,11)(11,14))$.

In general, we can identify the set of (class of) paths in C with origin at 0_Γ with $\mathbb{N}^r$, where $\mathbf{a} = (a_1, \ldots, a_r)$ is a path with a_i chords of type i, i.e., joining a vertex g with $g + s_i$. Let R be the mapping that associates a path originating at 0_Γ with its destination:

$$\begin{array}{rccc} R: & \mathbb{N}^r & \longrightarrow & \Gamma \\ & \mathbf{a} & \mapsto & a_1 s_1 + \cdots + a_r s_r. \end{array} \tag{3.5}$$

The set of paths $\mathbb{N}^r$ is labelled by R with its destination. It is easy to extend R to a homomorphism from $\mathbb{Z}^r$:

$$\begin{array}{rccc} \bar{R}: & \mathbb{Z}^r & \longrightarrow & \Gamma \\ & \mathbf{a} & \mapsto & a_1 s_1 + \cdots + a_r s_r. \end{array} \tag{3.6}$$

This is the natural extension for considering "undirected" paths, i.e., paths in the undirected graph $(V, A \cup A^t)$. Let Λ be the kernel of $\bar{R}$. Obviously, it is an integer lattice and induces an equivalence relation in $\mathbb{Z}^r$: two paths are equivalent if and only if they have the same destination.

Definition 3.3 *Let Γ be a finite abelian group and $S = \{s_1, \ldots, s_r\} \subseteq \Gamma$. The lattice associated with the Cayley digraph $C(\Gamma, S)$ is the kernel of the mapping in Equation (3.6).*

By selecting one path (in $\mathbb{N}^r$) with label g for every vertex g in the graph we build a routing scheme that can be regarded as a handbook to use the graph: for any pair of vertices (g, h), it proposes a path (that labelled by $h-g$) with origin g and destination h. Figure 3.2 depicts a routing scheme with a convenient property: the path proposed for any destination g is optimal, for there is no path with fewer chords connecting 0_Γ and g. Note that the length of a path represented as an element in $\mathbb{N}^r$ equals its ℓ_1 norm.

Definition 3.4 *Let Γ be a finite abelian group and S a generating set of Γ. A Minimum Distance Diagram (MDD) of the (connected) digraph $C(\Gamma, S)$ is a mapping:*

$$D: \ \Gamma \ \longrightarrow \ \mathbb{N}^r \ ,$$

such that

$$R(D(g)) = g, \ \forall g \in \Gamma \quad \text{and} \quad \|D(g)\|_1 = \min\{\|\mathbf{x}\|_1 \mid \mathbf{x} \in R^{-1}(g)\}.$$

Sometimes we refer with MDD to the a graphical representation of the mapping, like those from Figures 3.2 and 3.7.

In general, a Minimum Distance Diagram need not be unique, as is shown in Figure 3.7. This happens when the set $R^{-1}(g)$ contains two or more elements with minimum ℓ_1 norm, for some $g \in \Gamma$.

In digraphs of degree two, we can characterize this situation in terms of lattices.

Proposition 3.5 *Let D be an MDD for $C(\Gamma, \{s_1, s_2\})$, where Γ is a finite and abelian group generated by $\{s_1, s_2\}$. There is a different MDD for the same graph if and only if there exists a vector $(T, -T) \in \Lambda$ with $0 < T \leq \max\{a_1, a_2\}$, for some $\mathbf{a} = (a_1, a_2) \in D(\Gamma)$.*

In the example from Figure 3.7, $C_{33}(5, 14)$, the associated lattice is generated by $\{(-16, 1), (-1, -2)\}$:

$$(T, -T) = \alpha(-16, 1) + \beta(-1, -2) \in \Lambda \iff T \in (11).$$

In consequence, this graph admits exactly four MDDs, two of them presenting an "L" shape. A method for obtaining an MDD for circulant digraphs was introduced in [Wong and Coppersmith, 1974]. It corresponds with Algorithm 3.1 below and the graded lexicographic ordering $X_1 \prec \cdots \prec X_r$. It proves that in the degree two case, the obtained diagram has an "L" shape, which can be determined by four integer parameters, as is shown in Figure 3.8. These four parameters a, b, c, d can be easily obtained for a degree two circulant (see [Cheng and Hwang, 1988]) and permit the computation of the diameter and the average distance:

$$d = a + b - \min\{c, d\} - 2,$$

$$\bar{d} = \frac{a^2b + ab^2 + c^2d + cd^2 - 2cd(a+b)}{2(ab - cd)} - 1.$$

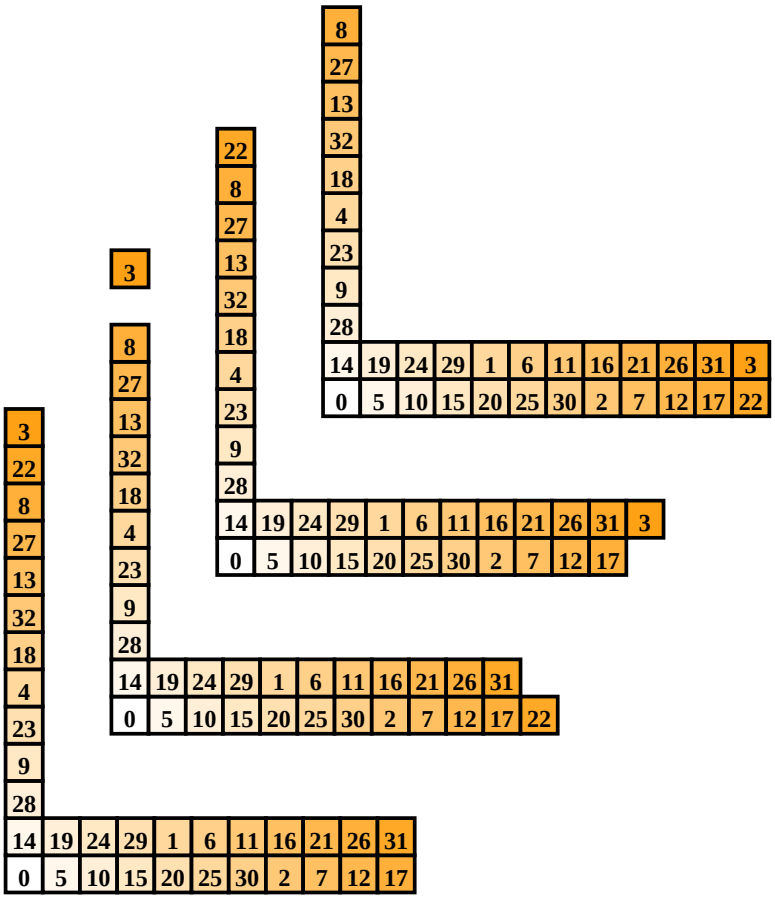


Figure 3.7: Different MDDs for $C_{33}(5, 14)$.

Not any diagram with an "L" shape is the MDD of a circulant digraph. The characterization of those is given in [Fiol et al., 1987].

For Cayley digraphs of degree greater than two, the obtained minimum distance diagrams become more complicated to describe. Monomial ideals are a useful tool to represent these diagrams. In accordance with the previous discussion, a well-ordering in $\mathbb{N}^r$ compatible with the norm ℓ_1 determines a unique MDD. Note than the degree of a monomial equals its ℓ_1 under the identification given in Section 3.1.1. Then, one way of determining a MDD is fix a graded monomial ordering $\prec$ and define:

$$\begin{array}{rccc} D: & \Gamma & \longrightarrow & \mathbb{N}^r \\ & g & \mapsto & \min_{\prec}(R^{-1}(g)). \end{array} \tag{3.7}$$

Every graded monomial ordering defines a mapping

$$s : \mathbb{N}^r \longrightarrow \mathbb{N}^r$$

such that $\mathbf{a} \prec s(\mathbf{a})$ and $\mathbf{a} \prec \mathbf{b} \Rightarrow s(\mathbf{a}) \preceq \mathbf{b}$. This means that the elements in $\mathbb{N}^r$ can be arrayed in an infinite list $\mathbf{a}_1 \prec \mathbf{a}_2 \prec \cdots$. Note that this need not be possible for any monomial ordering, as for instance for a pure lexicographic ordering. This mapping provides a method of constructing the MDD with respect to a fixed monomial ordering.

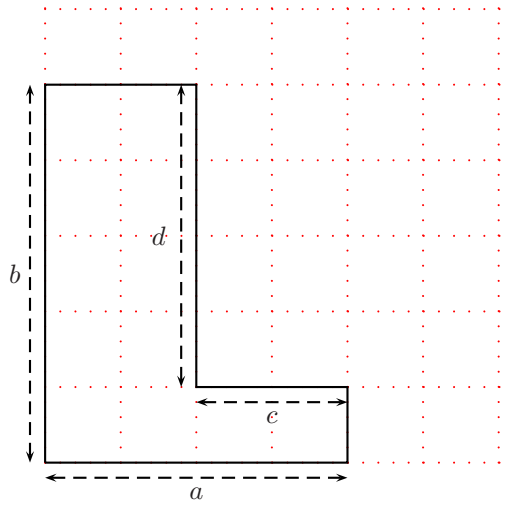


Figure 3.8: An "L" shape.

Algorithm 3.1: MDD construction.

Input: $\Gamma = \{g_i \mid 0 \leq i < N\}$, abelian group, $\{s_1, \ldots, s_r\}$, generating set; s.
Output: $D(g_i),\ i = 0, \ldots, N-1$.

```
1 D[g_0, ..., g_{N-1}] ← ∅̄, S ← 0, a ← 0;
2 while S < N do
3 |   g ← R(a);
4 |   if D(g) = ∅ then
5 |   |   D(g) ← a;
6 |   |   S ← S + 1;
7 |   end
8 |   a ← s(a);
9 end
```

Of course, computing the whole diagram $D[0,\ldots,N-1]$ of a circulant cannot be computationally efficient, its size being exponential in the input size. Furthermore, Algorithm 3.1 performs an exhaustive search that can last at most for $\binom{d+r}{d} \overset{r\ll d}{\sim} \frac{1}{r!}d^r$ loops until reaching its ending, where d is the graph's diameter. The examples in Figure 3.9 illustrate the algorithm's output.

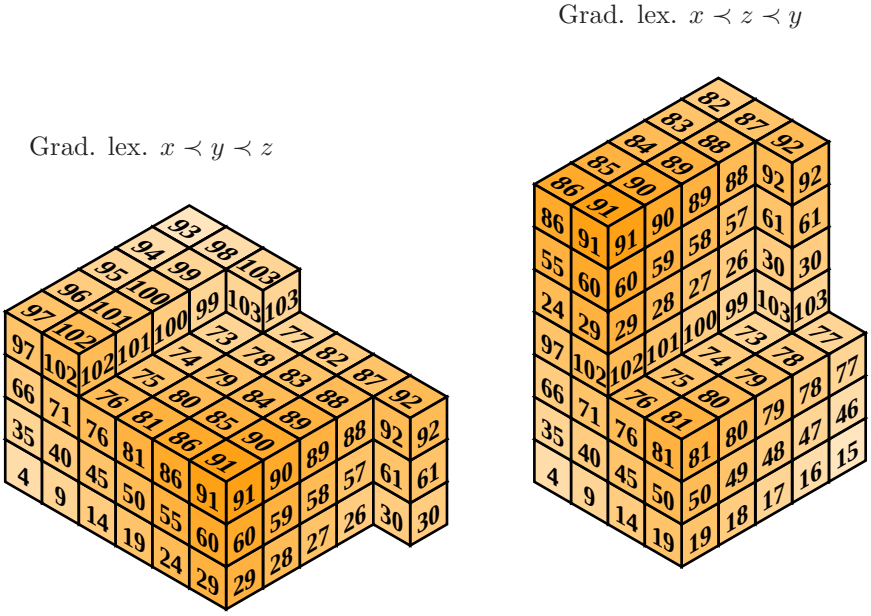


Figure 3.9: MDDs of $C_{37}(3,14,25)$.

Proposition 3.6 *Let Γ be a finite abelian group, S a generating set of Γ, and $\prec$ a graded monomial ordering. Let D be the minimum distance diagram output by Algorithm 3.1 from that input. The monomial ideal generated by paths outside the diagram*

$$I(C(\Gamma,S),\prec) := (\mathbb{N}^r \backslash D(\Gamma))$$

has no monomial in the diagram. This is, $\mathbb{N}^r\backslash D(\Gamma)$ is an ideal in the semigroup of monomials $\mathbb{N}^r$. $I(C(\Gamma,S),\prec)$ is called the monomial ideal associated with $C(\Gamma,S)$ and $\prec$.

Proof. Let $\mathbf{a}$ be an element in the ideal generated by $\mathbb{N}^r\backslash D(\Gamma)$, then $\exists \mathbf{b}\in\mathbb{N}^r, \exists \mathbf{z}\in \mathbb{N}^r\backslash D(\Gamma)$ such that $\mathbf{a}=\mathbf{b}+\mathbf{z}$. Now, $\mathbf{z}\notin D(\Gamma)$, then $\exists \mathbf{u}\in\mathbb{N}^r$, with $R(\mathbf{u})=R(\mathbf{z})$, $\mathbf{u}\prec\mathbf{z}$. Since $\mathbf{u}+\mathbf{b}\prec\mathbf{z}+\mathbf{b}$ and R is a linear map, $R(\mathbf{u}+\mathbf{b})=R(\mathbf{a})$ and $\mathbf{a}\notin D(\Gamma)$. ■

In the Figure 3.9 examples, we have two monomial ideals (J_1 and J_2) associated with $C_{37}(3,14,25)$ and with graded lex $x\prec z\prec y$, and $x\prec y\prec z$, respectively:

$$\begin{aligned} J_1 = {} & (x^5, x^4y, y^2, yz^4, z^5, x^4z) = (x^5,y,z)\cap(x^4,y,z^5)\cap(x^4,y^2,z^4).\\ J_2 = {} & (x^5, x^4y, x^3y^2, x^2y^4, xy^6, y^8, y^7z, y^5z^2, y^3z^3, yz^4, z^5, xz) = \\ & (x,y^7,z^2)\cap(x,y^5,z^3)\cap(x,y^3,z^4)\cap(x,y,z^5)\cap(x^5,y,z)\cap \\ & (x^4,y^2,z)\cap(x^3,y^4,z)\cap(x^2,y^6,z)\cap(x,y^8,z). \end{aligned} \tag{3.8}$$

With Definition 3.4, not every minimum distance diagram is the complement of an ideal in $\mathbb{N}^r$ (there are two examples in Figure 3.7). Moreover, not every minimum distance diagram whose complement is an ideal in the semigroup of monomials arises from a graded monomial ordering (following Algorithm 3.1). In [Sabariego, 2008], MDDs output by Algorithm 3.1 are called *coherent*.

Obviously, a minimum distance diagram D is an injective map and $\#\big(D(\Gamma)\big) = \#\Gamma < \infty$. So, the monomial ideal $I := I(C(\Gamma, S), \prec)$ always contains generators of the form $X_1^{a_1}, \ldots, X_r^{a_r}$; that is, the quotient ring $\mathbb{K}[X_1 \ldots, X_r]/I$ is artinian (see Theorem 3.1). We say that an MDD built from a graded monomial ordering is *degenerated* if I is an irreducible ideal, this is, when the minimal system of generators of I only contains as many generators as the cardinal of S. In general, this is not the case, as is illustrated in the above examples. The following concept is a generalization of L-shapes to arbitrary dimension:

Definition 3.7 *Let I be a monomial ideal and let A be the minimal system of generators of I. We say that I is an L-shape if there exists at most one element $\mathbf{x}^{\mathbf{a}} = X_1^{a_1} \cdots X_r^{a_r} \in A$ such that $a_i > 0$, for all $i = 1, \ldots, r$.*

We say that an MDD built following Algorithm 3.1 is an L-shape if the associated monomial ideal is an L-shape.

This definition is satisfied by every minimum distance diagram whose complement is an ideal in $\mathbb{N}^r$ (as is the case of those output by Algorithm 3.1). We use the following technical result to prove it:

Lemma 3.8 *Let Γ be a finite abelian group, $S = \{s_1, \ldots, s_r\}$ a generating set, and D a minimum distance diagram of $C(\Gamma, S)$ such that $I := \mathbb{N}^r \backslash D(\Gamma)$ is an ideal in $\mathbb{N}^r$. Let A be the minimal system of generators of I. If the exponent of $\mathbf{x}^{\mathbf{a}} \in A$ has some component a_i positive, then $\mathbf{b} = (b_1, \ldots, b_r) := D(R(\mathbf{a}))$ satisfies $b_i = 0$, where R is the routing function defined in (3.5).*

Proof. Since $\mathbf{a}$ is an element of A, $\mathbf{a} \notin D(\Gamma)$. We must have $\mathbf{a} - \mathbf{e}_i \in D(\Gamma)$, because otherwise $\mathbf{a}$ would not be a minimal generator. Now, $\mathbf{b} \prec \mathbf{a}$ and $R(\mathbf{b}) = R(\mathbf{a})$. If we suppose $b_i > 0$ then:

$$R(\mathbf{b} - \mathbf{e}_i) = R(\mathbf{a} - \mathbf{e}_i),\ \mathbf{b} - \mathbf{e}_i \prec \mathbf{a} - \mathbf{e}_i,$$

which contradicts $\mathbf{a} - \mathbf{e}_i \in D(\Gamma)$. ■

Now, we state the main result of this section:

Proposition 3.9 *Let Γ be a finite abelian group, $S = \{s_1, \ldots, s_r\}$ a generating set, and D a minimum distance diagram of $C(\Gamma, S)$ such that $I := \mathbb{N}^r \backslash D(\Gamma)$ is an ideal in $\mathbb{N}^r$. Then, D is an L-shape.*

Proof. Let A be the minimal system of generators of I. If $\mathbf{a} \in A$ is such that $a_i > 0, \forall i$, then by Lemma 3.8 we have $R(\mathbf{a}) = R(\mathbf{0}) = 0_\Gamma$.

Moreover, $\mathbf{a} - \mathbf{e}_1 \in D(\Gamma)$, and $R(\mathbf{a} - \mathbf{e}_1) = -s_1$. So, if $\mathbf{a} \in A$ and $\mathbf{b} \in A$ are two generators with every component positive, then $\mathbf{a} - \mathbf{e}_1 = D(-s_1) = \mathbf{b} - \mathbf{e}_1$. That

completes the proof. ■

Now, the problem is to find the list of generators that describe the ideal associated to a circulant digraph in a convenient way. The following section answers this question.

3.3 Lattice ideals and L-shapes

We present in this section a method to obtain a description of the minimum distance diagram associated to a circulant digraph and a graded monomial ordering. This description is made of a (minimal) system of generators of its complement, which is an ideal in the semigroup of monomials. This description has not polynomial size on the number of vertices, but is substantially more efficient than an exhaustive listing of every monomial outside the ideal, as that provided by Algorithm 3.1.

Later on, we will need to restrict ourselves to circulant graphs. Now, we start in the general case of a connected Cayley digraph $C = C(\Gamma, S)$ whose vertex set is finite and abelian.

In the previous section we have used the integer lattice Λ defined as the kernel of the extended routing mapping $\bar{R}$ (see Definition 3.3). Considering integer vectors as paths in the undirected graph corresponding to C, this lattice consists of the *loops* of the graph: paths with same origin and destination.

We recall the definition of the *leading monomial* of a multivariate polynomial $f = \sum_{i\in A} f_i \mathbf{x}^{\mathbf{a}_i} \in \mathbb{K}[X_1, \ldots, X_r]$ with respect to a monomial ordering $\prec$:

$$\mathrm{lm}_\prec(f) = \mathrm{lm}(f) = \mathbf{x}^{\mathbf{a}}, \text{ where } \mathbf{a} := \max_\prec(A).$$

Note that $\prec$ is a well-ordering in the set of monomials.

We consider the binomial ideal I_Λ associated with Λ, as defined in Section 1.3. As usual, given an ideal J in $\mathbb{K}[X_1, \ldots, X_r]$ and a monomial ordering $\prec$, we denote by $\mathrm{lm}_\prec(J) = \mathrm{lm}(J)$ the monomial ideal generated by the leading monomials of all nonzero elements of J, i.e.,

$$\mathrm{lm}(J) := (\mathrm{lm}_\prec(f) \mid f \in J^*).$$

The following is one of the main results in this section. It uses the Gröbner bases theory, which was introduced by B. Buchberger [1965]. Its use has become widespread in Commutative Algebra and Algebraic Geometry. The theory of Gröbner bases is related to several areas in Mathematics and Computer Science, see [Becker and Weispfenning, 1993; Gutierrez and Rubio San Miguel, 1998; Sturmfels, 1996].

Proposition 3.10 *Let Γ be a finite abelian group, $S = \{s_1, \ldots, s_r\}$ a generating set, and $\prec$ a graded monomial ordering in $\mathbb{N}^r$. We have:*

$$\mathrm{lm}(I_\Lambda) = I(C(\Gamma, S), \prec).$$

Proof. The ideal I_Λ is generated by binomials of the form $\mathbf{x}^{\mathbf{a}} - \mathbf{x}^{\mathbf{b}}$. Then, it admits a Gröbner basis G also consisting of that kind of binomials, for the two type of operations performed in Buchberger' algorithm to compute a Gröbner basis (building an S-polynomial and computing a remainder) are internal in the set of binomials with form $\mathbf{x}^{\mathbf{a}} - \mathbf{x}^{\mathbf{b}}$. Let $\mathbf{x}^{\mathbf{a}}$ be a monomial in $\mathrm{lm}(I_\Lambda)$. There exists a binomial $\mathbf{x}^{\mathbf{a}} - \mathbf{x}^{\mathbf{b}}$ in

basis G, and by Proposition 1.18, $\mathbf{a}-\mathbf{b}\in\Lambda$. Now, as $\mathbf{a}\succ\mathbf{b}$ and both paths have the same destination, $\mathbf{a}\notin D(\Gamma)$. Conversely, let $\mathbf{x}^{\mathbf{a}}\in I$. We take $\mathbf{b}:=D(R(\mathbf{a}))\prec\mathbf{a}$. It is clear that $\mathbf{a}-\mathbf{b}\in\Lambda$, and so, $\mathbf{x}^{\mathbf{a}}-\mathbf{x}^{\mathbf{b}}$ is a binomial in I_Λ, whose leading monomial is $\mathbf{x}^{\mathbf{a}}$. ■

As a consequence of the previous result, if G is a minimal or reduced Gröbner basis of the ideal I_Λ, then the leading monomials of the elements of G constitute a minimal system of generators of our MDD. In order to apply Buchberger's algorithm for computing a finite Gröbner basis of an ideal, we need to start with a finite set of generators. It is easy to do for a circulant digraph using Lemma 1.19.

Proposition 3.11 *Let $C_N(j_1,\ldots,j_r)$ be a connected circulant digraph with associated lattice Λ. We have $I_\Lambda=(X_1^NX_2^N\cdots X_r^N-1,\quad \mathbf{x}^{\mathbf{a}^+}-\mathbf{x}^{\mathbf{a}^-}\mid\mathbf{a}\in Q)$, where*

$$\begin{aligned}Q=\ &\{(N\alpha_1,\ldots,N\alpha_r),(\alpha_1j_1-1,\alpha_2j_2,\ldots,\alpha_rj_r),(\alpha_1j_1,\alpha_2j_2-1,\ldots,\alpha_rj_r),\ldots,\\&(\alpha_1j_1,\ldots,\alpha_{r-1}j_{r-1},\alpha_rj_r-1)\},\end{aligned}\tag{3.9}$$

and $\beta,\alpha_i\in\mathbb{Z},(i=1,\ldots,r)$ satisfy $1=\alpha_1j_1+\cdots+\alpha_rj_r+\beta N$.

Using Propositions 3.10 and 3.11 we can compute a minimal system of generators of I for circulant digraphs. The paper [Sturmfels et al., 1995] also contains results on the complexity of computing the reduced Gröbner basis of lattice ideals and on its size. In particular, it provides an upper bound for the number of elements and shows an example lattice Λ with exponential size in the bit complexity of a basis of Λ. 4ti2 [4ti2 team] is a useful software for computing the reduced Gröbner basis of a binomial ideal.

3.4 Optimal Routing

In this section we show an algorithm to compute a shortest path between two vertices for a circulant digraph using a finite Gröbner basis of I_Λ.

Given a pair of nodes (i,j) in a graph, there are several paths which join the origin i and the destination j. We are interested in *optimal paths*, i.e., those with minimum length. For general graphs, finding a shortest path between two vertices is a well-known and important problem. Efficient polynomial time algorithms have been developed for various routing problems. However, for the family of circulant graphs, there is an important distinction to be made, and that concerns the natural input size to a problem. For an arbitrary graph it is common to consider the input size to be $O(N^2)$, which is the number of bits in its adjacency matrix. However, any circulant graph can be described by only r integers. In this representation the input size is $O(r\log N)$. Thus, polynomial time algorithms for general graphs may exhibit exponential complexity in the special case of circulant graphs, for this compact input representation. In [Cai et al., 1999] it is shown that the Shortest Path problem is NP-hard for this concise representation.

As we have already pointed out, in our case the routing problem is reduced to pairs of nodes $(0,i)$ where the origin node is fixed. Using the well-known Extended Euclidean Algorithm we compute a path $\mathbf{c}$ from vertex 0 to vertex i if it exists.

We can apply general Integer Programming techniques ([Schrijver, 1986]) to find a shortest path for circulant digraphs as follows:

Lemma 3.12 *Paths in $C_N(j_1, j_2, \ldots, j_r)$ with minimum length from node 0 to node i are the solutions to the following integer program:*

$$\min\{\mathbf{d} \cdot \mathbf{x} \mid A\mathbf{x} \geq \mathbf{b},\ \mathbf{x} \in \mathbb{Z}^{r+1}\},$$

where $\mathbf{x} = (x_1, x_2, \ldots, x_r, y) \in \mathbb{Z}^{r+1}$, $\mathbf{d} = (1, 1, \ldots, 1, 0) \in \mathbb{Z}^{r+1}$, $\mathbf{b} = (j, -j, 0, \ldots, 0) \in \mathbb{Z}^{r+2}$, and

$$A = \begin{pmatrix} j_1 & j_2 & \ldots & j_r & N \\ -j_1 & -j_2 & \ldots & -j_r & -N \\ 1 & 0 & \ldots & 0 & 0 \\ 0 & 1 & \ldots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \ldots & 1 & 0 \end{pmatrix} \in \mathbb{Z}^{(r+2)\times(r+1)}.$$

So, with the number of jumps r fixed, we can derive an algorithm to compute a shortest path in circulant digraphs requiring $O(r + \log r \log N)$ arithmetic operations on rational numbers of size $O(\log N)$, see [Eisenbrand, 2003; Gómez et al., 2005c; Kannan, 1987; Lenstra, 1983].

Proposition 3.13 *Let Γ be a finite abelian group, $S = \{s_1, \ldots, s_r\}$ a generating set, and Λ the integer lattice associated with $C(\Gamma, S)$ (Definition 3.3). Let G be a Gröbner basis of I_Λ with respect to a graded monomial ordering $\prec$ and $\mathbf{c} \in \mathbb{N}^r$ a path (not necessarily a shortest one) in $R^{-1}(i)$. Then the normal form of $\mathbf{x}^{\mathbf{c}} - 1$ with respect to G is $\mathbf{x}^{\mathbf{d}} - 1$, where $\mathbf{d}$ is the shortest path from vertex 0 to vertex i with respect to the monomial ordering $\prec$.*

Proof. We have $\mathbf{c} - \mathbf{d} \in \Lambda$, which implies $(\mathbf{x}^{\mathbf{c}} - 1) - (\mathbf{x}^{\mathbf{d}} - 1) \in I_\Lambda$. Clearly, $\mathbf{x}^{\mathbf{d}} - 1$ is a normal form, because $\mathbf{x}^{\mathbf{d}} \notin I$, where I is the initial ideal of I_Λ. ■

This result provides a convenient algorithm to compute a shortest path and then to design optimal routings.

In the rest of this section, we develop a specific method for the case of degree two circulant digraphs and their associated undirected graphs. We address now the question of how to compute a vector that is closest to another given vector in a two dimensional lattice with respect to ℓ_1 norm. We denote by $\|\cdot\|$ this norm acting over a vector. As we are dealing with vectors $\mathbf{u} \in \mathbb{R}^2$, we denote their components by $\mathbf{u} = (u_1, u_2)$. For every real number $x \in \mathbb{R}$, as usual, we denote by $\operatorname{sgn}(x)$ its sign.

3.4.1 Reduction by a vector

Given $\mathbf{u}, \mathbf{v}$ in $\mathbb{R}^2$, with $\mathbf{v} \neq 0$, we can find $\alpha \in \mathbb{Z}$ such that the value $\|\mathbf{u} - \alpha\mathbf{v}\|$ is minimal, that is, we want to make $\mathbf{u}$ as short as possible by subtracting an integer multiple of $\mathbf{v}$ (see [Kaib and Schnorr, 1996; Micciancio and Goldwasser, 2002]). The algorithm `REDUCE` (see Algorithm 3.2) is the basic tool of this section.

Lemma 3.14 *Algorithm 3.2 is correct.*

Algorithm 3.2: REDUCE procedure

Input: $\mathbf{u}, \mathbf{v} \in \mathbb{R}^2,\ \mathbf{v} \neq \mathbf{0}$.

Output: $\text{Reduce}_{\mathbf{v}}(\mathbf{u}) \in \mathbf{u} + \mathbb{Z}\mathbf{v}$ s.t. $\|\text{Reduce}_{\mathbf{v}}(\mathbf{u})\| = \min\{\|\mathbf{u} + \alpha\mathbf{v}\| \mid \alpha \in \mathbb{Z}\}$.

1 Select $i \in \{1,2\},\ j \in \{1,2\}\backslash\{i\}$ such that $|v_i| > |v_j|$. If $|v_1| = |v_2|$, then $i = 1$;

2 Return the vector with minimum norm between:

$$\mathbf{u} - \left\lfloor \frac{u_i}{v_i} \right\rfloor \mathbf{v}, \quad \mathbf{u} - \left\lceil \frac{u_i}{v_i} \right\rceil \mathbf{v}.$$

If both have the same norm, return the one with ith non-negative component;

Proof. We just look at the mapping:

$$\begin{array}{rccc} f: & \mathbb{R} & \longrightarrow & \mathbb{R} \\ & \alpha & \mapsto & \|\mathbf{u} + \alpha\mathbf{v}\| \end{array}$$

We aim to minimize $f(\alpha) = |u_1 + \alpha v_1| + |u_2 + \alpha v_2|$ among integers. Let's separate two cases:

- $v_j = 0$
  It must be $v_i \neq 0$, because we are supposing $\mathbf{v} \neq 0$. Then, $f(\alpha) = |u_j| + |u_i + \alpha v_i|$ takes its minimum at $\dfrac{-u_i}{v_i}$.

- $v_1 v_2 \neq 0$
  We have then two singular points:

$$\frac{-u_1}{v_1},\ \frac{-u_2}{v_2},$$

$$f\left(\frac{-u_k}{v_k}\right) = \frac{|u_1 v_2 - u_2 v_1|}{|v_k|}.$$

  So, f reaches also its minimum at $\dfrac{-u_i}{v_i}$; although the map takes the same value in the gap limited by both singular points whenever $|v_1| = |v_2|$.

Once this is seen, it is clear that the integer that minimizes f must be in:

$$\left\{ \left\lfloor \frac{-u_i}{v_i} \right\rfloor, \left\lceil \frac{-u_i}{v_i} \right\rceil \right\}.$$

■

Next result collects several properties of this concept for later use.

Lemma 3.15 *Let $\mathbf{u}, \mathbf{v} \in \mathbb{R}^2$ be two vectors such that $\mathbf{v} \neq 0$. Let $i \in \{1,2\},\ j \in \{1,2\}\backslash\{i\}$ as in Algorithm 3.2, that is, $|v_i| > |v_j| \ \vee (i = 1 \ \wedge \ |v_1| = |v_2|)$. We have the following properties:*

1. $\mathbf{r} = \text{Reduce}_{\mathbf{v}}(\mathbf{u}) \Rightarrow |r_i| < |v_i|$.

2. $\mathbf{h} \in \mathbf{u} + \mathbb{Z}\mathbf{v} \Rightarrow \mathrm{Reduce}_{\mathbf{v}}(\mathbf{h}) = \mathrm{Reduce}_{\mathbf{v}}(\mathbf{u})$.

3. $\mathrm{Reduce}_{\mathbf{v}}(\mathrm{Reduce}_{\mathbf{v}}(\mathbf{u})) = \mathrm{Reduce}_{\mathbf{v}}(\mathbf{u})$.

4. $\mathrm{Reduce}_{\mathbf{v}}(\mathbf{u}) = \mathrm{Reduce}_{-\mathbf{v}}(\mathbf{u})$.

Proof.

1. The number r_i is

$$u_i - \left\lfloor \frac{u_i}{v_i} \right\rfloor v_i, \quad \text{or} \quad u_i - \left\lfloor \frac{u_i}{v_i} \right\rfloor v_i - v_i.$$

We divide the study into several cases:

- If $v_i > 0$:

$$\left\lfloor \frac{u_i}{v_i} \right\rfloor \leq \frac{u_i}{v_i} \Rightarrow \left\lfloor \frac{u_i}{v_i} \right\rfloor v_i \leq u_i \Rightarrow u_i - \left\lfloor \frac{u_i}{v_i} \right\rfloor v_i \geq 0,$$

$$\left\lfloor \frac{u_i}{v_i} \right\rfloor + 1 > \frac{u_i}{v_i} \Rightarrow \left\lfloor \frac{u_i}{v_i} \right\rfloor v_i + v_i > u_i \Rightarrow u_i - \left\lfloor \frac{u_i}{v_i} \right\rfloor v_i < v_i.$$

- If $v_i < 0$:

$$\left\lfloor \frac{u_i}{v_i} \right\rfloor \leq \frac{u_i}{v_i} \Rightarrow \left\lfloor \frac{u_i}{v_i} \right\rfloor v_i \geq u_i \Rightarrow u_i - \left\lfloor \frac{u_i}{v_i} \right\rfloor v_i \leq 0,$$

$$\left\lfloor \frac{u_i}{v_i} \right\rfloor + 1 > \frac{u_i}{v_i} \Rightarrow \left\lfloor \frac{u_i}{v_i} \right\rfloor v_i + v_i < u_i \Rightarrow u_i - \left\lfloor \frac{u_i}{v_i} \right\rfloor v_i > v_i.$$

For the last one:

- If $v_i > 0$:

$$0 = v_i - v_i > u_i - \left\lfloor \frac{u_i}{v_i} \right\rfloor v_i - v_i \geq 0 - v_i.$$

And when $u_i - \left\lfloor \frac{u_i}{v_i} \right\rfloor v_i - v_i = -v_i$, we have:

$$u_i = \left\lfloor \frac{u_i}{v_i} \right\rfloor v_i \Rightarrow \frac{u_i}{v_i} = \left\lfloor \frac{u_i}{v_i} \right\rfloor \Rightarrow \frac{u_i}{v_i} \in \mathbb{Z} \Rightarrow r_i = 0.$$

- If $v_i < 0$:

$$0 = v_i - v_i < u_i - \left\lfloor \frac{u_i}{v_i} \right\rfloor v_i - v_i \leq 0 - v_i.$$

And when $u_i - \left\lfloor \frac{u_i}{v_i} \right\rfloor v_i - v_i = -v_i$, we have:

$$u_i = \left\lfloor \frac{u_i}{v_i} \right\rfloor v_i \Rightarrow \frac{u_i}{v_i} = \left\lfloor \frac{u_i}{v_i} \right\rfloor \Rightarrow \frac{u_i}{v_i} \in \mathbb{Z} \Rightarrow r_i = 0.$$

2. It must be $\exists \alpha \in \mathbb{Z} \mid \mathbf{h} = \mathbf{u} + \alpha\mathbf{v}$. Then,

$$\left\lfloor \frac{h_i}{v_i} \right\rfloor = \left\lfloor \frac{u_i}{v_i} + \alpha \right\rfloor = \left\lfloor \frac{u_i}{v_i} \right\rfloor + \alpha.$$

So, the two possibilities for $\text{Reduce}_{\mathbf{v}}(\mathbf{h})$ are:

$$\mathbf{h} - \left(\left\lfloor \frac{u_i}{v_i} \right\rfloor + \alpha \right) \mathbf{v} = \mathbf{u} - \left\lfloor \frac{u_i}{v_i} \right\rfloor \mathbf{v},$$

$$\mathbf{h} - \left(\left\lfloor \frac{u_i}{v_i} \right\rfloor + \alpha + 1 \right) \mathbf{v} = \mathbf{u} - \left(\left\lfloor \frac{u_i}{v_i} \right\rfloor + 1 \right) \mathbf{v}.$$

And, when the norms are coincident, we observe:

$$\frac{h_i}{v_i} \in \mathbb{Z} \iff \frac{u_i}{v_i} \in \mathbb{Z}.$$

3. It is straightforward from previous item.

4. - When $\frac{u_i}{v_i} \in \mathbb{Z}$, it must be

$$\text{Reduce}_{\mathbf{v}}(\mathbf{u}) = \mathbf{u} - \frac{u_i}{v_i}\mathbf{v},$$

$$\text{Reduce}_{-\mathbf{v}}(\mathbf{u}) = \mathbf{u} - \frac{u_i}{-v_i}(-\mathbf{v}) = \mathbf{u} - \frac{u_i}{v_i}\mathbf{v}.$$

- In any other case, we show

$$\left\lfloor \frac{u_i}{-v_i} \right\rfloor = -\left\lceil \frac{u_i}{v_i} \right\rceil,$$

and $\left\lfloor \frac{u_i}{v_i} \right\rfloor + 1 = \left\lceil \frac{u_i}{v_i} \right\rceil$. So,

$$\text{Reduce}_{\mathbf{v}}(\mathbf{u}) \in \left\{ \mathbf{u} - \left\lfloor \frac{u_i}{v_i} \right\rfloor \mathbf{v}, \mathbf{u} - \left(\left\lfloor \frac{u_i}{v_i} \right\rfloor + 1 \right) \mathbf{v} \right\},$$

$$\text{Reduce}_{-\mathbf{v}}(\mathbf{u}) \in \left\{ \mathbf{u} + \left\lceil \frac{u_i}{v_i} \right\rceil (-\mathbf{v}), \mathbf{u} + \left(1 - \left\lceil \frac{u_i}{v_i} \right\rceil \right) \mathbf{v} \right\}.$$

And the algorithm takes the same output in both cases.

■

The properties 2, 3 and 4 in the above lemma show that $\text{Reduce}_{\mathbf{v}}(\mathbf{u})$ is invariant in the set $\mathbf{u} + \mathbb{Z}\mathbf{v}$. In order to gauge the norm reduction performed by `REDUCE` procedure, the next result provides the following bound:

Proposition 3.16 *Let $\mathbf{u}, \mathbf{v} \in \mathbb{R}^2$ be two vectors such that $\mathbf{v} \neq \mathbf{0}$. Let $i \in \{1,2\}$, $j \in \{1,2\}\backslash\{i\}$ as in Algorithm 3.2; with $|v_i| = \beta|v_j|$ for some $1 \leq \beta \in \mathbb{R}$. If $|v_i| \leq |u_i|$ and $|u_j| \neq 0$, then:*

$$\|\text{Reduce}_{\mathbf{v}}(\mathbf{u})\| \leq \frac{\alpha\left(1+\frac{1}{\beta}\right)+2}{2\alpha+2}\|\mathbf{u}\|,$$

where $|u_i| = \alpha|u_j|$, for some $\alpha \in \mathbb{R}$.

Proof. Firstly, let us prove that $\exists \mathbf{h} \in \mathbf{u} + \mathbb{Z}\mathbf{v}$ with $|h_i| < \frac{|u_i|}{2}$, and $\big(h_i \neq 0 \Rightarrow \text{sgn}\,(h_i) = \text{sgn}\,(u_i)\big)$.

We just see that there are two vectors $\mathbf{r}$ and $\mathbf{s}$ in $\mathbf{u} + \mathbb{Z}\mathbf{v}$, with: $r_i = u_i - \left\lfloor \frac{u_i}{v_i} \right\rfloor v_i$, $s_i = u_i - \left\lfloor \frac{u_i}{v_i} \right\rfloor v_i - v_i$.

As $|v_i| \leq |u_i|$, it must be $|u_i| > 0$. Also, $\left|\frac{u_i}{v_i}\right| \geq 1 \Rightarrow \left\lfloor \frac{u_i}{v_i} \right\rfloor \neq 0$. We also see that $\left\lfloor \frac{u_i}{v_i} \right\rfloor = -1 \Rightarrow \frac{u_i}{v_i} = -1 \Rightarrow r_i = 0$. We suppose then $\frac{u_i}{v_i} \notin \mathbb{Z}$, $\left\lfloor \frac{u_i}{v_i} \right\rfloor \notin \{-1, 0\}$.

We divide the proof into several cases:

- If $v_i > 0$, $u_i - \left\lfloor \frac{u_i}{v_i} \right\rfloor v_i \geq 0$ and $u_i - \left\lfloor \frac{u_i}{v_i} \right\rfloor v_i - v_i < 0$.
  - If $u_i > 0$, then $\left\lfloor \frac{u_i}{v_i} \right\rfloor \geq 1 \Rightarrow \left\lfloor \frac{u_i}{v_i} \right\rfloor < 2\left\lfloor \frac{u_i}{v_i} \right\rfloor$.

    $$\left\lfloor \frac{u_i}{v_i} \right\rfloor < 2\left\lfloor \frac{u_i}{v_i} \right\rfloor \Rightarrow \frac{u_i}{v_i} < 2\left\lfloor \frac{u_i}{v_i} \right\rfloor \Rightarrow \frac{u_i}{v_i} - \frac{u_i}{2v_i} < \left\lfloor \frac{u_i}{v_i} \right\rfloor \Rightarrow u_i - \left\lfloor \frac{u_i}{v_i} \right\rfloor v_i < \frac{u_i}{2}$$

  - If $u_i < 0$, then $\left\lfloor \frac{u_i}{v_i} \right\rfloor \leq -2 \Rightarrow \left\lfloor \frac{u_i}{v_i} \right\rfloor \geq 2 + 2\left\lfloor \frac{u_i}{v_i} \right\rfloor$.

    $$\left\lfloor \frac{u_i}{v_i} \right\rfloor \geq 2\left\lfloor \frac{u_i}{v_i} \right\rfloor + 2 \Rightarrow 2\left\lfloor \frac{u_i}{v_i} \right\rfloor + 2 < \frac{u_i}{v_i} \Rightarrow \frac{u_i}{2} > \left\lfloor \frac{u_i}{v_i} \right\rfloor v_i + v_i \Rightarrow$$
    $$\Rightarrow u_i - \left\lfloor \frac{u_i}{v_i} \right\rfloor v_i - v_i > \frac{u_i}{2}.$$

- If $v_i < 0$, $u_i - \left\lfloor \frac{u_i}{v_i} \right\rfloor v_i - v_i > 0$ and $u_i - \left\lfloor \frac{u_i}{v_i} \right\rfloor v_i \leq 0$.
  - If $u_i > 0$, then $\left\lfloor \frac{u_i}{v_i} \right\rfloor \leq -2$

    $$\left\lfloor \frac{u_i}{v_i} \right\rfloor \geq 2\left\lfloor \frac{u_i}{v_i} \right\rfloor + 2 \Rightarrow 2\left\lfloor \frac{u_i}{v_i} \right\rfloor + 2 < \frac{u_i}{v_i} \Rightarrow \frac{u_i}{v_i} - \left\lfloor \frac{u_i}{v_i} \right\rfloor - 1 > \frac{u_i}{2v_i} \Rightarrow$$
    $$\Rightarrow u_i - \left\lfloor \frac{u_i}{v_i} \right\rfloor v_i - v_i < \frac{u_i}{2}.$$

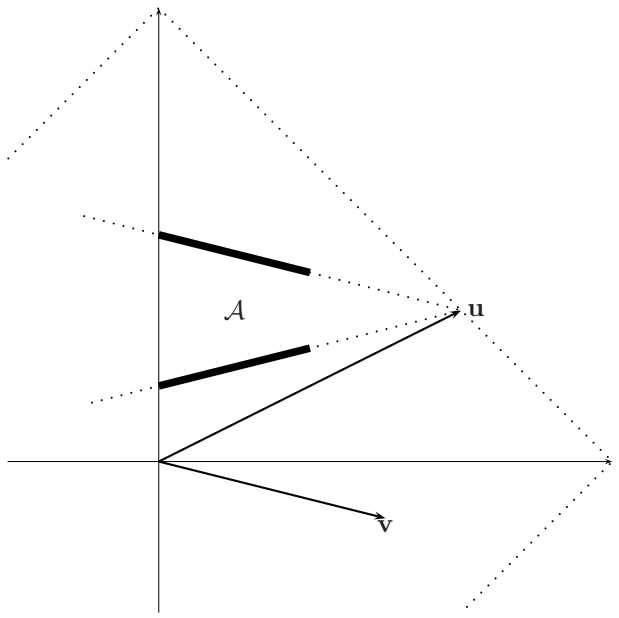


Figure 3.10: The set $\mathcal{A}$

- If $u_i < 0$, then $\left\lfloor \frac{u_i}{v_i} \right\rfloor \geq 1$. So,

$$\frac{u_i}{v_i} < 2\left\lfloor \frac{u_i}{v_i} \right\rfloor \Rightarrow \frac{u_i}{v_i} - \left\lfloor \frac{u_i}{v_i} \right\rfloor < \frac{u_i}{2v_i} \Rightarrow u_i - \left\lfloor \frac{u_i}{v_i} \right\rfloor v_i > \frac{u_i}{2}.$$

In the rest of the proof, we use the following notation:

$$[a,b] := \begin{cases} (a,b), \text{if } i = 1 \\ (b,a), \text{if } i = 2 \end{cases}$$

We define the following set:

$$\left\{ \left[h_i, \frac{v_j}{v_i}(h_i - u_i) + u_j \right] \mid 0 \leq |h_i| < \frac{|u_i|}{2},\ u_i h_i \geq 0 \right\} \subseteq$$

$$\subseteq \left\{ \left[h_i, \pm\frac{|v_j|}{|v_i|}(h_i - u_i) + u_j \right] \mid 0 \leq h_i \leq \frac{|u_i|}{2},\ u_i h_i \geq 0 \right\} =: \mathcal{A}$$

As seen before, $\|\text{Reduce}_{\mathbf{v}}(\mathbf{u})\| \leq \max_{\mathbf{w} \in \mathcal{A}} \|\mathbf{w}\|$. Now, this maximum is reached at:

$$\left[\frac{u_i}{2}, u_j + \frac{\text{sgn}\,(u_j)\,|u_i|}{2\beta} \right].$$

Then, $\|\text{Reduce}_{\mathbf{v}}(\mathbf{u})\| \leq \frac{|u_i|}{2} + |u_j| + \frac{|u_i|}{2\beta}$. And so,

$$\frac{\|\text{Reduce}_{\mathbf{v}}(\mathbf{u})\|}{\|\mathbf{u}\|} \leq \frac{|u_i|\left(\frac{1+\beta}{2\beta}\right) + |u_j|}{|u_1| + |u_2|} = \frac{\alpha|u_j|\left(\frac{1+\beta}{2\beta}\right) + |u_j|}{(1+\alpha)|u_j|} = \frac{2+\alpha\left(1+\frac{1}{\beta}\right)}{2+2\alpha}.$$

■

We note that:

$$\frac{\alpha\left(1+\frac{1}{\beta}\right)+2}{2+2\alpha} \leq 1,$$

because $\beta \geq 1$. And if $\beta > 1$ then the inequality is strict. The requirement $|v_i| = \beta|v_j|$ excludes from this result the reduction by a vector with one vanished component. We are also excluding the possibility $|u_j| = 0$. In the next result, we are dealing with these cases. In a similar way we can obtain the following result:

Proposition 3.17 *Let $\mathbf{u}, \mathbf{v} \in \mathbb{R}^2$ be two vectors such that $\mathbf{v} \neq 0$. Let $i \in \{1,2\}$, $j \in \{1,2\}\backslash\{i\}$ as Algorithm 3.2 with $|v_i| \leq |u_i|$. We have:*

1. *If $v_j = 0 \wedge u_j \neq 0$, then:*

$$\|\text{Reduce}_{\mathbf{v}}(\mathbf{u})\| \leq \frac{2+\alpha}{2+2\alpha}\|\mathbf{u}\|, \text{ where } \alpha = |u_i|/|u_j|.$$

2. *If $u_j = 0 \wedge v_j \neq 0$, then:*

$$\|\text{Reduce}_{\mathbf{v}}(\mathbf{u})\| \leq \frac{1+\beta}{2\beta}\|\mathbf{u}\|, \text{ where } \beta = |v_i|/|v_j|.$$

3. *If $|v_j| = |u_j| = 0$, then:*

$$\|\text{Reduce}_{\mathbf{v}}(\mathbf{u})\| \leq \frac{1}{2}\|\mathbf{u}\|.$$

3.4.2 The method's core

Let us describe precisely our goal. We start with three vectors in $\mathbb{R}^2$, two of them linearly independent:

$$\mathbf{w} + \mathbb{Z} < \mathbf{u}, \mathbf{v} >, \ \text{rank}(\mathbf{u}, \mathbf{v}) = 2.$$

We are going to find the shortest element in that set with respect to ℓ_1 norm. The method consists on recursively apply `REDUCE` algorithm to the "translation" vector $\mathbf{w}$ by some vectors in $\mathbb{Z} < \mathbf{u}, \mathbf{v} >$. Our purpose is to guarantee that each step reduces the vector norm by a constant factor, until we reach an ending condition. In order to perform this goal, we select a particular basis of the lattice $\mathbb{Z} < \mathbf{u}, \mathbf{v} >$:

Definition 3.18 *A lattice basis $\{\mathbf{u}, \mathbf{v}\}$ is called extra-reduced when:*

$$\text{Reduce}_{\mathbf{v}}(\mathbf{u}) = \mathbf{u} \ \wedge \text{Reduce}_{\mathbf{u}}(\mathbf{v}) = \mathbf{v}.$$

In order to study some properties of this kind of lattice basis, we introduce a new concept:

Definition 3.19 *We classify vectors* $\mathbf{u} = (u_1, u_2) \in \mathbb{R}^2$ *into two types:*

- **horizontal**, *if* $|u_1| \geq |u_2|$.
- **vertical**, *if* $|u_1| < |u_2|$.

Lemma 3.20 *Let* $\{\mathbf{u}, \mathbf{v}\}$ *be an extra-reduced basis. Then,*

1. *The basis is also Gauss-reduced.*
2. *One of the basis vectors is vertical, and the other is horizontal.*

Proof. The proof of the first claim is straightforward. For the second, we first suppose both basis vectors are vertical, that is,

$$\mathbf{u} = (u_1, u_2),\ |u_1| < |u_2| \text{ and } \mathbf{v} = (v_1, v_2),\ |v_1| < |v_2|.$$

We look at the vectors:

$$\mathbf{u} + \mathbf{v} = (u_1 + v_1, u_2 + v_2), \quad \mathbf{u} - \mathbf{v} = (u_1 - v_1, u_2 - v_2).$$

We select the one that minimizes the second component modulus, whose norm is:

$$||u_1| \pm |v_1|| + ||u_2| - |v_2|| \leq \begin{cases} & |u_1| + |v_1| + |u_2| - |v_2| < \|\mathbf{u}\| \\ \vee & \\ & |u_1| + |v_1| + |v_2| - |u_2| < \|\mathbf{v}\| \end{cases}$$

It is clear, in both options, that the basis is Gauss-reduced.

We suppose now that both vectors are horizontal, that is,

$$\mathbf{u} = (u_1, u_2),\ |u_1| \geq |u_2|$$

$$\mathbf{v} = (v_1, v_2),\ |v_1| \geq |v_2|.$$

Using the first claim of Lemma 3.15, we have that:

$$\mathbf{u} = \text{Reduce}_{\mathbf{v}}(\mathbf{u}) \Rightarrow |u_1| < |v_1|$$

$$\mathbf{v} = \text{Reduce}_{\mathbf{u}}(\mathbf{v}) \Rightarrow |v_1| < |u_1|$$

■

Note 3.21 *As a consequence of the above proof, we can classify Gauss-reduced basis, as well as they consist on:*

- *A vertical vector and an horizontal one.*
- *Two horizontal vectors.*

Algorithm 3.3: Extra-reduced basis algorithm

Input: $\mathbf{u}, \mathbf{v} \in \mathbb{R}^2$, Gauss-reduced basis of a lattice.
Output: $\mathbf{U}, \mathbf{V} \in \mathbb{R}^2$, extra-reduced basis of the same lattice.

```
Set the vectors U, V so that {U, V} = {u, v} and ‖U‖ ≤ ‖V‖;
if ‖U‖ < ‖V‖ then
    V ← Reduce_U(V)
else
    if U and V are horizontal then
        if |U_1| ≠ |U_2| then
            Swap U and V;
        end
        V ← Reduce_U(V);
    else
        if U is vertical then
            Swap U and V;
        end
    end
    if U_2 < 0 then
        U ← −U;
    end
    if V_1 < 0 then
        V ← −V;
    end
end
```

In the second option, one of them (at least) must be diagonal ($|u_1| = |u_2|$), because in other case, the same argument as in the first proof's part would reach a contradiction.

Kaib and Schnorr [1996] showed how to get a reduced basis (referred to any norm, in particular ℓ_1) of a lattice with rank two. Then, it is easy to reach an extra-reduced basis, by just applying `REDUCE` procedure, see Algorithm 3.3.

Lemma 3.22 *Algorithm 3.3 is correct.*

Proof.

- If both input vectors' norms are different, the algorithm ends at Step 3. If $\mathbf{U}$ and $\mathbf{V}$ are the input vectors, ordered such that $\|\mathbf{U}\| < \|\mathbf{V}\|$, we output:

$$\{\mathbf{U}, \text{Reduce}_{\mathbf{U}}(\mathbf{V})\}.$$

  This output is a basis of the input lattice, because the change matrix takes the form:

$$\begin{pmatrix} 1 & \lambda \\ 0 & 1 \end{pmatrix},$$

  for an integer λ. Furthermore, it is an extra-reduced basis: firstly, by Lemma 3.15.3:

$$\text{Reduce}_{\mathbf{U}}(\text{Reduce}_{\mathbf{U}}(\mathbf{V})) = \text{Reduce}_{\mathbf{U}}(\mathbf{V}).$$

  On the other hand, $\text{Reduce}_{\text{Reduce}_{\mathbf{U}}(\mathbf{V})}(\mathbf{U})$ is a vector in the lattice, and since $\{\mathbf{U}, \mathbf{V}\}$ is an ordered Gauss-reduced basis, it must be:

$$\|\mathbf{U}\| \leq \|\text{Reduce}_{\text{Reduce}_{\mathbf{U}}(\mathbf{V})}(\mathbf{U})\|.$$

  The converse comes from the meaning of `REDUCE` process, so

$$\|\mathbf{U}\| = \|\text{Reduce}_{\text{Reduce}_{\mathbf{U}}(\mathbf{V})}(\mathbf{U})\|.$$

  Now, since $\|\mathbf{V}\|$ is the second Minkowski's minimum in the lattice, and its norm is greater than $\|\mathbf{U}\|$, it must be $\text{Reduce}_{\text{Reduce}_{\mathbf{U}}(\mathbf{V})}(\mathbf{U}) \in \mathbb{Z} < \mathbf{U} >$, and so,

$$\mathbf{U} = \text{Reduce}_{\text{Reduce}_{\mathbf{U}}(\mathbf{V})}(\mathbf{U}).$$

- We analyse now the case $\|\mathbf{u}\| = \|\mathbf{v}\|$.

  According to Note 3.21, after Step 8, we have $|U_1| = |U_2|$. Now, if we perform Step 9, we reach a basis where $\mathbf{U}$ is horizontal, and $\mathbf{V}$ is vertical:

$$|V_1| + |V_2| = \|\mathbf{V}\| = \|\mathbf{U}\| = 2|U_1| > 2|V_1|.$$

  Last inequality follows from Lemma 3.15.1.

  Then, $|V_2| > |V_1|$.

  Even if we have skipped Steps 6–9, after 14, $\mathbf{U}$ is horizontal, and $\mathbf{V}$ is vertical.

We have to see now that after Step 20 we have an extra-reduced basis. The situation is as follows:

$$0 \leq U_2 \leq |U_1|.$$
$$0 \leq V_1 < |V_2|.$$

We have that $|U_2| < |V_2|$ and that $|V_1| < |U_1|$, because $\|\mathbf{U}\| = \|\mathbf{V}\|$. Then,

$$\left\lfloor \frac{U_2}{V_2} \right\rfloor, \left\lfloor \frac{V_1}{U_1} \right\rfloor \in \{-1, 0\},$$

and then the basis $\{\mathbf{U}, \mathbf{V}\}$ is extra-reduced.

■

We will use then this extra-reduced basis to perform iterative reductions of the vector $\mathbf{w}$, as in explained in Algorithm 3.4.

Algorithm 3.4: Iterative Reduce

Input: $\mathbf{w}, \mathbf{u}, \mathbf{v} \in \mathbb{R}^2$; $\{\mathbf{u}, \mathbf{v}\}$, extra − reduced basis ($\mathbf{u}$, hor., $\mathbf{v}$, ver.).
Output: $\mathbf{W} \in \mathbf{w} + \mathbb{Z} < \mathbf{u}, \mathbf{v} >$, $|W_1| < |u_1|$, $|W_2| < |v_2|$.

```
1 W ← w;
2 while |W_1| ≥ |u_1| ∨ |W_2| ≥ |v_2| do
3 |   if |W_1| ≥ |u_1| then
4 |   |   W ← Reduce_u(W);
5 |   else
6 |   |   W ← Reduce_v(W);
7 |   end
8 end
```

From Lemma 3.15, Proposition 3.16 and Proposition 3.17 we can obtain the following:

Lemma 3.23 *Algorithm 3.4 is correct and the number of loops is in* $O(\log \|\mathbf{w}\|)$.

Proof. By Lemma 3.15.1, in consecutive loops different substeps must be performed. So, each two loops, one reduction by $\mathbf{v}$ is made. But by Propositions 3.16 and 3.17, there exists a constant $\lambda < 1$ such that every step of this kind produces:

$$\|\mathbf{W}^{new}\| \leq \lambda \|\mathbf{W}^{old}\|.$$

That constant λ is the minimum reduction factor provided by Proposition 3.16 or Proposition 3.17. Note that the parameter α cannot the arbitrary small: suppose that $\mathbf{W}^n$ is a vector obtained at step n, and $|W_1| < |u_1|$. Then, the parameter α corresponding to vector $\mathbf{W}$ at step $n + 2k$ satisfies:

$$\alpha \geq \frac{|v_i|}{\|\mathbf{W}^n\| - |v_i|}.$$

So, as a lattice is a discrete subset, the lemma follows. ■

3.4.3 The whole process

We have reached a vector in $\mathbf{w}+\mathbb{Z} < \mathbf{u}, \mathbf{v} >$ with some properties. We need to conclude our job by getting the shortest vector among all. We will use the following technical result:

Lemma 3.24 *Let $\{\mathbf{u}_1, \mathbf{u}_2\}$ be a reduced basis (respect to any norm) of a lattice in $\mathbb{R}^m$. Let $\mathbf{w} = \alpha_1\mathbf{u}_1 + \alpha_2\mathbf{u}_2$ be a lattice vector $(\alpha_1, \alpha_2 \in \mathbb{Z})$. Then, we have:*

$$\|\alpha_1\mathbf{u}_1\|, \|\alpha_2\mathbf{u}_2\| \leq 2\|\mathbf{w}\|.$$

Proof.

- If $\mathbf{w} = 0$, the result is clear.
- If $\alpha_1\alpha_2 = 0$, we have:
$$\|\mathbf{w}\| = \left\{ \begin{array}{c} \\ \vee \\ \\ \end{array} \begin{array}{l} \|\alpha_1\mathbf{u}_1\| \\ \\ \|\alpha_2\mathbf{v}_2\| \end{array} \right.$$
- If $\alpha_1\alpha_2 \neq 0$, let us write $\sigma \in \Sigma_2$, such that
$$|\alpha_{\sigma(1)}| \leq |\alpha_{\sigma(2)}|.$$
Then, writing $\varepsilon := \operatorname{sgn}\left(\dfrac{\alpha_{\sigma(1)}}{\alpha_{\sigma(2)}}\right) \in \{-1, 1\}$,
$$\|\mathbf{u}_{\sigma(1)}\| \leq \|\mathbf{u}_{\sigma(1)} + \varepsilon\mathbf{u}_{\sigma(2)}\| \Rightarrow \|\mathbf{u}_{\sigma(1)}\| \leq \|\mathbf{u}_{\sigma(1)} + \frac{\alpha_{\sigma(2)}}{\alpha_{\sigma(1)}}\mathbf{u}_{\sigma(2)}\| \Rightarrow$$
$$\Rightarrow \quad \|\alpha_{\sigma(1)}\mathbf{u}_{\sigma(1)}\| \leq \|\mathbf{v}\|.$$
Then,
$$\|\alpha_{\sigma(2)}\mathbf{u}_{\sigma(2)}\| \leq \|\mathbf{v}\| + \|\alpha_{\sigma(1)}\mathbf{u}_{\sigma(1)}\|.$$

■

Let us suppose now we have reached a description $\mathbf{w} + \mathbb{Z} < \mathbf{u}, \mathbf{v} >$ of the original set, where $\{\mathbf{u}, \mathbf{v}\}$ is an extra-reduced basis ($\mathbf{u}$ is horizontal and $\mathbf{v}$ is vertical), and with $|w_1| < |u_1|$, $|w_2| < |v_2|$. Let $\mathbf{W}$ be a closest vector to $\mathbf{w}$ with respect to ℓ_1 norm.

Then, $\mathbf{d} := \mathbf{W} - \mathbf{w} \in \mathbb{Z} < \mathbf{u}, \mathbf{v} >$ and

$$\|\mathbf{d}\| \leq 2\|\mathbf{w}\| = 2(|w_1| + |w_2|) < 2(|u_1| + |v_2|).$$

We try to bound the coefficients of $\mathbf{d}$ as a lattice member:

$$\mathbf{d} = \alpha\mathbf{u} + \beta\mathbf{v}.$$

We distinguish two cases and applying previous Lemma 3.24

- $|u_1| \geq |v_2|$

$$\|\alpha\mathbf{u}\| \leq 2\|\mathbf{d}\| \Rightarrow |\alpha| \leq \frac{4(|u_1| + |v_2|)}{|u_1| + |u_2|} \leq 8$$

- $|u_1| < |v_2|$

$$\|\beta\mathbf{v}\| \leq 2\|\mathbf{d}\| \Rightarrow |\beta| \leq \frac{4(|u_1| + |v_2|)}{|v_1| + |v_2|} \leq 8$$

So, we can now give a method to obtain a shortest vector: Algorithm 3.5.

Algorithm 3.5: Final reduce

Input: $\mathbf{u}, \mathbf{v}$, extra-reduced basis ($\mathbf{u}$, hor. $\mathbf{v}$, ver.), $\mathbf{w}$, with $|w_1| < |u_1|,\ |w_2| < |v_2|$

Output: $\mathbf{W}$, shortest vector in $\mathbf{w} + Z < \mathbf{u}, \mathbf{v} >$.

```
1 U ← u, V ← v;
2 if |U_1| < |V_2| then
3 |   Swap U and V;
4 end
5 for α = [−8, ..., 8] do
6 |   W_α ← Reduce_V(w + αU);
7 end
8 Return a vector with minimum norm in {W_α | |α| ≤ 8};
```

To sum up, jointing Algorithms 3.3, 3.4 and 3.5, we reach our goal.

3.4.4 Complexity

When studying lattices from a complexity point of view, it is customary to assume that the basis vectors (and therefore any lattice vector) have rational coordinates. It is easy to see that rational lattices can be converted to integer lattices (i.e., sublattices of $\mathbb{Z}^m$) by multiplying all coordinates by an appropriate integer scaling factor.

Definition 3.25 *If a, b are two integers such that $b \neq 0$, we denote by* $\mathrm{quo}(a, b)$, $\mathrm{rem}(a, b)$ *the unique integers satisfying:*

$$a = b \cdot \mathrm{quo}(a, b) + \mathrm{rem}(a, b),\ 0 \leq \mathrm{rem}(a, b) < |b|,$$

i.e., $\mathrm{quo}(a, b)$, $\mathrm{rem}(a, b)$ *are the quotient and the remainder of the Euclidean division of a by b.*

In the case of lattices with integer coefficients, Algorithm 3.2 can be substituted with Algorithm 3.6.

It is straightforward check both Algorithm 3.2 and Algorithm 3.6 have same output for integer vectors.

Given three vectors $\{\mathbf{u}, \mathbf{v}, \mathbf{w}\}$, such that $\mathbf{u}, \mathbf{v}$ are linearly independent. Let $M = \max(\|\mathbf{u}\|, \|\mathbf{v}\|, \|\mathbf{w}\|)$. We claim that our algorithm for solving the closest vector problem costs $O(\log^2 M \log\log M \log\log\log M)$ bit operations:

Algorithm 3.6: Integer reduce

Input: $\mathbf{u}, \mathbf{v} \in \mathbb{Z}^2,\ \mathbf{v} \neq \mathbf{0}$.

Output: $\text{Reduce}_{\mathbf{v}}(\mathbf{u}) \in \mathbf{u} + \mathbb{Z}\mathbf{v} \mid \|\text{Reduce}_{\mathbf{v}}(\mathbf{u})\| = \min\{\|\mathbf{u} + \alpha\mathbf{v}\| \mid \alpha \in \mathbb{Z}\}$.

1 Find $i \in \{1, 2\},\ j \in \{1, 2\}\backslash\{i\}$ such that $|v_i| > |v_j|$. If $|v_1| = |v_2|$, select $i \leftarrow 1$;

2 Return the vector with minimum norm between:

$$\mathbf{u} - \text{quo}(u_i, v_i)\mathbf{v},\ \mathbf{u} - (\text{quo}(u_i, v_i) + \text{sgn}\,(v_i))\mathbf{v}.$$

If both share the same norm, return $\mathbf{u} - \text{quo}(u_i, v_i)\mathbf{v}$.

- Clearly, $\text{Reduce}_{\mathbf{v}}(\mathbf{u}) \in \mathbf{u} + \mathbb{Z}\mathbf{v}$ is bounded by computing four Euclidean division on integer number of size $O(\log M)$.
- The algorithm in [Kaib and Schnorr, 1996] computes a Gauss-reduced basis from a base $\{\mathbf{u}, \mathbf{v}\}$ of a lattice $\mathcal{L} \subset \mathbb{Z}^2$ in $O(\log M)$ arithmetic operations on integer numbers of size $O(\log M)$.
- Now, find an extra-reduced basis from a Gauss-reduced one using Algorithm 3.3 requires compute four Euclidean division on integer of size $O(\log M)$.
- Algorithm 3.4 requires $O(\log M)$ arithmetic (see Lemma 3.23) steps and any step consists on two Euclidean division on integer of size $O(\log M)$.
- Finally, Algorithm 3.5 computes 16 `REDUCE` procedures.

To finish this analysis we use the famous algorithm of Schönhage and Strassen [1971] for multiplication integers.

From practical point of view, it is interesting to remark that the Algorithm 3.5 only requires to perform the `REDUCE` procedure 8 times instead of 16. However, the proof of this fact is a little longer.

Given a vertex $j \in \mathbb{Z}_N$, we can compute by Extended Euclidean algorithm a path $\mathbf{w}$ from 0 to j. All of this on $O(\log N \log\log N \log\log\log N)$, see for instance [von zur Gathen and Gerhard, 2003].

Corollary 3.26 *Given an undirected circulant graph $C_N(\pm j_1, \pm j_2)$ and vertex $j \in \mathbb{Z}_N$, we can decide if there exists a path from vertex 0 to vertex j and, in the affirmative case, we can compute a shortest one on $O(\log^2 N \log\log N \log\log\log N)$ bit operations.*

3.4.5 The directed case

Our method can be easily extended to *directed circulant graphs* $C_N(j_1, j_2)$. First, we describe the tool of *positive reduction of a vector*. Given two vectors $\mathbf{u} = (u_1, u_2)$ and $\mathbf{v} = (v_1, v_2) \neq \mathbf{0}$, we are looking for $\alpha \in \mathbb{Z}$ such that the value $\|\mathbf{u} + \alpha\mathbf{v}\|$ is minimal and having both components positive. In general, such vector does not exist, for instance, take $\mathbf{u} = (3, -5)$ and $\mathbf{v} = (12, 0)$. If it exists, we denote this vector by $\text{PRed}_{\mathbf{v}}(\mathbf{u}) \in \mathbf{u} + \mathbb{Z}\mathbf{v}$, that is,

$$\|\text{PRed}_{\mathbf{v}}(\mathbf{u})\| = \min\{\|\mathbf{u} + \alpha\mathbf{v}\| :\ \alpha \in \mathbb{Z},\ \mathbf{u} + \alpha\mathbf{v} \in \mathbb{R}^2_{\geq 0}\},$$

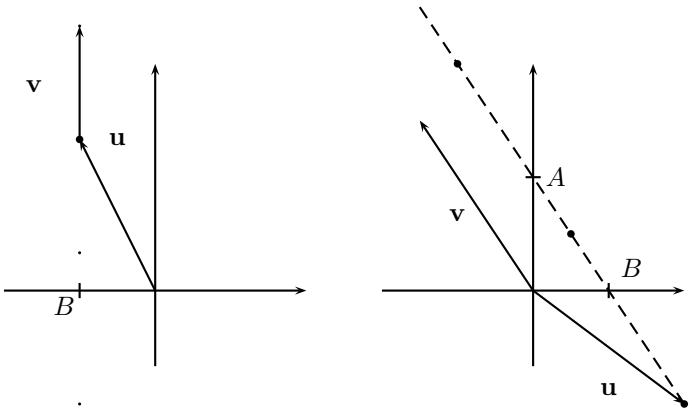


Figure 3.11: Algorithm 3.7

where $\mathbb{R}^2_{\geq 0}$ is the set of real points (x_1, x_2) with $x_i \geq 0, i = 1, 2$.

It is easy to check that the Algorithm 3.7 is correct. On the other hand, if we are dealing with integer vectors, we can formulate the main steps, similar as in Algorithm 3.6, using the Euclidean quotient (see Algorithm 3.8).

Therefore, a bound for the bit complexity on computing $\mathrm{PRed}_{\mathbf{v}}(\mathbf{u})$ is

$$O(\log^2 M \log\log M \log\log\log M),$$

where $M = \max(\|\mathbf{u}\|, \|\mathbf{v}\|)$.

Once this tool is fixed, let us describe the method to reach a shortest path in a directed circulant graph. Firstly, we act as seen in the previous section to compute an extra reduced basis of the associated lattice $\{\mathbf{u}, \mathbf{v}\}$, and a shortest path for the corresponding undirected circulant graph $\mathbf{w}$.

We can state that there always exists one path in the directed circulant graph with bounded length.

Lemma 3.27 *Let $\mathbf{w}, \mathbf{u}, \mathbf{v} \in \mathbb{Z}^2$, such that $\{\mathbf{u}, \mathbf{v}\}$ is an extra reduced basis for the lattice they generate. Then,*

$$\exists \mathbf{d} \in (\mathbf{w} + \mathbb{Z} < \mathbf{u}, \mathbf{v} >) \cap \mathbb{N}^2,\ \|\mathbf{d}\| \leq 6 \max\{\|\mathbf{u}\|, \|\mathbf{v}\|\}.$$

Proof: Let $M = \max\{\|\mathbf{u}\|, \|\mathbf{v}\|\}$. We consider the translated lattice

$$\mathbf{w} - (2M, 2M) + \mathbb{Z} < \mathbf{u}, \mathbf{v} > .$$

By Algorithm 3.4, this set contains an element $\mathbf{z}$, with $|z_1| < |u_1|,\ |z_2| < |v_2|$. So, $\|\mathbf{z}\| \leq 2M$. Clearly $\mathbf{z} + (2M, 2M)$ belongs to the set $(\mathbf{w} + \mathbb{Z} < \mathbf{u}, \mathbf{v} >) \cap \mathbb{N}^2$, and its norm is bounded by $6M$. ■

Finally, we follow a similar argument than in Section 3 to reach the shortest path for the directed graph.

Algorithm 3.7: Postive reduction

Input: $\mathbf{u}, \mathbf{v} \in \mathbb{R}^2,\ \mathbf{v} \neq \mathbf{0}$.

Output: $\mathrm{PRed}_{\mathbf{v}}(\mathbf{u})$, if it exists, and $\emptyset$ otherwise.

Select $i \in \{1,2\},\ j \in \{1,2\}\backslash\{i\}$ such that $|v_i| > |v_j|$. If $|v_1| = |v_2|$, then $i = 1$;

Set $\varepsilon = \begin{cases} -1, & \text{if } |v_1| \geq |v_2| \\ 1\ , & \text{if } |v_1| < |v_2| \end{cases}$;

Compute $\Delta \leftarrow \varepsilon(u_1 v_2 - u_2 v_1)$;

if $v_j = 0$ **then**

  $B \leftarrow \dfrac{\Delta}{v_i}$;

  if $B < 0$ **then**

    Output $\emptyset$;

  else

    Output $\mathbf{u} - \mathrm{sgn}\,(v_i) \left\lfloor \dfrac{u_i}{|v_i|} \right\rfloor \mathbf{v}$;

  end

else

  $A \leftarrow \dfrac{-\Delta}{v_j},\ B \leftarrow \dfrac{\Delta}{v_i}$;

  if $(A < 0 \wedge B < 0)$ **then**

    Output $\emptyset$;

  end

  if $(A < 0 \wedge B \geq 0)$ **then**

    Output $\mathbf{u} - \mathrm{sgn}\,(v_i) \left\lfloor \dfrac{u_i}{|v_i|} \right\rfloor \mathbf{v}$;

  end

  if $(A \geq 0 \wedge B < 0)$ **then**

    Output $\mathbf{u} - \mathrm{sgn}\,(v_j) \left\lfloor \dfrac{u_j}{|v_j|} \right\rfloor \mathbf{v}$;

  end

  if $(A \geq 0 \wedge B \geq 0)$ **then**

    Set $\mathbf{w} \leftarrow \mathbf{u} - \mathrm{sgn}\,(v_i) \left\lfloor \dfrac{u_i}{|v_i|} \right\rfloor \mathbf{v}$;

    if $\mathbf{w} \in \mathbb{R}^2_{\geq 0}$ **then**

      Output $\mathbf{w}$;

    else

      Output $\emptyset$;

    end

  end

end

Algorithm 3.8: Positive reduction with integer components

Input: $\mathbf{u}, \mathbf{v} \in \mathbb{Z}^2,\ \mathbf{v} \neq \mathbf{0}$.

Output: $\mathrm{PRed}_{\mathbf{v}}(\mathbf{u})$, if it exists, and $\emptyset$ otherwise.

Select $i \in \{1,2\},\ j \in \{1,2\}\backslash\{i\}$ such that $|v_i| > |v_j|$. If $|v_1| = |v_2|$, then $i = 1$;

Set $\varepsilon = \begin{cases} -1, & \text{if } |v_1| \geq |v_2| \\ 1\ , & \text{if } |v_1| < |v_2| \end{cases}$;

Compute $\Delta \leftarrow \varepsilon \mathrm{sgn}\,(u_1 v_2 - u_2 v_1)$;

if $v_j = 0$ **then**

  $B \leftarrow \Delta \mathrm{sgn}\,(v_i)$;

  if $B = -1$ **then**

    Output $\emptyset$;

  else

    Output $\mathbf{u} - \mathrm{sgn}\,(v_i)\,\mathrm{quo}(u_i, |v_i|)\mathbf{v}$;

  end

else

  $A \leftarrow -\Delta \mathrm{sgn}\,(v_j)\,,\ B \leftarrow \Delta \mathrm{sgn}\,(v_i)$;

  if $(A = -1 \wedge B = -1)$ **then**

    Output $\emptyset$;

  end

  if $(A = -1 \wedge B \geq 0)$ **then**

    Output $\mathbf{u} - \mathrm{sgn}\,(v_i)\,\mathrm{quo}(u_i, |v_i|)\mathbf{v}$;

  end

  if $(A \geq 0 \wedge B = -1)$ **then**

    Output $\mathbf{u} - \mathrm{sgn}\,(v_j)\,\mathrm{quo}(u_j, |v_j|)\mathbf{v}$;

  end

  if $(A \geq 0 \wedge B \geq 0)$ **then**

    Set $\mathbf{w} \leftarrow \mathbf{u} - \mathrm{sgn}\,(v_i)\,\mathrm{quo}(u_i, |v_i|)\mathbf{v}$;

    if $\mathbf{w} \in \mathbb{N}^2_{\geq 0}$ **then**

      Output $\mathbf{w}$;

    else

      Output $\emptyset$;

    end

  end

end

Lemma 3.28 *Let* $\mathbf{w}, \mathbf{u}, \mathbf{v} \in \mathbb{Z}^2$*, such that* $\{\mathbf{u}, \mathbf{v}\}$ *is an extra reduced basis for the lattice they generate. Let* $\mathbf{W}$ *be a shortest element in a translated lattice* $\mathbf{w} + \mathbb{Z} < \mathbf{u}, \mathbf{v} >$ *and let* $\mathbf{d}$ *be a shortest element in* $(\mathbf{w} + \mathbb{Z} < \mathbf{u}, \mathbf{v} >) \cap \mathbb{N}^2$*. We have:* $\mathbf{d} - \mathbf{W} = \alpha\mathbf{u} + \beta\mathbf{v}$*, verifying*

$$|\alpha| \leq 16 \text{ if } \|\mathbf{u}\| \geq \|\mathbf{v}\|, \quad |\beta| \leq 16 \text{ if } \|\mathbf{v}\| \geq \|\mathbf{u}\|.$$

Proof: Let $M = \max\{\|\mathbf{u}\|, \|\mathbf{v}\|\}$, by Lemma 3.24 and Lemma 3.27 and since $\{\mathbf{u}, \mathbf{v}\}$ is an extra reduced basis, we have:

$$\|\alpha\mathbf{u}\|, \|\beta\mathbf{v}\| \leq 2\|\mathbf{d} - \mathbf{w}\| \leq 2\|\mathbf{d}\| + 2\|\mathbf{W}\| \leq 12M + 4M \leq 16M.$$

So, the lemma follows. ■

Using the above lemma it is straightforward to prove that the next Algorithm 3.9 computes a shortest vector with positive components.

Algorithm 3.9: Positive shortest path

Input: $\mathbf{w} \in \mathbb{Z}^2$, $\{\mathbf{u}, \mathbf{v}\}$, extra reduced basis.
Output: $\mathbf{d}$, shortest element in $(\mathbf{w} + \mathbb{Z} < \mathbf{u}, \mathbf{v} >) \cap \mathbb{N}^2$.

```
Find a shortest element z in w + Z < u, v >;
if ||u|| ≥ ||v|| then
    for α = −16, ..., 16 do
        d_α ← PRed_v(z + αu);
    end
else
    for α = −16, ..., 16 do
        d_α ← PRed_u(z + αv)
    end
end
Output min{d_α | |α| ≤ 16};
```

As an immediate consequence is that we can compute a shortest path in a directed circulant graph of degree two on $O(\log^2 N \log\log N \log\log\log N)$ bit operations.

3.4.6 Weighted circulant graphs

In this section, we consider weighted circulant graphs with two jumps and weights.

Theorem 3.29 *Given a undirected circulant graph* $C_N(\pm j_1, \pm j_2)$ *with weights* $w = (w_1, w_2)$ *we can find a shortest path on* $O(\log^2 N \log\log N \log\log\log N)$ *bit operations.*

Proof. The distance of a path $\mathbf{c} = (c_1, c_2)$ in the weighted circulant graph is

$$\|\mathbf{c}\|_w = w_1|c_1| + w_2|c_2|.$$

Let $\mathbf{c} \in \mathbb{Z}^2$ then $\|\mathbf{c}\|_w = \|\Phi(\mathbf{c})\|_{\ell_1}$ where Φ is the injective group homomorphism

$$\Phi : \mathbb{Z}^2 \to \mathbb{Z}^2, \quad \Phi((x, y)) = (w_1 x, w_2 y).$$

Let $j \in \mathbb{Z}_N$ be a vertex of the graph and let $\mathbf{u}, \mathbf{v}$ be an extra-reduced basis of the circulant graph $C_N(j_1, j_2)$. By Corollary 3.26 we compute on cubic polynomial time a shortest path $\mathbf{c}$ from vertex 0 to vertex j. Let $\mathbf{d}$ a solution to CVP for the lattice generated by $< \Phi(\mathbf{u}), \Phi(\mathbf{v}) >$, shifted by $\Phi(\mathbf{c})$, then $\Phi^{-1}(\mathbf{d})$ is a shortest path in the weighted undirected circulant graph. ■

The algorithm in the above theorem can be adapted in a natural way to weighted directed circulant graphs.

3.5 An algorithm for MDD of triple-loop computer networks

In this section we provide an algorithm specifically tailored for computing the minimal system of generators for a triple-loop computer network, which requires $O(s \log N)$ arithmetic operations, where s is the number of generators in the minimal system.

The case of degree two circulants is very simple. We always have two generators of the form x^a, y^b; and two possibilities can happen: there is another one generator $x^c y^d$ $(c < a,\ d < b)$ or those two are the only generators (irreducible ideal case). We can obtain this representation in an efficient way, for instance, using the algorithm in [Cheng and Hwang, 1988].

We present a procedure, Algorithm 3.10, to compute the minimal generators of the ideal I_S associated to a circulant digraph of degree 3. Once we have fixed a graded monomial ordering, we need as an intermediate step a procedure to decide, given a path $\mathbf{b}$ whether or not it lies in the MDD. For $\mathbf{b} \in \mathbb{N}^3$, we define the Boolean function $P(\mathbf{b})$ to be the truth value of $D(R(\mathbf{b})) = \mathbf{b}$.

Algorithm 3.10: MDD description. The three jumps case. (I)

Input: $j_1, j_2, j_3, N \in \mathbb{N},\ \gcd(j_1, j_2, j_3, N) = 1,\ P$.
Output: $\mathbf{a}_1, \ldots, \mathbf{a}_s \in \mathbb{N}^3 \mid (\mathbf{x}^{\mathbf{a}_1}, \ldots, \mathbf{x}^{\mathbf{a}_s}) = (\mathbb{N}^3 \backslash D(\mathbb{Z}_N))$; $a_i \not\leq a_j$, if $i \neq j$.

```
1  k ← 1;
2  for i = 1, 2, 3 do
3  |  m ← 0, M ← N.;
4  |  while M − m > 1 do
   |  |  l ← ⌊(m + M)/2⌋;
5  |  |
6  |  |  if P(l e_i) then
7  |  |  |  m ← l;
8  |  |  else
9  |  |  |  M ← l;
10 |  |  end
11 |  end
12 |  a_k ← M e_i;
13 |  k ← k + 1;
14 end
```

Algorithm 3.10: MDD description. The three jumps case. (II)

```
15 for i = {(1,2),(1,3),(2,3)} do
16 |  T ← a_{i[2]}[i[2]] − 1;
17 |  Q ← 0;
18 |  repeat
19 |  |  m ← Q, M ← a_{i[1]}[i[1]];
20 |  |  while M − m > 1 do
   |  |  |  l ← ⌊(m + M)/2⌋;
21 |  |  |  if P(l e_{i[1]} + T e_{i[2]}) then
22 |  |  |  |  m ← l;
23 |  |  |  else
24 |  |  |  |  M ← l;
25 |  |  |  end
26 |  |  end
27 |  |  Q ← M;
28 |  |  if Q < a_{i[1]}[i[1]] then
29 |  |  |  m ← 0, M ← T;
30 |  |  |  while M − m > 1 do
   |  |  |  |  l ← ⌊(m + M)/2⌋;
31 |  |  |  |  if P(Q e_{i[1]} + l e_{j[2]}) then
32 |  |  |  |  |  m ← l;
33 |  |  |  |  else
34 |  |  |  |  |  M ← l;
35 |  |  |  |  end
36 |  |  |  end
37 |  |  |  a_k ← Q e_{i[1]} + l e_{i[2]};
38 |  |  |  k ← k + 1;
39 |  |  |  T ← l − 1;
40 |  |  end
41 |  until Q = a_{i[1]}[i[1]] ;
42 end
```

Algorithm 3.10 works by computing, one by one, every generator in the ideal's minimal system. For each generator we use one or two binary searches. So, its complexity is $O(s \ \log N)$ steps, where s is the number of generators. In the worst case, an upper bound for s is $2N + 1$, see [Sturmfels et al., 1995]. In practice, most of the time consumed in each step is used calling up the boolean function P, which will be proved to be computable in polynomial time.

Proposition 3.30 *Algorithm 3.10 is correct.*

Proof. By Theorem 3.1, among the generators of I_S are monomials of the form: x^a, y^b, and z^c. These are computed in lines 2–14. Lines 15–44 find every generator involving

Algorithm 3.10: MDD description. The three jumps case. (III)

```
45 c ← N − j_1 mod N;
46 b ← D(c);
47 b[1] ← b[1] + 1;
48 for i = 1, ..., k − 1 do
49 |   if a_i ≤ b then
50 |   |   k ← k − 1;
51 |   |   STOP;
52 |   end
53 |   a_k ← b;
54 end
```

two variables, and lines 45–54 work for the (possibly missing) generator with all three variables.

The key fact is that if $(a, 0, 0)$ is one of the generators we are looking for in the first part, then, for any $l \in \mathbb{N}$, $P(l, 0, 0) \iff l < a$. We can then perform a binary search to obtain the three generators.

In the second part, we start with generators involving the first two variables, continue with the one without the y, and so on. For instance, for the first case, we look at the generator $(0, a, 0)$, found in the previous step. Then, if $(q, *, 0)$ is the generator with lowest first component involving the first two variables, we can use $P(l, a-1, 0) \iff l < q$ to find q by a binary search. Once this is done, we fix the generator's second component $*$, aided by $P(q, l, 0) \iff l < *$. In a similar way, we continue to discover all the generators in this form.

Finally, there is only one generator possibly missing, which must satisfy $R(\mathbf{b}) = 0$. So, Steps 45–47 find a candidate. This possible generator is checked for possible redundancy in the remaining lines. ■

To finish the algorithm, we need a way to decide $P(\mathbf{b})$. In fact, we can use Integer Programming to solve the problem of finding a shortest path, see Lemma 3.12.

But, we need to find the minimum element according to the ordering $\prec$. We can follow Algorithm 3.11, which takes as input a matrix A to represent the monomial ordering, (see [Becker and Weispfenning, 1993]) in this way:

$$\mathbf{x} \prec \mathbf{y} \iff A\mathbf{x} \underset{\text{lex}}{<} A\mathbf{y}.$$

We represent the matrix rows with subindices: $A_1, \ldots, A_m$. Then, we obtain Algorithm 3.11.

Proposition 3.31 *Algorithm 3.11 is correct.*

Proof. Steps 1–6 are clear. The only trouble arises when the vector that we get as result of the Integer Programming-type search $\mathbf{c}$ has the same ℓ_1-norm as $\mathbf{b}$, and $\mathbf{b} \preceq \mathbf{c}$. In this case, we have to decide whether there is another vector $\mathbf{d} \in \mathbb{N}^r$, satisfying:

$$\|\mathbf{d}\|_1 = \|\mathbf{b}\|_1 = \|\mathbf{c}\|_1,\ \mathbf{d} \prec \mathbf{b}.$$

Algorithm 3.11: Deciding if a given path lies in an MDD.

Input: $j_1, \ldots, j_r, N \in \mathbb{N},\ \gcd(j_1, \ldots, j_r, N) = 1,\ A \in \mathbb{R}^{m \times r},\ \mathbf{b} \in \mathbb{N}^r$.
Output: Boolean value $P(\mathbf{b}) \leftarrow (\mathbf{b} = D(R(\mathbf{b})))$.

```
1  Execute an Integer Programming-type algorithm to get c, an element with
   minimum ℓ1-norm in R^{-1}(R(b));
2  if ||c||_1 < ||b||_1 then
3  |  OUTPUT false;
4  else
5  |  if c ≺ b then
6  |  |  OUTPUT false;
7  |  else
8  |  |  Compute a basis for the lattice
   |  |  Λ ← {c ∈ N^r | < c, (j_1, ..., j_r) >= 0, < c, (1, ..., 1) >= 0};
9  |  |  for i = 1, ..., m do
10 |  |  |  Set * ← (⟨A_i, b⟩ − (min A)/2);
11 |  |  |  Set the boolean value α, depending on whether there is a point in the
   |  |  |  set (b + Λ) ∩ N^r ∩ {c ∈ N^r | ⟨c, A_1⟩ = ⟨b, A_1⟩, ..., ⟨c, A_{i−1}⟩ =
   |  |  |  ⟨b, A_{i−1}⟩, ⟨c, A_i⟩ ≤ *};
12 |  |  |  if α then
13 |  |  |  |  OUTPUT false;
14 |  |  |  end
15 |  |  end
16 |  |  OUTPUT true;
17 |  end
18 end
```

Obviously, if such a vector $\mathbf{d}$ exists, it lies in the set $(\mathbf{b}+\Lambda)\cap\mathbb{N}^r$. So, we check in Steps 9–14 if there is another path $\mathbf{c}$ such that $A\mathbf{c} \underset{\text{lex}}{<} A\mathbf{b}$. ■

3.6 Diameter and average distance

Two notable parameters in a digraph are the diameter and the average distance. The former represents the worst delay in the communication between two nodes and the latter, the average delay. In this section we show formulae to compute those parameters in a Cayley digraph given by the irredundant irreducible decomposition of the monomial ideal associated with this graph and any graded monomial ordering.

3.6.1 Diameter

Given an MDD of a digraph $C(\Gamma,S)$, it is easy to obtain the diameter:

$$d(C(\Gamma,S)) = \max\{\|\mathbf{a}\|_1 \mid \mathbf{a}\in D(\Gamma)\}.$$

The description of the associated monomial ideal in terms of its irreducible components permits the following simplification:

Proposition 3.32 *Let Γ be a finite abelian group, $S=\{s_1,\ldots,s_r\}$ a generating set, and $\prec$ a graded monomial ordering in $\mathbb{K}[X_1,\ldots,X_r]$. If $\mathfrak{m}^{\mathbf{b}_1}\cap\cdots\cap\mathfrak{m}^{\mathbf{b}_f}$ is the irredundant irreducible decomposition of the ideal $I:=I(C(\Gamma,S),\prec)$, we have:*

$$d(C(\Gamma,S)) = \max\{\|\mathbf{b}_i\|_1 - r \mid i=1,\ldots,f\}.$$

Proof. Let D the minimum distance diagram defined by Equation (3.7). We define the *corners* of $I(C(\Gamma,S),\prec)$ as

$$E(D) := \{\mathbf{a}\in D(\Gamma) \mid \mathbf{a}+\mathbf{e}_i\notin D(\Gamma),\ \forall i=1,\ldots,r\},$$

then, it is clear that $d(C(\Gamma,S)) = \max\{\|\mathbf{a}\|_1 \mid \mathbf{a}\in E(D)\}$. We will prove that $\{\mathbf{a}+\mathbf{1} \mid \mathbf{a}\in E(D)\} = \{\mathbf{b}_1,\ldots,\mathbf{b}_n\}$,

Let $i\in\{1,\ldots,f\}$. By Theorem 3.1, we have $\mathbf{b}_i\geq\mathbf{1}$. Let us check that $\mathbf{a}:=\mathbf{b}_i-\mathbf{1}\in E(D)$. If $\mathbf{x}^{\mathbf{a}}\in I$, we would have $\mathbf{x}^{\mathbf{a}}\in\mathfrak{m}^{\mathbf{b}_i}\Rightarrow\exists j\in\{1,\ldots,r\} \mid a_j\geq b_{ij}=a_j+1$. So, $\mathbf{x}^{\mathbf{a}}\notin I$. Further, if $\exists j\in\{1,\ldots,r\}$ such that $\mathbf{x}^{\mathbf{a}+\mathbf{e}_j}\notin I$, then $\exists k\in\{1,\ldots,n\},\ k\neq i \mid \mathbf{x}^{\mathbf{a}+\mathbf{e}_j}\notin\mathfrak{m}^{\mathbf{b}_k}\Rightarrow\mathbf{a}+\mathbf{e}_j\sqsubset\mathbf{b}_k\Rightarrow\mathbf{b}_i\leq\mathbf{b}_k\Rightarrow\mathfrak{m}^{\mathbf{b}_k}\subseteq\mathfrak{m}^{\mathbf{b}_i}$. So, $\mathbf{x}^{\mathbf{a}+\mathbf{e}_j}\in I$ and $\mathbf{a}\in E(D)$.

On the other hand, let $\mathbf{a}\in E(D)$. First we will see that $I\subseteq\mathfrak{m}^{\mathbf{a}+\mathbf{1}}$. Suppose that $\mathbf{x}^{\mathbf{u}}\in I\backslash\mathfrak{m}^{\mathbf{a}+\mathbf{1}}$, then $\mathbf{a}+\mathbf{1}>\mathbf{u}\Rightarrow\mathbf{a}\geq\mathbf{u}$. Since $\mathbf{x}^{\mathbf{u}}\in I$, then $\mathbf{x}^{\mathbf{a}}\in I$; this is a contradiction because $\mathbf{a}\in E(D)$. If it was the case that $\mathfrak{m}^{\mathbf{a}+\mathbf{1}}$ were not an irreducible component in the decomposition of I, it would be satisfied:

$$\exists j\in\{1,\ldots,f\} \mid \mathfrak{m}^{\mathbf{b}_j}\subsetneq\mathfrak{m}^{\mathbf{a}+\mathbf{1}}\Rightarrow\left\{\begin{array}{l}\mathbf{a}+\mathbf{1}\leq\mathbf{b}_j\\ \mathbf{a}+\mathbf{1}\neq\mathbf{b}_j\end{array}\right\}\Rightarrow\exists i\in\{1,\ldots,r\} \mid \mathbf{x}^{\mathbf{a}+\mathbf{e}_i}\in D(\Gamma).$$

■

3.6.2 Average distance

Again, given an MDD of a Cayley digraph, it is easy to obtain the average distance. With the previous notation, let $N := \#\Gamma$ be the number of nodes.

$$\bar{d}(C(\Gamma, S)) = \frac{\sum_{g\in\Gamma} \|D(g)\|_1}{N} = \frac{\sum_{\mathbf{x}^{\mathbf{u}}\notin I} \|\mathbf{u}\|_1}{N}.$$

The following result provides a formula for computing the average distance in Cayley digraphs with a degenerated MDD.

Lemma 3.33 *Let $I = \mathfrak{m}^{\mathbf{a+1}} = \mathfrak{m}^{\mathbf{b}}$. Then,*

$$\sum_{\mathbf{x}^{\mathbf{u}}\notin I} \|\mathbf{u}\|_1 = \frac{b_1\cdots b_r}{2}(b_1+\cdots+b_r-r) = \frac{a_1+\cdots+a_r}{2}\prod_{i=1}^{r}(a_i+1).$$

Proof. Let $D := \{\mathbf{u}\in\mathbb{N}^r \mid \mathbf{x}^{\mathbf{u}}\notin I\}$. By Equation (3.3),

$$\mathbf{u}=(u_1,\ldots,u_r)\in D \iff \forall i\in\{1,\ldots,r\},\ 0\le u_i<b_i.$$

Therefore, for every $\mathbf{u}\in D$, vector $\mathbf{a}-\mathbf{u}$ lies in D. Note that these two vectors are different if and only if $2u_i = a_i$, for every component. Then,

$$\sum_{\mathbf{u}\in D}\|\mathbf{u}\|_1 = \frac{1}{2}\sum_{\substack{\mathbf{x}^{\mathbf{u}}\in D\\ \exists i: 2u_i\neq a_i}}(\|\mathbf{u}\|_1+\|\mathbf{a}-\mathbf{u}\|_1) + \sum_{\substack{\mathbf{x}^{\mathbf{u}}\in D\\ \forall i, 2u_i=a_i}}\|\mathbf{u}\|_1 = \frac{\|\mathbf{a}\|_1\#D}{2}.$$

■

Last results imply that a Cayley digraph $C(\Gamma, S)$ which admits an irreducible minimum distance diagram, i.e., a minimum distance diagram which is the complement of an irreducible ideal in the semigroup of monomials satisfies:

$$\bar{d}(C(\Gamma,S)) = \frac{1}{2}d(C(\Gamma,S)).$$

We introduce some new notation in order to discuss the general case. Let $\mathfrak{m}^{\mathbf{b}_1}\cap\cdots\cap\mathfrak{m}^{\mathbf{b}_f}$ be the irreducible decomposition of the monomial ideal I. For a nonempty set $\Delta\subseteq\{1,\ldots,n\}$, we define $\mathbf{d}_\Delta$ as the exponent of $\gcd(\mathbf{x}^{\mathbf{b}_i}\mid i\in\Delta)$. We also write:

$$\sigma(\mathbf{u}) := \frac{u_1\cdots u_r}{2}(u_1+\cdots+u_r-r).$$

Our next goal is to find a formula for the average distance. We will apply the general Inclusion-Exclusion Principle as follows:

Proposition 3.34 *Let $\mathfrak{m}^{\mathbf{b}_1}\cap\cdots\cap\mathfrak{m}^{\mathbf{b}_f}$ the irreducible decomposition of the ideal I. We have:*

$$\sum_{\mathbf{x}^{\mathbf{u}}\notin I}\|\mathbf{u}\|_1 = \sum_{\emptyset\subsetneq\Delta\subseteq\{1,\ldots,f\}}(-1)^{\#\Delta+1}\sigma(\mathbf{d}_\Delta).$$

Proof. Applying Lemma 3.33, we obtain:

$$\sum_{\emptyset\subsetneq\Delta\subseteq\{1,\dots,f\}}(-1)^{\#\Delta+1}\sigma(\mathbf{d}_\Delta)=\sum_{\Delta}(-1)^{\#\Delta+1}\sum_{\mathbf{x^u}\notin\mathfrak{m}^{\mathbf{b}_i},\ \forall i\in\Delta}\|\mathbf{u}\|_1$$

If $\mathbf{x^u}\notin I$, this is, if $\exists i\in\{1,\dots,f\}\mid\mathbf{x^u}\notin\mathfrak{m}^{\mathbf{b}_i}$; the above sum includes $\|\mathbf{u}\|_1$ the following number of times, $j\in\{1,\dots,f\}$ being the quantity of indexes i satisfying the $\mathbf{x^u}\notin\mathfrak{m}^{\mathbf{b}_i}$:

$$\binom{j}{1}-\binom{j}{2}+\cdots+(-1)^{j+1}\binom{j}{j}=1.$$

■

Considering the ideal $I_1=(x^4,x^2y^2,y^3)$ from Figure 3.4, the sum of the degrees of the monomials outside this ideal is:

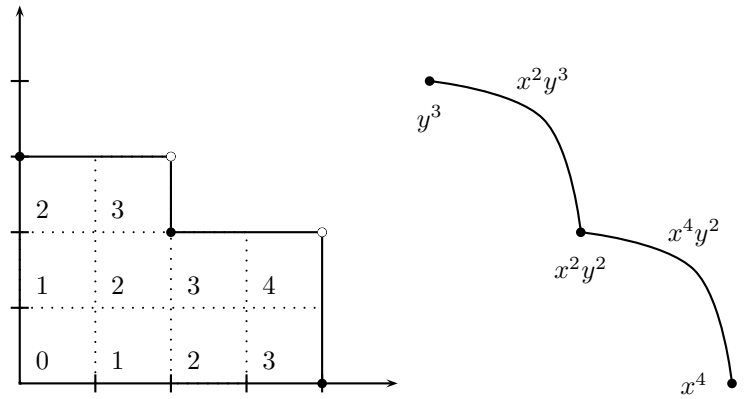


Figure 3.12:

$$\sum_{\mathbf{x^u}\notin I_1}\|\mathbf{u}\|_1=\sigma(2,3)+\sigma(4,2)-\sigma(2,2)=9+16-4=21.$$

The number of terms involved in the formula provided by Proposition 3.34 may be strongly reduced. For instance, consider Example 3.2, Proposition 3.34 solves:

$$\begin{aligned}\sum_{\mathbf{x^u}\notin I}\|\mathbf{u}\|_1=\quad&\sigma(8,2,1)+\sigma(7,2,4)+\sigma(3,3,5)+\sigma(4,3,2)+\sigma(4,5,1)-\\&-[\sigma(7,2,1)+\sigma(3,2,1)+\sigma(4,2,1)+\sigma(4,2,1)+\sigma(3,2,4)+\\&+\sigma(4,2,2)+\sigma(4,2,1)+\sigma(3,3,2)+\sigma(3,3,1)+\sigma(4,3,1)]+\\&+\sigma(3,2,1)+\sigma(4,2,1)+\sigma(4,2,1)+\sigma(3,2,1)+\sigma(3,2,1)+\\&+\sigma(4,2,1)+\sigma(3,2,2)+\sigma(3,2,1)+\sigma(4,2,1)+\sigma(3,3,1)-\\&-[\sigma(3,2,1)+\sigma(3,2,1)+\sigma(4,2,1)+\sigma(3,2,1)+\sigma(3,2,1)]+\\&+\sigma(3,2,1)=\\&=\sigma(8,2,1)+\sigma(7,2,4)+\sigma(3,3,5)+\sigma(4,3,2)+\sigma(4,5,1)-\\&-[\sigma(7,2,1)+\sigma(3,2,4)+\sigma(4,2,2)+\sigma(3,3,2)+\sigma(4,3,1)]+\\&+s(3,2,2)=454\end{aligned}$$

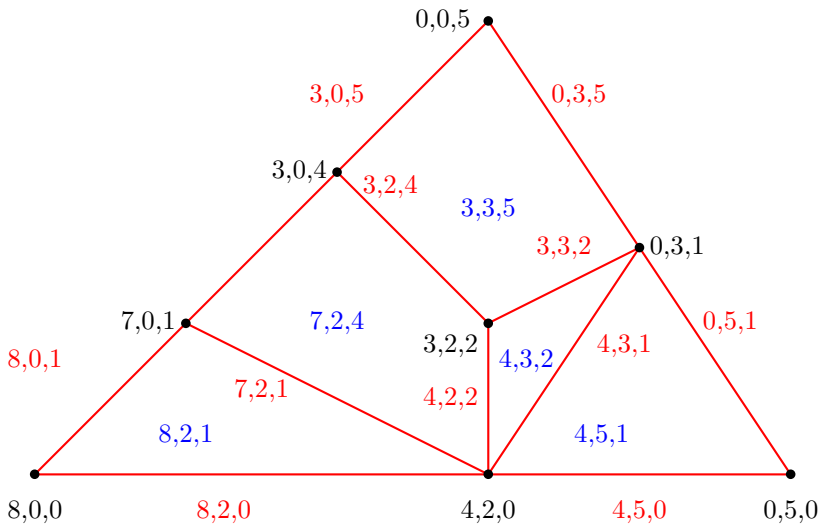


Figure 3.13:

Clearly, if $\mathbf{b} \in \mathbb{N}^r$ has a zero coordinate, then $\sigma(\mathbf{b}) = 0$. This fact produces several cancellations in the formula of Proposition 3.34. We end up with a sum of the simplex labels, affected with the sign: $+$ for faces, $-$ for edges, and $+$ for nodes.

In Cayley digraphs of degree two the associated monomial ideal only has one or two irreducible components (see Proposition 3.9), then the computation of the average distance is immediate. For digraphs of degree three we can follow this strategy:

- Construct the Miller-Sturmfels graph G as in Example 3.2 such that each irreducible component corresponds with the least common multiple of some generators of the minimal system.
- Let E be the set of all edges, F the set of faces and N the set of vertices of G

$$\bar{d} = \frac{1}{N}\left(\sum_{e\in F}\sigma(e) - \sum_{e\in E}\sigma(e) + \sum_{e\in N}\sigma(e)\right).$$

3.7 Degenerated L-Shapes

We recall that an MDD is degenerated if the associated monomial ideal is irreducible, that is, of the form $\left(X_1^{\alpha_1}, \ldots, X_r^{\alpha_r}\right)$. In general, the family of graphs with this property need not present an optimal nodes/diameter ratio. In this section we present families of circulant digraphs with a degenerated MDD and a relatively small diameter.

Proposition 3.35 *Let a, s, k be natural numbers such that $\gcd(a, s) = 1$ and $a < s$. The monomial ideal associated with $C_{sk}(a, s)$ and any monomial ordering is $I = (x^s, y^k)$.*

Proof. Since $\mathbb{K}[x, y]/I$ is an artinian ring (see Theorem 3.1), the minimal system of generators of I contains monomials of the form x^α, y^β. We claim that $\beta = k$. In order

to prove it, let us see $D(si) = (0, i)$, for $i = 0, \ldots, k-1$. Indeed, let $i \in \{0, \ldots, k-1\}$ and suppose that $\exists (u, v) \in \mathbb{N}^2$ such that $u + v \leq i$ and $R(u, v) = si$. Then,

$$si \equiv_{sk} au + sv \Rightarrow \exists h \in \mathbb{N} \mid si = au + sv + hsk \Rightarrow s|au \Rightarrow s|u \Rightarrow$$

$$\Rightarrow \begin{cases} u = 0 \\ \vee \\ \exists t \in \mathbb{N}^* \mid u = st \end{cases}$$

In the first case, we have:

$$i = v + kh \leq k - 1 \Rightarrow h = 0 \Rightarrow v = i.$$

In the second,

$$i = at + v + kh \leq k - 1 \Rightarrow h = 0,\ i = at + v \geq u + v \Rightarrow at \geq u,$$

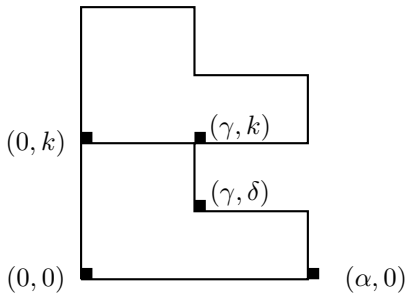
but this a contradiction because $a < s$. So, $\beta \geq k$. On the other hand, $D(0, k) = 0 = D(0, 0)$ implies $\beta = k$. Finally, suppose that

$$I = (x^\alpha, x^\gamma y^\delta, y^k),\ \gamma < \alpha,\ \delta < k,\ R(\gamma, \delta) = R(0, k) = 0.$$

Thus,

$$R(\gamma, k) = R(\gamma, \delta) + R(0, k - \delta) = R(0, k - \delta),$$
$$R(\gamma, k) = R(\gamma, 0) + R(0, k) = R(\gamma, 0).$$

Therefore, one of the two vectors $(0, k - \delta)$, $(\gamma, 0)$ should be in I, which is false. Consequently, I is degenerated and $sk = \dim \left(\mathbb{K}[x,y]/(x^\alpha, y^k)\right) = \alpha k \Rightarrow s = \alpha$. ■



The following example shows that we cannot omit from the above result hypotheses the requirement $a < s$.

Example 3.36 *The monomial ideal I associated with $C_{60}(7, 6)$ and any graded monomial ordering in $\mathbb{K}[x, y]$ is $I = (x^{12}, x^6y^3, y^7)$.*

Using Gröbner bases theory and previous results we can generalize Proposition 3.35 from two jumps to an arbitrary number of them:

Proposition 3.37 *Let $\alpha_1, \ldots, \alpha_r$ be positive integers, neither of them equal to one. Then, setting $N = \alpha_1 \cdots \alpha_r$, the monomial ideal associated with the digraph*

$$C_N(1, \alpha_1, \alpha_1\alpha_2, \ldots, \alpha_1 \cdots \alpha_{r-1})$$

and any graded monomial ordering is $(X_1^{\alpha_1}, \ldots, X_r^{\alpha_r})$. The reduced Gröbner basis of the associated binomial ideal is $\{X_i^{\alpha_i} - X_{i+1} \mid \ i = 1, \ldots, r-1\} \cup \{X_r^{\alpha_r} - 1\}$.

Proof. Firstly, every element of the proposed basis lies in the binomial ideal, for their associated lattice points $\{(0, \ldots, \overset{i}{\alpha_i}, -1, \ldots, 0) \mid i = 1, \ldots, r-1\} \cup \{(0, \ldots, 0, \alpha_r)\}$ are paths with node 0 as destination. Then, the initial ideal I of this lattice ideal satisfies:

$$(X_1^{\alpha_1}, \ldots, X_r^{\alpha_r}) \subseteq I.$$

Now, we know that the dimension of the quotient vector space $\mathbb{K}[\mathbf{x}]/I$ equals the number of nodes $N = \alpha_1 \cdots \alpha_r$. Yet, the dimension of $\mathbb{K}[\mathbf{x}]/(X_i^{\alpha_i} \mid i = 1, \ldots, r)$ is N, so both ideals must coincide. Then, in order to obtain a reduced Gröbner basis, we must have one binomial for each generator in the initial ideal. That is, the reduced Gröbner basis is:

$$\{X_i^{\alpha_i} - m_i(\mathbf{x}) \mid i = 1, \ldots, r\},$$

where m_i is a monomial satisfying $m_i \notin (X_1^{\alpha_1}, \ldots, X_r^{\alpha_r})$. Then, $m_i = \mathbf{x}^{\mathbf{a}}$, with $a_i < \alpha_i, i = 1, \ldots, r$. Let us set $X_{r+1} := 1$. Now, $(X_i^{\alpha_i} - X_{i+1}) - (X_i^{\alpha_i} - m_i) = m_i - X_{i+1}$ is an ideal element. If $m_i \neq X_{i+1}$, we would have $\alpha_{i+1} = 1$, a contradiction. ■

The following result is an immediate consequence:

Corollary 3.38 *Let d, r be two positive integers. Let k be the residue class of d modulo r. Then, if we fix:*

$$\alpha_1 = \cdots = \alpha_k = \frac{d-k}{r} + 2, \ \alpha_{k+1} = \cdots = \alpha_r = \frac{d-k}{r} + 1,$$

the following is a directed circulant graph with r jumps, $N := \alpha_1 \cdots \alpha_r$ nodes, and diameter d:

$$\mathcal{C}_N(1, \alpha_1, \alpha_1\alpha_2, \ldots, \alpha_1 \cdots \alpha_{r-1}).$$

We note that the number of vertices is:

$$N = \left(\frac{d-k}{r} + 2\right)^k \left(\frac{d-k}{r} + 1\right)^{r-k}.$$

Once r is fixed, increasing the diameter d makes the number of nodes in this graph family increase as $O(d^r)$.

Proposition 3.35 provides a family with diameter $2\sqrt{N} - 2$ and average distance $\sqrt{N} - 1$. Let $d > 1$ be a natural number:

$$C_{\left(\frac{d+2}{2}\right)^2}\left(1, \frac{d+2}{2}\right), \ if \ d \equiv 0 \bmod 2 \quad and \quad C_{\frac{(d+1)(d+3)}{4}}\left(1, \frac{d+1}{2}\right), \ if \ d \equiv 1 \bmod 2.$$

Basically, this family was discovered in the paper [Wong and Coppersmith, 1974]. However, determining $d_2(N)$ and finding the optimal $C_N(j_1, j_2)$ is an open problem.

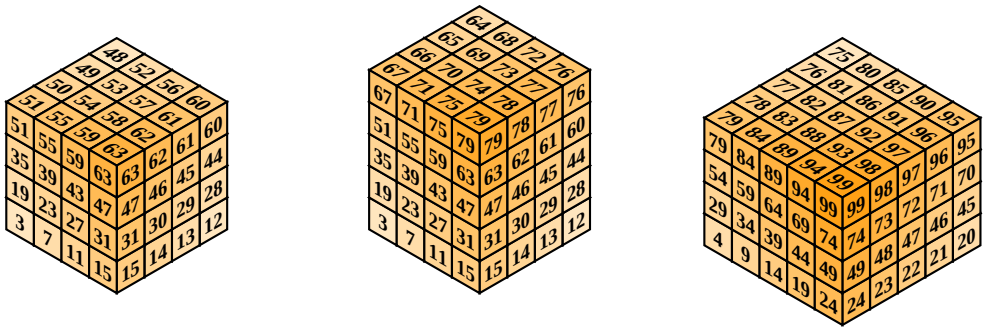
In the case of undirected circulant graphs of degree four, i.e, $C_N(j_1, -j_1, j_2, -j_2)$, several papers have shown that the lower bound $\frac{1}{2}\left(\sqrt{2N-1}-1\right)$ can be achieved by taking $j_1 = \frac{1}{2}\left(\sqrt{2N-1}-1\right)$ and $j_2 = \frac{1}{2}\left(\sqrt{2N-1}-1\right)+1$, (see the survey [Bermond et al., 1995]). In the middle, that is, between circulant digraphs of degree two and circulant graphs of degree four, Proposition 3.37 and the above Corollary provide a very attractive family of circulant graph of degree 3 (see Figure 3.14). Let $d > 2$ be a natural number:

$$C_{\left(\frac{d+3}{3}\right)^3}\left(1, \frac{d+3}{3}, \left(\frac{d+3}{3}\right)^2\right), \text{ if } d \equiv 0 \bmod 3,$$

$$C_{\frac{(d+2)^2(d+5)}{27}}\left(1, \frac{d+2}{3}, \left(\frac{d+2}{3}\right)^2\right), \text{ if } d \equiv 1 \bmod 3,$$

$$C_{\frac{(d+4)^2(d+1)}{27}}\left(1, \frac{d+4}{3}, \left(\frac{d+4}{3}\right)^2\right), \text{ if } d \equiv 2 \bmod 3.$$

Graphs in this family have diameter d and average distance $d/2$.



d=9=4+4+4-3 d=10=4+4+5-3 d=11=5+5+4-3

Figure 3.14: Family of circulant digraphs

Chapter IV

Software

In this chapter we comment two software applications we have performed as a complement to the research in the two topics we have studied. On the one hand, we explain how the algorithms from Chapter II have been implemented and show the obtained numerical results. On the other one, we have developed the software CIRCULE, for drawing circulant digraphs and associated diagrams.

4.1 Small Roots

Chapter II presents algorithms for finding small roots of certain systems of multivariate polynomials over $\mathbb{F}_p$. The problem can also be regarded as finding a root of a multivariate system of polynomials from a known approximation to that root. Theoretical bounds on the approximation errors permitted for the algorithms to work are given, and some experiments have been done showing that the empirical results match the predicted behaviour. In this section we explain how those experiments were carried out.

We have performed C++ implementations of the proposed algorithms using NTL library [Shoup], which permits the usage of arbitrary big integers. The core of every algorithm in Chapter II is finding a shortest solution of a linear system of congruences. It is simple to compute the set of solutions as a translated lattice $\mathbf{c} + \Lambda$, reducing the problem to solving a closest vector problem. For that task, we have followed the algorithm proposed in [Agrell et al., 2002], which is based on a method of M. Pohst [1981]. In short, that method consists of three steps:

- We would like to obtain a basis as close to orthogonal as possible. We build an LLL reduced basis $\{\mathbf{b}_1, \ldots, \mathbf{b}_n\}$ and select the biggest vector $\mathbf{b}_n$ in that basis.

- Any lattice element can be expressed as an integer combination of the reduced basis $\mathbf{v} = x_1\mathbf{b}_1 + \cdots + x_n\mathbf{b}_n$. The lattice is divided in layers $\{x_n = k\}_{k\in\mathbb{Z}}$, and the distance between adjacent layers equals $\|\mathbf{b}_n^*\|$.

- We can reduce recursively the problem to a bunch of CVP instances of dimension $n-1$, as sketched in Figure 4.1, taking advantage of known bounds on the distance from $\mathbf{c}$ to its closest lattice vector.

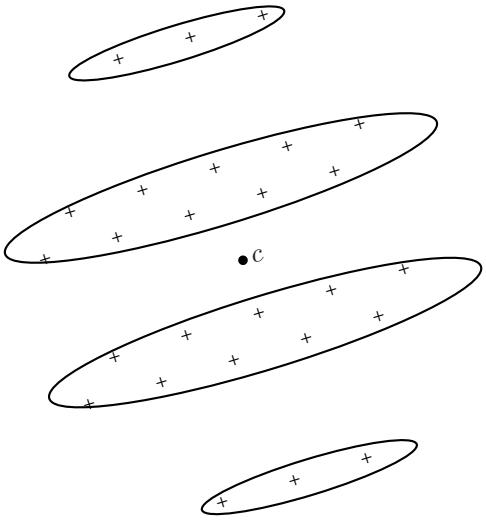


Figure 4.1: Layers to check for the CVP search

In the following, we analyse the obtained results comparing them with the expected behaviour.

4.1.1 Pollard Generator

We analyse the obtained result for the Pollard Generator, which is the most involved method from Section 2.1. Testing the algorithm from [Blackburn et al., 2005] that leads to Theorem 2.3 with this implementation we can confirm the threshold $\delta = 1/3$ for the maximum fraction of bits hidden. However, the two-round algorithm leading to Theorem 2.4 obtains successfully the expected solution in instances even more difficult than $\delta = 5/14 \simeq 0.35714$. The following table shows these tests results for a 1000-bit prime generator, following the one-round algorithm from [Blackburn et al., 2005] and the two-round one presented in this thesis.

δ	0.3	0.3307	0.3333	0.3385	0.3615	0.3641
1r	100%	100%	50%	0	0	0
2r	100%	100%	100%	100%	98%	0

Tests with generators over a finite field of bigger size set empirically the threshold in $\delta \simeq 0.363$. However, we fail to give a rigorous proof for a bound better than $\delta = 5/14$.

When the shift c remains unknown as well, we can improve the bound $\delta = 0.25$ provided by Blackburn et al. [2005] up to $\delta \simeq 0.2613$. Anyway, this is clearly worse than the heuristic method proposed in [Blackburn et al., 2005], which attains $\delta = 1/3$.

4.1.2 Linear congruential generator over elliptic curves

In order to perform these tests, we have implemented a simple template to operate with points in elliptic curves over an arbitrary finite prime field with characteristic

distinct to 2 or 3.

Firstly, we generate an elliptic curve over a prime finite field of a desired size by choosing randomly in $\mathbb{F}_p$ parameters a, b to fix Equation (2.32). Then, we generate randomly points in the curve by choosing their first coordinate and trying to solve Equation (2.32). For several pairs of points, an EC-LCG is simulated, and approximations to some consecutive values are given as input to our algorithms.

We summarize its results in the following table. We have selected primes of several sizes, and note the obtained success threshold. As we can see, for the first method 1/6 appears as the correct threshold:

$\log_2(p)$	50	100	500	1000
$\log_p(\Delta)$	0.15	0.156	0.164	0.165

As for the second algorithm proposed, for the case when the composer G is unknown, the threshold has been obtained using the so-called Gaussian heuristic. We fail to obtain results according to our predictions, and we think this is due to the small approximation errors permitted. Recall that we prove that, roughly, the algorithm would work when $\Delta < p^{1/46}$. In order to test examples with a reasonable high tolerance Δ, we would need a very large size for the prime p, exceeding our computing possibilities. Note also that in this case the dimension of the employed lattice is bigger, increasing the constant hidden in the O-notation.

4.2 Circule

In this Section we present Circule [Ibeas], a tool for drawing circulant digraphs and associated diagrams. This software has been developed with two objectives. On the one hand, it provides a simple tool for compiling figures to illustrate didactic or research material. On the other, it is a simple resource to quickly view the associated diagram of a circulant, and therefore, it may be useful to develop or check conjectures on this topic.

Circule runs under a UNIX system with X-windows. It works with a current circulant digraph, which can be loaded from a text file with the format:

```
(N;j1,j2,...,jr)
```

Graphs with degree up to 10 are allowed. A user-friendly interface is also provided for setting the graph parameters'. The resulting figure is output as an EPS file.

When the current circulant $C(N; j_1, \ldots, j_r)$ is connected and its degree r is not grater than 3, an associated diagram is drawn, following next formula:

$$D(g) = \min_{\prec}\{R^{-1}(g)\},\ g = 0, \ldots, N-1,$$

where R is the routing function introduced in Equation (3.5) and $\prec$ is any monomial ordering, not necessarily graded. This means that the diagram need not be a minimum distance diagram.

The monomial ordering $\prec$ is represented as an r-row matrix $O \in \mathbb{R}^{r \times m}$ such that

$$\mathbf{a} \prec \mathbf{b} \iff O\mathbf{a} < O\mathbf{b},$$

where $<$ represents the pure lexicographic ordering in $\mathbb{R}^m$. The ordering can be read from an input file, or selected directly from the three listed choices: graded lexicographic, reverse graded lexicographic, and pure lexicographic.

Algorithm 3.1, introduced in Chapter III, works only for graded monomial algorithms. For the general case, we have used Algorithm 4.1. This procedure is more efficient than Algorithm 3.1, for one need not visit all the elements in $\mathbb{N}^r$ up to a certain degree.

Algorithm 4.1: MDD construction.

Input: $\Gamma = \{g_i \mid 0 \leq i < N\}$, abelian group, $\{s_1, \ldots, s_r\}$, generating set, $\prec$, monomial ordering in $\mathbb{N}^r$.
Output: $D(g_i),\ i = 0, \ldots, N-1$.

```
1  D[g_0, ..., g_{N-1}] ← ∅̄, S ← 0, a ← 0, L ← {∅};
2  while S < N do
3  |  g ← R(a);
4  |  if D(g) = ∅ then
5  |  |  D(g) ← a;
6  |  |  S ← S + 1;
7  |  |  for i = 1, ..., r do
8  |  |  |  L ← L ∪ {a + e_i};
9  |  |  end
10 |  end
11 |  a := min(L);
12 |  L ← L\{a};
13 end
```

The diagram can be coloured in an uniform way, or according to the length of each depicted path. Finally, in the 3-D case, the diagram can be viewed under an axonometric, conic, or cavalier perspective.

Further work

In Chapter II we have presented several variations of an algorithm for predicting pseudorandom sequences from partial information. For each of these variations we have needed to compute a theoretical threshold for the maximum approximation errors allowed. It would be interesting to delve into the understanding of these methods, maybe being able to determine that threshold for more general situations.

On the other hand, the two-round approach explained in Section 2.1.3 seems to be a feasible way to improve the predicting results for many generators, or even for other applications of lattice basis reduction. However, the dimension of the employed lattices grows with this approach. Moreover, the obtained threshold is not tight, as is shown with the experiments (see Section 4.1.1). It might be possible to obtain tighter bounds by a more careful examination of the exceptional set. In order to extend this technique to other situations, we might need a generalization of Lemma 1.12 for LLL reduced basis in arbitrary dimensions.

In all of these algorithms, the prime p is assumed to be known. It would be interesting to explore the possibility of developing attacks able to discover that parameter as well. A heuristic approach for that purpose in the linear case is presented in [Joux and Stern, 1998]. However, it is not clear how to extend that approach (even just heuristically) to the case of nonlinear generators.

Concerning the linear generator over elliptic curves, it is natural to look for a predicting algorithm that does not require the knowledge of parameter G with better performance that the one presented in Section 2.2.4. In a more general setting, it may be possible to extend this kind of linearizing techniques to other generators over elliptic curves.

In Chapter III we have proposed monomial ideals as a natural tool to study Cayley digraphs with a finite abelian group as vertex set. We have generalized the L-shape concept in the plane to L-shape in the r-dimensional affine space. We think that this new point of view may shed light on problems in multi-loop computer networks. We also have introduced the Gröbner bases theory in this context, which seems quite useful. Many interesting questions remain unsolved. The problem of deciding whether a circulant digraph has a Hamiltonian loop [Fiol and Yebra, 1988] is solved for the degree two case but remains an open question for the general case. This problem has a beautiful connection with a combinatorial problem arising from bell ringing [Rankin, 1948]. Indeed, that paper solves this question for $r = 2$. Perhaps the use of monomial and binomial ideals can facilitate the research of that topic.

On the other hand, it would be interesting to provide fault tolerance routing algorithms. From a more practical point of view, it may be useful to investigate the implementation in computer networks of the proposed family of circulant digraphs under parameters such as routing, fault tolerance, etc.

Sumario

Esta memoria, titulada "Predicción de sucesiones pseudoaleatorias y descripción de digrafos de Cayley mediante retículas enteras", estudia dos problemas independientes que pueden inscribirse dentro del campo de la Teoría de la Comunicación. Por una parte, la predicción de sucesiones pseudoaleatorias tiene interés en el contexto de la Criptología. Por otra, los digrafos de Cayley generalizan un tipo de grafos muy estudiados en Arquitectura de Computadores y diseño de redes, los digrafos circulantes. En ambos problemas utilizamos retículas (lattices) enteras como herramienta básica.

El contenido está dividido en cinco partes redactadas en inglés y resumidas en este sumario. En el Capítulo I se repasan la definición y principales propiedades de las retículas, objeto matemático fundamental en esta memoria. A continuación, el Capítulo II presenta los resultados obtenidos para la predicción de algunos generadores concretos de números pseudoaleatorios, utilizando una técnica de reducción de bases de retículas. El Capítulo III estudia los digrafos de Cayley y sus diagramas de mínima distancia, proporcionando algoritmos para el cálculo de ciertos parámetros de los digrafos circulantes. En el Capítulo IV se exponen los resultados experimentales obtenidos mediante la implementación de los algoritmos propuestos en el Capítulo II y el programa CIRCULE, que puede resultar útil a los interesados en el estudio de los digrafos circulantes. Finalmente, se mencionan las líneas de trabajo que pueden continuar el estudio recogido en esta memoria.

1 Conceptos básicos sobre retículas

En el estudio sobre las formas cuadráticas de C.F. Gauss se encuentra implícito el concepto de retícula. En castellano se utiliza comúnmente la palabra "retículo" para referirse a este objeto. Nosotros, sin embargo, hemos preferido el femenino "retícula", en principio para evitar la confusión con la estructura algebraica homónima que abstrae las operaciones binarias mínimo y máximo en un conjunto parcialmente ordenado. La elección del femenino puede justificarse también atendiendo al significado común en castellano de la palabra, que refleja la interpretación intuitiva del objeto matemático, en el caso bidimensional.

Dejando de lado estas consideraciones, una *retícula* se define como un subrupo discreto de $(\mathbb{R}^m, +)$. Es decir, un subgrupo del grupo abeliano $\mathbb{R}^m$ que no tiene puntos de acumulación, como se formaliza en la Definición 1.1. A lo largo de esta memoria utilizamos letras en negrita para representar los vectores de $\mathbb{R}^m$ y la notación $A = [\mathbf{a}_1|\cdots|\mathbf{a}_n] \in \mathbb{R}^{m\times n}$ para agrupar varios vectores de $\mathbb{R}^m$ en una matriz. Con este

convenio, el subgrupo generado por las columnas de la matriz A es:

$$\mathcal{L}(A) := \{A\mathbf{x} \mid \mathbf{x} \in \mathbb{Z}^n\} \subset \mathbb{R}^m.$$

Si las columnas de A son linealmente independientes, el subgrupo $\mathcal{L}(A)$ que generan es una retícula. Lo mismo sucede cuando las componentes de la matriz son racionales. Es decir, el subgrupo generado por un conjunto de vectores racionales es una retícula.

Un conjunto de vectores linealmente independientes de $\mathbb{R}^m$ que genere una retícula se llama *base*. Toda retícula admite una base que la genera, y el conjunto de todas las bases se obtiene mediante transformaciones *unimodulares*. Es decir, dos matrices B_1, B_2 son bases de la misma retícula si y sólo si existe una matriz cuadrada P con determinante unidad (± 1) tal que $B_1 P = B_2$. Llamamos *dimensión* de una retícula a la cantidad de elementos de cualquier base.

Una matriz $B \in \mathbb{R}^{m\times n}$ de rango n determina una forma cuadrática definida positiva en $\mathbb{R}^n$, llamada forma grammiana, con matriz asociada B^tB. La cantidad $\sqrt{\det(B^tB)}$ es un invariante de la retícula $\mathcal{L}(B)$ que se denomina *volumen* o *determinante* de la retícula y se denota $\mathrm{vol}(\Lambda)$.

El principal problema computacional relacionado con este objeto es el llamado problema del vector menor (SVP):

Definición 1 (SVP) *Dadas una base $B \in \mathbb{Z}^{m\times n}$ de una retícula Λ y una norma $\|\cdot\|$ en $\mathbb{R}^m$, el problema del vector menor consiste en encontrar $\mathbf{u} \in \Lambda\backslash\{0\}$ tal que $\|\mathbf{u}\| \leq \|\mathbf{v}\|$, para todo $\mathbf{v} \in \Lambda\backslash\{0\}$.*

En el contexto original del estudio de las formas cuadráticas, este problema viene motivado por la búsqueda del menor valor representado por una forma cuadrática definida positiva $q(\mathbf{x})$ cuando sus argumentos son números enteros.

La teoría conocida como Geometría de los Números, introducida por H. Minkowski, aborda este problema mediante el estudio de dominios convexos en $\mathbb{R}^n$. El principal resultado de esta teoría es el siguiente:

Teorema 2 (Minkowski) *Sean Λ una retícula de dimensión n y $\mathbb{R}\langle\Lambda\rangle \subseteq \mathbb{R}^m$ el espacio vectorial que genera. Si $S \subseteq \mathbb{R}\langle\Lambda\rangle$ es un conjunto convexo, simétrico con respecto al origen y medible con medida $\mu(S) > 2^n \mathrm{vol}(\Lambda)$, se tiene $S \cap \Lambda \neq \{0\}$. Por tanto, el conjunto S contiene al menos tres puntos no nulos de la retícula.*

Como consecuencia inmediata, se obtiene una cota para la norma del menor vector no nulo en una retícula. Utilizamos la notación $\lambda_1(\Lambda)$ para el mínimo de las normas de los vectores no nulos de Λ, según la Definición 1.5, y $\mathrm{B}_{\|\cdot\|}(1)$ para la bola de centro 0 y radio 1 de la norma $\|\cdot\|$ en $\mathbb{R}^m$.

$$\lambda_1(\Lambda) \leq 2\left(\left(\mu(\mathrm{B}_{\|\cdot\|}(1))\right)^{-1}\mathrm{vol}(\Lambda)\right)^{1/n}.$$

Particularizando esta cota en el caso de la norma euclídea, $\lambda_1(\Lambda) = O(n^{1/2}(\mathrm{vol}\,\Lambda)^{1/n})$. La heurística Gaussiana sugiere que es improbable que una retícula contenga un vector no nulo con norma sustancialmente inferior a esa cota.

El problema del vector menor es NP-duro para la norma ℓ_∞, y se asume comúnmente que lo es también para cualquier norma ℓ_p, con $p \geq 1$. Esto último no está demostrado

para reducciones deterministas (a la Karp), pero sí para reducciones probabilistas [Ajtai, 1998]. De modo más simple, puede probarse que la versión no homogénea del SVP es un problema NP-duro.

Definición 3 (CVP) *Dados una base $B \in \mathbb{Z}^{m\times n}$ de una retícula Λ, un vector $\mathbf{s} \in \mathbb{R}^m$ y una norma $\|\cdot\|$ in $\mathbb{R}^m$, el problema del vector más próximo (CVP) consiste en encontrar $\mathbf{u} \in \mathbf{s} + \Lambda$ tal que $\|\mathbf{u}\| \leq \|\mathbf{v}\|$, para todo $\mathbf{v} \in \mathbf{s} + \Lambda$.*

Los algoritmos habituales para resolver o aproximar los problemas SVP y CVP consisten en transformar la base incial en otra *equivalente* (es decir, que genera la misma retícula) que presente propiedades interesantes. Por ejemplo, un modo de resolver SVP es hallar una base equivalente a la dada que cuente entre sus elementos con un vector de norma mínima entre los no nulos de la retícula. En general, es deseable obtener bases que minimicen la cantidad siguiente:

Definición 4 *Sea $B = [\mathbf{b}_1|\cdots|\mathbf{b}_n] \in \mathbb{R}^{m\times n}$ una matriz de rango n. Se define su defecto ortogonal como*

$$\mathrm{od}(B) := \frac{\prod_{i=1}^{n} \|\mathbf{b}_i\|}{\mathrm{vol}(\mathcal{L}(B))} = \prod_{i=2}^{n} \sin^{-1}(\alpha_i),$$

donde α_i es el ángulo entre $\mathbf{b}_i$ y el espacio vectorial $\mathbb{R}\langle \mathbf{b}_1, \ldots, \mathbf{b}_{i-1}\rangle$.

De la ecuación anterior se desprende que al acercase una base a la ortogonalidad, decrece la norma de sus elementos. Por lo tanto, dada una retícula mediante una base, el objetivo idóneo sería obtener una base ortogonal equivalente. Sin embargo, no toda retícula admite una base ortogonal. En el caso bidimensional, existe una definición de base reducida de modo que una base de ese tipo tiene el menor defecto ortogonal entre todas las de su clase de equivalencia y que toda retícula admite una base reducida.

Definición 5 (Gauss) *Sea $B = [\mathbf{b}_1|\mathbf{b}_2] \in \mathbb{R}^{m\times 2}$ una matriz de rango 2. Se dice que B es reducida si se verifica:*

$$\|\mathbf{b}_1\| \leq \|\mathbf{b}_2\| \leq \|\mathbf{b}_1 - \mathbf{b}_2\|, \|\mathbf{b}_1 + \mathbf{b}_2\|.$$

Por lo tanto, para resolver el SVP en una retícula bidimensional basta con obtener una base reducida de la retícula de entrada. Existe un método eficaz (Algoritmo 1.1) para este propósito que generaliza el algoritmo centrado de Euclides. Este método fue propuesto por C.F. Gauss en el contexto de las formas cuadráticas (restringiéndose por tanto a la norma euclídea) y extendido en [Kaib and Schnorr, 1996] para cualquier norma computable.

Este buen comportamiento del caso bidimensional no ha podido extenderse al caso general. Si se procura obtener para toda retícula de dimensión arbitraria una base con propiedades tan exigentes como en el caso de las bases reducidas de Gauss, no se consigue un algoritmo eficaz para computar esa base. Recordamos que SVP es NP-duro, según una conjetura ampliamente asumida, por lo que es improbable obtener un algoritmo eficaz que logre el cálculo para una retícula arbitraria de una base que incluya un vector de norma mínima. Si bien se han propuesto varias definiciones de "base

reducida", es a raíz de la aparición de la llamada reducción LLL en [Lenstra et al., 1982] que las técnicas de reducción de retículas han cobrado más popularidad. En ese trabajo, se propuso un algoritmo de complejidad polinómica en la talla de la base de entrada que devuelve una base con propiedades menos exigentes que las empleadas hasa entonces (Minkowski y Korkine-Zolotarev, principalmente). La norma euclídea del primer vector de la base devuelta por el algoritmo de reducción LLL es una aproximación exponencial de la del más pequeño:

$$\|\mathbf{b}_1\|_2 \leq 2^{(n-1)/2}\lambda_1(\Lambda).$$

Esta aproximación es suficiente para lograr el objetivo original del algoritmo LLL: desarrollar un algortimo de complejidad polinómica para factorizar polinomios univariados sobre los racionales. Del mismo modo, la reducción LLL se ha revelado útil en otros problemas.

Terminamos el Capítulo I con una breve introducción a los *ideales reticulares*, estudiados en [Sturmfels et al., 1995], que serán de utilidad en el capítulo dedicado a la descripción de digrafos de Cayley. Representamos los monomios de $\mathbb{K}[X_1, \ldots, X_m]$ con la notación $\mathbf{x}^{\mathbf{a}} := X_1^{a_1} \cdots X_m^{a_m}$, donde $\mathbf{a} = (a_1, \ldots, a_m) \in \mathbb{N}^m$. Utilizamos las notaciones $\mathbf{a}^+, \mathbf{a}^-$ para las *partes positiva* y *negativa*, respectivamente, de un vector $\mathbf{a} \in \mathbb{Z}^m$. Es decir, los únicos vectores con componentes no negativas que satisfacen

$$\mathbf{a} = \mathbf{a}^+ - \mathbf{a}^-.$$

En el mencionado artículo, a cada retícula entera se le asocia un ideal binomial, del modo siguiente:

$$I_\Lambda := (\mathbf{x}^{\mathbf{a}^+} - \mathbf{x}^{\mathbf{a}^-} \mid \mathbf{a} \in \Lambda).$$

A partir de un conjunto de binomios generadores de I_Λ, se obtiene de forma inmediata una base de la retícula Λ, puesto que todo binomio en el ideal ha de ser de la forma $\alpha\mathbf{x}^{\mathbf{a}} - \alpha\mathbf{x}^{\mathbf{b}}$, para ciertos $\mathbf{a}, \mathbf{b} \in \mathbb{N}^m$ y $\alpha \in \mathbb{K}$. De esta forma, si el ideal I_Λ está generado por $\{\alpha_1\mathbf{x}^{\mathbf{a}_1} - \alpha_1\mathbf{x}^{\mathbf{b}_1}, \ldots, \alpha_n\mathbf{x}^{\mathbf{a}_n} - \alpha_n\mathbf{x}^{\mathbf{b}_n}\}$, la retícula Λ está generada por $\{\mathbf{a}_1 - \mathbf{b}_1, \ldots, \mathbf{a}_n - \mathbf{b}_n\}$. El paso contrario, es decir, obtener un conjunto de binomios generadores de I_Λ a partir de una base de Λ no es tan inmediato, si bien puede utilizarse el resultado siguiente para obtener un sistema generador de Λ que sí se traduzca en un conjunto de binomios que genere el ideal asociado.

Lema 6 *Sea $A = [\mathbf{a}_1|\cdots|\mathbf{a}_n] \in \mathbb{Z}^{m\times n}$. Si para cada índice $i = 1, \ldots, m$ hay una columna a_j de A que no tiene ninguna componente negativa y tal que su i-ésima componente no es nula, entonces*

$$I_{\mathcal{L}(A)} = (\mathbf{a}^{\mathbf{a}_i^+} - \mathbf{x}^{\mathbf{a}_i^-} \mid 1 \leq i \leq n).$$

Por ejemplo, consideramos la matriz entera

$$B := \begin{bmatrix} 1 & -2 \\ -1 & 1 \end{bmatrix},$$

cuyo determinante es -1, lo que es suficiente para probar que genera la retícula $\mathbb{Z}^2$. Por lo tanto, el ideal binomial asociado es $I_{\mathbb{Z}^2} = (x - 1, y - 1)$. Sin embargo, si tratáramos de calcularlo directamente a partir de las columnas de B, obtendríamos

$(x - y, y - x^2) \subsetneq I_{\mathbb{Z}^2}$. Para hacer uso del lema anterior, podemos ampliar B a un sistema generador que satisfaga sus hipótesis, como

$$C := \begin{bmatrix} 1 & -2 & 1 & 0 \\ -1 & 1 & 0 & 1 \end{bmatrix},$$

y completar el cálculo del ideal asociado.

2 Predicción de sucesiones pseudoaleatorias

Los generadores de números pseudoaleatorios conforman una alternativa práctica a los procesos ortodoxos para seleccionar números al azar, con la salvedad de que la aleatoriedad de la sucesión obtenida es solamente aparente, puesto que estos generadores están gobernados por un proceso determinista. Cuando un proceso automatizado hace uso de una fuente de aleatoriedad, suele requerirse que esa fuente sea rápida y continuada. En esta situación, recurrir a mediciones de procesos físicos para obtener cantidades al azar no es lo suficientemente eficaz. Se han elaborado tablas de números aleatorios con la finalidad de ser utlizadas, por ejemplo, para calcular una integral determinada mediante el método Monte Carlo. En general, esta opción tampoco resulta del todo eficiente (o económica).

Consideraremos generadores que producen recursivamente una sucesión de números, atendiendo aparentemente a una distribución uniforme en el rango donde se toman. En concreto, utilizaremos generadores de números en un cuerpo finito primo $\mathbb{F}_p$. En ocasiones trataremos los elementos de este conjunto como números enteros en el rango $\{0, \ldots, p-1\}$, y recíprocamente, identificaremos los números enteros con su clase de restos módulo p. Estos generadores tienen aplicaciones en Criptología, como por ejemplo el criptosistema simétrico de cifrado en flujo, en el que el mensaje a enviar se combina bit a bit con una "máscara" calculada independientemente por los dos comunicantes mediante un generador de números pseudoaleatorios, a partir de una clave secreta compartida.

Esta aplicación explota la característica fundamental que diferencia los generadores de números psedualeatorios de las sucesiones "verdareramente" aleatorias, consistente en que las sucesiones obtenidas por los primeros pueden reproducirse con exactitud a partir de cierta información (los parámetros que gobiernan el generador). En el contexto de la Criptología, es natural exigir a los generadores utilizados que las sucesiones resultantes sean difíciles de predecir por parte de un observador externo. En el Capítulo II exponemos varios métodos que permiten la reproducción de una sucesión pseudoaleatoria a partir de información parcial de varios de sus elementos. Estos algoritmos están inspirados en los trabajos de D. Coppersmith para el cómputo de raíces pequeñas de polinomios utilizando reducción de bases de retículas [Coppersmith, 1997]. Hacen uso de técnicas similares a las publicadas en [Blackburn et al., 2003, 2005, 2006]. La información inicial que se precisa para predecir el generador son aproxiamaciones de varios valores consecutivos. Para un cierto entero positivo Δ, diremos que $w \in \mathbb{F}_p$ es una *aproximación* de $u \in \mathbb{F}_p$ con precisión Δ si $u - w \in \{-\Delta, 1-\Delta, \ldots, \Delta\}$.

A continuación exponemos el planteamiento común a este tipo de algoritmos. Se considera un generador de números pseudoaleatorios que produce recursivamente una

sucesión $u_0, u_1, \ldots$ de elementos de $\mathbb{F}_p$ a partir de la *semilla* u_0 y de unos parámetros (por ejemplo, en un generador polinómico, los coeficientes del polinomio que se evalúa en cada valor para obtener el siguiente). Suponemos que un atacante externo tiene acceso a aproximaciones w_i de varios valores consecutivos de la sucesión. De este modo, se tiene $u_i = w_i + \varepsilon_i$, donde los errores de aproxiamación ε_i tienen un valor absoluto acotado por la tolerancia Δ que se permite. Expresaremos este parámetro como potencia del cardinal de $\mathbb{F}_p$. Así, $\Delta = p^\delta$ indica que el atacante, por conocer la aproximación w_i, tiene acceso a $(1-\delta)\log_2(u_i)$ bits del valor buscado u_i (ver Ilustración 2.3). Se construye un algoritmo que a partir de las aproximaciones w_i, y en algunos casos, de todos o parte de los parámetros del generador, puede recuperar los errores de aproximación ε_i y por lo tanto, los valores exactos u_i a partir de los cuales se puede reproducir la sucesión. Se demuestra que el algoritmo funciona correctamente siempre que el primer elemento aproximado u_0 no pertenezca a cierto conjunto de valores "malos", cuyo cardinal se acota, pongamos por $O(\Delta^t)$. Suponiendo que los elementos producidos por el generador siguen una distribución uniforme en $\mathbb{F}_p$, podemos acotar la probabiliad de fallo del algoritmo propuesto por $O(p^{\delta t-1})$. Así pues, si se admite una tolerancia $\Delta \geq p^{1/t}$, este razonamiento no permite asegurar que el algoritmo vaya a predecir la sucesión en ningún caso, y de hecho sugiere que va a fallar siempre. Por otra parte, cuando δ es significativamente menor que $1/t$, la probabilidad de éxito es grande. Mediante una implementación de los algoritmos se puede confirmar este límite teórico para la proporción de los bits ocultos que permiten recuperar.

En la primera parte del capítulo se estudia el generador cuadrático definido recursivamente por:

$$u_{n+1} \equiv a\, u_n^2 + c \bmod p. \tag{6.1}$$

En [Blackburn et al., 2005] se exponen algoritmos como los que planteamos, con la diferencia de que requieren excluir un cierto conjunto de parámetros para los cuales no se garantiza que el algoritmo devuelva la solución deseada.

En un primer apartado, se analiza este generador cuadrático en el caso en el que atacante tiene acceso a los dos coeficientes a y c del polinomio. En [Blackburn et al., 2005] se propone un método para este contexto que utiliza dos aproximaciones. El tamaño del conjunto de valores "malos" se acota por $O(\Delta^4)$, pero además se requiere excluir los generadores cuyo parámetro a, llamado *multiplicador*, pertenece a otro conjunto excepcional de tamaño $O(\Delta^3)$. El método desarrollado en el Apartado 2.1.1 (ver Teorema 7) permite la eliminación de ese segundo conjunto excepcional para la prueba de la corrección del algoritmo. Una versión preliminar se ha publicado en [Gómez et al., 2005b].

Teorema 7 *Sean p un número primo y Δ un entero positivo. Para parámetros cualesquiera $a \in \mathbb{F}_p^*$ y $c \in \mathbb{F}_p$, existe un conjunto $\mathcal{U}(\Delta; a) \subseteq \mathbb{F}_p$ de cardinal $\#\mathcal{U}(\Delta; a) = O(\Delta^4)$ con la propiedad siguiente. Existe un algoritmo que, tomando como entrada a, c y aproximaciones w_0, w_1 con precisión Δ de dos valores consecutivos u_0, u_1 producidos por el generador cuadrático (6.1) tales que $u_0 \notin \mathcal{U}(\Delta; a)$, devuelve el valor u_0 en tiempo polinómico.*

En el Apartado 2.1.2 se desarrolla un algoritmo similar para el caso en que se conoce el multiplicador a, mientras que el *desplazamiento* c permanece oculto. Este algoritmo,

que esencialmente coincide con el presentado en [Gómez et al., 2005b], requiere de tres aproximaciones w_0, w_1, w_2 y cada una de ellas debe proporcionar 4/5 de los bits del valor que se aproxima.

Teorema 8 *Sean p un número primo y Δ un entero positivo. Para parámetros cualesquiera $a \in \mathbb{F}_p^*$ y $c \in \mathbb{F}_p$, existe un conjunto $\mathcal{U}(\Delta; a, c) \subseteq \mathbb{F}_p$ de cardinal $\#\mathcal{U}(\Delta; a, c) = O(\Delta^5)$ con la propiedad siguiente. Existe un algoritmo que, tomando como entrada a y aproximaciones w_0, w_1, w_2 con precisión Δ de tres valores consecutivos u_0, u_1, u_2 producidos por el generador cuadrático (6.1) tales que $u_0 \notin \mathcal{U}(\Delta; a, c)$, devuelve el valor u_0 y el parámetro c en tiempo polinómico.*

Concluyendo la sección dedicada al generador cuadrático, el Apartado 2.1.3 estudia el llamado *generador de Pollard*, caso particular del generador cuadrático considerado hasta ahora en el que el polinomio es mónico. Considerando el parámetro c conocido, el algoritmo de [Blackburn et al., 2005] funciona para tres aproximaciones excluyendo un conjunto de cardinal $O(\Delta^4)$. En esta memoria exponemos un algoritmo similar (publicado en [Gómez et al., 2006]) que ejecuta dos reducciones de bases de retículas, mejorando sensiblemente el resultado: se pasa de una tolerancia máxima de $p^{1/3}$ a $p^{5/14}$.

Teorema 9 *Sean p un número primo y Δ un entero positivo. Para cualquier parámetro $c \in \mathbb{F}_p$, existe un conjunto $\mathcal{V}(\Delta, c) \subseteq \mathbb{F}_p \times [-\Delta, \Delta]$, cuyo cardinal satisface*

$$\#\mathcal{V}(\Delta, c) = O\left(\max\{\Delta^{15} p^{-4}, \Delta^{19/5}\}\right),$$

con la propiedad siguiente. Existe un algoritmo que, tomando como entrada aproximaciones w_0, w_1 con precisión Δ de dos valores consecutivos u_0, u_1 producidos por el generador de Pollard tales que $(u_0, u_0 - w_0) \notin \mathcal{V}(\Delta, c)$, devuelve el valor u_0 en tiempo polinómico.

En el Apartado 2.2 aplicamos el método de reducción de bases de retículas a la predicción del generador lineal de puntos pseudoaleatorios sobre curvas curvas elípticas. Este generador, propuesto en [Hallgren, 1994], es un ejemplo de cómo el grupo abeliano $(E, \oplus)$ de una curva elíptica proporciona aplicaciones en muchos aspectos de la Criptología.

Para definir este generador, consideramos una curva elíptica sobre un cuerpo finito primo $\mathbb{F}_p$ de característica distinta de 2 ó 3:

$$E: \quad Y^2 = X^3 + aX + b, \quad \text{con } 4a^3 + 27b^2 \not\equiv_p 0.$$

El generador obtiene recursivamente puntos en este curva a partir de una semilla U_0 según la fórmula:

$$U_{n+1} = U_n \oplus G, \tag{6.2}$$

donde G es un punto de la curva que actúa como parámetro.

Consideramos la situación en la que un observador externo tiene acceso a este parámetro G, junto con aproximaciones (componente a componente) de dos puntos consecutivos producidos por el generador. Es poco probable que la abscisa de G coincida con la de uno de los puntos U_i que se aproximan. Podemos trasladar entonces la

ecuación $U_1 = U_0 \oplus G$ en dos identidades entre las componentes de estos tres puntos, y desarrollar un método análogo a los expuestos para otros generadores.

Se obtiene un resultado de predicción cuando la abscisa del primero de los puntos x_0 está fuera de un conjunto excepcional de tamaño $O(\Delta^6)$. Los experimentos realizados confirman este límite teórico. En este caso, Δ representa la tolerancia al error de aproximación de cada una de las componentes. En otras palabras, decimos que $W = (x_W, y_W) \in \mathbb{F}_p^2$ es una *aproximación* de $U = (x_U, y_U)$ con precisión Δ si las dos diferencias $x_U - x_W$, $y_U - y_W$ están en el rango $\{-\Delta, 1-\Delta, \ldots, \Delta\}$.

Teorema 10 *Sean p un número primo mayor que 3 y $a, b \in \mathbb{F}_p$ tales que $4a^3+27b^2 \neq 0$, y sea $G = (x_G, y_G)$ un punto afín de la curva elíptica $E(\mathbb{F}_p)$ definida por el polinomio $Y^2 - (X^3 + aX + b)$. Existe un conjunto $\mathcal{U}(\Delta; a, x_G, y_G) \subseteq \mathbb{F}_p$ de cardinal $\#\mathcal{U}(\Delta; a, x_G, y_G) = O(\Delta^6)$ con la propiedad siguiente. Existe un algoritmo que, tomando como entrada el parámetro $G = (x_G, y_G)$ y dos aproximaciones W_0,W_1 con precisión Δ de dos puntos afines consecutivos $U_0 = (x_0, y_0)$, U_1 producidos por el generador lineal (6.2) tales que $x_0 \notin \mathcal{U}(\Delta; a, x_G, y_G)$, devuelve el valor U_0 en tiempo polinómico.*

Por otra parte, en el caso de que el parámetro G permanezca oculto, hemos propuesto un método distinto para abordar la predicción del generador. Partirmos de tres aproximaciones a tres puntos producidos por el generador, pero únicamente tenemos en cuenta la pertenencia de estos puntos a la curva elíptica para desarrollar el método de reducción de retículas. De este modo, obviamos la información que pudiéramos obtener a partir del hecho de que estos puntos se han producido por un generador lineal. Por lo tanto, no es necesario que los valores aproximados sean consecutivos. La cota teórica que obtenemos para la tolerancia es $\Delta = O(p^{1/46})$.

3 Diagramas de mínima distancia en digrafos de Cayley

Los grafos circulantes proporcionan una solución sencilla para el diseño de redes de comunicación y de algunos dispositivos utilizados en Arquitectura de Computadores [Stone, 1970]. En esta memoria, recogiendo los resultados publicados en [Gómez et al., 2005a, 2007a,b], nos restringimos al estudio de los grafos circulantes dirigidos, también llamados digrafos circulantes.

Dados r elementos distintos en $\mathbb{Z}/N\mathbb{Z}$, se define $C_N(j_1, \ldots, j_r)$ como el grafo dirigido cuyos N nodos vienen etiquetados por las clases de restos módulo N y cuyo conjunto de arcos es

$$A := \{(g, g + j_i) \in (\mathbb{Z}/N\mathbb{Z})^2 \mid g \in \mathbb{Z}/N\mathbb{Z},\ i = 1, \ldots, r\}.$$

De este modo, $C_N(j_1, \ldots, j_r)$ es un grafo regular de grado r y cuenta con Nr arcos que pueden clasificarse en r tipos o "colores". Estos grafos son un caso particular de los llamados *digrafos de Cayley* en los que el grupo de vértices es cíclico.

Definición 11 *Sean Γ un grupo y $S \subseteq \Gamma$ un subconjunto. Se define el digrafo de Cayley $C(\Gamma, S)$ como el grafo dirigido cuyos conjuntos de vértices y arcos son, respectivamente, $V = \Gamma$ y $A = \{(g, h) \in \Gamma^2 \mid g^{-1}h \in S\}$.*

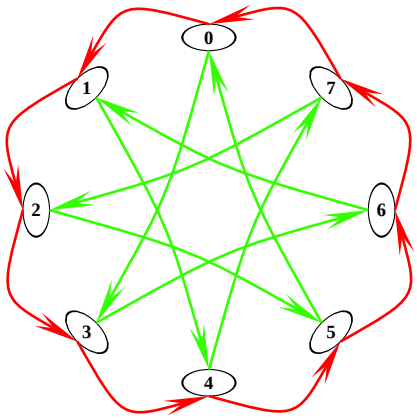


Ilustración 1: $C_8(1,3)$ tiene arcos de dos tipos.

Consideramos el problema del *encaminamiento* en un digrafo de Cayley: a partir de un par de nodos del grafo, se trata de buscar un camino que los comunique con la mínima cantidad posible de arcos. Los grafos de este tipo son transitivos con respecto a los vértices, es decir, para todo par de vértices $(g,h) \in \Gamma^2$, existe una permutación $\sigma \in \Sigma_\Gamma$ tal que $\sigma(g) = h$ e $(i,j) \in A \iff (\sigma(i), \sigma(j)) \in A$. Esta propiedad significa que cada vértice del grafo es indistinguible del resto. Por lo tanto, podemos reducir el problema del encaminamiento al caso en que el nodo de origen es el elemento neutro de Γ.

De aquí en adelante nos restringiremos al caso en que Γ es un grupo finito y abeliano. Por lo tanto, S también es finito y podemos enumerar sus elementos: $\{s_1, \ldots, s_r\}$. El ser Γ abeliano provoca que, en lo que concierne al encaminamiento, podamos identificar entre sí los caminos que constan de igual número de arcos de cada color, independientemente del orden en que se recorran. Por lo tanto, podemos representar un camino mediante un vector de r números naturales cuya componente i-ésima representa la cantidad de arcos de la forma $(g, g+s_i)$ que componen el camino. Con esta notación, construimos la función de encaminamiento:

$$\begin{array}{rccc} R: & \mathbb{N}^r & \longrightarrow & \Gamma \\ & \mathbf{a} & \mapsto & a_1 s_1 + \cdots + a_r s_r \end{array} \tag{6.3}$$

que asocia a cada camino el nodo con que conecta al nodo 0_Γ.

En [Wong and Coppersmith, 1974] se introdujo un método para representar un encamiento óptimo para para pareja de vértices en un digrafo circulante conexo de grado dos, mediante un diagrama llamado "de mínima distancia". Según se observa en la Ilustración 2, un diagrama de este tipo representa, para cada nodo del grafo $g \in \Gamma$, un camino óptimo (con la menor cantidad posible de arcos) $D(g) \in \mathbb{N}^r$ que enlaza los nodos 0_Γ y g.

Según se explica en el Capítulo III, dado un orden monomial graduado $\prec$, se puede asociar un diagrama de mínima distancia a cada digrafo de Cayley conexo mediante la elección:

$$D(g) := \min_{\prec}(R^{-1}(g)). \tag{6.4}$$

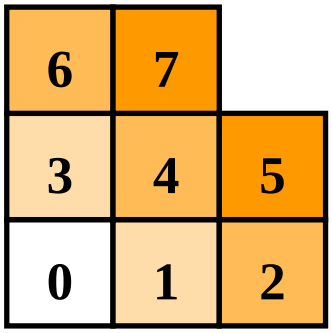


Ilustración 2: Diagrama de mínima distancia para $C_8(1,3)$.

Los diagramas así obtenidos pueden ser representados mediante un ideal monomial, puesto que su complementario es un ideal en el semigrupo $\mathbb{N}^r$.

Definición 12 *Sean Γ un grupo finito y abeliano, $S = \{s_1, \ldots, s_r\} \subseteq G$ un subconjunto generador y $\prec$ un orden monomial graduado en $\mathbb{N}^r$. Se define el ideal monomial asociado como:*

$$I(C(\Gamma, S), \prec) := (\mathbb{N}^r \backslash D(\Gamma)),$$

siendo D la aplicación definida por la Ecuación (6.4).

En el caso bidimensional, el sistema minimal de generadores de cualquiera de estos ideales consta de dos o de tres monomios. Por ejemplo, el ideal correspondiente a la Ilustración 2 es (x^3, x^2y^2, y^3). Independientemente de la dimensión (el grado del grafo), a partir del sistema minimal de generadores de un ideal monomial se puede computar la descomposición irredundante en ideales monomiales irreducibles, $(x^3, y^2) \cap (x^2, y)$ en el ejemplo anterior. Esta descomposición permite el cálculo del diámetro y la distancia media del grafo, según el resultado siguiente:

Proposición 13 *Sean Γ un grupo finito y abeliano, y $S = \{s_1, \ldots, s_r\} \subseteq \Gamma$ un subconjunto generador. Para cualquier orden monomial graduado $\prec$ en $\mathbb{N}^r$, dada la descomposición irredundante del ideal monomial asociado por ideales monomiales irreducibles:*

$$I(C(\Gamma, S), \prec) = \mathfrak{m}^{\mathbf{b}_1} \cap \cdots \mathfrak{m}^{\mathbf{b}_f},$$

las fórmulas siguientes calculan el diámetro y la distancia media del grafo $C(\Gamma, S)$.

$$d = \max\{\|\mathbf{b}_i\|_1 - r \mid i = 1, \ldots, f\},$$

$$\bar{d} = \frac{1}{N} \sum_{\emptyset \subsetneq \Delta \subseteq \{1,\ldots,f\}} (-1)^{\#\Delta+1} \sigma(\mathbf{d}_\Delta),$$

siendo $\mathbf{d}_\Delta$ el vector cuya componente i-ésima coincide con el grado en X_i del monomio $\operatorname{mcd}\{\mathbf{x}^{\mathbf{b}_i} \mid i \in \Delta\}$ y $\sigma(\mathbf{d}) := d_1 \cdots d_r(d_1 + \cdots + d_r - r)/2$.

Al generalizar esta construcción en el caso general de un digrafo de grado arbitrario r, encontramos la dificultad de acotar el número de elementos del sistema generador minimal del ideal monomial. Se puede demostrar que existe a lo sumo un generador que involucre todas las variables, como es el caso de x^2y^2 en nuestro ejemplo. Por otro lado, existen r generadores que son una potencia pura de una variable, pero la cantidad de generadores que involucran una cantidad de variables intermedia entre 1 y r crece de modo que no somos capaces de dar un algoritmo eficaz para el cálculo del diámetro y la distancia media.

En todo caso, podemos aprovechar el concepto de ideal reticular (ver Secciones 1 y 1.3) para obtener esa lista de generadores usando la teoría de las bases de Gröbner. Para ello, basta comprobar que el ideal monomial que hemos asociado a un digrafo conexo y un orden monomial graduado coincide con el ideal inicial del ideal binomial I_Λ, donde Λ es el núcleo de la aplicación de encaminamiento extendida a $\mathbb{Z}^r$:

$$\begin{array}{rccc} \bar{R}: & \mathbb{Z}^r & \longrightarrow & \Gamma \\ & \mathbf{a} & \mapsto & a_1 s_1 + \cdots + a_r s_r. \end{array}$$

De este modo, como es inmediato obtener una base de la retícula $\Lambda = \ker(\bar{R})$ en el caso de digrafos circulates, obtenemos un procedimiento para el cálculo de su diámetro y distancia media mediante el cálculo de la base de Gröbner del ideal reticular correspondiente.

Una base de Gröbner de I_Λ también es útil para resolver el problema del encaminamiento. Obtenemos:

Proposición 14 *Sea Γ un grupo finito y abeliano, $S = \{s_1, \ldots, s_r\} \subseteq \Gamma$ un subconjunto generador y Λ la retícula entera asociada. Sean G una base de Gröbner de I_Λ con respecto al orden monomial graduado $\prec$ y $\mathbf{c} \in \mathbb{N}^r$ un camino en $R^{-1}(i)$. En estas condiciones, la forma normal de $\mathbf{x}^{\mathbf{c}} - 1$ con respecto a G es $\mathbf{x}^{\mathbf{d}} - 1$, donde $\mathbf{d}$ es el camino mínimo con respecto al orden $\prec$ que tiene origen 0 y destino i.*

En general, el problema del encaminamiento se reduce a un problema de programación entera, una vez que se dispone de la retícula asociada $\Lambda = \ker(R)$. En la Sección 3.4 hacemos explícita esta reducción y la Sección 3.4 damos un algoritmo específico para el caso bidimensional. Este algoritmo, publicado en [Gómez et al., 2007b], estudia un versión del algoritmo de Gauss para la norma ℓ_1. Se analiza el coste del cálculo de un camino de longitud mínima en los casos dirigido, no dirigido, y para grafos con pesos, en los que es diferente el coste de recorrer saltos de colores distintos.

Concluimos el capítulo definiendo una familia de digrafos circulantes cuyos miembros tienen la propiedad de que su ideal monomial asociado con respecto a cualquier orden monomial graduado es irreducible. Se obtienen de esta manera redes de grado r arbitrario que conectan N vértices y cuyo diámetro es aproximadamente $N^{1/r}$.

Teorema 15 *Sean d y r dos enteros positivos, y sea k el resto de la división euclídea de d por r. Entonces, fijando:*

$$\alpha_1 = \cdots = \alpha_k = \frac{d-k}{r} + 2,\ \alpha_{k+1} = \cdots = \alpha_r = \frac{d-k}{r} + 1,$$

el siguiente es un digrafo circulante de grado r, $N := \alpha_1 \cdots \alpha_r$ nodos y diámetro d:

$$\mathcal{C}_N(1, \alpha_1, \alpha_1\alpha_2, \ldots, \alpha_1 \cdots \alpha_{r-1}).$$

4 Software

En el último capítulo de la memoria reflejamos las tareas de implementación que hemos realizado como complemento a la investigación en los dos problemas estudiados.

En primer lugar, exponemos la implementación en C++ de los algoritmos presentados en el Capítulo II. Hemos utilizado la librería NTL de V. Shoup que permite el tratamiento de números enteros de talla arbitraria e incluye el algoritmo LLL para reducción de bases de retículas. Todos los algoritmos que implementamos hacen una llamada (o dos) al problema del vector más próximo (CVP) para calcular una solución de norma mínima en un sistema lineal de congruencias no homogéneo. Al tratar con sistemas con un número pequeño de variables (22 en el caso peor), este problema puede resolverse eficientemente. Hemos recurrido al método descrito en [Agrell et al., 2002], que utiliza como paso previo la reducción LLL.

Recogemos los resultados experimentales que hemos obtenido aplicando los algoritmos para distintas tolerancias Δ. De este modo, podemos confirmar empíricamente las cotas teóricas deducidas para algunos casos; mientras que en otros, se observa como la cota predicha no es óptima y el algoritmo correspondiente recupera correctamente errores de aproximación mayores de lo esperado. Debido a la naturaleza asintótica de las cotas que consideramos, es necesario emplear generadores sobre un cuerpo $\mathbb{F}_p$ de tamaño suficientemente grande para observar el comportamiento que comentamos.

Por último, presentamos el programa Circule. Esta aplicación permite dibujar de forma sencilla grafos circulantes, modificando la curvatura de los arcos, su color, etc. También se obtiene el dibujo del diagrama asociado a un digafo circulante de grado no superior a tres, con respecto a cualquier orden monomial (graduado o no).

5 Trabajo futuro

En lo que respecta a los algoritmos descritos en el Capítulo II, llama la atención la necesidad de acotar, para cada una de las situaciones que se consideran, el tamaño del llamado conjunto malo de valores iniciales. Sería interesante poder profundizar en estos métodos de forma que pudieran obtenerse resultados aplicables a situaciones más generales.

Por otra parte, parece que la técnica de la reducción doble que se detalla en el Apartado 2.1.3 podría aplicarse en la predicción de más generadores e incluso de otras aplicaciones de la reducción de bases de retículas. El principal incoveniente de esta técnica es el aumento de la dimensión de las retículas con las que se opera. Además, el resultado teórico que obtenemos se mejora significativamente en los experimentos realizados. Esto sugiere que seguramente sea posible mejorar la acotación del conjunto excepcional de parámetros que hemos hecho. Como continuación natural, surge la idea de un algoritmo de predicción que realice múltiples iteraciones. Para la concreción de esta idea, sería necesario un tratamiento más sistematizado que el que hemos llevado a cabo en los casos particulares que tratamos en esta memoria.

En todos nuestros algoritmos, el tamaño p del espacio base se asume conocido por el atacante. En [Joux and Stern, 1998] se propone un método heurístico para la predicción del generador lineal con el primo p oculto. Sería interesante desarrollar métodos similares para otros generadores (no lineales).

En lo que respecta al generador lineal sobre curvas elípticas, es de esperar que exista un algoritmo que no requiera el conocimiento del parámetro G pero presente un comportamiento mejor que el que hemos propuesto en el Apartado 2.2.4, que no aprovecha el hecho de que los puntos para los que contamos con aproximaciones se han obtenido mediante un generador concreto. De modo más general, podrían buscarse aplicaciones del método de reducción de bases de retículas a otros generadores sobre curvas elípticas.

En el Capítulo III hemos utilizado los ideales monomiales como una herramienta básica para el estudio de los digrafos de Cayley cuyo conjunto de vértices es un finito y abeliano. Es posible que este punto de vista sea útil en otras custiones relativas a grafos circulantes de grado arbitrario. Por ejemplo, el problema de decidir si un digrafo circulante contiene un circuito de Hamilton [Fiol and Yebra, 1988] está resuelto para digrafos de grado dos pero permanece como problema abierto en el caso general. Este problema tiene una conexión curiosa con un problema relativo a la combinatoria de un carillón [Rankin, 1948].

Por último, no hemos tratado en esta memoria aspectos concernientes a la búsqueda de caminos con tolerancia a fallos en algunos arcos de un grafo circulante. Podría analizarse el comportamiento de la familia de digrafos circulantes que hemos propuesto frente a estos problemas.

Bibliography

4ti2 team. 4ti2—a software package for algebraic, geometric and combinatorial problems on linear spaces. Available at www.4ti2.de.

E. Agrell, T. Eriksson, A. Vardy, and K. Zeger. Closest point search in lattices. *IEEE Trans. Inform. Theory*, 48(8):2201–2214, 2002.

M. Ajtai. Generating hard instances of lattice problems (extended abstract). In *Proceedings of the Twenty-eighth Annual ACM Symposium on the Theory of Computing (Philadelphia, PA, 1996)*, pages 99–108, New York, 1996. ACM.

M. Ajtai. The shortest vector problem in l_2 is NP-hard for randomized reductions (extended abstract). In *STOC*, pages 10–19, 1998.

L. Babai. On Lovász' lattice reduction and the nearest lattice point problem. *Combinatorica*, 6(1):1–13, 1986.

T. Becker and V. Weispfenning. *Gröbner bases*, volume 141 of *Graduate Texts in Mathematics*. Springer-Verlag, New York, 1993. A computational approach to commutative algebra, In cooperation with Heinz Kredel.

P. H. T. Beelen and J. M. Doumen. Pseudorandom sequences from elliptic curves. In *Finite fields with applications to coding theory, cryptography and related areas (Oaxaca, 2001)*, pages 37–52. Springer, Berlin, 2002.

J.-C. Bermond, F. Comellas, and D. F. Hsu. Distributed loop computer networks: A survey. *J. Parallel Distrib. Comput.*, 24(1):2–10, 1995.

S. R. Blackburn, D. Gómez, J. Gutierrez, and I. E. Shparlinski. Predicting the inversive generator. In *Cryptography and coding*, volume 2898 of *Lecture Notes in Comput. Sci.*, pages 264–275. Springer, Berlin, 2003.

S. R. Blackburn, D. Gómez, J. Gutierrez, and I. E. Shparlinski. Predicting nonlinear pseudorandom number generators. *Math. Comp.*, 74(251):1471–1494 (electronic), 2005.

S. R. Blackburn, D. Gómez, J. Gutierrez, and I. E. Shparlinski. Reconstructing noisy polynomial evaluation in residue rings. *J. Algorithms*, 61(2):47–59, 2006.

I. F. Blake, G. Seroussi, and N. P. Smart. *Elliptic curves in cryptography*, volume 265 of *London Mathematical Society Lecture Note Series*. Cambridge University Press, Cambridge, 2000. Reprint of the 1999 original.

H. F. Blichfeldt. A new principle in the geometry of numbers, with some applications. *Trans. Amer. Math. Soc.*, 15(3):227–235, 1914.

F. Boesch and R. Tindell. Circulants and their connectivities. *J. Graph Theory*, 8(4): 487–499, 1984.

J. Boyar. Inferring sequences produced by pseudo-random number generators. *J. Assoc. Comput. Mach.*, 36(1):129–141, 1989a.

J. Boyar. Inferring sequences produced by a linear congruential generator missing low-order bits. *J. Cryptology*, 1(3):177–184, 1989b.

E. F. Brickell and A. M. Odlyzko. Cryptanalysis: a survey of recent results. In *Contemporary cryptology*, pages 501–540. IEEE, New York, 1992.

B. Buchberger. *An algorithm for findind the bases elements of the residue class ring modulo a zero dimensional polynomial ideal*. PhD thesis, University of Innsbruck, 1965.

J.-Y. Cai, G. Havas, B. Mans, A. Nerurkar, J.-P. Seifert, and I. Shparlinski. On routing in circulant graphs. In *Computing and combinatorics (Tokyo, 1999)*, volume 1627 of *Lecture Notes in Comput. Sci.*, pages 360–369. Springer, Berlin, 1999.

J. W. S. Cassels. *An introduction to the geometry of numbers*. Classics in Mathematics. Springer-Verlag, Berlin, 1997. Corrected reprint of the 1971 edition.

Y. Cheng and F. K. Hwang. Diameters of weighted double loop networks. *J. Algorithms*, 9(3):401–410, 1988.

Y. Cheng, F. K. Hwang, I. F. Akyildiz, and D. F. Hsu. Routing algorithms for double loop networks. *Internat. J. Found. Comput. Sci.*, 3(3):323–331, 1992.

H. Cohen, G. Frey, R. Avanzi, C. Doche, T. Lange, K. Nguyen, and F. Vercauter. *Handbook of elliptic and hyperelliptic curve cryptography*. Discrete Mathematics and its Applications (Boca Raton). Chapman & Hall/CRC, Boca Raton, FL, 2005.

D. Coppersmith. Finding a small root of a bivariate integer equation; factoring with high bits known. In *Advances in cryptology—EUROCRYPT '96 (Saragossa, 1996)*, volume 1070 of *Lecture Notes in Comput. Sci.*, pages 178–189. Springer, Berlin, 1996.

D. Coppersmith. Small solutions to polynomial equations, and low exponent RSA vulnerabilities. *J. Cryptology*, 10(4):233–260, 1997.

F. Eisenbrand. Fast integer programming in fixed dimension. In *Algorithms—ESA 2003*, volume 2832 of *Lecture Notes in Comput. Sci.*, pages 196–207. Springer, Berlin, 2003.

D. Eisenbud and B. Sturmfels. Binomial ideals. *Duke Math. J.*, 84(1):1–45, 1996.

E. El Mahassni and I. E. Shparlinski. On the uniformity of distribution of congruential generators over elliptic curves. In *Sequences and their applications (Bergen, 2001)*, Discrete Math. Theor. Comput. Sci. (Lond.), pages 257–264. Springer, London, 2002.

P. Erdős and D. F. Hsu. Distributed loop network with minimum transmission delay. *Theoret. Comput. Sci.*, 100(1):223–241, 1992.

M. Espona and O. Serra. Cayley digraphs based on the de Bruijn networks. *SIAM J. Discrete Math.*, 11(2):305–317 (electronic), 1998.

M. A. Fiol and J. L. A. Yebra. Ciclos de hamilton en redes de paso conmutativo y de paso fijo. *Stochastica*, 12(2-3):113–129, 1988.

M. A. Fiol, J. L. A. Yebra, I. Alegre, and M. Valero. A discrete optimization problem in local networks and data alignment. *IEEE Trans. Comput.*, 36(6):702–713, 1987.

A. M. Frieze, J. Håstad, R. Kannan, J. C. Lagarias, and A. Shamir. Reconstructing truncated integer variables satisfying linear congruences. *SIAM J. Comput.*, 17(2): 262–280, 1988. Special issue on cryptography.

C. F. Gauss. *Disquisitiones arithmeticae*. Translated into English by Arthur A. Clarke, S. J. Yale University Press, New Haven, Conn., 1966.

D. Gómez. *Cryptanalysis of nonlinear pseudorandom number generators*. PhD thesis, Universidad de Cantabria, 2006.

D. Gómez, J. Gutierrez, and A. Ibeas. Circulant digraphs and monomial ideals. In *Computer algebra in scientific computing*, volume 3718 of *Lecture Notes in Comput. Sci.*, pages 196–207. Springer, Berlin, 2005a.

D. Gómez, J. Gutierrez, and A. Ibeas. Cryptanalysis of the quadratic generator. In *Progress in cryptology—INDOCRYPT 2005*, volume 3797 of *Lecture Notes in Comput. Sci.*, pages 118–129. Springer, Berlin, 2005b.

D. Gómez, J. Gutierrez, A. Ibeas, C. Martínez, and R. Beivide. On finding a shortest path in circulant graphs with two jumps. In *Computing and combinatorics*, volume 3595 of *Lecture Notes in Comput. Sci.*, pages 777–786. Springer, Berlin, 2005c.

D. Gómez, J. Gutierrez, and A. Ibeas. Attacking the Pollard generator. *IEEE Trans. Inform. Theory*, 52(12):5518–5523, 2006.

D. Gómez, J. Gutierrez, and A. Ibeas. Cayley digraphs of finite abelian groups and monomial ideals. *SIAM J. Discrete Math.*, 21(3):763–784 (electronic), 2007a.

D. Gómez, J. Gutierrez, and A. Ibeas. Optimal routing in double loop networks. *Theoret. Comput. Sci.*, 381(1-3):68–85, 2007b.

G. Gong and C. C. Y. Lam. Linear recursive sequences over elliptic curves. In *Sequences and their applications (Bergen, 2001)*, Discrete Math. Theor. Comput. Sci. (Lond.), pages 182–196. Springer, London, 2002.

G. Gong, T. A. Berson, and D. R. Stinson. Elliptic curve pseudorandom sequence generators. In *Selected areas in cryptography (Kingston, ON, 1999)*, volume 1758 of *Lecture Notes in Comput. Sci.*, pages 34–48. Springer, Berlin, 2000.

M. Grötschel, L. Lovász, and A. Schrijver. *Geometric algorithms and combinatorial optimization*, volume 2 of *Algorithms and Combinatorics*. Springer-Verlag, Berlin, second edition, 1993.

P. M. Gruber and C. G. Lekkerkerker. *Geometry of numbers*, volume 37 of *North-Holland Mathematical Library*. North-Holland Publishing Co., Amsterdam, second edition, 1987.

D. J. Guan. An optimal message routing algorithm for double-loop networks. *Inform. Process. Lett.*, 65(5):255–260, 1998.

J. Gutierrez and A. Ibeas. Inferring sequences produced by a linear congruential generator on elliptic curves missing high-order bits. *Des. Codes Cryptogr.*, 45(2):199–212, 2007.

J. Gutierrez and R. Rubio San Miguel. Reduced Gröbner bases under composition. *J. Symbolic Comput.*, 26(4):433–444, 1998.

S. Hallgren. Linear congruential generators over elliptic curves. Preprint CS-94-143, Dept of Comp. Sci., Cornegie Mellon Univ., 1994.

F. Hess and I. E. Shparlinski. On the linear complexity and multidimensional distribution of congruential generators over elliptic curves. *Des. Codes Cryptogr.*, 35(1): 111–117, 2005.

N. Howgrave-Graham. Finding small roots of univariate modular equations revisited. In *Cryptography and coding (Cirencester, 1997)*, volume 1355 of *Lecture Notes in Comput. Sci.*, pages 131–142. Springer, Berlin, 1997.

D. F. Hsu and X. D. Jia. Extremal problems in the construction of distributed loop networks. *SIAM J. Discrete Math.*, 7(1):57–71, 1994.

F. K. Hwang. A complementary survey on double-loop networks. *Theoret. Comput. Sci.*, 263(1-2):211–229, 2001. Combinatorics and computer science (Palaiseau, 1997).

F. K. Hwang. A survey on multi-loop networks. *Theoret. Comput. Sci.*, 299(1-3): 107–121, 2003.

A. Ibeas. Circule. Available at `http://grupos.unican.es/amac/circule`.

A. Joux and J. Stern. Lattice reduction: a toolbox for the cryptanalyst. *J. Cryptology*, 11(3):161–185, 1998.

M. Kaib and C. P. Schnorr. The generalized Gauss reduction algorithm. *J. Algorithms*, 21(3):565–578, 1996.

R. Kannan. Improved algorithms for integer programming and related lattice problems. In *STOC*, pages 193–206, 1983.

R. Kannan. Minkowski's convex body theorem and integer programming. *Math. Oper. Res.*, 12(3):415–440, 1987.

D. E. Knuth. *The art of computer programming. Vol. 2.* Addison-Wesley Publishing Co., Reading, Mass., second edition, 1981. Seminumerical algorithms, Addison-Wesley Series in Computer Science and Information Processing.

D. E. Knuth. Deciphering a linear congruential encryption. *IEEE Trans. Inform. Theory*, 31(1):49–52, 1985.

A. Korkine and G. Zolotareff. Sur les formes quadratiques. *Math. Ann.*, 6(3):366–389, 1873.

H. Krawczyk. How to predict congruential generators. *J. Algorithms*, 13(4):527–545, 1992.

J. C. Lagarias. Pseudorandom number generators in cryptography and number theory. In *Cryptology and computational number theory (Boulder, CO, 1989)*, volume 42 of *Proc. Sympos. Appl. Math.*, pages 115–143. Amer. Math. Soc., Providence, RI, 1990.

D. H. Lehmer. Mathematical methods in large-scale computing units. In *Proceedings of a Second Symposium on Large-Scale Digital Calculating Machinery, 1949*, pages 141–146, Cambridge, Mass., 1951. Harvard University Press.

A. K. Lenstra, H. W. Lenstra, Jr., and L. Lovász. Factoring polynomials with rational coefficients. *Math. Ann.*, 261(4):515–534, 1982.

H. W. Lenstra, Jr. Integer programming with a fixed number of variables. *Math. Oper. Res.*, 8(4):538–548, 1983.

H. W. Lenstra, Jr. Factoring integers with elliptic curves. *Annals of Math.*, 126: 649–673, 1987.

L. Lovász and H. E. Scarf. The generalized basis reduction algorithm. *Math. Oper. Res.*, 17(3):751–764, 1992.

A. May. *New RSA vulnerabilities using lattice reduction methods.* PhD thesis, University of Paderborn, 2003.

D. Micciancio and S. Goldwasser. *Complexity of lattice problems.* The Kluwer International Series in Engineering and Computer Science, 671. Kluwer Academic Publishers, Boston, MA, 2002. A cryptographic perspective.

E. Miller. *Resolutions and Duality for Monomial Ideals.* PhD thesis, University of California, Berkeley, 2000.

E. Miller and B. Sturmfels. Monomial ideals and planar graphs. In *Applied algebra, algebraic algorithms and error-correcting codes (Honolulu, HI, 1999)*, volume 1719 of *Lecture Notes in Comput. Sci.*, pages 19–28. Springer, Berlin, 1999.

H. Minkowski. *Geometrie der Zahlen.* Bibliotheca Mathematica Teubneriana, Band 40. Johnson Reprint Corp., New York, 1968.

P. Q. Nguyen and D. Stehlé. Low-dimensional lattice basis reduction revisited. In *Algorithmic number theory*, volume 3076 of *Lecture Notes in Comput. Sci.*, pages 338–357. Springer, Berlin, 2004.

P. Q. Nguyen and J. Stern. The two faces of lattices in cryptology. In *Cryptography and lattices (Providence, RI, 2001)*, volume 2146 of *Lecture Notes in Comput. Sci.*, pages 146–180. Springer, Berlin, 2001.

H. Niederreiter. Design and analysis of nonlinear pseudorandom number generators. In G. I. Schuëller and P. D. Spanos, editors, *Monte Carlo Simulation*, pages 3–9, Rotterdam, 2001. A.A. Balkema.

H. Niederreiter. New developments in uniform pseudorandom number and vector generation. In *Monte Carlo and quasi-Monte Carlo methods in scientific computing (Las Vegas, NV, 1994)*, volume 106 of *Lecture Notes in Statist.*, pages 87–120. Springer, New York, 1995.

A. M. Odlyzko. The rise and fall of knapsack cryptosystems. In *Cryptology and computational number theory (Boulder, CO, 1989)*, volume 42 of *Proc. Sympos. Appl. Math.*, pages 75–88. Amer. Math. Soc., Providence, RI, 1990.

M. Pohst. On the computation of lattice vectors of minimal length, successive minima and reduced bases with applications. *SIGSAM Bull.*, 15(1):37–44, 1981.

R. A. Rankin. A campanological problem in group theory. *Proc. Cambridge Philos. Soc.*, 44:17–25, 1948.

M. d. P. Sabariego. *Some problems on tilings*. PhD thesis, Universidad de Cantabria, 2008.

C.-P. Schnorr. Factoring integers and computing discrete logarithms via Diophantine approximation. In *Advances in computational complexity theory (New Brunswick, NJ, 1990)*, volume 13 of *DIMACS Ser. Discrete Math. Theoret. Comput. Sci.*, pages 171–181. Amer. Math. Soc., Providence, RI, 1993.

A. Schönhage and V. Strassen. Schnelle Multiplikation grosser Zahlen. *Computing (Arch. Elektron. Rechnen)*, 7:281–292, 1971.

A. Schrijver. *Theory of linear and integer programming*. Wiley-Interscience Series in Discrete Mathematics. John Wiley & Sons Ltd., Chichester, 1986. A Wiley-Interscience Publication.

C. E. Shannon. Communication theory of secrecy systems. *Bell System Tech. J.*, 28: 656–715, 1949.

V. Shoup. Number theory C++ library (NTL). Available at `www.shoup.net/ntl`.

I. E. Shparlinski. *Cryptographic applications of analytic number theory*, volume 22 of *Progress in Computer Science and Applied Logic*. Birkhäuser Verlag, Basel, 2003.

I. E. Shparlinski. Orders of points on elliptic curves. In *Affine algebraic geometry*, volume 369 of *Contemp. Math.*, pages 245–251. Amer. Math. Soc., Providence, RI, 2005.

I. E. Shparlisnki. Pseudorandom points on elliptic curves over finite fields. In *Algebraic geometry and its applications. Proc. first SAGA conference.*, volume 5 of *Ser. Number Theory Appl.*, pages 116–134. World Sci. Publ., Hackensack, NJ, 2008.

J. H. Silverman. *The arithmetic of elliptic curves*, volume 106 of *Graduate Texts in Mathematics*. Springer-Verlag, New York, 1992. Corrected reprint of the 1986 original.

H. S. Stone. The organization of high-speed memory for parallel block transfer of data. *IEEE Trans. Comput.*, C-19(1):47–53, 1970.

B. Sturmfels. *Gröbner bases and convex polytopes*, volume 8 of *University Lecture Series*. American Mathematical Society, Providence, RI, 1996.

B. Sturmfels, R. Weismantel, and G. M. Ziegler. Gröbner bases of lattices, corner polyhedra, and integer programming. *Beiträge Algebra Geom.*, 36(2):281–298, 1995.

B. Vallée. Gauss' algorithm revisited. *J. Algorithms*, 12(4):556–572, 1991.

P. van Emde Boas. Another np-complete problem and the complexity of computing short vectos in a lattice. Technical Report 81-04, Mathematische Instituut, University of Amsterdam, 1981.

J. von zur Gathen and J. Gerhard. *Modern computer algebra*. Cambridge University Press, Cambridge, second edition, 2003.

C. K. Wong and D. Coppersmith. A combinatorial problem related to multimodule memory organizations. *J. Assoc. Comput. Mach.*, 21:392–402, 1974.