On Finding a Shortest Path in Circulant Graphs with Two Jumps

Domingo Gómez, Jaime Gutierrez, Álvar Ibeas,
Carmen Martínez, and Ramón Beivide

Faculty of Sciences, University of Cantabria
Santander E–39071, Spain
jaime.gutierrez@unican.es

Abstract. In this paper we present algorithms for finding a shortest path between two vertices of any weighted undirected and directed circulant graph with two jumps. Our shortest path algorithm only requires $O(\log N)$ arithmetic steps and the total bit complexity is $O(\log^3 N)$, where N is the number of the graph's vertices. This method has been derived from some Closest Vector Problems (CVP) of lattices in dimension two and with ℓ_1-norm.

1 Introduction

An *undirected circulant graph* $C_N(j_1, j_2, \ldots, j_m)$ with N vertices, labeled with integers modulo N, and jumps $j_1, j_2, \ldots, j_m$, is a graph in which each vertex n, $0 \le n \le N-1$, is adjacent to all the vertices $n \pm j_i$ *mod* N, with $1 \le i \le m$. In contrast, a *directed circulant graph* $DC_N(j_1, j_2, \ldots, j_m)$ with N vertices, and jumps $j_1, j_2, \ldots, j_m$ is a graph in which each vertex n, $0 \le n \le N-1$, is adjacent to all the vertices $n+j_i$ *mod* N, with $1 \le i \le m$. Throughout the paper we employ the term circulant graph for both undirected and directed circulant graphs.

This kind of graphs have a vast number of applications in telecommunication networking, VLSI design and distributed computation. Their properties, such as diameters and reliabilities, have been the focus of many research in computer network design [1–3, 6, 15, 19].

Every circulant graph can be associated to a lattice $\mathcal{L}$ which consists of the integer solutions $(x_1, \ldots, x_m) \in \mathbb{Z}^m$ to the system of congruences

$$j_1 x_1 + \cdots + j_m x_m \equiv 0 \bmod N. \tag{1}$$

Given two vertices r and s, a path from r to s in $C_N(j_1, j_2, \ldots, j_m)$ can be described by an integer vector $\mathbf{x} = (x_1, \ldots, x_m) \in \mathbb{Z}^m$ such that

$$\sum_{i=1}^{m} x_i j_i \equiv s - r \bmod N.$$

And a shortest path $\mathbf{x}$ is a path with minimum ℓ_1-norm. In contrast, a path from r to s in the directed circulant graph $DC_N(j_1, j_2, \ldots, j_m)$ can be described by a integer positive vector $\mathbf{x} = (x_1, \ldots, x_m) \in \mathbb{N}^m$ verifying the above equation, and a shortest path $\mathbf{x}$ is a path with minimum ℓ_1-norm.

L. Wang (Ed.): COCOON 2005, LNCS 3595, pp. 777–786, 2005.

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 N^2, which is the number of bits in its adjacency matrix. However, any circulant graph can be described by only m integers. In this representation the input size is $O(m \log N)$. Thus polynomial time algorithms for general graphs may exhibit exponential complexity in the special case of circulant graphs, for this compact input representation.

In [4] the authors establish relations between several routing applications for undirected circulant graphs and the problem of finding the shortest vector in the ℓ_1-norm in the above lattice. They present an algorithm which solves the Shortest-Loop problem in polynomial time for this input measure. In contrast, they show that the Shortest-Path problem is NP-hard for this concise representation.

The particular case $m = 2$, that is, undirected circulant graphs of degree four or distributed double-loop networks and directed circulant graphs of degree two or double-loop networks has been extensively studied, see the surveys [2, 9]). When N is given as a unary input and the time complexity is measured in terms of N, there are several shortest path algorithms for circulant graphs of degree four and for directed circulant graph of degree two, see for instance [6, 8, 10, 17, 18]. Typically, they require $O(N)$ arithmetic steps or $O(\log N)$ time for preprocessing and constant processing time at each node on the route. But a lower bound of the diameter for circulant graph and directed circulant graphs is $\Omega(\sqrt{N})$ (see [2]). So, they are in both cases exponential in the input size $\log N$. The paper [5] shows an algorithm to compute a shortest path in the circulant graph $C_N(1, h)$ (the so called chordal ring graphs) in $O(h/g + \log h)$ time, where $g = \gcd(h, N)$. Obviously, it has also exponential time complexity.

We remark in Section 2 that given a path $\mathbf{c}$, not necessarily a shortest one, in an undirected circulant graph then the problem of finding a shortest path is equivalent to problem of finding a vector that is closest to vector $-\mathbf{c}$ in the lattice defined by Equation 1 with respect to ℓ_1 norm. The well known paper [12] presents an algorithm for solving the Closest Vector Problem in a lattice given by a basis with respect to ℓ_1, ℓ_2 and ℓ_∞ norms. Moreover, fixed the dimension of the lattice, Kannan's algorithm is polynomial in the bit-size of the lattice's basis. So, fixed the number of the jumps m, the paper [12] provides a polynomial time algorithm of finding a shortest path in undirected circulant graphs. According with the above paragraph, this simple observation could be considered as a minor contribution of the present paper.

In this article we give a polynomial time deterministic algorithm to compute a shortest path between two vertices in any weighted circulant graph with N vertices and two jumps. Our algorithm only requires $O(\log N)$ arithmetic steps and the total bit operations is $O(\log^3 N)$. It is based on Closest Vector Problems for ℓ_1-norm. The paper is divided into five sections. In Section 2 we introduce some known lattice concepts and show the relation between Shortest Path Problem and the Closest Vector Problem in a lattice for ℓ_1-norm. The Section 3 is devoted to describe a cubic polynomial time algorithm for solving the two dimensional CVP for ℓ_1-norm. As consequence of the two previous sections we obtain an algorithm for finding a shortest path in any undirected circulant graph of degree four. Section 4 is dedicated for finding a shortest path in double-loop networks. Finally, in Section 5, we analyze the problem in weighted circulant graphs.

2 Closest Vector Problem Versus Shortest Path Problem

The fundamental objects we are dealing with in this section are *lattices*, defined as a discrete subgroup of the space $\mathbb{R}^m$. Equivalently, a lattice is the set of integer linear combinations of some linearly independent vectors. Here we collect some definitions and well-known facts about lattices which can be found, for instance, in [7, 14, 16].

Let $\{\mathbf{b}_1, \ldots, \mathbf{b}_s\}$ be a set of linearly independent vectors in $\mathbb{R}^m$. The set

$$\mathcal{L} = \{\mathbf{z} \ : \ \mathbf{z} = c_1\mathbf{b}_1 + \ldots + c_s\mathbf{b}_s, \quad c_1, \ldots, c_s \in \mathbb{Z}\}$$

is called an *s-dimensional lattice* with *basis* $\{\mathbf{b}_1, \ldots, \mathbf{b}_s\}$.

One basic lattice problem is the *Shortest Vector Problem (SVP)*: given a lattice $\mathcal{L}$ and norm $\|\cdot\|$, finding a nonzero lattice with the smallest norm among all non-zero vectors in the lattice. Unfortunately, there are several indications that this problem is NP-complete. This study has suggested several definitions of a *reduced* basis for a lattice. The following concept is the generalization of the reduced basis concept in celebrated LLL algorithm [13] for lattices of rank 2 to an arbitrary norm [11].

Definition 1. *A basis $\{\mathbf{u}, \mathbf{v}\}$ is called reduced or Gauss-reduced respect to a norm $\|\cdot\|$ if* $\quad \|\mathbf{u}\|, \|\mathbf{v}\| \leq \|\mathbf{u} + \mathbf{v}\|, \|\mathbf{u} - \mathbf{v}\|$.

The algorithm in [11] computes a Gauss-reduced basis from a basis $\{\mathbf{u}, \mathbf{v}\}$ of the lattice $\mathcal{L} \subset \mathbb{Z}^2$ for any computable norm in $O(\log M)$ arithmetic steps where $M = \max(\|\mathbf{u}\|, \|\mathbf{v}\|)$ and a bound for total bit complexity is $O(\log^3 M)$. Notice that, possibly, swapping $\mathbf{u}$ and $\mathbf{v}$, we can always assume that $\|\mathbf{u}\| \leq \|\mathbf{v}\|$. Then, $\|\mathbf{u}\|$ and $\|\mathbf{v}\|$ are the two successive Minkowski minima.

Another related problem for which no polynomial time solution exists is the *Closest Vector Problem, CVP*:

Definition 2. *Given a basis B generating the lattice $\mathcal{L} \subseteq \mathbb{R}^m$, a vector $\mathbf{v} \in \mathbb{R}^m$, and a norm in $\mathbb{R}^m$; the Closest Vector Problem consists on finding a vector in the set $\mathbf{v} + L$ with minimum norm.*

The vertex-symmetry of circulants allows their analysis starting from any vertex, which simplifies their study. We may assume the routing is from vertex 0 to vertex $j \in \mathbb{Z}_N$. Using the well known Extended Euclidean Algorithm we compute a path $\mathbf{c} = (c_1, \ldots, c_m)$ from 0 to vertex j, that is, a solution of the congruence equation: $j_1x_1 + j_2x_2 + \cdots + j_mx_m \equiv j \bmod N$.

We consider the lattice $\mathcal{L}$ given in Equation (1), then we have the following observation:

Lemma 1. *With the above notation, we have:*
- A vector $\mathbf{w}$ is a shortest path between 0 and j in $C_N(j_1,\dots,j_m)$ if and only if $\mathbf{w}$ solves the CVP in the lattice $\mathcal{L}$ with norm ℓ_1 and for the vector $\mathbf{c}$.
- A vector $\mathbf{w}=(x_1,\dots,x_m)$ is a shortest path between 0 and j in $DC_N(j_1,\dots,j_m)$ if and only if $\mathbf{w}$ solves the CVP in the lattice $\mathcal{L}$ with norm ℓ_1 and for the vector $\mathbf{c}$ verifying that $x_i \geq c_i, \quad i=1,\dots,m$.

It is well known that CVP is NP-hard. However, for any fixed dimension CVP can be solved exactly in polynomial time for the Euclidean norm ℓ_2. The algorithm presented in [12] can also be adapted to find the ℓ_1 closest and the ℓ_∞ closest vectors. His algorithm requires cubic polynomial time (polynomial in the bit-size of the lattice's basis) for solving the two dimensional case.

In the next section we present an elementary cubic polynomial time algorithm for solving the two dimensional CVP with respect ℓ_1-norm which can also be extended for solving the shortest path in directed circulant graphs.

3 An Algorithm for Solving Two Dimensional CVP

The main problem addressed in this section is how to compute a vector that is closest to another given vector in a two dimensional lattice with respect to ℓ_1-norm.

For the rest of the paper we only consider the 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 $\mathbf{u}=(u_1,u_2)$.

3.1 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 [11, 16]).

The main goal of this subsection is to obtain a such smallest vector with extra properties for our purpose. The algorithm `REDUCE` is an important tool of this paper:

Algorithm 1.

```
INPUT:  $\mathbf{u},\mathbf{v} \in \mathbb{R}^2,\ \mathbf{v}\neq \mathbf{0}$.
OUTPUT: $\mathrm{Reduce}_{\mathbf{v}}(\mathbf{u}) \in \mathbf{u}+\mathbb{Z}\mathbf{v}\ /\ \|\mathrm{Reduce}_{\mathbf{v}}(\mathbf{u})\|$
        $= \min\{\|\mathbf{u}+\alpha\mathbf{v}\|\ /\ \alpha\in\mathbb{Z}\}$.
– Select $i \in \{1,2\},\ j\in\{1,2\}\backslash\{i\}$ such that $|v_i| > |v_j|$. If $|v_1| = |v_2|$, then
  $i := 1$.
– Return the vector with minimum norm between:
  $\mathbf{u}-\left\lfloor \frac{u_i}{v_i}\right\rfloor \mathbf{v} \quad \wedge \quad \mathbf{u}-\left\lceil \frac{u_i}{v_i}\right\rceil \mathbf{v}$.
  If both have the same norm, return the one with $i$th non-negative component.
```

The next result collects several properties of this concept for later use.

Lemma 2. *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 1, that is, $|v_i| > |v_j| \ \vee (i = 1 \ \wedge \ |v_1| = |v_2|)$. We have the following properties:*

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

The properties 2, 3 and 4 in last Lemma show that $\mathrm{Reduce}_{\mathbf{v}}(\mathbf{u})$ 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 this bound.

Proposition 1. *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 1 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: $\|\mathrm{Reduce}_{\mathbf{v}}(\mathbf{u})\| \leq \frac{\alpha(1+\frac{1}{\beta})+2}{2\alpha+2} \|\mathbf{u}\|$, where $|u_i| = \alpha |u_j|$, for some $\alpha \in \mathbb{R}$.*

A first consequence of the above result is that: $\frac{\alpha(1+\frac{1}{\beta})+2}{2+2\alpha} \leq 1$, because $\beta \geq 1$. And if $\beta > 1$ then the inequality is strict. This result can be extended for the cases $v_j u_j = 0$.

3.2 The Method's Core

We start with three vectors in $\mathbb{R}^2$, two of them linearly independent: $\mathbf{w} + \mathbb{Z} < \mathbf{u}, \mathbf{v} >, \ \mathrm{rank}(\mathbf{u}, \mathbf{v}) = 2$.

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's norm by a constant factor, until we reach some property. To perform this goal, we select a particular basis of the lattice $\mathbb{Z} < \mathbf{u}, \mathbf{v} >$:

Definition 3. *A lattice basis $\{\mathbf{u}, \mathbf{v}\}$ is call extra-reduced when:*

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

In order to study some properties of this kind of lattice basis, we classify vectors $\mathbf{u} = (u_1, u_2) \in \mathbb{R}^2$ in two types: **horizontal**, if $|u_1| \geq |u_2|$ and **vertical**, if $|u_1| < |u_2|$.

Lemma 3. *Let $\{\mathbf{u}, \mathbf{v}\}$ be two linear independent vectors: We have*

1. *If $\{\mathbf{u}, \mathbf{v}\}$ is a Gauss-reduced basis then one of the basis vectors is vertical and the other one is horizontal or both basis vectors are horizontal.*

2. *If $\{\mathbf{u}, \mathbf{v}\}$ is an extra-reduced basis then it is also Gauss-reduced and one of the vector is vertical and the other oner horizontal.*

Kaib and Schonrr showed in [11] how to get a reduced basis (referred to any norm, in particular ℓ_1) of lattice in two dimensions. Thanks Lemma 3, it is easy to reach an extra-reduced basis, by just applying `REDUCE` procedure.

Algorithm 2.

```
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.

1. Set the vectors $\mathbf{U}$, $\mathbf{V}$ so that $\{\mathbf{U}, \mathbf{V}\} = \{\mathbf{u}, \mathbf{v}\}$ and $\|\mathbf{U}\| \leq \|\mathbf{V}\|$.
2. if $\|\mathbf{U}\| < \|\mathbf{V}\|$,  do  $\mathbf{V} := \text{Reduce}_{\mathbf{U}}(\mathbf{V})$.
3. else
   1. if $\mathbf{U}$ and $\mathbf{V}$ are horizontal,
       i. if $|U_1| \neq |U_2|$, swap $\mathbf{U}$ and $\mathbf{V}$.
      ii. $\mathbf{V} := \text{Reduce}_{\mathbf{U}}(\mathbf{V})$.
   2. else
       i. if $\mathbf{U}$ is vertical, swap $\mathbf{U}$ and $\mathbf{V}$.
   3. if $U_2 < 0$, $\mathbf{U} := -\mathbf{U}$.
   4. if $V_1 < 0$, $\mathbf{V} := -\mathbf{V}$.
```

We will use then this extra-reduced basis to perform iterative reductions of the vector $\mathbf{w}$.

Algorithm 3.

```
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: $\mathbf{W} \in \mathbf{w} + \mathbb{Z} < \mathbf{u}, \mathbf{v} >$,  $|W_1| < |u_1|$,  $|W_2| < |v_2|$.

– $\mathbf{W} := \mathbf{w}$.
– while $|W_1| \geq |u_1| \;\vee\; |W_2| \geq |v_2|$
   • if $|W_1| \geq |u_1|$,  do  $\mathbf{W} := \text{Reduce}_{\mathbf{u}}(\mathbf{W})$.
   • else  do  $\mathbf{W} := \text{Reduce}_{\mathbf{v}}(\mathbf{W})$.
```

From Lemma 2 and Proposition 1 we can obtain the following:

Lemma 4. *Algorithm 3 is correct and the number of performed loops is $O(\log \|\mathbf{w}\|)$.*

3.3 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 5. *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:*

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

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 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 5

- $|u_1| \geq |v_2|$ then $\|\alpha\mathbf{u}\| \leq 2\|\mathbf{d}\| \Rightarrow |\alpha| \leq \frac{4(|u_1|+|v_2|)}{|u_1|+|u_2|} \leq 8$.
- $|u_1| < |v_2|$ then $\|\beta\mathbf{v}\| \leq 2\|\mathbf{d}\| \Rightarrow |\beta| \leq \frac{4(|u_1|+|v_2|)}{|v_1|+|v_2|} \leq 8$.

To sum up, jointing Algorithms 2, 3 and 4, we reach our goal:

Algorithm 4.

```
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 $\mathbf{w} + Z < \mathbf{u}, \mathbf{v} >$.

– $\mathbf{U} := \mathbf{u}, \ \mathbf{V} := \mathbf{v}$.
– if $|U_1| < |V_2|$, swap $\mathbf{U}$ and $\mathbf{V}$.
– for $\alpha = [-8, \ldots, 8]$   do
    • $\mathbf{W}_\alpha := \text{Reduce}_{\mathbf{V}}(\mathbf{w} + \alpha\mathbf{U})$.
– Return a vector with minimum norm in   $\{\mathbf{W}_\alpha \ / \ |\alpha| \leq 8\}$.
```

When studying lattices from a complexity point of view, it is customary to assume that the basis vectors (and therefore any lattice vector) have all 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.

If a, b are two integers, such that $b \neq 0$, we denote by $\text{quo}(a, b), \ \text{rem}(a, b)$ the unique integers verifying: $a = b \cdot \text{quo}(a, b) + \text{rem}(a, b), \ 0 \leq \text{rem}(a, b) < |b|$, i.e., $\text{quo}(a, b), \ \text{rem}(a, b)$ are the quotient and the remainder of the Euclidean division of a by b.

For every real number $x \in \mathbb{R}$, as usual we denote by $\text{sgn}\,(x)$ its sign.

In the case of lattices with integer coefficients, Algorithm 1 admits the following form:

Algorithm 5.

```
INPUT:  $\mathbf{u}, \mathbf{v} \in \mathbb{Z}^2, \ \mathbf{v} \neq \mathbf{0}$.
OUTPUT: $\text{Reduce}_{\mathbf{v}}(\mathbf{u}) \in \mathbf{u} + \mathbb{Z}\mathbf{v} \ / \ \|\text{Reduce}_{\mathbf{v}}(\mathbf{u})\|$
        $= \min\{\|\mathbf{u} + \alpha\mathbf{v}\| \ / \ \alpha \in \mathbb{Z}\}$.

– Find $i \in \{1, 2\}, \ j \in \{1, 2\} \backslash \{i\}$ such that $|v_i| > |v_j|$. If $|v_1| = |v_2|$, select
  $i := 1$.
– 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 $\mathbf{u} - \text{quo}(u_i, v_i)\mathbf{v}$.
```

It is straightforward to check both Algorithm 1 and Algorithm 5 have same output for integer vectors. Then, clearly given an undirected circulant graph $C_N(j_1, j_2)$ and vertex $j \in \mathbb{Z}_N$, we can decide if there exists a shortest path from vertex 0 to vertex j and, in the affirmative case, we can compute one on $O(\log^3 N)$ bit operations.

4 Directed Circulant Graphs

Our method can be easily extended to *directed circulant graphs* $DC_N(j_1, j_2)$. First, we need to introduce the concept 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, $\mathbf{u} - \alpha\mathbf{v}$ with both components positive may not achieve the minimum norm over all integer component $\mathbf{u} - \beta\mathbf{v}$, with α, β integral.

Algorithm 6.

```
INPUT   $\mathbf{u}, \mathbf{v} \in \mathbb{Z}^2,\ \mathbf{v} \neq 0$
OUTPUT  $\mathrm{PRed}_{\mathbf{v}}(\mathbf{u}) \in (\mathbf{u} + \mathbb{Z}\mathbf{v}) \cap \mathbb{N}^2,$
        $\|\mathrm{PRed}_{\mathbf{v}}(\mathbf{u})\| \leq \|\mathbf{u} + \alpha\mathbf{v}\|\ \forall \alpha \in \mathbb{Z} / \mathbf{u} + \alpha\mathbf{v} \in \mathbb{N}^2 \qquad \vee$
        $\emptyset$, if $(\mathbf{u} + \mathbb{Z}\mathbf{v}) \cap \mathbb{N}^2 = \emptyset$

1. Find $i \in \{1,2\},\ j \in \{1,2\} \backslash \{i\}$ such that $|v_i| > |v_j|$. If $|v_1| = |v_2|$, select $i := 1$.
2. Set $\varepsilon = \begin{cases} -1, & \text{if } |v_1| \geq |v_2| \\ 1, & \text{if } |v_1| < |v_2|. \end{cases}$
3. Compute $\Delta := \varepsilon \,\mathrm{sgn}\,(u_1 v_2 - u_2 v_1)$.
4. If $v_j = 0$,
       $B := \Delta \mathrm{sgn}\,(v_i)\,.$
   1. If $B = -1$, OUTPUT $\emptyset$.
   2. If $B \geq 0$, OUTPUT $\mathbf{u} - \mathrm{sgn}\,(v_i)\,\mathrm{quo}(u_i, |v_i|)\mathbf{v}$.
5. If $v_j \neq 0$,
       $A := -\Delta \mathrm{sgn}\,(v_j)\,,\ B := \Delta \mathrm{sgn}\,(v_i)\,.$
   1. If $(A = -1 \wedge B = -1)$, OUTPUT $\emptyset$.
   2. If $(A = -1 \wedge B \geq 0)$, OUTPUT $\mathbf{u} - \mathrm{sgn}\,(v_i)\,\mathrm{quo}(u_i, |v_i|)\mathbf{v}$.
   3. If $(A \geq 0 \wedge B = -1)$, OUTPUT $\mathbf{u} - \mathrm{sgn}\,(v_j)\,\mathrm{quo}(u_j, |v_j|)\mathbf{v}$.
   4. If $(A \geq 0 \wedge B \geq 0)$,
        i. $\mathbf{w} := \mathbf{u} - \mathrm{sgn}\,(v_i)\,\mathrm{quo}(u_i, |v_i|)\mathbf{v}$.
       ii. If $\mathbf{w} \in \mathbb{N}^2$, OUTPUT $\mathbf{w}$.
      iii. Else, OUTPUT $\emptyset$.
```

It is easy to check that the previous algorithm is correct and a bound for the bit complexity on computing $\mathrm{PRed}_{\mathbf{v}}(\mathbf{u})$ is $O(\log^2 M)$, 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 6. *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, 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 7.

```
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 $\mathbf{z}$ in $\mathbf{w} + \mathbb{Z} < \mathbf{u}, \mathbf{v} >$.
– if $\|\mathbf{u}\| \geq \|\mathbf{v}\|$
    • for $\alpha = -16, \ldots, 16$   do   $\mathbf{d}_\alpha := \mathrm{PRed}_{\mathbf{v}}(\mathbf{z} + \alpha\mathbf{u})$.
– else
    • for $\alpha = -16, \ldots, 16$   do   $\mathbf{d}_\alpha := \mathrm{PRed}_{\mathbf{u}}(\mathbf{z} + \alpha\mathbf{v})$.
– Return a vector with minimum norm in $\{\mathbf{d}_\alpha \ / \ |\alpha| \leq 16\}$.
```

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

Let $M = \max\{\|\mathbf{u}\|, \|\mathbf{v}\|\}$, by Lemma 5 and Lemma 6 and since $\{\mathbf{u}, \mathbf{v}\}$ is an extra reduced basis, we have:

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

□

5 Weighted Circulant Graphs

In this section, we consider weighted circulant graphs with two jumps $C_N(j_1, j_2)$ and weights $w = (w_1, w_2)$.

Theorem 8. *Given a circulant graph $C_N(j_1, j_2)$ with weights $w = (w_1, w_2)$ we can find a shortest path on cubic polynomial time.*

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 Section 3 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 $< \Phi(\mathbf{u}), \Phi(\mathbf{v}) >$ and target vector $\Phi(\mathbf{c})$, 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.

References

1. R. Beivide, E. Herrada, J.L. Balcázar and A. Arruabarrena. *Optimal Distance Networks of Low Degree for Parallel Computers.* IEEE Transactions on Computers, Vol. C-40, No. 10, pp. 1109-1124, 1991.
2. J.-C. Bermond, F. Comellas and D.F. Hsu. *Distributed Loop Computer Networks: A Survey.* Journal of Parallel and Distributed Computing, Vol. 24, pp. 2-10, 1995.
3. F.T. Boesch and R. Tindell. *Circulants and their connectivity.* J. Graph Theory, Vol. 8, pp. 487-499, 1984.
4. J.-Y. Cai, G. Havas, B. Mans, A. Nerurkar, J.-P. Seifert and I. Shparlinski. *On Routing in Circulant Graphs.* Proc. Fifth Annual International COCOON- 1999, LNCS vol. 1627, Springer-Verlag, T. Asano, H. Imai, D.T. Lee, S. Nakano, and T. Tokuyama (Eds.), pp. 360-369.
5. N. Chalamaiah and B. Ramamurthy. *Finding shortest paths in distributed loop networks.* Information Processing Letters, Vol. 67, pp. 157-161, (1998).
6. Y. Cheng and F. K. Hwang. *Diameters of Weighted Double Loop Networks*, Journal of Algorithms 9, 401-410, 1988.
7. M. Grötschel, L. Lovász and A. Schrijver, *Geometric algorithms and combinatorial optimization*, Springer-Verlag, Berlin, 1993.
8. D. J. Guan. *An Optimal Message Routing Algorithm for Double-Loop Networks.* Information Processing Letters 65(5): 255-260, 1998.
9. F. K. Hwang. *A complementary survey on double-loop networks*, Theoretical Computer Science 263, 2001. 211-229.
10. F. K. Hwang. *A survey on multi-loop networks*, Theoretical Computer Science 299, 2003. 107-121.
11. M. Kaib and C.P. Schnorr: "The Generalized Gauss Reduction Algorithm". Journal of Algorithms **21**, 3 (1996): 565-578.
12. R. Kannan. *Minkoswski's convex body theorem and integer programing*, *Mathematics of operation research* , **12**(3), 415–440, 1987.
13. A. K. Lenstra, H. W. Lenstra and L. Lovász. 'Factoring polynomials with rational coefficients', *Mathematische Annalen*, **261**, 515–534, 1982.
14. L. Lovász and H. Scarf: "The Generalized Basis Reduction Algorithm". Mathematics of Operations Research **17**, 3 (1992): 751-764.
15. B. Mans. *Optimal Distributed algorithms in unlabeled tori and chordal rings* Journal of Parallel and Distributed Computing, Vol. 46, pp. 80-90, 1997.
16. D. Micciancio and S. Goldwasser. *Complexity of Lattices Problems*, The Kluwer International Series in Engineering and Computer Science, vol. 671, 2002.
17. K. Mukhopadhyaya and B.P. Sinha. *Fault-Tolerant Routing Algorithm in distributed Loop Networks.* IEEE Transactions on Computers 44(12): 1452-1456, 1995.
18. Yu-Liang Liu, Yue-Li Wang and D. J. Guan. *An Optimal Fault-Tolerant Routing Algorithm for Double-Loop Networks.* IEEE Transactions on Computers 50(5): 500-505, 2001.
19. J. Žerovnik, and T. Pisanski. *Computing the Diameter in Multiple-Loop Networks.* J. Algorithms 14(2): 226-243, 1993.