Theoretical Computer Science 412 (2011) 4814–4823

Contents lists available at ScienceDirect

Theoretical Computer Science

journal homepage: www.elsevier.com/locate/tcs

Computing efficiently the lattice width in any dimension

Émilie Charrier [a,*], Fabien Feschet [b], Lilian Buzer [c]

[a] Université de Savoie, Laboratoire de Mathématiques, UMR 5127 CNRS, F-73376 Le Bourget du Lac, France
[b] Clermont Université, Université d'Auvergne, Santé-Technologie, 49 Bd F. Mitterrand, BP 32, F-63001 Clermont-Ferrand, France
[c] Université Paris-Est, LABINFO-IGM, UMR CNRS 8049, ESIEE Paris, 2 Bd Blaise Pascal, BP 99, F-93162 Noisy le Grand Cedex, France

ARTICLE INFO

Keywords:
Lattice width
Shortest vector
Greedy algorithm

ABSTRACT

We provide an algorithm for the exact computation of the lattice width of a set of points K in $\mathbb{Z}^2$ in linear-time with respect to the size of K. This method consists in computing a particular surrounding polygon. From this polygon, we deduce a set of candidate vectors allowing the computation of the lattice width. Moreover, we describe how this new algorithm can be extended to an arbitrary dimension thanks to a greedy and practical approach to compute a surrounding polytope. Indeed, this last computation is very efficient in practice as it processes only a few linear time iterations whatever the size of the set of points. Hence, it avoids complex geometric processings.

1. Introduction

Integer Linear Programming is a fundamental tool in optimization, in operational research and in economics.... Moreover, it is interesting in itself since the problem is NP-hard in the general case. Several works were done for the planar case [20,28,15] before Lenstra [23] proved that Integer Linear Programming can be solved in polynomial time when the dimension is fixed. Faster and faster algorithms are nowadays developed and available, making the use of Integer Linear Programming reliable even for high dimensional problems. The approach of Lenstra uses the notion of *lattice width* for precise lattice definition to detect directions for which the polyhedron of solutions is thin. In polynomial time, the problem is then reduced to a feasibility question: given a polyhedron P, determine whether P contains an integer point. To solve it, Lenstra approximates the width of the polyhedron and gives a recursive solution solving problems of smaller dimension. The approximate lattice width is also used in the recent algorithms of Eisenbrand and Rote [10] and Eisenbrand and Laue [9] for the 2-variable problem.

Not surprisingly, following the arithmetical approach of Reveillès [26,7], the lattice width is also a fundamental tool in digital geometry since it corresponds to the notion of width for digital objects [11]. Moreover, as an application of the lattice width computation, we mention the intrinsic characterization of linear structures [12]. Indeed, the lattice width can be computed for any digital set but it does not correspond to a direct measure of linearity. However, when combining the lattice width along a direction and along its orthogonal, it can be used as a linearity measure. The work in [12] is currently extended, by the second author of the present paper, to higher dimensions for detecting either linear or tubular structures. A preliminary algorithm for the computation of the lattice width in the two-dimensional case was given in [11] with a geometrical interpretation. It has the advantage to be extensible to the incremental and to the dynamic case but it seems difficult to extend it to an arbitrary dimension. We proposed in [4] a new method, efficient in any dimension, and we extend it here by detailing its main steps and by providing experimental results. This approach is based on the computation of a particular surrounding polytope which is used to bound the set of candidate vectors to define the lattice width. This

* Corresponding author. Tel.: +33 479759438; fax: +33 479758142.
E-mail addresses: Emilie.Charrier@univ-savoie.fr (É. Charrier), feschet@acm.org (F. Feschet), buzerl@esiee.fr (L. Buzer).

0304-3975/$ – see front matter
doi:10.1016/j.tcs.2011.02.009

algorithm runs in linear time for the two-dimensional case. In higher dimensions, we propose a greedy method to compute the surrounding polytope. This approach is efficient as it computes a maximal simplex over the set of points in few iterations whatever the size of the set of points. Then, we directly deduce an appropriate surrounding polytope from this simplex.

The paper is organized as follows. In Section 2, the main definitions and tools are presented. Then, we describe the two-dimensional algorithm introduced in [11] in Section 3. As this method cannot be easily extensible to higher dimensions, we introduce in Section 4 a new geometric approach to estimate the lattice width in any dimension. This algorithm is based on the computation of a particular surrounding polytope and we describe in Section 5 geometric methods to compute it. We first focus on the two-dimensional case and we provide two linear-time approaches. Then, since these geometric constructions might be difficult to extend in arbitrary dimensions, we provide a greedy algorithm which runs in any dimension. Some conclusions and perspectives end the paper.

2. Definitions from integer lattice theory

In this section, we review some definitions from algorithmic number theory and we provide a precise formulation of the problem we solve. Definitions are taken from [2,10,29].

Let K be a set of n points of $\mathbb{Z}^d$. Moreover, we suppose that all numbers appearing in the points and in the vector coordinates have their bit size bounded by $\log s$. The *width* of K along a direction $c \neq 0$ in $\mathbb{R}^d$ is defined as:

$$\omega_c(K) = \max\left\{c^T x \mid x \in K\right\} - \min\left\{c^T x \mid x \in K\right\}. \tag{1}$$

Geometrically, if a set K has a width of l along the direction c then any integer point which lies in the interior of the convex hull of K also lies on a hyperplane of the form $c^T x = \lambda$ where λ corresponds to an integer value between $\min\{c^T x \mid x \in K\}$ and $\max\{c^T x \mid x \in K\}$. We say that K can be covered by these $\lfloor l \rfloor + 1$ parallel hyperplanes. It is straightforward to see that $\omega(K) = \omega(\mathrm{conv}(K))$ where $\mathrm{conv}(K)$ denotes the convex hull of K.

Let $\mathbb{Z}^{d*} = \mathbb{Z}^d \setminus \{0\}$ denote the set of integer vectors different from zero. The *lattice width* of K is defined as follows:

$$\omega(K) = \min_{c \in \mathbb{Z}^{d*}} \omega_c(K). \tag{2}$$

We notice that the lattice width is an integer value. We briefly recall some basic and important properties about inclusion and translation:

Lemma 1. *For any sets of points A and B, such that conv(A) $\subset$ conv(B) and for any vector $c \in \mathbb{Z}^{d*}$, we have $\omega_c(A) \leq \omega_c(B)$. Thus, it follows that $\omega(A) \leq \omega(B)$.*

Lemma 2. *Suppose that A' corresponds to the points of A translated in the same direction. By definition, we know that for any $c \in \mathbb{Z}^{d*}$, $\omega_c(A) = \omega_c(A')$ and so we have $\omega(A) = \omega(A')$. The lattice width is invariant under translation.*

The problem we would like to solve is the following one:

Problem (Lattice Width)
Given a set of integer points $K \subset \mathbb{Z}^d$, find its lattice width $\omega(K)$ as well as all the vectors $c \in \mathbb{Z}^d$ such that $\omega_c(K) = \omega(K)$.

It is known [23] that the lattice width of a convex set K is obtained for the shortest vector with respect to the *dual norm* whose unit ball is the polar set of the set $\frac{1}{2}(K + (-K))$. In the general case, computing the shortest vector is NP-hard. Thus, approximations of the solution can be computed via standard arguments [29,19,27], but it does not lead us to an easy exact algorithm in arbitrary dimension.

3. Computing lattice width in the planar case

3.1. The 2006 algorithm design

In 2006, Feschet proposed in [11] a method to compute in $O(n + n \log s)$ time the lattice width of a set of two-dimensional integer points. We recall in this part this algorithm via connections with the notion of digital straightness and more precisely with the notion of arithmetical digital lines [26]. This two-dimensional algorithm requires a convex polygon as input; as a consequence we have to compute the convex hull H of K in $O(n)$ time [18].

The main idea in [11] is based on the principle that the lattice width of K is necessarily reached for two *opposite* vertices of its convex hull. To define the notion of *opposite*, we rely on the notion of *supporting lines* well known in computational geometry [6]. A *supporting line* of H is a line D such that $D \cap K \neq \emptyset$ and H is contained entirely in one of the half-planes bounded by D. For each supporting line D, there exists at least one vertex v of H such that the parallel line D_v to D passing through v is such that H entirely lies in the strip bounded by D and D_v. If s denotes a vertex of H belonging to D then s and v are called *opposite* (see Fig. 1, left). Opposite pairs are also called *antipodal* pairs. Note that in general, a supporting line intersects H at only one point. The supporting line D intersecting H along an edge is called *principal* supporting line.

We now suppose H to be oriented counter-clockwise. As in the classical Rotating Calipers algorithm of Toussaint [16], we can rotate the principal supporting lines D around the right vertex of $D \cap H$. D_v is also rotated around v to keep it parallel