Random generation of Q-convex sets

Sara Brunetti [a] and Alain Daurat [b]

[a] *Dipartimento di Scienze Matematiche e Informatiche, Università di Siena, Pian dei Mantellini 44, 53100, Siena, Italy*

[b] *LSIIT UMR 7005 CNRS-ULP, Pôle API, Boulevard Sébastien Brant, 67400 Illkirch-Graffenstaden, France*

Abstract

The problem of randomly generating Q-convex sets is considered. We present two generators. The first one uses the Q-convex hull of a set of random points in order to generate a Q-convex set included in the square $[0, n)^2$. This generator is very simple, but is not uniform and its performance is quadratic in n. The second one exploits a coding of the salient points, which generalizes the coding of a border of polyominoes. It is uniform, and is based on the method by rejection. Experimentally, this algorithm works in linear time in the length of the word coding the salient points. Besides, concerning the enumeration problem, we determine an asymptotic formula for the number of Q-convex sets according to the size of the word coding the salient points in a special case, and in general only an experimental estimation.

Key words: uniform generator, lattice sets, convexity, salient points

1 Introduction

A *lattice set* is a non-empty finite subset of the integer plane $\mathbb{Z}^2$. In this paper we address the random generation of a class of special lattice sets, called Q-convex sets. The class of Q-convex sets generalizes both the class of HV-convex polyominoes and the one of convex sets. HV-convex polyominoes are well-known combinatorial objects, and the large interest for this class is testified by the numerous results concerning the enumeration, the generation, and the reconstruction [10,11,2]. Convex sets are mainly studied in computational and discrete geometry [15,16]. The class of Q-convex sets has been introduced

Email addresses: `sara.brunetti@unisi.it` (Sara Brunetti), `daurat@dpt-info.u-strasbg.fr` (Alain Daurat).

Preprint submitted to Elsevier Science *20 April 2006*

for its interest in Discrete Tomography ([5,6,8]). Results concerning the reconstruction and uniqueness problems have been established generalizing the corresponding problems for HV-convex polyominoes and convex sets. We present here two random generators. In Section 3 we describe a very simple algorithm that generates at random a set in the square $[0, n)^2$, and returns its Q-convex hull (a generalization of the convex hull). Unfortunately, this algorithm is not uniform, and so we propose in Section 4 a probabilistic algorithm that achieves uniformity. This algorithm is based on the method by rejection, that is, consists in uniformly generating an element of an enlarged class of Q-convex sets, then keeping the element if it is Q-convex or otherwise refusing it and trying again. The uniformity is not destroyed by filtering out. Hochstattler et al. [11] designed a probabilistic generator for HV-convex polyominoes using an encoding of the border of the studied polyominoes by a word on an alphabet of two letters: H(orizontal step), V(ertical step). Since the definition of Q-convex sets does not refer to the coordinate directions like HV-convex polyominoes, but to any couple of lattice directions, our generator generalizes the algorithm in [11] as it uses a coding of "salient points" by a word on an alphabet whose letters depend on the considered couple of directions. Thus, the efficiency of the generator depends on the length of the word coding the salient points and on the fixed couple of directions. We report on the experiments conducted in order to estimate the success probability.

2 The class of Q-convex sets

In this section we introduce the necessary notations to get the definition of Q-convex sets. This definition does not refer to the coordinate directions, but to any couple of lattice directions. A direction is given by its equation $ax + by =$ constant, and a *lattice direction* has a and b integers. In the whole paper two lattice directions $p = ax+by$ and $q = cx+dy$ are fixed, and without loss of generality we suppose that $\gcd(a, b) = 1$, $\gcd(c, d) = 1$. The vectors $\vec{p}$ and $\vec{q}$ are defined by $\vec{p} = (-b, a)$ and $\vec{q} = (d, -c)$.

Let $\delta = |\det(p, q)| = |ad - bc|$. We denote by $\langle i, j\rangle_{pq}$ the point M such that $p(M) = i$ and $q(M) = j$. It must be noticed that $\langle i, j\rangle_{pq}$ is not in $\mathbb{Z}^2$ in general. More precisely there exists κ coprime with δ such that

$$\langle i, j\rangle_{pq} \in \mathbb{Z}^2 \iff j \equiv \kappa i \mod \delta$$

(see for example [9]).

Figure 1 illustrates the case $p = x - y$ and $q = 2x + y$: with these directions we have $\delta = 3$ and $\kappa = 2$. All the examples in the paper refer to these two directions (except in subsection 4.3).

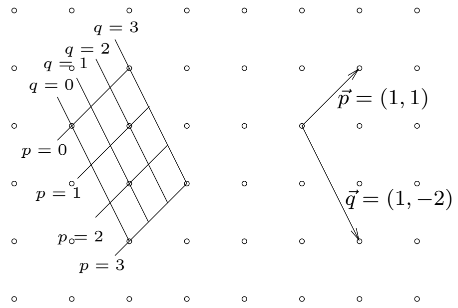

Fig. 1. Directions $p = x - y$ and $q = 2x + y$.

For any $M \in \mathbb{R}^2$, we can define the four quadrants around M along the directions p and q by:

$$\begin{aligned}
R_0(M) &= \{N \in \mathbb{Z}^2 \ / \ p(N) \leq p(M) \text{ and } q(N) \leq q(M)\} \\
R_1(M) &= \{N \in \mathbb{Z}^2 \ / \ p(N) \geq p(M) \text{ and } q(N) \leq q(M)\} \\
R_2(M) &= \{N \in \mathbb{Z}^2 \ / \ p(N) \geq p(M) \text{ and } q(N) \geq q(M)\} \\
R_3(M) &= \{N \in \mathbb{Z}^2 \ / \ p(N) \leq p(M) \text{ and } q(N) \geq q(M)\}
\end{aligned}$$

(see Fig. 2).

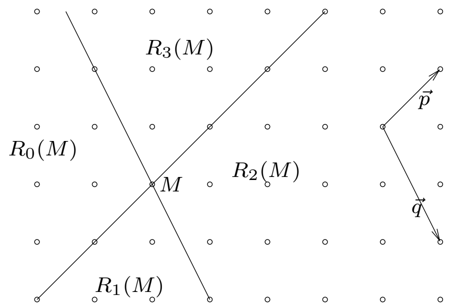

Fig. 2. The four quadrants

Definition 1 *The Q-convex hull of a lattice set E is the set of points $M \in \mathbb{Z}^2$ such that $R_k(M) \cap E \neq \varnothing$ for all k.*

We denote the Q-convex hull of a set $E \subset \mathbb{Z}^2$ by $\mathcal{Q}(E)$.

Definition 2 *A lattice set E is* Q-convex *(quadrant-convex) if $E = \mathcal{Q}(E)$.*

In this paper we refer to this definition even if usually we derive it from the following one: a lattice set E is Q-convex if $R_k(M) \cap E \neq \varnothing$ for all $k \in \{0, 1, 2, 3\}$ implies $M \in E$.

The Q-convex hull of E can be easily computed by starting from E (see also [7]). Let $\Delta = \{(i, j) \in \mathbb{Z}^2 : \min p(E) \leq i \leq \max p(E), \min q(E) \leq j \leq \max q(E)\}$ and let $(V_k(i, j))_{(i,j) \in \Delta}$ be the array of boolean variables defined by:

$$V_k(i, j) = 1 \text{ if } R_k(\langle i, j\rangle_{pq}) \cap E \neq \varnothing, \text{ else } 0,$$

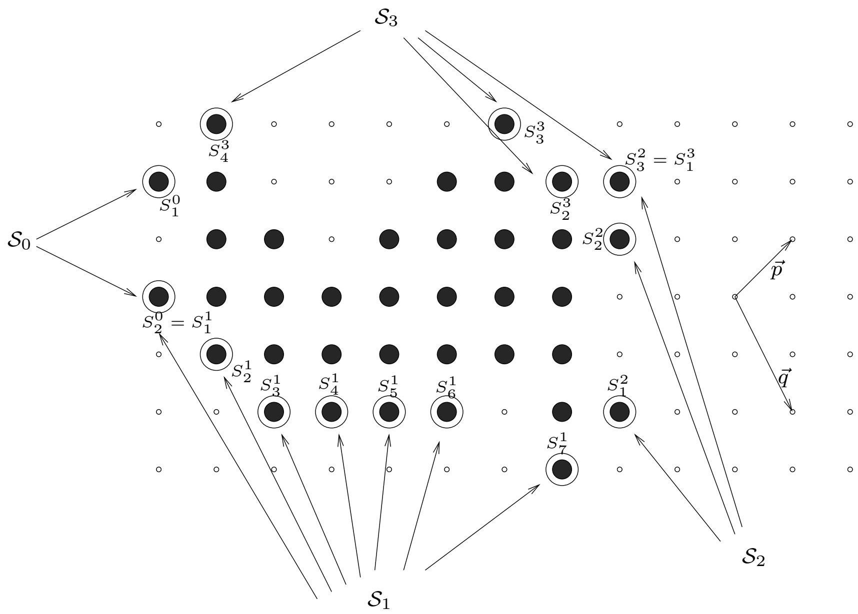

Fig. 3. A Q-convex set. The encircled points are the salient points of the set.

for $k = 0, 1, 2, 3$. We can compute these arrays by the induction:

$$\begin{aligned} V_0(i,j) &= V_0(i-1,j) \vee V_0(i,j-1) \vee \text{“}\langle i,j\rangle_{pq} \in E\text{”} \\ V_1(i,j) &= V_1(i+1,j) \vee V_1(i,j-1) \vee \text{“}\langle i,j\rangle_{pq} \in E\text{”} \\ V_2(i,j) &= V_2(i+1,j) \vee V_2(i,j+1) \vee \text{“}\langle i,j\rangle_{pq} \in E\text{”} \\ V_3(i,j) &= V_3(i-1,j) \vee V_3(i,j+1) \vee \text{“}\langle i,j\rangle_{pq} \in E\text{”}. \end{aligned}$$

Finally the Q-convex hull of E is obtained by the formulas:

$$\mathcal{Q}(E) = \{\langle i,j\rangle_{pq} \in \Delta : \; V_0(i,j) \wedge V_1(i,j) \wedge V_2(i,j) \wedge V_3(i,j)\},$$

and the computation requires a number of operations proportional to the size of Δ, i.e., $O((\max p(E) - \min p(E)) \cdot (\max q(E) - \min q(E)))$ operations.

Definition 3 *Let E be any lattice set. A point $M \in E$ is a salient point of E if $M \notin \mathcal{Q}(E \setminus \{M\})$.*

The set of salient points of E is denoted by $\mathcal{S}(E)$.

If M is a salient point of E, then there exists k such that $R_k(M) \cap E = \{M\}$. So $\mathcal{S}(E) = \mathcal{S}_0(E) \cup \mathcal{S}_1(E) \cup \mathcal{S}_2(E) \cup \mathcal{S}_3(E)$ where $\mathcal{S}_k(E) = \{M \in E : R_k(M) \cap E = \{M\}\}$.

The set of salient points of a lattice set can be easily computed as we have:

$$\begin{aligned}
\mathcal{S}_0(E) &= \{\langle i,j\rangle_{pq} \in E \ : \ \overline{V_0(i-1,j)} \wedge \overline{V_0(i,j-1)}\} \\
\mathcal{S}_1(E) &= \{\langle i,j\rangle_{pq} \in E \ : \ \overline{V_1(i+1,j)} \wedge \overline{V_1(i,j-1)}\} \\
\mathcal{S}_2(E) &= \{\langle i,j\rangle_{pq} \in E \ : \ \overline{V_2(i+1,j)} \wedge \overline{V_2(i,j+1)}\} \\
\mathcal{S}_3(E) &= \{\langle i,j\rangle_{pq} \in E \ : \ \overline{V_3(i-1,j)} \wedge \overline{V_3(i,j+1)}\}
\end{aligned}$$

where $\overline{V_0}$, $\overline{V_1}$, $\overline{V_2}$, $\overline{V_3}$ are the negations of the arrays of boolean variables defined above.

We recall Proposition 5 and Theorem 6 of [7]:

Proposition 4 *For any lattice set E: $\mathcal{Q}(E) = \mathcal{Q}(\mathcal{S}(E))$ and $\mathcal{S}(E) = \mathcal{S}(\mathcal{Q}(E))$.*

From this proposition we can deduce:

Corollary 5 *For any lattice sets E and F:*

$$\mathcal{S}(E) = \mathcal{S}(F) \iff \mathcal{Q}(E) = \mathcal{Q}(F).$$

Therefore, every Q-convex set is characterized by its set of salient points.

3 A simple but non-efficient generator

In this section we study a very simple generator. This generator produces a lattice set that is Q-convex with respect to the two directions p and q and included in the square $\{0, \ldots, n-1\}^2$, for a given n. The idea of the algorithm is the following: the generated set F is the Q-convex hull of an ordinary set E. Therefore, first the algorithm generates E and then it computes $\mathcal{Q}(E)$. (This idea is quite common for the classes stable by intersection, see for example [17]). We suppose that rand is a function that returns a real in $[0, 1[$ and that is uniform, i.e. the probability of rand $< \alpha$ with $0 \leq \alpha < 1$ is α.

```
GENQ1(n)
  E ← ∅
  for x ← 0 to n − 1 do
    for y ← 0 to n − 1 do
      if rand < 1/2 then
        E ← E ∪ {(x, y)}
      end if
    end for
  end for
  F ← Q(E)
  return(F)
```

By Corollary 5, given a Q-convex set $F \in [0, n-1]^2$, the procedure GENQ1 outputs F if and only if the intermediate set E satisfies $\mathcal{S}(F) \subset E \subset F$. For example the set of Fig. 3 is produced by any set E which contains the encircled points and is contained in the set. So each Q-convex set F has a probability to be produced which is proportional to $2^{\mathrm{card}(F)-\mathrm{card}(\mathcal{S}(F))}$. It implies that GENQ1 is not uniform.

We can transform this generator in an uniform but probabilistic generator:

GENQ2(n)

$F \leftarrow$ GENQ1(n)
if rand $< \frac{1}{2^{\mathrm{card}(F)-\mathrm{card}(\mathcal{S}(F))}}$ **then**
  return(F)
else
  return(FAILURE)
end if

The probability of success of this generator is given by the mean of $\frac{1}{2^{\mathrm{card}(F)-\mathrm{card}(\mathcal{S}(F))}}$ on the produced sets F weighted with the probability of F to be produced. We can also compute this mean by considering the intermediate sets E of GENQ1 which have all the same probability so:

$$\mathrm{psuccess(GENQ2)} = \frac{1}{2^{n^2}} \sum_E \frac{1}{2^{\mathrm{card}(\mathcal{Q}(E))-\mathrm{card}(\mathcal{S}(E))}}$$

On each line $p = cste$ there are at most two salient points, so $\mathrm{card}(\mathcal{S}(E)) \leq M \cdot n$ where $p = ax + by$, $M = 2(|a| + |b|)$. It implies that:

$$\begin{aligned}
\mathrm{psuccess(GENQ2)} &\leq \frac{2^{M\cdot n}}{2^{n^2}} \sum_E \frac{1}{2^{\mathrm{card}(\mathcal{Q}(E))}} \leq \frac{2^{M\cdot n}}{2^{n^2}} \sum_E \frac{1}{2^{\mathrm{card}(E)}} \\
&= \frac{2^{M\cdot n}}{2^{n^2}} \sum_{k=0}^{n^2} \frac{\binom{n^2}{k}}{2^k} \qquad \text{(variable change } k = \mathrm{card}(E)) \\
&= \frac{2^{M\cdot n}}{2^{n^2}} \sum_{k=0}^{n^2} \binom{n^2}{k} \left(\frac{1}{2}\right)^k 1^{n-k} = \frac{2^{M\cdot n}}{2^{n^2}} \left(\frac{1}{2}+1\right)^{n^2} = 2^{M\cdot n} \left(\frac{3}{4}\right)^{n^2}.
\end{aligned}$$

Thus, the probability of success converges exponentially to zero as n tends to ∞.

We can transform GENQ2 in a generator which does not fail:

GENQ3(n)

---

repeat
  $F \leftarrow$ GENQ2(n)
until $F \neq$ FAILURE
return(F)

By standard results on the geometric distribution, the number of iterations in the **repeat** loop is $\frac{1}{\text{psuccess(GENQ2)}} \geq \frac{1}{2^{M\cdot n}}\left(\frac{4}{3}\right)^{n^2}$ on average, and so GENQ3(n) has an average time-complexity which is exponential.

4 A fast uniform generator

In this section we present a probabilistic algorithm that uniformly generates Q-convex sets. This algorithm adopts the well known approach of studying the language defined on the boundary of the sets of the class in examination. In the first part we show that this language uses an alphabet whose letters are determined by the directions p and q, and whose cardinality is $\delta + 1$. (Recall that $\delta = |\det(p, q)|$). The encoding passes through two steps: since a Q-convex set is characterized by its salient points, and couples of consecutive salient points can be seen as vectors, we first decompose these vectors into "primitive vectors" in such a way that each primitive vector can be encoded by a letter; then we determine a word encoding the boundary of the Q-convex set as a concatenation of letters. In the second part we provide the algorithm generating a word of the language of length n and an integer α. From these information the algorithm attempts to generate a Q-convex set.

4.1 Consecutive salient points, and salient-word

Let E be a Q-convex set. Thus, it is the Q-convex hull of its salient points, that is, $E = \mathcal{S}(E)$. We now focus on the subsets $\mathcal{S}_0(E), \mathcal{S}_1(E), \mathcal{S}_2(E), \mathcal{S}_3(E)$ of $\mathcal{S}(E)$.

Two points A and B of $\mathcal{S}_0(E)$ are said to be 0-*consecutive* if $R_0(C) \cap E = \varnothing$ with $C = \langle \max(p(A), p(B)) - 1, \max(q(A), q(B)) - 1\rangle$. This gives to $\mathcal{S}_0(E)$ a graph relation. It is easy to see that this graph is a chain. More precisely $\mathcal{S}_0(E)$ can be written as $\{S^0_1, S^0_2, S^0_3, \ldots, S^0_{m_0}\}$ with S^0_j, S^0_{j+1} consecutive, $p(S^0_1) = \min p(E)$, $q(S^0_{m_0}) = \min q(E)$. Besides, 0-consecutive points have the property that $S^0_{j+1} \in R^\circ_1(S^0_j)$, where, more in general, $R^\circ_k(M)$ is the quadrant $R_k(M)$ without its border: $R^\circ_k(M) = R_k(M) \setminus (R_{(k-1)\%4}(M) \cup R_{(k+1)\%4})$ ($x\%y$ is the reminder in the division of x by y).

Analogously we can define the k-consecutivity between two points of $\mathcal{S}_k(E)$; $\mathcal{S}_k(E) = \{S^k_1, S^k_2, S^k_3, \ldots, S^k_{m_i}\}$ with S^k_j, S^k_{j+1} k-consecutive, for $j = 1, \ldots, m_j$ and $k = 1, 2, 3$, and $q(S^0_{m_0}) = q(S^1_1) = \min q(E)$, $p(S^1_{m_1}) = p(S^2_1) = \max p(E)$, $q(S^2_{m_2}) = q(S^3_1) = \min q(E)$, $p(S^3_{m_3}) = p(S^0_1) = \min p(E)$. Besides, k-consecutive points have the property that $S^k_{j+1} \in R^\circ_{(k+1)\%4}(S^k_j)$, for $j = 1, \ldots, m_j$ and $k = 1, 2, 3$ (see Fig. 3).

The sequence of the salient points is the concatenation of the four previous defined sequences:

$$(S) = (S_1^0, \dots, S_{m_0}^0, S_1^1, \dots, S_{m_1}^1, S_0^2 \dots, S_{m_2}^2, S_1^3, \dots, S_{m_3}^3, S_1^0). \tag{1}$$

We are going to encode (S) by a word on the alphabet $\mathcal{A} = \{0, 1, 2, \dots, \delta\}$. Hereafter let us denote the set of words on $\mathcal{A}$ by $\mathcal{A}^*$, and the concatenation of $w_1, w_2 \in \mathcal{A}^*$ by $w_1 \diamond w_2$.

A sequence of salient points can be seen as a path of oriented steps, the latter defined by couples of consecutive points. Thus, by definition, every step is a vector, and the vectors leave the lattice set on their left-hand side. We distinguish among the vectors those "primitive" ones. We need to introduce some additional notions. Let us define two functions $\chi, \chi' : \mathcal{A} \to \mathcal{A}$ by $\chi(0) = \chi'(0) = \delta$, $\chi(\delta) = \chi'(\delta) = 0$, $\chi(i) = (\kappa i)\%\delta$, $\chi'(i) = \delta - \chi(i)$ for $0 < i < \delta$. Let $O = (0, 0)$ be the origin, and consider lattice vectors in the origin. The set of *primitive vectors* is the set $\mathcal{V} = \{\vec{u} = \langle u_p, u_q \rangle_{pq} \neq 0 : \ |u_p| \leq \delta, |u_q| \leq \delta, \text{ and } (|u_p|, |u_q|) \neq (\delta, \delta)\} = \{\pm\frac{1}{\delta}(u_p\vec{q} + \chi(u_p)\vec{p})\} \cup \{\pm\frac{1}{\delta}(u_p\vec{q} - \chi'(u_p)\vec{p})\}$. A primitive vector $\vec{u}$ is simply encoded by a word consisting of one letter: $c(\vec{u}) = |p(\vec{u})| \in \mathcal{A}$. In particular, $c(\vec{p}) = 0$ and $c(\vec{q}) = \delta$.

For the general case we decompose any vector $\vec{v}$ as a sum of primitive vectors as follows.

- If $\vec{v} \in R_0(O)$, i.e. $\vec{v} = \langle -v_p, -v_q \rangle_{pq} = \frac{1}{\delta}(-v_q\vec{p} - v_p\vec{q})$ with $v_p, v_q \geq 0$, then

$$\vec{v} = \frac{1}{\delta}(\underbrace{-\vec{p} - \vec{p} \dots - \vec{p}}_{\lfloor \frac{v_q}{\delta} \rfloor \text{times}} + \langle -v_p\%\delta, -v_q\%\delta \rangle_{pq} \underbrace{-\vec{q} - \vec{q} \dots - \vec{q}}_{\lfloor \frac{v_p}{\delta} \rfloor \text{times}})$$

  The encoding $c(\vec{v})$ of $\vec{v}$ is:

$$c(\vec{v}) = \begin{cases} \underbrace{00\dots0}_{\lfloor \frac{v_q}{\delta} \rfloor \text{times}} \underbrace{\delta\delta\dots\delta}_{\lfloor \frac{v_p}{\delta} \rfloor \text{times}} & \text{if } v_p\%\delta = 0 \\ \underbrace{00\dots0}_{\lfloor \frac{v_q}{\delta} \rfloor \text{times}} (v_p\%\delta) \underbrace{\delta\delta\dots\delta}_{\lfloor \frac{v_p}{\delta} \rfloor \text{times}} & \text{if } v_p\%\delta > 0 \end{cases} \tag{2}$$

- If $\vec{v} \in R_1(O)$, i.e. $\vec{v} = \langle v_p, -v_q \rangle_{pq}$ with $v_p, v_q \geq 0$, then

$$\vec{v} = \frac{1}{\delta}(\underbrace{+\vec{q} + \vec{q} \dots + \vec{q}}_{\lfloor \frac{v_p}{\delta} \rfloor \text{times}} + \langle -v_p\%\delta, +v_q\%\delta \rangle_{pq} \underbrace{-\vec{p} - \vec{p} \dots - \vec{p}}_{\lfloor \frac{v_q}{\delta} \rfloor \text{times}}).$$

The encoding $c(\vec{v})$ of $\vec{v}$ is:

$$c(\vec{v}) = \begin{cases} \underbrace{\delta\delta\ldots\delta}_{\lfloor\frac{v_p}{\delta}\rfloor\text{times}}\ \underbrace{00\ldots0}_{\lfloor\frac{v_q}{\delta}\rfloor\text{times}} & \text{if } v_p\%\delta = 0 \\ \underbrace{\delta\delta\ldots\delta}_{\lfloor\frac{v_p}{\delta}\rfloor\text{times}}(v_p\%\delta)\ \underbrace{00\ldots0}_{\lfloor\frac{v_q}{\delta}\rfloor\text{times}} & \text{if } v_p\%\delta > 0 \end{cases} \tag{3}$$

- If $\vec{v} \in R_2(O)$ i.e. $\vec{v} = \langle v_p, v_q\rangle_{pq}$ with $v_p, v_q \geq 0$, then $c(\vec{v})$ is defined by the formula (2).
- If $\vec{v} \in R_3(O)$ i.e. $\vec{v} = \langle -v_p, v_q\rangle_{pq}$ with $v_p, v_q \geq 0$, then $c(\vec{v})$ is defined by the formula (3).

Remark 6 *A vector $\vec{v}$ is uniquely determined by the word $c(\vec{v})$ and the quadrant $R_k(O)$ containing $\vec{v}$.*

Now we are in the position to extend the encoding from the vectors to the oriented path. Let $(M_k)_{1\leq k\leq m} = (M_1, M_2, \ldots, M_m)$ be a sequence of points. We define the encoding of the sequence by:

$$c((M_k)_{1\leq k\leq m}) = c(\overrightarrow{M_1M_2}) \diamond c(\overrightarrow{M_2M_3}) \diamond \ldots \diamond c(\overrightarrow{M_{m-1}M_m}).$$

Therefore a sequence of salient points is encoded by a word of $\mathcal{A}^*$. Notice that the length of the sequence is in general lower than the length of its encoding word, while this latter is equal to the number of primitive vectors in the decomposition.
Actually for our application we are interested in the converse: given a word of $\mathcal{A}^*$, is it an encoding of a sequence of salient points? In other words, we wonder if the encoding provides a bijection between the set of salient points and a word of $\mathcal{A}^*$. Let $\mathbb{S}_j$ be the set of the finite sequences of points $(M_k)_{1\leq k\leq m} = (M_1, M_2, \ldots, M_m)$ such that

$$\begin{aligned} M_2 &\in R_j^\circ(M_1) \cup (R_j(M_1) \cap R_{(j-1)\%4}(M_1) \setminus \{M_1\}) \\ M_{k+1} &\in R_j^\circ(M_k) \text{ for } 1 < k < m-1 \\ M_m &\in R_j^\circ(M_{m-1}) \cup (R_j(M_{m-1}) \cap R_{(j+1)\%4}(M_{m-1}) \setminus \{M_{m-1}\}) \end{aligned}$$

(see Fig. 4). In particular the subsequence $(S_1^{j-1}, S_2^{j-1}, \ldots, S_{m_j}^{j-1})$ of the sequence (1) is in $\mathbb{S}_j$ for $j \in \{0, \ldots, 3\}$. We also denote by $\mathbb{S}_j^N$ the set of sequences $(M) \in \mathbb{S}_j^N$ such that $M_1 = N$. By looking at c as the function mapping $\mathbb{S}_j^N$ into $\mathcal{A}^*$, we can state the following:

Lemma 7 *For every $j \in \{0, 1, 2, 3\}$ and every point $N \in \mathbb{Z}^2$ the function c restricted to $\mathbb{S}_j^N$ is a bijection from $\mathbb{S}_j^N$ to $\mathcal{A}^*$.*

PROOF. We suppose for example $j = 0$. Let w be a word of $\mathcal{A}^*$. This word can be uniquely written as $w = s_1 \diamond s_2 \diamond \ldots s_l$ where every sub-word s_k is given

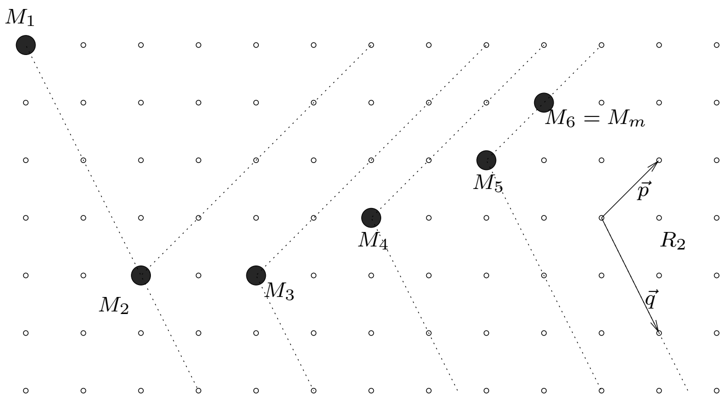

Fig. 4. A sequence of $\mathbb{S}_2$

by (2). By Remark 6 each word s_k encodes an unique vector $\vec{v}_k \in R_0(O)$, and so the unique sequence (M) such that $c((M)) = w$ is given by $M_1 = N$ and $M_{k+1} = M_k + \vec{v}_k$ □

Remark 8 *Let $w = w_1 \dots w_n$ a word of $\mathcal{A}^*$. The sequence $(M_k) \in \mathbb{S}_0^N$ such that $c((M_k)) = w$ is computed by the following algorithm:*

```
1: M ← N; M_1 ← N; k ← 1
2: for i ← 1 to n do
3:    if i > 1 and w_{i−1} ≠ 0 and w_i ≠ δ then
4:       M_k ← M; k ← k + 1
5:    end if
6:    M ← M − (1/δ)(w_i q⃗ + χ(w_i) p⃗)
7: end for
8: M_k ← M
```

The reconstruction of a sequence in $\mathbb{S}_j^N$ with $j \neq 0$ can be done by a similar algorithm (the same with the lines 3 and/or 6 modified).

Definition 9 *The* salient-word *of a Q-convex set E is $c((S))$, where (S) is the ordered sequence of its salient points defined by the equation* (1).

The salient-word of a Q-convex set E is denoted by $c(E)$. Figure 5 illustrates a Q-convex set and its salient-word.

Proposition 10 *A Q-convex set E is completely determined by its salient point $S_1^0 \in \mathcal{S}_0(E)$ minimizing p, the integer* $\min q(E)$ *and the salient-word $c(E) \in \mathcal{A}^*$.*

PROOF. Let $c(E) = w = w_1 w_2 \dots w_n$ be the salient-word of E. Since the jth letter w_j corresponds to one of the vectors in $\{\pm\frac{1}{\delta}(w_j\vec{q}+\chi(w_j)\vec{p})\} \cup \{\pm\frac{1}{\delta}(w_j\vec{q} - \chi'(w_j)\vec{p})\}$, we have $\sum_{j=1}^{n} w_j = 2(\max p(E) - \min p(E)) = lp$. The knowledge of $S_1^0 \in \mathcal{S}_0(E)$, and the integer $\min q(E)$ permits to decompose $w = s_1 \diamond s_2 \diamond s_3 \diamond s_0$

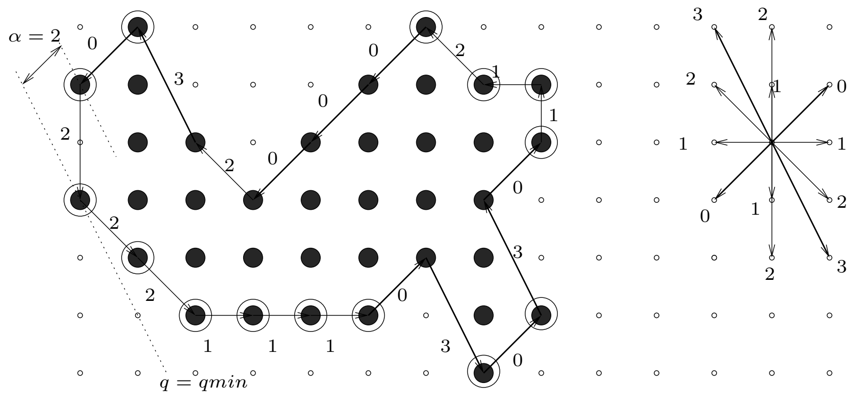

Fig. 5. The salient-word of this Q-convex set is 22211103030112000230.

such that each sub-word s_i necessarily codes a subsequence of the salient points which is in $\mathbb{S}_i$.

Precisely $s_1 = w_1 w_2 \dots w_{n_1}$ is such that $-\sum_{i=1}^{n_1} \chi'(w_j) = \min q(E) - q(S_1^0)$, $s_2 = w_{n_1+1} \dots w_{n_2}$ is such that $\sum_{i=1}^{n_2} w_i = lp$. Let $\gamma = \min q(E) + \sum_{i=n_1+1}^{n_2} \chi(w_i)$. Then $s_3 = w_{n_2+1} \dots w_{n_3}$ and $s_4 = w_{n_3+1} \dots w_n$ are such that $q(S_1^0) + \sum_{i=n_3+1}^{n} \chi(w_i) = \gamma + \sum_{i=n_2}^{n_3} \chi(w_i)$.

As a consequence of Lemma 7, the sequence (S) of salient points satisfies $(S) = (c_1^{-1}(s_1), c_2^{-1}(s_2), c_3^{-1}(s_3), c_4^{-1}(s_0))$ where c_1 is c restricted to $\mathbb{S}_1^{S_1^0}$, and c_k for $k \neq 1$ is c restricted to $\mathbb{S}_k^{N_k}$ where N_k is the last point of $c_{(k-1)\%4}^{-1}(s_{(k-1)\%4})$, and so the salient points only depend on the word w, $\min p(E)$ and S_1^0. But by Corollary 5 any Q-convex set is determined by its salient points, so any Q-convex set E only depends on the word w, $\min q(E)$ and S_1^0. $\square$

4.2 The probabilistic algorithm

The algorithm is derived from the proof of Proposition 10 and Remark 8. In input we are given the length n of the salient-word of the generated set. Since, in combinatorics, sets up to a translation are of interest, we can suppose that the generated set always satisfies $S_1^0 = (0, 0)$. By Proposition 10, it is sufficient to generate an integer $\alpha = q(S_1^0) - \min q(E)$ which is always in the set $\{0, 1, \dots, \lfloor \frac{n\delta}{2} \rfloor - 1\}$ and the word $w = c(E)$ on $\mathcal{A}^*$ of size n in order to produce a set E. After generating α and w, the algorithm reconstructs the salient points coded by w, and then it checks if they characterize the Q-convex set obtained as their Q-convex hull.

Salient points are recognized from w using the algorithm in Remark 8. Consider the decomposition $w = s_1 \diamond s_2 \diamond s_3 \diamond s_0$ of the proof of Proposition 10. The salient points in S^0 and S^1 are computed by scanning the letters of s_1 and then s_2 with the algorithm of Remark 8. Then the salient points in S^2 and S^3

are computed simultaneously reading the word $s_3 \diamond s_0$ from the left hand side for S^2 and from the right hand side for S^3.

The last step of the algorithm consists in checking if the sequence (S^0, S^1, S^2, S^3) is really the sequence of salient points of any Q-convex set. It is sufficient to check if any two consecutive points of the sequence S^k are two k-consecutive salient points. Suppose for example to check if S^0_i and S^0_{i+1} are 0-consecutive. If the quadrant $R^\circ_0(\langle p(S^0_{i+1}), q(S^0_i)\rangle)$ contains a point of E, then it must contain a point of the sequence (S^2). Let j be the largest integer such that $q(S^2_j) < q(S^0_i)$. Suppose that $S^2_j \notin R^\circ_0(\langle p(S^0_{i+1}), q(S^0_i)\rangle)$. Then $p(S^2_j) \geq p(S^0_{i+1})$, so for any other point $S^2_{j'}$ we have $q(S^2_{j'}) \geq q(S^0_i)$ if $j' > j$ and $p(S^2_{j'}) > p(S^2_j) \geq p(S^0_{i+1})$ if $j' < j$. Thus S^0_i and S^0_{i+1} are 0-consecutive if and only if $S^2_j \notin R^\circ_0(\langle p(S^0_{i+1}), q(S^0_i)\rangle)$. This can be done for all the couples (S^0_i, S^0_{i+1}) and since j is decreasing when i is increasing, the time required for the check of the points of S^0 is linear in the size of S^0 and S^2. The same approach can be made for the points of S^1, S^2, S^3.

Now we present the algorithm. If $(M) = (M_1, \ldots, M_n)$ is a finite sequence of points and if N is any point then $\text{appendl}((M), N) = (N, M_1, \ldots, M_n)$, $\text{appendr}((M), N) = (M_1, \ldots, M_n, N)$, $\text{size}((M)) = n$, $\text{supp}((M)) = \{M_1, \ldots, M_n\}$.

GENQ4(n)

Generate a word $w = w_1 w_2 \ldots w_n$ in $\mathcal{A}^*$ of length n.
Generate a non-negative integer $\alpha < \frac{n\delta}{2}$
if $\sum_{i=1}^n w_i$ is not even **then**
  return(FAILURE)
end if
{The algorithm would work without the following check, since in the failure case, the point $(0,0)$ is not in $\mathcal{S}_0(E)$ (see also Remark 12)}
if $w_1 = 0$ and $\alpha \neq 0$ **then**
  return(FAILURE)
end if
$lp \leftarrow \frac{1}{2}\sum_{i=1}^n w_i$; $M \leftarrow (0,0)$; $S^0 \leftarrow (M)$; $i \leftarrow 1$
{Computation of $\mathcal{S}_0(E)$ with the sub-word s_1}
while $q(M) > -\alpha$ and $i \leq n$ **do**
  if $i > 1$ and $w_{i-1} \neq \delta$ and $w_i \neq 0$ **then**
    $(S^0) \leftarrow \text{appendr}((S^0), M)\}$
  end if
  $M \leftarrow M + \frac{1}{\delta}(w_i\vec{q} - \chi'(w_i)\vec{p})$; $i \leftarrow i+1$
end while
{At this point i is equal to the number n_1+1 of the proof of Proposition 10}
if $q(M) \neq -\alpha$ **then**
  return(FAILURE)
end if

$(S^0) \leftarrow \mathrm{appendl}((S^0), M)$; $(S^1) \leftarrow ()$
{Computation of $\mathcal{S}_1(E)$ with the sub-word s_2}
while $p(M) < lp$ and $i \leq n$ **do**
  if (size((S^1)) $= 0$ or $w_{i-1} \neq 0$) and $w_i \neq \delta$ **then**
    $(S^1) \leftarrow \mathrm{appendr}((S^1), M)$, }
  end if
  $M \leftarrow M + \frac{1}{\delta}(w_i\vec{q} + \chi(w_i)\vec{p})$; $i \leftarrow i + 1$
end while
{At this point i is equal to the number n_2+1 of the proof of Proposition 10}
if $p(M) \neq lp$ **then**
  return(FAILURE)
end if
$(S^1) \leftarrow \mathrm{appendr}((S^1), M)$; $M' \leftarrow (0, 0)$; $i' \leftarrow n$; $(S^2) \leftarrow ()$; $(S^3) \leftarrow ()$
{Parallel computation of $\mathcal{S}_2(E)$ and $\mathcal{S}_3(E)$ with the sub-word s_3s_0}
while $i \leq i'$ **do**
  if $q(M) < q(M')$ **then**
    if (size((S^2)) $= 0$ or $w_{i-1} \neq \delta$) and $w_i \neq 0$ **then**
      $(S^2) \leftarrow \mathrm{appendr}((S^2), M)$
    end if
    $M \leftarrow M + \frac{1}{\delta}(-w_i\vec{q} + \chi'(w_i)\vec{p})$; $i \leftarrow i + 1$
  else
    if (size((S^3)) $= 0$ or $w_{i'+1} \neq \delta$) and $w_{i'} \neq 0$ **then**
      $(S^3) \leftarrow \mathrm{appendl}((S^3), M')$}
    end if
    $M' \leftarrow M' + \frac{1}{\delta}(w_{i'}\vec{q} + \chi(w_{i'})\vec{p})$; $i' \leftarrow i' - 1$
  end if
end while
{At this point i is equal to $n_3 + 1$ and i' is equal to n_3}
if $M \neq M'$ **then**
  return(FAILURE)
end if
if size((S^2)) $= 0$ or $w_{i'} \neq \delta$ **then**
  $(S^2) \leftarrow \mathrm{appendl}((S^2), M)$
end if
if size((S^3)) $= 0$ or $w_i \neq \delta$ **then**
  $(S^3) \leftarrow \mathrm{appendl}((S^3), M)$
end if
if CHECKSALIENT(S^0, S^1, S^2, S^3) **then**
  return$((S^0, S^1, S^2, S^3))$
else
  return(FAILURE)
end if

```
CHECKSALIENT((S^0), (S^1), (S^2), (S^3))
  { Check if (S^0, S^1, S^2, S^3) is the sequence of the salient points of E with
  E = Q(supp((S^0)) ∪ supp((S^1)) ∪ supp((S^2)) ∪ supp((S^3))).}
  j ← size((S^2))
  for i ← 1 to size((S^0)) − 1 do
    while q(S^2_j) ≥ q(S^0_i) and j > 1 do
      j ← j − 1
    end while
    if p(S^2_j) < p(S^0_{i+1}) and q(S^2_j) < p(S^0_i) then
      return(FALSE)
    end if
  end for
  j ← size((S^0))
  for i ← 1 to size((S^2)) − 1 do
    while q(S^0_j) ≤ q(S^2_i) and j > 1 do
      j ← j − 1
    end while
    if p(S^0_j) > p(S^2_{i+1}) and q(S^0_j) > p(S^2_i) then
      return(FALSE)
    end if
  end for
  j ← size((S^3))
  for i ← 1 to size((S^1)) − 1 do
    while p(S^3_j) ≤ p(S^1_i) and j > 1 do
      j ← j − 1
    end while
    if p(S^3_j) > p(S^1_i) and q(S^3_j) < p(S^1_{i+1}) then
      return(FALSE)
    end if
  end for
  j ← size((S^1))
  for i ← 1 to size((S^3)) − 1 do
    while p(S^1_j) ≥ p(S^3_i) and j > 1 do
      j ← j − 1
    end while
    if p(S^1_j) < p(S^3_i) and q(S^1_j) > p(S^3_{i+1}) then
      return(FALSE)
    end if
  end for
  return(TRUE)
```

Algorithm GENQ4 either fails (α and w do not correspond to any set), or returns the sequence $((S^0, S^1, S^2, S^3))$ of salient points of a Q-convex set.

This representation of a Q-convex set has the advantage to have a linear size in n and permits to compute the corresponding binary image by using the Q-convex hull (even if this operation takes a quadratic time in n).

Anyway every Q-convex set has the same probability to be generated given by $1/(\lfloor \frac{n\delta}{2} \rfloor (\delta+1)^n)$.

If we consider that an arithmetical operation and the function rand take a constant time, the algorithm GENQ4 takes a linear time. We summarize the results in the following:

Theorem 11 GENQ4(n) *is a uniform random generator of Q-convex sets and has linear running time in* n.

Like in the previous section, we can transform this probabilistic algorithm in an algorithm which never fails:

```
GENQ5(n)
  repeat
    F ← GENQ4(n)
  until F ≠ FAILURE
  return(F)
```

This generator has been implemented using Java language. An example of a generated set is given by Figure 6 (always with the directions $p = x - y$ and $q = 2x + y$.)

Fig. 6. A Q-convex set produced by GENQ5(1000).

The average time-complexity of GENQ5 is $O(\frac{n}{p(n)})$ where $p(n)$ is the success probability of GENQ4. Since $p(n) = \frac{q_n}{\lfloor \frac{n\delta}{2} \rfloor (\delta+1)^n}$, where q_n is the number of Q-convex sets encoded by a word of length n, the complexity of the algorithm depends on q_n. In next sections we will derive an asymptotic formula for q_n for

Q-convex sets with respect to the coordinate directions, while for any couple of directions we will experimentally estimate $p(n)$.

4.3 Estimation of the success probability for the directions $\{x, y\}$

In this subsection we suppose that $p = x$ and $q = y$. We use the notations $l(E) = \max_{M\in E} x_M - \min_{M\in E} x_M$ and $h(E) = \max_{M\in E} y_M - \min_{M\in E} y_M$. As the sets are all considered up-to a translation, all the lattice sets E of this subsection will satisfy $\min_{M\in E} x_M = \min_{M\in E} y_M = 0$.

The set of primitive vectors is $\mathcal{V} = \{(-1, 0), (0, -1), (1, 0), (0, 1)\}$ and hence E has a salient-word of size n if and only if $n = 2(l(E) + h(E))$. We are going to estimate the number of Q-convex sets E with respect to $\{x, y\}$ such that $l(E) + h(E) = m$.

We recall that a polyomino is a lattice set which is 4-connected. If its intersection with any horizontal or vertical line is also a polyomino, it is called HV-convex. A SW-directed convex polyomino (resp. NE-directed) is a HV-convex polyomino containing the source point $(\min_{M\in E} x_M, \min_{M\in E} y_M)$ (resp. $(\max_{M\in E} x_M, \max_{M\in E} y_M)$). A parallelogram polyomino is a polyomino which is SW-directed and NE-directed.

In [10], it is proved that the number c_m of the HV-convex polyominoes E such that $l(E) + h(E) = m$ (with $m \geq 2$) is [1]

$$\begin{aligned} c_m &= (2m + 7)4^{m-2} - 4(2m - 3)\binom{2(m-2)}{m-2} \\ &= 4^m \left(\frac{m}{8} - \frac{\sqrt{m}}{2\sqrt{\pi}} + \frac{7}{16} + O\left(\frac{1}{\sqrt{m}}\right)\right). \end{aligned} \tag{4}$$

An HV-convex polyomino is always a Q-convex set (Proposition 2.3 of [5]), and so to enumerate the Q-convex sets we only have to count the Q-convex sets which are not HV-convex polyominoes.

For this we consider the following generating function

$$\mathrm{QNC}(X, Y) = \sum_{\substack{E \text{ Q-convex} \\ \text{not HV-convex polyomino}}} X^{l(E)} Y^{h(E)}.$$

Consider a Q-convex set which is not a HV-convex polyomino: it has at least

[1] The precise link between the number p_{2n} of [10, Theorem 1.1] and c_m is $c_m = p_{2m+4}$ because here points instead of cells are considered.

two 4-components. Since a horizontal or vertical line cannot intersect two distinct 4-components of E by convexity, we can order the 4-components by increasing abscissa. Let $E_1, E_2, \ldots, E_k$ be the 4-components in this order. Suppose that there are three components $E_{i_1}, E_{i_2}, E_{i_3}$ with $i_1 < i_2 < i_3$ and three points $A_1 \in E_{i_1}$, $A_2 \in E_{i_2}$, $A_3 \in E_{i_3}$ such that $y_{A_1} > y_{A_2} < y_{A_3}$ (or $y_{A_1} < y_{A_2} > y_{A_3}$). By Q-convexity the point $(x_{A_2}, \max(y_{A_1}, y_{A_3}))$ $((x_{A_2}, \min(y_{A_1}, y_{A_3}))$, resp.) is in E and connects two components of E. So there are two kinds of Q-convex sets: the Q-convex sets whose components are in increasing order with respect to the ordinates and those whose components are in decreasing order with respect to the ordinates. By symmetry the number of Q-convex sets for each kind are equal, so we can suppose that the components are of the first kind.

By Q-convexity it is easy to see that E_1 is a SW-directed polyomino, E_k is NE-directed polyomino and that $E_2, \ldots, E_{k-1}$ are parallelogram polyominoes. Conversely if a sequence of polyominoes $E_1, E_2, \ldots, E_k$ has this property and is increasing with respect to the ordinates, then the union of this components is Q-convex (see Fig. 7).

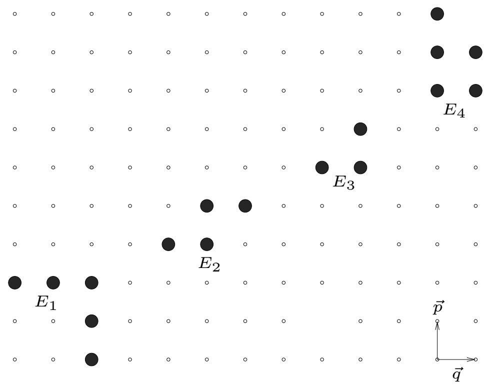

Fig. 7. A Q-convex set which is not an HV-convex polyomino.

We recall that the generating function of the parallelogram polyominoes ([4, page 12])) is:

$$\mathrm{P} = \sum_{E \text{ parallelogram}} X^{l(E)} Y^{h(E)} = \frac{1 - X - Y - \sqrt{1 - 2X - 2Y - 2XY + X^2 + Y^2}}{2XY}$$

and that of the directed polyominoes ([14, formula (25)]) is:

$$\mathrm{D} = \sum_{E \text{ SW-directed}} X^{l(E)} Y^{h(E)} = \sum_{E \text{ NE-directed}} X^{l(E)} Y^{h(E)} = \frac{1}{\sqrt{1 - 2\,X - 2\,Y - 2\,XY + X^2 + Y^2}}$$

The vector joining the NE-extremity of the component E_i and the SW-extremity of the component E_{i+1} is any lattice vector in $R_2^\circ(O)$. The generating function of these vectors is:

$$\mathrm{J} = \sum_{x,y>0} X^x Y^y = \frac{XY}{(1-X)(1-Y)}.$$

Hence the generating function of the Q-convex sets which have $k+2$ 4-components is $2\,\mathrm{D}\,\mathrm{J}\,(\mathrm{P}\,\mathrm{J})^k\,\mathrm{D}$. (The factor 2 comes from the two kinds of Q-convex-sets).

The generating function of the Q-convex sets which are not HV-convex polyominoes is:

$$\begin{aligned}\mathrm{QNC} &= \sum_{k=0}^{\infty} 2\,\mathrm{D}\,\mathrm{J}\,(\mathrm{P}\,\mathrm{J})^k\,\mathrm{D} \\ &= 2\mathrm{D}\,\mathrm{J}\,\frac{1}{1-\mathrm{P}\,\mathrm{J}}\,\mathrm{D} \\ &= \frac{4XY}{R(X,Y)\,(1-2\,X-2\,Y-2\,XY+X^2+Y^2)}\end{aligned}$$

where $R(X,Y) = \left(1-Y-X+2\,XY+\sqrt{1-2\,X-2\,Y-2\,XY+X^2+Y^2}\right)$.
If we substitute X and Y by T we obtain:

$$\mathrm{QNC}(T,T) = \sum_{\substack{E \text{ Q-convex} \\ \text{not HV-convex polyomino}}} T^{l(E)+h(E)} = \frac{4T^2}{\left(1-2\,T+2\,T^2+\sqrt{1-4\,T}\right)(1-4\,T)}.$$

The singularity of QNC with the smallest modulus is $\frac{1}{4}$ and the asymptotic expansion around $\frac{1}{4}$ is $QNC(T,T) = \frac{2}{5(1-4t)} + O(\frac{1}{\sqrt{1-4t}})$ so the number a_m of Q-convex sets E which are not HV-convex polyominoes such that $l(E)+h(E) = m$ satisfies:

$$\mathrm{a}_m = 4^m\left(\frac{2}{5} + O(\frac{1}{\sqrt{m}})\right).$$

Finally the number b_m of Q-convex sets E such that $l(E)+h(E) = m$ satisfies:

$$\mathrm{b}_m = q_{2m} = \mathrm{c}_m + \mathrm{a}_m = 4^m\left(\frac{m}{8} - \frac{\sqrt{m}}{2\sqrt{\pi}} + \frac{67}{80} + O\left(\frac{1}{\sqrt{m}}\right)\right).$$

As a result, we have an asymptotic expansion of the success-probability of the generator GENQ4 for any even size n:

$$p(n) = \frac{\mathrm{b}_{\frac{n}{2}}}{\frac{n}{2}2^n} = \frac{1}{8} - \frac{1}{\sqrt{2\pi n}} + \frac{67}{40n} + O(\frac{1}{n\sqrt{n}}) \qquad (5)$$

Remark 12 *We could multiply this success probability by* $4\frac{\frac{n}{2}}{\frac{n}{2}+1}$ *by modifying a little bit algorithm* GENQ4*:*

- *The number of* 0*'s of* w *is always even when the algorithm succeeds, so the last letter* w_n *of* w *can be omitted, multiplying by two the success probability.*
- *When* $\alpha \neq 0$ *the first letter* w_1 *is always* 1*, so we can omit this letter if we add the possibility* $\alpha = \frac{n}{2}$ *which in fact codes* $\alpha = 0$ *and* $w_1 = 0$*. This trick multiplies the success probability by* $2\frac{\frac{n}{2}}{\frac{n}{2}+1}$*.*

In fact this shrewdness is used in the generator of HV-convex polyominoes of [11], and with them, GENQ4 *has the same asymptotic success-probability:* $\frac{1}{2}$*.*

Notice that for a general set of directions, the first improvement cannot be done because there are more than two letters, and that the second improvement multiplies the success probability by $\frac{(\delta+1)\lfloor\frac{n}{2}\rfloor}{\lfloor\frac{n}{2}\rfloor+1}$ *by coding the case* $\alpha = 0$ *and* $w_1 = i \neq 0$ *by* $\alpha = \lfloor\frac{n}{2}\rfloor + i$*.*

Remark 13 *If a couple of directions* (p, q) *satisfies* $\delta = \det(p, q) = 1$*, then there is a linear bijection from* $\mathbb{Z}^2$ *to* $\mathbb{Z}^2$ *which maps* (p, q) *in* (x, y)*, so the estimation* (5) *is also true in this case.*

Remark 14 *If* n *is odd, since* $n = 2(l(E) + h(E))$ *we have* $p(n) = 0$*. This property is not true for a general couple of directions (see Fig. 8).*

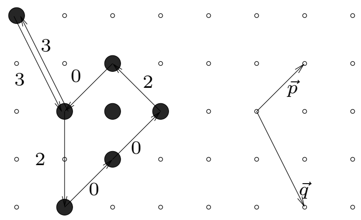

Fig. 8. The salient-word of this Q-convex set (3200203) has an odd size

4.4 Experimental estimation of the success probability

In this section we are going to estimate $p(n)$, the success probability of GENQ4, since the average time-complexity of GENQ5 depends on $p(n)$.

The algorithm GENQ4 behaves as a Bernoulli trial and hence the number of calls to GENQ4 to halting GENQ5 has a geometric distribution with same parameter $p(n)$. So, we decided to estimate $p(n)$ as a parameter of the Bernoulli distribution. In this case the mean is $p(n)$ and the variance is $p(n)(1 - p(n))$, and an estimation of $p(n)$ is provided by the success frequency, that is, number of successes on number of attempts. It is possible to show that the success frequency maximizes the joint probability mass function for the Bernoulli distribution, since the data are independent. (So it gives the maximum probability of obtaining the observed data of our experiment). In order to assess the precision of this estimation we construct a confidence interval for $p(n)$. The test data were obtained as follows. We have chosen three couples of directions having different values of δ: $\{x, y\}$, $\{x+y, x-y\}$, $\{x-y, 2x+y\}$. For every couple of directions, GENQ4 was executed $l = 10000$ times on an Athlon 2000+ for each of the different sizes $n = 10, 20, 50, 100, 200, 201, 400 + 200k$ of the generated word w. The results are summarized in the Figures 9 and 10.

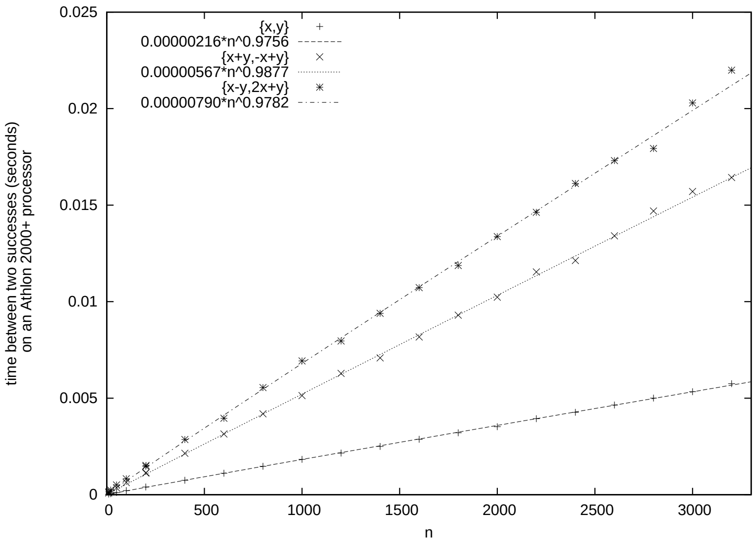

Fig. 9. The time between two successes of the generator GENQ4 in function of n.

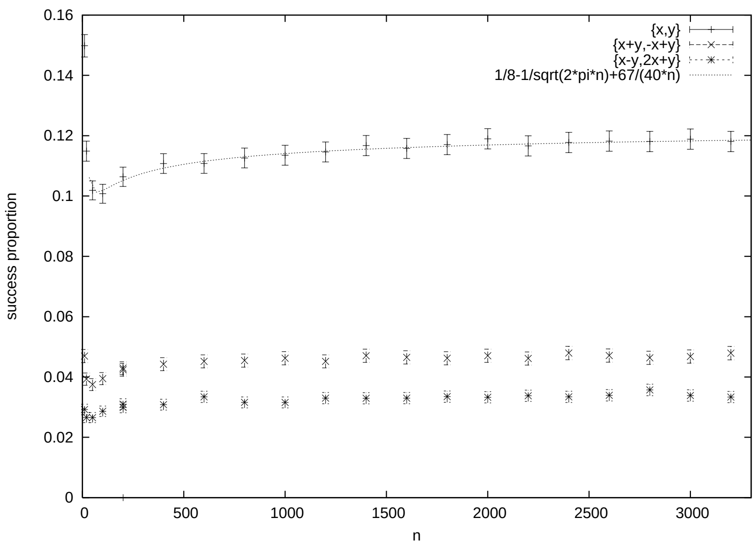

Fig. 10. The success proportion of the generator GENQ4 in function of n.

Figure 9 shows the average time between two successes of the generator

GENQ4 in function of n. So it gives an estimation of the time-complexity of the generator GENQ5. The curves show the power-regression of this data. Since the exponent of this approximation is near 1, it suggests that the time-complexity of the generator GENQ5 is linear.

Figure 10 illustrates the ratio $\frac{k}{l}$ of the number k of successes on the number $l = 10000$ of attempts. The intervals around each points are the 99% confidance intervals, that is, the proportion of intervals covering $p(n)$. These intervals can be computed as the success frequencies are asymptotically normally distributed, with parameters mean $p(n)$ and variance $\sigma^2 = \frac{p(n)(1-p(n))}{l}$. So, for large l, the 99% confidence interval for $p(n)$ is given by $[\frac{k}{l} - 2.576\sqrt{\frac{\frac{k}{l}(1-\frac{k}{l})}{l}}, \frac{k}{l} + 2.576\sqrt{\frac{\frac{k}{l}(1-\frac{k}{l})}{l}}$. We refer to [13, p.367-371] for an exhaustive treatment of estimation of parameters in probability distributions.

The results suggest that $p(n)$ tends to a constant (which looks to be approximatively 0.048 for the directions $\{x+y, x-y\}$, and 0.033 for $\{x-y, 2x+y\}$). We know that $p(n) = \frac{q_n}{\lfloor \frac{n\delta}{2} \rfloor (\delta+1)^n}$ so we can also use the estimation of $p(n)$ to derive an estimation of the number q_n of Q-convex sets encoded by a word of length n:

Conjecture 15 *If $\delta \geq 2$ then the ratio $\frac{q_n}{n(\delta+1)^n}$ tends to a non-null constant when n tends to infinity.*

For $\delta = 1$, we have proved in the last subsection that $q_{2m+1} = 0$ and $\frac{q_{2m}}{m(\delta+1)^{2m}}$ tends to $\frac{1}{8}$.

5 Conclusion and perspectives

The main contribution of this paper is a uniform probabilistic generator for the class of Q-convex sets generalizing the algorithm in [11]. Our generator runs in time linear in the size of the word coding its salient points.

This algorithm allowed to estimate the number of elements of this class according to the size of the word coding the salient points. While an asymptotic formula for the case of coordinate directions is provided, the enumeration in the general case is an open problem. We conjectured that $q_n \sim Cn(\delta+1)^n$ where C only depends on the pair of directions.

An interesting perspective would consist in a generalization of the generator for Q-convex sets along more than two directions ([6,5]) because the class of convex sets can be seen as the limit of the class of Q-convex sets when the set of directions tends to the set of all the lattice directions, and a few papers have studied the convex sets from a combinatorial point of view ([12,1]).

Dedication

In October 2000, Alberto Del Lungo told us that we should try to extend the generator of [11] to the class of Q-convex sets which we had just introduced.

This paper is dedicated to his Memory. His advices and enthusiasm always made us believe in our work.

References

[1] I. Bárány. The limit shape of convex lattice polygons. *Discrete Comput. Geom.*, 13(3-4):279–295, 1995.

[2] E. Barcucci, A. Del Lungo, M. Nivat, and R. Pinzani. Reconstructing convex polyominoes from horizontal and vertical projections. *Theoret. Comput. Sci.*, 155(2):321–347, 1996.

[3] E. Barcucci, R. Pinzani, and R. Sprugnoli. The random generation of directed animals. *Theoret. Comput. Sci.*, 127(2):333–350, 1994.

[4] M. Bousquet-Mélou. A method for the enumeration of various classes of column-convex polygons. *Discrete Math.*, 154(1-3):1–25, 1996.

[5] S. Brunetti and A. Daurat. An algorithm reconstructing convex lattice sets. *Theoret. Comput. Sci.*, 304(1-3):35–57, 2003.

[6] S. Brunetti, A. Daurat, and A. Del Lungo. Approximate X-rays reconstruction of special lattice sets. *Pure Math. Appl.*, 11(3):409–425, 2000.

[7] A. Daurat. Salient points of Q-convex sets. *Int. J. Pattern Recognition & Artificial Intelligence*, 15(7):1023–1030, 2001.

[8] A. Daurat. Determination of Q-convex sets by X-rays. *Theoret. Comput. Sci.*, 332(1-3):19–45, 2005.

[9] A. Daurat, A. Del Lungo, and M. Nivat. Median points of discrete sets according to a linear distance. *Discrete Comput. Geom.*, 23(4):465–483, 2000.

[10] M. P. Delest and G. Viennot. Algebraic languages and polyomino enumeration. *Theoret. Comput. Sci.*, 112:65–79, 1984.

[11] W. Hochstättler, M. Loebl, and C. Moll. Generating convex polyominoes at random. *Discrete Math.*, 153(1-3):165–176, 1996.

[12] A. Ivić, J. Koplowitz, and J. Žunić. On the number of digital convex polygons inscribed into an (m,m)-grid. *IEEE Trans. Inform. Theory*, 40(5):1681–1686, 1994.

[13] A. M. Law and W. D. Kelton. *Simulation modeling and analysis.* McGraw-Hill, New York, second edition, 1991.

[14] K. Y. Lin and S. J. Chang. Rigorous results for the number of convex polygons on the square and honeycomb lattices. *J. Phys. A*, 21(11):2635–2642, 1988.

[15] J. O'Rourke. *Computational geometry in C.* Cambridge University Press, Cambridge, second edition, 1998.

[16] C. Ronse. A bibliography on digital and computational convexity. *IEEE trans. on PAMI*, 11(2):181–190, 1989.

[17] C. Valenti. An experimental study of the stability problem in discrete tomography. In *Proc. of IWCIA 2003*, volume 12 of *Electronic Notes in Discrete Mathematics*. Elsevier, 2003.