On computing the degree of convexity of polyominoes*

Stefano Brocchi
Dipartimento di Matematica e Informatica 'U. Dini'
Università di Firenze, Italy
stefano.brocchi@unifi.it

Giusi Castiglione
Dipartimento di Matematica e Informatica
Università di Palermo, Italy
giusi@math.unipa.it

Paolo Massazza
Dipartimento di Scienze Teoriche e Applicate
Università dell'Insubria, Italy
paolo.massazza@uninsubria.it

Abstract

In this paper we present an algorithm which has as input a convex polyomino P and computes its degree of convexity, defined as the smallest integer k such that any two cells of P can be joined by a monotone path inside P with at most k changes of direction. The algorithm uses space $O(m + n)$ to represent a polyomino P with n rows and m columns, and has time complexity $O(\min(m, rk))$, where r is the number of corners of P. Moreover, the algorithm leads naturally to a decomposition of P into simpler polyominoes.

1 Introduction

A polyomino (cf. [12]) is a finite and connected union of unitary squares (called cells) in the plane $Z \times Z$, considered up to translations. The number of cells of a polyomino is its *area* and the length of the boundary is its *perimeter*. One of the most investigated class of polyominoes is the class of *hv-convex* (or, more simply, convex) *polyominoes* constituted by polyominoes whose intersection with any vertical or horizontal line is connected. The notion of convexity is well defined in the continuous case, whereas in the discrete case several definitions have been considered. In particular, in [8] the authors provided a

---

linear algorithm to detect whether an hv-convex polyomino is convex with respect to the definition of convexity for discrete regions given in [13].

In literature, the class of convex polyominoes and some of its subclasses have been studied under several points of view as, for instance, their enumeration and exhaustive generation [10, 2, 9].

In [6] the authors observe that in a convex polyomino each pair of cells can be joined by a monotone path (a path that contains only two kinds of steps among East, North, West and South steps). Then they introduce the *degree of convexity* of a convex polyomino as the number of changes of direction needed to connect any pair of cells. A convex polyomino is said to be *k-convex* if it has a degree of convexity k, i.e. every pair of its cells can be connected by means of a monotone path, internal to the polyomino, with at most k changes of direction. The classification of convex polyominoes with respect to their degree of convexity provides us an index describing the regularity and the shape of the polyomino. Recent studies have considered the properties of these objects under many aspects, attempting to achieve knowledge about this class and to extend the well-known results about convex polyominoes.

Many of these studies have been approached by taking into consideration a specific degree of convexity, starting from $k = 1$ (L-convex polyominoes, cf. [4, 14]) and $k = 2$ (Z-convex polyominoes, cf. [11]). Very recently, in [15] the authors have shown an asymptotic estimate of a lower bound for the number of k-convex polyominoes with perimeter p, for a generic k. This results to be $\mu(k) \, p \, 4^p$, where $\mu(k)$ is a fractional function of k that has not been explicitly determined.

In this paper we give an efficient algorithm that is able to determine the degree of convexity k of a convex polyomino P in time $O(\min(m, rk))$, where m is the number of columns of P and r the number of its corners (cells of P with two adjacent sides belonging to the boundary).

We take advantage of the property that the degree of convexity of a polyomino depends only on the paths connecting some cells on the boundary of the polyomino; starting from these cells, we can identify with ease some areas of P where every pair of cells can be connected by a given number of changes of direction. Extending these areas to the whole P, we can identify pairs of cells that require the maximum number of changes of direction to be connected, and hence detect the degree of convexity. The method is based on the properties studied at first in [3], where the class of L-convex polyominoes was recognized by a bidimensional language thanks to a similar procedure, extended in this work to generic k-convex polyominoes.

From the proposed algorithm, we directly obtain a method for the decomposition of a k-convex polyomino in simpler shapes. Such a procedure may be useful for future research in a variety of situations. As an example, we may think that by being able to see a k-convex polyomino as a $(k-1)$-convex polyomino combined with some extra cells that are easily determinable, we could define a recursive generation algorithm. Further, the decomposition of a polyomino in simpler polyominoes has been used to define an efficient encoding, for example in image compression [1], and our procedure could find similar motivations.

2 Notation and preliminaries

Let P be a convex polyomino with a minimal bounding rectangle of size $m \times n$. We number the m columns and the n rows from left to right and from bottom to top, respectively. Thus, we consider the bottom (resp. top) row of P as its first (resp. last) row, and the leftmost (resp. rightmost) column of P as its first (resp. last) column.

By (i, j) we denote a cell in the i-th row and j-th column of P. We represent the convex polyomino by means of four integer vectors. The vectors $F = [f_1, \ldots, f_n]$ and $L = [l_1 \ldots l_n]$ contain the positions (i.e. the indices of columns) of the first and of the last cell of the polyomino on each row, respectively. Symmetrically, $F' = [f'_1, \ldots, f'_m]$ and $L' = [l'_1, \ldots, l'_m]$ provide the positions of the first and of the last cell in each column, respectively. Given $c = (i, j) \in P$ we denote by $Op(c)$ the cell (l'_j, l_i); c and $Op(c)$ identify a rectangle not necessarily in P. A *path* connecting two cells a and b of a polyomino P, is a sequence $(a = (i_1, j_1), (i_2, j_2), \ldots, (i_r, j_r) = b)$ of distinct edge-adjacent cells all belonging to P. The step $((i_s, j_s), (i_{s+1}, j_{s+1}))$ is called:

- an *East* step if $j_{s+1} = j_s + 1$ and $i_{s+1} = i_s$;

- a *North* step if $i_{s+1} = i_s + 1$ and $j_{s+1} = j_s$;

- a *West* step if $j_{s+1} = j_s - 1$ and $i_{s+1} = i_s$;

- a *South* step if $i_{s+1} = i_s - 1$ and $j_{s+1} = j_s$.

A path, is *monotone* if it is made of steps in only two directions: North and East (NE-path), North and West (NW-path), South and East (SE-path) or South and West (SW-path). A polyomino is convex if and only if every pair of its cells is joined by a monotone path (see [6]). Given $k \in \mathbb{N}$, a convex polyomino is said to be *k-convex* if every pair of its cells can be joined by a monotone path with at most k changes of direction. Obviously, a k-convex polyomino is also h-convex for every $h > k$. We define the *degree of convexity* of a convex polyomino P as the smallest $k \in \mathbb{N}$ such that P is k-convex. Figure 1(a) shows a 2-convex polyomino, whereas Figure 1(b) represents a polyomino which is 3-convex, but not 2-convex, since there exist two cells (highlighted in the figure) that can be connected only by paths with at least three changes of direction. Hence its degree of convexity is 3.
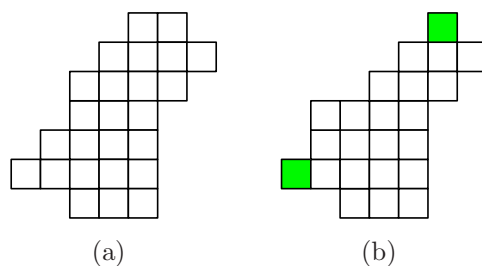

(a) (b)

Figure 1: (a) A 2-convex polyomino, (b) a 3-convex polyomino.

Given two cells $a, b \in P$, we write $a \nearrow b$ ($a \nwarrow b$) if there is a NE-path (NW-path) from a to b, and we define the function $D_{\mathrm{NE}}(a,b)$ ($D_{\mathrm{NW}}(a,b)$) which returns the smallest number of changes of direction in any NE-path (NW-path) connecting a to b, called the NE-distance of b from a. We let $D_{\mathrm{NE}}(a,b) = \infty$ if there is not a NE-path from a to b. Note that if P has degree of convexity k, then one can find $a, b \in P$ such that $D_{\mathrm{NE}}(a,b) = k$ or $D_{\mathrm{NW}}(a,b) = k$. A NE-path (NW-path) from a to b is called *minimal* if it has $D_{\mathrm{NE}}(a,b)$ ($D_{\mathrm{NW}}(a,b)$) changes of direction.

We say that a convex polyomino P has NE-degree of convexity (resp. NW-degree of convexity) k, and we write $D_{\mathrm{NE}}(P) = k$ (resp. $D_{\mathrm{NW}}(P) = k$), if for any $a, b \in P$ such that $a \nearrow b$ (resp. $a \nwarrow b$) one has $D_{\mathrm{NE}}(a,b) \leqslant k$ (resp. $D_{\mathrm{NW}}(a,b) \leqslant k$). It is clear that the degree of convexity of a polyomino P is $\max_{a,b \in P}\{D_{\mathrm{NE}}(a,b), D_{\mathrm{NW}}(a,b)\}$. We say that a cell $(i,j) \in P$ is a *SW-corner* (*SE-corner*) of P if and only if $(i-1,j), (i,j-1) \notin P$ ($(i-1,j),(i,j+1) \notin P$). Analogously, $(i,j) \in P$ is a *NE-corner* (*NW-corner*) of P if and only if $(i+1,j),(i,j+1) \notin P$ ($(i+1,j),(i,j-1) \notin P$). In the following we denote by $C_{\mathrm{SW}}(P)$ (resp. $C_{\mathrm{SE}}(P)$, $C_{\mathrm{NE}}(P)$ and $C_{\mathrm{NW}}(P)$) the set of SW-corners (resp. SE-corners, NE-corners, NW-corners) of P. Lastly, the following definition plays a crucial role for computing the degree of convexity.

Definition 1. Given P, we denote by $P_{(i,j)} \subseteq P$ the polyomino consisting of the cells $(i', j') \in P$ with $i' > i$ and $j' > j$.

3   Properties

In this section we give some basic properties and lemmas on which our algorithm is based.

Property 2. *Let $a = (i,j)$ and b be two cells of P such that $a \nearrow b$. If $a' = (i, j-1)$ ($a' = (i-1, j)$) belongs to P then $D_{NE}(a,b) \leqslant D_{NE}(a',b) \leqslant D_{NE}(a,b) + 1$.*

Proof. Let $r = D_{\mathrm{NE}}(a,b)$. Suppose that $a' = (i, j-1) \in P$ and $D_{\mathrm{NE}}(a',b) < r$. Consider the second cell a'' in a minimal path from a' to b. Obviously $a'' \neq a$, otherwise one would have $D_{\mathrm{NE}}(a,b) < r$. So, one has $a'' = (i+1, j-1)$. Eventually, after p North steps, the path crosses column j with a change of direction in $(i+p, j-1)$, reaching a cell $b' = (i+p, j)$ with $D_{\mathrm{NE}}(b',b) < r-1$. Since $D_{\mathrm{NE}}(a,b') = 0$ one has the contradiction $D_{\mathrm{NE}}(a,b) < r$. The relation $D_{\mathrm{NE}}(a',b) \leqslant D_{\mathrm{NE}}(a,b) + 1$ follows by considering an East step from a' to a and the minimal path from a to b. A similar reasoning holds if $a' = (i-1, j) \in P$. $\qquad \square$

Because of symmetry, we also have:

Property 3. *Let a and $b = (i,j)$ be two cells of P such that $a \nearrow b$. If $b' = (i, j+1)$ ($b' = (i+1, j)$) belongs to P then $D_{NE}(a,b) \leqslant D_{NE}(a,b') \leqslant D_{NE}(a,b) + 1$.*

The previous two properties imply that, in order to determine the degree of convexity of a polyomino P, it is sufficient to consider the distance between opposite corners of P.

Property 4. *Let P be a convex polyomino with $D_{NE}(P) = k$ (resp. $D_{NW}(P) = k$). Then, there are $a' \in C_{SW}(P)$, $b' \in C_{NE}(P)$, (resp. $a'' \in C_{SE}(P)$ and $b'' \in C_{NW}(P)$) such that $D_{NE}(a',b') = k$ (resp. $D_{NW}(a'',b'') = k$).*

We exploit Property 4 to define an efficient algorithm for determining the degree of convexity of a convex polyomino P. Informally, the idea consists of labelling some suitable cells of P with an integer that represents the smallest number of changes of direction necessary to join a given corner to a cell above and to the right of the labelled cell. To this aim, we introduce the following:

Definition 5. A cell $c = (i, j) \in P$ is labelled e_s, with $e \in \mathbb{N}$ and $s \in C_{\text{SW}}(P)$, iff

    a) for every $d \in P_c$ one has $D_{\text{NE}}(s, d) > e$;

    b) for every $d \in P \setminus P_c$ such that $s \nearrow d$ one has $D_{\text{NE}}(s, d) \leqslant e$.

Figure 2(a) shows a cell with label 2 whereas Figure 2(b) illustrates the division of P into regions of increasing distance D. The previous definition admits some interesting
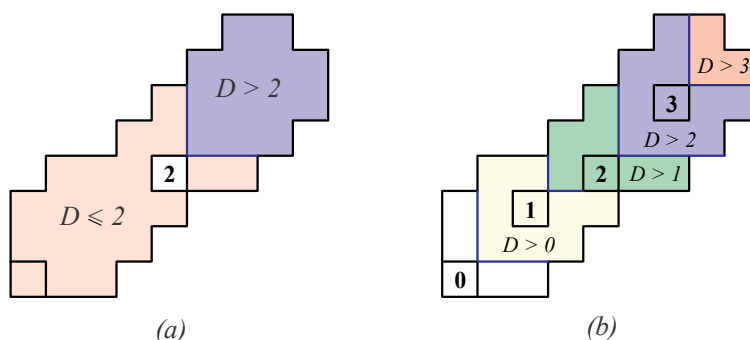

(a)   *(b)*

Figure 2: A 4-convex polyomino with one SW-corner.

properties as shown in the following lemmas.

Lemma 6. *Let P be a convex polyomino. Then any $c \in C_{SW}(P)$ has label 0_c.*

Proof. Let $c = (i, j) \in C_{\text{SW}}(P)$. A cell d in $P \setminus P_c$ such that $c \nearrow d$ is either in column j or in row i, then $D_{\text{NE}}(c, d) = 0$. Moreover, at least one change of direction is needed to reach a cell $d = (i', j') \in P_c$ since $i' > i$ and $j' > j$. $\square$

Lemma 7. *Let P be a convex polyomino with a cell (i, j) labelled e_s. Then, for any $p > 0$ such that $(i, j + p) \in P$ (resp. $(i + p, j) \in P$) one has $D_{NE}(s, (i, j + p)) = e$ (resp. $D_{NE}(s, (i + p, j)) = e$) and every minimal NE-path from s to $(i, j + p)$ (resp. $(i + p, j)$) contains $(i, j + p - 1)$ (resp. $(i + p - 1, j)$).*

Proof. Without loss of generality we consider $c = (i, j+p) \in P$. Suppose $D_{\text{NE}}(s, c) < e$ and let $\bar{c} = (\bar{\imath}, j)$ be the the first cell belonging to column j in a minimal NE-path from s to c. Since at least one change of direction is needed to reach c from $\bar{c}$, one has $D_{\text{NE}}(s, \bar{c}) < e - 1$ and then $D_{\text{NE}}(s, (i + 1, j + q)) < e$ for all $q \leqslant p$ such that $(i + 1, j + q) \in P$, whereas the label of (i, j) implies $D_{\text{NE}}(s, (i + 1, j + q)) > e$. Lastly, consider a cell $(i + 1, j + q) \in P$ with $q \leqslant p$. If the last step in a minimal NE-path from s to $(i, j + q)$ were a North-step, then we would have $D_{\text{NE}}(s, (i + 1, j + q)) = e$. $\square$

Lemma 8. *Let P be a convex polyomino with a cell $c = (i, j)$ labelled e_s. If $Op(c)$ belongs to P then it can be labelled $(e + 1)_s$.*

Proof. First, recall that $Op(c) = (l'_j, l_i)$ where $L = [l_1 \ldots l_n]$ and $L' = [l'_1, \ldots, l'_m]$ are the two vectors providing the position of the last cell on each row and on each column, respectively. By applying Lemma 7 we may use the cell c labelled e_s to detect another cell that can be labelled: every cell $c' = (\bar{\imath}, \bar{\jmath}) \in P$ with $\bar{\imath} > i$ and $j < \bar{\jmath} \leqslant l_i$ can be reached by extending a minimal path from s to $(i, \bar{\jmath})$ with one more change of direction, that is, $D_{\mathrm{NE}}(s, c') = e + 1$. Similarly, a cell $c'' = (\hat{\imath}, \hat{\jmath}) \in P$ with $i < \hat{\imath} \leqslant l'_j$ and $\hat{\jmath} > l_i$ satisfies $D_{\mathrm{NE}}(s, c'') = e + 1$. Lastly, consider the cell $Op(c)$. A path from s to a cell $d \in P_{Op(c)}$ has necessarily a change of direction at $(\tilde{\imath}, \tilde{\jmath})$, with either $\tilde{\imath} > l'_j$ and $j < \tilde{\jmath} \leqslant l_i$ or $i < \tilde{\imath} \leqslant l'_j$ and $\tilde{\jmath} > l_i$. As a consequence, one has $D_{\mathrm{NE}}(s, d) > e + 1$. $\qquad\square$

Figure 3 contains a graphical representation of the property described by the previous lemma.
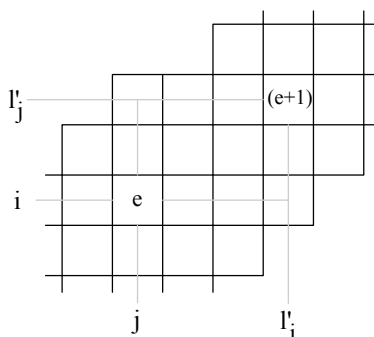

Figure 3: Derivation of new labels using the L and L' vectors.

4 The algorithm

Our algorithm to determine the degree of convexity of a convex polyomino uses the properties of labelled cells to compute the smallest number of changes of direction needed to join any two cells in P. Actually, the described procedure computes the *NE-degree of convexity* of a polyomino P. A similar procedure can compute the *NW-degree of convexity*, and by taking the maximum of the two values we obtain the degree of convexity of P.

The algorithm exploits Lemma 6 to label all the SW-corners and then continues to label other cells of P according to Lemma 8. When no more labels can be computed, the largest ones are examined to determine the NE-degree of convexity. Not all the cells that could be labelled are computed, but we show in the following that, thanks to the properties described in the previous section, the labelled cells which are computed, starting from the SW-corners, are sufficient to obtain the degree of convexity, as the SW-corners determine the maximal NE-distance.

In the algorithm's labelling, for the sake of simplicity, we omit the subscript indicating a corner associated with a label, since we are looking for the largest distance and we are not interested in finding the cells that originate it.

From here on, given a cell $c = (i, j)$ in P, we indicate by $\hat{c}$ the cell $\mathrm{Op}(c)$.

Algorithm 1 Degree of convexity.

1: **for** $c \in C_{\mathrm{SW}}(P)$ **do** assign label 0 to c; **end for**
2: **for** each c labelled e s.t. $\hat{c} \in P, \hat{c} \neq c$ and $\hat{c}$ is not labelled or has label less than $e + 1$ **do**
3:    assign label $e + 1$ to $\hat{c}$;
4: **end for**
5: let k be the maximum label of the labelled cells c with $P_c \neq \emptyset$;
6: return $k + 1$;

Figure 4 illustrates a run of Algorithm 1 on a simple polyomino.
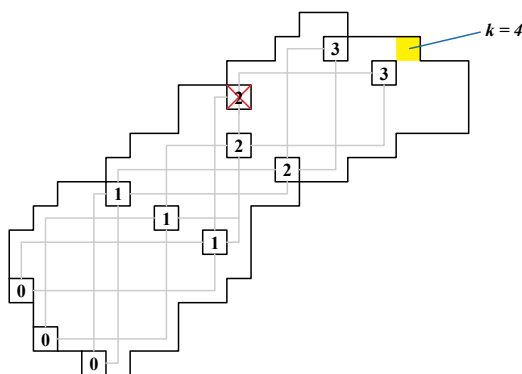

Figure 4: An example of run of the algorithm.

Theorem 9. *Algorithm 1 computes the NE-degree of convexity of a convex polyomino.*

Proof. The algorithm exploits Lemma 6 and assigns a label 0 to all SW-corners (the for-loop at line 1). Then, Lemma 8 is used to iteratively label cells of P (the for-loop at lines 2-4). Note that if a cell $\hat{c}$ is already labelled and a greater label is found for $\hat{c}$ (with respect to a different corner), then the old label can be discarded since it can not lead to the degree of convexity thanks to Lemma 7. At the end of the loop, for all cells c with label e one has that either $\hat{c}$ is not in P or $\hat{c}$ is in P and has a label greater than e. Now, observe that, by Lemma 6, for any cell c with label e and such that $\hat{c} \notin P$ one has $D_{\mathrm{NE}}(s, d) = e + 1$ for a suitable $s \in C_{\mathrm{SW}}$ and for all $d \in P_c$. Therefore, since any NE-corner d is either in the same row of c or in the same column or in P_c, the NE-degree of convexity is just the largest label of a labelled cell c with $P_c \neq \emptyset$, incremented by 1. $\qquad \square$

4.1 Implementation and Time Complexity

In this section we show how to develop an efficient implementation of Algorithm 1. To this aim, we keep track in a queue Q of the labelled cells c for which the successors $\hat{c}$ have not yet been computed. Initially, the queue contains all SW-corners, that is, cells with label 0. Then, the main loop removes a cell c from Q, and inserts $\hat{c}$ in Q with label 1 if it belongs to P. More general, it is straightforward to see that at any time Q may contain only a sequence of cells with label e, eventually followed by a sequence of cells with label $e+1$. By storing the label of the latest cell c inserted in Q such that $P_c \neq \emptyset$, we can immediately obtain the NE-degree of convexity once we exit the loop.

Note that, since P is represented by the two vectors L and L', at every iteration we can check in constant time the conditions $\hat{c} \in P$ and $P_c \neq \emptyset$. Furthermore, to improve the time complexity, we make use of the following definition.

Definition 10. Consider two cells c and d labelled e_a and e'_b, respectively. If for all $f \in P_d$ there is $g \in P_c$ such that $D_{\mathrm{NE}}(a, g) \geqslant e'$ and $D_{\mathrm{NE}}(a, g) \geqslant D_{\mathrm{NE}}(b, f)$, then we write $d \preceq_{\mathrm{NE}} c$.

It should be clear that if $d \preceq_{\mathrm{NE}} c$, then the cell d can be discarded from the computation, as it does not contribute to the degree of convexity of P (see Lemmas 7 and 8). Thus, the following properties provide us with rules which can be used for reducing the number of labelled cells computed by the algorithm.

Property 11. *Let c and d be two cells with labels e_a and e_b, respectively. If $P_d \subseteq P_c$ then $d \preceq_{NE} c$.*

Proof. Let $P_d \neq \emptyset$ (if $P_d = \emptyset$ the result holds immediately). Since $P_d \subseteq P_c$, for each cell $f \in P_d$ one has $a \nearrow f$ and so it is sufficient to prove that $D_{\mathrm{NE}}(a, f) \geqslant D_{\mathrm{NE}}(b, f)$.

Suppose there is $f \in P_d$ such that $D_{\mathrm{NE}}(a, f) = p$ and $D_{\mathrm{NE}}(b, f) = q$ with $p < q$. If $d = (i, j) \in P_c$, consider the first cell x in a minimal NE-path from a to f such that $x = (i, j_1)$ or $x = (i_1, j)$ with $j_1 \geqslant j$, $i_1 \geqslant i$. Since $D_{\mathrm{NE}}(a, x) \geqslant e + 1$, the number of changes of direction in a path from x to f is at most $D_{\mathrm{NE}}(x, f) \leqslant p - e - 1$. Thanks to the property described by Lemma 7, this implies the existence of a path from b to f which contains x and has at most $e + 1 + p - e - 1 = p$ changes of directions, whereas $D_{\mathrm{NE}}(b, f) = q > p$.

The case where d is in the same row or in the same column of c can be proved similarly. $\square$

Property 12. *Let $c = (i, j)$ and $d = (i_2, j)$, with $i_2 > i$, be two cells with labels e_a and e'_b, respectively. If $e' > e$ then $c \preceq_{NE} d$, otherwise $d \preceq_{NE} c$.*

Proof. The result follows from Property 11 in the case $e = e'$. Otherwise, let us consider $e' > e$ (the case $e > e'$ follows by symmetry). If $e' = e + 1$ we consider the cell $\hat{c}$ (with label $(e+1)$). It is easy to verify that $P_{\hat{c}} \subseteq P_d$, hence Property 11 implies $\hat{c} \preceq_{\mathrm{NE}} d$. Therefore, one has also $c \preceq_{\mathrm{NE}} d$ as $D_{\mathrm{NE}}(a, g) \leqslant e + 1$ for all $g \in P_c \setminus P_{\hat{c}}$. If $e' = e + 2$, we can apply the same idea to $\widehat{\hat{c}}$, and so on for any e', with $e' - e > 2$. $\square$

We can make use of Property 12 in order to consider only one labelled cell for each column; when a new labelled cell is found by the algorithm and there is already a labelled cell in the same column, the algorithm can discard one of the two cells. By using a support vector to keep track of labelled cells (indexed by columns), we can execute this check in constant time. Consequently, the loop iterates at most m (number of columns) times and then the time complexity of the algorithm is at most $O(m)$. An execution of the algorithm, including also this feature, is shown in Figure 5.
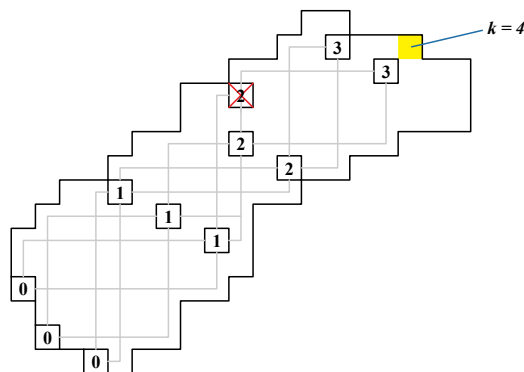

Figure 5: An example of execution of the algorithm. The presence of the highlighted cell implies that the NE-degree of convexity of the polyomino is 4.

Further, we observe that for a polyomino with degree of convexity k, every SW-corner produces at most k labelled cells, so the complexity is also bounded by $O(|C_{\mathrm{sw}}|\,k)$. Thus, we can state:

Theorem 13. *Algorithm 1 admits an implementation with runtime $O(\min(m, |C_{sw}|\,k))$, where m is the number of columns, k the degree of convexity and C_{sw} the set of SW-corners of P.*

In the worst case, it may happens that $k = \Theta(m)$ and $|C_{\mathrm{sw}}| = \Theta(m)$, but we suspect that these conditions do not hold simultaneously for any given convex polyomino. In fact, we expect that the average complexity of the algorithm is $O(|C_{\mathrm{sw}}|\,k)$, and for large m it should be $|C_{\mathrm{sw}}|\,k \ll m$.

5   Decomposition in L-convex polyominoes

Algorithm 1 allows a decomposition of a convex polyomino into NE L-convex polyominoes (having NE-degree of convexity 1) obtained by considering its labelled cells. More precisely, let $\{0, 1, 2, \ldots, e\}$ be the set of labels that the algorithm assigns to some cells c of P with $P_c \neq \emptyset$. For each h, with $0 \leqslant h \leqslant e$, let C_h be the set of cells of P that the algorithm labels by h.

We define a sequence of NE L-convex polyominoes $L_1, L_2, \ldots, L_e$ as follows.

- $L_1 = P \setminus \bigcup_{(i,j) \in C_1} P_{(i,j)}$;

- for each h, with $1 < h \leqslant e$, $L_h = (P \setminus \bigcup_{(i,j) \in C_h} P_{(i,j)}) \setminus \bigcup_{i<h} L_i$;

Each polyomino in the sequence $L_1, L_2, \ldots, L_e$ is a NE L-convex polyomino by Definition 5. Indeed, if $a, b \in L_i$ and $a \nearrow b$, then there exists some NE-path connecting a to b with at most 1 change of direction. For example, Figure 6 shows a decomposition of a 4-convex polyomino into three L-convex polyominoes L_1, L_2, L_3, and two rectangles ($P \setminus (L_1 \cup L_2 \cup L_3)$ is not connected).

By definition, $L_1, L_2, \ldots, L_e$ are uniquely determined; in general they are not maximal because can be contained in a bigger NE L-convex polyomino, but for each m, with $1 \leqslant m \leqslant e$, the polyomino $Q_m = \bigcup_{1 \leqslant h \leqslant m} L_h$ turns out to be a maximal NE m-convex polyomino. In fact, we can easily deduce the following property.

Property 14. Q_m is a maximal NE m-convex polyomino of P, i.e. if Q' is NE m-convex and $Q_m \subseteq Q' \subseteq P$ then $Q_m = Q'$.
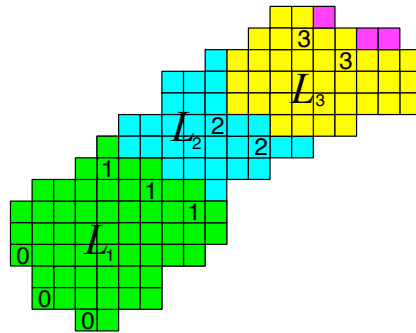
Figure 6: The decomposition of a 4-convex polyomino into 3 L-convex polyominoes and 2 rectangles.

In order to investigate about the NW-degree of convexity of the polyominoes L_i and Q_m we give the following proposition.

Proposition 15. Let P be a convex polyomino such that $D_{NE}(P) > D_{NW}(P)$. If $D_{NE}(P) > 2$ then $D_{NW}(P) = 1$.

Proof. Since P is convex, the sequence $T = [t_1, \ldots, t_m]$ of y coordinates of the top cells of the columns is unimodal, as well as the the sequence $B = [b_1, \ldots, b_m]$ of y coordinates of the bottom cells. Let h, h' be the largest indices such that $t_h \geqslant t_e$ and $b_{h'} \leqslant b_e$, for all e with $1 \leqslant e \leqslant m$. Since $D_{NE}(P) > 2$ there are $a = (i, j) \in C_{SW}$, $b = (i', j') \in C_{NE}$ with $D_{NE}(a, b) > 2$ (the red and the green cells, respectively, depicted in Figure 7) Let us consider the two columns j, j'. Notice that $t_j < b_{j'}$ (otherwise, for convexity reasons, one would have $D_{NE}(a, b) \leqslant 2$) and that it is impossible to find a column q, with $j < q < j'$ such that $t_q \geqslant t_{j'}$ and $b_q \leqslant b_j$ ($D_{NE}(a, b) = 2$ would hold in this case). Moreover, due

to convexity, one has $j \leqslant h, h' \leqslant j'$. In particular, one necessarily has $h > h'$ otherwise $t_h \geqslant t_{j'}$ and $b_h \leqslant b_j$.

Figure 7 illustrates how P looks like. The boundary of P is divided into 4 parts associated with the four different kinds of corners. More precisely, SW-corners can appear only along the red part of the boundary, NW-corners along the blue part, NE-corners along the green part and, finally, SE-corners along the yellow part. From the figure it is also immediate to see that $D_{NW}(P) = 1$. Indeed, starting from any yellow corner, all the blue corners that can be reached (by North or West steps) require at most one change of direction.

More formally, suppose $D_{NW}(P) \geqslant 2$ and let $c = (r, s) \in C_{SE}$, $d = (r', s') \in C_{NW}$ be two corners such that $D_{NW}(c, d) \geqslant 2$. This implies $t_s < t_{s'}$ and $b_s < b_{s'}$ (otherwise $D_{NW}(c, d) = 1$). Now, by unimodality of T and B it follows $s \geqslant h$ (otherwise $t_s \geqslant t_{s'}$) and $s' \leqslant h'$ (otherwise $b_{s'} \leqslant b_s$). In particular, one has $t_s \geqslant t_{j'}$ and $b_s \leqslant b_{j'}$. Furthermore, one has also $s' \geqslant j$ (otherwise $t_{s'} < t_s$) and then $t_{s'} \geqslant t_j$ and $b_{s'} \leqslant b_j$. As a consequence, since $t_{j'} \geqslant b_j \geqslant b_{s'}$ and $t_{s'} > t_s \geqslant t_{s'}$, we can easily get a NE-path which starts at a, makes $s' - i$ East steps, changes direction at (i, s'), continues with $i' - i$ North steps towards (i', s'), makes a change of direction and reach b after $j' - s'$ East steps, showing that $D_{NE}(a, b) = 2$ (contradiction). $\qquad\square$
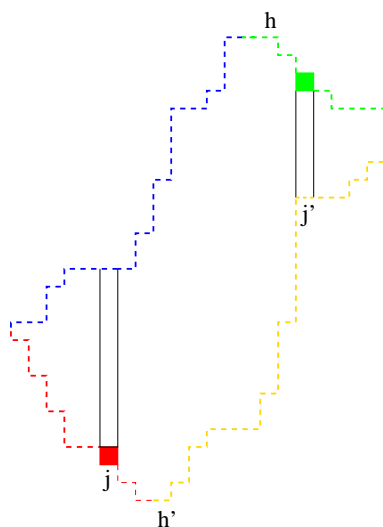
Figure 7: A polyomino P with $D_{NE}(P) > 2$.

An analogous and symmetric result holds if P is such that $D_{NE}(P) < D_{NW}(P)$. Note that the case $D_{NE}(P) = 2$ and $D_{NW}(P) = 2$ may occur, as in the polyomino shown in Figure 8. Observe that if P has degree of convexity $k > 2$ with $D_{NE}(P) > D_{NW}(P)$ then $D_{NE}(P) = k$ and $D_{NW}(P) = 1$. As a consequence, each L_i is both NE and NW L-convex and Q_m is NE m-convex and NW L-convex. If P has degree of convexity k with $D_{NE}(P) = D_{NW}(P)$ then k is at most 2 and some L_i can be not L-convex in some of the directions.

Figure 8: A polyomino P with $D_{\mathrm{NE}}(P) = D_{\mathrm{NW}}(P) = 2$.

For completeness, we point out that $P \setminus Q_e$ is not necessarily connected (it may be empty or a set of NE L-convex polyominoes).

6   Conclusions and further work

Since, in general, the L-convexity is easier to handle than the k-convexity (with $k > 1$), the decomposition here presented suggests a possible approach to solve problems concerning k-convex polyominoes. For example, it is known (cf. [5]) that an L-convex polyomino can be uniquely reconstructed by its orthogonal projections, or by partial information (cf. [7]), whereas nothing is known in general for k-convex polyominoes. By extending the result in [3], a tiling system for k-convex polyominoes can be investigated. In the same way, enumeration problems can be handed. Lastly, the labelling of a k-convex polyomino might be used to develop an efficient encoding with some applications to compression methods.

References

[1] Elena Barcucci, Srecko Brlek, and Stefano Brocchi. Pcif: An algorithm for lossless true color image compression. In *IWCIA*, volume 5852 of *Lecture Notes in Comput. Sci.*, pages 224–237. Springer, 2009.

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

[3] Stefano Brocchi, Andrea Frosini, Renzo Pinzani, and Simone Rinaldi. A tiling system for the class of l-convex polyominoes. *Theor. Comput. Sci.*, 475:73–81, 2013.

[4] Giusi Castiglione, Andrea Frosini, Emanuele Munarini, Antonio Restivo, and Simone Rinaldi. Combinatorial aspects of l-convex polyominoes. *Eur. J. Comb.*, 28(6):1724–1741, 2007.

[5] Giusi Castiglione, Andrea Frosini, Antonio Restivo, and Simone Rinaldi. A tomographical characterization of l-convex polyominoes. In *DGCI05*, volume 3429 of *Lecture Notes in Comput. Sci.*, pages 115–125. Springer, 2005.

[6] Giusi Castiglione and Antonio Restivo. Reconstruction of l-convex polyominoes. *Electronic Notes in Discrete Mathematics*, 12:290–301, 2003.

[7] Giusi Castiglione, Antonio Restivo, and Roberto Vaglica. A reconstruction algorithm for l-convex polyominoes. *Theor. Comput. Sci.*, 356(1-2):58–72, 2006.

[8] Isabelle Debled-Rennesson, Jean-Luc Remy, and Jocelyne Rouyer-Degli. Detection of the discrete convexity of polyominoes. *Discrete Applied Mathematics*, 125(1):115–133, 2003.

[9] Alberto Del Lungo, Enrica Duchi, Andrea Frosini, and Simone Rinaldi. On the generation and enumeration of some classes of convex polyominoes. *Electr. J. Comb.*, 11(1), 2004.

[10] Marie-Pierre Delest and Grard Viennot. Algebraic languages and polyominoes enumeration. *Theoretical Computer Science*, 34(1-2):169–206, 1984.

[11] Enrica Duchi, Simone Rinaldi, and Gilles Schaeffer. The number of z-convex polyominoes. *Advances in Applied Mathematics*, 40(1):54 – 72, 2008.

[12] W. Solomon Golomb. Checker boards and polyominoes. *The American Mathematical Monthly*, 61:675–682, 1954.

[13] C. E. Kim and Azriel Rosenfeld. On the convexity of digital regions. *Pattern Recognition*, 5:1010–1015, 1980.

[14] Paolo Massazza. On the generation of l-convex polyominoes. In *Proc. of GasCom12, Bordeaux June 25-27*, 2012.

[15] Anne Micheli and Dominique Rossin. Counting k-convex polyominoes. *Electr. J. Comb.*, 20(2), 2013.