LATTICE SIZE OF POLYGONS WITH RESPECT TO THE STANDARD SIMPLEX

ANTHONY HARRISON, JENYA SOPRUNOVA, AND PATRICK TIERNEY

Abstract. In this paper we study the lattice size $l_{\Sigma}(P)$ of a lattice polygon $P$ with respect to the standard simplex $\Sigma \subset \mathbb{R}^2$. The lattice size $l_{\Sigma}(P)$ is the smallest integer $l$ such that $P$ is contained in an $l$-dilate of $\Sigma$ after some unimodular transformation.

This invariant was studied by Schicho in the context of simplifying parametrizations of rational surfaces. Schicho gave an “onion skins” algorithm for mapping a lattice polygon $P$ into $l\Sigma$ for a small integer $l$. Castryck and Cools proved that Schicho’s algorithm computes $l_{\Sigma}(P)$ and provided a similar algorithm for finding the lattice size of a plane polygon with respect to the unit square.

In this paper we give a new algorithm for computing $l_{\Sigma}(P)$ of a lattice polygon $P$, which does not require enumeration of lattice points in $P$. We also discuss a possible 3D generalization of our algorithm.

Introduction

The lattice size $l_X(P)$ of a non-empty lattice polytope $P \subset \mathbb{R}^n$ with respect to a set $X$ with positive Jordan measure was defined in [4] as the smallest integer $l$ such that $T(P)$ is contained in the $l$-dilate of $X$ for some affine unimodular transformation $T$. Note that when $X = [0, 1] \times \mathbb{R}^{n-1}$, the lattice size of a lattice polytope $P$ with respect to $X$ is the lattice width $w(P)$ (see Definition 1.4), a very important and much studied invariant.

In the case when $X$ is the standard simplex, this invariant comes up in the context of simplifying parameterizations of surfaces, as explained in [1], [3], [7].

The lattice size is also useful when dealing with questions that arise when studying lattice polytopes. Let $\square = [0, 1]^2$ be the unit square. In [3] the lattice size $l_{\square}(P)$ of a lattice polygon $P$ with respect to the unit square is used to classify small lattice polygons and corresponding toric codes. This notion also appears implicitly in the work of Arnold [1], Bárány and Pach [2], and Lagarias and Ziegler [6].

In [7] Schicho provided an “onion skins” algorithm for mapping a lattice polygon $P$ into $l\Sigma$ for a small integer $l$. In [4] Castryck and Cools proved that this algorithm computes $l_{\Sigma}(P)$. The idea of this algorithm is that when one passes from a lattice polygon $P$ to the convex hull of its interior lattice points, its lattice size $l_{\Sigma}(P)$ drops.

2010 Mathematics Subject Classification. 52B20.

Work of Soprunova and Tierney was partially supported by NSF Grant DMS-1156798.
by 3 unless \( P \) belongs to a list of exceptional cases. One can then compute \( \text{ls}_\Sigma(P) \) by successively peeling off “onion skins” of \( P \).

The downside of the “onion skins” algorithm is that it is quite time-consuming since one needs to enumerate all interior lattice points of \( P \). In this paper we provide a new algorithm for computing \( \text{ls}_\Sigma(P) \) for a plane lattice polygon \( P \), which does not require enumeration of lattice points in \( P \).

Our work is based on our earlier algorithm for computing \( \text{ls}_\Box(P) \), the lattice size with respect to the unit square \( \Box \). In [5] we developed a fast algorithm for computing \( \text{ls}_\Box(P) \), as well as provided a generalization of this algorithm to the lattice polytopes \( P \subset \mathbb{R}^3 \). In the 2D case the algorithm is particularly simple. Define \( \text{nls}_\Box(P) \) to be the smallest integer \( k \) such that \( P \) is contained in \( k \Box \) after a lattice translation. Then the algorithm is based on the following statement proved in [5]: If the lattice width of \( P \) in both of the diagonal directions \((1,1)\) and \((1,-1)\) is at least its lattice width in each of the standard basis directions \((1,0)\) and \((0,1)\), then \( \text{ls}_\Box(P) = \text{nls}_\Box(P) \). (See Definition 1.4 for the definition of the lattice width \( w_v(P) \) of a lattice polytope \( P \) in a given direction \( v \).)

The algorithm for computing \( \text{ls}_\Box(P) \) is then very straightforward: Start with \( P \), find \( \text{nls}_\Box(P) \), check if the lattice width of \( P \) in the directions \((1,1)\) and \((1,-1)\) is at least \( \text{nls}_\Box(P) \). If this is the case, conclude that \( \text{ls}_\Box(P) = \text{l} \). If not, pass to the polygon \( T_A(P) \), where \( T_A \) is the linear map defined by \( A = \begin{bmatrix} 1 & 0 \\ 1 & \pm 1 \end{bmatrix} \) or \( \begin{bmatrix} 0 & 1 \\ 1 & \pm 1 \end{bmatrix} \), and repeat the step.

In this paper we prove that this algorithm can also be used for computing \( \text{ls}_\Sigma(P) \). We define \( \text{nls}_\Sigma(P) \) to be the smallest \( l \) such that \( P \) is contained in \( l \Sigma \) after a transformation which is a composition of a unimodular matrix of the form \( \begin{bmatrix} \pm 1 & 0 \\ 0 & \pm 1 \end{bmatrix} \) and a lattice translation. Our main result is the following theorem.

**Theorem 0.1.** Let \( P \subset \mathbb{R}^2 \) be a lattice polygon. Suppose that \( \text{ls}_\Box(P) = \text{nls}_\Box(P) \) and \( w(P) = w_{(1,0)}(P) \). Then \( \text{ls}_\Sigma(P) = \text{nls}_\Sigma(P) \).

In Section 2 we address a natural question of whether a similar statement holds true for \( P \subset \mathbb{R}^3 \), which would then provide an algorithm for computing \( \text{ls}_\Sigma(P) \) for 3D lattice polytopes. We give an example that demonstrates that the answer to this question is negative.

In Section 3 we modify our 2D algorithm. In this modified algorithm, at each step we check whether the naive lattice size \( \text{nls}_\Sigma(P) \) drops after we apply to \( P \) linear maps defined by matrices \( U = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} \) and \( L = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix} \). If in both cases the lattice size does not drop, we prove that \( \text{ls}_\Sigma(P) = \text{nls}_\Sigma(P) \). While this algorithm takes longer than checking the lattice widths in the diagonal directions, there is hope that this algorithm might be generalizable to lattice polytopes \( P \subset \mathbb{R}^3 \).

It is explained in [3, 5] how one can generalize the standard width algorithm to computing lattice size of a lattice polytope \( P \subset \mathbb{R}^n \) with respect to the unit cube.
[0, 1]^n$. This algorithm is very time-consuming, but it works in any dimension n. In Section 4 we explain how to adjust this algorithm for computing $l_{\Sigma}(P)$, where $\Sigma$ is the standard n-dimensional simplex.

1. Lattice Size of Polygons

Let $\Sigma = \text{conv}\{(0, 0), (1, 0), (0, 1)\} \subset \mathbb{R}^2$ be the standard simplex in the plane. Recall that a square integer matrix $A$ is unimodular if $\det A = \pm 1$. For such a $2 \times 2$ matrix $A$ and an integer vector $v$, the transformation $T: \mathbb{R}^2 \to \mathbb{R}^2$, defined by $T(x) = Ax + v$, is called an affine unimodular transformation of the plane. Such transformations preserve the integer lattice $\mathbb{Z}^2 \subset \mathbb{R}^2$. Let $P \subset \mathbb{R}^2$ be a lattice polygon. We will simply write $AP$ for the image $T(P)$ of $P$ under the linear transformation $T: \mathbb{R}^2 \to \mathbb{R}^2$ defined by $T(x) = Ax$.

**Definition 1.1.** We define the *lattice size* $l_{\Sigma}(P)$ of a plane lattice polygon $P$ with respect to the standard simplex $\Sigma$ to be the smallest $l$ such that $P$ is contained in the $l$-dilate $l\Sigma$ of the standard simplex $\Sigma$ after an affine unimodular transformation. A unimodular transformation which minimizes $l$ is said to compute $l_{\Sigma}(P)$.

We also define *naive lattice size* $n_{\Sigma}(P)$ to be the smallest $l$ such that $P$ is contained in $l\Sigma$ after a transformation which is a composition of $A = \begin{pmatrix} \pm 1 & 0 \\ 0 & \pm 1 \end{pmatrix}$ and a lattice translation. That is, $n_{\Sigma}(P)$ is the smallest of the four numbers below, each of which corresponds to fitting a dilate of $\begin{pmatrix} \pm 1 & 0 \\ 0 & \pm 1 \end{pmatrix} \Sigma$ around $P$:

$$l_1(P) := \max_{(x,y) \in P} (x + y) - \min_{(x,y) \in P} x - \min_{(x,y) \in P} y,$$

$$l_2(P) := \max_{(x,y) \in P} x + \max_{(x,y) \in P} y - \min_{(x,y) \in P} (x + y),$$

$$l_3(P) := \max_{(x,y) \in P} y - \min_{(x,y) \in P} x + \max_{(x,y) \in P} (x - y),$$

$$l_4(P) := \max_{(x,y) \in P} x - \min_{(x,y) \in P} y + \max_{(x,y) \in P} (y - x).$$

**Example 1.** Let $P = \text{conv}\{(0, 0), (4, 1), (5, 2)\}$. Then, as demonstrated in the diagram below, $l_1(P) = 7$, $l_2(P) = 7$, $l_3(P) = 5$, and $l_4(P) = 5$, and hence $n_{\Sigma}(P) = 5$. If we apply $A = \begin{pmatrix} 1 & -2 \\ 0 & 1 \end{pmatrix}$ to this $P$ we get a triangle with the vertices $(0, 0), (2, 1), (1, 2)$, so $l_1(AP) = 3$. Since $P$ has a lattice point inside, while $2\Sigma$ does not, it is impossible to unimodularly map $P$ inside $2\Sigma$. We can conclude that $l_{\Sigma}(P) = 3$ and that $A = \begin{pmatrix} 1 & -2 \\ 0 & 1 \end{pmatrix}$ computes the lattice size of $P$.

We next record a few straight-forward observations.

**Lemma 1.2.** Let $P \subset \mathbb{R}^2$ be a lattice polygon.
(a) If $P$ is reflected in the line $x = y$ then $l_1$ and $l_2$ are fixed while $l_3$ and $l_4$ switch.

(b) If $P$ is reflected in the $x$-axis then
   (i) $l_1$ and $l_3$ switch;
   (ii) $l_2$ and $l_4$ switch.

(c) If $P$ is reflected in the $y$-axis then
   (i) $l_1$ and $l_4$ switch;
   (ii) $l_2$ and $l_3$ switch.

(d) The naive lattice size $\mathrm{nls}_\Sigma(P)$ is preserved under the reflection in the line $y = x$ as well as under the reflections in the $x$- and $y$-axis.

Lemma 1.3. Let $P \subset \mathbb{R}^2$ be a lattice polygon.

(a) 

$$l_1 \left( \begin{bmatrix} a & b \\ c & d \end{bmatrix} P \right) = \max_{(x,y) \in P} ((a + c)x + (b + d)y) - \min_{(x,y) \in P} (ax + by) - \min_{(x,y) \in P} (cx + dy);$$

(b) 

$$l_1 \left( \begin{bmatrix} a & b \\ c & d \end{bmatrix} P \right) = l_1 \left( \begin{bmatrix} c & d \\ a & b \end{bmatrix} P \right);$$

(c) 

$$l_1 \left( \begin{bmatrix} a & b \\ c & d \end{bmatrix} P \right) = l_1 \left( \begin{bmatrix} -(a + c) & -(b + d) \\ c & d \end{bmatrix} P \right) = l_1 \left( \begin{bmatrix} a & b \\ -(a + c) & -(b + d) \end{bmatrix} P \right).$$

Proof. First equality of part (c) is equivalent to claiming that $l_1(\dot{A}P) = l_1(\dot{S}A^T)$, where $A = \begin{bmatrix} a & b \\ c & d \end{bmatrix}$ and $S = \begin{bmatrix} -1 & -1 \\ 0 & 1 \end{bmatrix}$, so replacing $\dot{A}P$ with $P$, it is enough to
show that \( l_1(SP) = l_1(P) \). We have
\[
\begin{align*}
l_1(SP) &= \max_{(x,y) \in P} (-x) - \min_{(x,y) \in P} (-x - y) - \min_{(x,y) \in P} y \\
&= \max_{(x,y) \in P} (x + y) - \min_{(x,y) \in P} x - \min_{(x,y) \in P} y = l_1(P).
\end{align*}
\]

\[\square\]

**Definition 1.4.** Let \( P \subset \mathbb{R}^2 \) be a lattice polygon. Recall that an integer vector \( v = (a, b) \in \mathbb{Z}^2 \) is primitive if \( \gcd(a, b) = 1 \). The width of \( P \) in the direction of a primitive \( v \in \mathbb{Z}^2 \) is defined by
\[
w_v(P) = \max_{x \in P} v \cdot x - \min_{x \in P} v \cdot x,
\]
where \( v \cdot x \) denotes the standard dot-product. The **lattice width** \( w(P) \) of \( P \) is the minimum of \( w_v(P) \) over all primitive vectors \( v \in \mathbb{Z}^2 \).

**Lemma 1.5.** Suppose that for \( A = \begin{bmatrix} a & b \\ c & d \end{bmatrix} \) we have \( l_1 := l_1(AP) < l \). Then the width of \( P \) in the three directions, \((a, b), (c, d), \) and \((a + c, b + d)\) is less than \( l \).

**Proof.** For any primitive lattice vector \( v \in \mathbb{Z}^2 \) we have
\[
w_v(AP) = \max_{x \in P} v \cdot (Ax) - \min_{x \in P} v \cdot (Ax) = \max_{x \in P} (A^T v) \cdot x - \min_{x \in P} (A^T v) \cdot x = w_{A^T v}(P).
\]

Let \( e_1 \) and \( e_2 \) be the standard basis vectors in \( \mathbb{R}^2 \). Then
\[
w_{(a,b)}P = w_{e_1}(AP) \leq l_1 < l,
\]
where we used \( AP \subset l_1 \Sigma \), which implies that \( w_{e_1}(AP) \leq w_{e_1}(l_1 \Sigma) = l_1 \). The same argument works for \( e_2 \) and \( e_1 + e_2 \). \[\square\]

Next, let \( \square = \text{conv}\{(0,0), (1,0), (0,1), (1,1)\} \subset \mathbb{R}^2 \) be the unit square.

**Definition 1.6.** The **lattice size** \( ls_{\square}(P) \) of a plane lattice polygon \( P \) with respect to \( \square \) is the smallest \( k \) such that \( P \) is contained in \( k\square \) after an affine unimodular transformation. A unimodular transformation which minimizes \( k \) is said to compute \( ls_{\square}(P) \).

The **naive lattice size** \( nls_{\square}(P) \) with respect to the unit square is the smallest \( k \) such that \( P \) is contained in \( k\square \) after a lattice shift.

A fast algorithm for computing \( ls_{\square}(P) \) in dimension 2 and 3 is explained in [5]. The 2D algorithm is based on the following theorem.

**Theorem 1.7.** [5] Let \( P \subset \mathbb{R}^2 \) be a lattice polygon. Suppose that \( nls_{\square}(P) = k \) and \( w_{(1,\pm1)}(P) \geq k \). Then \( w_{(a,b)}(P) \geq k \) for all primitive directions \( (a,b) \in \mathbb{Z}^2 \), except, possibly, for \((a,b) = (\pm1,0)\) or \((0,\pm1)\). This implies that \( ls_{\square}(P) = k \) and \( w(P) = \min\{w_{(1,0)}(P), w_{(0,1)}(P)\} \).
The algorithm is now simple: If \( w_{(1, \pm 1)}(P) \geq \nls_{\Sigma}(P) \) then \( \ls_{\Sigma}(P) = \nls_{\Sigma}(P) \).
Otherwise, replace \( P \) with \( AP \), where \( A = \begin{bmatrix} 1 & 0 \\ 1 & \pm 1 \end{bmatrix} \) or \( \begin{bmatrix} 1 & \pm 1 \\ 0 & 1 \end{bmatrix} \) and repeat the step. At the end we also reflect \( P \) in the line \( y = x \), if needed, to ensure that \( w(P) = w_{(1,0)}(P) \). The product \( B \) of the matrices \( A \) used at each of the steps, including the possible reflection at the end, computes \( \ls_{\Sigma}(P) \) and \( w(P) \), that is, \( \ls_{\Sigma}(P) = \nls_{\Sigma}(BP) \) and \( w(P) = w_{(1,0)}(BP) \).

We now prove our main result: If a unimodular matrix \( B \) computes \( \ls_{\Sigma}(P) \) and \( w(P) \), then it also computes \( \nls_{\Sigma}(P) \). Therefore, the algorithm of [5] that we explained above can be used to compute \( \ls_{\Sigma}(P) \).

**Theorem 1.8.** Let \( P \subset \mathbb{R}^2 \) be a lattice polygon. Suppose that \( \ls_{\Sigma}(P) = \nls_{\Sigma}(BP) \) and \( w(P) = w_{(1,0)}(BP) \) for some unimodular matrix \( B \). Then \( \ls_{\Sigma}(P) = \nls_{\Sigma}(BP) \).

**Proof.** Replacing \( BP \) with \( P \) we can reformulate this statement: If \( \ls_{\Sigma}(P) = \nls_{\Sigma}(P) \) and \( w(P) = w_{(1,0)}(P) \), then \( \ls_{\Sigma}(P) = \nls_{\Sigma}(P) \). We reflect \( P \) in the \( x \)- and \( y \)-axis, if necessary, to ensure that \( l_1(P) = \nls_{\Sigma}(P) \) and denote \( l_1(P) = \nls_{\Sigma}(P) = l \). We then translate \( P \) by an integer vector so that \( \min_{(x,y) \in P} x = 0 \) and \( \min_{(x,y) \in P} y = 0 \) and get \( l = l_1(P) = \max_{(x,y) \in P} (x+y) \).

Our goal now is to show that there is no unimodular matrix \( A \) that satisfies \( l_1(AP) < l \). If such a matrix existed, the width of \( P \) in the direction of each of the rows of \( A \) and in the direction of the sum of rows of \( A \) would be less than \( l \), as observed in Lemma [15].

Our plan is to show that for many primitive directions the corresponding width of \( P \) is at least \( l \), so these vectors cannot appear as rows or the sum of rows in \( A \). We will then work with the remaining primitive directions to see how they can be used as rows and the sum of rows to form a unimodular matrix \( A \), and then for each of these matrices we will show that \( l_1(AP) \geq l \).

We denote \( m := \max_{(x,y) \in P} x \) and \( k := \max_{(x,y) \in P} y \). Since \( w(P) = w_{(1,0)}(P) \) and \( \ls_{\Sigma}(P) = \nls_{\Sigma}(P) \), we have \( m \leq k \) and \( \ls_{\Sigma}(P) = k \). We also have \( w_{(1, \pm 1)}(P) \geq k \) since otherwise, in the case \( m < k \), we can apply one of \( \begin{bmatrix} 1 & 0 \\ 1 & \pm 1 \end{bmatrix} \) or \( \begin{bmatrix} 1 & \pm 1 \\ 0 & 1 \end{bmatrix} \) and decrease \( \nls_{\Sigma}(P) \).

In the case when \( m = k \) we again have \( w_{(1, \pm 1)}(P) \geq k \), since otherwise this would give a direction with the corresponding width of \( P \) less than \( m \).

We next denote \( s := \min_{(x,y) \in P} (x+y) \). Then we have

\[
(1.1) \quad w_{(1,1)}(P) = l - s \geq k \\
(1.2) \quad l_2(P) = m + k - s \geq l.
\]

Let \((a, b) \in \mathbb{Z}^2\) be primitive. Since \( w_{(a,b)}(P) = w_{(-a,-b)}(P) \) we can assume that \( a \geq 0 \). Suppose first that \( a \geq b \geq 0 \).
Since \( P \subset \Sigma \) has a point on the segment that joins points \((l - k, k)\) and \((m, l - m)\) we have

\[
\max_{(x,y) \in P} (ax + by) \geq \min \{a(l - k) + bk, am + b(l - m)\} = a(l - k) + bk,
\]

since \( a(l - k) + bk \leq am + b(l - m) \) reduces to \( b(k + m - l) \leq a(k + m - l) \), which holds true since \( a \geq b > 0 \) and by (1.2) we have \( k + m - l \geq s \geq 0 \). Since \( P \) has a point on the segment that joins points \((0, s)\) and \((s, 0)\) using (1.1) we get

\[
\min_{(x,y) \in P} (ax + by) \leq \max \{as, bs\} = as \leq a(l - k).
\]

Hence for \( b \geq 2 \) using (1.2) we get

\[
w_{(a,b)} P \geq a(l - k) + bk - a(l - k) = bk \geq 2k \geq m + k \geq l + s \geq l,
\]

so we can rule out all primitive directions with \( a \geq b > 0 \) except for \((1, 0)\) and the ones with \( a \geq b = 1 \).

We next assume that \( b > a \geq 0 \). Since \( P \) has a point on the segment connecting \((0, k)\) and \((l - k, k)\) we get

\[
\max_{(x,y) \in P} (ax + by) \geq \min \{bk, a(l - k) + bk\} = bk.
\]

Adding up (1.1) and (1.2) we get \( m \geq 2s \), so

\[
\min_{(x,y) \in P} (ax + by) \leq \max \{as, bs\} = bs \leq b \cdot \frac{m}{2} \leq \frac{b}{2} \cdot k.
\]

Hence for \( b \geq 4 \) we get

\[
w_{(a,b)} P \geq \frac{b}{2} \cdot k \geq 2k \geq m + k \geq l + s \geq l.
\]

Now let \( a = 2 \) and \( b = 3 \). Then

\[
\max_{(x,y) \in P} (2x + 3y) \geq \min \{2m + 3(l - m), 2(l - k) + 3k\} = 3l - m,
\]

since \( 2m + 3(l - m) \leq 2(l - k) + 3k \) is equivalent to \( l \leq k + m \), which holds true under our assumptions. We also have \( \min_{(x,y) \in P} (2x + 3y) \leq 3s \), so

\[
w_{(2,3)} P \geq 3l - m - 3s \geq l,
\]
since the last inequality is equivalent to \( m + 3s \leq 2l \), and we have \( m + s \leq k + s \leq l \) and \( 2s \leq m \leq l \), which add up to \( m + 3s \leq 2l \).

Next let \( a = 1 \) and \( b = 3 \). We have

\[
\max_{(x, y) \in P} (x + 3y) \geq \min\{m + 3(l - m), l - k + 3k\} = 3l - 2m
\]

since \( 3l - 2m \leq l + 2k \) is equivalent to \( l \leq m + k \). Also, plugging a point with \( y = 0 \) we get \( \min_{(x, y) \in P} (x + 3y) \leq m \). Hence \( w_{(1, 3)} P \geq 3l - 3m \), so we can rule out this direction if \( 3l - 3m \geq l \). Otherwise, \( m > \frac{2l}{3} \) and using a point in \( P \) with \( y = k \) we get

\[
\max_{(x, y) \in P} (x + 3y) \geq 3k \geq 3m.
\]

Then \( w_{(1, 3)} P \geq 3m - m = 2m > \frac{4l}{3} > l \).

We have proved the following lemma.

**Lemma 1.9.** Let \( w(P) = w_{(1, 0)}(P) \) and \( l_{\square}(P) = \text{nls}_{\square}(P) \). If \( l_2(P) \geq l_1(P) = l \) then \( w_{(a, b)} \geq l \) for all primitive directions \( (a, b) \) with \( a, b \geq 0 \), except, possibly, for \( (a, b) \) of the form \((a, 1), (1, 0), (1, 2), \) and \((0, 1)\).

We next deal with the case \( a \geq 0 \geq b \). If \( l_4(P) \geq l_3(P) \) we reflect \( P \) in the \(-x\)-axis and then shift the reflection by a lattice vector so that for the obtained polygon \( P' \) we have \( \min_{(x, y) \in P'} y = 0 \). We then have \( w_{(a, b)} P = w_{(a, -b)} P' \) and

\[
\max_{(x, y) \in P'} x = m, \quad \max_{(x, y) \in P'} y = k, \quad w_{(1, 1)}(P') = w_{(1, -1)}(P) \geq k.
\]

Using Lemma 1.2 we have \( l_1(P') = l_3(P) \leq l_4(P) = l_2(P') \).

Similarly, if \( l_4(P) \leq l_3(P) \) we reflect \( P \) in the \(-y\)-axis and then shift the reflection by a lattice vector and get \( l_1(P) = l_4(P) \leq l_3(P) = l_2(P') \). Hence by Lemma 1.9 \( w_{(a, b)}(P) = w_{(a, -b)}(P') \geq l \) unless \( (a, -b) \) is of the form \((a, 1), (1, 0), (1, 2), (0, 1)\).

Hence we can assume that the rows and the row sum of \( A \) are from the list

\[
\{(a, \pm 1), (\pm 1, 0), (\pm 1, \pm 2)\}, \quad \text{where} \quad a \in \mathbb{Z}.
\]

Our next goal is to show that unimodular matrices of this form are listed in (1.5) below, up to reductions allowed by Lemma 1.3. Since \((\pm 1, 0)\) and \((\pm 1, \pm 2)\) cannot be the two rows of a unimodular matrix, we can assume that first row of \( A \) is of the form \((a, \pm 1)\). If now the second row is of the form \((\pm 1, \pm 2)\), that is, \( A = \begin{bmatrix} a & 1 \\ \pm 1 & -2 \end{bmatrix} \) or \( A = \begin{bmatrix} a & -1 \\ \pm 1 & 2 \end{bmatrix} \), using Lemma 1.3 we can change second row to one of the form \((b, \pm 1)\) for some \( b \in \mathbb{Z} \).

If the second row of \( A \) is of the form \((\pm 1, 0)\) we get \( A = \begin{bmatrix} a & \pm 1 \\ \pm 1 & 0 \end{bmatrix} \) for some \( a \in \mathbb{Z} \).
We now identify unimodular matrices both of whose rows are of the form \((a, \pm 1)\). Let first \(A = \begin{bmatrix} a & 1 \\ b & -1 \end{bmatrix}\) for some \(a, b \in \mathbb{Z}\). Then the sum of rows \((a+b, 0)\) is supposed to be on the list (1.4), so we get \(a + b = \pm 1\). Hence \(A = \begin{bmatrix} a & 1 \\ \pm 1 - a & -1 \end{bmatrix}\) for some \(a \in \mathbb{Z}\) and using Lemma 1.3 we can replace \(A\) with \(\begin{bmatrix} a & 1 \\ \pm 1 & 0 \end{bmatrix}\).

Next let \(A = \begin{bmatrix} a & 1 \\ b & 1 \end{bmatrix}\). Then \(a + b = \pm 1\) and \(a - b = \pm 1\), where we have the second condition since \(A\) is unimodular. We get \(A = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}\) or \(\begin{bmatrix} -1 & 1 \\ 0 & -1 \end{bmatrix}\). If \(A\) is of the form \(\begin{bmatrix} a & -1 \\ b & -1 \end{bmatrix}\) we get \(A = \begin{bmatrix} 1 & -1 \\ 0 & -1 \end{bmatrix}\) or \(\begin{bmatrix} -1 & -1 \\ 0 & 1 \end{bmatrix}\).

Here is the list matrices \(A\) for which we need to show that \(l_1(AP) \geq l\):

\[
(1.5) \quad \left\{ \begin{bmatrix} \pm a & \pm 1 \\ \pm 1 & 0 \end{bmatrix}, \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}, \begin{bmatrix} -1 & 1 \\ 0 & 1 \end{bmatrix}, \begin{bmatrix} 1 & -1 \\ 0 & -1 \end{bmatrix}, \begin{bmatrix} -1 & -1 \\ 0 & -1 \end{bmatrix} \right\},
\]

where \(a\) is a non-negative integer.

**Lemma 1.10.** Let \(l_1(P) = l \leq l_2(P)\), \(\text{lsc}(P) = \text{uls}(P)\), and \(w(P) = w(1,0)(P)\). Then \(l_1(AP) \geq l\) for \(A = \begin{bmatrix} a & 1 \\ 1 & 0 \end{bmatrix}, \begin{bmatrix} a & 1 \\ -1 & 0 \end{bmatrix}, \begin{bmatrix} -a & -1 \\ 1 & 0 \end{bmatrix},\) and \(\begin{bmatrix} -a & -1 \\ -1 & 0 \end{bmatrix}\), where \(a \geq 0\) in the first and fourth of these matrices, and \(a \geq 1\) in the second and third.

**Proof.** Let first \(A = \begin{bmatrix} a & 1 \\ 1 & 0 \end{bmatrix}\) with \(a \geq 0\). Then

\[
l_1(AP) = \max_{(x,y) \in P} ((a+1)x + y) - \min_{(x,y) \in P} (ax + y) - \min_{(x,y) \in P} (x).
\]

If \(a \geq 1\), using (1.3) we get \(\max_{(x,y) \in P} ((a+1)x + by) \geq (a+1)(l-k) + k\). We also have \(\min_{(x,y) \in P} (ax + y) \leq as\), so we get \(l_1(AP) \geq (a+1)(l-k) + k - as\). Then \((a+1)(l-k) + k - as \geq l\) is equivalent to \(a(l-k) \geq as\), which holds true since \(a \geq 1\) and \(l \geq k + s\). If \(a = 0\) then \(l_1(AP) = l_1(P) = l\).

We next work with \(A = \begin{bmatrix} a & 1 \\ -1 & 0 \end{bmatrix}\) with \(a \geq 1\). If \(a \geq 2\) we have

\[
l_1(AP) = \max_{(x,y) \in P} ((a-1)x + y) - \min_{(x,y) \in P} (ax + y) - \min_{(x,y) \in P} (-x) \geq (a-1)(l-k) + k - as + m,
\]

and \((a-1)(l-k) + k - as + m \geq l\) if and only if \((a-2)(l-k) + m \geq as\), and since \(l-k \geq s\) and \(m \geq 2s\) this holds true. If \(a = 1\) then \(l_1(AP) = l_2(P) \geq l\).

Using Lemma 1.3 we get

\[
l_1 \left( \begin{bmatrix} -a & -1 \\ 1 & 0 \end{bmatrix} P \right) = l_1 \left( \begin{bmatrix} a-1 & 1 \\ 1 & 0 \end{bmatrix} P \right) \geq l.
\]
and
\[ l_1 \left( \begin{bmatrix} -a & -1 \\ -1 & 0 \end{bmatrix} P \right) = l_1 \left( \begin{bmatrix} a + 1 & 1 \\ -1 & 0 \end{bmatrix} P \right) \geq l. \]

\[ \square \]

**Lemma 1.11.** Suppose that whenever a lattice polygon \( P \) satisfies \( l_1(P) = l \leq l_2(P) \), \( l_3(P) = l_5(P) = l_7(P) \), and \( w(P) = w_{(1, 0)}(P) \), we have \( l_1(AP) \geq l \) for some fixed unimodular matrix \( A = \begin{bmatrix} a & b \\ c & d \end{bmatrix} \). Then if we additionally assume that \( nls_\Sigma(P) \geq l \), that is, \( l_3(P) \geq l \) and \( l_4(P) \geq l \), then we can also conclude that \( l_1(BP) \geq l \) for \( B = \begin{bmatrix} -a & b \\ -c & d \end{bmatrix} \) or \( B = \begin{bmatrix} a & -b \\ c & -d \end{bmatrix} \).

**Proof.** If \( l_4(P) \geq l_3(P) \), let \( P' \) be the reflection of \( P \) in the \( x \)-axes. Then
\[ l_1(P') = l_3(P) \leq l_4(P) = l_2(P'). \]

We also have \( l_5(P') = l_7(P') = l_9(P') \), and \( w(P') = w(P) = w_{(1, 0)}(P) = w_{(1, 0)}(P') \). Hence we can apply the assumption of the lemma to \( P' \) and get
\[ l_1 \left( \begin{bmatrix} a & -b \\ c & -d \end{bmatrix} P \right) = l_1 \left( \begin{bmatrix} a & b \\ c & d \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} P \right) = l_1 \left( \begin{bmatrix} a & b \\ c & d \end{bmatrix} P' \right) \geq l. \]

Similarly, if \( l_3(P) \geq l_4(P) \) we conclude that \( l_1 \left( \begin{bmatrix} -a & b \\ -c & d \end{bmatrix} P \right) \geq l. \)

\[ \square \]

Using this lemma, we can conclude that \( l_1(AP) \geq l \) for
\[ A = \begin{bmatrix} a & -1 \\ 1 & 0 \end{bmatrix}, \begin{bmatrix} a & -1 \\ -1 & 0 \end{bmatrix}, \begin{bmatrix} -a & 1 \\ 1 & 0 \end{bmatrix}, \begin{bmatrix} -a & 1 \\ -1 & 0 \end{bmatrix}, \]
where \( a \geq 1 \). The reason is that whenever we flip the sign in a column of any of these matrices we get a matrix which is already covered by Lemma 1.10. Note that such matrices with \( a = 0 \) are also covered since \( l_i(P) \geq l \) for \( i = 1 \ldots 4 \).

It remains to show that \( l_1(AP) \geq l \) for \( A \) on the list
\[ \left\{ \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}, \begin{bmatrix} -1 & 1 \\ 0 & 1 \end{bmatrix}, \begin{bmatrix} 1 & -1 \\ 0 & -1 \end{bmatrix}, \begin{bmatrix} -1 & -1 \\ 0 & -1 \end{bmatrix} \right\}, \]
which by our sign-flipping argument reduces to working with \( \left\{ \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}, \begin{bmatrix} -1 & -1 \\ 0 & -1 \end{bmatrix} \right\} \).

If \( A = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} \) then
\[ l_1(AP) = \max_{(x, y) \in P} (x + 2y) - \min_{(x, y) \in P} (x + y) - \min_{(x, y) \in P} y \]
\[ = \max_{(x, y) \in P} (x + 2y) - s \geq 2k - s \geq k + m - s \geq l. \]
If $A = \begin{bmatrix} -1 & -1 \\ 0 & -1 \end{bmatrix}$ we get

$$l_1(AP) = \max_{(x,y) \in P} (-x - 2y) - \min_{(x,y) \in P} (-x - y) - \min_{(x,y) \in P} -y$$

$$= - \min_{(x,y) \in P} (x + 2y) + \max_{(x,y) \in P} (x + y) + \max_{(x,y) \in P} y \geq -m + l + k \geq l,$$

where we used $\min_{(x,y) \in P} (x + 2y) \leq m$, which holds true since $P$ has a point of the form $(x,0)$ with $x \leq m$.

We have checked that $l_1(AP) \geq l$ for all unimodular matrices $A$ and hence $l_{\Sigma}(P) = l$. \hfill $\square$

### 2. Counterexample in the 3-space.

Let now $\Sigma$ and $\square$ denote the standard simplex and the unit cube in the 3-space and let $P \subset \mathbb{R}^3$ be a lattice polytope. As in the plane case, we define $l_{\Sigma}(P)$ to be the smallest integer $l$ such that $AP \subset l\Sigma$ after an affine unimodular transformation $A$. Similarly, $l_{\square}(P)$ is the smallest integer $k$ such that $AP \subset k\square$ after an affine unimodular transformation $A$. As before, we say that such $A$ computes the corresponding lattice size.

We then define $n_{l_{\Sigma}}(P)$ to be the smallest $l$ such that $P$ is contained in $l\Sigma$ after a transformation which is a composition of $A = \begin{bmatrix} \pm 1 & 0 & 0 \\ 0 & \pm 1 & 0 \\ 0 & 0 & \pm 1 \end{bmatrix}$ and a lattice translation. We also define $n_{l_{\square}}(P)$ to be the the smallest $k$ such that $P$ is contained in $k\square$ after a lattice shift.

A fast algorithm for computing $l_{\square}(P)$ for 3D lattice polytopes is explained in [5]. It is natural to ask whether our 2D result of the previous section extends to the 3D space and therefore the 3D algorithm for computing $l_{\square}(P)$ could be used to also find $l_{\Sigma}(P)$. That is, we are asking if it is true that a matrix $A$ that computes $l_{\square}(P)$ also computes $l_{\Sigma}(P)$, where $P \subset \mathbb{R}^3$. We can also relax this question: Is it true that there exists a matrix $A$ that computes both $l_{\square}(P)$ and $l_{\Sigma}(P)$? The following example demonstrates that the answer to both of these questions is negative.

**Example 2.** Let $P$ be the convex hull of the set

$$\{(1,1,2), (4,4,4), (0,2,2), (3,0,3), (4,3,0)\} \subset \mathbb{R}^3.$$

Then $n_{l_{\square}}(P) = 4$. One can check using Mathematica that the only directions with corresponding width of $P$ less than or equal to 4 are the standard basis directions $\pm e_1, \pm e_2$, and $\pm e_3$. Hence $l_{\square}(P) = 4$, and the matrices that compute $l_{\square}(P)$ have $\pm e_1, \pm e_2, \pm e_3$ as their rows. One can easily check that $n_{l_{\Sigma}}(P) = 8$, while $l_{\Sigma}(P) = 7$, so none of these matrices compute $n_{l_{\Sigma}}(P)$. 


Matrix $A = \begin{bmatrix} -1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 1 & 1 \end{bmatrix}$ is one of the matrices that computes $\text{ls}_\Sigma(P)$. Since the only vectors $v$ with $w_v(P) \leq 4$ are the standard basis ones, the width of $P$ in the direction of $(0,1,1)$ is larger than 4, so $A$ does not compute $\text{nls}_\Sigma(P)$. Same is true about other matrices that compute $\text{ls}_\Sigma(P)$. This is because at least one of the rows in each such matrix is not of the form $\pm e_1, \pm e_2, \pm e_3$, so the corresponding width is greater than 4, and hence none of these matrices compute $\text{ls}_\Sigma(A)$. We have checked that in this example there is no overlap between the sets of matrices that compute $\text{ls}_\Sigma(P)$ and $\text{ls}_\Sigma(P)$, and therefore one cannot extend our 2D algorithm to a similar 3D one, based on the algorithm of [3] for computing $\text{ls}(P)$ for $P \subset \mathbb{R}^3$.

3. A similar 2D algorithm

Here we explain an algorithm for computing $\text{ls}_\Sigma(P)$ which takes longer than the algorithm from Section 1, but which might be generalizable to 3-polytopes. This algorithm is based on the following statement.

**Theorem 3.1.** Let $P \subset \mathbb{R}^2$ be a lattice polygon that satisfies $\text{nls}_\Sigma(P) = l_1(P) = l$. Suppose that we also have $\text{nls}_\Sigma(U_P) \geq l$ and $\text{nls}_\Sigma(L_P) \geq l$, where $U = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}$ and $L = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix}$. Then $\text{ls}_\Sigma(P) = \text{nls}_\Sigma(P)$.

**Proof.** Let $S = \begin{bmatrix} 0 & 1 \\ -1 & -1 \end{bmatrix}$ and $F = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$. Our goal is to check that we have $\text{ls}_\Sigma(Q) = \text{nls}_\Sigma(Q)$ and $w(Q) = w_{(1,0)}(Q)$, where $Q = AP$ for some $A$ in the subgroup $\langle S, F \rangle$, generated by $S$ and $T$, and then apply Theorem 1.8.

We observe that $w_{(1,0)}(SP) = w_{(0,1)}(P)$, $w_{(0,1)}(SP) = w_{(1,1)}(P)$, $w_{(1,1)}(SP) = w_{(1,0)}(P)$, while $F$ switches width in the basis directions and preserves the width in the direction of $(1,1)$. Hence for some $A \in \langle S, F \rangle$ we have $w_{(1,0)}(Q) \leq w_{(0,1)}(Q) \leq w_{(1,1)}(Q)$, where $Q = AP$.

If we apply $F$ to $P$, then as we observed in Lemma 1.3, $l_1$ and $l_2$ are fixed, while $l_3$ and $l_4$ switch, so $\text{nls}_\Sigma(FP) = l_1(FP) = l$. Also, since $UF = FL$ we get $\text{nls}_\Sigma(UFP) = \text{nls}_\Sigma(FLP) = \text{nls}_\Sigma(LP) \geq l$ and $\text{nls}_\Sigma(LFP) = \text{nls}_\Sigma(FUP) = \text{nls}_\Sigma(UP) \geq l$, that is, $FP$ satisfies the assumptions of the theorem.

We next check that this is also the case for $SP$. We saw in part (c) of Lemma 1.3 that $l_1(SP) = l_1(P) = l$. Similarly, $l_2(SP) = l_2(P)$, $l_3(SP) = l_1(U_P)$, and $l_4(SP) = l_2(U_P)$, so $\text{nls}_\Sigma(SP) = l_1(SP) = l$. Also, $\text{nls}_\Sigma(USP) = \text{nls}_\Sigma(-LP) = \text{nls}_\Sigma(LP) \geq l$ and $\text{nls}_\Sigma(LSP) = \text{nls}_\Sigma(P) \geq l$. Hence $SP$ satisfies the assumptions of the theorem.

We have checked that whenever a polygon $P$ satisfies the assumption of the theorem the same is true about $FP$ and $SP$, so we can now conclude that $Q = AP$ also satisfies the assumptions. Note that in particular we have $\text{nls}_\Sigma(Q) = l = \text{nls}_\Sigma(P)$. 
As before, we next shift $Q$ so that $\min_{(x,y) \in Q} x = \min_{(x,y) \in Q} y = 0,$ and denote

$$w_{(1,0)}(Q) = \max_{(x,y) \in Q} x = m, \quad w_{(0,1)}(Q) = \max_{(x,y) \in Q} y = k.$$  

We then have $m \leq k$ and $nls_{\square}(Q) = k$. Next,

$$l \leq l_3(Q) = \max_{(x,y) \in Q} y - \min_{(x,y) \in P} x + \max_{(x,y) \in P} (x - y) = k + \max_{(x,y) \in P} (x - y),$$

so we can conclude that $\max_{(x,y) \in P} (x - y) \geq l - k$. Since $Q$ has a point of the segment with the endpoints $(0, k)$ and $(l - k, k)$ we also have

$$\min_{(x,y) \in Q} (x - y) \leq \max\{-k, l - 2k\} = l - 2k.$$  

This implies that $w_{(1,-1)}(Q) \geq l - k - (l - 2k) = k$. Recall that we chose $Q = AP$ so that $w_{(1,1)}(Q) \geq k$. By Theorem 1.7 we conclude that $ls_{\square}(Q) = k = nls_{\square}(Q)$ and $w(Q) = w_{(1,0)}(Q) = m$. Then by By Theorem 1.8 we get $ls_{\Sigma}(Q) = nls_{\Sigma}(Q)$. Finally, we conclude $ls_{\Sigma}(P) = ls_{\Sigma}(Q) = nls_{\Sigma}(Q) = nls_{\Sigma}(P)$. 

Now one can use the above theorem for computing $ls_{\Sigma}(P)$: At each step of the algorithm first reflect $P$ in the $x$- and $y$-axes so that $nls_{\Sigma}(P) = l_1(P)$ and then check whether at least one of the inequalities $nls_{\Sigma}(LP) < l$ and $nls_{\Sigma}(UP) < l$ holds true. If this is the case, we pass from $P$ to $UP$ or $LP$, and repeat the step. Otherwise, we conclude that $ls_{\Sigma}(P) = nls_{\Sigma}(P)$. While, clearly, this algorithm takes longer than the one explained in Section 1, there is hope that this algorithm might be generalizable to 3-polytopes.

Such a generalization would be of the form: Suppose that $l_1(P) = nls_{\Sigma}(P) = l$ and that $l_1(AP) \geq l$ for some fixed finite set $S$ of unimodular matrices. Then $ls_{\Sigma}(P) = l$. Motivated by the 2D Theorem, one may hope that this would work if we take $S$ to be the set of all unimodular size 3 matrices whose entries are in the set $\{\pm 1, 0\}$. An example, provided by Abdulrahman Alajmi (private communication), demonstrates that this is not the case.

**Example 3.** Let $P = \text{conv}\{(0,0,0), (0,1,6), (1,3,1), (1,1,4)\}$. Then $l_1(P) = 7$ and $l_1(AP) \geq 7$ for all unimodular size 3 matrices whose entries are in the set $\{\pm 1, 0\}$.

(The second statement was checked using Mathematica.) Let $B = \begin{bmatrix} 2 & -1 & 0 \\ -1 & 0 & 1 \\ 1 & 0 & 1 \end{bmatrix}$. Then $l_1(BP) = 6$.

4. Lattice Size of $n$-dimension Lattice Polytopes.

It is explained in [3, 5] how one can generalize the standard width algorithm to obtain an algorithm for computing the lattice size of a lattice polytope $P \subset \mathbb{R}^n$ with respect to the unit cube $[0,1]^n$. This algorithm is quite time-consuming, but it works in any dimension $n$. We now explain how one can adjust this algorithm to compute lattice size of $P \subset \mathbb{R}^n$ with respect to the standard $n$-dimensional simplex.
Similarly to the 2D case, denote
\[ l_1(P) = \max_{(x_1, \ldots, x_n) \in P} (x_1 + x_2 + \cdots + x_n) - \min_{(x_1, \ldots, x_n) \in P} x_1 - \cdots - \min_{(x_1, \ldots, x_n) \in P} x_n. \]

Let \( l_1(P) = l \). If there exists a unimodular \( n \times n \) matrix \( A \) such that \( l_1(AP) \leq l - 1 \) then the width of \( P \) in the direction of each row vector of \( A \) is at most \( l - 1 \), since \( w_{e_i}(AP) \leq w_{e_i}((l - 1)\Sigma) \leq l - 1 \) for standard basis vectors \( e_i \) and

\[ w_{e_i}(AP) = \max_{x \in P} e_i \cdot (Ax) - \min_{x \in P} e_i \cdot (Ax) = \max_{x \in P} (A^T e_i) \cdot x - \min_{x \in P} (A^T e_i) \cdot x = w_{A^T e_i}(P). \]

Let \( M \) be the center of mass of \( P \) and let \( R \) be the radius of the largest sphere \( C \) centered at \( M \) that fits inside \( P \). We shift \( P \) so that the origin is at \( M \). If \( ||v|| > \frac{l - 1}{2R} \) then

\[ w_v(P) \geq w_v(C) = 2||v||R > l - 1. \]

Hence if we want to find \( A \) such that \( l_1(AP) \leq l - 1 \) we need to consider lattice vectors \( v \) with \( ||v|| \leq \frac{l - 1}{2R} \) and check if there are \( n \) of them that can be used as rows to form a unimodular matrix \( A \). The algorithm would then search through all possible size \( n \) collections of primitive lattice vectors in \( \mathbb{Z}^n \) with norm at most \( \frac{l - 1}{2R} \). For each such collection we would then check if it spans a parallelepiped of volume 1, and if \( l_1(AP) < l \) for some way of creating a unimodular matrix \( A \) using these vectors as rows in some order. The output is a unimodular matrix \( A \) with the smallest \( l_1(AP) \), which implies \( l_{\Sigma}(P) = l_1(AP) \).

Acknowledgments. Work of Soprunova and Tierney was partially supported by NSF Grant DMS-1156798.

References

[1] V. Arnold, *Statistics of integral convex polygons*, Functional Analysis and Its Applications 14(2), 1-3 (1980).
[2] I. Bárány and J. Pach, *On the number of convex lattice polygons*, Combinatorics, Probability and Computing 1, Issue 4 (1992).
[3] G. Brown, A. Kasprzyk, *Small polygons and toric codes*, Journal of Symbolic Computation, 51 p. 55 (2013).
[4] W. Castryck, F. Cools, *The lattice size of a lattice polygon*, Journal of Combinatorial Theory Series A 136, Issue C, 64-95 (2015).
[5] A. Harrison, J. Soprunova, *Lattice Size of 2D and 3D polytopes with respect to the unit cube*, preprint, arXiv:1709.03451 (2017).
[6] J. Lagarias, G. Ziegler, *Bounds for lattice polytopes containing a fixed number of interior points in a sublattice*, Canadian Journal of Mathematics 43(5), 1022-1035 (1991).
[7] J. Schicho, *Simplification of surface parametrizations - a lattice polygon approach*, Journal of Symbolic Computation 36(3-4), 535-554 (2003).
