Computing Minimal Generating Systems for Some Special Toric Ideals

Dimitrios I. Dais and Ioannis Markakis

Abstract. Let \( X_P \) be the projective toric surface associated to a lattice polygon \( P \). If the number of lattice points lying on the boundary of \( P \) is at least 4, it is known that \( X_P \) is embeddable into a suitable projective space as zero set of finitely many quadrics. In this case, the determination of a minimal generating system of the toric ideal defining \( X_P \) is reduced to a simple Gaussian elimination.

1. Introduction

Let \( P \subset \mathbb{R}^2 \) be a lattice polygon, i.e., a (convex, 2-dimensional) polygon, all of whose vertices belong to \( \mathbb{Z}^2 \). \( P \) is known to be normal and very ample, and to have a canonical presentation

\[
P = \left\{ x \in \mathbb{R}^2 \mid \langle x, u_F \rangle \geq -a_F \text{ for all } F \in \mathcal{F}(P) \right\},
\]

where \( \mathcal{F}(P) \) is the set of the facets (edges) of \( P \), \( a_F \in \mathbb{Z} \) and \( u_F \in \mathbb{Z}^2 \) the inward-pointing facet normal, i.e., the minimal generator of the ray \( \mathbb{R}_{\geq 0} u_F \). The corresponding compact complex toric surface \( X_P \) is therefore normal and projective, and

\[
D_P := \sum_{F \in \mathcal{F}(P)} a_F \text{orb}(\mathbb{R}_{\geq 0} u_F)
\]

is a very ample Cartier divisor on \( X_P \), where \( \text{orb}(\mathbb{R}_{\geq 0} u_F) \) is the Zariski closure of the orbit of the ray \( \mathbb{R}_{\geq 0} u_F \) w.r.t. the natural \( \text{Hom}_\mathbb{Z}(\mathbb{Z}^2, \mathbb{C}^*) \)-action. (See [4, Corollaries 2.2.13 and 2.2.19 (b), pp. 70-71, (4.2.6), p. 182, and Proposition 6.1.10 (c), p. 269].)

Setting \( \delta_P := \#(P \cap \mathbb{Z}^2) - 1 \), the complete linear system \( |D_P| \) induces the closed embedding \( \Phi|_{D_P}| \).

\[
\begin{array}{c}
\mathbb{T} \xrightarrow{\iota} X_P \xrightarrow{\Phi|_{D_P}|} \mathbb{P}^{\delta_P} \\
\pi \downarrow \quad \quad \quad \quad \downarrow \\
\mathbb{P}^{\delta_P} \end{array}
\]

with

\[
\mathbb{T} \ni t \mapsto (\Phi|_{D_P}| \circ \iota)(t) := [... : z_{(i,j)} : ...]|_{(i,j) \in P \cap \mathbb{Z}^2} \in \mathbb{P}^{\delta_P}, \quad z_{(i,j)} := \chi^{(i,j)}(t),
\]

where \( \chi^{(i,j)} : \mathbb{T} \to \mathbb{C}^* \) is the character associated to the lattice point \( (i,j) \) (with \( \mathbb{T} \) denoting the algebraic torus \( \text{Hom}_\mathbb{Z}(\mathbb{Z}^2, \mathbb{C}^*) \)), for all \( (i,j) \in P \cap \mathbb{Z}^2 \). The image \( \Phi|_{D_P}|(X_P) \) of \( X_P \) under \( \Phi|_{D_P}| \) is the Zariski closure of \( \text{Im}(\Phi|_{D_P}| \circ \iota) \) in \( \mathbb{P}^{\delta_P} \) and can be viewed as the projective variety \( \text{Proj}(S_P) \), where

\[
S_P := \mathbb{C}[C(P) \cap \mathbb{Z}^3] = \bigoplus_{k=0}^{\infty} \bigoplus_{(i,j) \in (\kappa P) \cap \mathbb{Z}^2} \mathbb{C} \chi^{(i,j)}(t)^k
\]

2010 Mathematics Subject Classification. 52B20 (Primary); 14M25, 14Q10 (Secondary).
Theorem 1.1. The zero set $V_{239-240}$ and Theorem 7.1.13, pp. 325-326. Equivalently, it can be viewed as the classical (smooth) quadric hypersurface in $\pi$ which is naturally graded by setting $\deg(C) = 2$. D.I. DAIS AND I. MARKAKIS

Proof. If $HP_2(P_\mathbb{C}^2) := \{\text{homogeneous polynomials (in } \delta + 1 \text{ variables)} \}$, then the $\mathbb{C}$-vector space homomorphism

$$f : HP_2(P_\mathbb{C}^2) \longrightarrow \mathbb{C}[x^{\pm 1}, y^{\pm 1}]$$

mapping $z_{(i,j)}$ onto $x^{i+1}y^{j+1}$, has as kernel $\text{Ker}(f)$ the $\mathbb{C}$-vector space of homogeneous polynomials of degree 2 which belong to $I_{\mathcal{A}_P}$ and as image $\text{Im}(f)$ the linear span of $\{x^iy^j \mid (i, j) \in 2P \cap \mathbb{Z}^2\}$ (because every lattice point in $2P$ is the sum of two lattice points of $P$, cf. [4, Theorem 2.2.12, pp. 68-69]). Taking into account Koelman's Theorem 1.1, [8, Lemma 4.1, p. 31], and the fact that $\mathbb{V}(I_{\mathcal{A}_P})$ is not contained in any hyperplane of $P_\mathbb{C}^2$, the equality $\dim_{\mathbb{C}}(\text{Ker}(f)) = \dim_{\mathbb{C}}(HP_2(P_\mathbb{C}^2)) - \dim_{\mathbb{C}}(\text{Im}(f))$ gives (1.1). \qed

Examples 1.3. (i) If $a, b$ are two positive integers, then the projective toric surface $X_{P_{a,b}} \cong \mathbb{V}(I_{\mathcal{A}_{P_{a,b}}}) \subset P_\mathbb{C}^{2\delta + 2}$ which is associated to the lattice quadrilateral

$$P_{a,b} := \text{conv}\{ (0, 0), (a, 0), (b, 1), (0, 1) \}$$

(where "conv" stands for convex hull, and $\delta_{P_{a,b}} = a + b + 1$), is isomorphic to the intersection of

$$\beta_{P_{a,b}} = \left(\frac{\delta_{P_{a,b}} + 2}{2}\right) - \frac{2P_{a,b} \cap \mathbb{Z}^2}{(a+b+2)(a+b+3)} - 3(a + b + 1) = \frac{1}{2} (a + b - 1)(a + b)$$

quadrics, i.e., to the rational normal scroll of type $(a, b)$ w.r.t. the homogeneous coordinates $[\ldots : z_{(i,j)} : \ldots]_{(i, j) \in P_{a,b} \cap \mathbb{Z}^2}$ satisfying the "2-minors condition"

$$\text{rank} \left( \begin{array}{cccccc} z_{(0,0)} & z_{(1,0)} & \cdots & z_{(a-1,0)} & z_{(0,1)} & z_{(1,1)} & \cdots & z_{(b-1,1)} \\ z_{(1,0)} & z_{(2,0)} & \cdots & z_{(a,0)} & z_{(1,1)} & z_{(2,1)} & \cdots & z_{(b,1)} \end{array} \right) \leq 1.$$
(ii) Let $d$ be a positive integer. The $\mathbb{C}$-vector space

$$\mathbb{C}[X_0, X_1, X_2]_d := \{ F \in \mathbb{C}[X_0, X_1, X_2] \mid F \text{ homogeneous of degree } d \} \cup \{0\}$$

has the set $\{X_0^{\nu_1} X_1^{\nu_2} X_2^{\nu_3} \mid (\nu_1, \nu_2, \nu_3) \in \mathbb{E}_{2,d}\}$ as one of its bases, where

$$\mathbb{E}_{2,d} := \{ \alpha = (\nu_1, \nu_2, \nu_3) \in \mathbb{Z}^3 \mid \nu_1, \nu_2, \nu_3 \in [0, d] \text{ and } \alpha_0 + \alpha_1 + \alpha_2 = d \}.$$  

For each $\alpha = (\nu_1, \nu_2, \nu_3) \in \mathbb{E}_{2,d}$ we write $X^\alpha := X_0^{\nu_1} X_1^{\nu_2} X_2^{\nu_3}$. Setting

$$\mathcal{T}_d := \text{conv}(\{(0,0), (d,0), (0,d)\}) = \{(x_1, x_2) \in \mathbb{R}^2 \mid x_1 \geq 0, x_2 \geq 0 \text{ and } x_1 + x_2 \leq d \}$$

we see that $\sharp(\partial \mathcal{T}_d \cap \mathbb{Z}^2) = 3d$ and $\sharp(\mathcal{T}_d \cap \mathbb{Z}^2) = \binom{d+2}{2}$, because

$$\mathcal{T}_d \cap \mathbb{Z}^2 \ni (m_1, m_2) \mapsto (m_1+1, m_1+m_2+2) \in \{(\xi_1, \xi_2) \in \mathbb{Z}_{\geq 0}^2 \mid 1 \leq \xi_1 < \xi_2 \leq d + 2\}$$

is a bijective map with $\sharp\{(\xi_1, \xi_2) \in \mathbb{Z}_{\geq 0}^2 \mid 1 \leq \xi_1 < \xi_2 \leq d + 2\} = \binom{d+2}{2}$. On the other hand,

$$\sharp(\mathbb{E}_{2,d}) = \sharp\{(\nu_1, \nu_2, \nu_3) \in \mathbb{Z}^3 \mid \nu_1, \nu_2, \nu_3 \in [0, d] \text{ and } \nu_0 + \nu_1 + \nu_2 \leq d \}$$

and

$$(\nu_1, \nu_2, \nu_3) \in \mathbb{Z}^3 \mid \nu_1, \nu_2, \nu_3 \in [0, d-1] \text{ and } \nu_0 + \nu_1 + \nu_2 \leq d - 1 \}$$

is isomorphic to the image of the so-called $d$-uple Veronese embedding

$$\nu_{2,d} : \mathbb{P}^1_{\mathbb{C}} \hookrightarrow \mathbb{P}^\delta_{\mathbb{C}}$$

where the monomials $\{X^\alpha \mid \alpha \in \mathbb{E}_{2,d}\}$ are arranged in a prescribed manner (e.g., lexicographically). In fact, in this case,

$$\beta_{\mathcal{T}_d} = \binom{3d+2}{2} - \sharp(\mathcal{T}_d \cap \mathbb{Z}^2)$$

is

$$= \frac{1}{2} \binom{d+2}{2} \left( \binom{d+2}{2} + 1 \right) - (2d^2 + 3d + 1)$$

and

$$= \frac{(d+1)(d+2)}{4} \left( \frac{(d+1)(d+2)}{2} + 1 \right) - (2d^2 + 3d + 1) = \frac{1}{2} d (d + 6) \left( d^2 - 1 \right).$$

\[(1.2)\]
An algorithm, implemented in Python to compute a minimal generating set for the ideal $I_{AP}$, given the vertex set $V(P)$ of the polygon $P$ is provided in the library \texttt{toricIdeal.py}. The algorithm is provided by the routine \texttt{minGenSet}, which receives a list of vertices and calls five subroutines to compute a minimal generating set of $I_{AP}$.

```python
import numpy

def minGenSet(p):
    intp = integerPoints(*vertToConst(p))
    (basis, genBin) = genBinom(intp)
    indepCol = findIndepCol(genBin)
    binom = findBinom(basis, genBin, indepCol)
    return (intp, binom)
```

More specifically, the first subroutine, \texttt{vertToConst}, produces a complete system of facet-defining inequalities from the list $V(P)$ and lower and upper bounds for the coordinates of the points of the polygon. First, the facets are distinguished among the line segments connecting any two vertices, using the fact that the polygon lies entirely in one of the two closed half-planes bounded by their supporting lines and then a constraint is created for each facet.

```python
def vertToConst(p):
    A = []
    b = []
    for i in range(len(p)-1):
        for j in range(i+1, len(p)):
            big = True
            small = True
            coef = numpy.array([[p[i,1]-p[j,1]], [p[j,0]-p[i,0]]])
            cst = numpy.inner(coef, p[j])
            for point in p:
                diff = numpy.inner(coef, point) - cst
                if diff > 0:
                    small = False
                elif diff < 0:
                    big = False
            if big:
                A.append(coef)
                b.append(cst)
            if small:
                A.append(-coef)
                b.append(-cst)
    A = numpy.array(A)
```
b=numpy.array(b)
x=[min(p[:,0]),max(p[:,0])]
y=[min(p[:,1]),max(p[:,1])]
return(A,b,x,y)

The second one, integerPoints, uses the constraints and by brute force finds the lattice points of the polygon.

def integerPoints(A,b,x,y):
    intPoints=[]
    for x0 in range(x[0],x[1]+1):
        for y0 in range(y[0],y[1]+1):
            point=numpy.array([x0,y0])
            diff=numpy.dot(A,point)-b
            inInterior=True
            for coord in diff:
                if coord<0:
                    inInterior=False
                    break
            if inInterior:
                intPoints.append(point)
    intPoints=numpy.array(intPoints)
    return(intPoints)

The third one, genBinom, orders the basis elements

$$B = \{z_{(i,j)}z_{(i',j')} : (i,j),(i',j') \in P \cap \mathbb{Z}^2\}$$

of the $\mathbb{C}$-vector space $\text{HP}^2(\mathbb{P}_C^d)$ and finds a generating set for the ideal $I_{A_P}$ by collecting all vectorial relations of the form

$$(i_1,j_1)+(i_2,j_2)=(i'_1,j'_1)+(i'_2,j'_2)$$

where $(i_1,j_1),(i_2,j_2),(i'_1,j'_1),(i'_2,j'_2) \in P \cap \mathbb{Z}^2$. This is returned as a matrix containing the coefficients of the generating binomials w.r.t. the ordered basis $B$.

def genBinom(intPoints):
    basis=[]
    for i in range(len(intPoints)):
        for j in range(i,len(intPoints)):
            bp=intPoints[i].tolist()+intPoints[j].tolist()
            basis.append(bp)
    basis=numpy.array(basis)
    genBin=[[]]
    for i in range(len(basis)):
        for j in range(i+1,len(basis)):
            if basis[i,0]+basis[i,2]==basis[j,0]+basis[j,2] and basis[i,1]+basis[i,3]==basis[j,1]+basis[j,3]:
                row=numpy.zeros(len(basis),dtype=numpy.int)
                row[i]=1
                row[j]=-1
                genBin.append(row)
    genBin=numpy.transpose(numpy.array(genBin))
    return(basis,genBin)
The fourth one, `findIndepCol`, performs a Gauss elimination on a matrix and finds a basis of its column space by collecting the non-zero columns of the row echelon form of it.

```python
from scipy.linalg import lu

def findIndepCol(A):
    U, permute, l = lu(A, permute_l=True)
    indepCol = []
    for i in range(len(U)):
        for j in range(len(U[i])):
            if U[i, j] != 0:
                indepCol.append(j)
                break
    return (indepCol)
```

Finally, the fifth subroutine, `findBinom`, uses the matrix given by `genBinom` and the list of $C$-linearly independent columns found by `findIndepCol` to produce a set of $C$-linearly independent generating binomials of the ideal $I_{AP}$.

```python
def findBinom(basis, genBin, indepCol):
    binom = []
    for i in indepCol:
        j1 = -1
        j2 = -1
        for j in range(len(genBin)):
            if genBin[j, i] == 1:
                j1 = j
            elif genBin[j, i] == -1:
                j2 = j
            if j1 != -1 and j2 != -1:
                binomial = "z_{{{}^{\{\{}\{\}}}}} - z_{{{}^{\{\{}\{\}}}}}.format(basis[j1]|0], basis[j1]|1], basis[j1]|2], basis[j1]|3], basis[j2]|0], basis[j2]|1], basis[j2]|2], basis[j2]|3])
                binom.append(binomial)
    return (binom)
```

The complexity of the `minGenSet` is polynomial of the class $O(m^4 + n^2)$, where $n$ is the number of vertices and $m$ an integer bounding absolutely the coordinates of the vertices.

3. Applications

**Veronese surfaces.** If $P = Tr_2 := \text{conv}(\{(0,0),(2,0),(0,2)\})$ (with $d = 2$ as in 1.3 (ii)), then the algorithm produces the following minimal generating set of $I_{Tr_2}$:

- $z_{(0,0)}z_{(2,0)} - z_{(1,1)}^2$,
- $z_{(0,0)}z_{(2,0)} - z_{(0,1)}^2$.

Analogously, if $P = Tr_3 := \text{conv}(\{(0,0),(3,0),(0,3)\})$ (with $d = 3$), then the 27 quadrics
generate minimally $I_{A_{\Delta_{3}}}$ (cf. (1.2)).

> **Toric log del Pezzo surfaces.** These are of the form $X_P$, where $P = tQ$ is the polar of an LDP-polygon $Q \subset \mathbb{R}^2$ dilated by its index $t$. (An LDP-polygon $Q \subset \mathbb{R}^2$ is a convex polygon which contains the origin in its interior, and its vertices belong to $\mathbb{Z}^2$ and are primitive. The index of a polygon of this kind is defined to be $\ell := \min\{ \kappa \in \mathbb{Z}_{>0} \mid \mathcal{V}(\kappa Q) \subset \mathbb{Z}^2 \}$.)

Kasprzyk, Kreuzer & Nill [6, §6] developed an algorithm by means of which one creates an LDP-polygon, for given $\ell \geq 2$, by fixing a “special” edge and following a prescribed successive addition of vertices, and produced in this way the long lists of all LDP-polygons for $\ell \leq 17$. An explicit study for each of these 15346 LDP-polygons is available on the webpage [1].

(i) Up to unimodular transformation the only reflexive hexagon (i.e., the only LDP-hexagon of index 1) is

$$Q := \text{conv}\left\{(0,1), (1,1), (1,0), (0,-1), (-1,-1), (-1,0)\right\}$$

having

$$\hat{Q} := \text{conv}\left\{(1,0), (1,-1), (0,-1), (-1,0), (-1,1), (0,1)\right\}$$

as its polar, and $X_{\hat{Q}} \cong \mathcal{V}(I_{A_{\hat{Q}}}) \subset \mathbb{P}^6_{\mathbb{C}}$ with $I_{A_{\hat{Q}}}$ minimally generated by the 9 quadrics:

$$z(-1,0)z(1,1) - z(0,-1)z(0,0), \quad z(-1,0)z(1,0) - z^2(0,0), \quad z(0,-1)z(1,0) - z(-1,1)z(1,-1),$$

$$z(-1,0)z(0,0) - z(-1,1)z(0,-1), \quad z(-1,0)z(0,1) - z(-1,1)z(0,0), \quad z(0,-1)z(1,0) - z(0,0)z(1,-1),$$

$$z(-1,0)z(0,1) - z(0,-1)z(0,0), \quad z(-1,1)z(0,0) - z(0,0)z(0,1), \quad z(0,0)z(1,0) - z(0,1)z(1,-1).$$

(ii) For the LDP-triangle $Q$ of index 2 with vertex set

$$\mathcal{V}(Q) := \{(0,1), (8,1), (-4,-1)\}$$

we obtain

$$\mathcal{V}(2\hat{Q}) = \{(1,-2), (0,-2), (-1,6)\}\text{ (cf. Fig. 2)}$$

and $X_{2\hat{Q}} \cong \mathcal{V}(I_{A_{2\hat{Q}}}) \subset \mathbb{P}^6_{\mathbb{C}}$ with $I_{A_{2\hat{Q}}}$ minimally generated by the 7 quadrics:

$$z(-1,0)z(1,-2) - z^2(0,2), \quad z(0,0)z(0,2) - z(0,1)z(0,1),$$

$$z(0,-1)z(0,2) - z(0,0)z(0,1), \quad z(0,-2)z(0,2) - z^2(0,0),$$

$$z(0,-2)z(0,1) - z(0,-1)z(0,0), \quad z(0,-2)z(0,0) - z(0,-1)z(0,1),$$

$$z(0,-2)z(0,1) - z(0,-1)z(0,0).$$
(iii) For the LDP-pentagon $Q$ of index 3 with vertex set
$$V(Q) := \{(0, 1), (1, 1), (1, 0), (-2, -1), (-3, -1)\}$$
(which is unimodularly equivalent to the pentagon “$Q_2^{[3]}$” of [5]) we obtain
$$V(3Q) = \{(2, -3), (0, -3), (-3, 0), (-3, -9), (0, 3)\},$$
and $X_{3Q} \cong \mathcal{V}(I_{A_{3Q}}) \subset \mathbb{P}_{\mathbb{C}}^{38}$ with $I_{A_{3Q}}$ minimally generated by a set of 646 quadrics!

(iv) For the LDP-quadrilateral $Q$ of index 4 with vertex set
$$V(Q) := \{(-1, 2), (3, 2), (-1, -1), (-3, -2)\}$$
we obtain $V(4Q) = \{(2, -1), (0, -2), (-12, 16), (-4, 8)\}$, and $X_{4Q} \cong \mathcal{V}(I_{A_{4Q}}) \subset \mathbb{P}_{\mathbb{C}}^{45}$ with $I_{A_{4Q}}$ minimally generated by a set of 918 quadrics!

(v) Finally, the LDP-triangle $Q$ of index 5 with vertex set
$$V(Q) := \{(0, 1), (15, 1), (-15, -2)\}$$
we obtain $V(5Q) = \{(1, -5), (0, -5), (-1, 10)\}$, and $X_{5Q} \cong \mathcal{V}(I_{A_{5Q}}) \subset \mathbb{P}_{\mathbb{C}}^{48}$ with $I_{A_{5Q}}$ minimally generated by the following 21 quadrics:

- $z_{(0, -5)} z_{(0, -1)} - z_{(0, -3)}^2$
- $z_{(0, -5)} z_{(0, 1)} - z_{(0, -4)} z_{(0, 0)}$
- $z_{(0, -5)} z_{(0, 0)} - z_{(0, -3)} z_{(0, -2)}$
- $z_{(0, -5)} z_{(0, 2)} - z_{(0, -4)} z_{(0, 1)}$
- $z_{(0, -5)} z_{(0, 1)} - z_{(0, -3)} z_{(0, -1)}$
- $z_{(0, -5)} z_{(0, 3)} - z_{(0, -4)}^2$
- $z_{(0, -5)} z_{(0, 0)} - z_{(0, -4)} z_{(0, -1)}$
- $z_{(0, -2)} z_{(0, 2)} - z_{(0, 0)}^2$
- $z_{(0, -2)} z_{(0, 1)} - z_{(0, 0)} z_{(0, 1)}$
- $z_{(0, -5)} z_{(0, -2)} - z_{(0, -4)} z_{(0, -3)}$
- $z_{(0, -5)} z_{(0, -3)} - z_{(0, -1)} z_{(0, 0)}$
- $z_{(0, -5)} z_{(0, 2)} - z_{(0, -3)} z_{(0, 1)}$
References

[1] Brown G. & Kasprzyk A.M.: The graded ring database homepage, online access via http://www.grdb.co.uk/.

[2] Castryck W. & Cools F.: A minimal set of generators for the canonical ideal of a non-degenerate curve, J. of the Australian Math. Soc. 98 (2015), 311-323.

[3] , canonical.m, Magma code accompanying the article [2]; access via http://math.univ-lille1.fr/~castryck/code/canonical.m.

[4] Cox D.A., Little J.B. & Schenck H.K.: Toric Varieties, Graduate Studies in Mathematics, Vol. 124, American Math. Soc., 2011.

[5] Dais D.I.: Toric log del Pezzo surfaces with one singularity, https://arxiv.org/abs/1705.06359, preprint, 2017.

[6] Kasprzyk A.M., Kreuzer M. & Nill B.: On the combinatorial classification of toric log del Pezzo surfaces, LMS Journal of Computation and Mathematics 13 (2010), 33-46.

[7] Koelma R.: A criterion for the ideal of a projectively embedded toric surface to be generated by quadrics, Beiträge zur Algebra und Geometric 34 (1993), 57-62.

[8] Sturmfels B.: Gröbner Bases and Convex Polytopes, University Lecture Series, Vol. 8, American Math. Soc., 1996.

University of Crete, Department of Mathematics and Applied Mathematics, Division Algebra and Geometry, Voutes Campus, P.O. Box 2208, GR-70013, Heraklion, Crete, Greece

E-mail address: ddais@math.uoc.gr/johnmarkakis95@gmail.com