Monomial Ideals

Serkan Hoşten[1] and Gregory G. Smith[2]

[1] Department of Mathematics, San Francisco State University, San Francisco, CA
[2] Department of Mathematics, University of California, Berkeley, CA

Abstract. Monomial ideals form an important link between commutative algebra and combinatorics. In this chapter, we demonstrate how to implement algorithms in *Macaulay 2* for studying monomial ideals. We illustrate these methods with a broad range of examples from combinatorics, integer programming and algebraic geometry.

Introduction

An ideal I in $S = \mathbb{Q}[x_1, \ldots, x_n]$ is called a monomial ideal if it satisfies one of the following equivalent conditions:

(a) I is generated by monomials,
(b) if $f = \sum_{\alpha \in \mathbb{N}^n} k_\alpha x^\alpha$ belongs to I then $x^\alpha \in I$ whenever $k_\alpha \neq 0$,
(c) I is torus-fixed; in other words I is fixed by the standard action of the diagonal matrices.

It follows that a monomial ideal is uniquely determined by the monomials it contains. Most operations are far simpler for a monomial ideal than for an ideal generated by arbitrary polynomials. In particular, many invariants can be effectively determined for monomial ideals. As a result, one can solve a broad collection of problems by reducing to or encoding data in a monomial ideal. The aim of this chapter is to develop the computational aspects of monomial ideals in *Macaulay 2* and demonstrate a wide range of applications.

This chapter is divided into five sections. Each section begins with a discussion of a computational procedure involving monomial ideals. Algorithms are presented as *Macaulay 2* functions. We demonstrate these procedures by solving problems from other areas of mathematics. In particular, we include the *Macaulay 2* code for generating many interesting families of monomial ideals. The first section introduces the basic functions on monomial ideals in *Macaulay 2*. As an example of these functions, we use the Stanley-Reisner ideal associated to a simplicial complex to compute its f-vector. Next, we present two algorithms for finding a primary decomposition of a monomial ideal. We illustrate these techniques by using graph ideals to study matchings in bipartite graphs. The third section focuses on the standard pairs of a monomial ideal; two methods are given for finding the set of standard pairs. As an application, we use standard pairs to solve integer linear programming problems. The fourth section examines Borel-fixed ideals and generic initial ideals. Combining these constructions with distractions, we demonstrate that a Hilbert scheme is connected. Finally, we look at the chains of associated primes in various families of monomial ideals.

1 The Basics of Monomial Ideals

Creating monomial ideals in *Macaulay 2* is analogous to creating general ideals. The monomial ideal generated by a sequence or list of monomials can be constructed with the function `monomialIdeal`.

```
i1 : load "monideal.m2"

i2 : S = QQ[a, b, c, d];

i3 : I = monomialIIdeal(a^2, a*b, b^3, a*c)

                     2        3
o3 = monomialIdeal (a , a*b, b , a*c)

o3 : MonomialIIdeal of S

i4 : J = monomialIIdeal{a^2, a*b, b^2}

                     2        2
o4 = monomialIdeal (a , a*b, b )

o4 : MonomialIIdeal of S
```

The type `MonomialIdeal` is the class of all monomial ideals. If an entry in the sequence or list is not a single monomial, then `monomialIdeal` takes only the leading monomial; recall that every polynomial ring in *Macaulay 2* is equipped with a monomial ordering.

```
i5 : monomialIIdeal( a^2+a*b, a*b+3, b^2+d)

                     2        2
o5 = monomialIdeal (a , a*b, b )

o5 : MonomialIIdeal of S
```

There are also several different methods of associating monomial ideals to arbitrary ideals in a polynomial ring. The most important of these is the initial ideal; the monomial ideal generated by the leading monomials of all elements in the given ideal. When applied to an ideal, the function `monomialIdeal` returns the initial ideal.

```
i6 : K = ideal( a^2, b^2, a*b+b*c)

             2   2
o6 = ideal (a , b , a*b + b*c)

o6 : Ideal of S

i7 : monomialIIdeal K

                     2        2      2
o7 = monomialIdeal (a , a*b, b , b*c )

o7 : MonomialIIdeal of S
```

This is equivalent to taking the leading monomials of a Gröbner basis for `K`. In our example, the given generators for `K` are not a Gröbner basis.

```
i8 : monomialIdeal gens K

                         2        2
o8 = monomialIdeal (a , a*b, b )

o8 : MonomialIdeal of S
```

One can also test if a general ideal is generated by monomials with the function `isMonomialIdeal`.

```
i9 : isMonomialIdeal L

o9 = false

i10 : isMonomialIdeal ideal(a^5, b^2*c, d^11)

o10 = true
```

The usual algebraic operations on monomial ideals are the same as on general ideals.

```
i11 : I+J

                          2        2
o11 = monomialIdeal (a , a*b, b , a*c)

o11 : MonomialIdeal of S

i12 : intersect(I, J)

                          2        3
o12 = monomialIdeal (a , a*b, b )

o12 : MonomialIdeal of S

i13 : I*J

                          4   3    2 2     3   5   3    2        2
o13 = monomialIdeal (a , a b, a b , a*b , b , a c, a b*c, a*b c)

o13 : MonomialIdeal of S

i14 : I:J

o14 = monomialIdeal (a, b)

o14 : MonomialIdeal of S

i15 : saturate(I,J)

o15 = monomialIdeal 1

o15 : MonomialIdeal of S

i16 : radical(I)

o16 = monomialIdeal (a, b)

o16 : MonomialIdeal of S
```

Example: Stanley-Reisner ideals

Radical monomial ideals — ideals generated by squarefree monomials — have a beautiful combinatorial interpretation in terms of simplicial complexes. More explicitly, a simplicial complex Δ on the vertex set $V = \{v_1, \dots, v_n\}$ corresponds to the ideal I_Δ in $S = \mathbb{Q}[x_1, \dots, x_n]$ generated by all monomials $x_{i_1} \cdots x_{i_p}$ such that $\{v_{i_1}, \dots, v_{i_p}\} \notin \Delta$. The ideal I_Δ is called the Stanley-Reisner ideal of Δ.

To illustrate the connections between Stanley-Reisner ideals and simplicial complexes, we consider the f-vector. Perhaps the most important invariant of a simplicial complex, the f-vector of a d-dimensional simplicial complex Δ is $(f_{-1}, f_0, f_1, \dots, f_d)$ in $\mathbb{N}^{d+2}$, where f_i denotes the number of i-dimensional faces in Δ. From the monomial ideal point of view, the f-vector is encoded in the Hilbert series of the quotient ring S/I_Δ as follows:

Theorem 1.1. *If Δ is a simplicial complex with f-vector $(f_{-1}, f_0, \dots, f_d)$, then the Hilbert series of I_Δ is*

$$H_{S/I_\Delta}(t) = \sum_{i=0}^{d+1} \frac{f_{i-1} t^i}{(1-t)^i} .$$

Since $H_{S/I_\Delta}(t)$ is typically expressed in the form $\frac{h_0 + h_1 t + \cdots + h_d t^d}{(1-t)^{d+1}}$, we can obtain the f-vector by using the identity $\sum_i h_i t^i = \sum_{j=0}^{d} f_{i-1} t^i (1-t)^{d-i}$. In particular, we can compute f-vectors from Stanley-Reisner ideals as follows:

```
i17 : fvector = I -> (
           R := (ring I)/I;
           d := dim R;
           N := numerator hilbertSeries R;
           h := apply(reverse toList(0..d), i ->
                if first exponents leadMonomial N === i then (
                     c := leadCoefficient N;
                     N = N - leadTerm N;
                     c)
                else 0);
           f := j -> sum(0..j+1, i -> binomial(d-i, j+1-i)*h#(d-i));
           apply(toList(0..d-1), j -> f(j)));
```

For example, we can demonstrate that the f-vector of the octahedron is $(6, 12, 8)$.

```
i18 : S = QQ[a .. f];

i19 : octahedron = monomialIIdeal(a*b, c*d, e*f)

o19 = monomialIdeal (a*b, c*d, e*f)

o19 : MonomialIIdeal of S

i20 : fvector(octahedron)

o20 = {6, 12, 8}

o20 : List
```

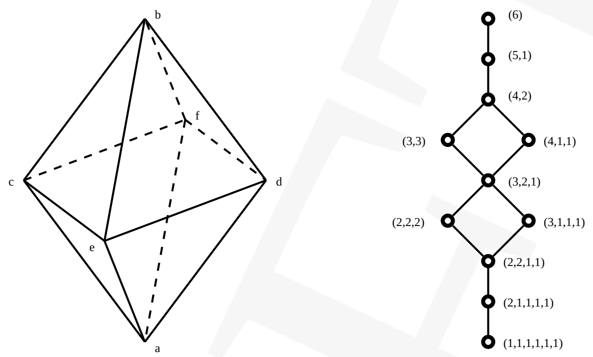

Fig. 1. The octahedron and dominance poset $(n = 6)$

For more examples, we consider the special class of simplicial complexes arising from posets, partially order sets. The order complex $\Delta(\Pi)$ of a poset Π is the simplicial complex whose simplices are the chains of Π. Recall that a subset C of Π is a chain if any two elements of C are comparable. In this situation, the Stanley-Reisner ideal $I_{\Delta(\Pi)}$ is generated by all monomials $x_i x_j$ for which $i \in \Pi$ and $j \in \Pi$ are incomparable. The next function produces the Stanley-Reisner ideal corresponding to the order complex of the set of permutations under the dominance order.

```
i21 : dominance = (p,q) -> all(toList(0 .. min(#p, #q)-1),
            i -> sum(p_{0 .. i}) <= sum(q_{0 .. i}));

i22 : dominancePoset = n -> (
            P := partitions n;
            X := apply(P, p -> x_p);
            -- NOTE: changes current ring
            S = QQ[X];
            X = gens S;
            monomialIIdeal flatten table(#P, #P, (i,j) ->
                      if dominance(P#i, P#j) or dominance(P#j, P#i)
                      then 0_S
                      else X#i * X#j));
```

For partitions of 6 and 7, we have

```
i23 : dominancePoset 6

o23 = monomialIdeal (x            x       , x               x          )
                      {4, 1, 1} {3, 3}    {3, 1, 1, 1} {2, 2, 2}

o23 : MonomialIIdeal of S

i24 : fvector(dominancePoset 6)

o24 = {11, 53, 147, 259, 301, 231, 113, 32, 4}

o24 : List

i25 : fvector(dominancePoset 7)

o25 = {15, 101, 404, 1070, 1978, 2618, 2500, 1709, 815, 257, 48, 4}

o25 : List
```

For more on the algebraic aspects of Stanley-Reisner ideals see Bruns and Herzog [2].

2 Primary Decomposition

A primary decomposition of an ideal I is an expression of I as a finite intersection of primary ideals; an ideal I is called primary if $r_1 r_2 \in I$ implies either $r_1 \in I$ or $r_2^\ell \in I$ for some $\ell > 0$. Calculating a primary decomposition of an arbitrary ideal in a polynomial ring is quite difficult. However, for monomial ideals the task is relatively simple.

We first present a recursive method for generating an irreducible primary decomposition. It is based on the following two observations.

Lemma 2.1. *Let I be a monomial ideal.*

(1) *If I is generated by pure powers of a subset of the variables, then it is a primary ideal.*
(2) *If r is minimal generator of I such that $r = r_1 r_2$ where r_1 and r_2 are relatively prime, then $I = \big(I + \langle r_1 \rangle\big) \cap \big(I + \langle r_2 \rangle\big)$.*

The following is an implementation of the resulting algorithm:

```
i26 : clearAll

i27 : load "monideal.m2"

i28 : supp = r -> select(gens ring r, e -> r % e == 0);

i29 : isIrreducible = I -> all(first entries gens I,
          r -> #supp(r) < 2);

i30 : monomialDecompose = method();

i31 : monomialDecompose List := L -> (
          P := select(L, I -> isIrreducible(I) === false);
          if #P > 0 then (
               I := first P;
               m := first select(first entries gens I,
                    r -> #supp(r) > 1);
               E := first exponents m;
               i := position(E, e -> e =!= 0);
               r1 := product apply(E_{0..i}, (gens ring I)_{0..i},
                    (j, r) -> r^j);
               r2 := m // r1;
               monomialDecompose( delete(I, L) | {I +
                    monomialIIdeal(r1), I + monomialIIdeal(r2)}))
          else L);

i32 : monomialDecompose MonomialIIdeal := I -> monomialDecompose {I};
```

Here are a couple of small examples illustrating this method.

```
i33 : S = QQ[a,b,c,d];

i34 : I1 = monomialIIdeal(a^2*b*c, a*b^2*c, a*b*c^2)

                       2        2        2
o34 = monomialIdeal (a b*c, a*b c, a*b*c )

o34 : MonomialIIdeal of S

i35 : P1 = monomialDecompose I1;
```

```
i36 : scan(P1, J -> << endl << J << endl);

monomialIdeal a

monomialIdeal b

monomialIdeal c

                  2
monomialIdeal (a , c)

                     2
monomialIdeal (a, b )

                  2
monomialIdeal (a , b)

                  2   2   2
monomialIdeal (a , b , c )

i37 : I1 == intersect(P1)

o37 = true

i38 : I2 = monomialIIdeal(a^3*b, a^3*c, a*b^3, b^3*c, a*c^3, b*c^3)

                        3       3    3    3       3      3
o38 = monomialIdeal (a b, a*b , a c, b c, a*c , b*c )

o38 : MonomialIIdeal of S

i39 : P2 = monomialDecompose I2;

i40 : scan(P2, J -> << endl << J << endl);

monomialIdeal (b, c)

monomialIdeal (a, c)

                  3   3   3
monomialIdeal (a , b , c )

monomialIdeal (a, b)

                  3      3
monomialIdeal (a , b, c )

monomialIdeal (a, b)

                     3   3
monomialIdeal (a, b , c )

i41 : I2 == intersect(P2)

o41 = true
```

As we see from these examples, this procedure doesn't necessarily produce an irredundant decomposition.

The second algorithm for finding a primary decomposition of a monomial ideal I is based on the Alexander dual of I. The Alexander dual was introduced first for squarefree monomial ideals. In this case, it is the monomial ideal of the dual of the simplicial complex corresponding to I. The following general definition appears in Miller [9]. If $I \subseteq \mathbb{Q}[x_1, \ldots, x_n]$ is a monomial ideal and x^λ is the least common multiple of the minimal generators of I, then the Alexander dual of I is

$$I^\vee = \left\langle \prod_{\beta_i > 0} x_i^{\lambda_i + 1 - \beta_i} : \begin{array}{l} \langle x_1^{\beta_1}, \ldots, x_n^{\beta_n} \rangle \text{ is an irredundant} \\ \text{irreducible component of } I \end{array} \right\rangle .$$

The next proposition provides a useful way of computing $I^\vee$ given a set of generators for I (see section 2 in Miller [9]).

Proposition 2.2. *If I is a monomial ideal and x^λ is the least common multiple of the minimal generators of I, then the generators for $I^\vee$ are those generators of the ideal $\left(\langle x_1^{\lambda_1+1}, \ldots, x_n^{\lambda_n+1} \rangle : I\right)$ that are not divisible by $x_i^{\lambda_i+1}$ for $1 \le i \le n$.*

The resulting algorithm for computing primary decomposition is available in *Macaulay 2* under the name `primaryDecomposition`.

```
i42 : code(primaryDecomposition, MonomialIdeal)

o42 = -- monideal.m2:261-270
      primaryDecomposition MonomialIdeal := (I) -> (
           R := ring I;
           aI := lcmOfGens I;
           M := first entries gens dual I;
           L := unique apply(#M, i -> first exponents M_i);
           apply(L, i -> monomialIdeal apply(#i, j -> (
                           if i#j === 0 then 0_R
                           else R_j^(aI#j+1-i#j)
                           )))
           )

o42 : Net
```

This direct algorithm is more efficient than our recursive algorithm. In particular, it gives an irredundant decomposition. For example, when we use it to determine a primary decomposition for the ideals `I1` and `I2` above, we obtain

```
i43 : L1 = primaryDecomposition I1;

i44 : scan(L1, J -> << endl << J << endl);

monomialIdeal c

monomialIdeal a

monomialIdeal b

                  2   2   2
monomialIdeal (a , b , c )

i45 : I1 == intersect L1

o45 = true

i46 : L2 = primaryDecomposition I2;
```

```
i47 : scan(L2, J -> << endl << J << endl);

                  3   3   3
monomialIdeal (a , b , c )

monomialIdeal (b, c)

monomialIdeal (a, b)

monomialIdeal (a, c)

i48 : I2 == intersect(L2)

o48 = true
```

For a family of larger examples, we consider the tree ideals:

$$\left\langle \left(\prod_{i \in F} x_i\right)^{n-|F|+1} : \emptyset \neq F \subseteq \{x_1, \ldots, x_n\} \right\rangle .$$

These ideals are so named because their standard monomials (monomials not in the ideal) correspond to trees on $n+1$ labeled vertices. We determine the number of irredundant irreducible components as follows:

```
i49 : treeIdeal = n -> (
          S = QQ[vars(0..n-1)];
          L := delete({}, subsets gens S);
          monomialIIdeal apply(L, F -> (product F)^(n - #F +1)));

i50 : apply(2..6, i -> #primaryDecomposition treeIdeal i)

o50 = (2, 6, 24, 120, 720)

o50 : Sequence
```

Example: graph ideals

As a simple application of primary decomposition, we consider graph ideals and matching theory. If G is a graph with vertices $V = \{v_1, \ldots, v_n\}$, then the associated ideal I_G in $\mathbb{Q}[x_1, \ldots, x_n]$ is generated by the set of monomials $x_i x_j$ such that v_i is connected to v_j by an edge of G. A subset $W \subseteq V$ is called a covering of G if each edge in G is incident to at least one vertex in W. Graph ideals encode coverings as follows:

Lemma 2.3. *A subset $W \subseteq V$ is a minimal covering of G if and only if the ideal $P = \langle x_i : v_i \in W \rangle$ is a minimal prime over I_G.*

Since I_G is a squarefree monomial ideal, its primary decomposition is the intersection of the minimal primes containing it.

A subset M of the edges of G is called a matching if no two edges in M have a vertex in common. The proposition below is one of the basic results linking matchings and coverings of bipartite graphs. A graph G is bipartite if its vertex set V can be partitioned into two subsets V_1 and V_2 such that every edge of G has one vertex in V_1 and the other V_2.

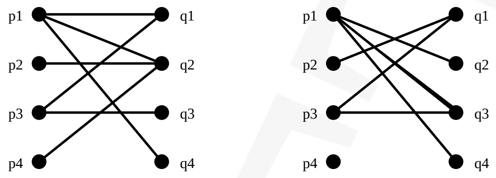

Fig. 2. Two bipartite graphs

Proposition 2.4 (König's minimax theorem). *If G is a bipartite graph then the largest number of edges in a matching of G equals the smallest number of vertices in a covering of G.*

Given a bipartite graph, we can compute the largest number of edges in a matching as follows:

```
i51 : matchingNumber = I -> min apply(primaryDecomposition I,
          J -> numgens J);
```

Let's use this function on the two bipartite graphs in Figure 2:

```
i52 : S = QQ[p_1 .. p_4, q_1 .. q_4];

i53 : I1 = monomialIdeal(p_1*q_1, p_1*q_2, p_1*q_4, p_2*q_2, p_3*q_1, ···

o53 = monomialIdeal (p q , p q , p q , p q , p q , p q , p q )
                      1 1   3 1   1 2   2 2   4 2   3 3   1 4

o53 : MonomialIdeal of S

i54 : m1 = matchingNumber I1

o54 = 3

i55 : select(primaryDecomposition I1, J -> numgens J === m1)

o55 = {monomialIdeal (p , p , q )}
                       1   3   2

o55 : List

i56 : I2 = monomialIdeal(p_1*q_2, p_1*q_3, p_1*q_4, p_2*q_1, p_3*q_1, ···

o56 = monomialIdeal (p q , p q , p q , p q , p q , p q )
                      2 1   3 1   1 2   1 3   3 3   1 4

o56 : MonomialIdeal of S

i57 : m2 = matchingNumber I2

o57 = 3

i58 : scan(select(primaryDecomposition I2, J -> numgens J === m2),
          L -> << endl << L << endl);

monomialIdeal (p , p , p )
                1   2   3

monomialIdeal (p , p , q )
                1   3   1

monomialIdeal (p , q , q )
                1   1   3
```

For more on the relationships between a graph and its associated ideal, see Villarreal [16].

3 Standard Pairs

In this section, we examine a combinatorial object associated to a monomial ideal. In particular, we present two algorithms for computing the standard pairs of a monomial ideal from its minimal generators. Before giving the definition of a standard pair, we consider an example.

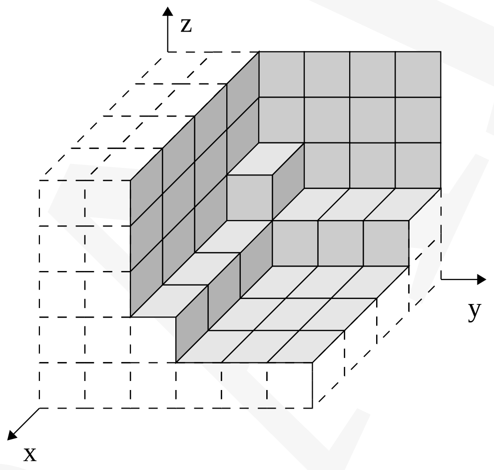

Fig. 3. Staircase diagram for $I = \langle xy^3z, xy^2z^2, y^3z^2, y^2z^3\rangle$

Example 3.1. Let $I = \langle xy^3z, xy^2z^2, y^3z^2, y^2z^3\rangle$ in $\mathbb{Q}[x,y,z]$. We identify the monomials in $\mathbb{Q}[x,y,z]$ with the lattice points in $\mathbb{N}^3$; see Figure 3. The standard monomials of I, those monomials which are not in I, can be enumerated as follows: (1) monomials corresponding to lattice points in the xy-plane, (2) monomials corresponding to lattice points in the xz-plane, (3) monomials corresponding to lattice points in the plane parallel to the xz-plane containing $(0,1,0)$, (4) monomials corresponding to lattice points on the line parallel to the y-axis containing $(0,0,1)$, (5) monomials corresponding to lattices point on the line parallel to the x-axis containing $(0,1,1)$, and (6) the monomial y^2z^2.

Following Sturmfels, Trung and Vogel [15], we make the following definitions. Given a monomial x^α and a subset F of the variables $\{x_1,\ldots,x_n\}$, we index the set of monomials of the form $x^\alpha \cdot x^\beta$ where $\operatorname{supp} x^\beta \subseteq F$ by the pair

(x^α, F). A standard pair of a monomial ideal I is a pair (x^α, F) satisfying the following three conditions:

(1) $\operatorname{supp} x^\alpha \cap F = \emptyset$,
(2) all of the monomials represented by this pair are standard, and
(3) $(x^\alpha, F) \not\subseteq (x^\beta, G)$ for any other pair (x^β, G) satisfying the first two conditions.

Hence, the six standard pairs in Example 3.1 are $(1, \{x, y\})$, $(1, \{x, z\})$, $(y, \{x, z\})$, $(z, \{y\})$, $(yz, \{x\})$, $(y^2z^2, \emptyset)$.

Observe that the set of standard pairs of I gives an irreducible decomposition of I:

$$I = \bigcap \langle x_i^{\alpha_i+1} \,:\, x_i \notin F \rangle \,,$$

where the intersection is over all standard pairs (x^α, F). Moreover, the prime ideal $P_F := \langle x_i \,:\, x_i \notin F\rangle$ is an associated prime of I if and only if there exists a standard pair of the form $(\bullet, F)$; see Sturmfels, Trung and Vogel [15] for details.

Our first algorithm for computing the set of standard pairs is taken from Hoşten and Thomas [8]. The ideas behind it are as follows: given a witness $w_1 = x^\alpha$ for the associated prime $P_F := \langle x_i \,:\, x_i \notin F\rangle$, that is $I : x^\alpha = P_F$, set $w_2 = \prod_{x_i \in \operatorname{supp}(w_1) \cap F^c} x_i^{\alpha_i}$. It follows that (w_2, F) is a standard pair of I. Now, consider the standard pairs of the slightly larger ideal $I + \langle w_1 \rangle$. Clearly (w_2, F) is not a standard pair of this ideal because w_1 "destroys" it. However, this larger ideal might have standard pairs which cover standard monomials in (w_2, F) that are not in the pair (w_1, F). On the other hand, all other standard pairs are the same as the original ideal I. Thus, the problem of finding all standard pairs of I reduces to determining if a standard pair of $I + \langle w_1 \rangle$ is a standard pair for I. To decide if a pair (x^β, G) of $I + \langle w_1 \rangle$ is a standard pair of I, we first check that P_F is an associated prime of I. If this is true, we determine if (w_2, F) is covered by (x^β, G).

The *Macaulay 2* version of this algorithms takes the following form:

```
i59 : clearAll

i60 : load "monideal.m2"

i61 : supp = r -> select(gens ring r, e -> r % e == 0);

i62 : stdPairs = I -> (
           S := ring I;
           X := gens S;
           std := {};
           J := I;
           while J != S do (
                w1 := 1_S;
                F := X;
                K := J;
                while K != 0 do (
                     g1 := (ideal mingens ideal K)_0;
                     x := first supp g1;
                     w1 = w1 * g1 // x;
                     F = delete(x, F);
                     K = K : monomialIIdeal(g1 // x);
                     L := select(first entries gens K,
                          r -> not member(x, supp r));
```

```
                if #L > 0 then K = monomialIIdeal L
                else K = monomialIIdeal 0_S;);
           w2 := w1;
           scan(X, r -> if not member(r, supp w1) or member(r, F)
                then w2 = substitute(w2, {r => 1}));
           P := monomialIIdeal select(X, r -> not member(r, F));
           if (I:(I:P) == P) and (all(std, p ->
                     (w2 % (first p) != 0) or not
                     isSubset(supp(w2 // first p) | F, last p)))
           then std = std | {{w2, F}};
           J = J + monomialIIdeal(w1););
      std);
```

We can compute the standard pairs of Example 3.1 using this *Macaulay 2* function:

```
i63 : S = QQ[x,y,z];

i64 : I = monomialIIdeal(x*y^3*z, x*y^2*z^2, y^3*z^2, y^2*z^3);

o64 : MonomialIIdeal of S

i65 : scan(stdPairs(I), P -> << endl << P << endl);

{y, {x, z}}

{1, {x, z}}

  2 2
{y z , {}}

{z, {y}}

  2
{y z, {x}}

{1, {x, y}}
```

The second algorithm appears in section 3.2 of Saito, Sturmfels and Takayama [13]. The proposition below provides the main ingredient for this algorithm. Let I be a monomial ideal. For $F \subseteq \{x_1, \ldots, x_n\}$, write I_F for the monomial ideal in $\mathbb{Q}[x_i : \; x_i \notin F]$ obtained by replacing each $x_i \in F$ with 1 in every minimal generator of I.

Proposition 3.2. *For (x^α, F) to be a standard pair of I, it is necessary and sufficient that $(x^\alpha, \emptyset)$ be a standard pair of I_F.*

Now, the definition of a standard pair implies that $(x^\alpha, \emptyset)$ is a standard pair of I_F if and only if x^α is one of the finitely many monomials in $(I_F : P_F^\infty)$ but not contained in I_F, where $P_F = \langle x_i : x_i \notin F \rangle$. Since ideal quotients and saturations are implemented in *Macaulay 2*, this reduces the problem to finding a set D which contains F for every associated prime P_F of I. One approach is to simply compute the associated primes of I form a primary decomposition.

The method `standardPairs` uses this algorithm to determine the set of standard pairs for a monomial ideal.

```
i66 : code(standardPairs, MonomialIIdeal, List)
```

```
o66 = -- monideal.m2:293-321
      standardPairs(MonomialIdeal, List) := (I,D) -> (
           R := ring I;
           X := gens R;
           S := {};
           scan(D, L -> (
                     Y := X;
                     M := vars R;
                     scan(L, i -> (
                               Y = delete(i,Y);
                               M = substitute(M, {i => 1});
                               ));
                     kk:= coefficientRing(R);
                     -- using monoid to create ring to avoid
                     -- changing global ring.
                     A := kk (monoid [Y]);
                     phi := map(A, R, substitute(M, A));
                     J := ideal mingens ideal phi(gens I);
                     Jsat := saturate(J, ideal vars A);
                     if (Jsat != J) then (
                          B := flatten entries (
                               super basis trim (Jsat / J));
                          psi := map(R, A, matrix{Y});
                          scan(B, b -> (
                                    S = append(S, {psi(b), L});
                                    ));
                          );
                     ));
           S
           )

o66 : Net
```

As an example, we will compute the standard pairs of the permutahedron ideal. Let $S = \mathbb{Q}[x_1, \ldots, x_n]$ and let Σ_n be the symmetric group of order n. We write ρ for the vector $(1, 2, \ldots, n)$ and $\sigma(\rho)$ for the vector obtained by applying $\sigma \in \Sigma_n$ to the coordinates of ρ. The n-th permutahedron ideal is $\langle x^{\sigma(\rho)} : \sigma \in \Sigma_n \rangle$. We compute the number of standard pairs for $2 \le n \le 4$.

```
i67 : x = symbol x;

i68 : permutohedronIdeal = n -> (
           -- NOTE: changes global ring
           S = QQ[x_1 .. x_n];
           monomialIdeal terms det matrix table(n ,gens S,
                (i,r) -> r^(i+1)));

i69 : L = apply({2,3,4,5}, j -> standardPairs(permutohedronIdeal(j)));

i70 : apply(L, i -> #i)

o70 = {3, 10, 53, 446}

o70 : List
```

Example: integer programming problems

As an application of standard pairs, we show how to solve integer linear programming problems. Let A be a $d \times n$ matrix with entries in $\mathbb{N}$, $\omega \in \mathbb{R}^n$ and $b \in \mathbb{Z}^d$. We focus on the following optimization problem

$$\mathrm{IP}_{A,\omega}(b) : \text{minimize } \omega \cdot x \text{ subject to } Ax = b,\ x \in \mathbb{N}^n.$$

We view this integer linear program as a family depending on the vector b. The algorithm we present for solving $\mathrm{IP}_{A,\omega}(b)$ depends on the following proposition. The toric ideal associated to A is denoted I_A; see Chapter [?].

Proposition 3.3. (1) *A monomial x^α is a standard monomial of $\mathrm{in}_\omega(I_A)$ if and only if α is the optimal solution to the integer program $\mathrm{IP}_{A,\omega}(A \cdot \alpha)$.*
(2) *If $(\bullet, F)$ is a standard pair of $\mathrm{in}_\omega(I_A)$, then the columns of A corresponding to F are linearly independent.*

The first statement implies that the standard pairs of $\mathrm{in}_\omega(I_A)$ covers all optimal solutions to all integer programs in $\mathrm{IP}_{A,\omega}$. If β is the optimal solution to $\mathrm{IP}_{A,\omega}(b)$ covered by the standard pair (x^α, F), then the second statement guarantees there exists a unique $x \in \mathbb{N}^n$ such that $A \cdot x = b - A \cdot \alpha$. Therefore, $\beta = x + \alpha$. We point out that the complexity of this algorithm is dominated by determining the set of standard pairs of $\mathrm{in}_\omega(I_A)$ which depends only on A and ω. As a result, this method is particularly well suited to solving $\mathrm{IP}_{A,\omega}(b)$ as b varies.

Here is the corresponding *Macaulay 2* code.

```
i71 : clearAll;

i72 : load "monideal.m2";

i73 : load "toric.m2";

i74 : solveParametricInteger = (Std, A, b) -> (
           n := rank source A;
           x := {};
           P := first select(1, Std, P -> (
                     F := apply(last P, r -> index r);
                     a := transpose matrix exponents first P;
                     K := transpose syz (submatrix(A, F) | (A*a - b));
                     X := select(entries K, k -> abs last(k) === 1);
                     scan(X, k -> if (all(k, j -> j >= 0) or
                               all(k, j -> j <= 0))
                          then x = apply(n, j -> if member(j, F) then
                               last(k) * k_(position(F, i -> i === j))
                               else 0));
                     #x > 0));
           if #P > 0 then (matrix {x}) + (matrix exponents first P)
           else 0);
```

We illustrate this function with a couple of examples.

```
i75 : A = matrix{{1,1,1,1,1},{1,2,4,5,6}}

o75 = | 1 1 1 1 1 |
      | 1 2 4 5 6 |

                2        5
o75 : Matrix ZZ  <--- ZZ

i76 : Std = standardPairs(monomialIIdeal toricIdeal(A));

i77 : b = transpose matrix{{3,9}}

o77 = | 3 |
      | 9 |

                2        1
o77 : Matrix ZZ  <--- ZZ
```

```
i78 : solveParametricInteger(Std, A, b)

o78 = | 1 1 0 0 1 |

                  1       5
o78 : Matrix ZZ  <--- ZZ

i79 : b = transpose matrix{{5,16}}

o79 = | 5  |
      | 16 |

                  2       1
o79 : Matrix ZZ  <--- ZZ

i80 : solveParametricInteger(Std, A, b)

o80 = | 2 1 0 0 2 |

                  1       5
o80 : Matrix ZZ  <--- ZZ
```

4 Generic Initial Ideals

Gröbner basis calculations and initial ideals depend heavily on the choice of variables. By making a generic change of coordinates, we may eliminate this dependence. This procedure also endows the resulting monomial ideal with a rich combinatorial structure.

To describe this structure, we introduce the following notation. Let $S = \mathbb{Q}[x_0, \ldots, x_n]$. If $\mathsf{g} = [g_{i,j}] \in \mathrm{GL}_{n+1}(\mathbb{Q})$ and $f \in S$, then $\mathsf{g} \cdot f$ denotes the standard action of the general linear group on S: $x_i \mapsto \sum_{j=0}^{n} g_{i,j} x_j$. For an ideal $I \subseteq S$, $\mathsf{g} \cdot I = \{\mathsf{g} \cdot f | f \in I\}$. Let B denote the Borel subgroup of $\mathrm{GL}_{n+1}(\mathbb{Q})$ consisting of upper triangular matrices. A monomial ideal I is called Borel-fixed if it satisfies one of the following equivalent conditions:

(a) if r is a generator of I divisible by x_j then $\frac{r x_i}{x_j} \in I$ for all $i < j$;
(b) $\mathsf{g} \cdot I = I$ for every $\mathsf{g} \in B$;
(c) $\mathrm{in}(\mathsf{g} \cdot I) = I$ for every g is some open neighborhood of the identity in B.

In *Macaulay 2*, the function `isBorel` tests if a monomial ideal is Borel-fixed.

```
i81 : clearAll

i82 : load "monideal.m2"

i83 : S = QQ[a,b,c,d];

i84 : isBorel monomialIIdeal(a^2, a*b, b^2)

o84 = true

i85 : isBorel monomialIIdeal(a^2, b^2)

o85 = false
```

The function `borel` generates the smallest Borel-fixed ideal containing the given monomial ideal.

```
i86 : borel monomialIIdeal(b*c)

                      2         2
o86 = monomialIdeal (a , a*b, b , a*c, b*c)

o86 : MonomialIIdeal of S
```

```
i87 : borel monomialIIdeal(a,c^3)

                           3   2       2   3
o87 = monomialIdeal (a, b , b c, b*c , c )

o87 : MonomialIIdeal of S
```

The next theorem provides the main source of Borel-fixed ideals.

Theorem 4.1 (Galligo). *Fix a term order on S such that $x_0 > \ldots > x_n$. If I is a homogeneous ideal in S, then there is a Zariski open subset $U \subseteq \mathrm{GL}_{n+1}(\mathbb{Q})$ such that*

(1) *there is a monomial ideal $J \subseteq S$ such that $J = \mathrm{in}(\mathsf{g} \cdot I)$ for all $\mathsf{g} \in U$;*
(2) *the ideal J is Borel-fixed.*

The ideal J is called the generic initial ideal of I.

The following method allows one to compute generic initial ideals in *Macaulay 2*.

```
i88 : gin = method();

i89 : gin Ideal := I -> (
           S := ring I;
           StoS := map(S, S, random(S^{0}, S^{numgens S:-1}));
           monomialIIdeal StoS I);

i90 : gin MonomialIIdeal := I -> gin ideal I;
```

For example, we can determine the generic initial ideal of two generic forms of degree p and q in $\mathbb{Q}[a,b,c,d]$.

```
i91 : genericForms = (p,q) -> ideal(random(p,S), random(q,S));

i92 : gin genericForms(2,2)

                        2         3
o92 = monomialIdeal (a , a*b, b )

o92 : MonomialIIdeal of S

i93 : gin genericForms(2,3)

                        2      2   4
o93 = monomialIdeal (a , a*b , b )

o93 : MonomialIIdeal of S
```

Although the generic initial ideal is Borel-fixed, some non-generic initial ideals may also be Borel-Fixed.

```
i94 : J = ideal(a^2, a*b+b^2, a*c)

                2          2
o94 = ideal (a , a*b + b , a*c)

o94 : Ideal of S

i95 : ginJ = gin J

                        2         2      2
o95 = monomialIdeal (a , a*b, b , a*c )

o95 : MonomialIIdeal of S

i96 : inJ = monomialIIdeal J

                        2         3        2
o96 = monomialIdeal (a , a*b, b , a*c, b c)

o96 : MonomialIIdeal of S
```

```
i97 : isBorel inJ and isBorel ginJ

o97 = true
```

Finally, we show that the generic initial ideal does depend on the term order by computing lexicographic generic initial ideal for two generic forms of degree p and q in $\mathbb{Q}[a, b, c, d]$

```
i98 : S = QQ[a,b,c,d, MonomialOrder => Lex];

i99 : gin genericForms(2,2)

                     2         4     2
o99 = monomialIdeal (a , a*b, b , a*c )

o99 : MonomialIIdeal of S

i100 : gin genericForms(2,3)

                      2      2   6        2     6         2       4
o100 = monomialIdeal (a , a*b , b , a*b*c , a*c , a*b*c*d , a*b*d )

o100 : MonomialIIdeal of S
```

A more comprehensive treatment of generic initial ideals can be found in Green [4]. The properties of Borel-fixed ideals in characteristic p are discussed in Eisenbud [3].

Example: connectedness of the Hilbert scheme

Generic initial ideals are a powerful tool for studying the structure of the Hilbert scheme. For example, Reeves [12] proves that the radius of the incidence graph of irreducible components of the Hilbert scheme $\mathrm{Hilb}^{p(t)}(\mathbb{P}^n)$, parameterizing subschemes of $\mathbb{P}^n$ with Hilbert polynomial $p(t)$, is at most $\deg p(t) + 1$. Her proof can be divided into three major steps.

Step I: Connect an arbitrary ideal to a Borel-fixed ideal. Recall that passing to an initial ideal corresponds taking the limit in a flat family. Thus, Theorem 4.1 shows that generic initial ideals connect arbitrary ideals to Borel-fixed ideals.

Step II: Connect Borel-fixed ideals by projection. For a homogeneous ideal $I \subseteq S = \mathbb{Q}[x_0, \ldots, x_n]$, let $\pi(I)$ denote the ideal obtained by setting $x_n = 1$ and $x_{n-1} = 1$ in I. With this notation, we have

Theorem 4.2. *If J is a Borel-fixed ideal, then the set of Borel-fixed ideals I with Hilbert polynomial $p(t)$ and $\pi(I) = J$ consists of ideals defining subschemes in $\mathbb{P}^n$ which all lie on a single component of* $\mathrm{Hilb}^{p(t)}(\mathbb{P}^n)$.

This gives an easy method for partitioning Borel-fixed ideals into classes, each of which must lie in a single component.

Step III: Connect Borel-fixed ideals by distractions. Given a Borel-fixed ideal, we produce a new ideal via a two-step process called distraction. First, one converts the Borel-fixed ideal to a squarefree (polarized) ideal by adding a variable for each successive power of each original variable. One then pulls the result back to an ideal in the original variables by taking a linear section of the polarization. Hartshorne [5] shows that the distraction is connected to the original Borel-fixed ideal. Taking the lexicographic generic initial ideal of the distraction yields a new Borel-fixed ideal. The lexicographic component of the Hilbert scheme is the irreducible component containing the point corresponding to the unique saturated lexicographic ideal. Reeves proves that repeating this process one arrives at the lexicographic component after at most $\deg p(t) + 1$ iterations.

We can implement the projection and distraction operations in *Macaulay 2* as follows:

```
i101 : projection = I -> (
            S := ring I;
            n := numgens S;
            X := gens S;
            monomialIIdeal mingens substitute(ideal I,
               {X#(n-2) => 1, X#(n-1) => 1}));

i102 : polarization = I -> (
            n := numgens ring I;
            u := apply(first entries gens I, r -> first exponents r);
            I.lcm = max \ transpose u;
            Z := flatten apply(n, i -> apply(I.lcm#i, j -> z_{i,j}));
            R := QQ(monoid[Z]);
            Z = gens R;
            p := apply(n, i -> sum((I.lcm)_{0..i-1}));
            monomialIIdeal apply(u, e -> product apply(n, i ->
                      product(toList(0..e#i-1), j -> Z#(p#i+j)))));

i103 : distraction = I -> (
            S := ring I;
            n := numgens S;
            X := gens S;
            J := polarization I;
            W := flatten apply(n, i -> flatten apply(I.lcm#i,
               j -> X#i));
            section := map(S, ring J, apply(W, r -> r -
               random(500)*X#(n-2) - random(500)*X#(n-1)));
            section ideal J);
```

For example, we have

```
i104 : S = QQ[x_0 .. x_4, MonomialOrder => GLex];

i105 : I = monomialIIdeal(x_0^2, x_0*x_1^2*x_3, x_1^3*x_4)

                        2     2     3
o105 = monomialIdeal (x , x x x , x x )
                        0   0 1 3   1 4

o105 : MonomialIIdeal of S

i106 : projection I

                        2     2   3
o106 = monomialIdeal (x , x x , x )
                        0   0 1   1

o106 : MonomialIIdeal of S

i107 : polarization I

o107 = monomialIdeal (z       z       , z       z       z       z       , z   ...
                        {0, 0} {0, 1}   {0, 0} {1, 0} {1, 1} {3, 0}    { ...

o107 : MonomialIIdeal of QQ [z       , z       , z       , z       , z     ...
                               {0, 0}   {0, 1}   {1, 0}   {1, 1}   {1,   ...
```

```
i108 : distraction I

               2                           2                     2    ···
o108 = ideal (x  - 398x x  - 584x x  + 36001x  + 92816x x  + 47239x , ···
               0       0 3      0 4         3         3 4         4   ···

o108 : Ideal of S
```

To illustrate Reeves' method, we show that the $\mathrm{Hilb}^{4t+1}(\mathbb{P}^4)$ has radius 2. Note that the rational quartic curve in $\mathbb{P}^4$ has Hilbert polynomial $4t+1$.

```
i109 : m =  matrix table({0,1,2}, {0,1,2}, (i,j) -> (gens S)#(i+j))

o109 = {0} | x_0 x_1 x_2 |
       {0} | x_1 x_2 x_3 |
       {0} | x_2 x_3 x_4 |

               3      3
o109 : Matrix S  <--- S

i110 : rationalQuartic = minors(2, m);

o110 : Ideal of S

i111 : H = hilbertPolynomial(S/rationalQuartic);

i112 : hilbertPolynomial(S/rationalQuartic, Projective => false)

o112 = 4$i + 1

o112 : QQ [$i]
```

There are 12 Borel-fixed ideals with Hilbert polynomial $4t+1$.

```
i113 : L = {monomialIIdeal(x_0^2, x_0*x_1, x_0*x_2, x_1^2, x_1*x_2, x_ ···

i114 : scan(#L, i -> << endl << i+1 << " : " << L#i << endl);

                        2         2                 2
1 : monomialIdeal (x , x x , x , x x , x x , x )
                    0   0 1   1   0 2   1 2   2

                        2         2                 3
2 : monomialIdeal (x , x x , x , x x , x x , x , x x )
                    0   0 1   1   0 2   1 2   2   0 3

                            2     2   3
3 : monomialIdeal (x , x , x x , x , x x x )
                    0   1   1 2   2   1 2 3

                            2         4   3
4 : monomialIdeal (x , x , x x , x , x x )
                    0   1   1 2   2   2 3

                                5   4 3
5 : monomialIdeal (x , x , x , x x )
                    0   1   2   2 3

                            2         5         4 2
6 : monomialIdeal (x , x , x x , x , x x , x x )
                    0   1   1 2   2   1 3   2 3

                        2         2                 5                 4
7 : monomialIdeal (x , x x , x , x x , x x , x , x x , x x , x x )
                    0   0 1   1   0 2   1 2   2   0 3   1 3   2 3

                            2         5   4         2
8 : monomialIdeal (x , x , x x , x , x x , x x )
                    0   1   1 2   2   2 3   1 3

                        2         2                 4             2
9 : monomialIdeal (x , x x , x , x x , x x , x , x x , x x )
                    0   0 1   1   0 2   1 2   2   0 3   1 3
```

```
                        2     2   4             2
10 : monomialIdeal (x , x , x x , x , x x x , x x )
                    0   1   1 2   2   1 2 3   1 3

                        2         4     3
11 : monomialIdeal (x , x , x x , x , x x )
                    0   1   1 2   2   1 3

                            6    5     4 2
12 : monomialIdeal (x , x , x , x x , x x )
                    0   1   2   2 3   2 3

i115 : all(L, I -> isBorel I and hilbertPolynomial(S/I) == H)

o115 = true
```

The projection operation partitions the list L into 3 classes:

```
i116 : class1 = projection L#0

                      2         2               2
o116 = monomialIdeal (x , x x , x , x x , x x , x )
                      0   0 1   1   0 2   1 2   2

o116 : MonomialIIdeal of S

i117 : class2 = projection L#1

                          2         3
o117 = monomialIdeal (x , x , x x , x )
                      0   1   1 2   2

o117 : MonomialIIdeal of S

i118 : class3 = projection L#4

                              4
o118 = monomialIdeal (x , x , x )
                      0   1   2

o118 : MonomialIIdeal of S

i119 : all(1..3, i -> projection L#i == class2)

o119 = true

i120 : all(4..11, i -> projection L#i == class3)

o120 = true
```

Finally, we use the distraction to connect the the different classes.

```
i121 : all(L, I -> I == monomialIIdeal distraction I)

o121 = true

i122 : all(0..3, i -> projection gin distraction L#i == class3)

o122 = true
```

5 The Chain Property

The set of associated primes for an ideal has a natural partial order given by inclusion. Since a monomial ideal is prime if and only if it is generated by a subset of the variables, the set of associated primes for a monomial ideal is contained in the poset of all subset of the variables. We say that a monomial ideal I has the chain property if the following condition holds: for any embedded prime $P_F := \langle x_i : x_i \notin F \rangle$ of I there exists an associated prime $P_G \subseteq P_F$ such that $|G| = |F| - 1$. In other words, there exists a saturated chain from every embedded prime to some minimal prime. Miller, Sturmfels and Yanagawa [10] show that a "generic" monomial ideals have the chain property. We are interested in the question: Which monomial ideals have the chain property? To determine if a monomial ideal has the chain property, we use the following *Macaulay 2* code.

```
i123 : clearAll;

i124 : load "monideal.m2"

i125 : hasChainProperty = I -> (
           L := ass(I);
           radI := radical(I);
           all(L, P -> radI : (radI : P) == P or (
                   gensP := first entries gens P;
                   all(gensP, r -> (
                           Q := monomialIIdeal(delete(r, gensP));
                           I : (I : Q) == Q)))));
```

Examples and counterexamples

The chain property was first studied by Hoşten and Thomas [7] for the family of monomial ideals arising as initial ideals of toric ideals; see Chapter [?] for more information about toric ideals. Using a polyhedral characterization for the standard pairs, Hoşten and Thomas prove that initial ideals of toric ideals have the chain property. For example, the toric ideal associated with the matrix

$$A = \begin{bmatrix} 1&1&1&1&1&1&1 \\ 2&0&0&0&1&0&0 \\ 0&2&0&0&0&1&0 \\ 2&2&0&2&1&1&1 \end{bmatrix}$$

is the ideal $I = \langle xz - q^2, yz - r^2, pz - s^2 \rangle$. We verify that its initial ideal has the chain property as follows:

```
i126 : S = QQ[x,y,z,p,q,r,s];

i127 : I = ideal(x*z - q^2, y*z -r^2, p*z-s^2)

                      2          2          2
o127 = ideal (x*z - q , y*z - r , z*p - s )

o127 : Ideal of S

i128 : hasChainProperty(monomialIIdeal I)

o128 = true
```

Since toric ideals are prime, it is natural to ask if the initial ideal of any prime ideal has the chain property. The next example show this is not the case. By making a linear change of coordinates by $p \mapsto z-p$, we obtain a new prime ideal $J = \langle xz - q^2, yz - r^2, z^2 - zp - s^2 \rangle$. Using a reverse lexicographic term order where $x \succ y \succ z \succ p \succ q \succ r \succ s$, we show that the initial ideal of J with respect to this term order does not have the chain property:

```
i129 : J = ideal(x*z - q^2, y*z - r^2, z^2 - z*p - s^2);

o129 : Ideal of S

i130 : hasChainProperty(monomialIdeal J)

o130 = false
```

A close relative of initial ideals of toric ideals is the family of A-graded monomial ideals; again see Chapter [?]. When A is corank two, Peeva and Gasharov [11] establish that each A-graded monomial ideal is an initial ideal of the corresponding toric ideal. In particular, they have the chain property. On the other hand, Altmann [1] showed that when A of rank two then A-graded monomial ideals have the chain property. However, Altmann also provides though a counterexample when A has rank 3. We can verify his example in *Macaulay 2* as follows:

```
i131 : S = QQ[a_1..a_3, b_1..b_3, c_1..c_3, d_1..d_7];

i132 : I = monomialIdeal(b_1*d_1, b_1*d_2, b_1*d_3, b_1*d_4, b_1*d_5,
             b_1*d_6, b_1*d_7, b_2*d_1, b_2*d_2, b_2*d_3, b_2*d_4,
             b_2*d_5, b_2*d_6, b_2*d_7, b_3*d_1, b_3*d_2, b_3*d_3,
             b_3*d_4, b_3*d_5, b_3*d_6, b_3*d_7, c_1*d_1, c_1*d_2,
             c_1*d_3, c_1*d_4, c_1*d_5, c_1*d_6, c_1*d_7, c_2*d_1,
             c_2*d_2, c_2*d_3, c_2*d_4, c_2*d_5, c_2*d_6, c_2*d_7,
             c_3*d_1, c_3*d_2, c_3*d_3, c_3*d_4, c_3*d_5, c_3*d_6,
             c_3*d_7, c_1^2, c_1*c_2, c_1*c_3, c_2^2, c_2*c_3, c_3^2,
             d_1^2, d_1*d_2, d_1*d_3, d_1*d_4, d_1*d_5, d_1*d_6,
             d_1*d_7, d_2^2, d_2*d_3, d_2*d_4, d_2*d_5, d_2*d_6,
             d_2*d_7, d_3^2, d_3*d_4, d_3*d_5, d_3*d_6, d_3*d_7, d_4^2,
             d_4*d_5, d_4*d_6, d_4*d_7, d_5^2, d_5*d_6, d_5*d_7, d_6^2,
             d_6*d_7, d_7^2, b_1^2, b_2^2, b_3^2, b_1*c_2, b_2*c_3,
             b_3*c_1, b_3*b_1*c_3, b_1*b_2*c_1, b_2*b_3*c_2,
             a_3*b_1*c_1, a_1*b_2*c_2, a_2*b_3*c_3, a_3*b_2*c_1,
             a_1*b_3*c_2, a_2*b_1*c_3, a_3*b_1*b_2, a_1*b_2*b_3,
             a_2*b_3*b_1, a_3*a_2*b_1, a_1*a_3*b_2, a_2*a_1*b_3,
             a_3*a_2*c_1, a_1*a_3*c_2, a_2*a_1*c_3);

o132 : MonomialIdeal of S

i133 : hasChainProperty(I)

o133 = false
```

For the record, `I` is an A-graded ideal for the matrix

$$\begin{bmatrix} 2 & 0 & 0 & 1 & 0 & 0 & 2 & 1 & 1 & 3 & 2 & 2 & 2 & 3 & 3 & 3 \\ 0 & 2 & 0 & 0 & 1 & 0 & 1 & 2 & 1 & 2 & 3 & 2 & 3 & 2 & 3 & 3 \\ 0 & 0 & 2 & 0 & 0 & 1 & 1 & 1 & 2 & 2 & 2 & 3 & 3 & 3 & 2 & 3 \end{bmatrix}.$$

Lastly, we consider a different family of monomial ideals arising from toric ideals. The vertex ideal V_A of A is defined as

$$\bigcap_{\prec} in_{\prec}(I_A)$$

which is the intersection of all monomial initial ideals of the toric ideal of A. Although there are (in general) infinitely many distinct term orders on a polynomial ring, an ideal has only finitely many initial ideals; see Sturmfels [14]. In particular, the above intersection is finite. Vertex ideals were introduced and studied by Hoşten and Maclagan [6]. However, the following question is still open: Does the vertex ideal V_A have the chain property? If A is a unimodular matrix, then no initial ideal of I_A has embedded primes. Hence, V_A has no embedded primes and the chain property trivially holds. If the corank of A is 2, then although individual initial ideals can have embedded primes, V_A does not have embedded primes and thus the chain property holds.

References

1. Klaus Altmann. The chain property for the associated primes of a-graded ideals. math.AG/0004142.
2. Winfried Bruns and Jürgen Herzog. *Cohen-Macaulay rings.* Cambridge University Press, Cambridge, 1993.
3. David Eisenbud. *Commutative algebra.* Springer-Verlag, New York, 1995. With a view toward algebraic geometry.
4. Mark L. Green. Generic initial ideals. In *Six lectures on commutative algebra (Bellaterra, 1996)*, pages 119–186. Birkhäuser, Basel, 1998.
5. Robin Hartshorne. Connectedness of the Hilbert scheme. *Inst. Hautes Études Sci. Publ. Math.*, 29:5–48, 1966.
6. Serkan Hoşten and Diane Maclagan. The vertex ideal of a lattice. 20 pages, (2000).
7. Serkan Hoşten and Rekha R. Thomas. The associated primes of initial ideals of lattice ideals. *Math. Res. Lett.*, 6(1):83–97, 1999.
8. Serkan Hoşten and Rekha R. Thomas. Standard pairs and group relaxations in integer programming. *J. Pure Appl. Algebra*, 139(1-3):133–157, 1999. Effective methods in algebraic geometry (Saint-Malo, 1998).
9. Ezra Miller. Alexander duality for monomial ideals and their resolutions. math.AG/9812095.
10. Ezra Miller, Bernd Sturmfels, and Kohji Yanagawa. Generic and cogeneric monomial ideals. *J. Symbolic Comput.*, 29(4-5):691–708, 2000. Symbolic computation in algebra, analysis, and geometry (Berkeley, CA, 1998).
11. Irena Peeva and Gasharov Vesselin. Deformations of codimension 2 toric varieties. *Compositio Mathematica.* To appear.
12. Alyson A. Reeves. The radius of the Hilbert scheme. *J. Algebraic Geom.*, 4(4):639–657, 1995.
13. Mutsumi Saito, Bernd Sturmfels, and Nobuki Takayama. *Gröbner deformations of hypergeometric differential equations.* Springer-Verlag, Berlin, 2000.
14. Bernd Sturmfels. *Gröbner bases and convex polytopes.* American Mathematical Society, Providence, RI, 1996.
15. Bernd Sturmfels, Ngô Viêt Trung, and Wolfgang Vogel. Bounds on degrees of projective schemes. *Math. Ann.*, 302(3):417–432, 1995.
16. Rafael H. Villarreal. Cohen-Macaulay graphs. *Manuscripta Math.*, 66(3):277–293, 1990.