Journal of Symbolic Computation 45 (2010) 953–964

Contents lists available at ScienceDirect

Journal of Symbolic Computation

journal homepage: www.elsevier.com/locate/jsc

Computing the support of monomial iterated mapping cones

Eduardo Sáenz-de-Cabezón [1]

Universidad de La Rioja, Mathematics and Computer Science, Edificio Vives, c/ Luis de Ulloa s/n, Room 213, 26005Logroño, La Rioja, Spain

ARTICLE INFO

Article history:
Received 22 October 2009
Accepted 2 June 2010
Available online 16 June 2010

Keywords:
Monomial ideals
Betti numbers
Iterated mapping cones
Multigraded Hilbert series
Mayer–Vietoris trees

ABSTRACT

In this paper we compute and manipulate the support of monomial resolutions based on iterated mapping cones. We derive in this way algorithms to obtain homological and numerical invariants of monomial ideals without actually computing their resolution. Our computations include Betti diagrams, Hilbert series and irreducible decompositions. The algorithms derived by the method presented in the paper are efficient in practice as shown by experiments.

1. Introduction

Iterated mapping cones are a standard "divide and conquer" strategy to compute free resolutions of ideals in a polynomial ring. In particular, some well known monomial resolutions arise as iterated mapping cones (e.g. Taylor (1966), Lyubeznik (1998), Eliahou and Kervaire (1990)) and some families of monomial ideals are minimally resolved by iterated mapping cones (Charalambous and Evans, 1995; Herzog and Takayama, 2002; Francisco, 2005). The use of this kind of strategy in this context started with Bayer and Stillman (1992) and has been frequently used thereafter.

In this paper we focus on monomial ideals and our goal is to compute the support of a resolution based on iterated mapping cones in a combinatorial way, without computing the resolution itself. On this support we perform reductions eliminating some elements in a way that the remaining elements still support a resolution of the underlying ideal. We also analyze, by these methods, the computed support to obtain numerical invariants of the ideal such as Betti numbers, multigraded Hilbert series, etc.

E-mail address: eduardo.saenz-de-cabezon@unirioja.es.

[1] Tel.: +34 941 299458; fax: +34 941 299460.

0747-7171/$ – see front matter © 2010 Elsevier Ltd. All rights reserved.
doi:10.1016/j.jsc.2010.06.001

An advantage of this method is that it produces simple and efficient algorithms to perform computations on monomial ideals, avoiding the computation of the full minimal free resolution, which is very hard. Also, analyzing the support of a mapping cone resolution provides tools to analyze other properties of the ideal (Saenz-de-Cabezón, 2009).

2. Iterated mapping cones and their support. Mayer–Vietoris trees

The cone of a map, or mapping cone, is a standard tool coming from topology. Among other uses, it provides a recursive way to compute free resolutions of ideals in a polynomial ring.

Let $R = \mathbf{k}[x_1, \ldots, x_n]$ be the polynomial ring in n indeterminates with $\mathbf{k}$ a field. Let $I = \langle f_1, \ldots, f_r \rangle \subseteq R$ be an ideal. Let $I_i = \langle f_1, \ldots, f_i \rangle$ be the subideal of I generated by the first i generators of I. There is a short exact sequence

$$0 \to R/(I_{i-1} : f_i) \xrightarrow{\phi} R/I_{i-1} \xrightarrow{\mathrm{j}} R/I_i \to 0 \tag{1}$$

for all $i \leq r$. Assume that free resolutions $\tilde{\mathcal{P}}$ and $\mathcal{P}'$ are known for $R/(I_{i-1} : f_i)$ and R/I_{i-1} respectively, then a resolution of R/I_i is obtained as the mapping cone of the chain complex morphism that lifts ϕ to a map from $\tilde{\mathcal{P}}$ to $\mathcal{P}'$. The procedure works with every short exact sequence; the following one, equivalent to (1), is particularly convenient in our context:

$$0 \to \tilde{I}_i \xrightarrow{\phi} I_{i-1} \oplus \langle f_i \rangle \xrightarrow{\mathrm{j}} I_i \to 0 \tag{2}$$

where $\tilde{I}_i = I_{i-1} \cap \langle f_i \rangle$. It is called a Mayer–Vietoris sequence.

We will work with monomial ideals in R. We use the standard $\mathbb{N}^n$-multigrading $md(x_i) = (0, \ldots, \overset{i}{1}, \ldots, 0)$ in R and denote by $R(-\alpha)$ the free R-module generated by one element in multidegree α. A multigraded resolution of a monomial ideal I will be denoted by $\mathcal{P} : \cdots \mathcal{P}_i \xrightarrow{\delta_i} \mathcal{P}_{i-1} \to \cdots \to \mathcal{P}_1 \xrightarrow{\delta_1} \mathcal{P}_0 \to 0$, where the free modules $\mathcal{P}_i$ are $\mathbb{N}^n$-graded and each homomorphism δ_i is multidegree preserving. In the case $\mathcal{P}$ is a minimal resolution, if $\mathcal{P}_i = \bigoplus_{\alpha \in \mathbb{N}^n} R^{\beta_{i,\alpha}}(-\alpha)$ then we say that the (i, α)-th Betti number of I is the nonzero integer $\beta_{i,\alpha}$. These are called the *multigraded Betti numbers* of I. We say that $\alpha \in \mathbb{N}^n$ is a *Betti multidegree* of I if $\beta_{i,\alpha}(I) \neq 0$ for some $i \in \mathbb{N}$. When we speak of the *collection of Betti multidegrees* of I we take into account multiplicities.

In this paper we use sequence (2) to generate our mapping cone resolutions. For a monomial ideal I consider its minimal generating set as an ordered set $\{m_1, \ldots, m_r\}$. For each $1 \leq s \leq r$ we denote $I_s := \langle m_1, \ldots m_s \rangle$ and $\tilde{I}_s := I_{s-1} \cap \langle m_s \rangle = \langle m_{1,s}, \ldots, m_{s-1,s} \rangle$, where $m_{i,j}$ denotes $\mathrm{lcm}(m_i, m_j)$.

The ideals involved when using recursively sequence (2) can be displayed as a tree. The root of this tree is I and every node J has two *children*: $\tilde{J}$ and J'. If J is generated by r monomials, $\tilde{J}$ denotes $\tilde{J}_r$ and J' denotes J_{r-1}. This is what we call a *Mayer–Vietoris tree* of the monomial ideal I, denoted $\mathrm{MVT}(I)$. When we speak of the nodes in such a tree we refer to the ideal in the node. Each node in a Mayer–Vietoris tree has a *position*: the root has position 1 and the left and right children of the node in position p have positions $2p$ and $2p + 1$ respectively. The node of $\mathrm{MVT}(I)$ in position p is denoted $\mathrm{MVT}_p(I)$. We call *relevant nodes* those in an even position or in position 1. We also assign a *dimension* to each node: the root has dimension 0 and the left and right children of any node of dimension d have dimension $d + 1$ and d respectively. Note that the dimension of a node is the number of zeros of the binary expression of the position of that node. Clearly, the relevant nodes in the Mayer–Vietoris tree support the corresponding mapping cone resolution in the sense that the module $\mathcal{P}_i$ is the direct sum of copies of R generated in the multidegrees given by the generators of the relevant nodes of dimension i in the tree. See Fig. 1.

Remark 2.1. As the reader may notice, the choice of the last generator of the ideal I to be the one which defines the sequence is just a matter of convenience in notation. The important fact is that we select *some* particular generator to define the sequence. This generator is called the *pivot monomial* and is used to generate the sequence. Several selection strategies can be applied to select the pivot monomial, and they can be changed during the process. Among these strategies, the most relevant are those called *coherent*:

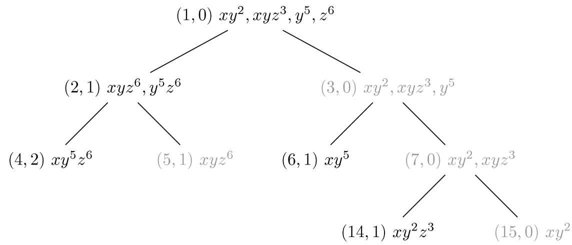


Fig. 1. A Mayer–Vietoris tree of $\langle xy^2, xyz^3, y^5, z^6\rangle$.

Definition 2.2. Let I be a monomial ideal. Let $J = \langle m_1, \ldots, m_r\rangle$ a node in $\mathrm{MVT}(I)$. Let m_i be the pivot monomial in J, then clearly $\tilde{J}$ is generated by a subset of $\{m_{ji} | 1 \leq j \leq r, j \neq i\}$. A strategy for the construction of $\mathrm{MVT}(I)$ is said to be *coherent* if whenever m_{ij} is the pivot monomial for $\tilde{J}$ then m_j is the pivot monomial for J' for every node $J \in \mathrm{MVT}(I)$.

In this paper, unless otherwise specified, we assume the pivot monomial is the first generator with respect to *lex* ordering, which is a *coherent* strategy, provided the generators of the nodes are numbered in *lex* ordering. This will not be relevant in this and the next section, but it is the strategy implemented in the algorithms described in Section 4. Observe that different ways of pivot selection might lead to very different trees.

The following propositions are a direct consequence of the fact that the relevant nodes of a Mayer–Vietoris tree support a resolution of the corresponding ideal. They are proved in Saenz-de-Cabezón (2008) together with other features of Mayer–Vietoris trees.

Proposition 2.3. *If $\beta_{i,\alpha}(I) \neq 0$ for some i, then x^α is a generator of some relevant node J in any Mayer–Vietoris tree* $\mathrm{MVT}(I)$.

Proposition 2.4. *If x^α appears only once as a generator of a relevant node J in* $\mathrm{MVT}(I)$ *then there exists exactly one $i \in \mathbb{N}$ such that $\beta_{i,\alpha}(I) = 1$ and $\beta_{j,\alpha}(I) = 0$ for all $i \neq j$.*

The homological degree i to which relevant multidegrees contribute is the dimension of the node of the Mayer–Vietoris tree in which it appears.

Example 2.5. Consider the ideal $I = \langle xy^2, xyz^3, y^5, z^6\rangle \subseteq \mathbf{k}[x, y, z]$. A Mayer–Vietoris tree of this ideal is shown in Fig. 1. Every node is given by a triple `(position, dimension) ideal` and the relevant nodes are the ones in strong black color. Observe that this tree has no repeated multidegree in the relevant nodes, therefore the multigraded Betti numbers of I are read from the tree. In this case we have $\beta_0(I) = 4$, $\beta_1(I) = 4$ and $\beta_2(I) = 1$. The Betti multidegrees are those of the generators of the relevant nodes in the tree.

3. Analysis of Mayer–Vietoris trees

We can use the properties of the nodes in a Mayer–Vietoris tree to discover properties of the mapping cone resolution supported on it, which we call *Mayer–Vietoris resolution*. Such study has led to characterizations of several families of ideals for which Mayer–Vietoris resolutions are minimal and, in some cases, to derive formulas for the (multi)graded Betti numbers of several families of ideals like Ferrers, Valla, k-out-of-n, consecutive k-out-of-n and others (Saenz-de-Cabezón, 2008; Saenz-de-Cabezón and Wynn, 2009; Saenz-de-Cabezón, 2009), see Example 3.5 below.

The modules in the cone of a map $\phi : A \to B$ are given by $\mathrm{Cone}(\phi)_i = \mathrm{B_i} \oplus \mathrm{A_{i-1}}$ and the differentials by

$$d_i^{\mathrm{Cone}(\phi)} = \begin{pmatrix} d_i^B & \phi_i \\ 0 & -d_{i-1}^A \end{pmatrix}.$$

Thus, when constructing a resolution as a mapping cone, if we keep minimality at each step, we know that the only possible part of the matrix which can be reduced is that corresponding to ϕ_i. Therefore, the search for pairs of non-minimal elements, also called reduction pairs, in the Mayer–Vietoris resolution (i.e. scalars in the matrices of its differentials) can be restricted to a search of scalars in the matrices of the morphisms ϕ used in the recursive process. As an abuse of notation, when we speak in the following paragraphs of *reduction pairs* we refer either to reduction pairs in the Mayer–Vietoris resolution or to their counterparts in ϕ.

We first give two obvious necessary conditions on the generators:

C1. Since both the Mayer–Vietoris resolution and the morphisms ϕ are multigraded, generators forming a reduction pair must have the same multidegree.
C2. Since reduction pairs correspond to scalars in ϕ there must be one step in the recursion process corresponding to some ideal J such that one of the generators in the pair is a generator of $\tilde{\mathcal{P}}_i$, the i-th module of the corresponding resolution of $\tilde{J}$ and the other is a generator of $\mathcal{P}'_i$, the module at the same homological degree in the resolution of J'.

For pairs of multidegrees satisfying the above conditions we must find ways to detect whether they actually form a reduction pair without computing the corresponding matrices. The following result is useful in this respect. We assume that MVT(I) is constructed using a coherent strategy to ensure that the lifting of the inclusion ϕ is again an inclusion. Note that if the strategy is not coherent we must first perform a minimization of each resolution in the process.

Lemma 3.1. *Let $\mu \in \mathbb{N}^n$ be a multidegree such that $\tilde{\mathcal{P}}_i = R(-\mu)^k \oplus \sum_{\alpha\in\mathcal{A}} R(-\alpha)$ and $\mathcal{P}'_i = R(-\mu)^l \oplus \sum_{\alpha\in\mathcal{B}} R(-\alpha)$ where $\mathcal{A}$ and $\mathcal{B}$ are collections of multidegrees.*

If there is no divisor of μ in $\mathcal{B}$ then the generators corresponding to the pieces of the multidegree μ in $\tilde{\mathcal{P}}_i$ and $\mathcal{P}'_i$ form k reduction pairs.

Proof. Consider the first k columns of the matrix corresponding to the inclusion ϕ at level i. If there is no divisor of μ in $\mathcal{B}$ then the last entries of these columns are zeros. Only the first l entries might be nonzero. Since ϕ is injective, the first column cannot be formed just by zeros, and then its first element is nonzero (we re-arrange rows if necessary). Since ϕ is multigraded, it must be a scalar and therefore the corresponding generators form a reduction pair. After deleting these generators, we are in the same situation which we can repeat k times. □

The multidegrees satisfying the above conditions and Lemma 3.1 can be found in the Mayer–Vietoris tree without the differentials of the resolution. From condition C1 we obtain that only those multidegrees that are repeated as generators of relevant nodes can be part of a reduction pair. For condition C2 we need some terminology:

Definition 3.2. Let b_1 and b_2 be two distinct binary numbers. We can say that b_1 and b_2 have the following form: $b_1 = u_1 \dots u_k 0 \tilde{u}_1 \dots \tilde{u}_{l_1}, b_2 = u_1 \dots u_k 1 u'_1 \dots u'_{l_2}$. We say that b_1 and b_2 are *compatible* if the number of zeros in $\tilde{u}_1 \dots \tilde{u}_{l_1}$ and $u'_1 \dots u'_{l_2}$ are equal. Observe that the total number of zeros of two compatible binary numbers differs by one.

We say that two positive integers $n_1, n_2 \in \mathbb{N}$ are *compatible* if their corresponding binary expressions are compatible. We say that two sets $\mathcal{A}, \mathcal{B} \subset \mathbb{N}$ are compatible if every pair (a, b), $a \in \mathcal{A}, b \in \mathcal{B}$ is compatible.

Nodes in compatible positions give those pairs that satisfy condition C2 above:

Proposition 3.3. *Let I be a monomial ideal, $\mu \in \mathbb{N}^n$ a multidegree appearing in the relevant nodes of positions p_1 and p_2 of a Mayer–Vietoris tree of I. Let e_1 and e_2 be their corresponding generators in the associated resolution of I.*

If e_1 and e_2 are a reduction pair then p_1 and p_2 are compatible.

Proof. Let b_1 and b_2 be the binary expressions of p_1 and p_2, given in the form shown in Definition 3.2. It is easy to see from the construction of Mayer–Vietoris trees that the number of zeros of the binary expression of the position of a node gives its dimension. It is also easy to see that the decimal expression the common part of b_1 and b_2, $d = u_1 \dots u_{k|10}$ gives the position of the nearest common ancestor J of the nodes with positions p_1 and p_2 in $\mathrm{MVT}(I)$.

Since e_1 and e_2 are a reduction pair in the resolution of I, which is multigraded, then they are a reduction pair of the resolution of J corresponding to the subtree of $\mathrm{MVT}(I)$ hanging from this node. Two generators of these nodes can be a reduction pair only if they are generators of $\mathcal{P}_i(\tilde{J})$ and $\mathcal{P}_i(J')$ for the same i. Since the number of zeros in $\tilde{u}_1 \dots \tilde{u}_{l_1}$ and $u'_1 \dots u'_{l_2}$ gives the dimension of the nodes p_1 and p_2 in the subtree hanging from J, they must be equal, and hence p_1 and p_2 are compatible. □

Lemma 3.1 is also reproduced in the tree via compatible nodes:

Proposition 3.4. *Let $\mathcal{A} = \{a_1, \dots, a_k\}$ and $\mathcal{B} = \{b_1, \dots, b_l\}$ be the positions of two sets of compatible nodes such that the nearest common ancestor J of every pair $(a \in \mathcal{A}, b \in \mathcal{B})$ coincides and such that the same monomial x^μ appears as a generator of each node in $\mathcal{A}$ and $\mathcal{B}$. Then,*

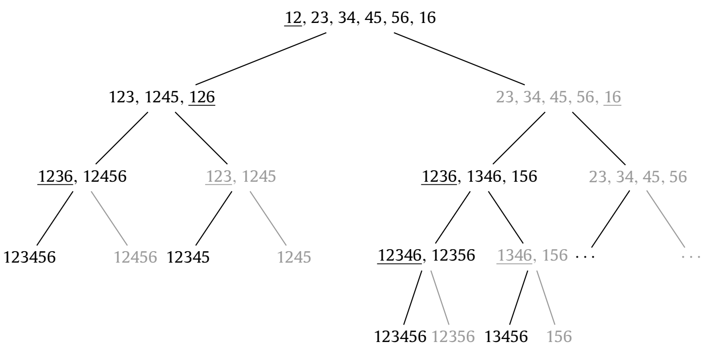
If there is no divisor of x^μ in any node of the subtree hanging from J' compatible with the nodes in $\mathcal{A}$ then the generators of multidegree μ in $\mathcal{A}$ and $\mathcal{B}$ form k reduction pairs.

Proof. The result is just a translation of Lemma 3.1 to $\mathrm{MVT}(I)$. □

Example 3.5. Consider the ideals $C_{k,n} \subseteq \mathbf{k}[x_1, \dots, x_n]$ given by

$$C_{k,n} = \langle x_1x_2 \cdots x_k, x_2x_3 \cdots x_{k+1}, \dots, x_nx_1 \cdots x_{k-1} \rangle.$$

These ideals are called *cyclic k-out-of-n*. The case $k = 2$ corresponds to the edge ideals of cycle graphs. These ideals appear in Bigatti (1997) as an example of ideals with particularly bad behaviour with respect to the computation of their Hilbert series. Using the results in the above sections, the Betti numbers of $C_{k,n}$ ideals (and other related ideals) can be computed without computing their minimal free resolution, generalizing the results in Jacques (2004); Visscher (2006). The proof consists on an enumeration too long for the scope of this paper, we show the kind of arguments used by working the example $C_{2,6}$. A Mayer–Vietoris tree of $C_{2,6}$ is the following (we write i instead of x_i and underline the pivot monomials):



The tree hanging from node 7 corresponds to a consecutive 2-out-of-5 ideal, which has a minimal mapping cone resolution, see Saenz-de-Cabezón and Wynn (2009). There is no x_1 involved in this subtree, therefore there is no generator in it appearing in the rest of the tree. A simple exploration of the rest of the tree shows that the only multidegrees that are repeated as generators of relevant nodes are $x_1x_2x_3x_4x_5x_6$ and $x_1x_2x_3x_6$. The first one, $x_1x_2x_3x_4x_5x_6$, appears in two nodes of dimension 3. Therefore by Proposition 3.3 they cannot form a reduction pair and then $\beta_{3,x_1x_2x_3x_4x_5x_6}(C_{2,6}) = 2$. The multidegree $x_1x_2x_3x_6$ appears in nodes 4 and 6 (which are compatible) and since there is no divisor of $x_1x_2x_3x_6$ in the relevant nodes of dimension 1 hanging from node 3, applying Proposition 3.4 we obtain that the two appearances of $x_1x_2x_3x_6$ are a reduction pair. Hence we conclude that the mapping Mayer–Vietoris resolution of $C_{2,6}$ is minimal after one reduction step. □

Of particular relevance are some *extremal* elements of the resolution, in particular those defining the *width* of the resolution, which are related to the Castelnuovo–Mumford regularity of the ideal. These extremal elements are also extremal in the tree. We define the regularity of a Mayer–Vietoris tree of a monomial ideal I as $reg(\mathrm{MVT}(I)) = \max\{\deg(m) - i | m \text{ is a generator of a node of dimension } i \text{ of } \mathrm{MVT}(I)\}$. If $reg(\mathrm{MVT}(I))$ is attained at a generator of a relevant node that is not part of a reduction pair in $\mathrm{MVT}(I)$ then $reg(\mathrm{MVT}(I))$ equals the Castelnuovo–Mumford regularity of I, if not, then it is an upper bound. However, to find $reg(\mathrm{MVT}(I))$ it is enough to look at the relevant leaves of the tree.

Proposition 3.6. *Let I be a monomial ideal and $\mathrm{MVT}(I)$ a Mayer–Vietoris tree of I. $reg(\mathrm{MVT}(I))$ is attained at one of the relevant leaves of $\mathrm{MVT}(I)$.*

Proof. For each node $\mathrm{MVT}_p(I)$ of dimension i define its regularity as $reg(\mathrm{MVT}_p(I)) = \max\{\deg(m) - i | m \text{ is a generator of } \mathrm{MVT}_p(I)\}$, then $reg(\mathrm{MVT}(I))$ is the maximum of the regularities of its nodes. It is clear that if $\mathrm{MVT}_p(I)$ has only two generators then $reg(\mathrm{MVT}_{2p}(I)) \geq reg(\mathrm{MVT}_p(I))$ and $\mathrm{MVT}_{2p}(I)$ is a relevant leaf of $\mathrm{MVT}(I)$. Now, if $\mathrm{MVT}_p(I)$ has more than two generators, then:

- If we choose as pivot monomial one of maximum degree, then $reg(\mathrm{MVT}_{2p}(I)) \geq reg(\mathrm{MVT}_p(I))$ and $\mathrm{MVT}_{2p}(I)$ has a number of generators strictly smaller than $\mathrm{MVT}_p(I)$.
- If we do not choose as pivot monomial one of maximum degree then $reg(\mathrm{MVT}_{2p+1}(I)) = reg(\mathrm{MVT}_p(I))$ and $\mathrm{MVT}_{2p+1}(I)$ has one generator less than $\mathrm{MVT}_p(I)$.

Then, for each node we know that there is another node with at least the same regularity and a strictly smaller number of generators. Iterating the process we reach a node with two generators and hence the result holds. □

4. Computations based on Mayer–Vietoris trees

The concepts and results in the previous section lead naturally to an algorithm that constructs and reduces Mayer–Vietoris trees. Such an algorithm consists of two stages. The first stage takes a monomial ideal I and constructs $\mathrm{MVT}(I)$ and the second stage uses the results in Section 3 to decide whether each of the multidegrees that are repeated in relevant nodes of $\mathrm{MVT}(I)$ is part of a reduction pair or not. The final output of the algorithm consists of two lists L_1 and L_2. The first list, L_1, is formed by all multidegrees in the relevant nodes of $\mathrm{MVT}(I)$ (together with their dimensions) that are not part of any reduction pair in the Mayer–Vietoris resolution of I. The second list, L_2, contains all those multidegrees for which the second stage of the algorithm could not decide whether they are part of a reduction pair. Observe that L_1 is contained in the support of the minimal free resolution of I and that L_1 and L_2 together support a Mayer–Vietoris resolution of I which is minimal whenever L_2 is empty. We will use the name `decided` to refer to the list L_1 and `undecided` to refer to L_2.

4.1. Pseudo Betti diagram

Recall that a Betti diagram of an ideal I is a matrix whose entry (i, j) is $\beta_{i,i-j}(I)$. The graded output of the algorithm just described can be seen as a *pseudo Betti diagram* which differs from the usual Betti diagram in that the entry (i, j) contains two numbers, the first one being a lower bound of $\beta_{i,j-i}(I)$ and the second one an upper bound. In our case the lower bound is the sum of elements of degree $j - i$ in the i-th entry of the `decided` list and for the upper bound we must add the corresponding number from the `undecided` list. If both bounds coincide (i.e. there is no undecided element of degree $j - i$ at dimension i) we just output the correct value of $\beta_{i,j-i}(I)$.

Remark 4.1. A well known strategy to compute the multigraded Betti numbers of monomial ideals consists in interpreting the Betti numbers at each multidegree as the reduced homology of a certain simplicial complex (Miller and Sturmfels, 2004). The problem is reduced then to compute simplicial homology with coefficients on a field, which amounts to linear algebra computations. However, the computational cost of this approach advises to reduce as much as possible the candidate multidegrees in which reduced homology of the corresponding simplicial complexes must be computed. In this respect, some research efforts have been made in the last decade, see Bayer and Taylor (2009) as a recent example. The approach using Mayer–Vietoris trees has several advantages in this context. First, it produces a typically small set of such candidates i.e. those in the `undecided` list. Second, for each such candidate it gives a list of possible degrees in which the reduced homology of the corresponding simplicial complex might not vanish, so that the size of the final linear algebra problem is reduced.

4.2. Hilbert series

The numerator of the *multigraded Hilbert series* of a monomial ideal I equals the alternating sum of the multidegrees that support any multigraded resolution of I counting multiplicities. This expression of the numerator of the multigraded Hilbert series is redundant (even in the case we use the minimal free resolution) in the sense that some cancellations can be done among the summands in it, but it is the most adequate form for some applications (Saenz-de-Cabezón and Wynn, 2009). Since any resolution can be used to obtain such an expression of the multigraded Hilbert series, we can use the relevant nodes of any Mayer–Vietoris trees avoiding the minimization step. This provides a fast algorithm for the computation of Hilbert series. After the minimization step, a more compact expression can be given, which still comes from a resolution of I.

4.3. Irreducible decompositions and related computations

The computation of the irredundant irreducible decomposition of a monomial ideal I is equivalent to the computation of the Alexander dual of I, the facets of its Scarf complex or the maximal standard monomials of I (Miller and Sturmfels, 2004; Bigatti and Sáenz-de-Cabezón, 2009; Roune, 2009) and all this is equivalent to the computation of its multigraded Betti numbers at dimension $n-1$. A specialization of the Mayer–Vietoris algorithm for the computation of the multigraded Betti numbers at dimension $n-1$ uses the Mayer–Vietoris tree of the artinian closure $\hat{I}$ of I and performs two types of prunings on this tree. The first type, *pruning by number of generators*, is done whenever we do not have enough generators to reach projective dimension $n-1$ (from Taylor resolution we know that to reach projective dimension k we need at least $k+1$ generators). The second one, *pruning by number of indeterminates*, is done when we do not have all the indeterminates involved in the intermediate nodes. In this way we obtain a set of multidegrees that are *candidates* to be in the set $\{\mu \in \mathbb{N}^n | \beta_{n-1,\mu}(\hat{I}) \neq 0\}$ and for these candidates there is a simple and fast test to detect whether there is homology at that multidegree. The algorithm is presented in Bigatti and Sáenz-de-Cabezón (2009) where the details are given.

4.4. Euler characteristic of simplicial complexes

Let $\chi(\Delta)$ be the Euler characteristic of a simplicial complex. The *Euler–Poincaré formula* states that $\chi(\Delta) = \sum_i (-1)^i \beta_i(\Delta)$ where $\beta_i(\Delta)$ are the Betti numbers of Δ. Every simplicial complex Δ has an associated *Stanley–Reisner ideal* I_Δ which is a squarefree monomial ideal. By Hochster's formula (see Miller and Sturmfels (2004) for instance) the Euler characteristic of Δ can be computed from the Betti numbers of I_Δ of multidegree $x_1 \cdots x_n$. For every monomial ideal I, a multidegree μ and any resolution $\mathcal{P}$ of I we have that

$$\sum_i (-1)^i \beta_{i,\mu}(I) = \sum_i (-1)^i \rho_{i,\mu}(\mathcal{P}),$$

where $\rho_{i,\mu}(\mathcal{P})$ is the rank of the multidegree μ piece of the module at homological degree i of $\mathcal{P}$. Then, we can compute the Euler characteristic of a simplicial complex Δ using any resolution of I_Δ. In particular, we can use Mayer–Vietoris resolutions. For this computation we use the Mayer–Vietoris tree of I_Δ, prune it by number of generators and keep only those nodes with the generator $x_1 \cdots x_n$ (note that these are leaves in the tree). Using the corresponding alternating sum, we obtain $\chi(\Delta)$.

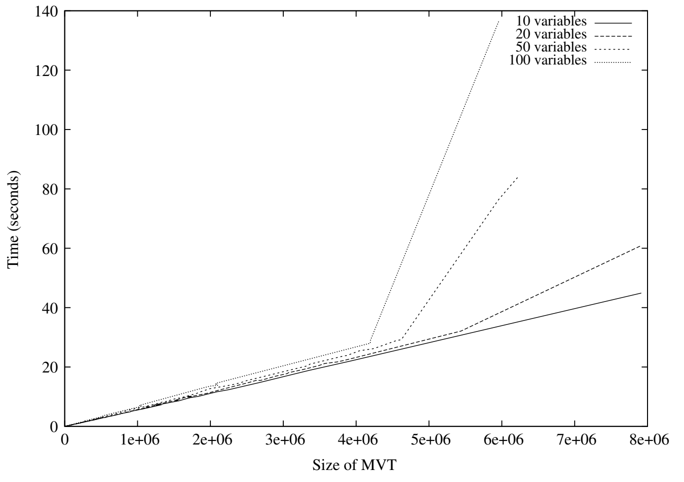


Fig. 2. Time in seconds against size of MVT resolution in random ideals.

4.5. Experiments

4.5.1. Basic algorithm

Our algorithms have been implemented using the C++ library `CoCoALib` (CoCoATeam, 2010), which is easy to use and brings together the capabilities of the C++ programming language and built-in algebraic structures. It is part of the CoCoA system (CoCoATeam, 2009).

We first show the behaviour of the construction and reduction of Mayer–Vietoris trees for random ideals. Fig. 2 shows the time of the construction of a Mayer–Vietoris tree against size of the corresponding resolution, where *size* is understood as the sum of the ranks of the modules in the resolution. We used ideals in 10, 20, 50 and 100 variables. The number of generators ranges between 0 and 50 and the exponent in each of the variables ranges also from 0 to 50. Fig. 3 shows the difference between the size of the resolution obtained running the first part of the MVT algorithm and the size of the resolution after applying the reduction described in Section 3. Both sizes are compared to the size of the minimal free resolution of these ideals. The examples are denoted by the number of variables v and number of minimal generators g. Observe that in Fig. 3 the size of the resolution of the last four examples (marked with an asterisk) has been multiplied by 1000 so that it could be seen in the figure.

4.5.2. Pseudo Betti diagram

The main goal of the algorithm we present is to compute multigraded Betti numbers without actually computing minimal free resolutions. Of course one point we must show is that this is faster than computing the full resolution, even if it is a priori obvious. Table 1 shows timings of algorithms computing the minimal free resolutions (when possible) in the computer algebra systems Macaulay2 (Grayson and Stillman, 2009) and Singular (Greuel et al., 2005) together with the times of the computations of Mayer–Vietoris trees.[2] The results of these experiments show that the computation of Mayer–Vietoris trees is, as expected, much faster than the computation of the full resolution. In particular, for large ideals for which the computation of the resolution is unfeasible, we can obtain information on the Betti numbers using this approach. Even if in principle the minimality of the corresponding Mayer–Vietoris resolution is not guaranteed, the output of the algorithm says whether there were some undecided multidegrees or not. In all examples computed there were no undecided elements, so we in fact obtained the actual Betti numbers.

[2] The entries marked with an asterisk in column Singular where computed using the command `lres` which does not guarantee the minimal resolution, in these cases the minimal resolution could not be computed using the Singular command `mres`. OOM in the table stands for Out Of Memory.

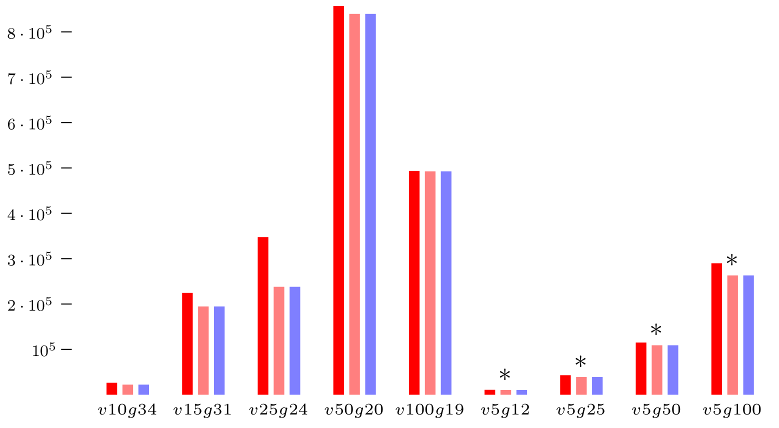


Fig. 3. Size of MVT resolution before and after the reduction step compared to size of minimal free resolution.

Table 1
Timings of the examples in Fig. 3.

Variables	Min. gens.	Macaulay2	Singular	MVT
10	34	2'40	1'66	0'02
15	31	21'3	148'38*	0'1
25	24	36'57	391'15*	0'62
50	20	OOM	3280'19*	6'32
100	19	OOM	922'71*	3'65
5	12	0	0	0
5	25	0	0	0
5	50	0'01	0	0'01
5	100	0'08	0'65	0'02

Table 2 shows the performance of the MVT algorithm for computing Castelnuovo–Mumford regularity vs. the algorithms implemented in Singular and Macaulay2 for different kinds of monomial ideals. The columns showing times for Singular and Macaulay2 show on one hand the algorithms in the `Singular` library `mreg.bib` that implements the algorithm by Bermejo, Gimenez and Greuel based on the results in Bermejo and Gimenez (2006) and in the other hand the Macaulay2 command `regularity`. The Mayer–Vietoris tree algorithm is not suitable for the computation of Castelnuovo–Mumford regularity in some cases. A trivial example, shown in Table 2 with the notation $n = k$ is the ideal generated by the indeterminates of the ring. It has a linear resolution, hence its regularity is one, which is immediately computed by algorithms like the one in Bermejo and Gimenez (2006), but its Mayer–Vietoris tree has size 2^n and the full tree is computed when we use this algorithm. On the other hand, in random ideals the comparison between the two algorithms in Table 2 shows that MVT is a good alternative for this kind of computations. Here the point is that the algorithms in Singular and Macaulay2 are much more general than ours, both work for homogeneous polynomial ideals in any characteristic. The results show that it is worth working on algorithms specifically targeted to monomial ideals. The examples computed gave always the correct result for the regularity although in general we can only expect bounds.

4.5.3. Hilbert series

For multigraded Hilbert series we consider the CoCoA implementation of Bigatti's algorithm (Bigatti, 1997) given in the function HilbertSeriesMultiDeg. Tables 3 and 4 show Mayer–Vietoris trees (MVT) vs. Bigatti's algorithm in two different kinds of ideals. In Table 3 we use *Valla ideals* (Valla, 2004), a class of zero-dimensional ideals whose resolution size is relatively big with respect to the number of variables. We see that Bigatti's algorithm is very efficient on these ideals. Table 4 shows the same

Table 2
Singular, Macaulay2 and MVT times for the computation of Castelnuovo–Mumford regularity.

Example	Singular	Macaulay2	MVT
$n = 5$	0	0	0
$n = 20$	0	0	6'17
$n = 25$	0	0	25'5
$v10g34$	1'66	1'29	0'02
$v15g31$	163'11	23'08	0'1
$v25g24$	428'60	35'20	0'62
$v50g20$	3904'71	OOM	6'32
$v100g19$	1103'45	OOM	3'65

Table 3
HilbertSeriesMultiDeg and MVT times for some Valla ideals.

Example	Min. gens.	HilbertSeriesMultiDeg	MVT
Valla(6,4,2)	126	0'08	0'02
Valla(8,4,2)	330	0'23	0'1
Valla(10,4,2)	715	0'78	0'62
Valla(12,4,2)	1365	4'72	3'42
Valla(14,4,2)	2380	24'23	19'09
Valla(16,4,2)	3876	OOT	117'23
Valla(8,5,3)	792	0'57	0'5
Valla(8,7,3)	3432	2'39	9'27
Valla(10,5,3)	2002	3'21	3'95
Valla(10,7,3)	11440	33'93	121'42
Valla(12,5,3)	4368	21'02	23'17
Valla(12,7,3)	31824	OOT	1037'77

Table 4
HilbertSeriesMultiDeg and MVT times for consecutive k-out-of-n ideals.

n	$I_{5,n}$	$I_{10,n}$	$I_{15,n}$
20	0'05::0'02	0'02::0	0'02::0
25	0'1::0'07	0'04::0	0'02::0
30	0'57::0'32	0'09::0'02	0'03::0
35	4'09::1'73	0'12::0'03	0'04::0'01
40	62'39::7'38	0'18::0'09	0'07::0'01
45	OOT::46'29	0'5::0'3	0'14::0'03

comparison in *consecutive k-out-of-n* ideals (Saenz-de-Cabezón and Wynn, 2009), whose resolution is relatively small with respect to the number of variables. In this case, we see that when the number of variables grows the MVT algorithm behaves better. These experiments show that the performance of the MVT algorithm is comparable to that of Bigatti's algorithm. We can also observe that the MVT algorithm is more sensitive to the growth of the number of generators when the number of variables is small, while Bigatti's algorithm seems to be more sensitive to the growth of the number of variables. The entries in Table 4 are of the form {Time taken by HilbertSeriesMultiDeg} :: {Time taken by MVT}. In these tables OOT stands for Out Of Time.

4.5.4. *Other computations*

We finish with a brief comment on other computations made using Mayer–Vietoris trees. The performance of the Mayer–Vietoris tree algorithm when computing irreducible decompositions of monomial ideals was described in Bigatti and Sáenz-de-Cabezón (2009) where it was shown that even

Table 5
Macaulay2 and MVT times for the computation of Euler characteristic of the simplicial complex corresponding to some squarefree monomial ideals.

Example	Macaulay2	MVT
v20g500	1’76	0’36
v20g622	2’34	0’53
v20g2000	7’54	3’49
v20g4000	OOM	12’77
v30g300	22’64	1’41
v20g600	OOM	4’68
v30g1253	OOM	3’49

if it is not an algorithm specifically targeted to irreducible decompositions, it is closer than others to the *Slice algorithm* (Roune, 2009), which is the fastest algorithm available for this computation.

In the case of Euler characteristic we show some times comparing the computation of Euler characteristic using the `fvector` function of the Macaulay2 package `SimplicialComplex` and using MVT, see Table 5. The table shows that the behaviour of our algorithm is quite efficient for this computation. In this case, the column *example* of Table 5 shows the number of variables and generators of I_Δ in each example.

5. Conclusions

We have presented in this paper a set of algorithms that perform homological computations on monomial ideals without computing their minimal free resolution. They are based on Mayer–Vietoris trees, which display the support of the iterated mapping cone resolution corresponding to Mayer–Vietoris short exact sequences. We have shown that this approach provides efficient computation of the basic homological and numerical invariants of monomial ideals. Moreover, simple modifications of the algorithm provide good methods for the computation of irreducible decompositions of monomial ideals and Euler characteristic of simplicial complexes.

There is a lack of specific methods for monomial ideals in the most widely used computer algebra systems. Since the study of monomial ideals has been significantly improved in the last years due to their applicability in several areas, it is reasonable to make efforts working on methods and algorithms specifically targeted to monomial ideals.

Acknowledgements

The author was partially supported by Ministerio de Ciencia e Innovación (Spain), grant MTM2009-13842-C02-01 and Comunidad Autónoma de La Rioja (Spain), grant FOMENTA 2007/03.

References

Bermejo, I., Gimenez, Ph., 2006. Saturation and Castelnuovo-Mumford regularity. Journal of Algebra 303, 592–617.
Bigatti, A.M., 1997. Computation of Hilbert–Poincaré series. Journal of Pure and Applied Algebra 119 (3), 237–253.
Bayer, D., Stillman, M., 1992. Computation of Hilbert functions. Journal of Symbolic Computation 14, 31–50.
Bigatti, A.M., Sáenz-de-Cabezón, E., 2009. Computation of the $(n-1)$-st Koszul homology of monomial ideals and related algorithms. In: May, J.P. (Ed.), Proceedings of ISSAC 2009. ACM Press, pp. 31–38.
Bayer, D., Taylor, A., 2009. Reverse search for monomial ideals. Journal of Symbolic Computation 44, 1477–1486.
Charalambous, H., Evans, E. G., 1995. Resolutions obtained by iterated mapping cones. Journal of Algebra 176, 750–754.
CoCoATeam, 2009. CoCoA: a system for doing Computations in Commutative Algebra. Available at http://cocoa.dima.unige.it.
CoCoATeam, 2010. CoCoALib: a GPL C++ library for doing Computations in Commutative Algebra. Available at http://cocoa.dima.unige.it.
Eliahou, S., Kervaire, M., 1990. Minimal resolutions of some monomial ideals. Journal of Algebra 129, 1–25.
Francisco, C.A., 2005. Resolutions of small sets of fat points. Journal of Pure and Applied Algebra 203, 220–236.
Greuel, G.-M., Pfister, G., Schönemann, H., 2005. Singular 3.0. A Computer Algebra System for Polynomial Computations, Centre for Computer Algebra, University of Kaiserslautern. http://www.singular.uni-kl.de.

Grayson, Daniel R., Stillman, Michael E., 2009. Macaulay 2, a software system for research in algebraic geometry. Available at http://www.math.uiuc.edu/Macaulay2/.
Herzog, J., Takayama, Y., 2002. Resolutions by mapping cones. Homology, Homotopy and Applications 4 (2), 277–294.
Jacques, S., 2004. Betti numbers of graph ideals. Ph.D. Thesis, Univ. Sheffield, (UK).
Lyubeznik, G., 1998. A new explicit finite free resolution of ideals generated by monomials in an R-sequence. Journal of Pure and Applied Algebra 51 (1–2), 193–195.
Miller, E., Sturmfels, B., 2004. Combinatorial Commutative Algebra. Springer Verlag.
Roune, B.H., 2009. The Slice algorithm for irreducible decomposition of monomial ideals. Journal of Symbolic Computation 44, 358–381.
Saenz-de-Cabezón, E., 2008. Combinatorial Koszul homology: Computations and Applications. Ph.D. Thesis, Univ. La Rioja, (Spain). Available at http://arxiv.org/abs/0803.0421.
Saenz-de-Cabezón, E., 2009. Multigraded Betti numbers without computing minimal free resolutions. Applicable algebra in Engineering, Communication and Computing 20 (5–6), 481–495.
Saenz-de-Cabezón, E., Wynn, Henry P., 2009. Betti numbers and minimal free resolutions for multi-state system reliability bounds. Journal of Symbolic Computation 44, 1311–1325.
Taylor, D., 1966. Ideals generated by monomials in an *R*-sequence. Ph.D. thesis, University of Chicago (USA).
Valla, G., 2004. Betti numbers of some monomial ideals. Proceedings of the AMS 133-1, 57–63.
Visscher, D., 2006. Minimal free resolutions of complete bipartite graph ideals. Communications in Algebra 34 (10), 3761–3766.