Chordal Graphs and Minimal Free Resolutions

David J. Marchette* David A. Johannsen*

Abstract

The problem of computing the minimal free resolution of the edge ideal of a graph has attracted quite a bit of interest in recent years. For chordal graphs, it is known that a recursive formula exists. We discuss the proof of this recursion, and provide implementation advice to guarantee that the recursion is achieved. We also discuss a simple addition to the algorithm that allows the calculation of the minimal free resolution of "nearly chordal" graphs within the same algorithm.

1 Introduction

A graph is a pair $G = (V, E)$, where $V = V(G)$ is a finite set $\{v_1, \ldots, v_n\}$, the vertices, and $E = E(G)$ is a set of pairs of vertices, the edges. We will usually write vw for the edge $\{v, w\}$. We assume all graphs are simple: edges are pairs of distinct vertices; there are no edges from a vertex to itself. The number of vertices of a graph is called the order of the graph. The number of edges is called the size.

An induced subgraph of a set of vertices $W = \{v_{i_1}, \ldots, v_{i_k}\}$ is the graph (W, E'), where E' is the subset of E consisting of those pairs containing only elements of W: $E' = \{w_i w_j \in E | w_i, w_j \in W\}$. Thus, the induced subgraph contains all the edges between its vertices that exist in the original graph.

A cycle is a set of vertices $\{v_{i_1} v_{i_2}, v_{i_2} v_{i_3}, \ldots, v_{i_n} v_{i_1}\}$. We will also refer to the corresponding graph as a cycle, and denote it C_n. A cycle has a chord if the subgraph induced by the vertices of the cycle has an edge between two non-adjacent vertices (non-adjacent as members of the cycle). The complete graph K_n is the graph on n vertices such that $uv \in E(K_n)$ for all $u \neq v \in V(K_n)$. The complete bipartite graph on n, m vertices is the graph for on the disjoint union of two sets ($V = X \cup Y$), where $|X| = n$ and $|Y| = m$, with all possible edges between the sets ($E = \{xy | x \in X \text{ and } y \in Y\}$). We denote the graph as $K_{n,m}$.

Definition 1.1. *A graph is chordal if all induced cycles have at least one chord.*

*Naval Surface Warfare Center, Code Q21, Dahlgren, VA 22448. {david.marchette,david.johannsen}@navy.mil

Definition 1.2. *The open neighborhood of a vertex* v*, denoted* $N(v)$*, is the set of vertices* $\{w \in V | vw \in E\}$*. That is, it consists of all neighbors of* v*. Note that* $v \notin N(v)$*. The closed neighborhood of* v*, denoted* $N[v]$ *is the set* $N(v) \cup \{v\}$*.*

Definition 1.3. *Given a graph* G*, define star*(G) *to be the graph defined by adding a single vertex* v *to* G*, with edges from* v *to each vertex of* G*. We will denote by star*(n) *the star on the trivial graph on* n *vertices. Thus star*$(n) = K_{1,n}$ *has order* $n+1$ *and contains one vertex of degree* n *and* n *vertices of degree* 1*. A wheel graph is defined to be star*(C_n)*.*

Definition 1.4. *A vertex* v *is simplicial if* $N(v) = K_n$ *for some* n*.*

2 Commutative Algebra

Suppose we have a graph G on n vertices $\{v_1, \ldots, v_n\}$. Let k be a field (we may assume for the purposes of this paper that $k = \mathbb{C}$). Let $S = k[x_1, \ldots, x_n]$, the ring of polynomials in n variables with coefficients in k. Our notation for edges, $v_i v_j$ is evocative of monomials, and we make use of this observation in the definition of the edge ideal of a graph.

Definition 2.1. *The edge ideal of* G *is the ideal of* S *generated by the monomials* $\{x_i x_j | v_i v_j \in E\}$*. We write* $\mathcal{I}(G)$ *for the edge ideal of* G*.*

Although there is a one-to-one correspondence between the vertices v_i and the variables x_i, we will keep to the usual naming convention so that it is clear when we are referring to elements of the graph and when we are referring to elements of the ring. For more information on edge ideals see Jacques [2004], Jacques and Katzman [2005], Stanley [1996], Villarreal [2001], Miller and Sturmfels [2005].

We can now use the tools of commutative and algebraic geometry to learn about a graph by studying its edge ideal. One such tool, the one we will be focused on in this paper, is the minimal free resolution.

Definition 2.2. *An augmented free resolution of an* S*-module* M *is an exact sequence of the form*

$$0 \longrightarrow F_m \longrightarrow F_{m-1} \cdots \longrightarrow F_1 \longrightarrow F_0 \longrightarrow M \longrightarrow 0$$

where each F_i *is a free* S*-module (a direct sum of* β_i *copies of* S*). The image of* F_i *in the sequence is called the* i^{th} syzygy *module. Such a resolution is minimal if* m *is minimal over all such, and each* β_i *is minimal. We define the minimal free resolution of the edge ideal to be the minimal free resolution of* $I = \mathcal{I}(G)$*, in which case the free resolution becomes:*

$$0 \longrightarrow S^{\beta_m} \xrightarrow{\phi_m} S^{\beta_{m-1}} \xrightarrow{\phi_{m-1}} \cdots \xrightarrow{\phi_2} S^{\beta_1} \xrightarrow{\phi_1} S^{\beta_0} \xrightarrow{\phi_0} I \longrightarrow 0.$$

The β_i *are called the* Betti *numbers. The length of the resolution is* m*.*

Definition 2.3. *The projective dimension of an edge ideal is the length of the minimal resolution.*

Note that $\beta_0 = \text{size}(G)$. There is a natural $\mathbb{N}^n$ grading on the ring $k[x_1, \dots, x_n]$, which gives a natural grading on the resolution.

$$0 \longrightarrow \bigoplus_j S(-j)^{\beta_{m,j}} \longrightarrow \cdots \longrightarrow \bigoplus_j S(-j)^{\beta_{1,j}} \longrightarrow \bigoplus_j S(-j)^{\beta_{0,j}} \longrightarrow I \longrightarrow 0,$$

where $S(-j)$ is the shifted module obtained by shifting the degrees by j, so that the corresponding maps remain degree 0. We will be concerned with methods for computing the graded Betti numbers $\beta_{i,j}$.

3 Splitting

Definition 3.1. *An edge uv is a splitting edge if $N[u] \subset N[v]$ or $N[v] \subset N[u]$. If uv is a splitting edge, we will assume that the vertices are ordered so that $N[u] \subset N[v]$.*

Theorem 3.1 (Hà and Van Tuyl [2006b]). *If uv is a splitting edge of G, then for all $i \geq 1$ and $j \geq 0$*

$$\beta_{i,j}(\mathfrak{I}(G)) = \beta_{i,j}(\mathfrak{I}(G \setminus \{uv\})) + \sum_{k=1}^{i} \binom{n}{k} \beta_{i-1-k,j-2-k}(\mathfrak{I}(H)), \qquad (1)$$

where $n = |N[v]| - 2$, $H = G \setminus N[v]$, $\beta_{-1,0} = 1$ and $\beta_{-1,j} = 0$ for $j > 0$. Recall that we are using the convention that uv is ordered so that $N[u] \subset N[v]$.

Proof. See Hà and Van Tuyl [2006b]. □

Lemma 3.1. *A graph G has the property that all (non-trivial) induced subgraphs H contain a splitting edge if and only if G is chordal. Here "non-trivial" refers to the condition that H contain at least one edge.*

Proof. ($\Leftarrow$) Assume G is not chordal. Then there is an induced cycle C_n with $n > 3$ with no chord. But it is easy to see that any such cycle does not have a splitting edge.

($\Rightarrow$) If G is chordal, then it contains a simplicial vertex u. Since $N(u)$ is a clique, uv is a splitting edge for any $v \in N(u)$. Since any induced subgraph of a chordal graph is chordal, we have the result. □

Lemma 3.2. *For any chordal graph G there is a splitting edge e such that $G \setminus \{e\}$ is chordal.*

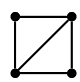
Proof. The only way removing and edge from G can make it non-chordal is if it opens up an induced C_4 with no chord. So in particular, we want to avoid chords. Let v be a simplicial vertex. Any edge e incident to v is a splitting edge. Further, it is not the chord of any cycle external to $N[v]$. Since $N[v]$ is complete, removing e cannot result in an induced cycle without a chord. Thus, any edge incident to a simplicial vertex can be removed. □

We call a function s that takes a graph containing splitting edges and returns one splitting edge a *splitting edge selection strategy*, or simply a *strategy*. We will say that Equation (1) is recursive for a class of graphs if there is a strategy for which it is recursive. We do not require that any strategy will work, only that there is one. The following theorem has been stated elsewhere (see Hà and Van Tuyl [2006a]), but is usually stated without explicit proof.

Theorem 3.2. *Under the strategy which selects splitting edges incident on simplicial vertices, Equation (1) is recursive for a graph G if and only if G is chordal.*

Proof. This is immediate from the two lemmas. □

This shows that the algorithm to apply Equation (1) recursively always works to compute the minimal resolution of G whenever G is chordal, provided the splitting edges are chosen appropriately. It is not the case, though, that it is recursive no matter what splitting edge is chosen at each step. The simplest example of this is shown below. It is clear that all edges in this graph are splitting and that this is a chordal graph. Further, if we remove any edge from the outside cycle, the resulting graph is chordal. However, if we remove the diagonal, the resulting graph is not chordal, and furthermore, does not contain any splitting edges. The two degree two vertices are simplicial, while the vertices on which the diagonal are incident are not.



The graph above is a specific case of a general class of graphs, $\{G = \text{star}(\text{star}(n))\}$, for $n > 1$, shown in Figure 1. G is not chordal. In fact there are a number of copies of C_4 in G. Note that every edge in star(star(n)) is splitting. However, all edges are not equal. If we choose uv, then $G \setminus uv$ contains no splitting edges, and Equation (1) fails to recurse. However, if we select any other edge (uv_i or vv_j) and continue to select such edges until only uv remains, Equation (1) recurses. At each step, H is the empty graph, since $N[u] = N[v] = V(G)$.

The strategy that works is to use a perfect vertex elimination scheme (Balakrishnan and Ranganathan [2000] also called a perfect (vertex) elimination ordering (PEO) (West [2001]). This is an ordering of the vertices such that each vertex is simplicial in the graph induced by the remaining vertices. If one uses the strategy of selecting edges incident on vertices in a PEO, the algorithm recurses.

Now consider the graph in Figure 1 with the edge uv removed: $K = G \setminus \{uv\}$ (we will call this a *m*ace (or a *m*ace head) because of its similarity to the head of a type of mace; see below).

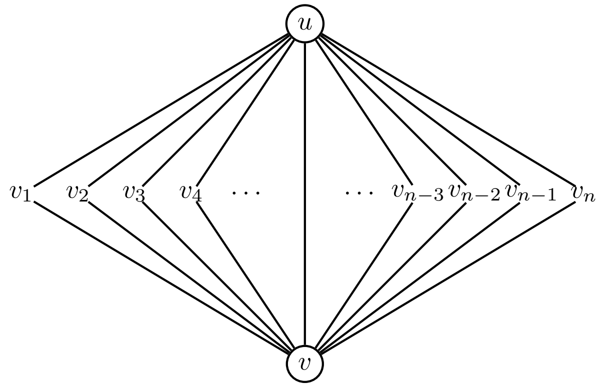


Figure 1: star(star(n)).

None of the edges is splitting, so we cannot even start the recursion. However, as we have noted, there is a strategy for using Equation (1) reflexively on G. Thus, we can turn Equation (1) around and recursively compute the Betti numbers for K using those of G:

$$\beta_{i,j}(\mathfrak{I}(K)) = \beta_{i,j}(\mathfrak{I}(K \cup \{uv\})) - \sum_{k=1}^{i} \binom{n}{k} \beta_{i-1-k,j-2-k}(\mathfrak{I}(H)). \tag{2}$$

Thus, if an edge uv is missing from a graph, but would be splitting if added to the graph, we can use Equation (2). We will say that we "add a splitting edge" when we mean an edge which is splitting in the resultant graph. Note that the subgraph H is computed on the augmented graph $K \cup \{uv\}$ using $N[v] \subset V(K \cup \{uv\})$. In computing the first summand of the right hand side, we need not select uv as the splitting edge (and in fact we must not, since this would bring us back to our starting point). Instead, for the equation to be recursive, the addition of uv must induce other edges to be splitting in the resultant graph. This is the case in our example of $K = \text{star}(\text{star}(n)) \setminus \{uv\}$. If the resultant graph is chordal, then our PEO strategy will select the appropriate edges, and we can recurse to the solution.

Thus if adding a splitting edge results in a chordal graph, then we can recurse. Similarly, if there is a sequence of splitting edges we can add for which the final graph is chordal, then we can recurse. Note that algorithms that produce PEOs on chordal graphs can be modified (if necessary) to produce partial PEOs: a set of vertices each of which is simplicial in the graph induced by the vertices in the rest of the list plus all vertices not in the list. If the graph is non-chordal (and contains at least one simplicial vertex), we can apply our recursion to this partial PEO until we have processed all the splitting edges. Other methods must be employed to compute the Betti numbers for the remaining subgraphs. At this point, one could attempt to apply Equation (2). It is unclear, from a practical standpoint, whether it is better to employ Equation (2) up front, on the original graph, or only once we hit the end of the partial PEO.

There are three ways to add a splitting edge to a graph: one can connect an isolated vertex, resulting in a pendant; one can connect a pendant, resulting in a triangle; one can add a diagonal to a square (4-cycle). The first two don't change the "chordal status" of the graph: they cannot make a non-chordal graph chordal. The third requires that one of the vertices of the square have the property that its neighborhood is contained in the neighborhood of its opposite. Thinking of the mace head depicted in Figure 1 (with the edge uv removed), the vertex can be the "point" of the mace, resulting in the added splitting edge being uv. Note that this vertex is not simplicial, even after adding the edge, and so the added edge will not be selected by the algorithm until all the other edges of the mace have been selected. Note that we could have added v_iv_j for any $i \neq j$, but this would not have resulted in a chordal graph (we would have to add many such edges). Thus, the strategy is to select splitting edges which make the graph "most chordal": fill in the maximum number of squares.

4 Conclusion

We have discussed the proof of Hà and Van Tuyl's theorem that Equation (1) is recursive on chordal graphs, and used this, with the notion of a perfect elimination ordering (PEO) to provide an explicit strategy for applying this theorem. A trivial observation allows us to apply the same algorithm to slightly more general graphs than chordal graphs.

More generally, the idea of using a partial PEO can reduce the computation of the Betti numbers of an edge idea in large graphs to that of computations on smaller graphs. This, combined with "filling in the squares" and other algorithms, or with specific formulas for different classes of graphs, can make the calculation of the Betti numbers of relatively large graphs practical.

Acknowledgments

This work was funded in part by the Office of Naval Research under the In-House Laboratory Independent Research program.

References

R. Balakrishnan and K. Ranganathan. *A Textbook of Graph Theory.* Springer, New York, 2000.

Huy Tài Hà and Adam Van Tuyl. Monomial ideals, edge ideals of hypergraphs, and their minimal graded free resolutions, 2006a.

Huy Tài Hà and Adam Van Tuyl. Splittable ideals and the resolutions of monomial ideals, 2006b.

Sean Jacques. *Betti numbers of graph ideals*. PhD thesis, 2004.

Sean Jacques and Mordechai Katzman. The betti numbers of forests, 2005. URL `http://arxiv.org/pdf/math.AC/0501226.pdf`.

Ezra Miller and Bernd Sturmfels. *Combinatorial Commutative Algebra*, volume 227 of *Graduate Texts in Mathematics*. Springer, New York, 2005.

Richard P. Stanley. *Combinatorics and Commutative Algebra*. Birkhuser, Basel, second edition, 1996.

Rafeal H. Villarreal. *Monomial Algebras*, volume 238 of *Monographs and Textbooks in Pure and Applied Mathematics*. Marcel Dekker, Inc., New York, 2001.

Douglas B. West. *Introduction to Graph Theory*. Prentice Hall, Upper Saddle River, NJ, 2001.