3

Algorithmic Graph Theory and Its Applications*

Martin Charles Golumbic

Caesarea Rothschild Institute
University of Haifa
Haifa, Israel

1. Introduction

The topic about which I will be speaking, algorithmic graph theory, is part of the interface between combinatorial mathematics and computer science. I will begin by explaining and motivating the concept of an intersection graph, and I will provide examples of how they are useful for applications in computation, operations research, and even molecular biology. We will see graph coloring algorithms being used for scheduling classrooms or airplanes, allocating machines or personnel to jobs, or designing circuits. Rich mathematical problems also arise in the study of intersection graphs, and a spectrum of research results, both simple and sophisticated, will be presented. At the end, I will provide a number of references for further reading.

I would like to start by defining some of my terms. For those of you who are professional graph theorists, you will just have to sit back and enjoy the view. For those of you who are not, you will be able to learn something about the subject. We have a mixed crowd in the audience: university students, high school teachers, interdisciplinary researchers and professors who are specialists in this area. I am gearing this talk so that it will be non-technical, so everyone should be able to enjoy something from it. Even when we move to advanced topics, I will not abandon the novice.

When I talk about a graph, I will be talking about a collection of vertices and edges connecting them, as illustrated in Figures 1 and 2. Graphs can be used in lots of different applications, and there are many deep theories that involve using graphs. Consider, for example, how cities may be connected by roads or flights, or how documents might

* This chapter is based on the Andrew F. Sobczyk Memorial Lecture delivered by the author on October 23, 2003 at Clemson University. For a biographical sketch of Andrew F. Sobczyk, see the website http://www.math.clemson.edu/history/sobczyk.html

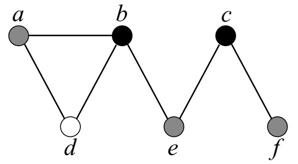

Figure 1. A graph and a coloring of its vertices.

be connected by similar words and topics. These are regularly modeled by graphs, and actions on them are carried out by algorithms applied to these graphs. Some of the terms that we need are:

- *Coloring a graph* – coloring a graph means assigning a color to every vertex, with the property that two vertices that are adjacent, i.e., connected by an edge, have different colors. As you can see in the example in Figure 1, we have colored the vertices white, grey or black. Notice that whenever a pair of vertices are joined by an edge, they have different colors. For example, a black vertex can be connected *only* to grey or white vertices. It is certainly possible to find pairs that have different colors yet are not connected, but every time we have an edge, its two end points must be different colors. That is what we mean by coloring.
- An *independent set* or a *stable set* – a collection of vertices, no two of which are connected. For example, in Figure 1, the grey vertices are pair-wise not connected, so they are an independent set. The set of vertices $\{d, e, f\}$ is also an independent set.
- A *clique* or a *complete* subset of vertices – a collection of vertices where everything is connected to each other, i.e., every two vertices in a clique are connected by an edge. In our example, the vertices of the triangle form a clique of size three. An edge is also a clique – it is a small one!
- The *complement* of a graph – when we have a graph we can turn it "inside out" by turning the edges into non-edges and vice-versa, non-edges into edges. In this way, we obtain what is called the complement of the graph, simply interchanging the edges and the non-edges. For example, the complement of the graph in Figure 1 is shown in Figure 3. We denote the complement of G by $\overline{G}$.
- An *orientation* of a graph – an orientation of a graph is obtained by giving a direction to each edge, analogous to making all the streets one way. There are many different ways to do this, since every edge could go either one way or the other. There are names for a number of special kinds of orientations. Looking at Figure 2, the first orientation of the pentagon is called *cyclic*, its edges are

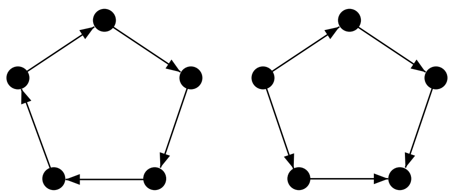

Figure 2. Two oriented graphs. The first is cyclic while the second is acyclic.

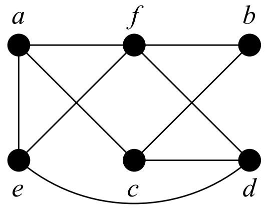

Figure 3. The complement of the graph in Figure 1.

directed consistently around in a circle, in this case clockwise; one can go round and round cycling around the orientation forever. The opposite of this idea, when there is no oriented cycle, is called an *acyclic* orientation. The second orientation of the pentagon is an acyclic orientation. You cannot go around and around in circles on this – wherever you start, it keeps heading you in one direction with no possibility of returning to your starting point.

Another kind of orientation is called a transitive orientation. An orientation is *transitive* if every path of length two has a "shortcut" of length one.[1] In Figure 4, the first orientation is not transitive because we could go from vertex d to vertex b and then over to vertex c, without having a shortcut. This is not a transitive orientation. The second orientation is transitive because for every triple of vertices x, y, z, whenever we have an edge oriented from x to y and another from y to z, then there is always a shortcut straight from x to z. Not all graphs have an orientation like this. For example, the pentagon cannot possibly be oriented in a transitive manner since it is a cycle of odd length.

2. Motivation: Interval Graphs

2.1. An Example

Let us look now at the motivation for one of the problems I will discuss. Suppose we have some lectures that are supposed to be scheduled at the university, meeting at certain hours of the day. Lecture a starts at 09:00 in the morning and finishes at 10:15; lecture b starts at 10:00 and goes until 12:00 and so forth. We can depict this on the real line by intervals, as in Figure 5. Some of these intervals intersect, for example, lectures a and b intersect from 10:00 until 10:15, the period of time when they are both in session. There is a point in time, in fact, where four lectures are "active" at the same time.

We are particularly interested in the intersection of intervals. The classical model that we are going to be studying is called an *interval graph* or the *intersection graph of a collection of intervals*. For each of the lectures, we draw a vertex of the interval graph, and we join a pair of vertices by an edge if their intervals intersect. In our

[1] Formally, if there are oriented edges $x \to y$ and $y \to z$, then there must be an oriented edge $x \to z$.

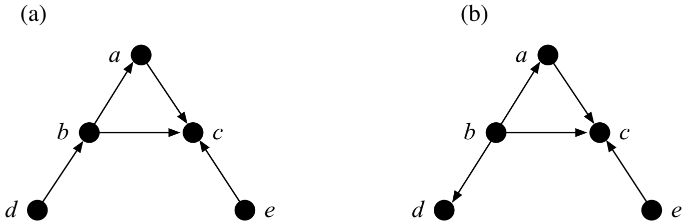

Figure 4. A graph with two orientations. The first is not transitive while the second is transitive.

example, lectures a and b intersect, so we put an edge between vertex a and vertex b, see Figure 6(a). The same can be done for lectures b and c since they intersect, and so forth. At time 13:15, illustrated by the vertical cut, we have c, d, e and f all intersecting at the same time, and sure enough, they have edges between them in the graph. They even form a clique, a complete subset of vertices, because they pair-wise intersect with each other. Some of the intervals are disjoint. For example, lecture a is disjoint from lecture d, so there is no edge between vertices a and d.

Formally, a graph is an *interval graph* if it is the intersection graph of some collection of intervals[2] on the real line.

Those pairs of intervals that do not intersect are called disjoint. It is not surprising that if you were to consider a graph whose edges correspond to the pairs of intervals that are disjoint from one another, you would get the complement of the intersection graph, which we call the *disjointness graph*. It also happens that since these are intervals on the line, when two intervals are disjoint, one of them is before the other. In this case, we can assign an orientation on the (disjointness) edge to show which interval is earlier and which is later. See Figure 6(b). Mathematically, this orientation is a partial order, and as a graph it is a transitive orientation. If there happens to be a student here from Professor Jamison's Discrete Math course earlier today, where he taught about Hasse diagrams, she will notice that Figure 6(b) is a Hasse diagram for our example.

2.2. Good News and Bad News

What can we say about intersecting objects? There is both good and bad news. Intersection can be regarded as a good thing, for example, when there is something important in common between the intersecting objects – you can then share this commonality, which we visualize mathematically as covering problems. For example, suppose I want to make an announcement over the loudspeaker system in the whole university for everyone to hear. If I pick a good time to make this public announcement, all the classes that are in session (intersecting) at that particular time will hear the announcement. This might be a good instance of intersection.

[2] The intervals may be either closed intervals which include their endpoints, or open intervals which do not. In this example, the intervals are closed.

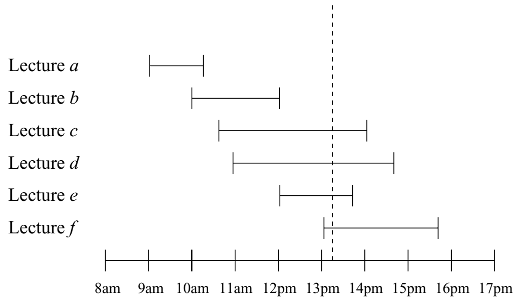

Figure 5. An interval representation.

Intersection can also be a bad thing, such as when intersecting intervals indicate a conflict or competition, and the resource cannot be shared. In our example of scheduling university lectures, we cannot put two lectures in the same classroom if they are meeting at the same time, thus, they would need different classes. Problems such as these, where the intervals cannot share the same resource, we visualize mathematically as coloring problems and maximum independent set problems.

2.3. Interval Graphs and their Applications

As mentioned earlier, not every graph can be an interval graph. The problem of characterizing which graphs could be interval graphs goes back to the Hungarian mathematician Gyorgy Hajös in 1957, and independently to the American biologist, Seymour Benzer in 1959. Hajös posed the question in the context of overlapping time intervals, whereas Benzer was looking at the linear structure of genetic material, what we call genes today. Specifically, Benzer asked whether the sub-elements could be arranged in a linear arrangement. Their original statements of the problem are quoted in [1] page 171. I will have more to say about the biological application later.

We have already seen the application of scheduling rooms for lectures. Of course, the intervals could also represent meetings at the Congress where we may need to

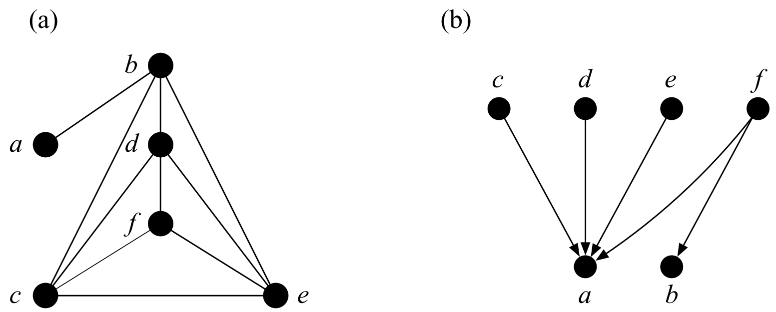

Figure 6. (a) The interval graph of the interval representation in Figure 5 and (b) a transitive orientation of its complement.

allocate TV crews to each of the meetings. Or there could be applications in which jobs are to be processed according to a given time schedule, with concurrent jobs needing different machines. Similarly, there could be taxicabs that have to shuttle people according to a fixed schedule of trips. The assignment problem common to all these applications, classrooms to courses, machines to jobs, taxis to trips, and so on, is to obtain a feasible solution – one in which no two courses may meet in the same room at the same time, and every machine or taxi does one job at a time.

3. Coloring Interval Graphs

The solution to this problem, in graph theoretic terms, is to find a coloring of the vertices of the interval graph. Each color could be thought of as being a different room, and each course needs to have a room: if two classes conflict, they have to get two different rooms, say, the brown one and the red one. We may be interested in a feasible coloring or a minimum coloring – a coloring that gives the fewest number of possible classrooms.

Those who are familiar with algorithms know that some problems are hard and some of them are not so hard, and that the graph coloring problem "in general" happens to be one of those hard problems. If I am given a graph with a thousand vertices with the task of finding a minimum feasible coloring, i.e., a coloring with the smallest possible number of colors, I will have to spend a lot of computing time to find an optimal solution. It could take several weeks or months. The coloring problem is an NP-complete problem, which means that, in general, it is a difficult, computationally hard problem, potentially needing an exponentially long period of time to solve optimally.

However, there is good news in that we are not talking about any kind of graph. We are talking about interval graphs, and interval graphs have special properties. We can take advantage of these properties in order to color them efficiently. I am going to show you how to do this on an example.

Suppose we have a set of intervals, as in Figure 7. You might be given the intervals as pairs of endpoints, [1, 6], [2, 4], [3, 11] and so forth, or in some other format like a sorted list of the endpoints shown in Figure 8. Figure 7 also shows the interval graph. Now we can go ahead and try to color it. The coloring algorithm uses the nice diagram of the intervals in Figure 8, where the intervals are sorted by their left endpoints, and this is the order in which they are processed. The coloring algorithm sweeps across from left to right assigning colors in what we call a "greedy manner". Interval *a* is the first to start – we will give it a color, solid "black". We come to *b* and give it the color "dashes", and now we come to *c* and give it the color "dots". Continuing across the diagram, notice "dashes" has finished. Now we have a little bit of time and *d* starts. I can give it "dashes" again. Next "black" becomes free so I give the next interval, *e*, the color "black". Now I am at a trouble spot because "dots", "dashes" and "black" are all busy. So I have to open up a new color called "brown" and assign that color

$a = [1,6]$
$b = [2,4]$
$c = [3,11]$
$d = [5,9]$
$e = [7,14]$
$f = [8,12]$
$g = [10,16]$
$h = [13,15]$

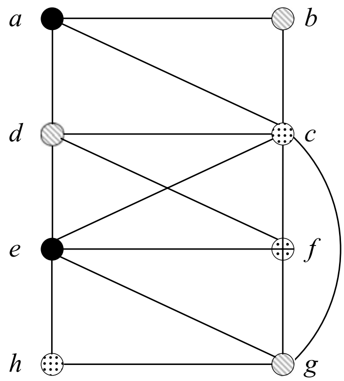

Figure 7. A set of intervals and the corresponding (colored) interval graph.

to interval f. I continue coloring from left to right and finally finish at the end. This greedy method gives us a coloring using 4 colors.

Is it the best we can do? Mathematicians would ask that question. Can you "prove" that this is the best we can do? Can we show that the greedy method gives the smallest possible number of colors? The answer to these questions is "yes".

Since this is a mathematics lecture, we must have a proof. Indeed, the greedy method of coloring is optimal, and here is a very simple proof. Let k be the number of colors that the algorithm used. Now let's look at the point P, as we sweep across the intervals, when color k was used for the first time. In our example, $k = 4$ and $P = 8$ (the point when we had to open up the color "brown".) When we look at the point P, we observe that all the colors 1 through $k - 1$ were busy, which is why we had to open up the last color k. How many intervals (lectures) are alive and running at that point P?

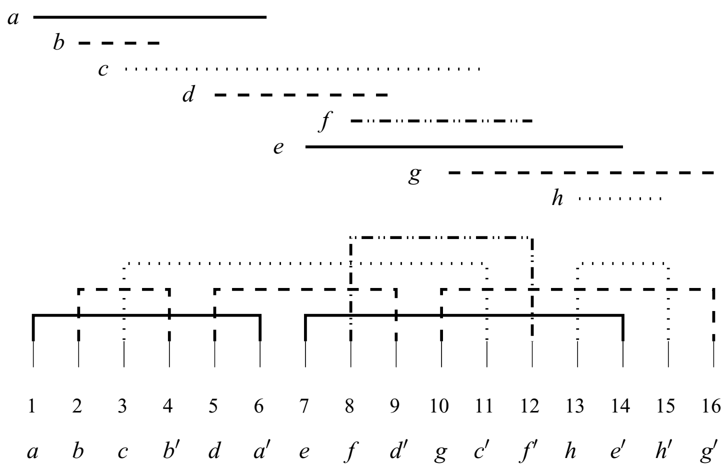

Figure 8. A sorted list of endpoints of the intervals in Figure 7.

The answer is k. I am forced to use k colors, and in the interval graph, they form a clique of size k. Formally, (1) the intervals crossing point P demonstrate that there is a k-clique in the interval graph – which means that at least k colors are required in any possible coloring, and (2) the greedy algorithm succeeded in coloring the graph using k colors. Therefore, the solution is optimal. Q.E.D.

It would be nice if all theorems had simple short proofs like this. Luckily, all the ones in this lecture will.

Interval graphs have become quite important because of their many applications. They started off in genetics and in scheduling, as we mentioned earlier. They have applications in what is called seriation, in archeology and in artificial intelligence and temporal reasoning. They have applications in mobile radio frequency assignment, computer storage and VLSI design. For those who are interested in reading more in this area, several good books are available and referenced at the end.

4. Characterizing Interval Graphs

What are the properties of interval graphs that may allow one to recognize them? What is their mathematical structure? I told you that not all graphs are interval graphs, which you may have believed. Now I am going to show you that this is true. There are two properties which together characterize interval graphs; one is the chordal graph property and the other is the co-TRO property.

A graph is *chordal* if every cycle of length greater than or equal to four has a chord. A chord means a diagonal, an edge that connects two vertices that are not consecutive on the cycle. For example, the hexagon shown in Figure 9 is a cycle without a chord. In an interval graph, it should not be allowed. In fact, it is forbidden.

Let's see why it should be forbidden. If I were to try to construct an interval representation for the cycle, what would happen? I would have to start somewhere by drawing an interval, and then I would have to draw the interval of its neighbor, which intersects it, and then continue to its neighbor, which intersects the second one but not the first one, and so forth, as illustrated in Figure 9. The fourth has to be disjoint from

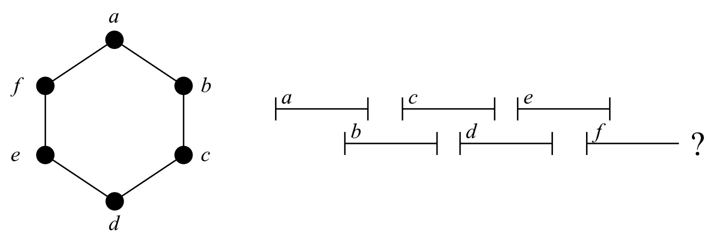

Figure 9. A cycle without a chord.

the second but hit the third, and the fifth has to hit the fourth but not the third. Finally, the sixth has to hit the fifth and not the fourth, yet somehow must close the loop and intersect the first. This cannot be done because we must draw intervals on a line. Thus, it is impossible to get an interval representation for this or any chordless cycle. It is a forbidden configuration. *A chordless cycle cannot be part of an interval graph.*

The second property of interval graphs is the transitive orientation property of the complement or co-TRO. Recall that the edges of the complement of an interval graph represent disjoint intervals. Since in a pair of disjoint intervals, one appears totally before the other, we may orient the associated edge in the disjointness graph from the later to the earlier. It is easy to verify that such an orientation is transitive: if a is before b, and b is before c, then a is before c. Now here is the punch line, a characterization theorem of Gilmore and Hoffman [10] from 1964.

Theorem 1 *A graph G is an interval graph if and only if G is chordal and its complement $\overline{G}$ is transitively orientable.*

Additional characterizations of interval graphs can be found in the books [1, 2, 3]. Next, we will illustrate the use of some of these properties to reason about time intervals in solving the Berge Mystery Story.

5. The Berge Mystery Story

Some of you who have read my first book, *Algorithmic Graph Theory and Perfect Graphs*, know the Berge mystery story. For those who don't, here it is:

> Six professors had been to the library on the day that the rare tractate was stolen. Each had entered once, stayed for some time and then left. If two were in the library at the same time, then at least one of them saw the other. Detectives questioned the professors and gathered the following testimony: Abe said that he saw Burt and Eddie in the library; Burt said that he saw Abe and Ida; Charlotte claimed to have seen Desmond and Ida; Desmond said that he saw Abe and Ida; Eddie testified to seeing Burt and Charlotte; Ida said that she saw Charlotte and Eddie. One of the professors lied!! Who was it?

Let's pause for a moment while you try to solve the mystery. Being the interrogator, you begin, by collecting the data from the testimony written in the story: Abe saw Burt and Eddie, Burt saw Abe and Ida, etc. Figure 10(a) shows this data with an arrow pointing from X to Y if X "claims" to have seen Y. Graph theorists will surely start attacking this using graph theory. How can we use it to solve the mystery?

Remember that the story said each professor came into the library, was there for an interval of time, during that interval of time he saw some other people. If he saw somebody, that means their intervals intersected. So that provides some data about the intersection, and we can construct an intersection graph G, as in Figure 10(b). This graph "should be" an interval graph if all the testimony was truthful and complete.

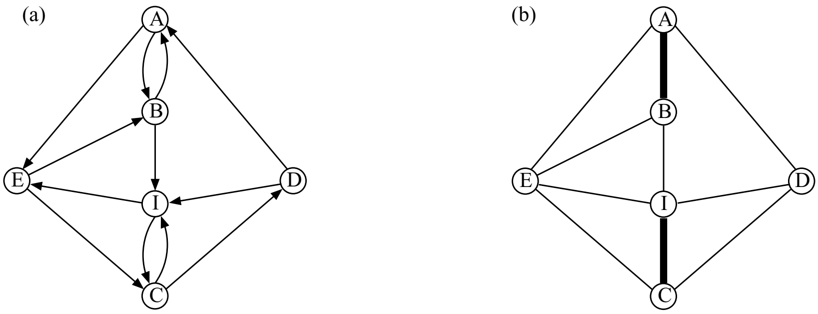

Figure 10. The testimony graphs.

However, we know that there is a lie here. Why? Because looking at the intersection graph G, we see a chordless cycle of length four which is an impossibility. This is supposed to be an interval graph, so we know something is wrong.

Notice in Figure 10(a), that some pairs have arrows going in both directions, for example, Burt saw Abe and Abe saw Burt, and other pairs are just one way. That gives us some further information. Some of the edges in the intersection graph are more confident edges than others. A bold black edge in Figure 10(b) indicates a double arrow in Figure 10(a), and it is pretty confident because B saw A and A saw B, so, if at least one of them is telling the truth, the edge really exists. Similarly, for I and C. But all the one-way arrows are possibly true and possibly false. How shall we argue? Well, if we have a 4-cycle, one of those four professors is the liar. I do not know which one, so I will list all the cycles and see who is common. ABID is a cycle of length 4 without a chord; so is ADIE. There is one more – AECD – that is also a 4-cycle, with no chord. What can we deduce? We can deduce that the liar is one of these on a 4-cycle. That tells us Burt is not a liar. Why? Burt is one of my candidates in the first cycle, but he is not a candidate in the second, so he is telling the truth. The same goes for Ida; she is not down in the third cycle, so she is also telling the truth. Charlotte is not in the first cycle, so she is ok. The same for Eddie, so he is ok. Four out of the six professors are now known to be telling the truth. Now it is only down to Abe and Desmond. What were to happen if Abe is the liar? If Abe is the liar, then ABID still remains a cycle because of the testimony of Burt, who is truthful. That is, suppose Abe is the liar, then Burt, Ida and Desmond would be truth tellers and ABID would still be a chordless cycle, which is a contradiction. Therefore, Abe is not the liar. The only professor left is Desmond. Desmond is the liar.

Was Desmond Stupid or Just Ignorant?

If Desmond had studied algorithmic graph theory, he would have known that his testimony to the police would not hold up. He could have said that he saw everyone, in which case, no matter what the truthful professors said, the graph would be an interval graph. His (false) interval would have simply spanned the whole day, and all the data would be consistent. Of course, the detectives would probably still not believe him.

6. Many Other Families of Intersection Graphs

We have seen a number of applications of interval graphs, and we will see one more a little bit later. However, I now want to talk about other kinds of intersections – not just of intervals – to give an indication of the breadth of research that goes on in this area. There is a mathematician named Victor Klee, who happened to have been Robert Jameson's thesis advisor. In a paper in the American Mathematics Monthly in 1969, Klee wrote a little article that was titled "What are the intersection graphs of arcs in a circle?" [21]. At that point in time, we already had Gilmore and Hoffman's theorem characterizing interval graphs and several other theorems of Lekkerkerker and Boland, Fulkerson and Gross, Ghouila-Houri and Berge (see [1]). Klee came along and said, *Okay, you've got intervals on a line, what about arcs going along a circle?* Figure 11 shows a model of arcs on a circle, together with its intersection graph. It is built in a similar way as an interval graph, except that here you have arcs of a circle instead of intervals of a line. Unlike interval graphs, circular arc intersection graphs may have chordless cycles. Klee wanted to know: Can you find a mathematical characterization for these circular arc graphs?

In fact, I believe that Klee's paper was really an implicit challenge to consider a whole variety of problems on many kinds of intersection graphs. Since then, dozens of researchers have begun investigating intersection graphs of boxes in the plane, paths in a tree, chords of a circle, spheres in 3-space, trapezoids, parallelograms, curves of functions, and many other geometrical and topological bodies. They try to recognize them, color them, find maximum cliques and independent sets in them. (I once heard someone I know mention in a lecture, "A person could make a whole career on algorithms for intersection graphs!" Then I realized, that person was probably me.)

Circular arc graphs have become another important family of graphs. Renu Laskar and I worked on domination problems and circular arc graphs during my second visit to Clemson in 1989, and published a joint paper [14].

On my first visit to Clemson, which was in 1981, I started talking to Robert Jamison about an application that comes up on a tree network, which I will now describe.

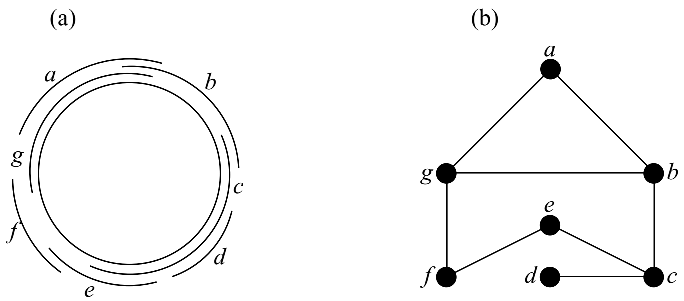

Figure 11. (a) Circular arc representation. (b) Circular arc graph.

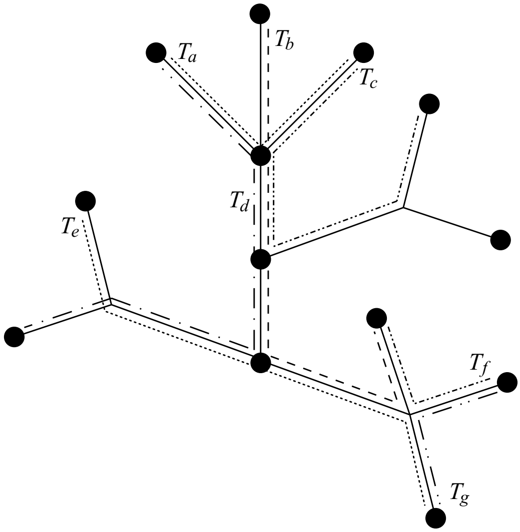

Figure 12. A representation of paths in a tree.

Figure 12 shows a picture of a tree, the black tree, which you may want to think of as a communication network connecting different places. We have pairs of points on the tree that have to be connected with paths – green paths, red paths and purple paths. They must satisfy the property that if two of these paths overlap, even a little bit, this intersection implies that they conflict and we cannot assign the same resource to them at the same time. I am interested in the intersection graph of these paths. Figure 13 gives names to these paths and shows their intersection graph. As before, if two paths intersect, you connect the two numbers by an edge, and if they are disjoint, you do not. Coloring this graph is the same as assigning different colors to these paths; if they intersect, they get different colors. I can interpret each color to mean a time slot when the path has exclusive use of the network. This way there is the red time slot, the purple

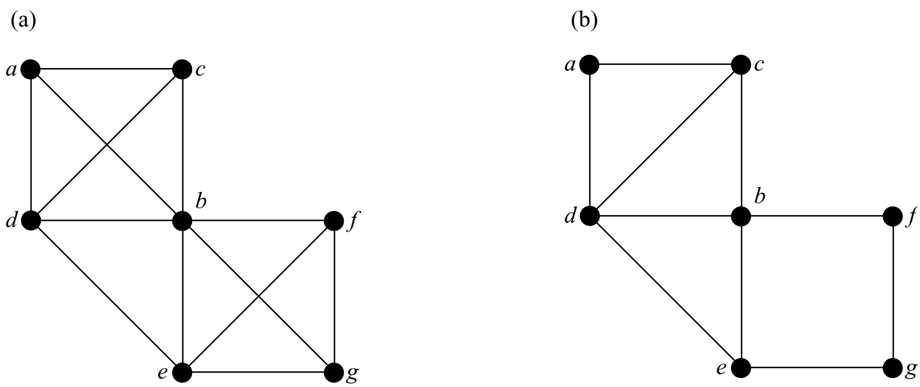

Figure 13. (a) Vertex Intersection Graph (VPT) and (b) Edge Intesection Graph (EPT), both of the paths shown on Figure 12.

Table 1. Six graph problems and their complexity on VPT graphs and EPT graphs.

Graph Problem	VPT graphs	EPT graphs
recognition	polynomial	NP-complete [12]
maximum independent set	polynomial	polynomial [25]
maximum clique	polynomial	polynomial [11]
minimum coloring	polynomial	NP-complete [11]
3/2 approximation coloring	polynomial	polynomial [25]
minimum clique cover	polynomial	NP-complete [11]

time slot, etc. All the red guys can use the network at the same time; all the purple guys can use it together some time later; brown guys use it at yet a different time. We might be interested to find (*i*) a maximum independent set, which would be the largest number of paths to be used simultaneously, or (*ii*) a minimum coloring, which would be a schedule of time periods for all of the paths.

I began investigating the intersection graphs of paths and trees, and immediately had to look at two kinds of intersections – one was sharing a vertex and one was sharing an edge. This gave rise to two classes of graphs, which we call vertex intersection graphs of paths of a tree (VPT graphs) and edge intersection graphs of paths of a tree (EPT graphs), quickly observing that they are different classes – VPT graphs are chordal and perfect, the EPT graphs are not.

After discussing this at Clemson, Robert and I began working together on EPT graphs, a collaboration of several years resulting in our first two joint papers [11, 12]. We showed a number of mathematical results for EPT graphs, and proved several computational complexity results. Looking at the algorithmic problems – recognition, maximum independent set, maximum clique, minimum coloring, 3/2 approximation (Shannon) coloring, and minimum clique cover – all six problems are polynomial for vertex intersection (VPT graphs), but have a real mixture of complexities for edge intersection (EPT graphs), see Table 1. More recent extensions of EPT graphs have been presented in [15, 20].

There are still other intersection problems you could look at on trees. Here is an interesting theorem that some may know. If we consider an intersection graph of subtrees of a tree, not just paths but arbitrary subtrees, there is a well known characterization attributed to Buneman, Gavril, and Wallace discovered independently by each of them in the early 1970's, see [1].

Theorem 2 *A graph G is the vertex intersection graph of subtrees of a tree if and only if it is a chordal graph.*

Here is another Clemson connection. If you were to look at subtrees not of just any old tree, but of a special tree, namely, a star, you would get the following theorem of Fred McMorris and Doug Shier [23] from 1983.

Table 2. Graph classes involving trees.

Type of Interaction	Objects	Host	Graph Class
vertex intersection	subtrees	tree	chordal graphs
vertex intersection	subtrees	star	split graphs
edge intersection	subtrees	star	all graphs
vertex intersection	paths	path	interval graphs
vertex intersection	paths	tree	path graphs or VPT graphs
edge intersection	paths	tree	EPT graphs
containment	intervals	line	permutation graphs
containment	paths	tree	? (open question)
containment	subtrees	star	comparability graphs

Theorem 3 *A graph G is a vertex intersection graph of distinct subtrees of a star if and only if both G and its complement $\overline{G}$ are chordal.*

Notice how well the two theorems go together: If the host tree is any tree, you get chordal, and if it is a star, you get chordal ∩ co-chordal, which are also known as split graphs. In the case of edge intersection, the chordal graphs are again precisely the edge intersection graphs of subtrees of a tree, however, *every possible graph can be represented as the edge intersection graph of subtrees of a star*. Table 2 summarizes various intersection families on trees. Some of them may be recognizable to you; for those that are not, a full treatment can be found in Chapter 11 of [4].

7. Tolerance Graphs

The grandfather of all intersection graph families is the family of interval graphs. Where do we go next? One direction has been to measure the size of the intersection and define a new class called the interval tolerance graphs, first introduced by Golumbic and Monma [16] in 1982. It is also the topic of the new book [4] by Golumbic and Trenk. We also go into trapezoid graphs and other kinds of intersection graphs.

Even though I am not going to be discussing tolerance graphs in detail, I will briefly state what they are and in what directions of research they have taken us. In particular, there is one related class of graphs (NeST) that I will mention since it, too, has a Clemson connection.

An undirected graph $G = (V, E)$ is a *tolerance graph* if there exists a collection $\mathcal{I} = \{I_v\}_{v \in V}$ of closed intervals on the real line and an assignment of positive numbers $t = \{t_v\}_{v \in V}$ such that

$$vw \in E \Leftrightarrow |I_v \cap I_w| \geq \min\{t_v, t_w\}.$$

Here $|I_u|$ denotes the length of the interval I_u. The positive number t_v is called the *tolerance* of v, and the pair $\langle \mathcal{I}, t \rangle$ is called an *interval tolerance representation* of G. Notice that interval graphs are just a special case of tolerance graphs, where each tolerance t_v equals some sufficiently small $\epsilon > 0$. A tolerance graph is said to be *bounded* if it has a tolerance representation in which $t_v \leq |I_v|$ for all $v \in V$.

The definition of tolerance graphs was motivated by the idea that a small, or "tolerable" amount of overlap, between two intervals may be ignored, and hence not produce an edge. Since a tolerance is associated to each interval, we put an edge between a pair of vertices when at least one of them (the one with the smaller tolerance) is "bothered" by the size of the intersection.

Let's look again at the scheduling problem in Figure 5. In that example, the chief university officer of classroom scheduling needs four rooms to assign to the six lectures. But what would happen if she had only three rooms available? In that case, would one of the lectures c, d, e or f have to be cancelled? Probably so. However, suppose some of the professors were a bit more tolerant, then an assignment might be possible.

Consider, in our example, if the tolerances (in minutes) were:

$$t_a = 10, t_b = 5, t_c = 65, t_d = 10, t_e = 20, t_f = 60.$$

Then according to the definition, lectures c and f would no longer conflict, since $|I_c \cap I_f| \leq 60 = \min\{t_c, t_f\}$. Notice, however, that lectures e and f remain in conflict, since Professor e is too intolerant to ignore the intersection. The tolerance graph for these values would therefore only erase the edge cf in Figure 6, but this is enough to admit a 3-coloring.

Tolerance graphs generalize both interval graphs and another family known as permutation graphs. Golumbic and Monma [16] proved in 1982 that every bounded tolerance graph is a cocomparability graph, and Golumbic, Monma and Trotter [17] later showed in 1984 that tolerance graphs are perfect and are contained in the class of weakly chordal graphs. Coloring bounded tolerance graphs in polynomial time is an immediate consequence of their being cocomparability graphs. Narasimhan and Manber [24] used this fact in 1992 (as a subroutine) to find the chromatic number of any (unbounded) tolerance graph in polynomial time, but not the coloring itself. Then, in 2002, Golumbic and Siani [19] gave an $O(qn + n \log n)$ algorithm for coloring a tolerance graph, given the tolerance representation with q vertices having unbounded tolerance. For details and all the references, see Golumbic and Trenk [4]. The complexity of recognizing tolerance graphs and bounded tolerance graphs remain open questions.

A several "variations on the theme of tolerance" in graphs have been defined and studied over the past years. By substituting a different "host" set instead of the real line, and then specifying the type of subsets of that host to consider instead of intervals, along with a way to measure the size of the intersection of two subsets, we obtain other classes of tolerance-type graphs, such as neighborhood subtree tolerance (NeST) graphs (see Section 8 below), tolerance graphs of paths on a tree or tolerance competition graphs. By changing the function *min* for a different binary function ϕ (for example, *max, sum, product*, etc.), we obtain a class that will be called ϕ*-tolerance* graphs. By replacing the measure of the length of an interval by some other measure μ of the intersection of the two subsets (for example, cardinality in the case of discrete sets, or number of branching nodes or maximum degree in the case of subtrees of trees), we could obtain yet other variations of tolerance graphs. When we restrict the tolerances to be 1 or ∞, we obtain the class of *interval probe graphs*. By allowing a separate leftside tolerance and rightside tolerance for each interval, various bitolerance graph models can be obtained. For example, Langley [22] in 1993 showed that *the bounded bitolerance graphs are equivalent to the class of trapezoid graphs*. Directed graph analogues to several of these models have also been defined and studied. For further study of tolerance graphs and related topics, we refer the reader to Golumbic and Trenk [4].

8. Neighborhood Subtree Tolerance (NeST) Graphs

On my third visit to Clemson, also in 1989, Lin Dearing told me about a class of graphs that generalized tolerance graphs, called neighborhood subtree tolerance (NeST) graphs. This generalization consists of representing each vertex $v \in V(G)$ of a graph G by a subtree T_v of a (host) tree embedded in the plane, where each subtree T_v has a center c_v and a radius r_v and consists of all points of the host tree that are within a distance of r_v from c_v. The size of a neighborhood subtree is twice its radius, or its diameter. The size of the intersection of two subtrees T_u and T_v is the Euclidean length of a longest path in the intersection, namely, the diameter of the subtree $T_u \cap T_v$. Bibelnieks and Dearing [9] investigated various properties of NeST graphs. They proved that bounded NeST graphs are equivalent to proper NeST graphs, and a number of other results. You can see their result in one of the boxes in Figure 14.

I bring this to your attention because it is typical of the research done on the relationships between graph classes. We have all these classes, some of which are arranged in a containment hierarchy and others are equivalent, shown in the same box of the figure. Figure 14 is an example of an incomplete hierarchy since some of the relationships are unknown. Interval graphs and trees are the low families on this diagram. They are contained in the classes above them, which are in turn contained in the ones above them, etc. The fact that NeST is contained in weakly chordal graphs is another Clemson result from [9].

You see in Figure 14 a number of question marks. Those are the open questions still to be answered. So as long as the relationships between several graph classes in the hierarchy are not known yet (this is page 222 of [4]), we remain challenged as researchers.

9. Interval Probe Graphs

Lastly, I want to tell you about another class of graphs called the interval probe graphs. They came about from studying interval graphs, where some of the adjacency information was missing. This is a topic that is of recent interest, motivated by computational biology applications. The definition of an interval probe graph is a graph

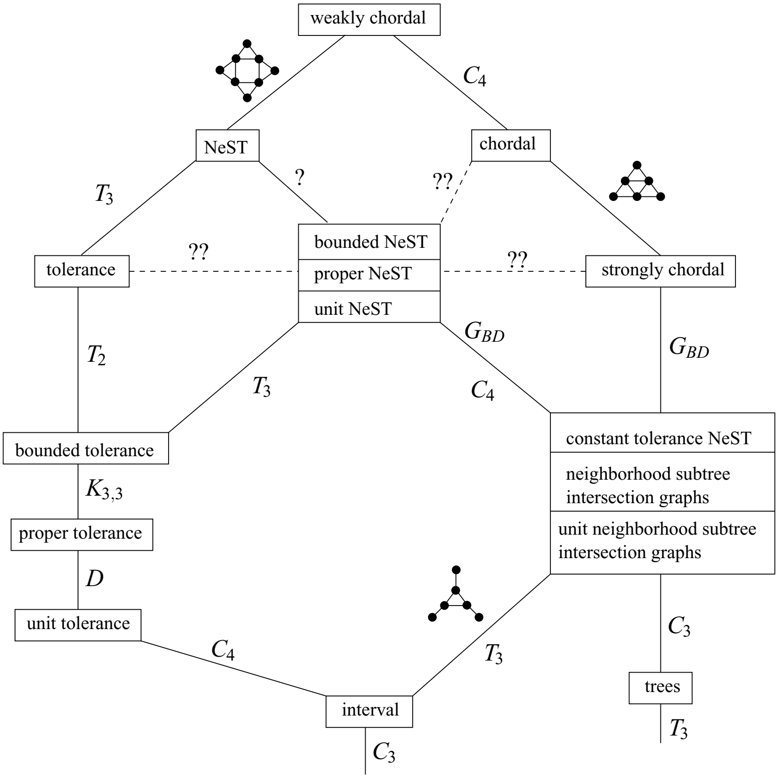

Figure 14. The NeST Hierarchy (reprinted from [4]).

whose vertices are partitioned into two sets: the probes P and non-probes N, where N is an independent set, and there must exist a(n interval) completion, by adding some extra edges between nodes in N so that this augmented graph is an interval graph. The names probe and non-probe come from the biological application. Partitioned into these two sets, the edges between pairs of P nodes and between P and N nodes are totally known, but there is nothing known about the connections between pairs of N nodes. Is it possible to fill in some of these missing edges in order to get an interval representation?

That is the mathematical formulation of it. You can ask, “What kinds of graphs do you have in this class?” Figure 15(a) shows an example of an interval probe graph and a representation for it; The black vertices are probes and the white vertices are non–probes. Figure 15(b) gives an example of a graph that is not interval probe, no matter how the vertices may be partitioned. I will let you prove this on your own, but if you get stuck, then you can find many examples and proofs in Chapter 4 of the Tolerance Graphs book [4].

I will tell you a little bit about how this problem comes about in the study of genetic DNA sequences. Biologists want to be able to know the whole structure of the DNA

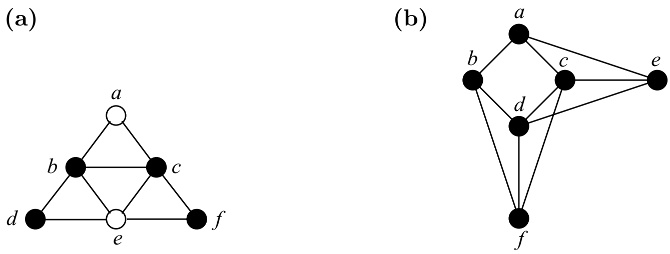

Figure 15. An example of an interval probe graph and a non-interval probe graph.

of some gene, which can be regarded as a long string of about a million letters. The Human Genome Project was to be able to identify the sequence of those strings in people. The biologists worked together with other scientists and mathematicians and computer scientists and so on, and here is one method that they use. They take the gene and they place it in a beaker with an enzyme, and then dump it out on the floor where it breaks into a bunch of pieces. These fragments are called gene fragments. Now they will take the same gene and put in a different enzyme, stir it and shake it and then dump it on the floor where it breaks up in a different way. They do it again and again with a bunch of different enzymes.

Now we can think of the problem as reconstructing several puzzles, each one having different pieces, but giving the same completed string. One could argue, "Ah, I have a string that says ABBABBABBA and someone else has a similar string BABBACCADDA and they would actually overlap nicely." By cooperating maybe we could put the puzzle back together, by recombining overlapping fragments to find the correct ordering, that is, if we are able to do all this pattern matching.

Imagine, instead, there is some experiment where you take this little piece of fragment and you can actually test somehow magically, (however a biologist tests magically), how it intersects with the other fragments. This gives you some intersection data. Those that are tested will be the probes. In the interval probe model, for every probe fragment we test, we know exactly whom he intersects, and for the unlucky fragments that we do not test, we know nothing regarding the overlap information between them. They are the non-probes. This is the intersection data that we get from the biologist. Now the interval probe question is: can we somehow fill-in the missing data between the pairs of non-probes so that we can get a representation consistent with that data?

Here is a slightly different version of the same problem – played as a recognition game. Doug has an interval graph H whose edges are a secret known only to him. A volunteer from the audience chooses a subset N of vertices, and Doug draws you a graph G by secretly erasing from H all the edges between pairs of vertices in N, making N into an independent set. My game #1 is, if we give you the graph G and the independent set N, can you fill-in some edges between pairs from N and rebuild an interval graph (not necessarily H)?

This problem can be shown to be solvable in time proportional to n^2 in a method that was found by Julie Johnson and Jerry Spinrad, published in SODA 2001. The following year there was a faster algorithm by Ross McConnell and Jerry Spinrad that solved the problem in time $O(m \log n)$, published in SODA 2002. Here n and m are the number of vertices and edges, respectively.

There is a second version of the game, which I call the unpartitioned version: this time we give you G, but we do not tell you which vertices are in N. My game #2 requires both choosing an appropriate independent set and filling in edges to complete it to an interval graph. So far, the complexity of this problem is still an open question. That would be recognizing unpartitioned interval probe graphs.

10. The Interval Graph Sandwich Problem

Interval problems with missing edges, in fact, are much closer to the problem Seymour Benzer originally addressed. He asked the question of reconstructing an interval model even when the probe data was only partially known. Back then, they could not answer his question, so instead he asked the 'simpler' interval graph question: "Suppose I had *all* of the intersection data, then can you test consistency and give me an interval representation?" It was not until much later, in 1993, that Ron Shamir and I gave an answer to the computational complexity of Benzer's real question.

You are given a partially specified graph, i.e., among all possible pairs of vertices, some of the pairs are definitely edges, some of them are definitely non-edges, and the remaining are unknown. Can you fill-in some of the unknowns, so that the result will be an interval graph? This problem we call the *interval sandwich problem* and it is a computationally hard problem, being NP-complete [18].

For further reading on sandwich problems, see [13], Chapter 4 of [4] and its references.

11. Conclusion

The goal of this talk has been to give you a feeling for the area of algorithmic graph theory, how it is relevant to applied mathematics and computer science, what applications it can solve, and why people do research in this area.

In the world of mathematics, sometimes I feel like a dweller, a permanent resident; at other times as a visitor or a tourist. As a mathematical resident, I am familiar with my surroundings. I do not get lost in proofs. I know how to get around. Yet, sometimes as a dweller, you can become jaded, lose track of what things are important as things become too routine. This is why I like different applications that stimulate different kinds of problems. The mathematical tourist, on the other hand, may get lost and may not know the formal language, but for him everything is new and exciting and interesting. I hope that my lecture today has given both the mathematical resident and the mathematical tourist some insight into the excitement and enjoyment of doing applied research in graph theory and algorithms.

12. Acknowledgements

I am honored to have been invited to deliver the 2003 Andrew F. Sobczyk Memorial Lecture. This has been my sixth visit to Clemson University, and I would like to thank my hosts from the Mathematics Department for their generous hospitality. Clemson University is a leader in the field of discrete mathematics, and building and maintaining such leadership is no small achievement. It takes hard work and a lot of excellent research. I salute those who have made this possible, and who have influenced my career as well: Renu Laskar, Robert Jamison, P.M. (Lin) Dearing, Doug Shier, Joel Brawley and Steve Hedetniemi. I would also like to thank Sara Kaufman and Guy Wolfovitz from the Univerity of Haifa for assisting with the written version of this lecture.

Further Reading

[1] M.C. Golumbic, "*Algorithmic Graph Theory and Perfect Graphs*", Academic Press, New York, 1980. Second edition: *Annals of Discrete Mathematics* 57, Elsevier, Amsterdam, 2004.
This has now become the classic introduction to the field. It conveys the message that intersection graph models are a necessary and important tool for solving real-world problems for a large variety of application areas, and on the mathematical side, it has provided rich soil for deep theoretical results in graph theory. In short, it remains a stepping stone from which the reader may embark on one of many fascinating research trails. The second edition of *Algorithmic Graph Theory and Perfect Graphs* includes a new chapter called *Epilogue 2004* which surveys much of the new research directions from the Second Generation. Its intention is to whet the appetite.
Seven other books stand out as the most important works covering advanced research in this area. They are the following, and are a must for any graph theory library.

[2] A. Brandstädt, V.B. Le and J.P. Spinrad, "*Graph Classes: A Survey*", SIAM, Philadelphia, 1999,
This is an extensive and invaluable compendium of the current status of complexity and mathematical results on hundreds of families of graphs. It is comprehensive with respect to definitions and theorems, citing over 1100 references.

[3] P.C. Fishburn, "*Interval Orders and Interval Graphs: A Study of Partially Ordered Sets*", John Wiley & Sons, New York, 1985.
Gives a comprehensive look at the research on this class of ordered sets.

[4] M.C. Golumbic and A.N. Trenk, "*Tolerance Graphs*", Cambridge University Press, 2004.
This is the youngest addition to the perfect graph bookshelf. It contains the first thorough study of tolerance graphs and tolerance orders, and includes proofs of the major results which have not appeared before in books.

[5] N.V.R. Mahadev and U.N. Peled, “*Threshold Graphs and Related Topics*”, North-Holland, 1995.
A thorough and extensive treatment of all research done in the past years on threshold graphs (Chapter 10 of Golumbic [1]), threshold dimension and orders, and a dozen new concepts which have emerged.

[6] T.A. McKee and F.R. McMorris, “*Topics in Intersection Graph Theory*”, SIAM, Philadelphia, 1999.
A focused monograph on structural properties, presenting definitions, major theorems with proofs and many applications.

[7] F.S. Roberts, “*Discrete Mathematical Models, with Applications to Social, Biological, and Environmental Problems*”, Prentice-Hall, Engelwood Cliffs, New Jersey, 1976.
This is the classic book on many applications of intersection graphs and other discrete models.

[8] W.T. Trotter, “*Combinatorics and Partially Ordered Sets*”, Johns Hopkins, Baltimore, 1992.
Covers new directions of investigation and goes far beyond just dimension problems on ordered sets.

Additional References

[9] E. Bibelnieks and P.M. Dearing, Neighborhood subtree tolerance graphs, *Discrete Applied Math.* 43: 13–26 (1993).

[10] P.C. Gilmore and A.J. Hoffman, A characterization of comparability graphs and of interval graphs. *Canad. J. Math.* 16: 539–548 (1964).

[11] M.C. Golumbic and R.E. Jamison, The edge intersection graphs of paths in a tree, *J. Combinatorial Theory, Series B* 38: 8–22 (1985).

[12] M.C. Golumbic and R.E. Jamison, Edge and vertex intersection of paths in a tree, *Discrete Math.* 55: 151–159 (1985).

[13] M.C. Golumbic and H. Kaplan and R. Shamir, Graph sandwich problems, *J. of Algorithms* 19: 449–473 (1995).

[14] M.C. Golumbic and R.C. Laskar, Irredundancy in circular arc graphs, *Discrete Applied Math.* 44: 79–89 (1993).

[15] M.C. Golumbic, M. Lipshteyn and M. Stern, The k-edge intersection graphs of paths in a tree, *Congressus Numerantium* (2004), to appear.

[16] M.C. Golumbic and C.L. Monma, A generalization of interval graphs with tolerances”, *Congressus Numerantium* 35: 321–331 (1982).

[17] M.C. Golumbic and C.L. Monma and W.T. Trotter, Tolerance graphs, *Discrete Applied Math.* 9: 157–170 (1984).

[18] M.C. Golumbic and R. Shamir, Complexity and algorithms for reasoning about time: A graph theoretic approach, *J. Assoc. Comput. Mach.* 40: 1108–1133 (1993).

[19] M.C. Golumbic and A. Siani, Coloring algorithms and tolerance graphs: reasoning and scheduling with interval constraints, *Lecture Notes in Computer Science* 2385, Springer-Verlag, (2002) pp. 196–207.

[20] R.E. Jamison and H.M. Mulder, Constant tolerance representations of graphs in trees, *Congressus Numerantium* 143: 175–192 (2000).

[21] V. Klee, What are the intersection graphs of arcs in a circle? *American Math. Monthly* 76: 810–813 (1969).

[22] L. Langley, *Interval tolerance orders and dimension*, Ph.D. Thesis, Dartmouth College", June, (1993).

[23] F.R. McMorris and D.R. Shier, Representing chordal graphs on $K_{1,n}$, *Comment. Math. Univ. Carolin.* 24: 489–494 (1983).

[24] G. Narasimhan and R. Manber, Stability number and chromatic number of tolerance graphs, *Discrete Applied Math.* 36: 47–56 (1992).

[25] R.E. Tarjan, Decomposition by clique separators, *Discrete Math.* 55: 221–232 (1985).