CAMBRIDGE

Cambridge University Press
978-0-521-82758-4 - Tolerance Graphs
Martin Charles Golumbic and Ann N. Trenk
Excerpt
More information

Chapter 1

Introduction

1.1 Background and motivation

Our mathematical adventure begins with a collection of intervals on the real line. The intervals may have come from an application, for example, they could represent the durations of a set of events on a time line, or fragments of DNA on the genome, or sectors of consecutive elements of a linearly ordered set. Some of the intervals may intersect one another, and others may be disjoint. No matter what they may represent, intervals are familiar to us as mathematical entities. There are many relationships between these intervals that we could study. In this book, we deal mostly with intersection.

When two intervals intersect, we might interpret this positively as their having something important in common, like an opportunity to share information. For example, if each interval represented the time period during which a group of school children would be visiting a science museum, then two groups whose intervals intersect could participate in a joint activity. We might then ask, how many times would we need to flash the new Artificial Bolt of Lightning so that each group would get to see it? Or we might interpret intersection negatively as having a major conflict, like competing for a resource that cannot be shared. For example, in a one-television household, when a parent wants to watch the News and at the same time a teenager wants to watch an old movie on a different channel, we have a temporal conflict.

In graph theory, the family of *interval graphs* was introduced to study such problems of intersecting intervals on the line. In this model, each vertex v in a graph $G = (V, E)$ is associated with an interval I_v, and two vertices are connected by an edge in G if their associated intervals have nonempty intersection. Formally, $uv \in E(G) \iff I_u \cap I_v \neq \emptyset$, for all $u, v \in V(G)$. The graph G is called an interval graph.

© Cambridge University Press www.cambridge.org

CAMBRIDGE

Cambridge University Press
978-0-521-82758-4 - Tolerance Graphs
Martin Charles Golumbic and Ann N. Trenk
Excerpt
More information

In our museum example, there is a well-defined minimum number α of how many times that the lightning must be flashed, and it is easy to calculate the number α and an optimal schedule for the flashes. Well, at least it is "easy" for the authors since we have been teaching students about interval graphs for a long time. But it is also "easy" in a computational sense since there are well-known linear time algorithms to do this.

But what do you do if the electricity requirements allow only $\alpha - 2$ flashings? Either some of the groups will be disappointed, or they will have to reschedule the time of their visit. Similarly, in our television example, when one spouse wants to watch a game show and the other spouse wants a basketball game, it is fair game to assume that a compromise is needed.

In this book, we study the class of tolerance graphs, which are a generalization of interval graphs. Tolerance graphs are constructed from intersecting intervals in a manner similar to interval graphs, but putting an edge between two vertices depends on measuring the size of the intersection of their two intervals before declaring that an edge exists. Informally, if both intervals are willing to "tolerate" or ignore the intersection, then no edge is added between their vertices in the graph.

Tolerance graphs were introduced by Golumbic and Monma (1982) to generalize some of the well-known applications associated with interval graphs. Their original motivation was the need to solve scheduling problems in which resources such as rooms, vehicles, support personnel, etc. may be needed on an exclusive basis, but where a measure of flexibility or tolerance would allow for sharing or relinquishing the resource if a solution is not otherwise possible. Let's look at simple example.

A motivating example

On a typical morning, six parliamentary or corporate meetings are to convene according to a fixed schedule, where meeting m_i is scheduled for the time interval $I_i = [a_i, b_i]$. Each meeting must be assigned a meeting room. Let us consider the example:

$$I_1 = [8{:}00\text{–}9{:}45], \quad I_2 = [9{:}00\text{–}11{:}30], \quad I_3 = [8{:}30\text{–}11{:}15],$$
$$I_4 = [10{:}00\text{–}11{:}00], \quad I_5 = [10{:}15\text{–}12{:}00], \quad I_6 = [10{:}45\text{–}12{:}30].$$

In our example, meeting m_1 could use the same room as either m_4 or m_5 or m_6 since its time interval I_1 does not intersect with the time intervals I_4, I_5 or I_6. Being very strict with these intervals, we see that at 10:50 five rooms are needed simultaneously (see Figure 1.1). But suppose there are only four meeting rooms! Should we cancel one of the meetings? Probably not. Rather,

© Cambridge University Press www.cambridge.org

CAMBRIDGE

Cambridge University Press
978-0-521-82758-4 - Tolerance Graphs
Martin Charles Golumbic and Ann N. Trenk
Excerpt
More information

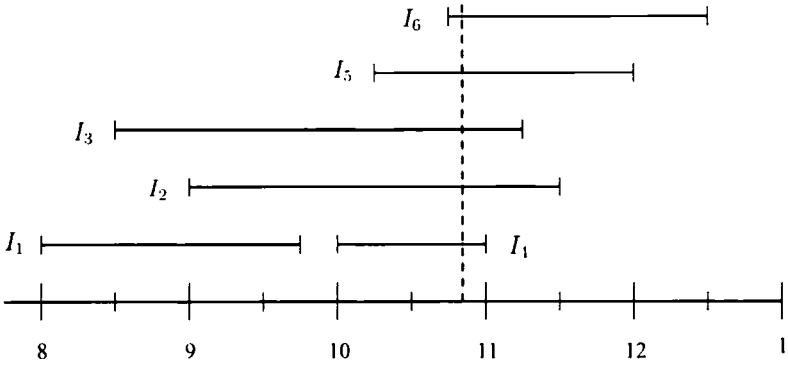


Figure 1.1. A motivating example.

we should try to identify some flexibility in these time constraints which may allow us to find an acceptable assignment of rooms.

The tolerance graph model, which we will formally define below, provides a mechanism for associating a numerical tolerance to each meeting to indicate the degree of its flexibility in allowing some intersection with other intervals. In this way, it may be possible to give an assignment of rooms to all the meetings by sharing the room for a short period or by moving the start or finish time. In our example, if both I_4 and I_6 were willing to tolerate an overlap of more than 15 minutes, then there would be a four room solution.

Resource assignment problems of this nature arise in many contexts: motorcycles for delivering express mail (or pizza), nurses for operating rooms, waterfront space for picnics, ovens for warming a caterer's dishes, etc. In a real world situation, some meetings or deliveries may indeed have strict deadlines which must be met, while others may be more flexible. By taking these tolerances into account, solutions can often be found which would otherwise not exist under the strict constraints. There would be a great benefit to having algorithmic methods for automatically resolving such conflicts.

This example, and the discussion on intersecting intervals, briefly motivates the topic of our book. The volume and scope of research in this area has expanded significantly both from the mathematical and algorithmic points of view. Many special families of graphs and ordered sets will be encountered along the way. Each will depend on the specific tolerance model being discussed.

In this chapter, we provide the formal definition of a tolerance graph and give some elementary properties. We also give a brief review of many of the important families of graphs which are related in some way to tolerance graphs.

© Cambridge University Press www.cambridge.org

CAMBRIDGE

Cambridge University Press
978-0-521-82758-4 - Tolerance Graphs
Martin Charles Golumbic and Ann N. Trenk
Excerpt
More information

1.2 Intersection graphs and interval graphs

Let $\mathcal{F}$ be a collection of sets. The *intersection graph* of $\mathcal{F}$ is the graph obtained by assigning a distinct vertex to each set in $\mathcal{F}$ and joining two vertices by an edge precisely when their corresponding sets have a nonempty intersection. When the types of sets allowed in $\mathcal{F}$ is limited, interesting classes of graphs result.

Most important to us will be the *interval graphs* which arise when the sets in $\mathcal{F}$ are intervals in the real line, that is, a graph $G = (V, E)$ is an *interval graph* if each vertex $v \in V$ can be assigned a real interval I_v so that $xy \in E \iff I_x \cap I_y \neq \emptyset$. The set of intervals $\{I_v \mid v \in V\}$ is an *interval graph representation* of G.

Interval graphs are important for their applications to scheduling problems, microbiology, and VLSI circuit design. In our previous motivating example (Figure 1.1), the intervals represented fixed time slots for a set of meetings which needed to be assigned rooms. The interval graph for this example is shown in Figure 1.2. Finding a consistent assignment of rooms can be viewed as a *coloring problem* on the interval graph, where the meeting rooms are the colors and adjacent vertices must be assigned different colors. There are efficient algorithms for coloring the vertices of an interval graph using a minimum number of colors (Golumbic, 1980). In our example, there cannot be a solution with four rooms since the interval graph has a *clique* (or *complete subgraph*) of size 5. Indeed, the only subsets that could be colored by the same color in this example are $\{1, 4\}$ or $\{1, 5\}$ or $\{1, 6\}$. A *stable set* (or *independent set*) is a subset of vertices no two of which are connected by an edge. Here there is no stable set larger than size 2.

In this book, we also consider other families of intersection graphs, such as *trapezoid graphs* and *parallelogram graphs* which are intersection graphs of trapezoids (resp. parallelograms) having two of their sides on two fixed

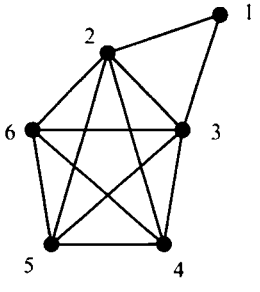


Figure 1.2. The interval graph for our motivating example.

© Cambridge University Press www.cambridge.org

CAMBRIDGE

Cambridge University Press
978-0-521-82758-4 - Tolerance Graphs
Martin Charles Golumbic and Ann N. Trenk
Excerpt
More information

parallel lines. Later in this chapter, we discuss permutation graphs which can be interpreted as intersection graphs of line segments in a matching diagram. Also, in Chapter 11, we present a variety of intersection graphs involving subtrees and paths in trees.

All of these families of intersection graphs satisfy the *hereditary property*, namely, if a graph $G = (V, E)$ is the intersection graph of a certain type (e.g., intervals, trapezoids, etc.), then every *induced subgraph* G_X of G is also an intersection graph of that same type, where $V(G_X) = X \subseteq V(G)$ and $E(G_X) = \{uv \in E(G) \mid u, v \in X\}$.

1.3 Tolerance graphs: definitions and examples

A graph $G = (V, E)$ is a *tolerance graph* if each vertex $v \in V$ can be assigned a closed interval I_v and a tolerance $t_v \in \mathbf{R}^+$ so that $xy \in E$ if and only if $|I_x \cap I_y| \geq \min\{t_x, t_y\}$. Such a collection $\langle \mathcal{I}, t \rangle$ of intervals and tolerances is called a *tolerance representation* where $\mathcal{I} = \{I_x \mid x \in V\}$ and $t = \{t_x \mid x \in V\}$. If graph G has a tolerance representation with $t_v \leq |I_v|$ for all $v \in V$, then G is called a *bounded tolerance graph* and the representation is called a *bounded tolerance representation.*

Consider once again our motivating example. If each of the tolerances were to be 5 minutes, then the tolerance graph would be the same as the interval graph since all of the nonempty intersections are longer than 5 minutes. However, if the tolerances of I_4 and I_6 were 20 minutes (or anything greater than 15 minutes) and each of the others 5 minutes, then the tolerance graph would have no edge between v_4 and v_6, as shown in Figure 1.3. In this case, the vertices of the tolerance graph can be colored using 4 colors, which provides a consistent assignment of meeting rooms.

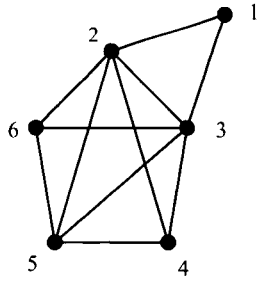


Figure 1.3. The tolerance graph for our motivating example, where I_4 and I_6 have a tolerance of 20 minutes and each of the others 5 minutes.

© Cambridge University Press www.cambridge.org

CAMBRIDGE

Cambridge University Press
978-0-521-82758-4 - Tolerance Graphs
Martin Charles Golumbic and Ann N. Trenk
Excerpt
More information

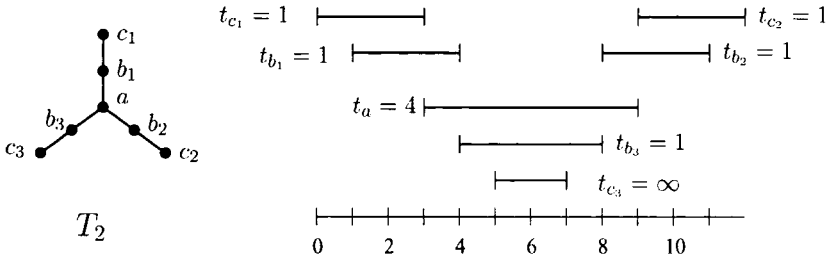


Figure 1.4. The graph T_2 and a tolerance representation of it.

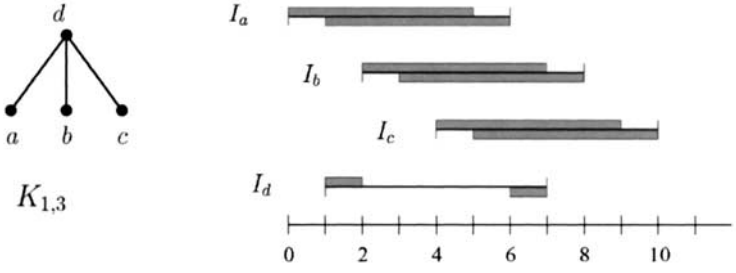


Figure 1.5. The graph $K_{1,3}$ and a bounded tolerance representation of it.

We next look at some additional examples of tolerance graphs. For tolerance representations, we draw the interval assigned to each vertex and list its tolerance next to it, as in the representation of the tree T_2 in Figure 1.4. Notice that the vertex c_3 has infinite tolerance. In fact, any tolerance greater than $|I_{c_3}|$ would work equally well. In Chapter 3, we will see that every tolerance representation of T_2 must have some vertex whose tolerance is greater than its interval length.

For bounded tolerance representations, the tolerance assigned to vertex v is at most the length of the interval $I_v = [L(v), R(v)]$ assigned to v. In this case, we sometimes find it clearer to show the tolerances visually using shading. We shade in the interval from $L(v)$ to $L(v) + t_v$ above I_v and shade in the interval from $R(v) - t_v$ to $R(v)$ below I_v. Figure 1.5 shows a bounded tolerance representation of the graph $K_{1,3}$ in which tolerances are indicated by shading.

The exercises at the end of this chapter will help the reader to become familiar with the concepts presented. Our formal study of tolerance graphs begins in Chapter 2. The remainder of this chapter is devoted to definitions, background and classical results.

© Cambridge University Press www.cambridge.org

CAMBRIDGE

Cambridge University Press
978-0-521-82758-4 - Tolerance Graphs
Martin Charles Golumbic and Ann N. Trenk
Excerpt
More information

1.4 Chordal graphs, comparability graphs, and properties of interval graphs

1.4.1 Chordal graphs and split graphs

A graph G is a *chordal graph* if every cycle of length greater than or equal to 4 has a *chord*, that is, an edge connecting two vertices that are not consecutive on the cycle. For example, the graph in Figure 1.3 is chordal, and the edge (3,5) is a chord of the cycle [3,4,5,6,3]. The chordal graphs are a well known classical family of graphs, and they appear in many interesting applications including relational databases, matrix theory, statistics, and biology. In the literature, chordal graphs are also called *triangulated graphs* (Berge, 1973; Golumbic, 1980) or *rigid circuit graphs* (Roberts, 1976). The family of chordal graphs includes all interval graphs but does not include all tolerance graphs.

There are several interesting characterizations of chordal graphs which we will now review. We present their equivalence below in Theorem 1.1.

A vertex v is called *simplicial* if its *neighborhood* $\mathcal{N}(v) = \{w \in V(G) \mid \forall w \in E(G)\}$ is a clique, that is, every pair of neighbors of v is connected by an edge of the graph. Let $\sigma = [v_1, v_2, \ldots, v_n]$ be an ordering of the vertices $V(G)$, and let $G_i = G_{\{v_i, \ldots, v_n\}}$ denote the subgraph remaining after deleting $\{v_1, \ldots, v_{i-1}\}$ from G. We define σ to be a *perfect elimination ordering (peo)* if v_i is a simplicial vertex in the graph G_i, for all i. For example, two possible perfect elimination orderings for the graph in Figure 1.3 are [4,6,5,1,3,2] and [1,4,3,5,2,6], but [3,4,5,6,1,2] is *not* a perfect elimination ordering for this graph.

A *maximum cardinality search (MCS)* of a graph G is done as follows: Initially all vertices are unnumbered and have counters set to zero. Choose an unnumbered vertex with largest counter, give it the next number, and add 1 to the counters of each of its neighbors. Continue doing this until all the vertices have been numbered. Suppose that the vertices were numbered in this way: $[x_1, x_2, \ldots, x_n]$, then we will call it a *maximum cardinality search ordering*. Such an MCS ordering for the graph in Figure 1.3 is [1,2,3,4,5,6].

Theorem 1.1. *The following conditions are equivalent:*

(i) *G is a chordal graph.*
(ii) *G has a perfect elimination ordering.*
(iii) *The reversal $[x_n, \ldots, x_2, x_1]$ of any MCS ordering of G is a perfect elimination ordering.*
(iv) *G is the intersection graph of a family of subtrees of a tree.*

The equivalence (i)⇔(ii) is due to Dirac; (i)⇔(iii) to Tarjan; (i)⇔(iv) independently to Buneman, Gavril and Walters; see Brandstädt, Le, and Spinrad

© Cambridge University Press www.cambridge.org

CAMBRIDGE

Cambridge University Press
978-0-521-82758-4 - Tolerance Graphs
Martin Charles Golumbic and Ann N. Trenk
Excerpt
More information

(1999), Golumbic (1980), Golumbic (1984), McKee and McMorris (1999) for a proof of this theorem and for additional references.

Both conditions (ii) and (iii) suggest algorithms for recognizing chordal graphs. Using (ii), one would repeatedly look for and eliminate a simplicial vertex, breaking ties arbitrarily, until either all vertices are eliminated (success) or no simplicial vertex can be found (failure). This greedy method is correct since once a vertex becomes simplicial, it remains simplicial in any induced subgraph. Using (iii), one would carry out a maximum cardinality search while testing its reversal to verify that it is a perfect elimination ordering (success) or is not a peo (failure). The latter method gives a more efficient algorithm, having complexity $O(n + e)$ for a graph with n vertices and e edges, see Berry, Blair, and Heggernes (2002), Golumbic (1980), Golumbic (1984) and Tarjan and Yannakakis (1984).

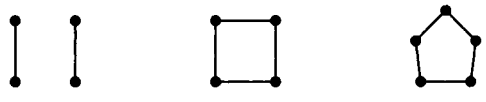
There are also efficient, polynomial time algorithms for finding a minimum coloring, maximum clique, maximum stable set, or a minimum clique cover of a chordal graph. In general, these graph problems are NP-complete, which means that chordal graphs are indeed a very special family of graphs.

We conclude this section by defining and characterizing the class of split graphs. A graph $G = (V, E)$ is called a *split graph* if its vertex set can be partitioned $V = X \cup Y$ into a stable set X and a clique Y. The graph in Figure 1.3 is a split graph with partition $X = \{1, 4\}$ and $Y = \{2, 3, 5, 6\}$.

The *complement* $\overline{G}$ of G is the graph where $V(\overline{G}) = V(G)$ and $E(\overline{G}) = \{xy \mid xy \notin E(G), x \neq y\}$. Since a stable set in G is a clique in the complement $\overline{G}$, and vice versa, G is a split graph if and only if $\overline{G}$ is a split graph. Földes and Hammer (1977) have given the following characterization of split graphs.

Theorem 1.2. *The following conditions are equivalent:*

(i) *G is a split graph.*
(ii) *G and $\overline{G}$ are chordal graphs.*
(iii) *G contains none of the graphs $2K_2$, C_4, C_5 as an induced subgraph, (see Figure 1.6).*



$2K_2$ C_4 C_5

Figure 1.6. The forbidden subgraphs characterizing split graphs.

© Cambridge University Press www.cambridge.org

CAMBRIDGE

Cambridge University Press
978-0-521-82758-4 - Tolerance Graphs
Martin Charles Golumbic and Ann N. Trenk
Excerpt
More information

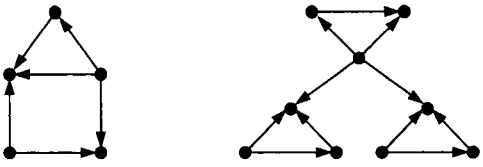


Figure 1.7. Some transitive orientations.

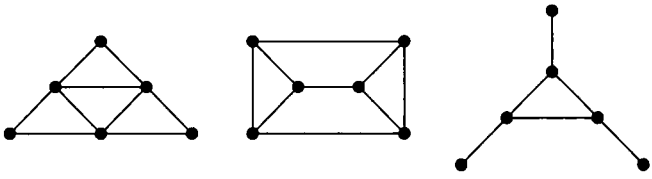


Figure 1.8. Some graphs which are *not* transitively orientable.

For a proof of this theorem and for further reading on chordal graphs and split graphs, see Brandstädt, Le, and Spinrad (1999), Golumbic (1980) and McKee and McMorris (1999). We will see split graphs again in Chapter 11.

1.4.2 Comparability graphs and transitive orientations

A *transitive orientation* F of graph $G = (V, E)$ is an assignment of a direction, or orientation, to each edge in E such that if $xy \in F$ and $yz \in F$ then $xz \in F$. A graph is called a *comparability graph* if it has a transitive orientation. For example, the even length chordless cycles $C_{2k} (k \geq 2)$ are comparability graphs, but the odd length chordless cycles C_5, C_7, etc. are not comparability graphs. Comparability graphs are also known as *transitively orientable (TRO)* graphs. Additional examples of comparability graphs and their transitive orientations can be found in Figure 1.7. Figure 1.8 shows several graphs which have no transitive orientation. Gallai (1967) gave a list of forbidden subgraphs that characterizes the class of comparability graphs, (see also Duchet, 1984). The name "comparability" graph comes from the observation that relation F is a strict partial ordering of V whose comparability relation is precisely E. We will discuss more about ordered sets in Section 1.5.

Comparability graphs can be recognized, and a transitive orientation can be produced, using the following well known greedy method. (a) Choose an orientation of an arbitrarily chosen edge. (b) Propagate all other orientations forced by this and all subsequently oriented edges (usually called the *implication class*). If at some point an edge is forced in both opposite directions, exit with failure. (c) When no other orientations are forced, add the oriented edges to F

© Cambridge University Press www.cambridge.org

CAMBRIDGE

Cambridge University Press
978-0-521-82758-4 - Tolerance Graphs
Martin Charles Golumbic and Ann N. Trenk
Excerpt
More information

and remove them from E. If the graph still has some edges, repeat this sequence of steps. When this algorithm finishes, F will be a transitive orientation. The reader unfamiliar with this topic is referred to Golumbic (1980, 1984). This method can be implemented to run in $O(n \cdot e)$ time for a graph with n vertices and e edges, or by a more careful counting $O(\Sigma_{v \in V} d_v^2)$, where d_v is the degree of v. (The *degree* of a vertex v is the number of edges that have v as an endpoint, that is, $d_v = |\mathcal{N}(v)|$.)

Asymptotically faster algorithms for recognizing comparability graphs, which use a technique called modular decomposition, have been given in McConnell and Spinrad (1999). In that paper, the authors show how to find an orientation F of an arbitrary graph G such that F is a TRO of G if and only if G is a comparability graph. This is very good if there is other information guaranteeing that G is a comparability graph. However, this alone does not recognize comparability graphs, since the algorithm simply produces an orientation which is *not* transitive when G is *not* a comparability graph. Hence, to complete it to a recognition algorithm, one must test F to determine whether it is transitive. The complexity of their method uses $O(n + e)$ time to produce F and $O(n^\alpha)$ to test whether F is transitive, where $O(n^\alpha)$ is the complexity to perform transitive closure or $n \times n$ matrix multiplication (currently $n^{2.376}$).

The complements of comparability graphs, called *cocomparability graphs*, are of particular interest in this book since, as we will see in the next chapter, all bounded tolerance graphs are cocomparability graphs. Cocomparability graphs also have a characterization as the intersection graphs of function diagrams Golumbic, Rotem, and Urrutia (1983), which we present in Section 1.6.

1.4.3 Interval graphs

We defined interval graphs in Section 1.2 as being the intersection graphs of intervals on a line. Interval graphs have several important characterizations which we review here. One of these is the equivalence of interval graphs and the graphs that are both chordal and cocomparability. A second relates to the notion of an asteroidal triple of vertices which we now define.

Three vertices $v_1, v_2, v_3 \in V(G)$ form an *asteroidal triple* (AT) of G if, for all permutations i, j, k of $\{1,2,3\}$, there is a path from v_i to v_j which avoids using any vertex in the *closed neighborhood* $\mathcal{N}[v_k] = \{v_k \cup \mathcal{N}(v_k)$. An easy way to verify this for v_k is to delete $\mathcal{N}[v_k]$ and test whether v_i and v_j remain in the same connected component of $G - \mathcal{N}[v_k]$. It also follows from the definition that the three vertices of an asteroidal triple are pairwise nonadjacent. For example, $\{c_1, c_2, c_3\}$ is an asteroidal triple in the tree T_2 in Figure 1.4.

© Cambridge University Press www.cambridge.org