Interval Graphs

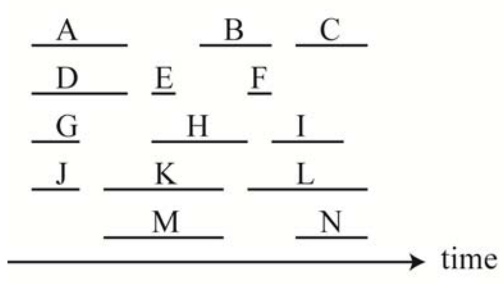
Complement to Chapter 3, “The Case of the Missing Files”

Let’s consider a set of tasks, each one with a precise start time and end time. Suppose we ask employees to carry out these tasks, but no employee can do two tasks at once. To model the situation, we simply need to create a graph in which each task is represented by a vertex and an edge links two vertices when the corresponding tasks overlap in time. The graph thus constructed is called an interval graph.

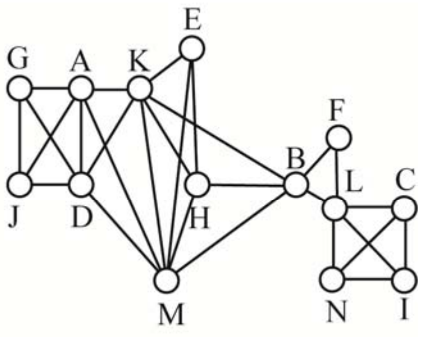
Definition

An “interval graph” is the graph showing intersecting intervals on a line. So, we associate a set of intervals $E=\{E_1,\ldots,E_n\}$ on a line with the interval graph $G=(V,E)$, where $V=\{1,\ldots,n\}$ and two vertices, x and y, are linked by an edge if and only if $E_x \cap E_y \neq \emptyset$.

Example



14 tasks to perform



Corresponding interval graph

In the case of the theft at the canton archives, Costello summarizes the suspects’ statements using the following table, in which an “x” in a box means that the person on that line says he saw the person in the column.

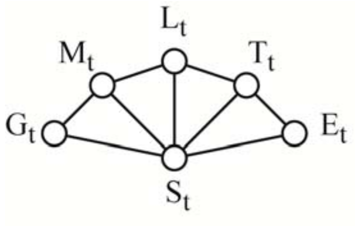
Thursday, April 9, 2009

	T	B	E	S	L	G	M
T			x	x	x		
B							
E	x			x			
S	x		x		x	x	x
L	x			x			x
G				x			x
M				x	x	x	

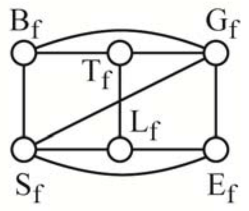
Friday, April 10, 2009

	T	B	E	S	L	G	M
T		x			x	x	
B	x			x		x	
E				x	x	x	
S		x	x		x	x	
L	x		x	x			
G	x	x	x	x			
M							

If each suspect only visited the canton archives a maximum of once per day, they each spent one time interval there. We can therefore associate an interval graph with these statements by creating one vertex per suspect and linking two vertices when the intervals pertaining to them overlap in time, or in other words, when the two suspects encountered one another (which means there’s an “x” on Costello’s table). Based on this, Manori draws the following graphs, which in principle correspond with two interval graphs, one for the Thursday intervals and one for the Friday intervals.



Thursday graph



Friday graph

To determine who's guilty, Manori demonstrates, among other things, that the right-hand graph is not an interval graph. We can ask, then, how to go about recognizing an interval graph. The following property is true for all interval graphs, and is due to the fact that time is linear and not circular.

Property

If G is an interval graph, then it contains no cycles with more than three edges and no shortcuts.

The concept of "no shortcuts" in this property is key. For example, the left-hand graph above is an interval graph even though there is a cycle with a length of 6 made up of the six edges forming the contour of the graph. The cycle effectively includes shortcuts. For example, instead of going from S_t to G_t and then M_t, we can take the shortcut represented by the edge linking S_t to M_t.

Manori showed his colleagues that the Friday graph contains two squares, meaning two cycles with four edges and no shortcuts. This means it is not an interval graph, which means that one of the suspects visited the archives more than once on Friday.

Interval graphs have other interesting properties as well. The property set out below will help us easily determine the minimum number of employees we need to perform all the tasks in the problem described at the beginning of this document.

Definitions

A vertex is "simplicial" if its neighbours (meaning the vertices to which it is linked by an edge) are all linked to one another by edges.

A "perfect elimination scheme" in a graph with n vertices is an ordering $v_1,\ldots,v_n$ of the vertices such that v_i is simplicial in the graph that contains only the vertices $v_i,\ldots,v_n$.

In other words, if $v_1,\ldots,v_n$ is a perfect elimination scheme, then all the neighbours of v_1 are linked to one another. Also, if we delete vertex v_1 from the graph, then all the neighbours of v_2 are related to one another. And so forth, meaning if we delete vertices $v_1,\ldots,v_{i-1}$ from the graph, all the neighbours of v_i are linked to one another.

Property

A graph is an interval graph if, and only if, a perfect elimination scheme exists.

It is very easy to determine a perfect elimination scheme in an interval graph. We simply need to choose a simplicial vertex (meaning one whose neighbours are all linked to one another) and name it v_1. We then delete this vertex from the graph and look for a simplicial vertex in the remaining graph, and name it v_2. We continue doing this until the remaining graph is empty.

Let's come back to the original problem, assigning tasks to employees, each task with a precise start and end time. This problem comes down to colouring the vertices of an interval graph under the constraint that two vertices linked by an edge cannot be of the same colour. In other words, each colour corresponds to an employee, and we cannot assign the same employee to two tasks that overlap in time. If we are looking to minimize the number of employees necessary for performing all these tasks, that's the equivalent of determining a colouring of the vertices that uses the minimum number of colours.

This colouring can be easily obtained with the following technique, in which we place the colours in a certain order.

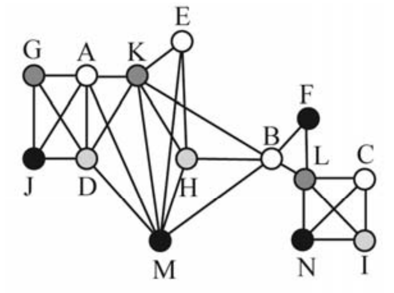
Colouring the vertices in an interval graph using the minimum possible number of colours:

1. Set out a perfect elimination scheme $v_1,\ldots,v_n$.
2. Colour the vertices one after another in reverse order, $v_n,\ldots,v_1$, always choosing the first available colour.

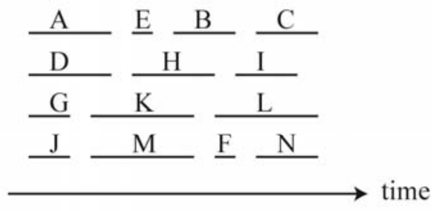
In the example of the 14 tasks, we can see for example that vertex C is simplicial, and so we can choose it for v_1. We could also have chosen G or J or I, etc. However, we could not, for instance, have chosen vertex L, because its neighbours, F and B, are not linked to its other neighbours, C, I and N. In second position, we can place, for instance, vertex I, and then vertex N in third. Vertex L then becomes simplicial because its neighbours in the remaining graph obtained by deleting C, I and N are F and B, which are linked together. So we can place vertex L in fourth position. We continue in this way to obtain, for instance, the following order:

C, I, N, L, F, B, E, H, K, M, A D, G, J

We can now colour the graph. Let's suppose that the colours are ordered from darkest to lightest. We start by assigning black to J, then dark grey to G, light grey to D and white to A. The M vertex, which is the next to be coloured, can be coloured in dark grey or black. Because black comes first, we choose it for vertex M. We continue in this way until we have coloured the whole graph, which is shown in the following figure. This colouring requires four colours, which means we need to hire four employees to carry out the 14 tasks. The assignment of these tasks to the four employees is represented to the right of the graph. Each line corresponds to one colour and so to one employee. For example, the employee represented by the colour white performs tasks A, B, C and E, since those vertices are coloured in white.



Coloring in reverse order of the perfect elimination scheme C,I,N,L,F,B,E,H,K,M,A,D,G,J



The tasks can be performed by four employees. Each line corresponds to one employee's work

In the example above, we are led to set up a partition of the vertices in an interval graph into the minimum possible number of subsets such that there is no edge linking two vertices in the same subset. Some situations can be modelled using an interval graph in which you need to determine a partition of its vertices into the minimum possible number of subsets such that, this time, there are no edges missing between any of the vertices in the same subset. In terms of colouring, this time we want to colour the vertices such that the vertices of the same colour are each linked to all the others by edges. Here are three examples of situations like this.

Example 1

A pharmaceuticals warehouse needs to keep a stock of products $p_1,\ldots,p_n$. Each p_i product has minimum m_i and maximum M_i storage temperatures. We want to determine the minimum possible number of refrigerated containers that will be needed to stock all the products.

To solve this problem, let's build an interval graph in which the vertices represent the products and two products, p_i and p_j, are linked by an edge if the intervals $[m_i,M_i]$ and $[m_j,M_j]$ have an intersection that is not empty. We want to create groups of products such that all the products in a given group are linked by an edge to all the others (meaning there is a storage temperature that's appropriate for all the products in that group).

Example 2

A number of people, $P_1,\ldots,P_n$, have decided to spend their vacation at the Bellevue Hotel. Person P_i will be there from day A_i until day B_i. The hotel director trusts his manager to make sure everything goes smoothly for the vacationers, and so rarely visits the hotel. In fact, the director only shows up at the hotel from time to time in order to meet the guests and to try and build their customer loyalty. He wants to meet his n guests while minimizing the number of days he needs to be at the Bellevue, since he is a very busy man and because the hotel is located in a distant area of the country that is hard to access.

To determine the minimum number of visits the director must make in order to meet all his guests, we can create an interval graph in which the vertices are guests and two guests, P_i and P_j, are linked by an edge if the intervals $[A_i,B_i]$ and $[A_j,B_j]$ have an intersection that is not empty. We want to create groups such that all the guests in a given group are linked to all the others by an edge (which means there is a date on which all those guests are present at the hotel and the director can therefore make their acquaintance).

Example 3

A theatre-lover wants to attend n plays, P_1, …, P_n, which will be performed in Montréal in the next little while. Each play, P_i, will run for a continuous period, from day B_i until day E_i. When several plays are performing on the same day, they are at different times so that people can attend them both on the same day. The theatre-lover lives an hour's drive away from Montréal and absolutely wants to sleep at home every night. So if he attends plays on two consecutive days, he'll have to drive to Montréal twice. What is the minimum number of times he can travel to Montréal and still see all the plays?

To solve this problem, we can create an interval graph in which the vertices are plays and two plays, P_i and P_j, are linked by an edge if the intervals $[B_i,E_i]$ and $[B_j,E_j]$ have an intersection that is not empty. We want to create groups such that the plays in a given group are all linked to one another by an edge (which means there is a date on which the theatre-lover could see all of them).

To solve these three problems, we need to partition the vertices in an interval graph into a minimum number of groups such that the vertices in a given group are all linked by an edge. To do this, we can work as follows. We start by ordering the vertices in ascending value from the bottom point of their interval. For the first example, we order the products by ascending m_i; for the second example, it's the A_i that need to be in ascending order, and for the third example it's the B_i.

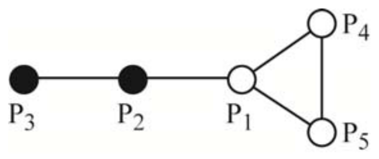
We go through the vertices in this order. We place the first vertex in a first group and the rule for the others is as follows: if the vertex being considered can be added to the existing group, we add it to the group; if not, we close the group and place the vertex in question into a new group.

As an illustration, let's consider example 3 with the *n*=5 following plays:

i	1	2	3	4	5
B_i	June 26	June 20	June 22	July 17	July 19
E_i	July 22	June 24	July 15	July 23	July 20

The order of the plays is therefore 2, 3, 1, 4, 5. So we place play 2 in a first group. Play 3 can be part of that group because it has a date in common with play 2. Play 1 cannot be part of this group because it starts after the end of play 2. So we create a new group for play 1. Plays 4 and 5 are then added to the same group as play 1, because they have July 19 and 20 in common. This way, the theatre-lover can come to Montréal on June 22, 23 or 24 to see plays 2 and 3, and on July 19 or 20 to attend plays 1, 4 and 5.

Here is the interval graph associated with this problem, with the vertex colouring using two colours such that the vertices of a given colour (meaning in a given group) are all linked together with an edge.


P_4
P_3
P_2
P_1
P_5