Network Optimisation

Social network analysis

Six degrees of Separation

Six degrees of separation was a an idea put forward by the Hungarian writer Frigyes Karinthy in 1929. The idea is that everyone is 'linked' to every other person in the world by, on average, six other people. This is an example of the small world property in complex networks.

This idea was popularized by a play of the same name, then later again, by a game invented by some US university students called six degrees of Kevin Bacon

The game is played by choosing an actor and linking them to Kevin Bacon through actors that have acted in the same films.

Kevin Bacon

A few good men

Jack Nicholson

Anger management

Adam Sandler

50 first dates

Drew Barrymore

Scream

David Arquette

Social network analysis

More generally, in social network analysis, nodes (also referred to as vertices or points) represent people.

Two nodes are connected by an edge if they interact with each other in some prescribed way.

Kevin Bacon was used in this example because he has high "centrality". In graph theory and network analysis, centrality refers to indicators which identify the most important nodes within a graph. Applications include identifying the most influential person(s) in a social network, key infrastructure nodes in the Internet or urban networks, and super spreaders of disease [Wikipedia].

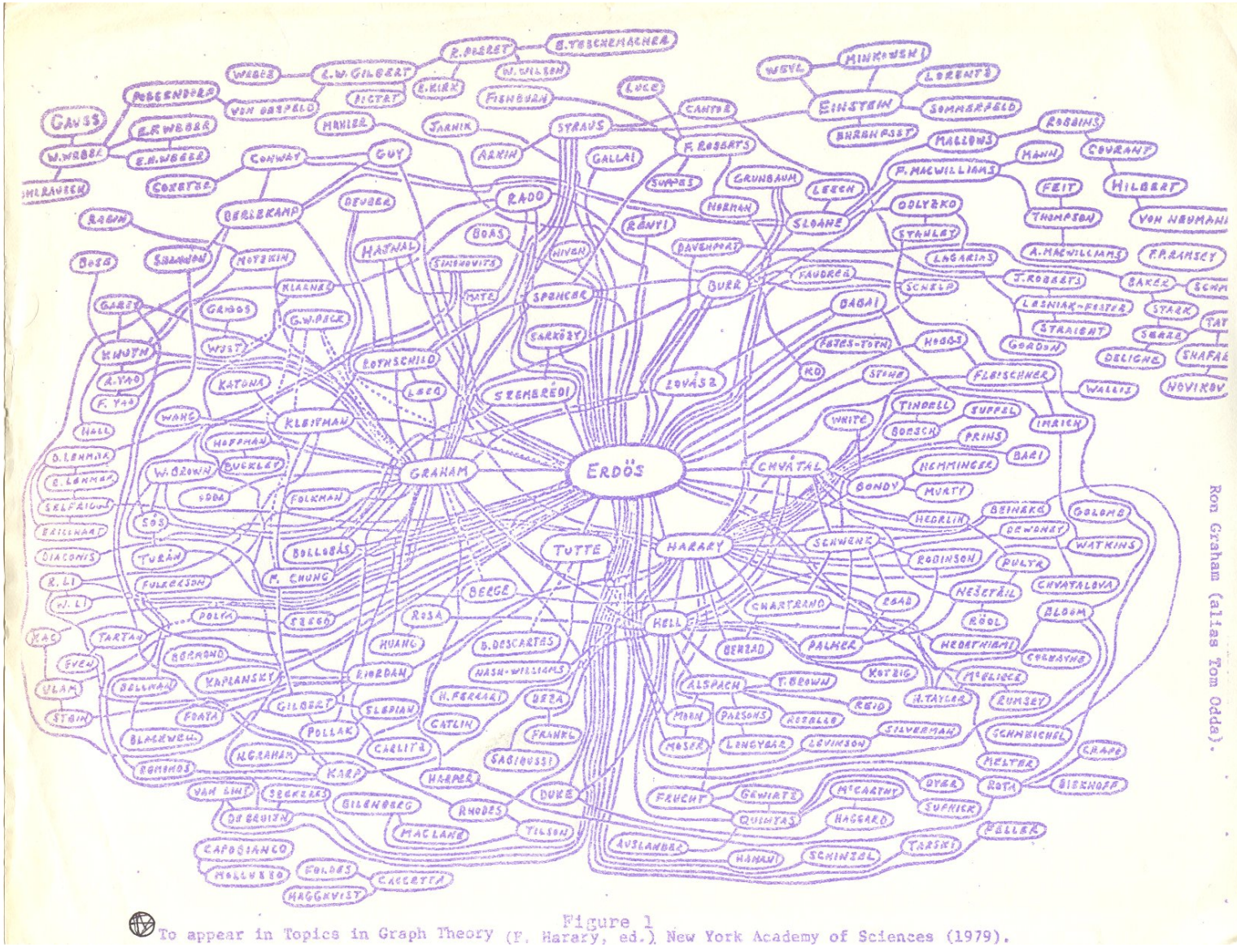


Figure 1

To appear in Topics in Graph Theory (F. Harary, ed.) New York Academy of Sciences (1979).

Another example, from the mathematical community, is Erdös number

Social network analysis

There are many ways to define node importance or centrality:

A) Degree centrality - nodes with many incident edges

B) Closeness centrality - nodes that are "close" (few hops away) to most other nodes

C) Betweenness centrality - nodes that frequently occur on shortest paths

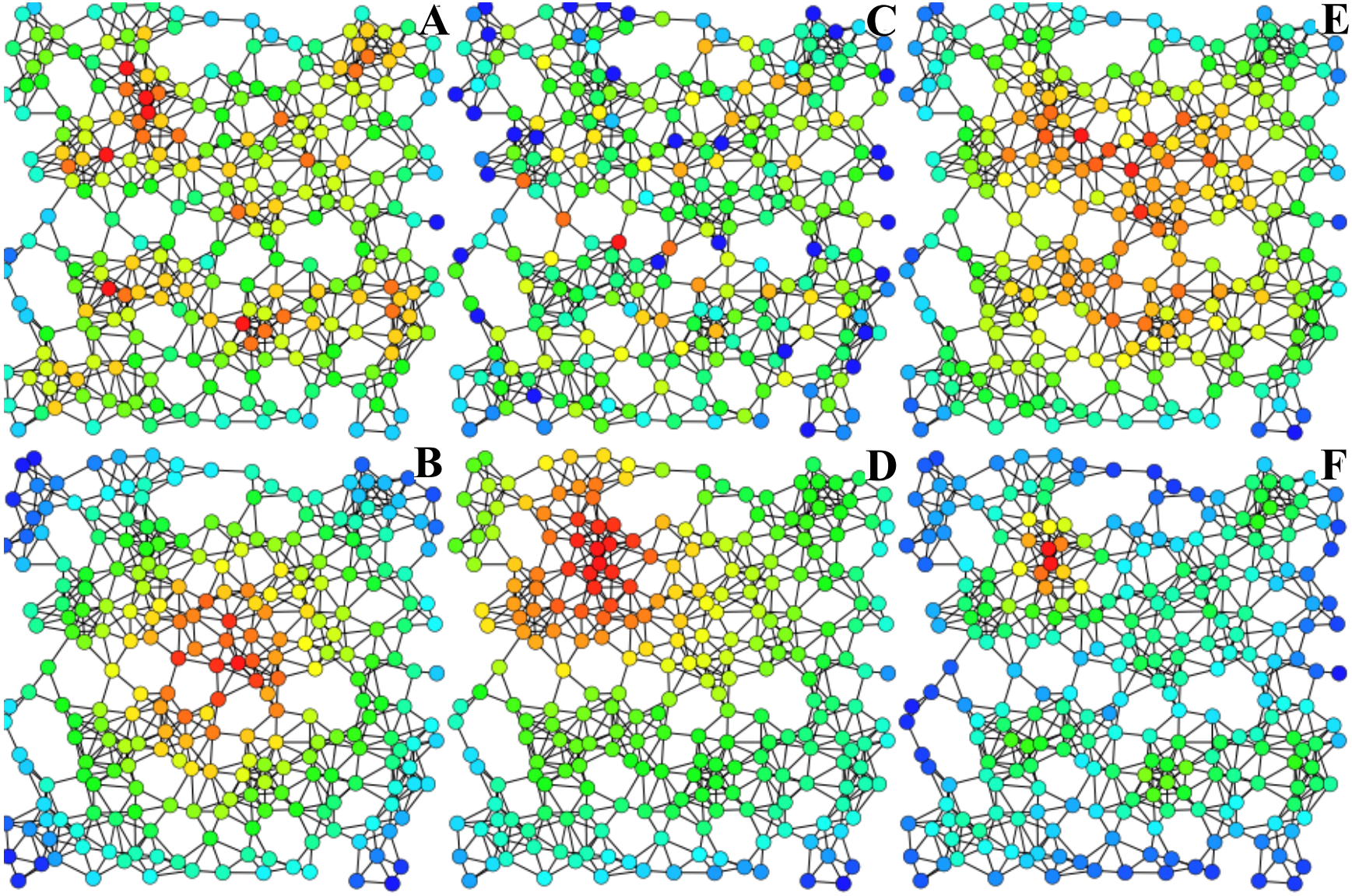
D) Eigenvector centrality - measure of "influence" of a node in a network. Eg. Google PageRank: "PageRank works by counting the number and quality of links to a page to determine a rough estimate of how important the website is. The underlying assumption is that more important websites are likely to receive more links from other websites"

Social network analysis

E) Katz centrality - measures influence by taking into account the total number of walks (not just shortest paths) between pairs of nodes

F) Alpha centrality - adaptation of eigenvector centrality with the addition that nodes are imbued with importance from external sources

etc., etc.



Hue (from blue=0 to red=max) shows the node importance for each of the above measures

Betweenness centrality

Betweenness centrality is based on shortest paths. Intuitively the idea is that if the shortest paths between all pairs of nodes in a connected network contain a certain node v more times that any other node, then v is the most important node. The formula reads

$$B(v) = \sum_{s \neq t \neq v} \frac{\sigma_{st}(v)}{\sigma_{st}}$$

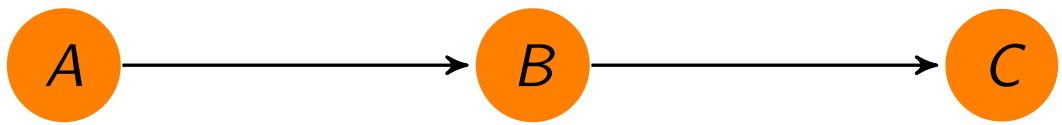
Here the sum is over all pairs of distinct nodes s, t of the network, excluding v.

$\sigma_{st}(v)$ is the number of shortest paths from s to t which pass through v.

σ_{st} is the number of shortest paths from s to t.

A simple example

Compute the betweenness centrality of each of the nodes in the following network:



Solution

Due to symmetry we only need to compute the betweenness of nodes A and B.

We have

$$B(A) = \frac{\sigma_{BC}(A)}{\sigma_{BC}}, \qquad B(B) = \frac{\sigma_{AC}(B)}{\sigma_{AC}}.$$

The shortest path from B to C is the edge BC, which does not pass through A. Hence $\sigma_{BC}(A) = 0$.

On the other hand, the shortest path from A to C is ABC, which does include B, and so $\sigma_{AC} = \sigma_{AC}(A) = 1$.

Hence

$$B(A) = 0, \qquad B(B) = 1, \qquad B(C) = 0.$$

For large networks we need an algorithm to find all shortest paths.

Edge-betweenness centrality and dendograms

As well as betweenness centrality for vertices, the same quantity can be defined for edges,

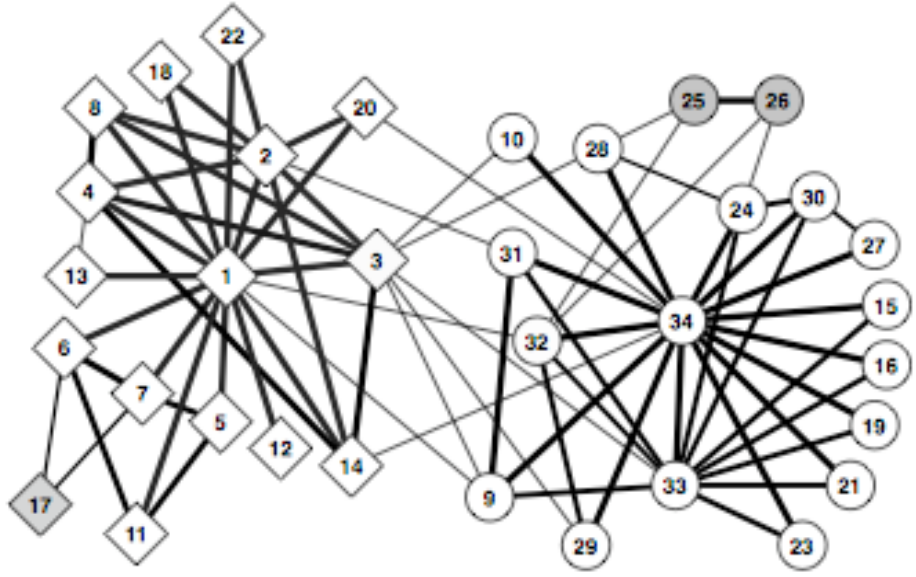
$$B(e) = \sum_{s \neq t} \frac{\sigma_{st}(e)}{\sigma_{st}},$$

where the sum is over all pairs of distinct nodes s, t. Here $\sigma_{st}(e)$ is the number of shortest paths between s and t that contain edge e. Edge-betweenness can be used to construct dendrograms.

Edge-betweenness centrality and dendograms

Girvan-Newman Algorithm - dendogram construction

1. Calculate the edge betweenness of every edge in the network.
2. Remove the edge or edges with the highest betweenness score, and collect the connected components on distinct branches of the tree.
3. Recalculate betweenness scores on each of the resulting connected components, and repeat Step 2.


22
18
8
20
2
4
10
28
25
26
24
30
27
13
1
3
31
15
34
6
32
16
7
5
12
14
19
17
9
33
21
11
29
23


34
1
5
9
7
11
17
2
8
14
3
4
18
20
22
13
12
25
26
6
31
29
32
10
15
16
19
21
23
24
27
28
30
33

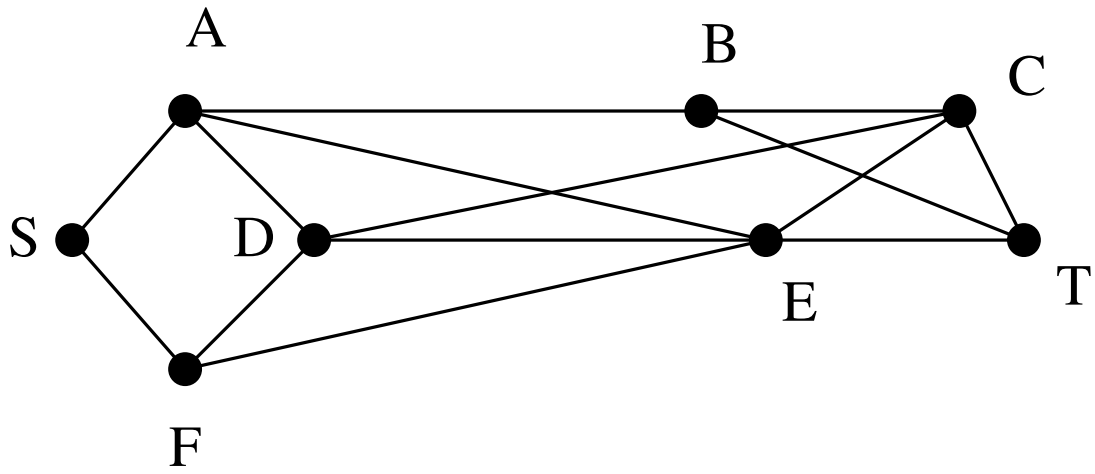
Finding shortest paths

In this section we will look at algorithms for finding shortest paths between given nodes in a network.

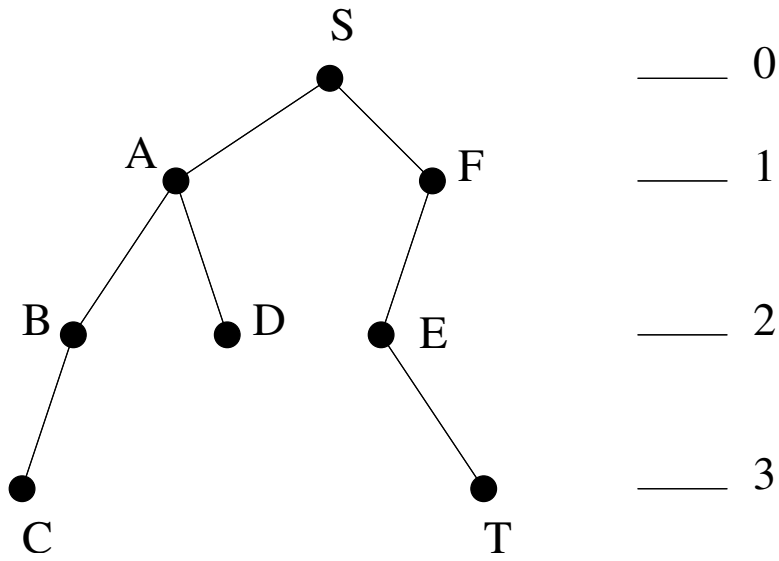
Algorithm - Breadth-First Search (BFS):
Given nodes S, T find a path (if it exists) from S to T which uses the least number of edges.

1. Label vertex S with 0. Set $i = 0$. We think of i as height from S
2. Find all unlabelled vertices in G which are adjacent to vertices labelled i. If there are no such vertices then T is not connected to S by a path. If there are such vertices, label them $i + 1$.
3. If T is labelled, go to Step 4. If not, increase i to $i + 1$ and go to Step 2.
4. The length of the shortest path from S to T is $i + 1$. Stop.

Example



A tree diagram of shortest paths



By backtracking from the tree diagram, we read off that one shortest path is $TEFS$.

Next we would like to calculate the total number of shortest paths σ_{ST} from S to T.

Recursive approach for the number of shortest paths

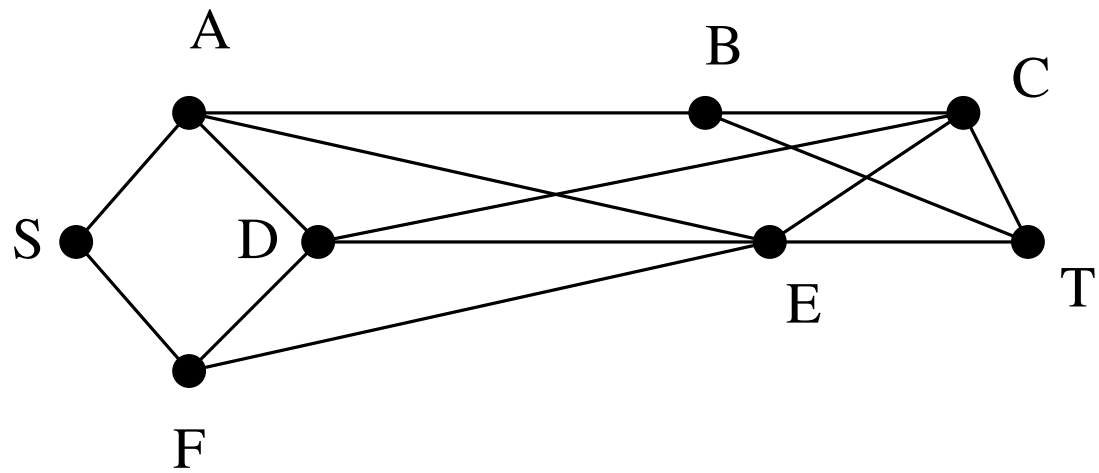
Algorithm

Given nodes S, T find the number of shortest paths from S to T.

1. Set $i := h(T)$, the height of T from S, and let $\mu(T|T) := 1$. All other vertices v for which $h(v) = h(T)$ are assigned $\mu(v|T) := 0$.
2. For each vertex v which satisfies $h(v) = i - 1$ compute

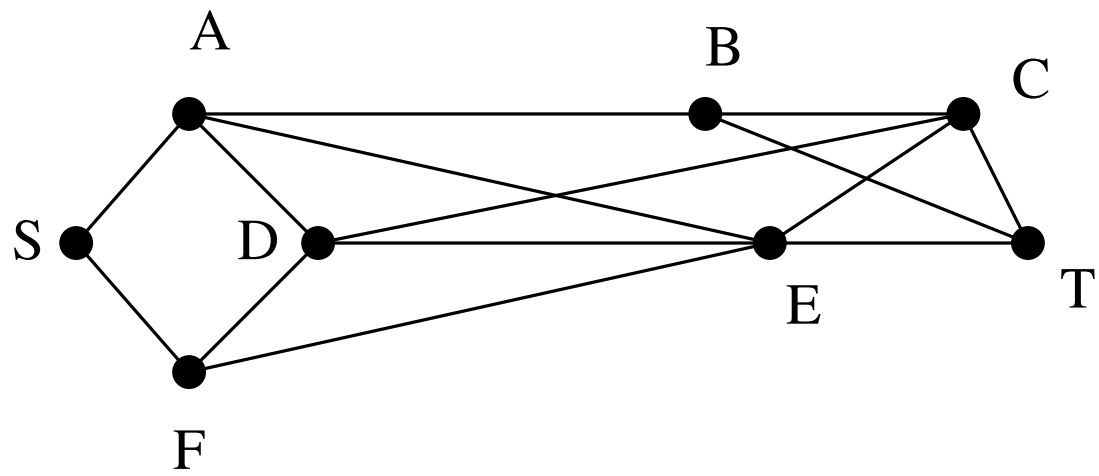
$$\mu(v|T) := \sum \mu(u|T)$$

   The sum is over all u's which satisfy the following condition: $h(u) = i$ and there is an edge from v to u. If there are no such u then $\mu(v|T) = 0$.
3. If $i = 0$ STOP and let $\sigma_{ST} = \mu(S|T)$. If not, decrease i to $i - 1$ and go to Step 2.



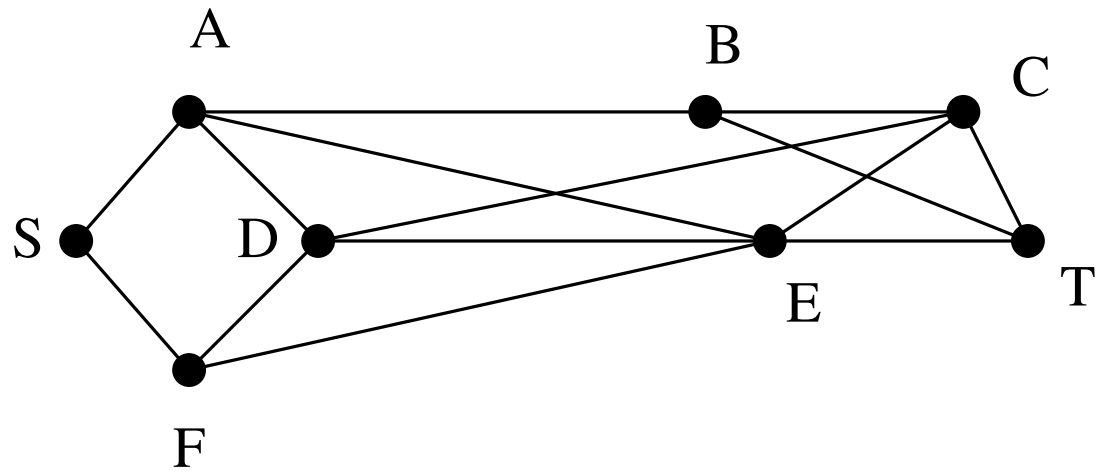
For $i = 3$ we have

$$\mu(T|T) = 1$$
$$\mu(C|T) = 0$$



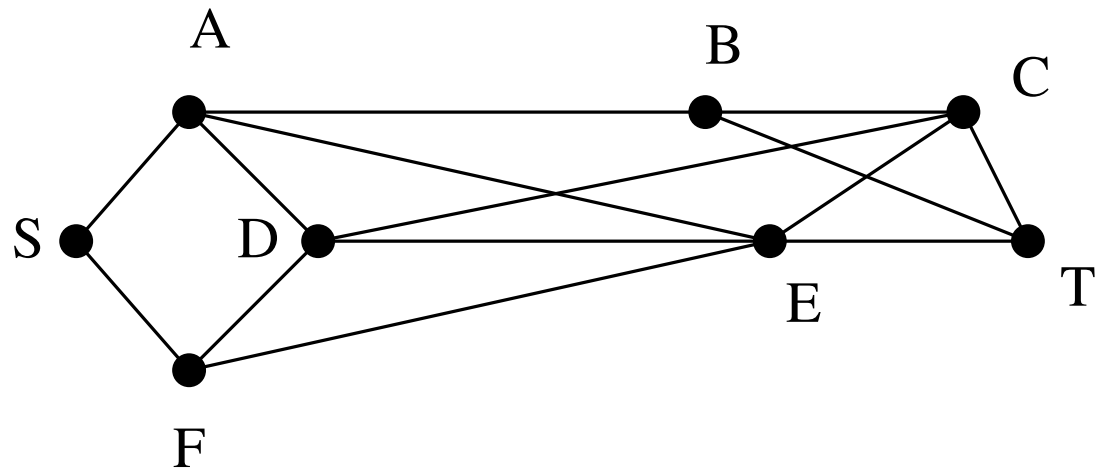
For $i = 2$ we have

$$\begin{aligned}
\mu(B|T) &= \mu(C|T) + \mu(T|T) \\
&= 0 + 1 = 1 \\
\mu(D|T) &= \mu(C|T) \\
&= 0 \\
\mu(E|T) &= \mu(C|T) + \mu(T|T) \\
&= 0 + 1 = 1
\end{aligned}$$



For $i = 1$ we have

$$\begin{aligned}\mu(A|T) &= \mu(B|T) + \mu(D|T) + \mu(E|T) \\ &= 1 + 0 + 1 = 2 \\ \mu(F|T) &= \mu(D|T) + \mu(E|T) \\ &= 0 + 1 = 1\end{aligned}$$



and for $i = 0$ we have

$$\begin{aligned}\sigma_{ST} = \mu(S|T) &= \mu(A|T) + \mu(F|T) \\ &= 2 + 1 = 3\end{aligned}$$

Therefore there are three paths having the shortest length from S to T. These are $SABT$, $SAET$ and $SFET$.

Matrix approach for the length of a shortest path

As in the case with activity networks and critical paths, there is a matrix approach to determining the length, say $h(S|T)$, of the shortest paths between two nodes S and T.

Let

$$a_{ij} = \begin{cases} 1, & \text{if } (i,j) \text{ is an edge or arc} \\ 0, & i = j \\ \infty, & \text{otherwise} \end{cases}$$

Min-plus matrix algebra

Let elements of row i of A, and elements of column j be

$$\vec{a}_i = [a_{i1}\, a_{i2}\, \cdots a_{iN}], \qquad \vec{b}_j = [b_{1j}\, b_{2j}\, \cdots b_{Nj}]$$

Then we define

$$\vec{a}_i \otimes_{\min} \vec{b}_j = \min\{a_{i1} + b_{1j}, a_{i2} + b_{2j}, \ldots a_{iN} + b_{Nj}\}$$

and

$$A \otimes_{\min} B = [\vec{a}_i \otimes_{\min} \vec{b}_j]_{i,j=1,\ldots,N}.$$

To find the lengths of the shortest paths from S to all other nodes we compute

$$\underbrace{A \otimes_{\min} \cdots \otimes_{\min} A}_{N-1 \text{ times}} = A^{N-1}$$

where N is the total number of nodes, and

$$A = [a_{ij}], \qquad a_{ij} = \begin{cases} 1 & \text{if edge } (i,j) \text{ exists,} \\ 0 & \text{if } i = j, \\ \infty & \text{otherwise.} \end{cases}$$

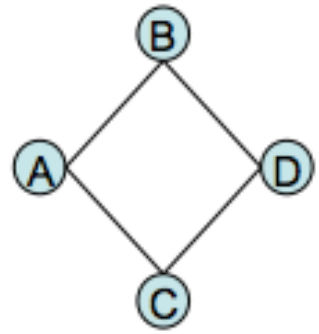
Notes

1. Since we are only interested in the row corresponding to vertex S, we can compute

$$[0 \; \infty \cdots \infty] \otimes_{\min} A^{N-1}.$$

2. The same formalism applies to weighted graphs, weighted $a_{ij} \geq 0$.

Exercise: Use the min-plus algebra to determine the shortest path length between A and D in the following graph.



The appropriate incident matrix is

$$A = \begin{bmatrix} 0 & 1 & 1 & \infty \\ 1 & 0 & \infty & 1 \\ 1 & \infty & 0 & 1 \\ \infty & 1 & 1 & 0 \end{bmatrix}$$

We have

$$\begin{bmatrix} 0 & \infty & \infty & \infty \end{bmatrix} \otimes_{\min} A = \begin{bmatrix} 0 & 1 & 1 & \infty \end{bmatrix}$$

$$\begin{bmatrix} 0 & 1 & 1 & \infty \end{bmatrix} \otimes_{\min} A = \begin{bmatrix} 0 & 1 & 1 & 2 \end{bmatrix}$$

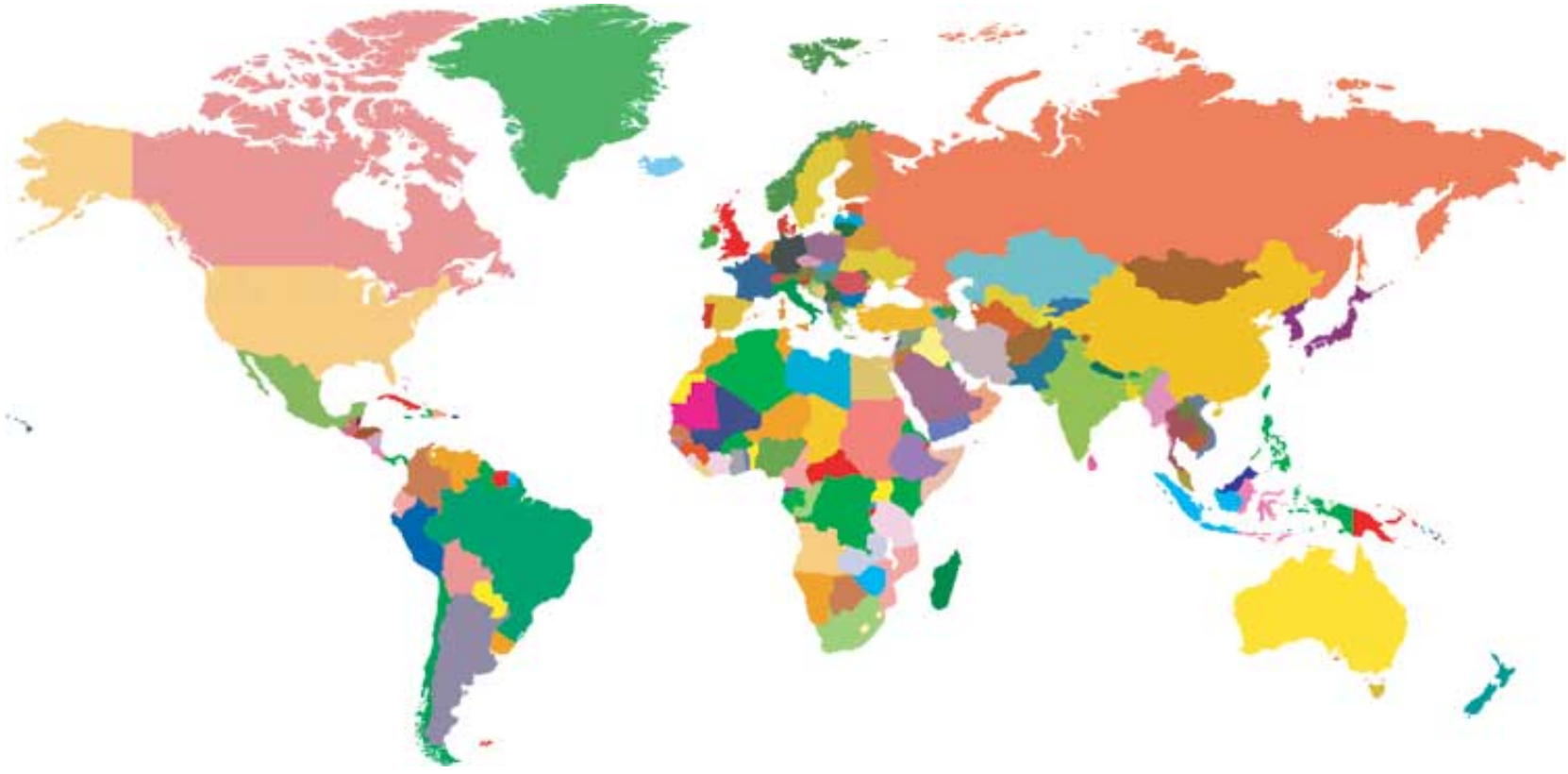
Graph colourings and interval graphs

A colouring of a graph is an assignment of colours to the nodes of the graph such that adjacent nodes have distinct colours.

The least number of colours needed to colour a graph is called the vertex chromatic number of the graph.

In the graph colouring problem we wish to find the vertex chromatic number of a graph or class of graphs.

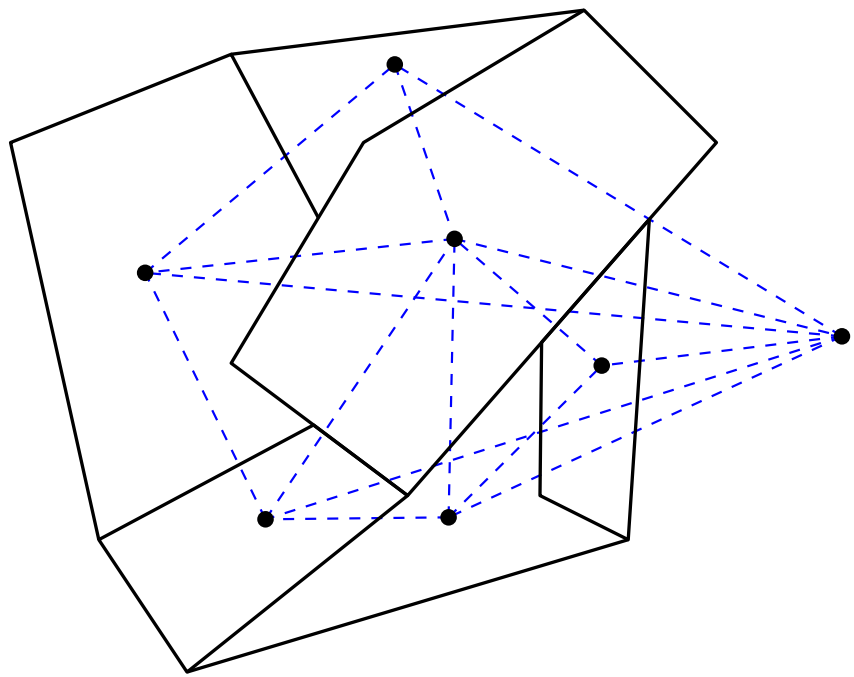
The problem was initially inspired by the map colouring problem: for any map, what is the least number of colours needed so that neighbouring countries have distinct colours?



Planar graphs

Every map can be represented as a planar graph (a graph that can be drawn without any crossing edges).

A vertex is placed in each region and two vertices are joined by an edge if their corresponding regions share a border.

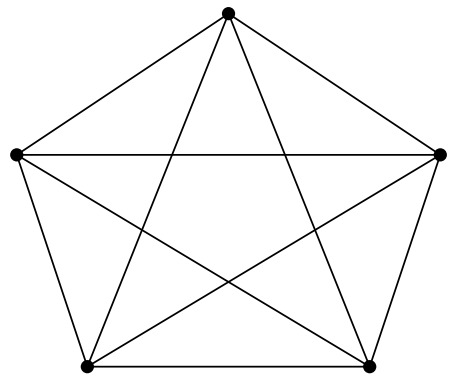


Planar graphs

In the above example the graph was drawn with some edges crossing. Can you draw the same graph without any crossing edges?

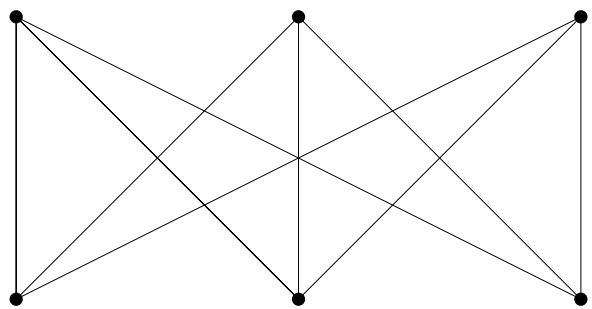
The below graph cannot be drawn without at least one pair of edges crossing (try!). This graph, known as K_5, is not planar.

K_5 has 5 vertices, and every pair of vertices is joined by an edge. Try to draw a map with 5 regions such that every pair of regions share a border.



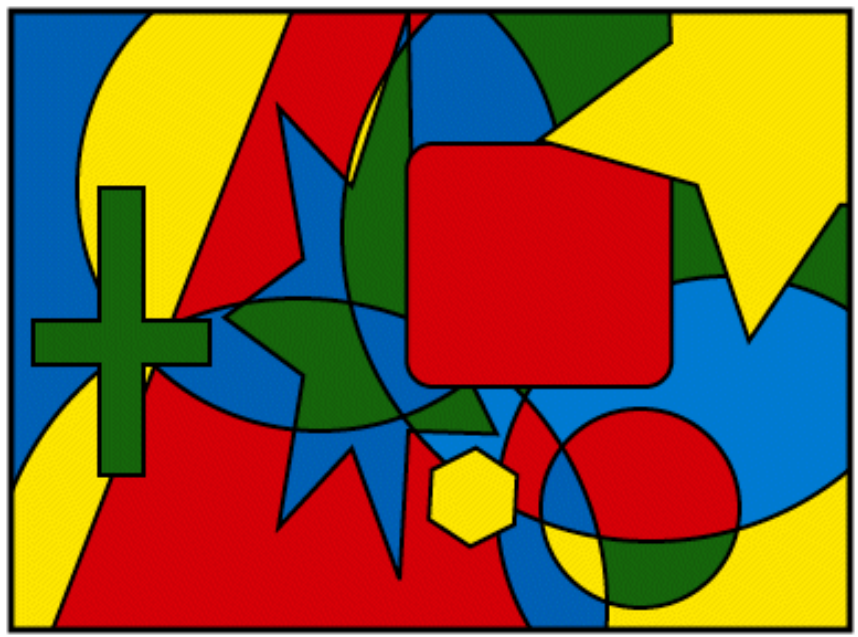
Planar graphs

The below graph also cannot be drawn without at least one pair of edges crossing. This graph is known as $K_{3,3}$ and is therefore not planar.



Planar graphs

The map colouring problem is therefore equivalent to the problem of finding the vertex chromatic number of planar graphs. It has been shown that no planar graph needs more than four colours. This is the famous Four-colour theorem.



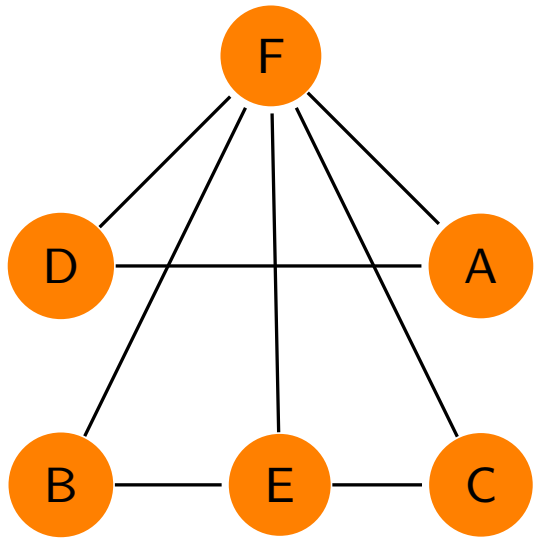
An application: hotel bookings

Graph colourings have many applications. One example we look at is hotel bookings.

Suppose, for a particular week, 6 people make the following bookings at a hotel

Adam		Tue	Wed				
Bob				Thur	Fri		
Cara						Sat	Sun
Dave	Mon	Tue	Wed				
Elle					Fri	Sat	
Fay		Tue	Wed	Thur	Fri	Sat	Sun

We can represent this information graphically, using nodes for the people and connecting them when both people require a room on the same day.



Our problem is to determine the least number of rooms required at the hotel to accommodate all the bookings.

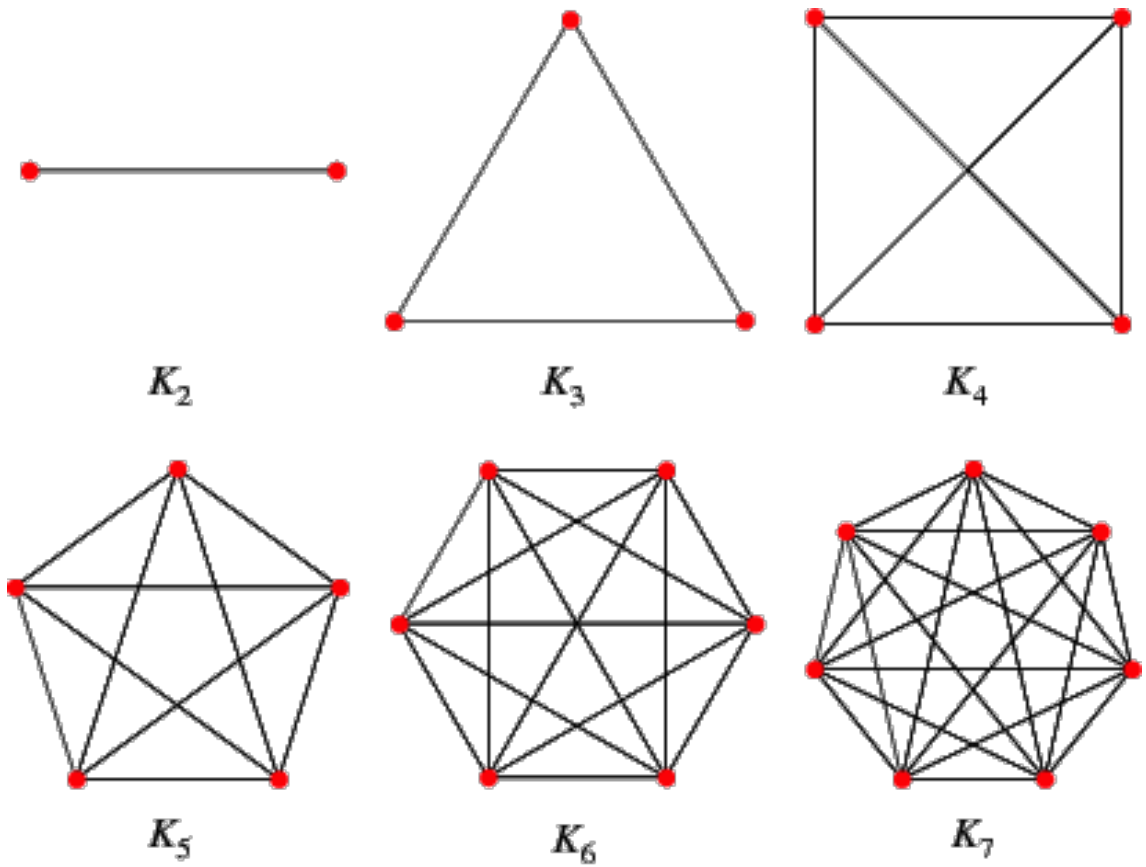
We colour the vertices of the graph according to the rooms the guests will stay in. Each colour represents a room.

We colour according to the rule: no adjacent vertices have the same colour.

To find the minimum number of rooms to use we calculate the vertex chromatic number of the bookings graph.

Cliques

A clique in a graph is a set of vertices such that each vertex in the set is adjacent to every other vertex in the set. Cliques are also known as complete graphs.



A lower bound

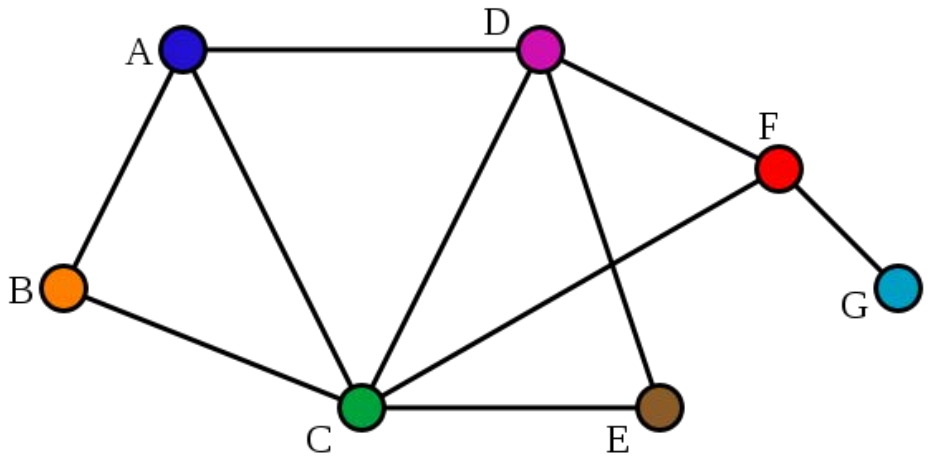
Theorem

For all graphs

$$\text{vertex chromatic number} \geq \text{size of largest clique}$$

Interval graphs

An interval graph is a graph where the nodes are intervals (connected sequences of numbers or symbols) and two nodes are adjacent if and only if the corresponding intervals intersect.



Observe that hotel booking graphs are interval graphs.

Interval graphs

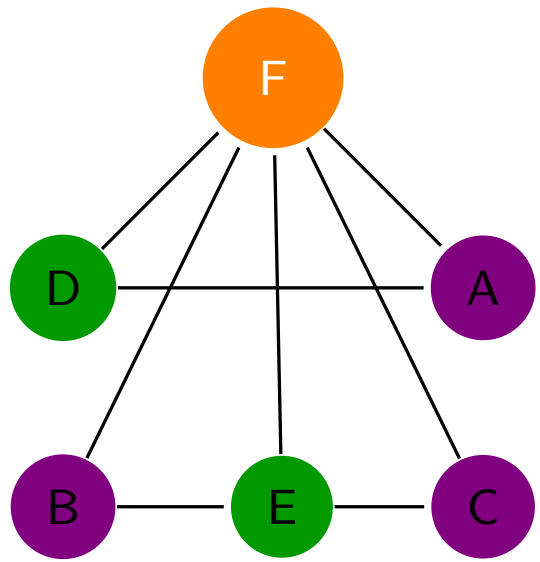
Theorem

For interval graphs

$$\text{vertex chromatic number} = \text{size of largest clique}$$

Example

D	M	T	W				
A		T	W				
F		T	W	Th	F	S	S
B				Th	F		
E					F	S	
C						S	S

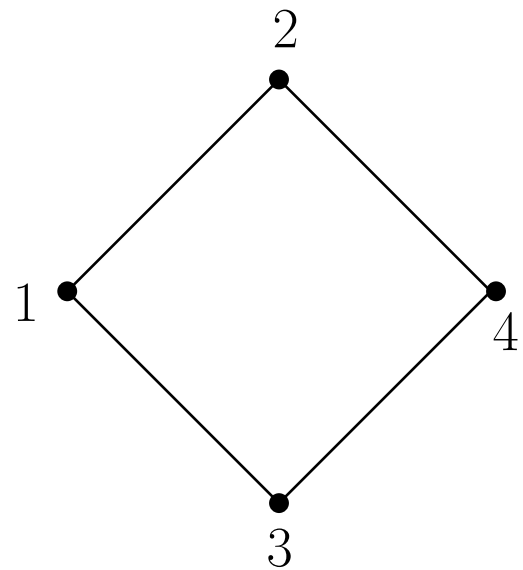


Interval graphs

Observe that the converse of the above theorem does not hold. In other words, there are graphs that are not interval graphs that also have vertex chromatic number equal to the size of the largest clique.

Example

What is the chromatic number of C_4, the cycle on four nodes? What is the size of the largest clique in C_4? Can you prove that C_4 is not an interval graph?

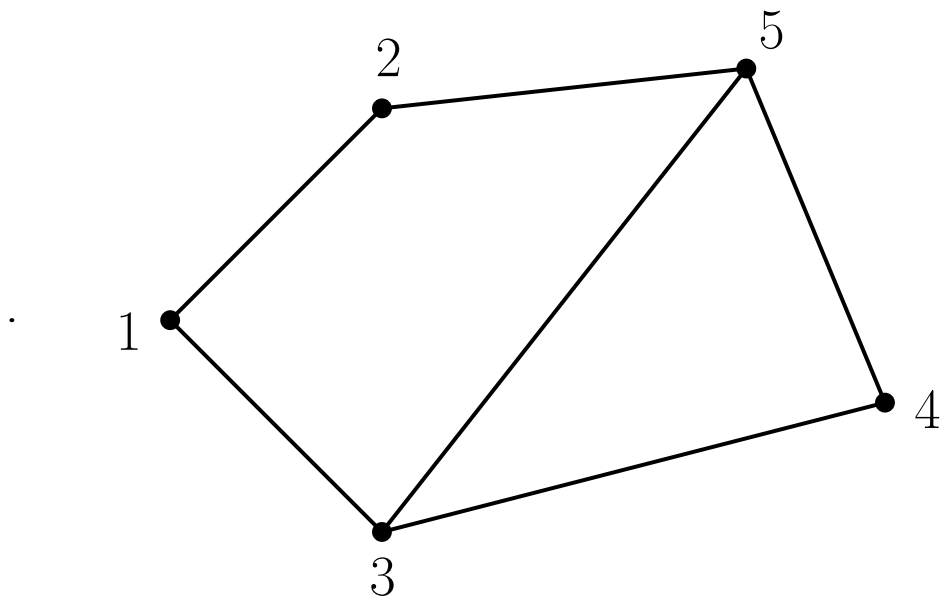


Chordal graphs

Definition

A chord of a graph is an edge joining two non-consecutive nodes on a cycle of the graph.

Example



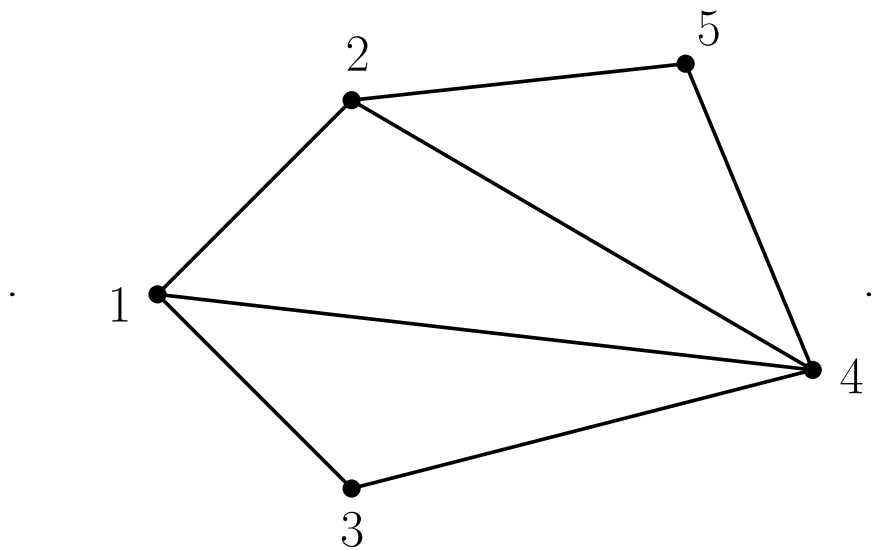
Chordal graphs

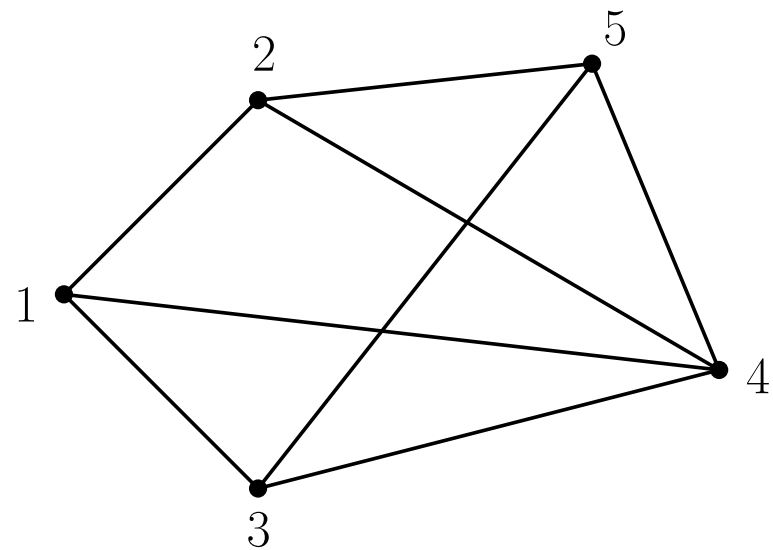
Definition

A chordal graph is a graph where every cycle of length at least four has a chord.

Example

The graph on the left is chordal. Is the graph on the right chordal?





Chordal graphs are also known as triangulated graphs.

Chordal graphs

Theorem

For chordal graphs

$$\text{vertex chromatic number} = \text{size of largest clique}$$

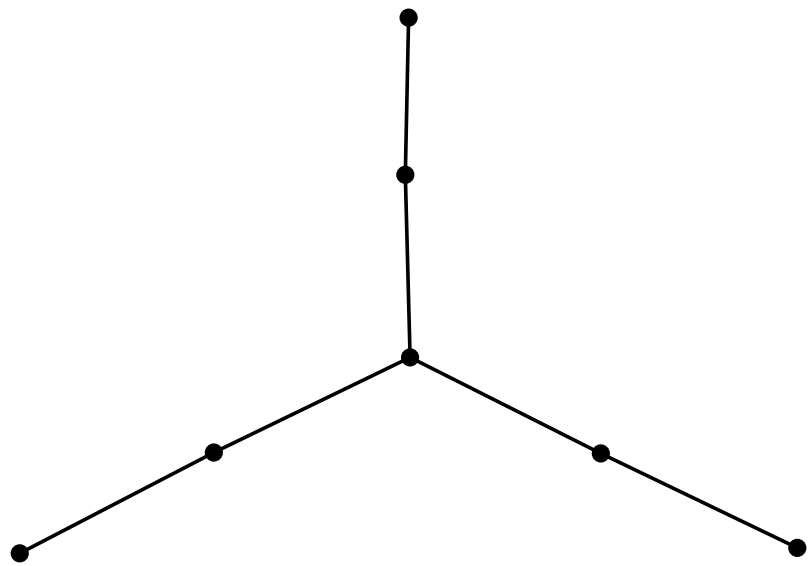
Theorem

Every interval graph is a chordal graph.

Chordal graphs

Example

The graph below is (vacuously) a chordal graph. Is it an interval graph?



Examples

There are many other examples of graphs for which vertex chromatic number equals size of largest clique.

More examples
Complete graphs
Even cycles

What about odd cycles?

An upper bound - Brook's Theorem

The degree of a node is the number of edges incident to the node.

Theorem: Brook's Theorem

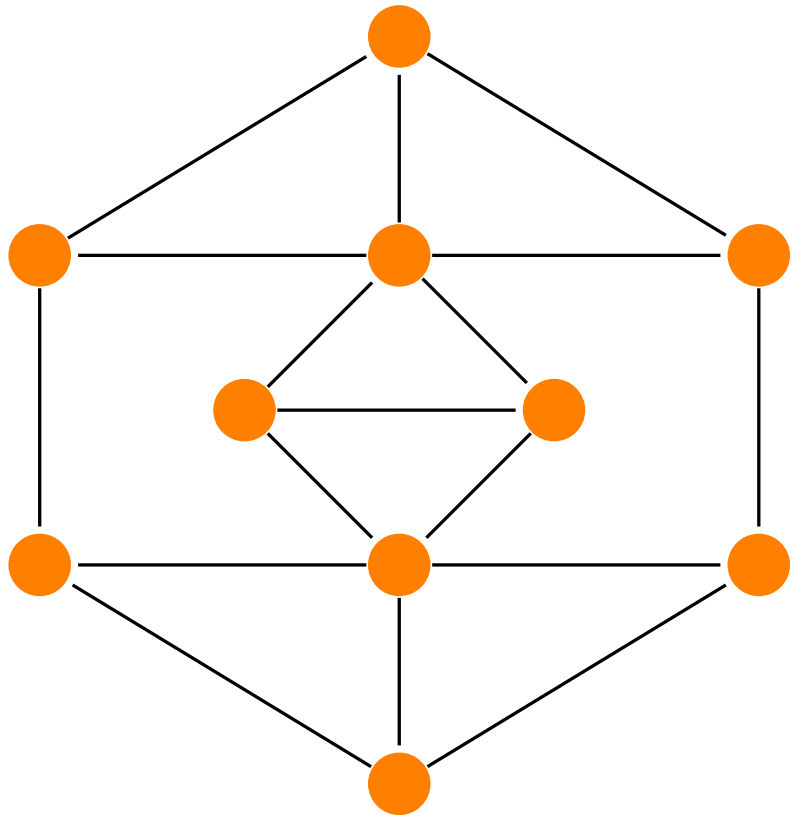
For any graph except complete graphs and cycles with an odd number of edges we have

$$\text{vertex chromatic number} \leq \text{maximum degree of any node}$$

In general it is difficult (NP-complete) to determine the vertex chromatic number of a graph.

Exercise

Find bounds on the vertex chromatic number of the following graph.



Now use trial and error to find the exact chromatic number.

Matching

We now move on to a different application that can also be solved using a graph model. Consider the problem of optimising timetables. We will first analyse the problem by means of matrices, and then look at how graphs can simplify the problem. Consider the setting of m teachers $x_1, x_2, \ldots, x_m$ and n classes $y_1, y_2, \ldots, y_n$.

Given that Teacher x_i is required to teach Class y_j for p_{ij} periods, the task is to schedule a timetable in the minimum possible number of time slots (periods), given that no class is taught more than once in a given period.

Example

Suppose the teacher/class allocation is given by the following table $P = [p_{ij}]$

	y_1	y_2	y_3	y_4	y_5
x_1	1	0	1	0	2
x_2	1	0	1	0	0
x_3	0	1	0	1	0
x_4	1	1	0	1	1

We decompose P into $0, 1$-matrices such that no column or row has more than one 1. This is called a single 1's decomposition.

$$\begin{bmatrix} 1 & 0 & 1 & 0 & 2 \\ 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix} + \begin{bmatrix} 0 & 0 & 1 & 0 & 2 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 1 \end{bmatrix}$$

So in Period 1, Teacher x_1 can teach Class y_1, Teacher x_2 can teach Class y_3, Teacher x_3 can teach Class y_2 and Teacher x_4 can teach Class y_4.

The significance of the $0,1$-matrix so constructed is that each teacher is matched to a distinct class, and so these classes can all be held at the same time.

Now we repeat:

$$\begin{bmatrix} 0 & 0 & 1 & 0 & 2 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 & 0 & 2 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 \end{bmatrix}$$

We read off from this that in Period 2, Teacher x_1 can teach Class y_3, Teacher x_2 can teach Class y_1, Teacher x_3 can teach Class y_4 and Teacher x_4 can teach Class y_5.

Finally

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 2 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \end{bmatrix}$$

So in Period 3 Teacher x_1 can teach Class y_5 and Teacher x_4 can teach Class y_1, while the other teachers have no classes. In Period 4 Teacher x_1 again teaches Class y_5 while Teacher x_4 teaches Class y_2 (and again, Teachers x_2 and x_3 have no classes).

We have found teaching allocations for 4 time slots. This is the minimum possible because both teachers x_1 and x_4 have four classes to teach.

Note that not all single 1's decompositions lead to just 4 matrices.

Suppose for example that, at Step 2, we instead wrote:

$$\begin{bmatrix} 0 & 0 & 1 & 0 & 2 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 & 0 & 2 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 \end{bmatrix}$$

We will then find (exercise) that more than 4 periods are needed. Observe that the sum of entries in column 5 has not decreased here.

Theorem

In a matrix which is not all 0's, a single 1's matrix can always be subtracted which reduces the maximum sum across all rows and columns

Theorem

Reducing the maximum sum at each step yields an optimal number of periods

Bipartite graphs

Now lets look at the graph model for the timetabling problem.

Definition

A bipartite graph is a graph with the property that its set of nodes can be partitioned into two sets such that there are no edges within either set.

or

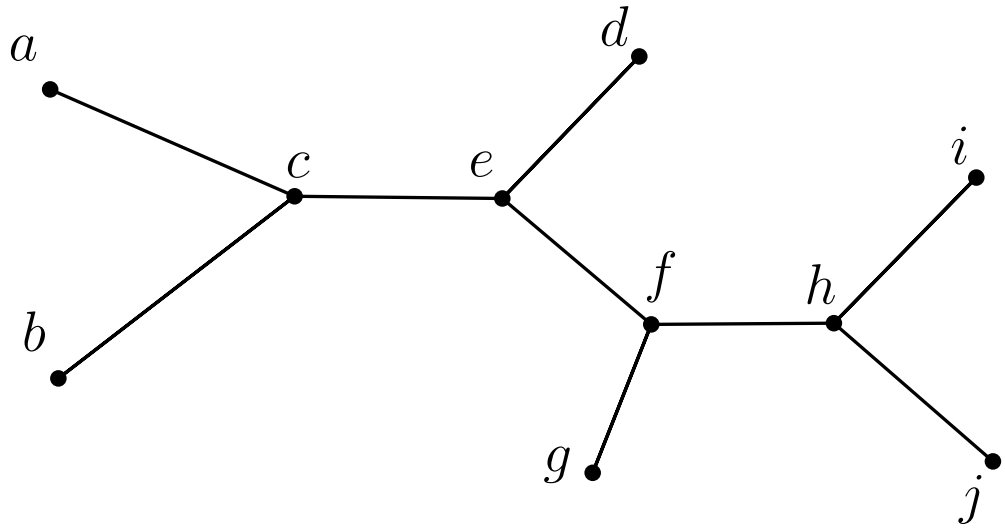
Definition

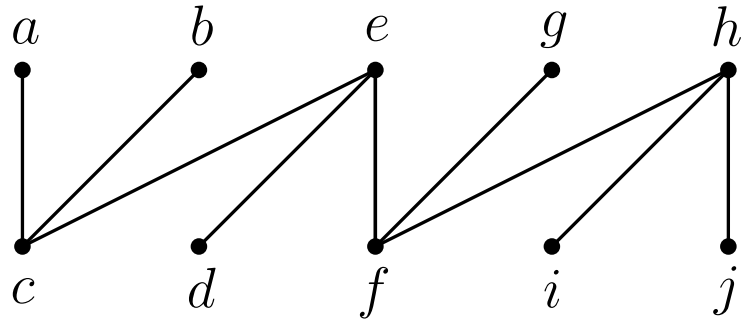
A bipartite graph is a graph that can be coloured using at most 2 colours.

Bipartite graphs

Example

Trees are bipartite.





Bipartite graphs

Examples

Even cycles are bipartite.

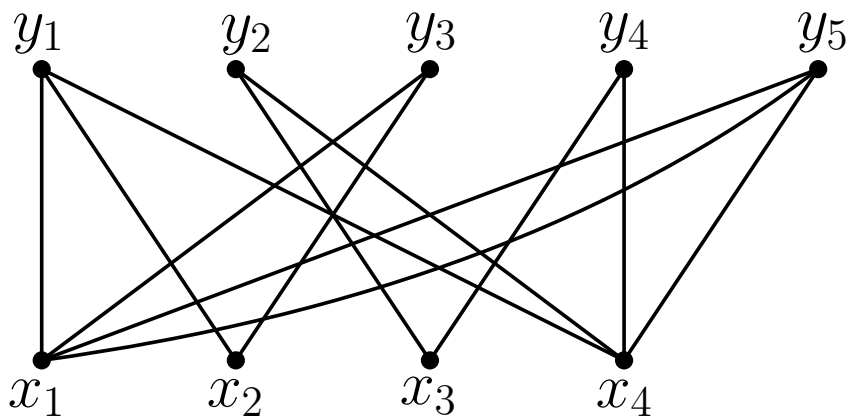
Odd cycles are not bipartite.

The empty graph (a graph without edges) is bipartite.

Bipartite graphs

In the timetabling problem the teacher/class allocation can be modelled as a bipartite graph.

	y_1	y_2	y_3	y_4	y_5
x_1	1	0	1	0	2
x_2	1	0	1	0	0
x_3	0	1	0	1	0
x_4	1	1	0	1	1



Matching

How do we model the periods in the timetabling problem?

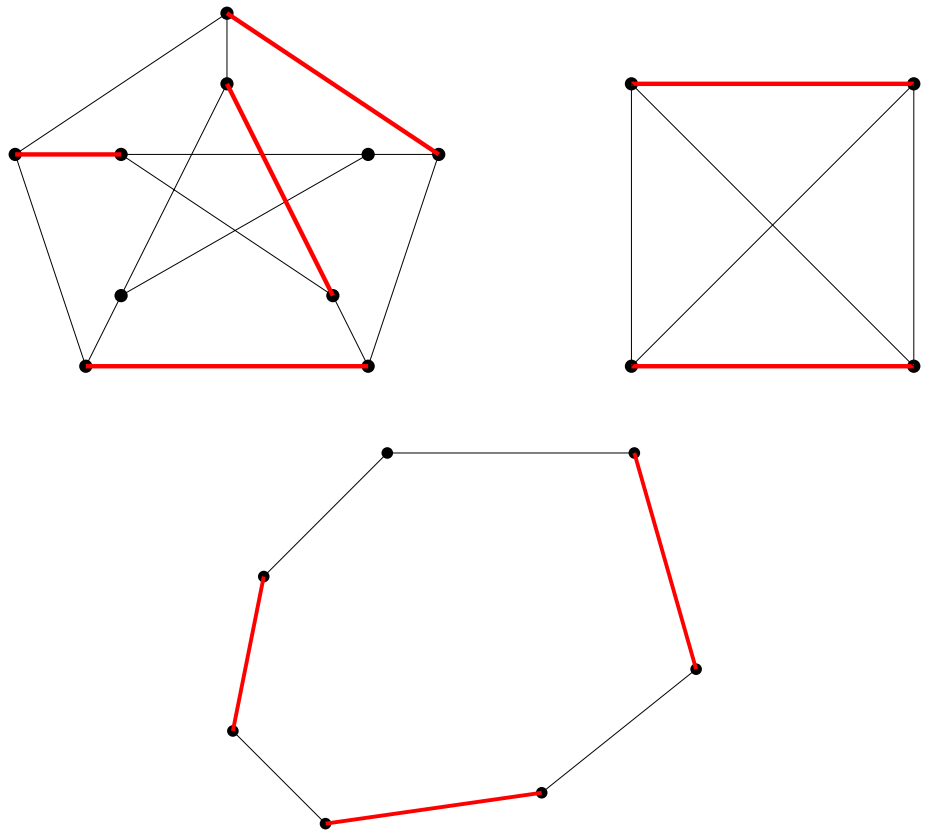
The key property is that no class is taught more than once in a given period and, of course, a teacher cannot teach two or more classes in the same period. This property is captured by so-called graph matchings.

Definition
A set of nodes (or edges) is called independent if no two nodes (or edges) in the set are adjacent.

Definition
A matching in a graph is a set of independent edges.

Matching

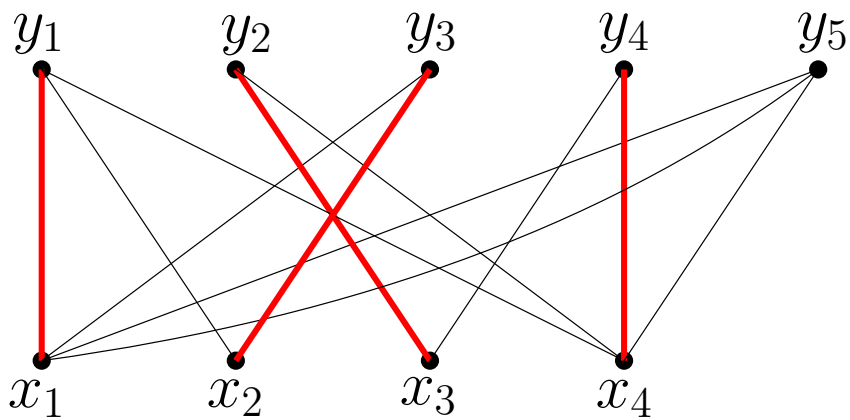
In the examples below the matchings are shown in red. In each case, can you tell if the matching is of maximum size?



Matching

A period in the timetabling problem is just a matching in the associated bipartite graph.

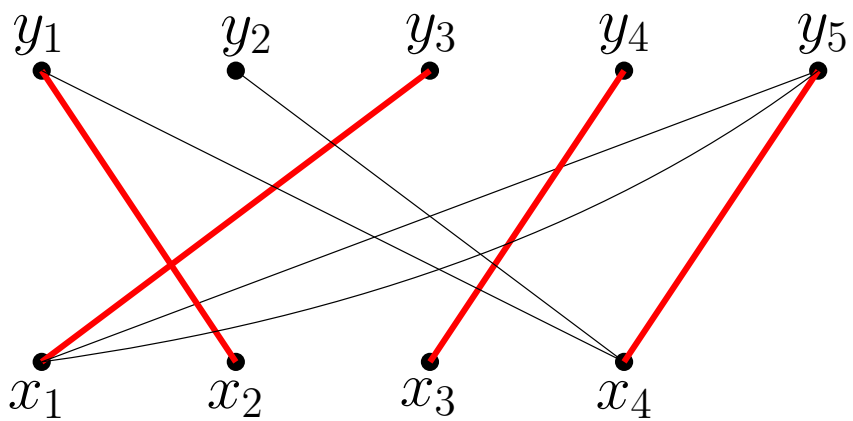
For example, Period 1 that we constructed above is represented by the set of red edges in the graph.



Period 1

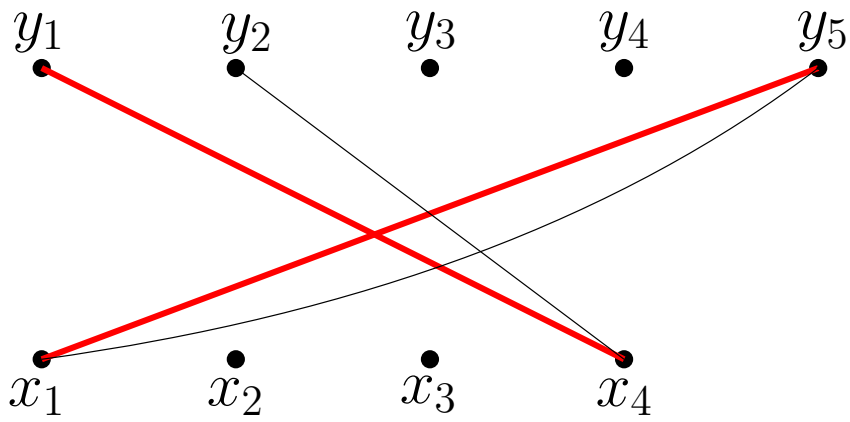
Matching

Removing this matching from the graph is equivalent to taking out a single 1's matrix. We then find a new matching in order to find Period 2. And repeat.

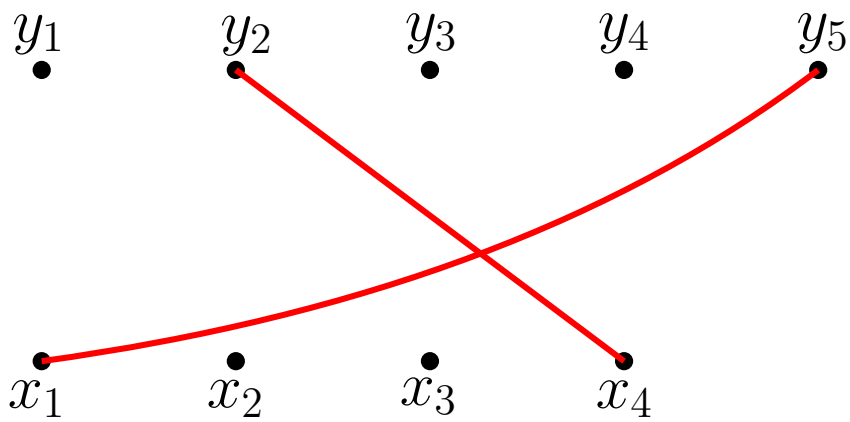


Period 2

Matching



Period 3



Period 4

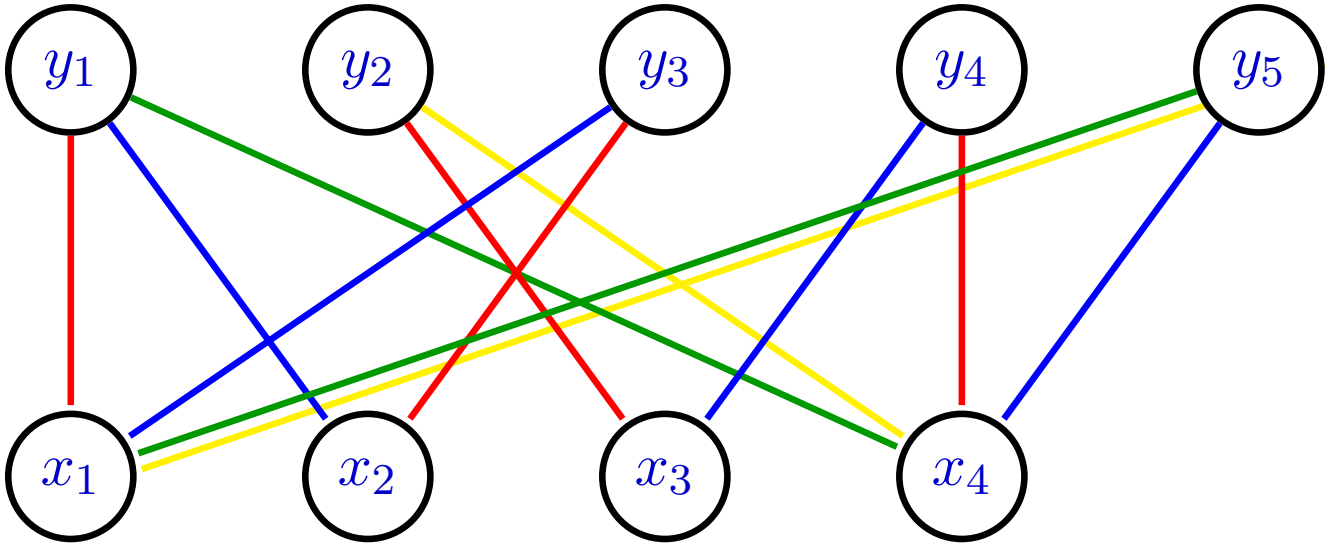
Matching

We also need to model the objective of the timetabling problem.

Since periods correspond to matchings, to solve the timetabling problem we need to solve the following.

> What is the **minimum number** of disjoint matchings that the edges of a bipartite graph can be partitioned into?

This question can also be stated in terms of so-called edge-colourings.



Notice that each matching (period) has its own colour.

No edges of the same colour can meet at the same node.

Edge-colourings

Definition

An edge-colouring of a graph is an assignment of colours to the edges such that no two edges with a common endpoint have the same colour.

Therefore a set of edges of the same colour in a graph forms a matching.

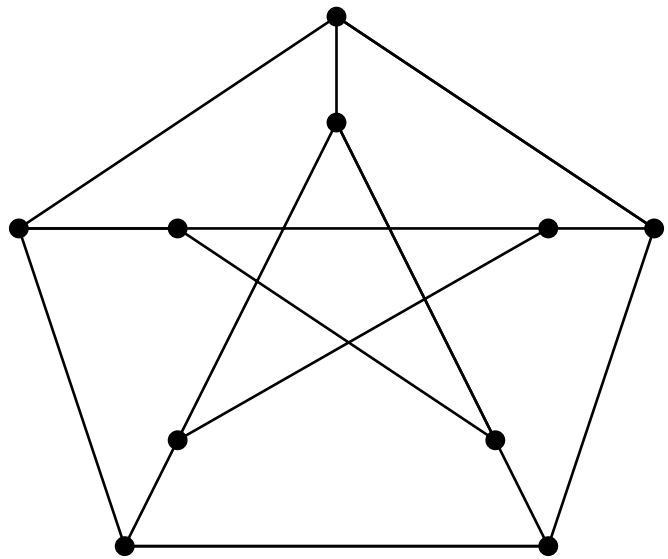
Definition

The edge-colouring problem requires one to find an edge-colouring of a given graph using the minimum number of colours. This minimum number is called the edge-chromatic number of the graph.

Notice then that the timetabling problem is an edge-colouring problem in bipartite graphs.

Edge-colourings

Example



Find an edge-colouring of the above graph. Can you also find the edge-chromatic number of the graph?

Edge-colourings

Theorem

In any bipartite graph, a matching can be found which contains all nodes of maximum degree

Theorem

Let G be a bipartite graph and let Δ be the maximum degree of any node of G. Then the edge-chromatic number of G is Δ

Compare these two theorems to the two theorems on single 1's decompositions.

Exercise: Assuming the first theorem, prove the second one.

Modifying matchings: augmenting paths

In our solution to the timetabling problem, we had four classes in the first and second periods, and two classes in the third and fourth periods.

We ask, would it be possible to have exactly three classes in each of the four time slots?

In terms of graph theory, this involves transforming a matching into a new matching of possibly different size.

For this we need the concept of an augmenting path.

Modifying matchings: augmenting paths

Let G be a graph and let M be a matching in G.

Definition

An exposed node of G with respect to M is a node that is not an endpoint of any edge in M.

Definition

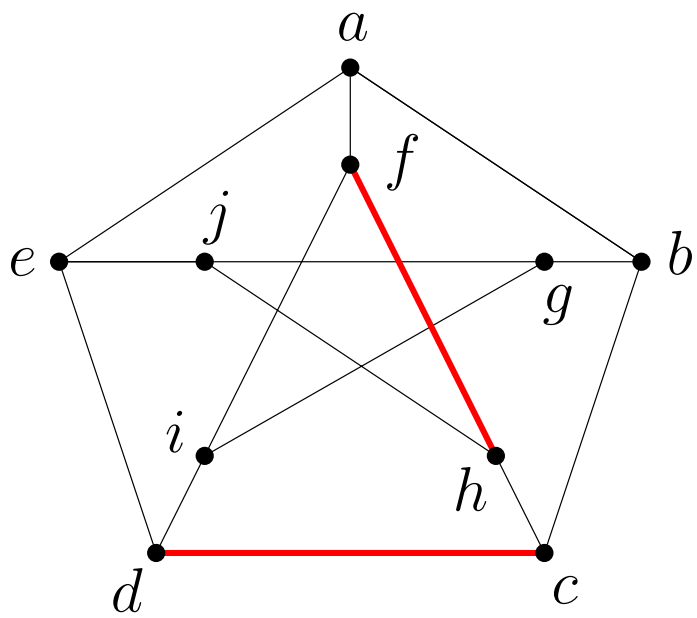
An alternating path in G with respect to M is a path with edges alternately in M and not in M.

Definition

An augmenting path in G with respect to M is a path of the form x, P, y, where x and y are exposed nodes and P is an alternating path with first and last edges in M.

Alternating paths and exposed nodes

Example



Some alternating paths

Path a, f, h

Path f, h, c, d

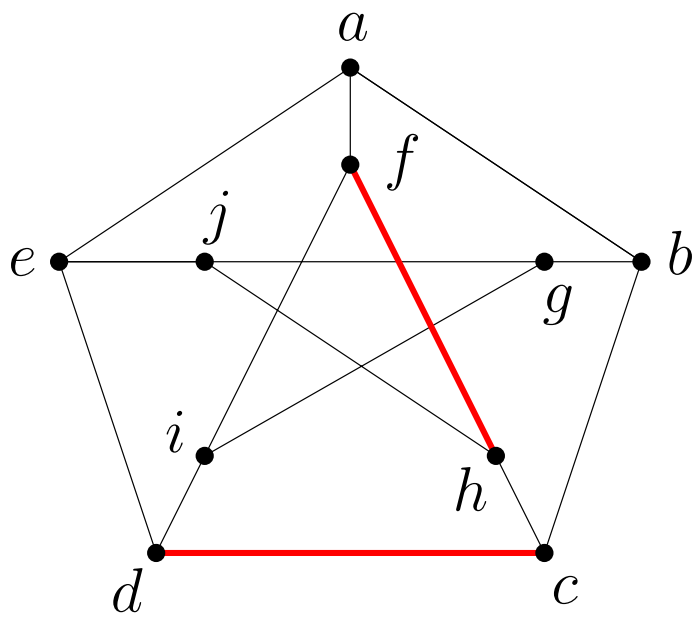
Path a, b

Exposed nodes

a, b, e, j, i, g

Augmenting paths

Example



Some augmenting paths

Path e, d, c, h, f, a. Here $P = d, c, h, f$

Path j, h, f, a. Here $P = h, f$

Path a, b. Here $P = \emptyset$

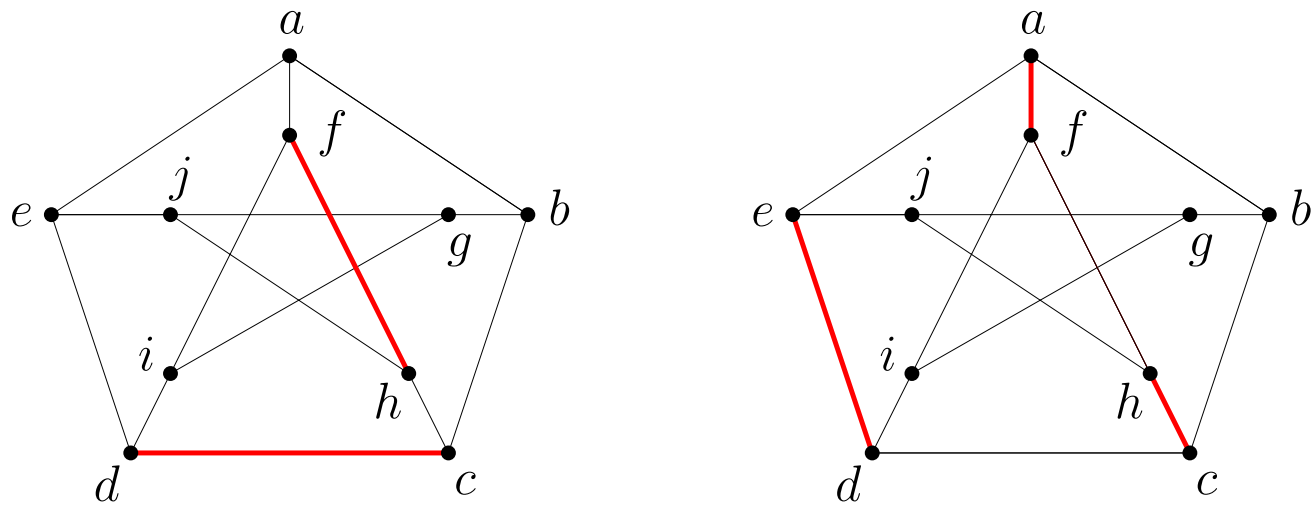
Augmenting paths

Theorem

If graph G has an augmenting path with respect to a matching M then M can be modified (augmented) into a matching of size $|M| + 1$

Augmenting paths

Example



Consider the augmenting path e, d, c, h, f, a.

A nice shorthand way of depicting the edges of this path which belong to the current matching is $e, (d, c), (h, f), a$.

The augmented matching is $(e, d), (c, h), (f, a)$.

Augmenting paths

Going back to our timetabling problem, the following theorem can help us to balance the number of classes per period.

Theorem

Suppose M_1 and M_2 are disjoint matchings in a bipartite graph G, with $|M_1| > |M_2|$. Then there are disjoint matchings $\tilde{M}_1$ and $\tilde{M}_2$ of G such that $|\tilde{M}_1| = |M_1| - 1$, $|\tilde{M}_2| = |M_2| + 1$ and $\tilde{M}_1 \cup \tilde{M}_2 = M_1 \cup M_2$

Augmenting paths

Proof

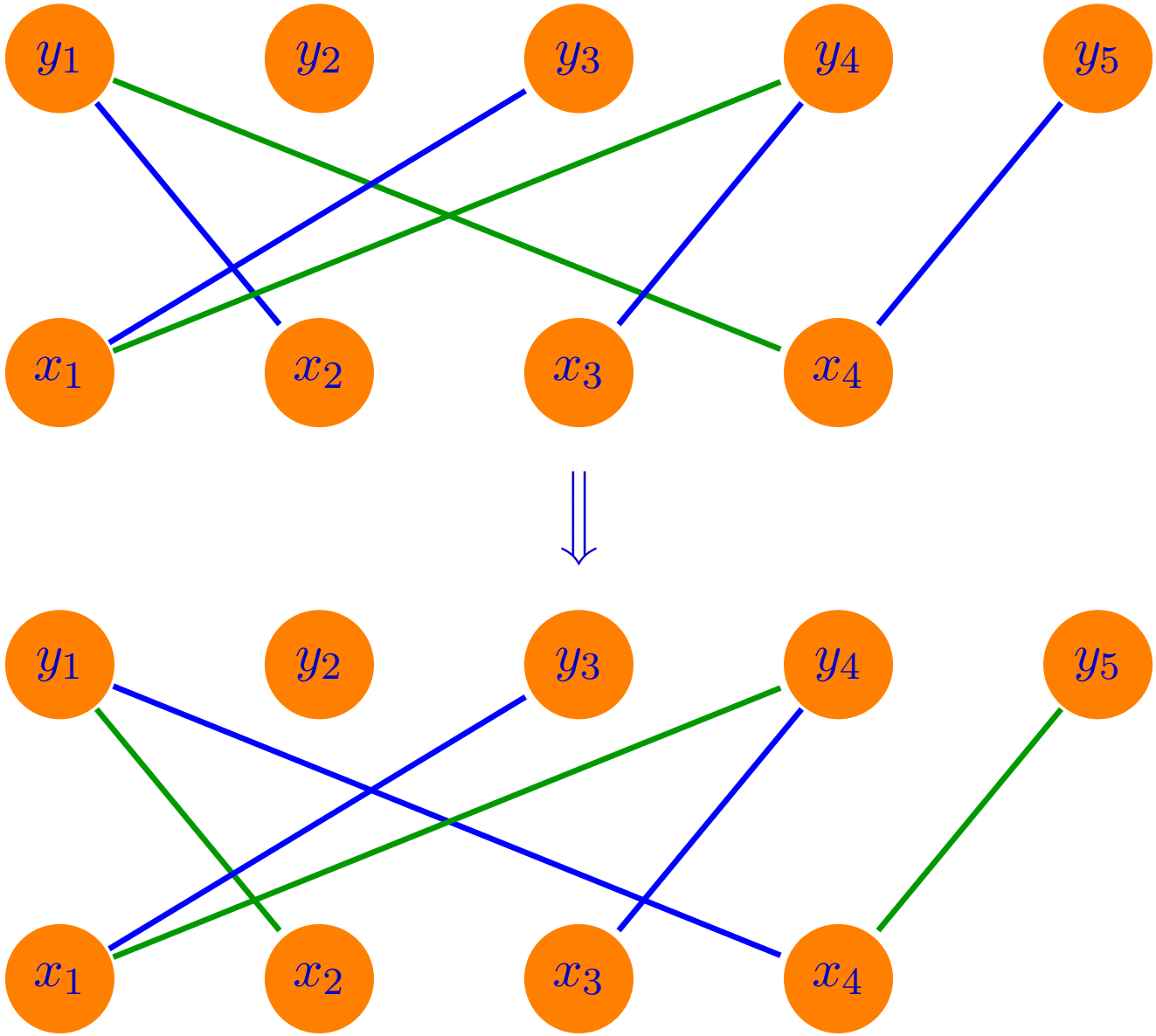
Consider the graph $G' = M_1 \cup M_2$.

Note that each connected component of G' is either a cycle consisting of an even number of edges, or a path with edges alternatively in M_1 and M_2.

Since $|M_1| > |M_2|$ some component which is a path must begin and end in M_1, and is therefore an augmenting path with respect to M_2.

If we augment this path so that it begins and ends in M_2, we get the desired new matchings.

□



The above augmentation converts a period with four classes and a period with two classes into two periods of size three.

Assignment

Our next application, the assignment problem, is closely related to the timetabling problem and can also be modelled using graphs and matchings.

As before, we will also model the problem in an alternative way for comparison; in this case, using matrices and set theory.

Our example problem is as follows: suppose you have 8 people that you would like to assign to 8 jobs. Each person is either suitable for a given job, or not.

How can you find a feasible assignment so that every person is assigned to one job they are suitable for, and every job has been assigned to one person?

Assignment

Let's consider the problem from a matrix perspective first. We denote the eight people by $a, b, \ldots, h$ and the eight jobs by $1, 2, \ldots, 8$.

Construct a matrix with rows labelled by people, and columns labelled by jobs, with entry 1 if the person is suited to the job, and 0 otherwise.

	1	2	3	4	5	6	7	8
a	1	0	0	1	1	0	1	0
b	0	1	0	0	0	1	0	0
c	1	0	1	0	1	0	0	0
d	0	1	0	1	0	1	0	0
e	0	1	1	0	1	1	0	0
f	0	1	0	1	0	1	0	0
g	1	0	0	0	1	1	1	1
h	1	0	0	1	0	1	0	0

Assignment

We now find an 8×8 single 1's matrix within the above suitability matrix. In other words, we find a set of 1's so that there is a 1 in each row, no two of which are in the same column.

But how do we find such a set of 1's, especially if the matrix is very large?

Can we be guaranteed that such a set of 1's even exists? Remember, we need every row and every column to contain a 1 in the set.

Assignment

The assignment problem is therefore a feasibility problem.

Consider the below suitability matrices and determine whether they give feasible assignments.

$$\begin{bmatrix} 0 & 0 \\ 1 & 1 \end{bmatrix} \qquad \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix} \qquad \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix} \qquad \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix}$$

Assignment

It is desirable to have some conditions we can check in order to determine whether a given suitability matrix yields a feasible assignment.

One way of doing this uses the following model from set-theory.

For the 8×8 job suitability matrix from before, lets associate with each person the set of jobs they are suited to:

$$S_1 = \{1, 4, 5, 7\}, \quad S_2 = \{2, 6\}, \quad S_3 = \{1, 3, 5, 8\},$$

$$S_4 = \{2, 4, 6\}, \quad S_5 = \{2, 3, 5, 6\}$$

$$S_6 = \{2, 4, 6\}, \quad S_7 = \{1, 5, 6, 7, 8\}, \quad S_8 = \{1, 4, 6\},$$

Systems of distinct representatives

We now choose a system of distinct representatives (SDR) from these sets.

In other words, we pick one element (representative) from each set such that each set has a unique representative.

For instance we could pick the following representatives:

$$R_1 = 7,\ R_2 = 2,\ R_3 = 3,$$

$$R_4 = 4,\ R_5 = 5,$$

$$R_6 = 6,\ R_7 = 8,\ R_8 = 1$$

Systems of distinct representatives

Fortunately, the previous example had a feasible assignment. The next theorem provides conditions under which a feasible assignment exists.

Theorem

A necessary and sufficient condition for the sets $S_1, \ldots, S_n$ to possess an SDR is that the union of any k of these sets, where $k \leq n$, contains at least k elements.

Example

Consider the sets
$S_1 = \{1\}$, $S_2 = \{1, 2\}$, $S_3 = \{1, 2, 3\}$, $S_4 = \{1, 2, 3, 4\}$,
$S_5 = \{1, 2, 3, 4, 5\}$.

The union of any k sets contains the set S_k and thus has at least k elements. Therefore Hall's theorem is satisfied. A set of distinct representatives for $S_1, ..., S_5$ is $\{1, 2, 3, 4, 5\}$.

What about this example:
$S_1 = \{1, 2\}, S_2 = \{2, 3\}, S_3 = \{1, 3\}, S_4 = \{1\}, S_5 = \{3\}$.

Proof of Hall's theorem

Proof

We only need to prove sufficiency. Given the condition of the theorem we will provide an algorithm to show how the distinct representatives can be chosen.

Now suppose $r < n$ distinct representatives have already been chosen from the sets $S_1, \ldots, S_r$. Our task is to choose distinct representatives for the sets $S_1, \ldots, S_{r+1}$.

Case 1

S_{r+1} contains an element which is not a distinct representative of $S_1, \ldots, S_r$. We choose this element as our distinct representative for the set S_{r+1}, and leave the distinct representatives of the sets $S_1, \ldots, S_r$ unchanged.

Proof . . .

Example for Case 1

$$S_1 = \{2, 3\} \qquad S_2 = \{1, 2, 3\} \qquad S_3 = \{2\}$$

Suppose the distinct representatives of S_1 and S_2, to be denoted R_1 and R_2 respectively, have been chosen as

$$R_1 = 3, \qquad R_2 = 1$$

Since S_3 contains the element 2 which is not already a distinct representative, we choose $R_3 = 2$.

Proof . . .

Case 2

All elements of S_{r+1} are distinct representatives for $S_1, \ldots, S_r$. Let e_1 be the first element of S_{r+1}. By assumption, this is the distinct representative of one of the sets $S_1, \ldots, S_r$, say S_{j_1}. Consider now

$$S_1' := (S_{r+1} \cup S_{j_1}) \setminus \{e_1\}$$

If there is a member of this set which is not the distinct representative of one of the sets $S_1, \ldots, S_r$ we call this e_2 and stop for now. If there is not we let e_2 be the first member of S_1' and continue. In the latter case suppose that e_2 is the distinct representative of S_{j_2}, and consider

$$S_2' := (S_{r+1} \cup S_{j_1} \cup S_{j_2}) \setminus \{e_1, e_2\}.$$

Proof . . .

If there is a member of this set which is not the distinct representative of one of the sets $S_1, \ldots, S_r$ we call this e_3 and stop for now. If there is not we let e_3 be the first member of S_2' and continue.

Eventually this procedure must yield an element e_{p+1} which is not the distinct representative for any of $S_1, S_2, \ldots, S_r$, and this must happen for $p \leq r$.

Example for Case 2

Using the same sets as before, suppose $r = 2$, and the distinct representatives have now been chosen as

$$R_1 = 2, \qquad R_2 = 3.$$

As all elements of S_3 are already distinct representatives, we have an example of Case 2.

Proof . . .

The first element of S_3 is 2, so $e_1 = 2$. It is the distinct representative of S_1, so

$$S_{j_1} = S_1.$$

We note $S_3 \cup S_1 = \{2\} \cup \{2, 3\} = \{2, 3\}$ and thus

$$(S_3 \cup S_1) \setminus \{e_1\} = \{2, 3\} \setminus \{2\} = \{3\}.$$

All the elements of this set are already distinct representatives, so we set $e_2 = 3$ and note that then $S_{j_2} = S_2$.

Noting $S_3 \cup S_1 \cup S_2 = \{2, 3\} \cup \{1, 2, 3\} = \{1, 2, 3\}$ we see

$$(S_3 \cup S_1 \cup S_2) \setminus \{e_1, e_2\} = \{1, 2, 3\} \setminus \{2, 3\} = \{1\}$$

This set contains the element 1, which is not a distinct representative. We set $e_3 = 1$, and we are done for the time being.

Proof . . .

Resuming the general argument, by construction the element e_{p+1} must be in S_{j_p}. We let the new distinct representative of S_{j_p} be e_{p+1}. This means that e_p, the old representative of S_{j_p}, is now freed up.

By definition e_p is in S'_{p-1}, and is therefore in one of the sets $S_{r+1}, S_{j_1}, \ldots, S_{j_{p-1}}$. If it is in S_{r+1} we let the representative of S_{r+1} be e_p and we are done. If not then e_p is in S_{j_k} for some $k < p$. We let the new representative of S_{j_k} be e_p, freeing up e_k. This process is repeated, where at each step the index k decreases. Therefore the process must terminate eventually with a freed up representative belonging to S_{r+1} (because eventually e_1, which we know is in S_{r+1}, will be freed up).

□

Proof . . .

Example
We continue the previous example, which had

$$S_1 = \{2, 3\} \qquad S_2 = \{1, 2, 3\} \qquad S_3 = \{2\}$$

$$R_1 = 2 \quad R_2 = 3 \qquad e_1 = 2, \quad e_2 = 3, \quad e_3 = 1.$$

Here $e_3 = 1$ is in S_2, so we make this the new distinct representative of S_2.

This frees up element 3, which is in S_1, and we make this the new distinct representative of S_1.

This now free up the element 2, which is in S_3, so we set $R_3 = 2$ and we are done.

Assignment

Hall's Theorem can be couched in terms of matrices too.

Theorem

Let A be a $0, 1$-matrix. A necessary and sufficient condition for it to be possible to choose a 1 in each row, no two of which are in the same column, is that for each subset of k rows, there are 1's in at least k different columns.

Go back to the previous four examples of matrices and use the above theorem to determine if they are feasible.

Assignment

Lastly, we will model the assignment problem using graphs. We need the folloiwng concept:

Definition

A perfect matching in a graph G is a matching that includes all nodes of G.

Note that G cannot have a perfect matching if G has an odd number of nodes.

One way to solve the assignment problem is to form a bipartite graph from the $0, 1$-matrix of suitabilities, and to seek a perfect matching.

Because such a matching involves all nodes, every person will be assigned to a job and every job will be assigned to a person.

Assignment

Example

From the previous suitability matrix we read off that a possible matching is

$$(a, 1), (b, 2), (c, 3), (d, 4), (e, 5), (f, 6), (g, 7)$$

However, this is not a perfect matching, as it does not involve h and 8. These nodes are exposed.

We therefore identify an augmenting path starting with node h and finishing with node 8.

Assignment

One possibility is

$$h, (1, a), (7, g), 8$$

This matching can be augmented into the larger matching

$$(h1)(a7)(g8).$$

By removing matched edges $(a1)$ and $(g7)$ from our very first matching, and then adding our augmented matching $(h1)(a7)(g8)$ we get the perfect matching

$$(a, 7), (b, 2), (c, 3), (d, 4), (e, 5), (f, 6), (g, 8), (h, 1).$$

Maximum matchings and vertex covers

Supposing that for some suitability matrix we cannot find a feasible assignment that includes all jobs and people.

In this case we could still try to match as many people to jobs as possible.

In terms of $0, 1$-matrices, we try to find the maximum number of independent 1's, i.e., the maximum number of 1's from distinct rows such that no two 1's are in the same column.

For an $n \times n$ matrix, we can then solve the job assignment problem provided the maximum number of independent 1's is equal to n.

Maximum matchings and vertex covers

Theorem: König-Egerváry theorem (matrices)

Let A be a $0,1$-matrix. The maximum number of independent 1's is equal to the minimum number of lines which one can draw to cover (cross out) the 1's.

Lines which cover the 1's must be drawn completely through rows or columns.

Maximum matchings and vertex covers

Example

Consider the $0, 1$-matrix

$$\begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \end{bmatrix}$$

In this example the maximum number of independent 1's is two: the elements in positions $(1, 2)$ and $(2, 1)$ for example.

On the other hand, the 1's can be covered by lines by ruling through the first row and first column, which is a total of 2 lines.

This is consistent with the theorem.

Maximum matchings and vertex covers

Now try these examples

$$\begin{bmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \end{bmatrix} \qquad \begin{bmatrix} 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}$$

Maximum matchings and vertex covers

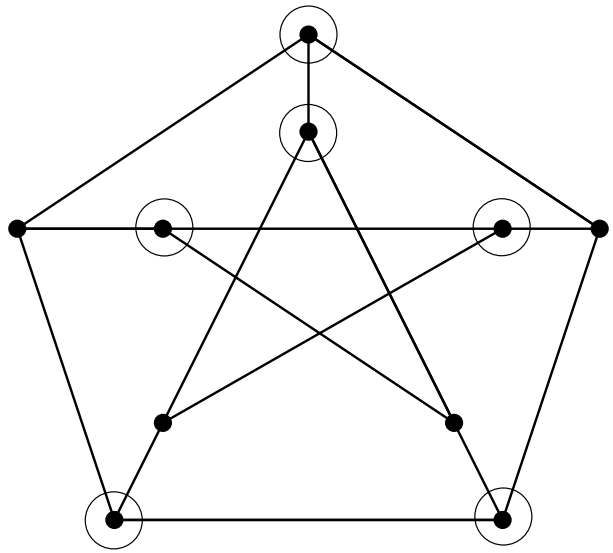
The König-Egerváry theorem is usually stated in terms of graphs.

We first need the following definition:

Definition

A vertex cover of a graph is a set of vertices such that every edge of the graph is incident to at least one of the vertices in the cover.

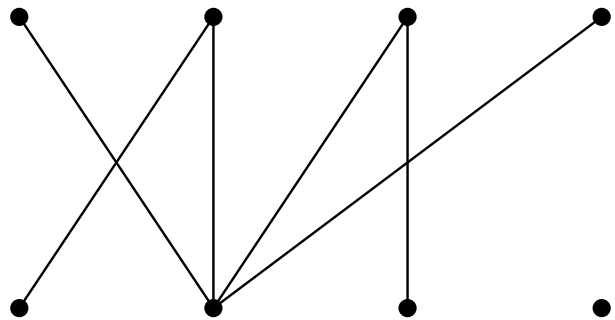
Maximum matchings and vertex covers



The circled nodes consitute a vertex cover of size 6 for the above graph. Compare this with the maximum size of a matching of this graph.

Note that this graph is not bipartite. How can you tell?

Maximum matchings and vertex covers



Compare the same two quantities for this graph. What do you notice?

Maximum matchings and vertex covers

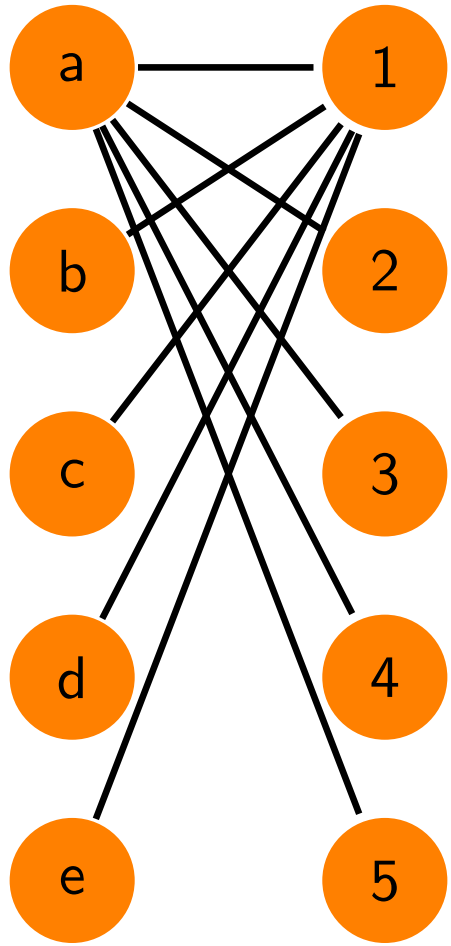
Theorem: König-Egerváry theorem (bipartite graphs)

In a bipartite graph, the maximum size of a matching is equal to the minimum size of a vertex cover.

In terms of matrices, a vertex cover in the corresponding bipartite graph is equivalent to a set of columns and rows that cover all 1's.

Also, a matching in the corresponding bipartite graph is equivalent to a set of independent 1's in the matrix.

Example



Here vertices a and 1 make up the minimum vertex cover, while a maximum matching is $(a, 2), (b, 1)$.

Maximum matchings and vertex covers

A systematic way to test if a matching is of maximum size (and to increase its size if it is not a maximum) follows from the following result.

Theorem: Berge's Lemma)

M is a maximum matching in a bipartite graph G if and only if M is not part of an augmenting path.

Proof

First we will show that if M is a maximum matching then M does not have an augmenting path. This will be done by showing the contrapositive: that if M is part of an augmenting path then M is not a maximum.

Proof . . .

Suppose that G has an augmenting path $x_1(y_1, x_2), (y_2, x_3) \cdots (y_n, x_{n+1}), y_{n+1}$. This in turn gives the matching $(x_1y_1)(x_2y_2) \cdots (x_{n+1}y_{n+1})$, which has one more edge in the matching than M. Therefore M is not maximal.

Next we will show that if M is not part of an augmenting path then M is of maximum cardinality. To this end, suppose there is a maximum matching M^* of size bigger than M. By assumption M does not have an augmenting path. Also, M^* does not have an augmenting path because it is of maximum cardinality.

Proof . . .

Consider the graph G' obtained by taking the union of the two edge-sets M and M^*. A vertex in G' can have at most two incident edges, one from M and one from M^*. Therefore G' consists of even cycles with edges alternating in M and M^*, and of paths with edges alternating between M and M^*. Since M^* is bigger than M, there must be at least one path which starts and ends in M^* – this would be an augmenting path for M, which is a contradiction.

Weighted Assignment

Our next example is a generalisation of the Assignment problem.

We will model this problem using matrices, but it can certainly also be modelled by graphs.

Suppose there are 4 jobs to be done by 4 different people. We devise a rating system using the numbers 0 up to 9. A rating of 0 means "perfectly suited" and a rating of 9 means "completely unsuited", with all grades in between.

For each person a 4-tuple can be formed out of their rating for the 4 jobs in order, together forming a weighted suitability matrix.

Weighted Assignment

$$\begin{bmatrix} 3 & 1 & 0 & 2 \\ 0 & 2 & 8 & 5 \\ 0 & 4 & 1 & 2 \\ 1 & 7 & 3 & 6 \end{bmatrix}$$

For instance, in this example, the suitability of Person 1 for Job 3 is 0, and the suitability of Person 3 for Job 4 is 2.

The weighted assignment problem is to choose an entry from each row, no two of which are in the same column, such that their sum is a minimum.

In graph theoretical terms this problem is also called the minimum weight perfect matching problem.

Algorithmic solution

Step 1

Subtract the smallest (non-zero) number from each row in turn.
Then subtract the smallest number from each column in turn.

This is justified since subtracting a constant from any row or column does not change the choice of elements which solve the weighted assignment problem.

The reason is that it changes the total sum by exactly the amount subtracted from the row or column.

Step 2

Cover the 0's by lines using the minimum number of rows and columns. If n rows or columns are covered then STOP.

For small matrices finding the minimum line cover be done by inspection, but more generally it requires its own strategy.

We are attempting to find n independent 0's. Therefore the reason we don't stop if there are less than n lines covering the 0's is that the König-Egerváry theorem tells us that there are less than n independent 0's.

Step 3

Let l denote the smallest number which is not covered by any line.

1. subtract l from all uncrossed numbers;
2. leave numbers which are crossed once unchanged;
3. add l to all numbers which are crossed twice.

Again, this leaves the solution unchanged since adding a constant to any row or column does not change the choice of elements which solve the problem.

We then return to Step 1. The algorithm always yields n independent 0's after a finite number of repetitions. This can be seen by showing that on each application of the procedure the sum of the entries in the matrix decreases.

Let a be the number of uncrossed numbers, b be the number crossed once, c the number crossed twice. Also, let r be the total number of crossing lines. Then

$$a + b + c = n^2, \qquad b + 2c = rn$$

Subtracting the second equation from the first gives

$$a - c = n(n - r) > 0.$$

Since l is subtracted from a numbers, and added to c numbers, the total sum of the entries must decrease, because $a > c$.

Step 4

Go to Step 2 with this new matrix.

Example

Solve the optimal assignment problem for

$$\begin{bmatrix} 3 & 1 & 0 & 2 \\ 0 & 2 & 8 & 5 \\ 0 & 4 & 1 & 2 \\ 1 & 7 & 3 & 6 \end{bmatrix}$$

Questions

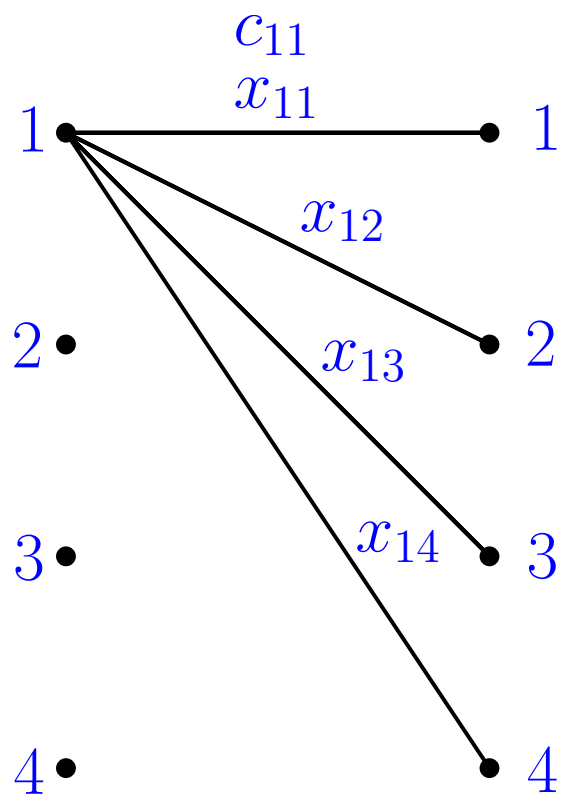
The above study leaves open the question of finding the minimal cover of 0's for a general $0, 1$-matrix, and of locating the independent 0's. For small matrices we used inspection.

In graph theoretic terms, the first question is the same as seeking minimum vertex covers in bipartite graphs, and the second as seeking matchings of maximum size. For the latter problem we can use Berge's lemma.

For large matrices Berge's lemma is an effective way to determine the maximum number of independent 0's.

LP formulation of the problem

The weighted assignment problem can be formulated as an integer linear program (ILP) as follows:



ILP for the weighted perfect matching (assignment) problem

$$\min \sum_{i=1}^{n} \sum_{j=1}^{n} c_{ij} x_{ij}$$

$$\sum_{j=1}^{n} x_{ij} = 1 \text{ for } i = 1, ..., n$$
$$\sum_{i=1}^{n} x_{ij} = 1 \text{ for } j = 1, ..., n$$
$$x_{ij} \in \{0, 1\} \text{ for } i, j = 1, ..., n$$

The formulation can be shown to be equivalent to the following linear program (LP):

$$\min \sum_{i=1}^{n} \sum_{j=1}^{n} c_{ij} x_{ij}$$

$$\sum_{j=1}^{n} x_{ij} = 1 \text{ for } i = 1, ..., n$$
$$\sum_{i=1}^{n} x_{ij} = 1 \text{ for } j = 1, ..., n$$
$$x_{ij} \geq 0 \text{ for } i, j = 1, ..., n$$

Note that the integer constraints are actually not needed.