Network optimisation

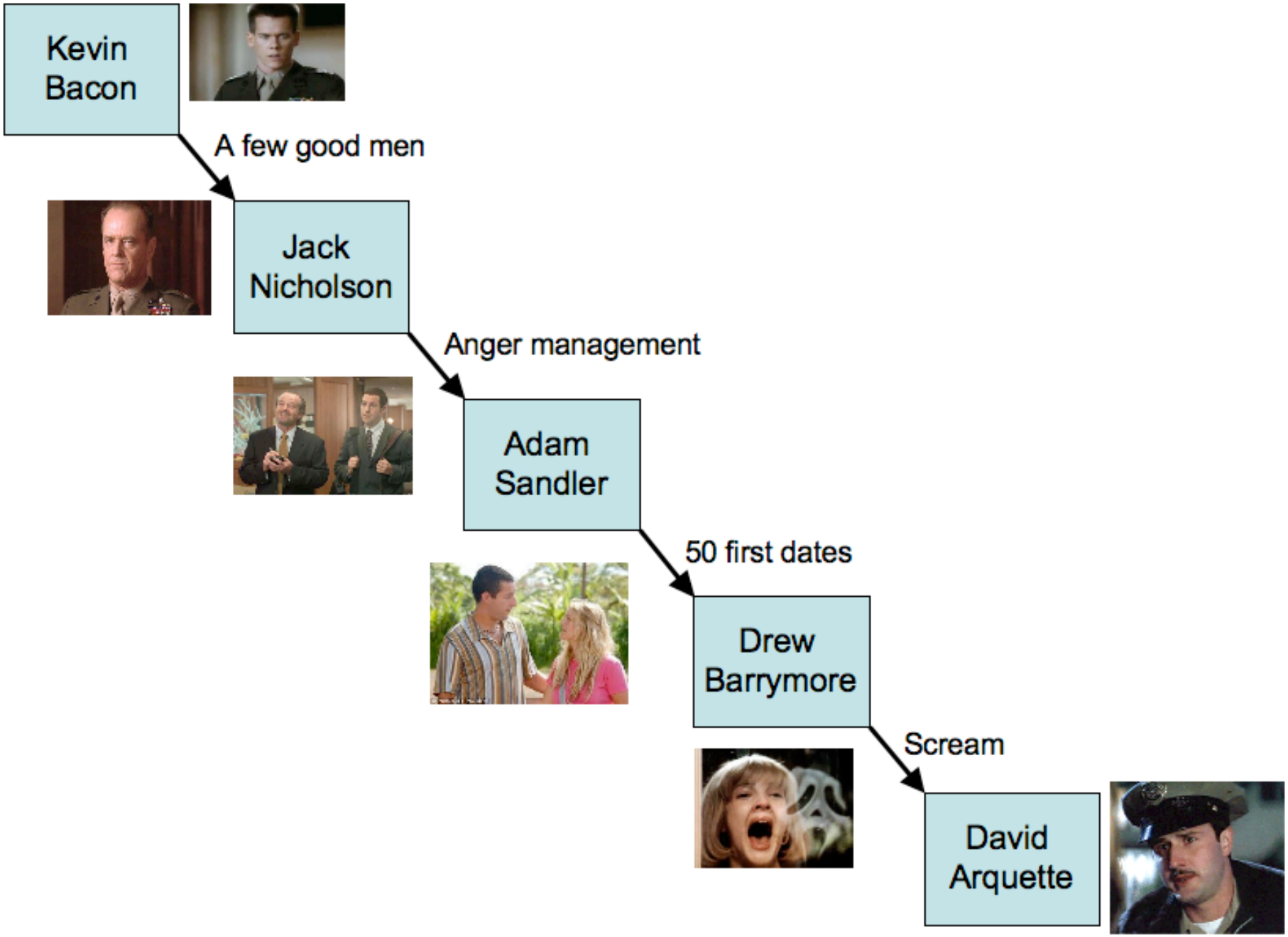
§1 – Social Network Analysis

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 edges 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].

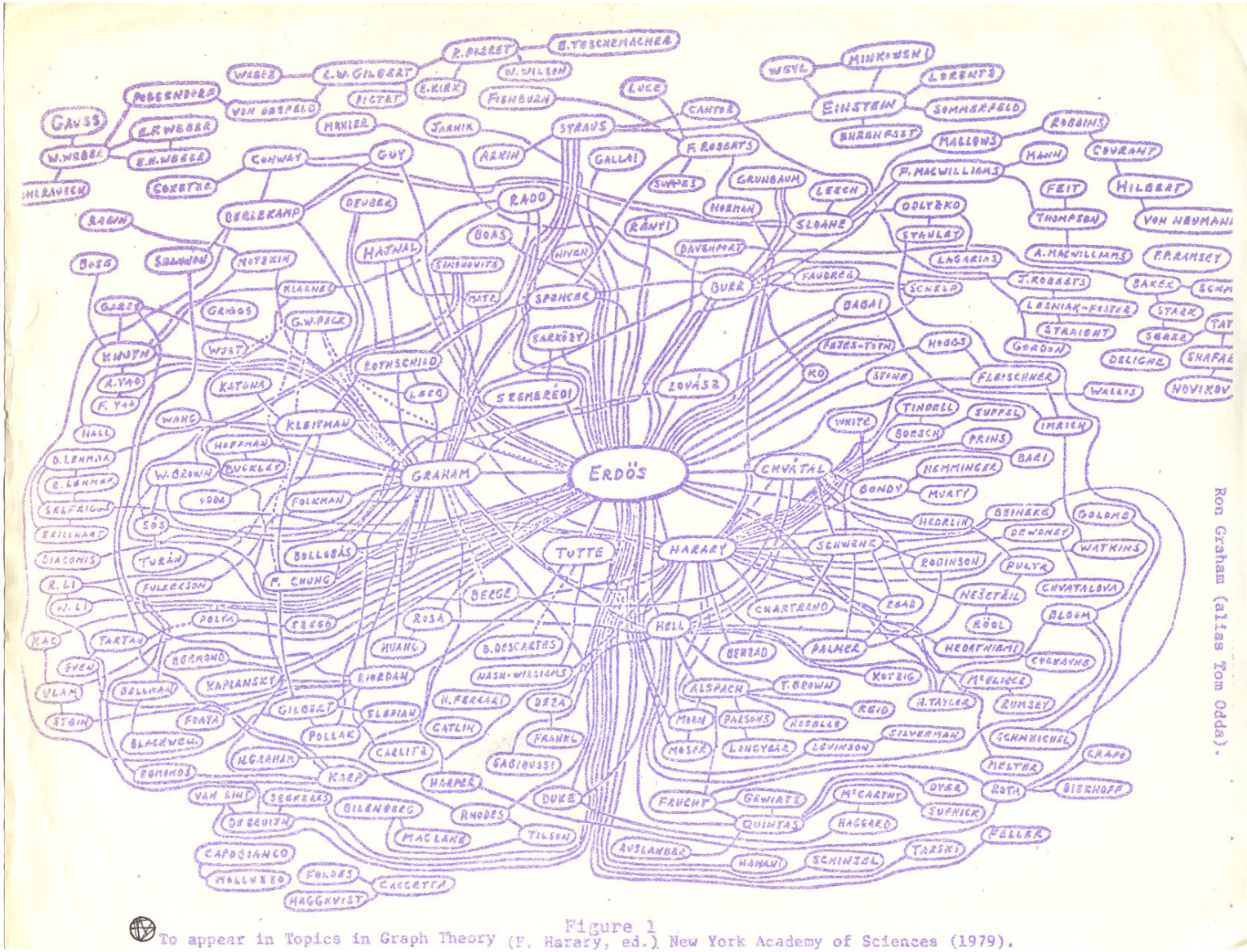


Figure 1

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

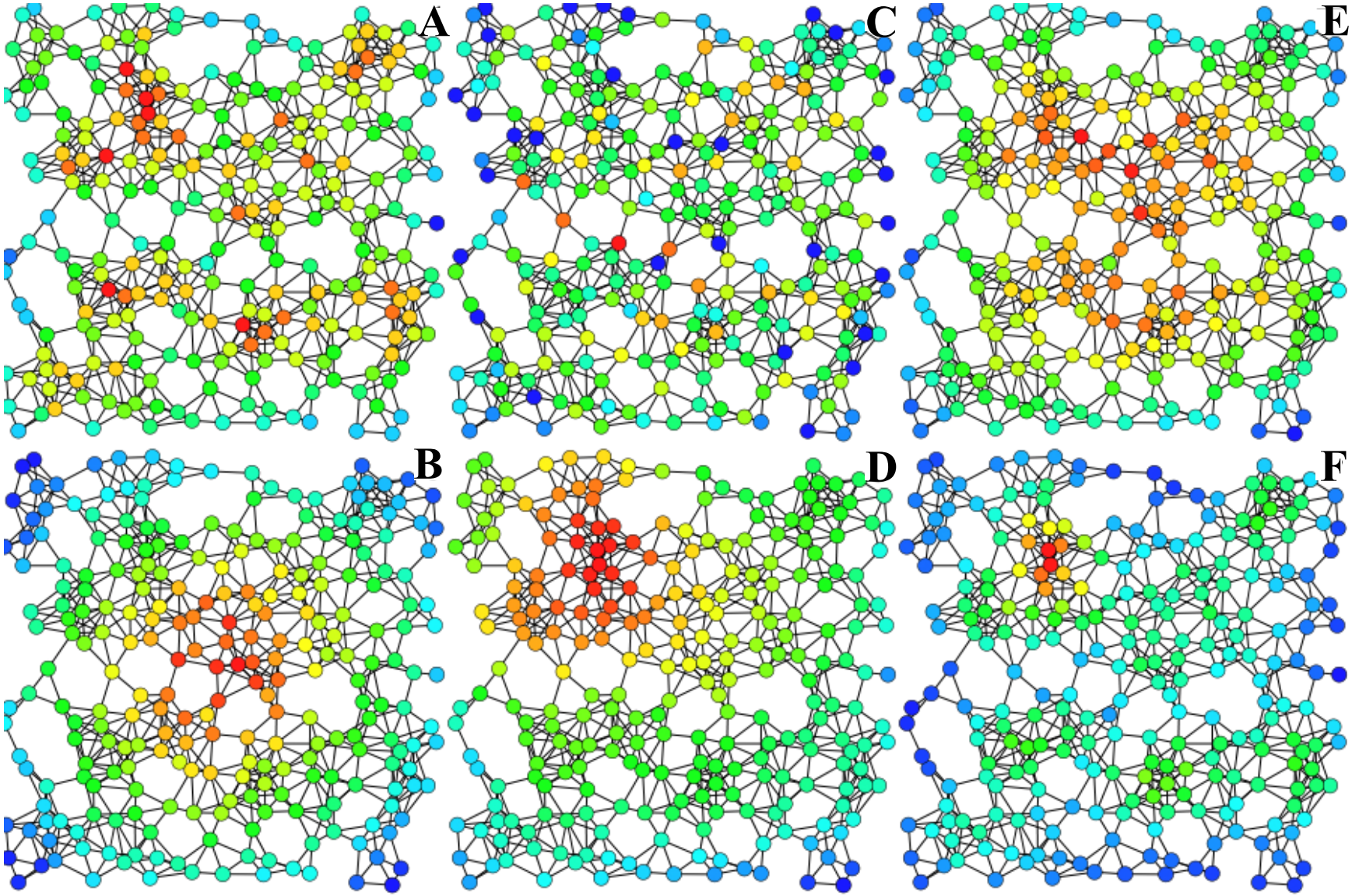
D) Eigenvector centrality - measure of "influence" of a node in a network. Eg. Google PageRank: "PageRank works by counting the number and quality of links to a page to determine a rough estimate of how important the website is. The underlying assumption is that more important websites are likely to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

§1.1 – 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

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

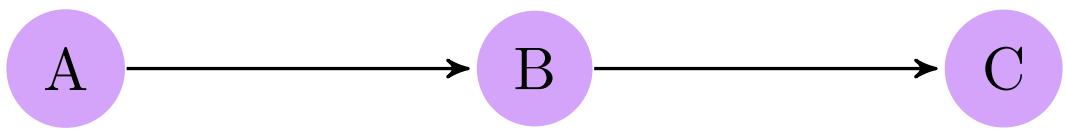
Here the sum is over all pairs of distinct nodes s, t of the network, excluding v.

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

μ_{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

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

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

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

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,

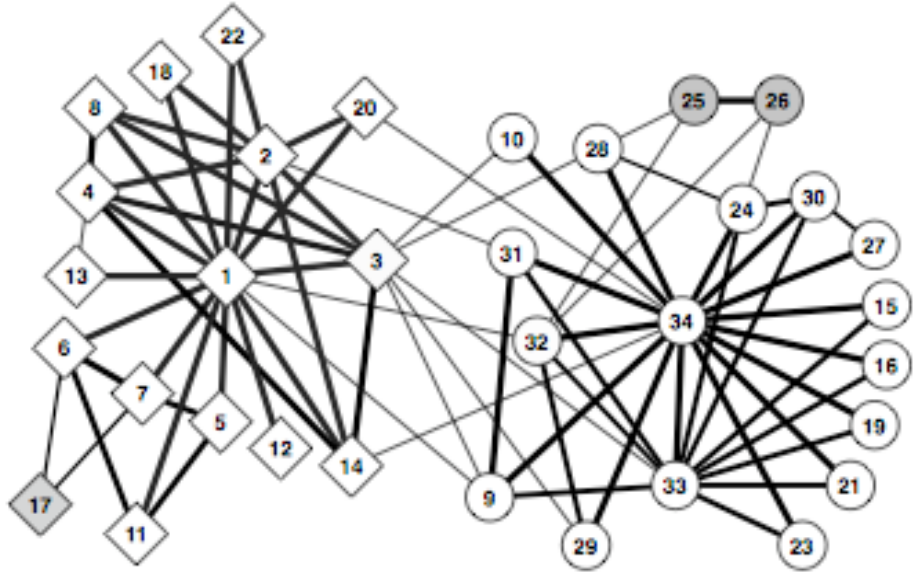
$$B(e) = \sum_{s \neq t} \frac{\mu_{st}(e)}{\mu_{st}},$$

where the sum is over all pairs of distinct nodes s, t. Here $\mu_{st}(e)$ 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
33
30
28
27
24
23
21
19
16
15
10
32
29
31
6
26
25
12
13
22
20
18
4
3
14
8
2

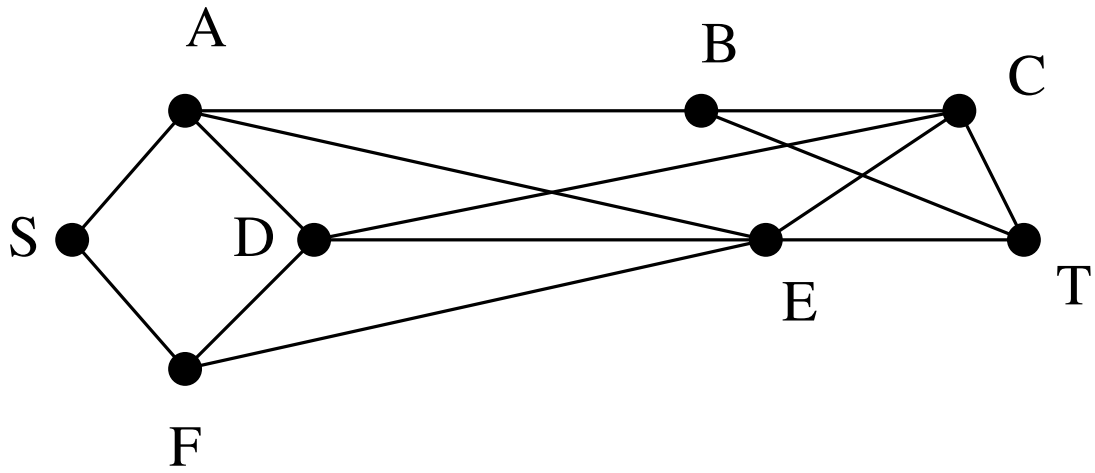
§2 – Finding shortest paths

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

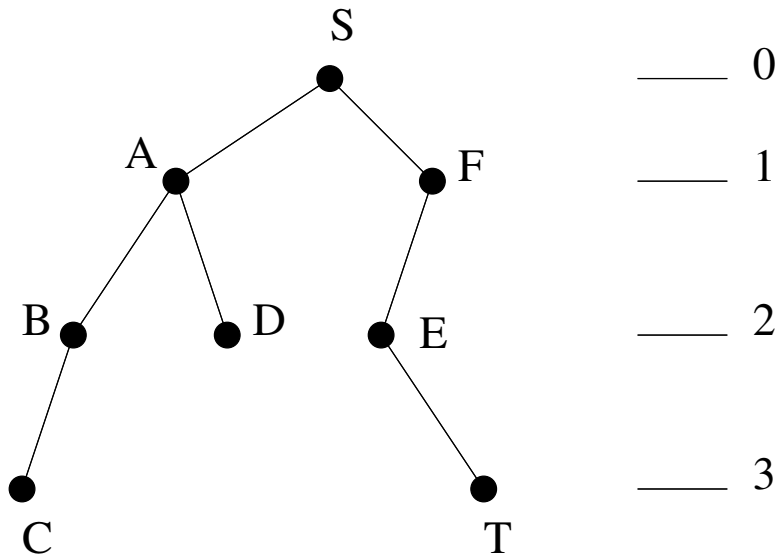
Algorithm (Breadth-First Search):
Given nodes S, T find a path (if it exists) from S to T which uses the least number of edges.

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 $\mu(S|T) := \mu_{ST}$ from S to T (note: we are using new notation to avoid subscripts).

Recursive approach for the number of shortest paths

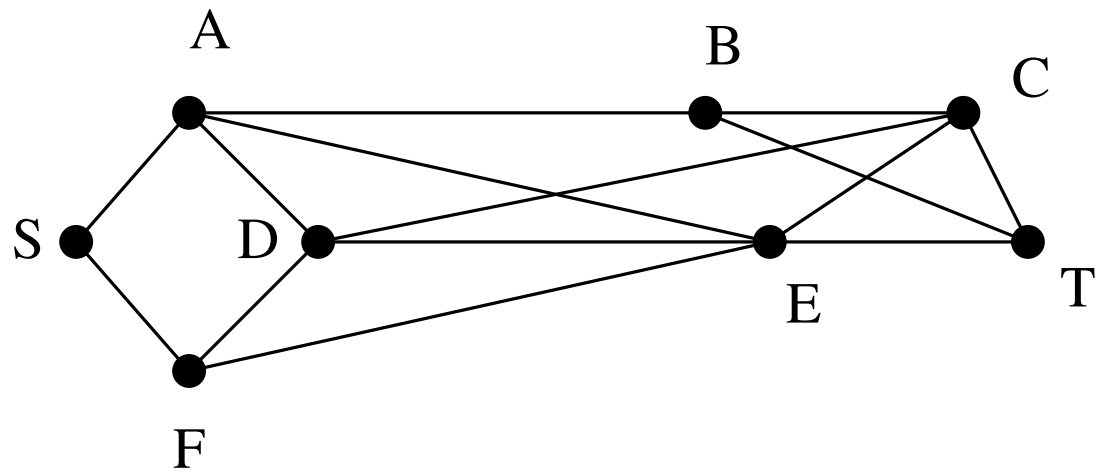
Algorithm

Given nodes S, T find $\mu(S|T)$, the number of shortest paths from S to T.

1. Set $i := h(T)$, the height of T, and $\mu(T|T) := 1$. 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

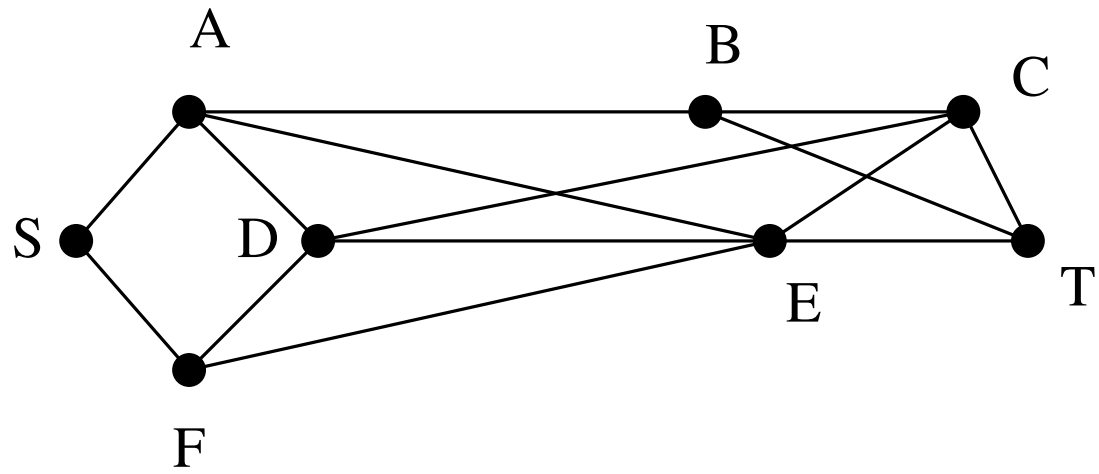
$$\mu(v|T) := \sum \mu(u|T)$$

   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. If not, decrease i to $i - 1$ and go to Step 2.



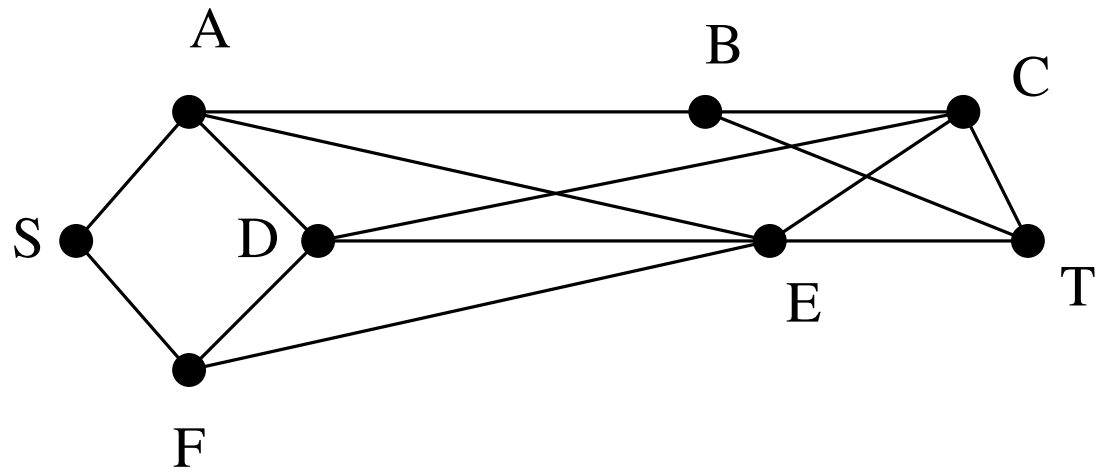
For $i = 3$ we have

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



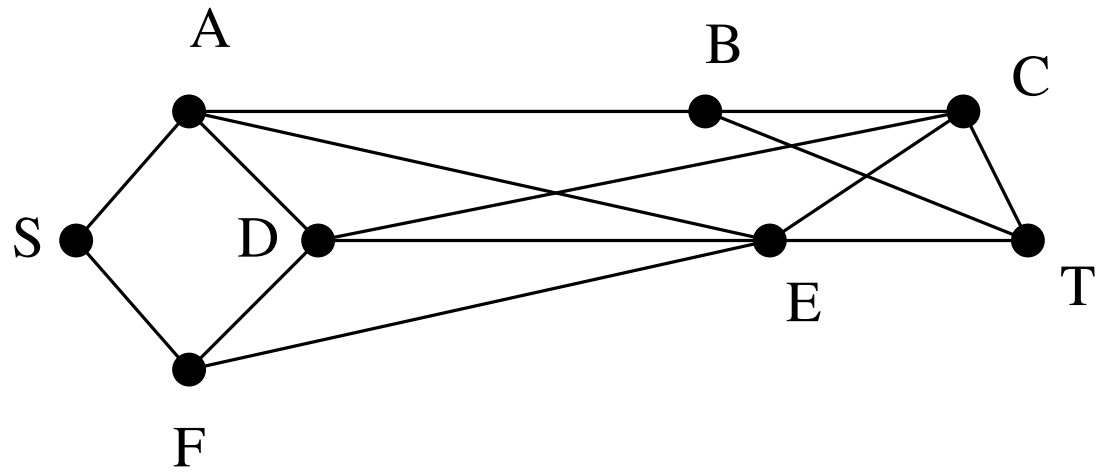
For $i = 2$ we have

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



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

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

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

Recursive 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

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

so that $[a_{ij}]$ is similar to the incidence matrix (except for the ∞ entries).

Let $h^{(m)}(S|j)$ be the length of the shortest path from vertex S to vertex j using no more than m edges. Then

$$h^{(0)}(S|S) = 0, \qquad h^{(0)}(S|j) = \infty \quad (j \neq S).$$

We then have the recursive formula:

$$h^{(m+1)}(S|j) = \min_k \{h^{(m)}(S|k) + a_{kj}\}.$$

To find $h(S|T)$ for some pair of nodes S, T we therefore calculate $h^{(m)}(S|T)$, where m chosen such that $h^{(m+i)}(S|T) = h^{(m)}(S|T)$ for all integers $i \geq 0$. For example, if

$$h^{(1)}(S|T) = h^{(2)}(S|T) = h^{(3)}(S|T) = h^{(4)}(S|T) = \infty$$

and

$$h^{(m)}(S|T) = 5 \qquad \text{for } m \geq 5.$$

then $h(S|T) = 5$ then

Min-plus Matrix Algebra

In studying activity networks we came across a similar recurrence, only it involved maximisations. As in that case, a type of matrix multiplication can be used to implement the recurrence.

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

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

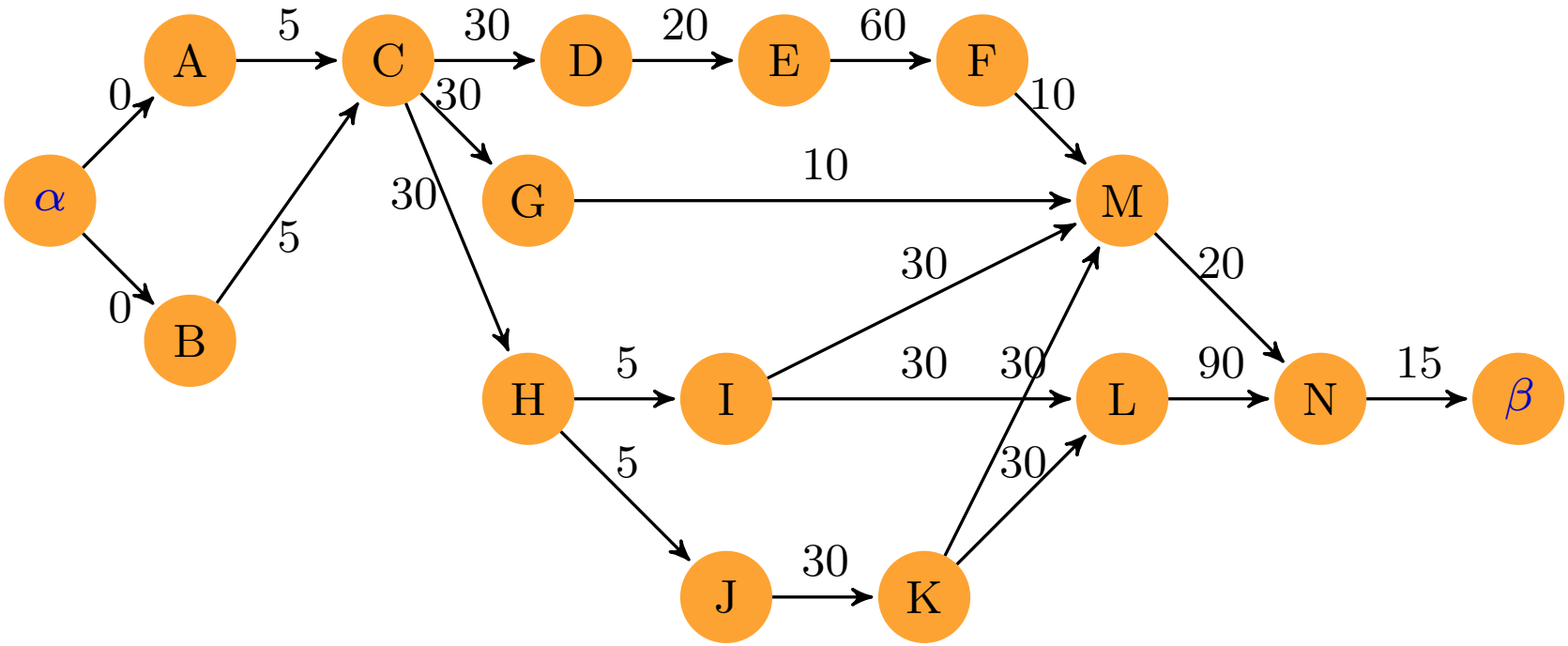
Notes

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

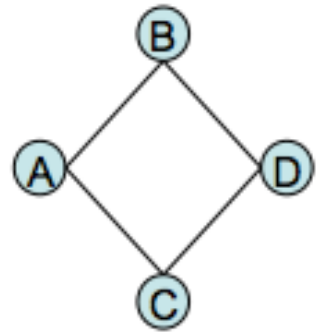
$$[0 \; \infty \cdots \infty] \otimes_{\min} A^{N-1}.$$

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

True or false: In an AON network, the earliest start time of any activity corresponds to the longest weighted path length from α to the node for the activity?



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

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

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

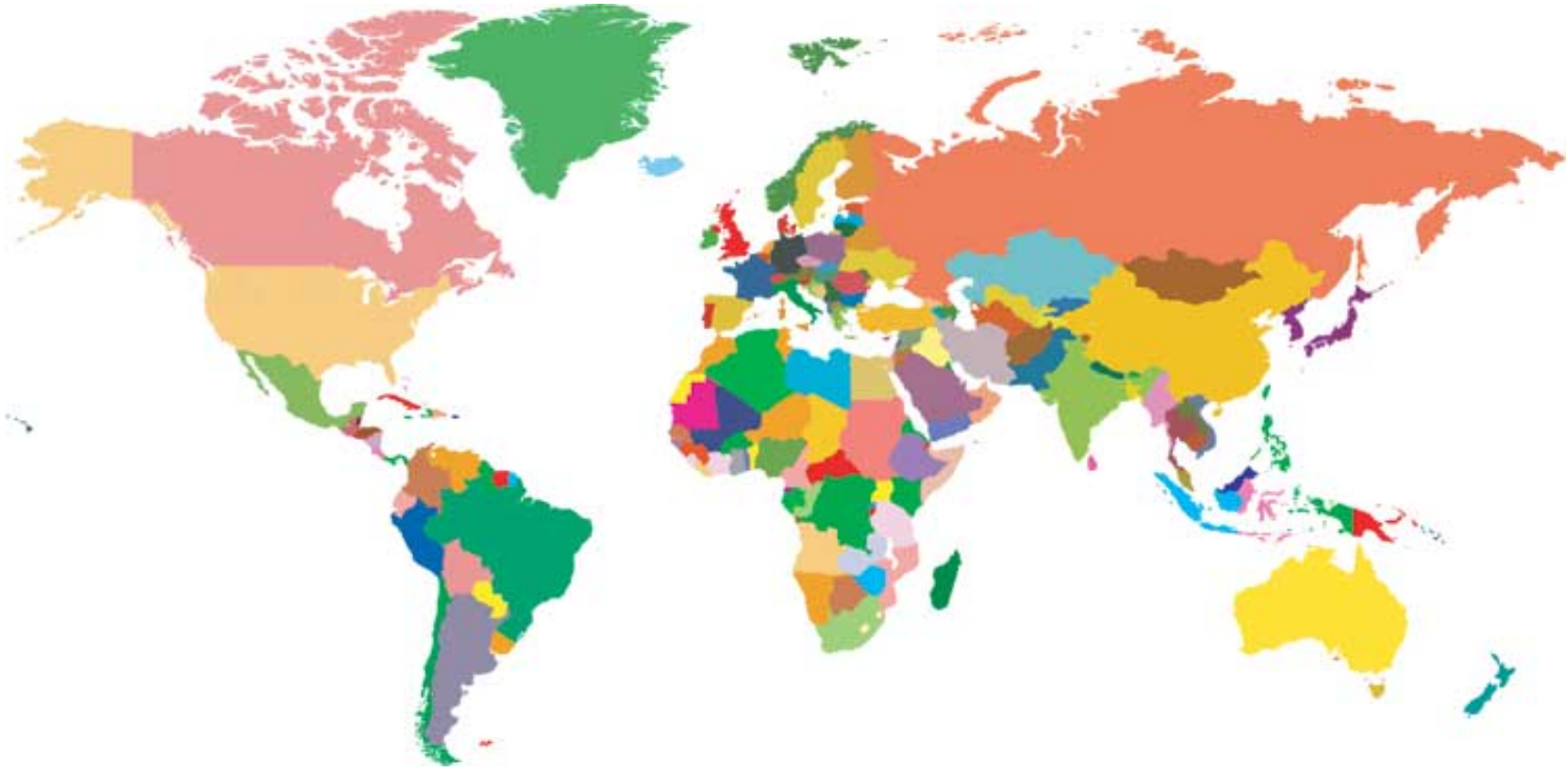
§3 – Graph Colourings and Interval Graphs

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

This smallest number of colours needed to colour a graph is called the vertex chromatic number of the graph.

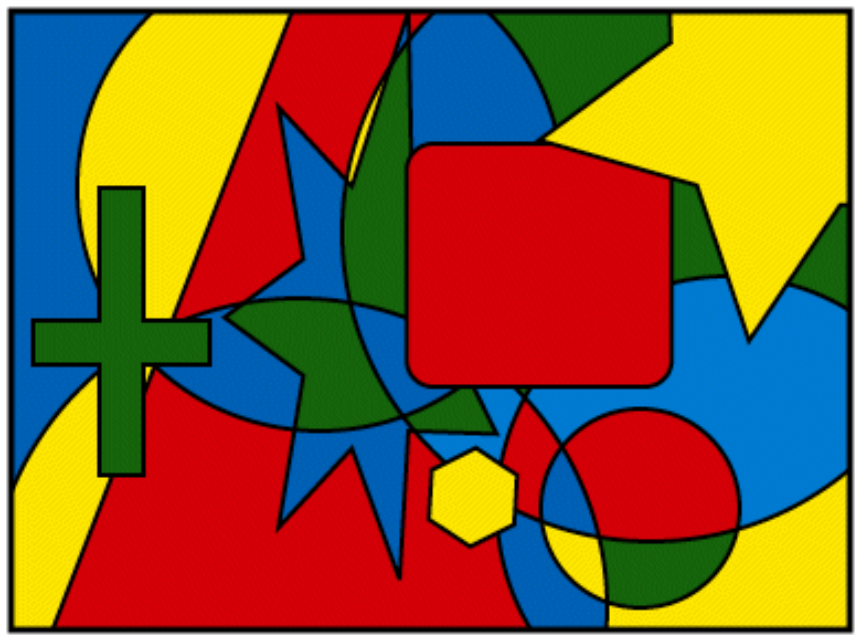
In the graph colouring problem we wish to find the vertex chromatic number of a graph or class of graphs.

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?



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

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.

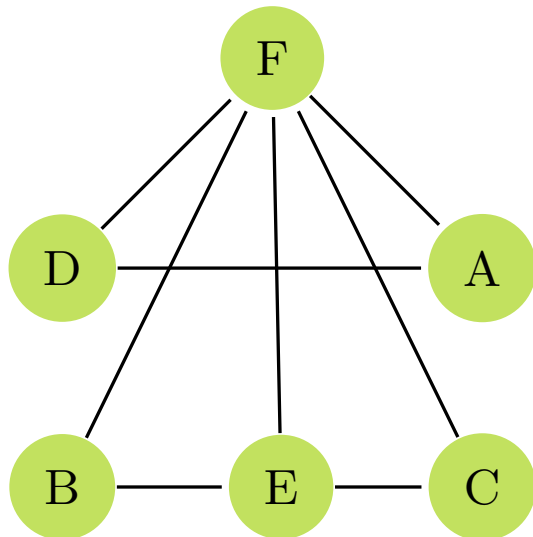


Graph colourings have many applications. The first example we look at is hotel bookings.

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 at the same time.



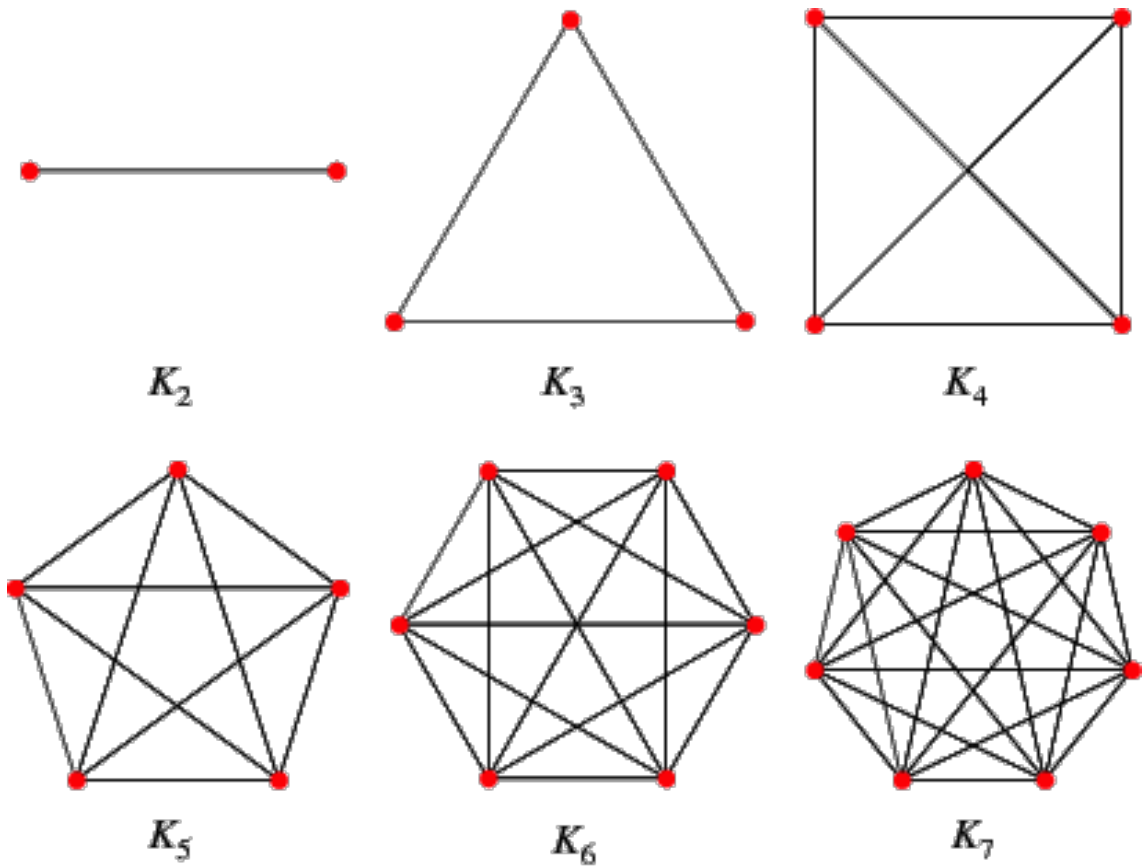
Our problem is to determine the smallest number of rooms required at the hotel to accommodate all the bookings.

Suppose we colour the vertex of the graph according to the rooms the guests will stay in. Then we must colour according to the rule: no vertices connected via an edge have the same colour.

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

Definition

A clique in a graph is a collection of vertices such that each vertex is connected to each other vertex by an edge. Cliques are also known as complete graphs.

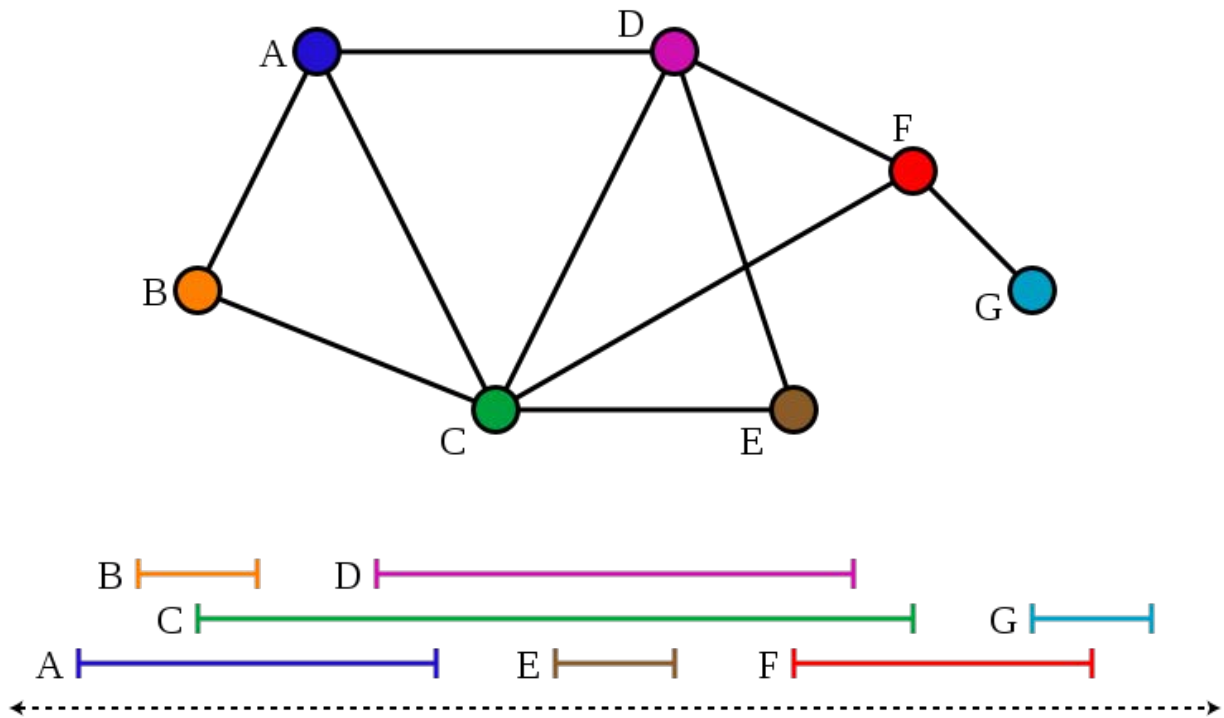


Theorem

$$vertex\ chromatic\ number \geq size\ of\ largest\ clique$$

Definition

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 overlap (intersect).



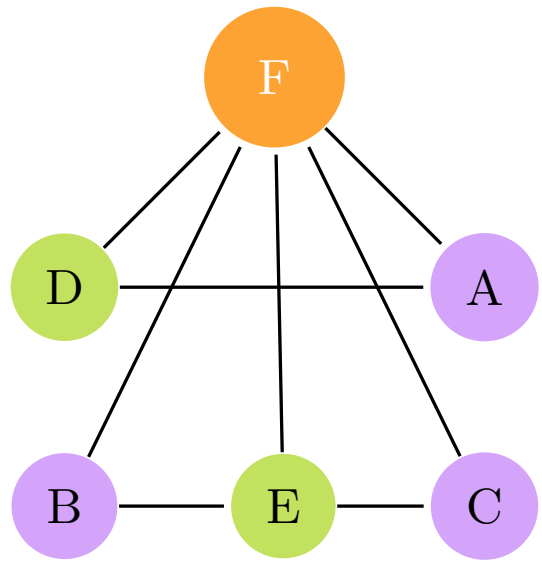
Observe that bookings graphs are interval graphs.

Theorem

For interval graphs it is always true that

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

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



Another applied problem where vertex colouring arises is in situations of conflict. An example is fighting fish, where certain fish are incompatible and so must reside in different tanks.

Theorem (Brook's Theorem)

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

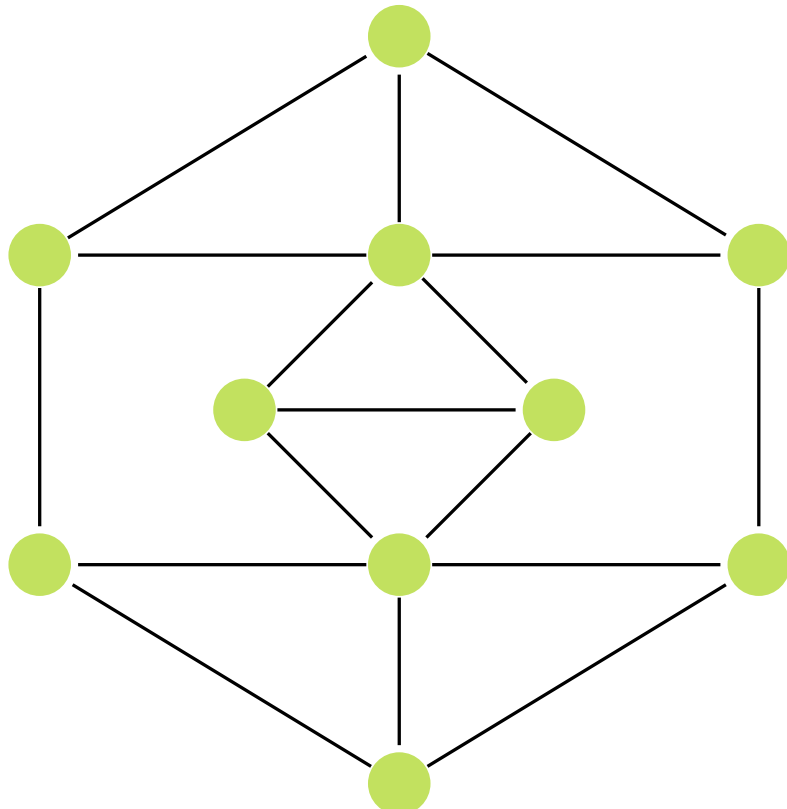
$$\textit{vertex chromatic number} \leq \textit{maximum degree of any node},$$

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

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

Exercise

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



Use trial and error to find the exact chromatic number.

§4 – Matching

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.

One application of graph matchings relates to 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 $x_1, x_2, \dots, x_m$ and n classes $y_1, y_2, \dots, y_n$.

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 a $0,1$-matrix 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.

In fact there is a theorem which tells us that:

A matching can always be found which reduces the sum of entries of the row and or column with the maximum sum

Performing each step by reducing the maximum sum leads to an optimal matching.

To get an optimal matching, another option at Step 2 would therefore have been:

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

Edge-colourings

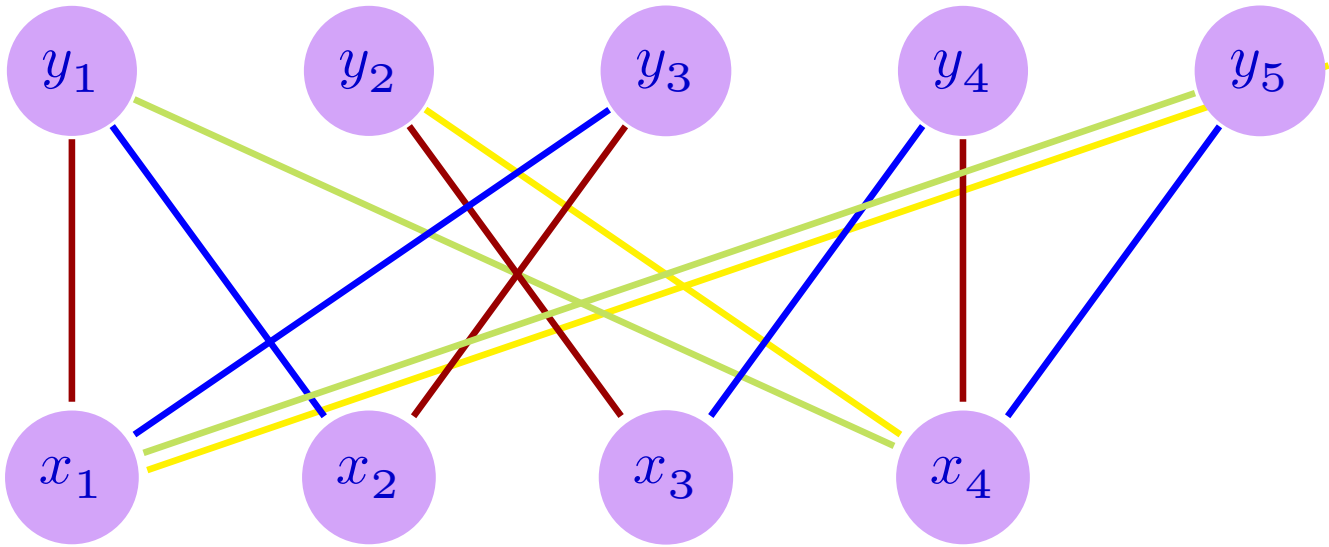
Definition

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.

Observe therefore that a set of edges of the same colour in a graph must be independent.

Definition

The node-set of a bipartite graph G can be partitioned into two subsets such that every edge of G joins nodes in different subsets.



Another viewpoint of the timetabling problem is that the matchings of teacher-class allocation specifies an edge colouring of a bipartite graph (note that there are two edges between x_1 and y_5).

The edges with the same colour correspond to the matchings obtained by decomposing the table of teacher-class allocations.

Theorem

In a bipartite graph a matching that involves all the vertices of maximum degree can always be found.

Proof

We will use induction on the number of edges m. The result is obviously true for $m = 1$.

Suppose the result holds true for any bipartite graph with a total of m edges. Let G be such a graph. We denote the maximum degree of G by Δ and we let M be a matching in G that involves all vertices of degree Δ. Suppose now that we add an edge e to G to get a new graph G' with $m + 1$ edges.

Proof...

We consider four cases depending on the degrees of the endpoints, say x_a, y_a, of e in G'.

1. Suppose that the degrees of both x_a and y_a are less than Δ in G'. Then x_a and y_a are not of maximum degree in G' and therefore M is the required matching in G' involving all vertices of maximum degree.
2. Suppose that the maximum degree of G' is $\Delta + 1$. This means either x_a or y_a (or both) are of degree Δ in G, and therefore occur in M. Hence all vertices of maximum degree in G' occur in matching M.

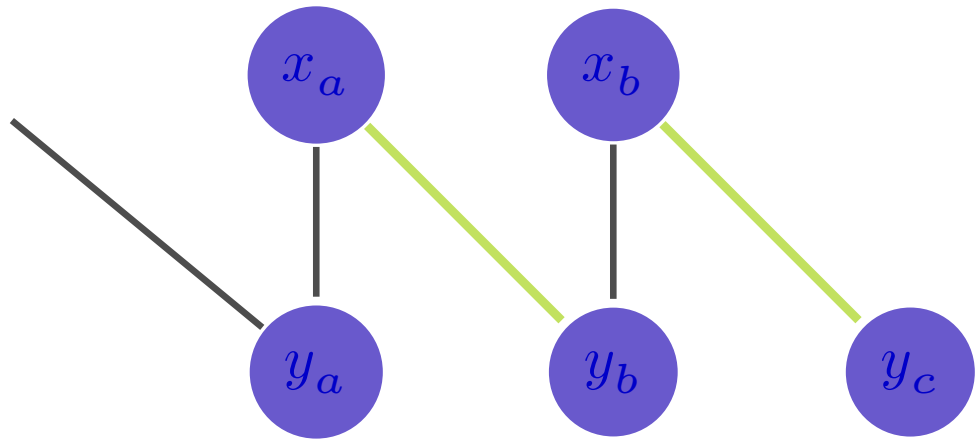
Proof...

3. Suppose that either x_a or y_a (or both) are of degree Δ in G', and suppose that M either includes both x_a and y_a or neither of them. If M includes both x_a and y_a then there is nothing to do. If M includes neither x_a nor y_a then we can add e to M to get a new matching in G' involving all nodes of degree Δ.

4. Finally, suppose that one of the endpoints of e, say y_a, is of degree Δ in G' and doesn't belong to M, and the other endpoint, x_a, does belong to M. Here x_a may or may not be of degree Δ. Note then that we can't simply add e to M to form a new matching because then x_a would be an endpoint of two edges in the matching, which is not allowed. Since G is bipartite the edges of M must form an alternating path, for example

$$y_a(x_a y_b)(x_b y_c)$$

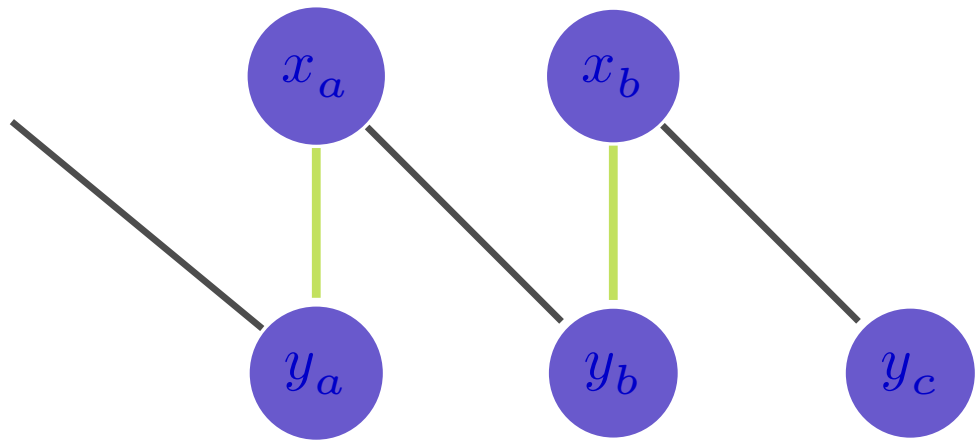
where y_c has degree less than Δ, and the brackets indicate a matching.



Edges of matching shown in green

Proof...

Since the degree of y_c is less than Δ we don't require it to be involved in the matching. We simply shift the brackets to obtain a matching involving y_a to get $(y_a x_a)(y_b x_b) y_c$



The four cases cover all possibilities. In each case it was found that a matching involving the vertices of maximum degree can be found in G'. Since any graph on $m+1$ edges can be constructed by adding an edge to a graph of m edges, the theorem is proved by induction.

Corollary

A bipartite graph of maximum degree Δ can be edge coloured using Δ different colours, which is the smallest possible.

Proof

Perform a matching according to the previous theorem so as to involve all vertices of degree Δ. Colour these using colour Δ say, and draw up a new bipartite graph with these edges removed. This new graph has maximum degree equal to $\Delta - 1$, so we repeat.

Corollary

In the teacher-class timetabling problem, the smallest number of timetable slots needed is equal to the maximum of the number of subjects a given teacher must teach and the number of times any one class must be taught.

In other words, it is equal to the maximum of the row or column sums of the matrix of periods, or the maximum vertex degree of the corresponding bipartite graph.

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

We ask, would it be possible to have exactly three classes in each of the four time slots? The answer is yes, due to the following result.

Theorem

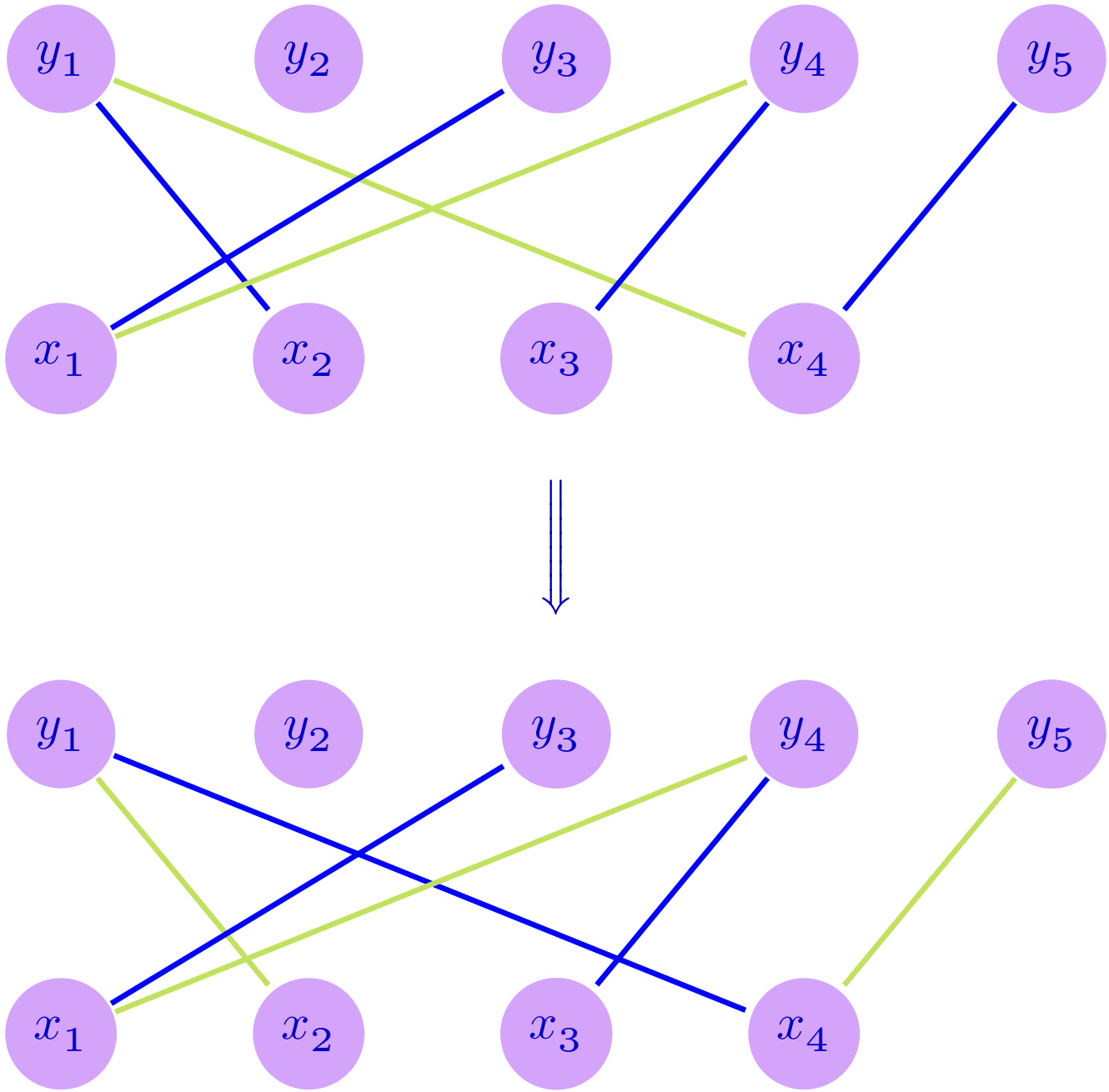
Suppose M_1 and M_2 are disjoint matchings in a bipartite graph G, with $|M_1| > |M_2|$. Then there are disjoint matchings $\tilde{M}_1$ and $\tilde{M}_2$ of G such that $|\tilde{M}_1| = |M_1| - 1$, $|\tilde{M}_2| = |M_2| + 1$ and $\tilde{M}_1 \cup \tilde{M}_2 = M_1 \cup M_2$.

Proof

Forming $M_1 \cup M_2$, we see that each resulting component 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.

If we change this matching as in the previous proof such that it begins and ends in M_2, we get the desired new matchings.

□


y_1
y_2
y_3
y_4
y_5
x_1
x_2
x_3
x_4
y_1
y_2
y_3
y_4
y_5
x_1
x_2
x_3
x_4

§5 – Assignment

Closely related to the above timetable problems is a problem on assigning people to jobs they are suited to. For example, suppose there are eight people $a, b, \ldots, h$ and eight jobs $1, 2, \ldots, 8$. 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.

For example

	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

One way to solve the problem is to form a bipartite graph from this $0,1$-matrix, and to seek a matching that involves all jobs and all people (a so-called perfect matching).

Another is to perform a single 1's decomposition so that there is a 1 in each row, no two of which are in the same column.

Yet another is to associate with each person the set of jobs they are suited to:

$$\{1,4,5,7\}, \quad \{2,6\}, \quad \{1,3,5,8\}, \quad \{2,4,6\}, \quad \{2,3,5,6\}$$

$$\{2,4,6\}, \quad \{1,5,6,7,8\}, \quad \{1,4,6\},$$

and then to choose distinct representatives from these sets to assign to the people.

We begin with an algorithm based on matchings. Let G be a bipartite graph with partite sets X, Y and let M be a matching in G.

Definition

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

(We saw examples of alternating paths in the previous section)

Definition

An augmenting path in G is a path of the form xPy, where $x \in X$, $y \in Y$, and P is an alternating path with first and last edges in M.

Observe that the matching within an augmenting path can easily be extended to a larger matching.

From the table, we read off that a possible matching is

$$(a1)(b2)(c3)(d4)(e5)(f6)(g7)$$

However, this is not a perfect matching, as it does not involve h and 8. We therefore identify an augmenting path starting with node h and finishing with node 8.
One possibility is

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

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

$$(a7)(b2)(c3)(d4)(e5)(f6)(g8)(h1).$$

Next we consider the problem in terms of distinct representatives. A fundamental question in relation to this problem is, under what conditions involving the $0, 1$-matrix of job suitability can we be sure that it is possible to choose a 1 in each row, no two of which are in the same column?

A necessary condition is that for any k rows, there are 1's in at least k different columns. It turns out that this is also a sufficient condition.

Theorem (Hall's theorem)

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

Theorem (Hall's theorem in terms of $0, 1$ matrices)

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.

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 $\{1, 2, 3, 4, 5\}$.

Proof of Hall's theorem

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. For this 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

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

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

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

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.

§5.1 – Maximum Covers of 0, 1-matrices

According to Hall's theorem written in terms of $0, 1$-matrices, there is a solution to the job assignment problem if and only if for any k rows there are 1's in at least k columns.

It is possible to make this criteria more practical to use.

For a $0, 1$-matrix, let the maximum number of independent 1's be 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.

Theorem (König-Egerváry max-min theorem)

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 the 1's.

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

As a rule of thumb, the minimum number of lines can be identified by drawing lines through rows and columns with the most number of 1's at each stage. This only works when there are NO TIES AT ANY STAGE!

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.

To prove the König-Egerváry max-min theorem we first need two lemmas.

Lemma

Consider the $m \times n$ matrix

$$B = \begin{bmatrix} C_{a\times b} & D_{a\times(n-b)} \\ E_{(m-a)\times b} & F_{(m-a)\times(n-b)} \end{bmatrix}$$

where F is free of 1's, and the minimal cover of B consists of the first a rows and first b columns. Then matrix D contains a independent 1's while matrix E contains b independent 1's.

Proof of first lemma

Note that all elements of C are covered by both a vertical and a horizontal line. Therefore the only time that a horizontal line is required is when there is a 1 in the corresponding row of D.

Similarly, the only time a vertical line is required is when there is a 1 in the corresponding column of E. Hence there are 1's in each of the a rows of D, and 1's in each of the b columns of E.

Our task is to show that there are in fact a independent 1's in D, and b independent 1's in E.

Proof of first lemma ...

Consider D (the argument relating to E is analogous). Suppose the number of independent 1's was in fact less than a. Then in the matrix form of Hall's theorem, there must be a set of k rows ($k \leq a$) in which all the 1's lie in $t < k \leq a$ columns.

If this were the case we could cover the 1's by t vertical lines, rather than the k horizontal lines (remember all 1's in C are already covered by a vertical line).

This would mean we would have covered all 1's by less than $a + b$ lines, which contradicts $a + b$ being minimal.

□

Lemma

If the minimum number of lines covering all the 1's in matrix A is h, then there are at least h independent 1's in A.

Proof

Suppose the lines in a minimal cover consist of a rows and b columns ($a + b = h$). By moving these rows and columns, these can be taken to be the first a rows and first b columns.

All the 1's lie in these lines, and so the matrix must be of the form

$$\begin{bmatrix} C_{a\times b} & D \\ E & F \end{bmatrix}_{m\times n}$$

where F has no 1's.

Proof of second lemma...

Application of the previous lemma tells us that D contains a independent 1's, while E contains b independent 1's.

Because D and E are disjoint, this means there are $a + b = h$ independent 1's, as required.

□

Proof of König-Egerváry max-min theorem

Because independent 1's must lie in distinct rows and columns, if the maximum number of independent 1's is k we must have at least k lines. Therefore

$$min\ \#\ lines\ covering\ 1's\ \geq max\ \#\ of\ independent\ 1's.$$

By the second lemma we also have

$$max\ \#\ of\ independent\ 1's \geq min\ \#\ lines\ covering\ 1's$$

The two inequalities together establish the result. $\square$

§5.2 – Vertex covers and maximum size matchings

The König-Egerváry max-min theorem is usually stated in terms of bipartite graphs:

Definition

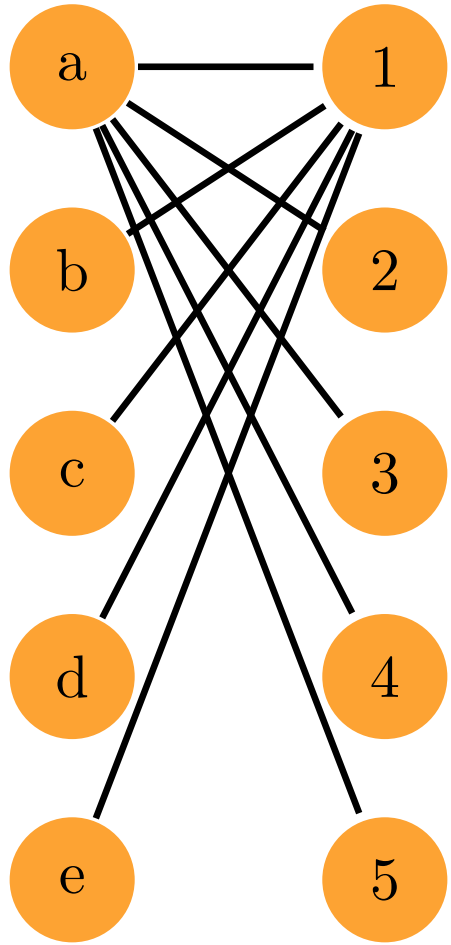
A vertex cover of a graph is a set of vertices such that every edge of the graph is incident to at least one of the vertices in the cover.

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.

Theorem (König's theorem for bipartite graphs)

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

Example



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

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 size matching in a bipartite graph 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 $M = (y_1x_2)(y_2x_3)\cdots(y_nx_{n+1})$. By assumption it is part of an augmenting path, say $x_1(y_1x_2)(y_2x_3)\cdots(y_nx_{n+1})y_{n+1}$. This in turn gives the matching $(x_1y_1)(x_2y_2)\cdots(x_{n+1}y_{n+1})$ which is one bigger than M, so 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 do this, suppose there is a 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 paths. The paths have 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.

□

§5.3 – Weighted Assignment

The following is a refinement of the job assignment problem.

Suppose there are 4 jobs to be done by 4 different people. Let a rating system be devised, say from the numbers 0 up to 9, with 0 being the most suited and 9 being the least suited. For each person a 4-tuple can be formed out of their rating for the 4 jobs in order.

From the 4-tuples a matrix can be formed. 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. This problem is also called the minimum weight perfect matching problem.

Algorithmic solution

Step 1
Subtract the smallest 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 max-min 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. Subtract l from the uncrossed columns, and add l to the crossed rows. Equivalently

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.

Step 3 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 a bipartite graphs, and the second as seeking matchings of maximum size. For the latter problem we can use Berge's lemma.

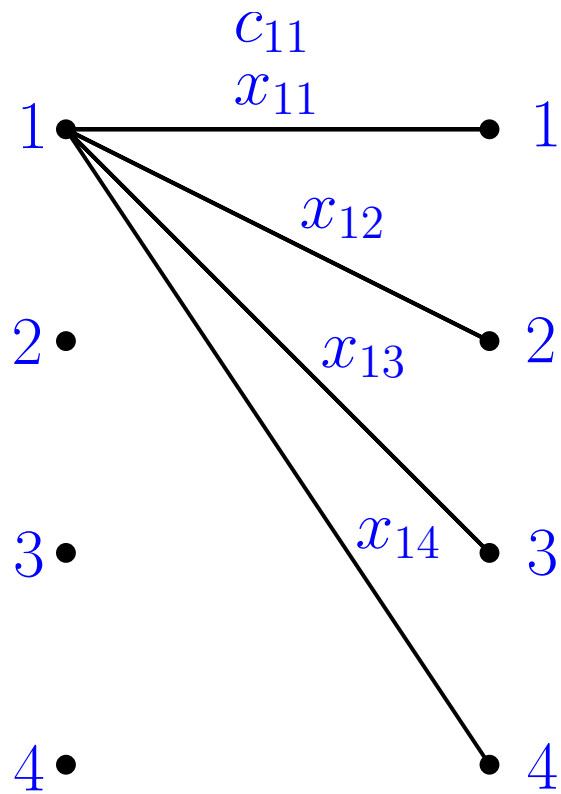
For large matrices a systematic approach for finding a minimum cover by lines first determines the maximum size matching (i.e., maximum number of independent 0's) using eg. Berge's lemma, and then deduces the minimum line cover.

Finding the minimum line cover once the independent 0's have been identified

1. Mark with a cross all other (non-independent) 0's,
2. Mark with a tick rows with no independent 0's,
3. Mark with a tick columns having crosses in ticked rows,
4. Mark with a tick rows having independent 0's in marked columns. Go to Step 2 and repeat until there is no change. Then
5. Draw lines through unmarked rows and marked columns. (Check that # lines = # of independent 0's.)

§5.4 – LP formulation of the minimum weight assignment 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):

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

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

Note that the integer constraints are not needed.