2456-8686, x(ii), 2026: 102 - 130
https://doi.org/10.26524/cm241

ORIGINAL RESEARCH Open Access

A STUDY ON VERTEX COLORING AND EDGE COLORING AND ITS APPLICATION

Nanthitha M M [1], Sampoornam M[2] and Kalaiarasi S [3]

Received: 01 June 2026/ Accepted: 01 July 2026 / Published online: 24 July 2026

Abstract

Graph coloring is an important area of graph theory that deals with assigning colors to the vertices and edges of a graph under certain conditions. Vertex coloring ensures that adjacent vertices have different colors, while edge coloring ensures that adjacent edges receive different colors. This study presents the basic concepts of vertex coloring and edge coloring and discusses their practical applications in scheduling, timetabling, frequency assignment, map coloring, and resource allocation. These coloring techniques help in minimizing conflicts and optimizing the efficient use of resources in various real-world problems.

Key words: Graph, Loop, Walk, Trail, Cycle,Simple Graph, Directed graph, Degree, In - Degree, Out - Degree, Eulerian, Tree, Vertex Coloring, Edge Coloring, Chromatic Number

AMS classification:

1 Introduction

The recent developments in Graph Theory have their impacts on computer science and communication networks, which too have their own contributions to the development of this subject. Modern graph theory is both structural and algorithmic. It plays a significant role in solving optimization problems and higher arithmetic. Graph colouring is a natural extension of graph theory. The colouring problems studied here are primarily vertex colouring and edge colouring in nature, satisfying simple algebraic constraints of adjacency.

In this project work, we proceed from the most general results about vertex colouring to stronger specific results about edge colouring. Here, we are concerned with the

[1]Department of Mathematics, Sacred Heart College(Autonomous) Tirupathur,Tamil Nadu, India. Email:nandhugms2003@gmail.com

[2]Department of Mathematics, Sacred Heart College(Autonomous) Tirupathur,Tamil Nadu, India. Email:sampooranamm@gmail.com

[3]Department of Mathematics, Sacred Heart College(Autonomous) Tirupathur,Tamil Nadu, India. Email:skalaiarasi2@gmail.com

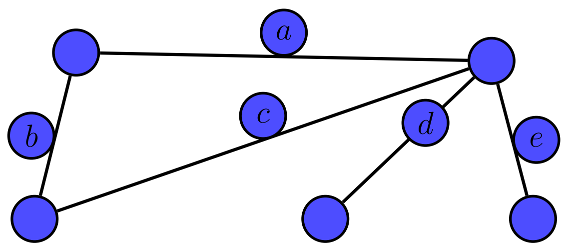
chromatic number, chromatic index, and the applications of colouring rather than concerning the algebraic structure of the groups or rings arising in the theory. Also, graph colourings are the roots of certain partitioning problems, and so we begin the chapters with basic definitions and theorems of graph theory.

Basic Definitions

Graph

A Graph $G = (V, E)$ consists of a set V of vertices(also called nodes) and a set E of edges.

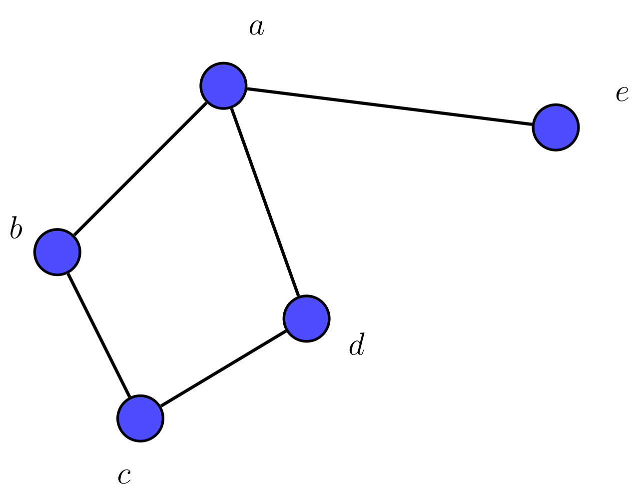
Example:



End Points

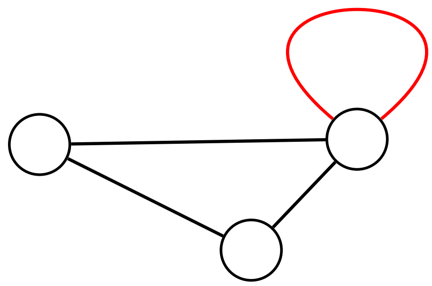
If an edge connects to a vertex. we say the edge is incident to the vertex and say the vertex is an endpoint of the edge.

Example:



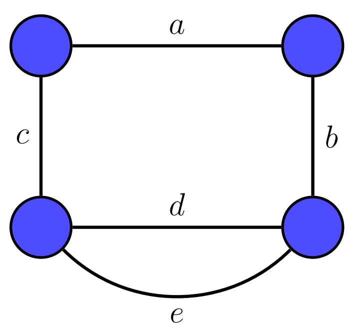
Loop edge

If an edge has only one end point. then it is called a loop edge.
Example: An edge e1 starts at vertex v_1 and ends at the same vertex v_1. Since it has only one endpoint, it is called a loop.



Multiple edge

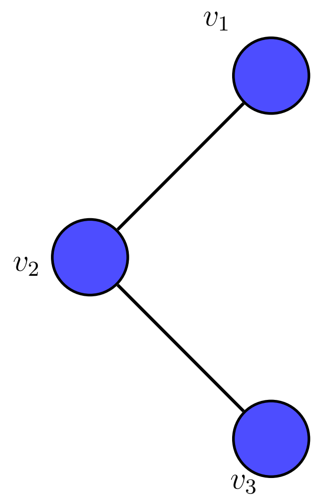
If two or more edges have the same endpoints then they are called multiple or parallel edges.
Example: If there are two distinct edges e_1 and e_2 connecting the same pair of vertices v_1 and v_2, these are called parallel edges.



Adjacent vertices

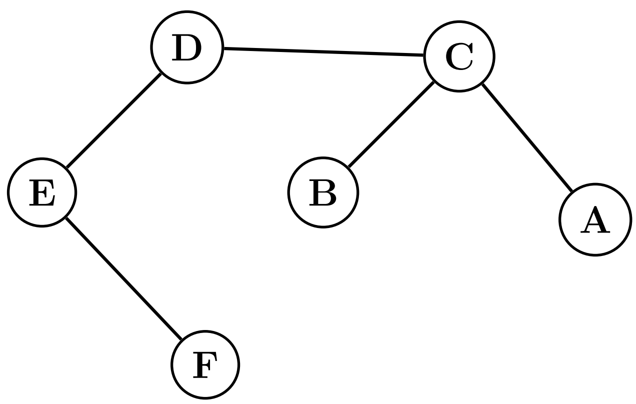
Two vertices that are joined by an edge are called adjacent vertices.
Example: In a triangle graph, any two vertices connected by a direct edge are adjacent. For example, v_1 and v_2 are adjacent if there is an edge (v_1, v_2) between them.

2456-8686, x(ii), 2026: 102 - 130
https://doi.org/10.26524/cm241



Pendent vertex

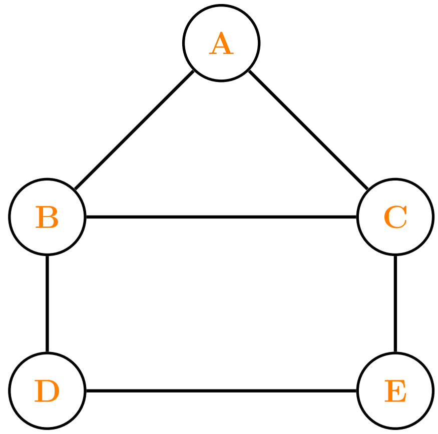
A pendant vertex is a vertex that is connected to exactly one other vertex by a single edge.
Example: A vertex v_2 is called a pendant vertex if it is connected to exactly one other vertex v_1 by a single edge. The degree of a pendant vertex is always 1.



Walk

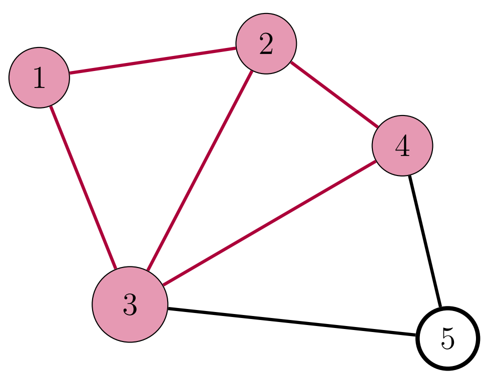
A walk in a graph is a sequence of alternating vertices and edges v_1 e_1 v_2 e_2 . . . v_n e_n $v_n + 1$ with n ≥ 0. If $v_1 = v_n + 1$ then the walk is closed. The length of the walk is the number of edges in the walk. A walk of length zero is a trivial walk.
Example: Consider a sequence $v_1 \rightarrow e_1 \rightarrow v_2 \rightarrow e_2$ $v_3 \rightarrow e_3$ $v_1 \rightarrow e_1 \rightarrow v_2$. In this walk, the vertex v_2 and edge e_1 are repeated, and the length of the walk is 4.

2456-8686, x(ii), 2026: 102 - 130
https://doi.org/10.26524/cm241



Trail

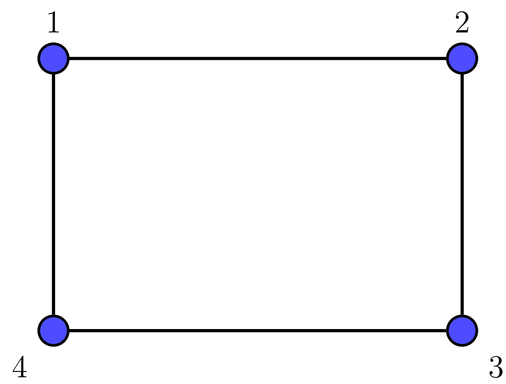
A trail is a walk with no repeated edges. A path is a walk with no repeated vertices. A circuit is a closed trail and a trivial circuit has a single vertex and no edges. A trail or circuit is Eulerian if it uses every edge in the graph.
Example: Consider the sequence $v_1 \to 1 \to v_2 \to e_2 \to v_3 \to e_3 \to v_1$. In this trail, all edges are distinct (no edge is repeated), but the vertex v_1 is repeated.



Cycle

A cycle is a nontrivial circuit in which the only repeated vertex is the first/last one.
Example: A cycle is a path that starts and ends at the same vertex, such as $v_1 \to v_2 \to v_3 \to v_1$ no other vertices are repeated.

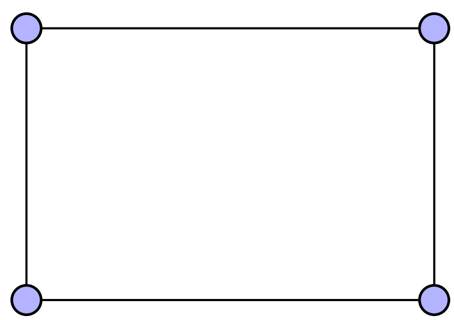
2456-8686, x(ii), 2026: 102 - 130
https://doi.org/10.26524/cm241



Simple graph

A simple graph is a graph with no loop edges or multiple edges. Edges in a simple graph may be specified by a set vi, vj of the two vertices that the edge makes adjacent. A graph with more than one edge between a pair of vertices is called a multigraph while a graph with loop edges is called a pseudograph.

Example: A square graph with vertices v_1, v_2, v_3, v_4 and edges $(v_1, v_2), (v_2, v_3), (v_3, v_4), (v_4, v_1)$ is a simple graph because it has no loops or parallel edges.

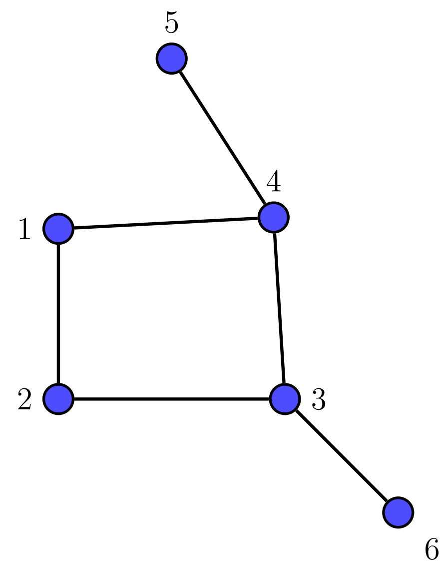


Directed graph

A directed graph is a graph in which the edges may only be traversed in one direction. Edges in a simple directed graph may be specified by an ordered pair (v_i, v_j) of the two vertices that the edge connects. We say that v_i is adjacent to v_i and v_j is adjacent from v_i.

Example: Consider a graph with an edge from v_1 to v_2, denoted as (v_1, v_2). You can only travel from v_1 to v_2, but not backward. Here, v_2 is adjacent from v_1.

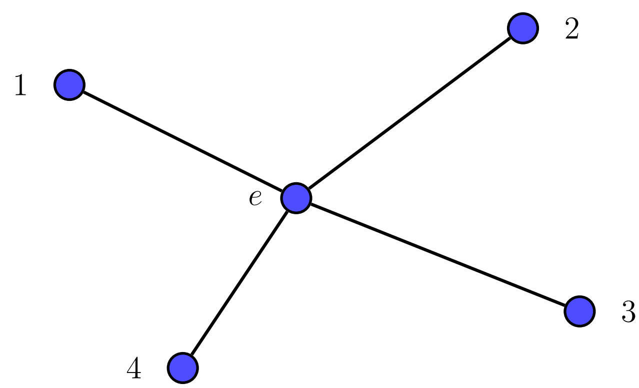
2456-8686, x(ii), 2026: 102 - 130
https://doi.org/10.26524/cm241



Degree

The degree of a vertex is the number of edges incident to the vertex and is denoted degree (v).

Example: In a square graph, each vertex is connected to two edges. Therefore, the degree of each vertex is degree.

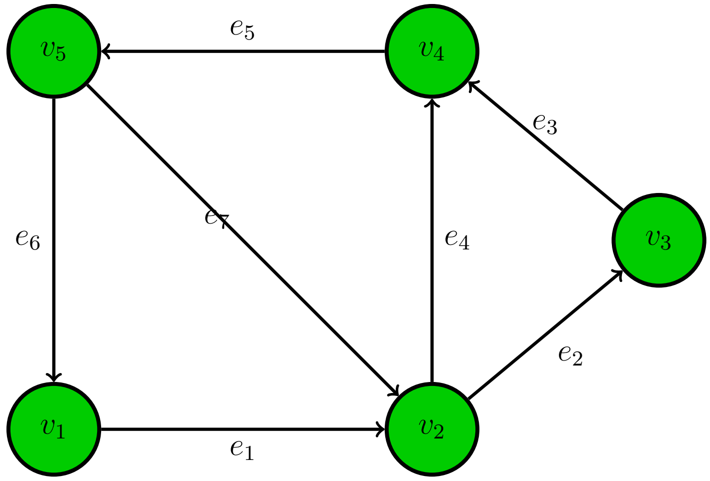


In-Degree and Out-Degree

In a directed graph, the in-degree of a vertex is the number of edges incident to the vertex and the out-degree of a vertex is the number of edges incident from the vertex.

Example: In a directed graph, if vertex v_1 has 2 arrows pointing towards it and 1 arrow pointing away, its in-degree is 2 and its out-degree is 1.

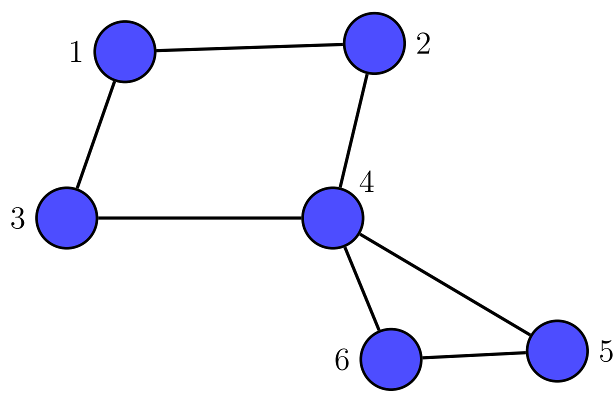
2456-8686, x(ii), 2026: 102 - 130
https://doi.org/10.26524/cm241



Connected

A graph is connected if there is a walk between every pair of distinct vertices in the graph.

Example: A triangle graph is a connected graph because you can reach any other vertex. If a h vertex from any graph has two separate pieces, it is disconnected.



Sub graph

A graph H is a subgraph of a graph G if all vertices and edges in H are also in G.

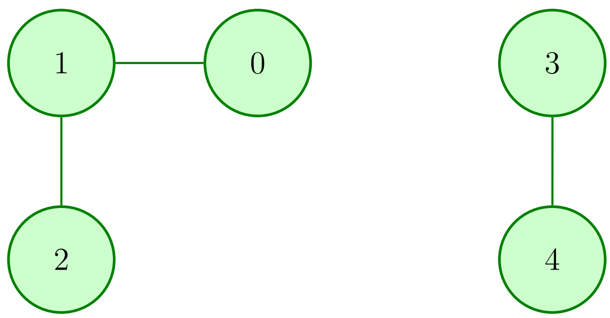
Example: Let Graph G be a triangle with vertices (v_1, v_2, v_3) and edges connecting them. If we take only two vertices (v_1, v_2) and the single edge between them, this smaller part is called a subgraph H.

2456-8686, x(ii), 2026: 102 - 130
https://doi.org/10.26524/cm241

Connected Component

A connected component of G is a connected sub graph H of G such that no other connected subgraph of G contains H.

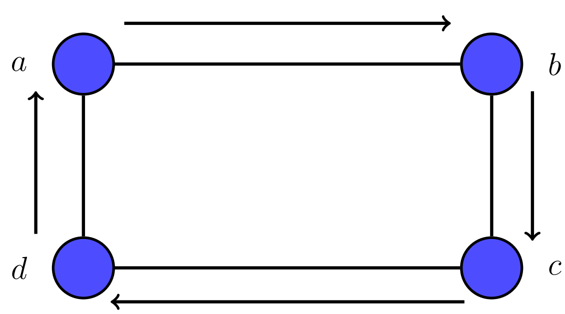
Example: Imagine a graph that looks like two separate triangles that are not joined together. Each triangle is a connected component of the whole graph because each piece is connected within itself but not to the other piece.



Eulerian

A graph is called Eulerian if it contains an, Eulerian circuit.

Example: A graph is Eulerian if you can start at one vertex, travel along every single edge exactly once, and return to the starting vertex.

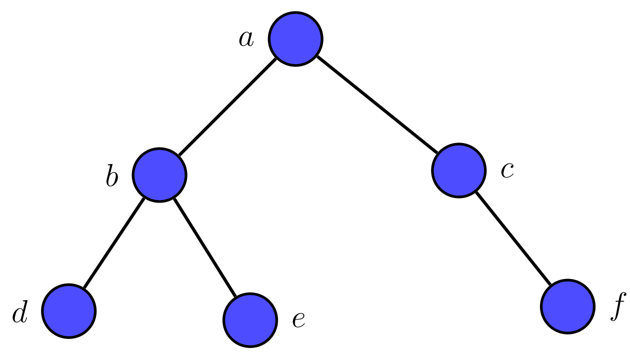


2456-8686, x(ii), 2026: 102 - 130
https://doi.org/10.26524/cm241

Tree

A tree is a connected, simple graph that has no cycles. Vertices of degree 1in a tree are called the leaves of the tree.

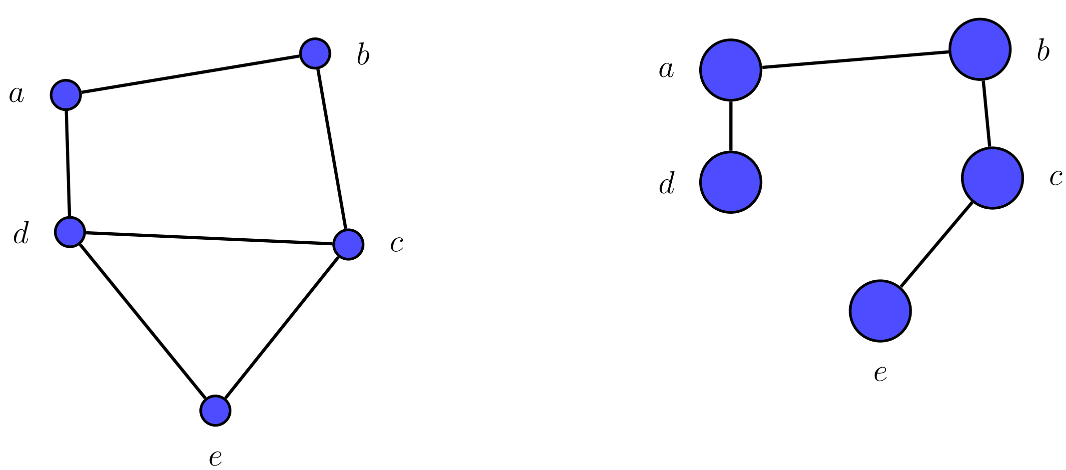
Example: Consider a graph with vertices (v_1, v_2, v_3, v_4) connected as $v_1 - v_2$, $v_2 - v_3$, and $v_2 - v_4$. This is a tree because it is connected and has no cycles. The vertices v_1, v_3, and v_4 have degree 1, so they are the leaves.



Spanning tree

Let G be a simple, connected graph. The subgraph T is a spanning tree of G if T is a tree and every node in G is a node in T.

Example: Let G be a square graph with four vertices and four edges. If we remove one edge, the remaining three edges form a spanning tree T because it connects all four vertices of G without forming a cycle.

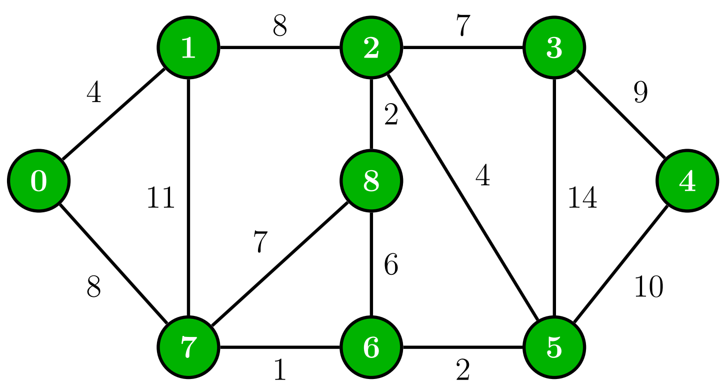


2456-8686, x(ii), 2026: 102 - 130
https://doi.org/10.26524/cm241

Weighted graph

A weighted graph is a graph $G = (V, E)$ along with a function W: E $\rightarrow R$ that associates a numerical weight to each edge. If G is a weighted graph, then T is a minimal spanning tree of G if it is a spanning tree and no other spanning tree of G has smaller total weight.

Example: In a weighted graph, edges have numbers (weights) like 5, 10, or 2. A Minimal Spanning Tree is a spanning tree where the sum of these weights is the lowest possible. For example, if a graph has edges with weights 2, 3, and 10, the MST will use the edges with weights 2 and 3.



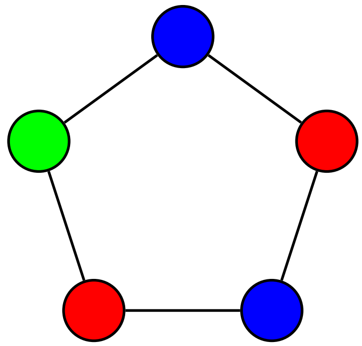
Vertex Colouring and Edge Colouring

Vertex Colouring: Vertex colouring is an assignment of labels or colours to the vertices of a graph G such that no two adjacent vertices share the same colour. A graph that can be properly coloured using k colours is called a k-colourable graph.

Chromatic Number: The minimum number of colours required to properly colour a graph G is called its chromatic number, denoted by $\chi(G)$.

Example:

Consider a Cycle Graph (C_5) with 5 vertices. To colour this graph properly We assign colours. such that, adjacent nodes are different. For an odd cycle like C_5, we need at least 3 colours. Therefore, the chromatic number $\chi(C_5) = 3$.



Theorem 2.1. *If G is k - critical, then $\delta \geq k-1$.*

Proof. By contradiction, If possible, let G be a k - critical graph with $\delta < k-1$, and let v be a vertex of degree δ in G. Since G is k-critical, $G–v$ is (k -l)-colourable. Let $(V_1 V_2, ..., v_{k-1})$-be a (k - l)-colouring of $G - V$. By definition, v is adjacent in G to $\delta < k-1$ vertices, and therefore v must be nonadjacent in G to every vertex of some Vj. t then $(V_1, V_2, \ldots, V_j) \cup \{\{v\}\}, \ldots, V_{k-1}$ is a $(k-1)$-colouring of G., a contradiction. Thus, $\delta \geq k-1$.

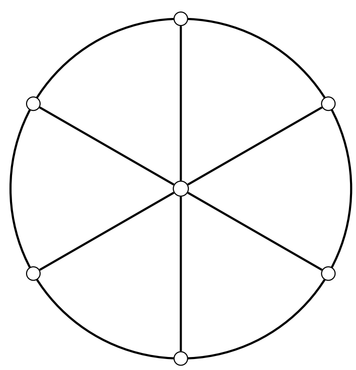


Figure 1.1. A 3-chromatic graph

□

Corollary 2.2. *Every k-chromatic graph has at least k vertices of degree at least $k-1$.*

Proof. Let, G be a k - chromatic graph, and let H be a k-critical subgraphs of G. each vertex of H has degree at least k - 1 in H, and hence also in G. The corollary now follows since H; being K- chromatic, clearly has at least k vertices.

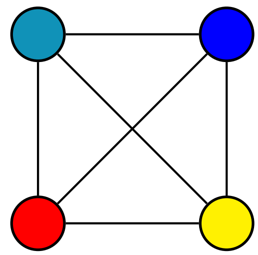


Figure 1.2 K_4 - chromatic graph.

□

Corollary 2.3. *For any graph G,* X $\Delta \leq$ *+ 1.*

Proof. Let, S be a vertex cut of a connected graph G, and let the components of $G - S$ have vertex sets $V_1, V_2, ..., V_n$. Then, the subgraphs G_i = $G[VUS]$ are called the S-components of G. We say that, colourings of $G_1, G_2, ..., G_n$. agree on S if, for every $v \in s$, vertex v is assigned the same colour in each of the colourings.

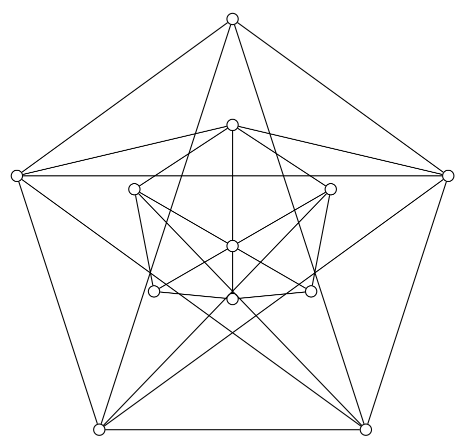


Figure 1.2 K_4 - chromatic graph.

□

Theorem 2.4. *In a critical graph, no vertex cut is a clique.*

2456-8686, x(ii), 2026: 102 - 130
https://doi.org/10.26524/cm241

Proof. By contradiction. Let G be a k-critical graph, and suppose that, G has a vertex cut S that is a clique. Denote the S-components of G be $G_1, G_2, ..., G_n$. Furthermore, because S is a clique, the vertices in S must receive distinct colours in any $(k-1)$ - colouring of Gi It follows that there are $(k-1)$ - colourings of $G_1, G2, ..., G_n$ which agree on S. But these colourings together yield a $(k-1)$ - colouring of G, a contradiction.

□

Corollary 2.5. *Every critical graph is a block.*

Proof. If v is a cut vertex, then v is a vertex cut which is also, trivially, a clique. It follows from theorem that no critical graph has a cut vertex; equivalently, every critical graph is a block. Another consequence of theorem is that if a k - critical graph G has a 2-vertex cut. u, v, then u and v cannot be adjacent.
We shall say that, A (u, v) - component G_i of G is of type 1 if every (k - 1) - colouring of G_i assigns the same colour to u and v, and of type 2 if every $(k - I)$ -colouring of Giassigns different colours to u and v. □

Theorem 2.6. *Let G be a k - critical graph with a 2 - vertex cut u, v. Then (i) G = $G_1 \cup G_2$, where G_i is a u, v - component of type i $(i = 1, 2)$ and*

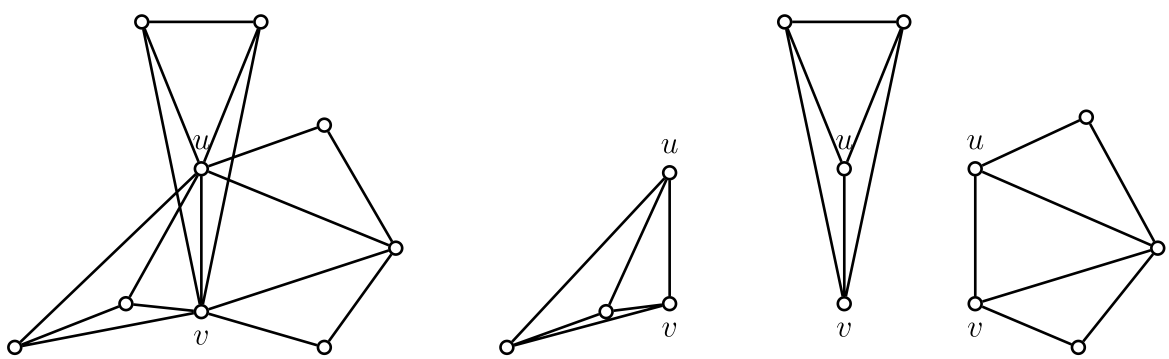


Figure 1.4 *(a) G; (b) the u, v - components of G.*

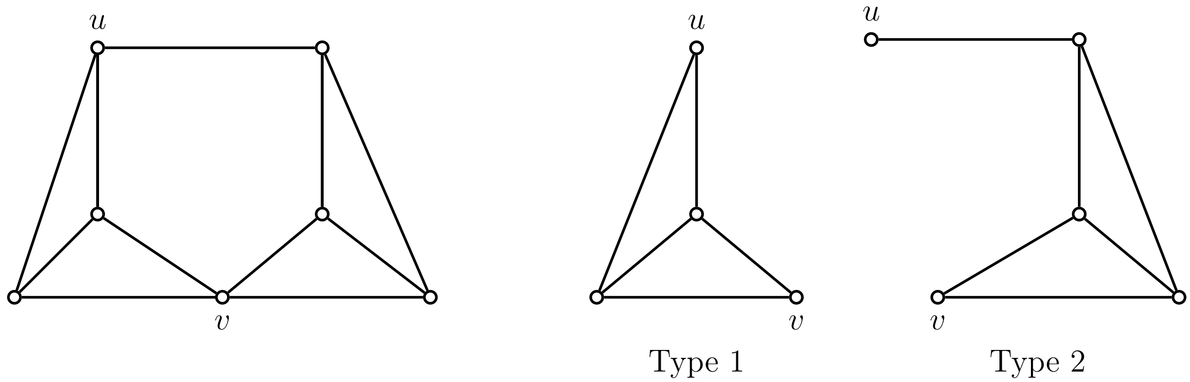


Figure 1.5

(ii) Set H_1 = G_1 + uv. Since G_1 is of type 1, H_1 is k - chromatic. We shall prove that H_1 is critical by showing that, for every edge e of H_1, H_1 - e is 29 $(k-l)$ - colourable. This is clearly so if $e = uv$, since then H_1 - e = G_1. Let e (iii) both G_1 + uv and $G_2 \bullet uv$ are k - critical (where $G_2 \bullet uv$ denotes the graph obtained from G_2 by identifying u and v).,

Proof. (i) Since, G is critical, each u, v - component of G is $(k-1)$ - colourable. Now there cannot exist $(k-1)$ - colourings of these u, v - components all of which agree on u, v,since such colourings would together yield a $(k-I)$ - colouring of G. Therefore, there are two u, v - components G_1 and G_2 such that no $(k-I)$ - colouring of G_1 agrees with any $(k-1)$ - colouring of G_2 Clearly one, say G_1 must be of type 1 and the other, G_2, of type 2. Since, G_1 and G_2 are of different types, the subgraph G_1 and G_2 are different is not $(k-1)$ - colourable. Therefore, because G is critical, we must have,

$$G = G_1 \cup G_2.$$

(ii) Set H_1 = G_1 + uv.
Since, Gt is of type 1, H_1 is k - chromatic.
We shall prove that,
H_1 is critical by showing that, for every edge e of H_1, H_1 - e is (kl) - colourable.

This is clearly so if,

$$e = uv,$$

Then,

$$H_1 \text{ - } e = G_1$$

Let, e be some other edge of H_1. In any $(k - I)$ - colouring of $G - e$, the vertices u and v must receive different colours, since G_2 is a subgraph of $G-e$. The restriction of such a colouring to the vertices of Gt is a $(k - l)$ - colouring of H_1 - e. Thus G_1 + uv is k - critical. An analogous argument G_2 uv is k-critical.

□

Corollary 2.7. *Let G be a k - critical graph with a 2 - vertex cut* u, v. *Then* $d(u)+d(v)$ $\geq 3k - 5$.

Proof. Let G_1 be the u, v - component of type 1 and G_2 the u, v - component of type 2. Set H_1 = G_1 + uv and H_2 = G_2 • uv.

$$d_{H1}(u) + d_{H1}(v) \geq 2k - 2$$

and

$$d_{H2}(w) \geq k - l$$

where w is the new vertex obtained by identifying u and v. It follows that,

$$d_{G1}(u) + d_{G1}(v) \geq 2k - 4$$

and,

$$d_{G2}(u) + d_{G2}(v) \geq k - 1$$

2456-8686, x(ii), 2026: 102 - 130
https://doi.org/10.26524/cm241

□

Edge Colouring:

Theorem 2.8. *If G is bipartite, then X' = Δ.*

Proof. Let, G be a graph with $X' > \Delta$, let C = $(E_1, E_2, ..., E_\Delta)$ be an optimal Δ-edge colouring of G, and let u be a vertex such that $c(u) < d(u)$.
Clearly, u satisfies the hypothesis. Therefore,
G contains an odd cycle and so is not bipartite. It follows from that if G is bipartite, Then,

$$X' = \Delta.$$

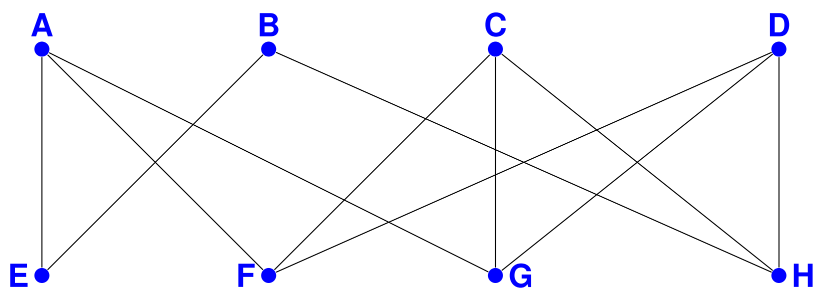
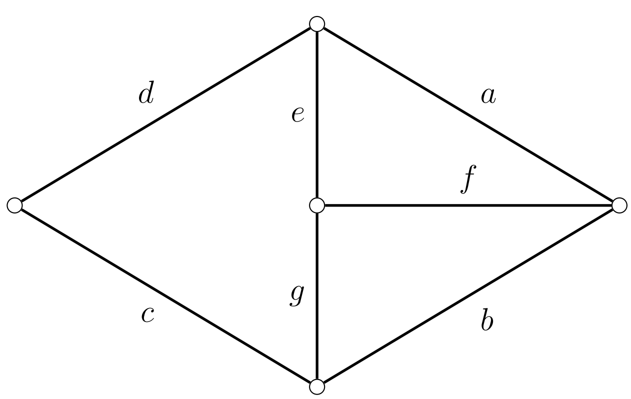


Figure: 1.6 Bipartite graph

□

Lemma 2.9. *Let G be a connected graph that is not an odd· cycle. Then*



2456-8686, x(ii), 2026: 102 - 130
https://doi.org/10.26524/cm241

Figure: 1.7

G has a 2-edge colouring in which both colours are represented at each vertex of degree at least two.

Proof. We may clearly assume that,

G is nontrivial.

Suppose, first that G is Eulerian. If G is an even cycle, the proper 2 - edge colouring of G has the required property. Otherwise, G has a vertex Vo of degree at least four. Let, $v_o e_1 v_1, ...e_\epsilon$ be an Euler tour of G, and set

E1 = ei — i odd and E2 = ei — i even

Then the 2-edge colouring (E_1, E_2) of G has the required property, since each vertex of G is an internal vertex of $v_o e_1 v_1, ...e_\epsilon v_o$. If G is not Eulerian, construct a new graph G^* by adding a new vertex v_o and joining it to each vertex of odd degree in G. Clearly G^* is Eulerian. Let $v_o e_1 v_1, ...e_\epsilon^*$be an Euler tour of G^* and define E_1 and E_2 as in. It is then easily verified that the 2 - edge colouring ($E_1 \cap E$, $E_2 \cap E$) of G has the required property. Given a k - edge colouring $\mathcal{C}$ of G we shall denote by $c(v)$ the number of distinct colours represented at v. Clearly, we always have,

c(v) $<$ d(v)

Moreover, $\mathcal{C}$ is a proper k-edge colouring. If and only if equality holds in for all vertices v of G. We shall call a k - edge colouring $\mathcal{C}'$ an improvement on $\mathcal{C}$ if

$$\sum_{v\in V} c'(v) > \sum_{v\in V} c(v)$$

where $c^{'}(v)$ is the number of distinct colours represented at v in the colouring $\mathcal{C}'$. An optimal k - edge colour. □

Lemma 2.10. *Let $\mathcal{C}$ = (E_1, E_2,...,E_k) bean optimal k - edge colouring of G. If there is a vertex u in G and colours i and j such that i is not represented at u and j is* *represented at least twice at u, then the component of G [$E_i \cup E_j$] that contains u is an odd cycle.*

Proof. Let, u be a vertex that satisfies the hypothesis of the lemma, and denote by H the component of G [$E_i \cup E_j$] containing u.
Suppose that H is not an odd cycle.
Then,
H has a 2-edge colouring in which both colours are represented attach vertex of degree at least two in H. When we recolour the edges of H with colours i and i in this way, we obtain a new k-edge colouring $\mathcal{C}$ = ($E_1^{'}$, $E_2^{'}$,...,$E_k^{'}$) of G. Denoting by $c^{'}(v)$ the number of distinct colours at v in the colouring $\mathcal{C}^{'}$.we have,

$$c^{'}(u) = c(u) + 1$$

since, now both i and j are represented at u, and also

$$c'(u) = c(u) + 1$$

Thus,

$$\sum_{v \in V} c'(v) > \sum_{v \in V} c(v)$$

contradicting the choice of $\mathcal{C}$. It follows that H is indeed an odd cycle. □

Applications

Applications for Vertex Colouring

The followings are some applications of vertex colourings.
A. Example Suppose that, Mg Aung, Mg Ba, Mg Chit, Mg Hla, Mg Kyaw, Mg Lin and Mg Tin are planning a ball. Mg Aung, Mg Chit, Mg Hla constitute the publicity committee. Mg Chit, Mg Hla and Mg Lin are the refreshment committee. Mg Lin and Mg Aung make up the facilities committee. Mg Ba, Mg Chit, Mg Hla, and Mg Kyaw form the decorations committee. Mg Kyaw, Mg Aung, and Mg Tin are the music committee.
Mg Kyaw, Mg Lin, Mg Chit form the cleanup committee. We can find the number

2456-8686, x(ii), 2026: 102 - 130
https://doi.org/10.26524/cm241

of meeting times, in order for each committee to meet once. We will construct a graph model with one vertex for each committee. Each vertex is labelled with the first letter of the name of the committee that is represents. Two vertices are adjacent whenever the committees have at least one member in common.
For example, P is adjacent to R,
Since,
Mg Chit is on both the publicity committee and the refreshment committee.

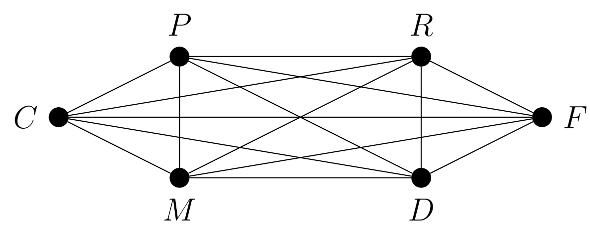


Fig. 1.8 (a) Graph Model for the Ball Committee

To solve this problem, we have to find the chromatic number of G. Each of the four vertices P, D, M, and C is adjacent to each of the other. Thus, in any colouring of G these four vertices must receive different colours say 1, 2, 3 and 4, respectively, as shown in Figure. Now, R is adjacent to all of these except M, so it cannot receive the colours 1, 2, or 4.
Similarly, vertex F cannot receive colours 1, 3, and 4, nor can it receive the colour that R receives. But we could colour R with colour 3 and F with colour 2.

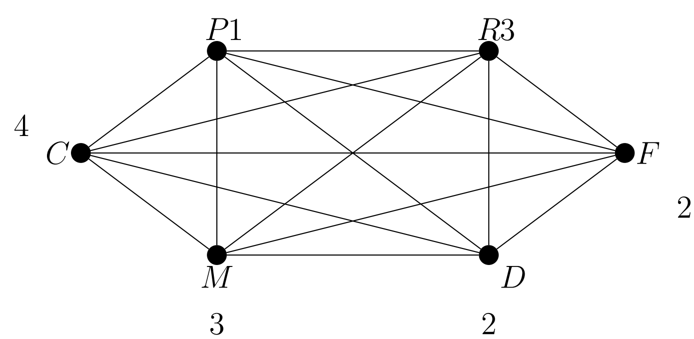


Fig. 1.8 (b)

2456-8686, x(ii), 2026: 102 - 130
https://doi.org/10.26524/cm241

Using a Colouring to Ball Committee Problem This completes a 4 - coloring of G. Since,the chromatic number of this graph is 4, four meeting times are required. The publicity committee meets in the first time slot, the facilities committee and the decoration committee meets in the second time slot, refreshment and music committee meet in the third time slot, and the cleanup committee meets last.

B. Example

We can find a schedule of the final exams for,

Math 115, Math 116, Math 185, Math 195, CS 101, CS 102, CS 273, and CS 473, using the fewest number of different time slots, if there are no students taking both Math 115 and CS 473, both Math 116 and CS 473, both Math 195 and CS 101, both Math 195 and CS 102, both Math 115 and Math 116, both Math 115 and Math 185, and Math 185 and Math 195, but there are students in every other combination of courses.

This scheduling problem can be solved using a graph model, with vertices representing courses and with an edge between two vertices if there is a common student in the courses they represent. Each time slot for a final exam is represented by a different colour.

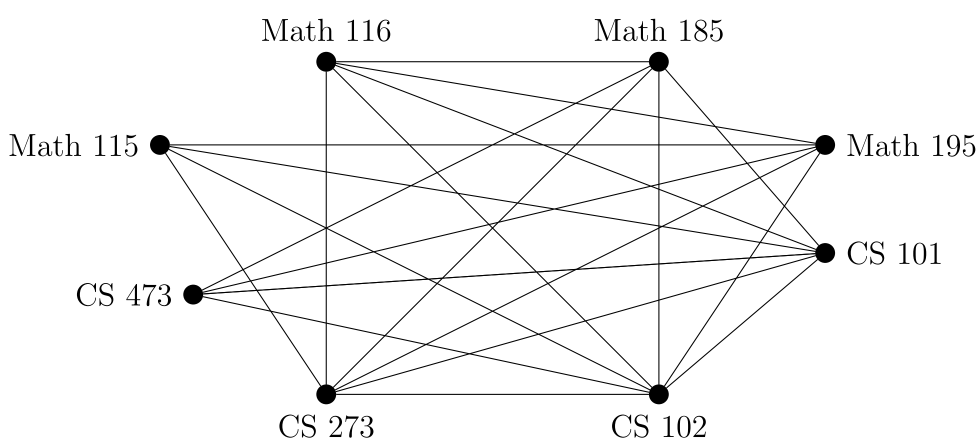


Fig. 1.9 (a)The Graph Representing the Scheduling of Final Exams

2456-8686, x(ii), 2026: 102 - 130
https://doi.org/10.26524/cm241

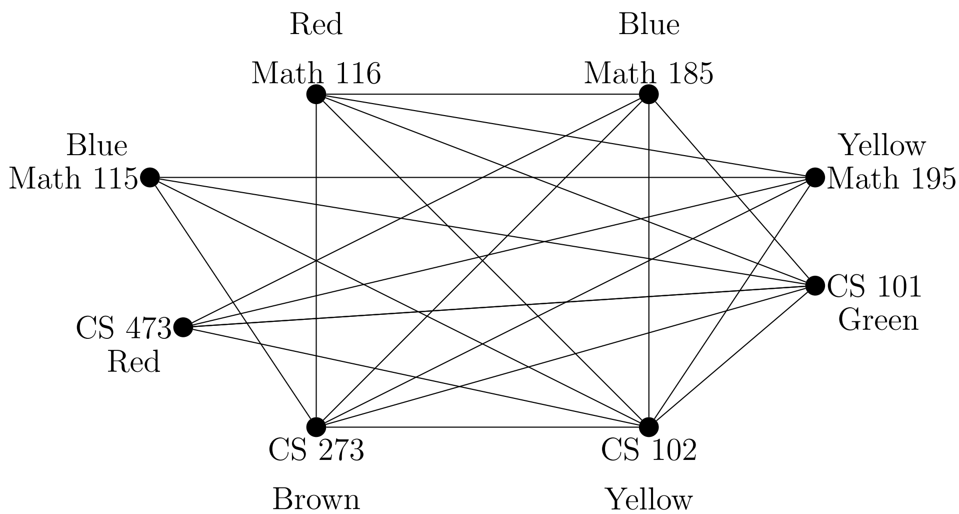


Fig. 1.9 (b)Using a Colouring to Schedule Final Exams

A scheduling of the exams corresponds to a colouring of the associated graph

Time Period	**Courses**
I	Math 116, CS 473
II	Math 115, Math 185
III	Math 195, CS 102
IV	CS 101
V	CS 273

Since, Math 116, Math 185, CS 101, CS 102, CS 2734, from acomplete sub graph of G,

$$\chi(G) \geq 5.$$

2456-8686, x(ii), 2026: 102 - 130
https://doi.org/10.26524/cm241

Thus, five time slots are needed. A colouring of the graph using five colours and the associated schedule are shown in Fig. 1.9 (b).

APPLICTIONS OF EDGE COLORING

The Timetabling Problem:

In a school there are m teachers $X_1, X_2, ..., X_m$ and n classes $Y_1, Y_2, ..., Y_n$. Given, the teacher X_i is required to teach class Yj for pij periods, schedule a complete timetable in the minimum possible number of periods.

This is known as the Timetabling Problem. We represent the teaching requirements by a bipartite graph G with bipartition (X, Y),where $X= x_1, x_2, ..., x_n$ and $Y = y_1, y_2, ..., y_n$ and vertices x_i and y_j are joined by p_{ij} edges. Now in any one period, each teacher can teach at most one class, and each class can be taught by at most one teacher. This is at least our assumption.

Thus, a teaching schedule for one period corresponds to a matching in the graph and, conversely, each matching corresponds to a possible assignment of teachers to classes for one period.

Our problem, therefore is to partition the edges of G into as few matching as possible or, equivalently, to properly colour the edges of G with as few colours as possible.

Since G is bipartite, we know, $\chi^{'}(G) = \Delta(G)$. Hence, if no teacher teaches for more than p periods, the teaching requirements can be scheduled in a p-period timetable. We thus have a complete solution to the problem.

Graph colouring techniques in scheduling:

Here some scheduling problems that uses variants of graph colouring methodologies. such as, precolouring, list Colouring, multi colouring, minimum sum colouring are given in brief.

Job scheduling:

Here the jobs are assumed as the vertices of the graph and there is an edge between two jobs if they cannot be executed simultaneously. There is a 1 - 1 correspondence between the feasible scheduling of the jobs and the colourings of the graph.

2456-8686, x(ii), 2026: 102 - 130
https://doi.org/10.26524/cm241

Aircraft scheduling:

Assuming that there are k aircrafts and they have to be assigned n flights. The ith flight should be during the time interval (ai, bi). If two flights overlap, then the same aircraft cannot be assigned to both the flights. This problem is modelled as a graph as follows. The vertices of the graph correspond to the flights. Two vertices will be connected, if the corresponding time intervals overlap. Therefore, the graph is an interval graph that can be coloured optimally in polynomial time.

Bi-processor tasks:

Assume that, there is a set of processors and set of tasks. Each task has to be executed on two processors simultaneously and these two processors must be pre assigned to the task. A processor cannot work on two jobs simultaneously. This type of tasks will arise when scheduling of file transfers between processors or in case of mutual diagnostic besting of processors.
This can be modelled by considering a graph whose vertices correspond to the processes and if there is any task that has to be executed on processors i and j, then and edge to be added between the two vertices.
Now, the scheduling problem is to assign colours to edges in such a way that every colour appears at most once at a vertex. If there are no multiple edges in the graph that is, no two tasks require the same two processors then the edge colouring technique can be adopted. The authors have developed an algorithm for multiple edges which gives an 1-1 approximate solution.

2456-8686, x(ii), 2026: 102 - 130
https://doi.org/10.26524/cm241

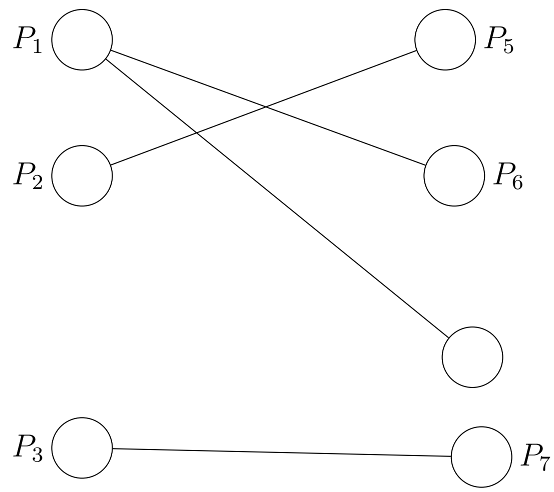


Figure – 1.10 Tasks allocated to processors.

The diagram shows, the tasks namely task1, task2, task3 and task4 are allocated to the processors (P1, P5); (P1, P6); (P2, P4) and (P3, P7) respectively.

Precolouring extension:

In certain scheduling problems, the assignments of jobs are already decided. In such cases precolouring technique can be adopted. Here, some vertices of the graph will have preassigned colour and the precolouring problem has to be solved by extending the colouring of the vertices for the whole graph using minimum number of colours.

List colouring:

In list colouring problem, each vertex v has a list of available colours and we have to find according where the colour of each vertex is taken from the list of available colours. This list colouring can be used to model situations where a job can be processed only in certain time slots or can be processed only by certain machines.

Minimum sum colouring:

In minimum sum colouring, the sum of the colours assigned to the vertices is minimal in the graph. The minimum sum colouring technique can be applied to the scheduling theory of minimizing the sum of completion times of the jobs. The multicolour version of the problemcan be used to model jobs with arbitrary lengths.

Here, the finish time of a vertex is the largest colour assigned to it and the sum of colouring is the sum of the finish time of the vertices. That is the sum of the finish times in a multi colouring is equal to the sum of completion times in the corresponding schedule.

Time table scheduling:

Allocation of classes and subjects to the professors is one of the major issues if the constraints are complex. Graph theory plays an important role in this problem. For m professors with n subjects the available number of p periods timetable has to be prepared.
This is done as follows,

P	n_1	n_2	n_3	n_4	n_5
M_1	2	0	1	1	0
M_2	0	1	0	1	0
M_3	0	1	1	1	0
M_4	0	0	0	1	1

Figure –1.11

A bipartite graph (or bi graph is a graph whose vertices can be divided into two disjoint sets U and V such that every edge connects a vertex in U to one in V; that is, U and V are independent sets) G where the vertices are the number of professors say $m_1, m_2, m_3, m_4, ..., m_k$ and n number of subjects say $n_1, n_2, n_3, n_4, ..., n_m$. such that, the vertices are connected by pi edges. It is presumed that at any one period each professor can teach at most one subject and that each subject can be taught by maximum one professor.
Consider, the first period. The timetable for this single period corresponds to a matching in the graph and conversely, each matching corresponds to a possible assignment of professors to subjects taught during that period. So, the solution for the timetabling problem will be obtained by partitioning the edges of graph G into minimum number of Matching.

2456-8686, x(ii), 2026: 102 - 130
https://doi.org/10.26524/cm241

Also, the edges have to be coloured with minimum number of colours. This problem can also be solved by vertex colouring algorithm."The line graph L(G) of G has equal number of vertices and edges of G and two vertices in L(G) are connected by an edge if and only if the corresponding edges of G have a vertex in common. The line graph L(G) is a simple graph and a proper vertex colouring of L(G) gives a proper edge colouring of G by the same number of colours. So, the problem can be solved by finding minimum proper vertex colouring of L(G)."
For example,
Consider, there are 4 professors namely $m_1, m_2, m_3, m_4 and m_5$ subjects say n_1, n_2, n_3, n_4, n_5 to be taught. The teaching requirement matrix $p = [p_{ij}]$ is given below. The teaching requirement matrix for four professors and five subjects.

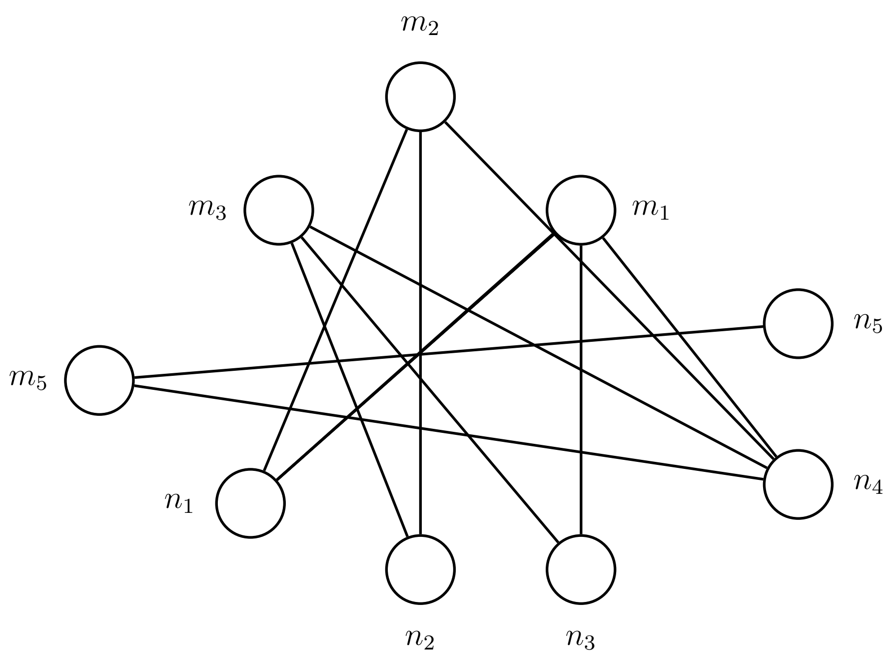


Figure 1.12 Bipartite graph with 4 professors and 5 subjects

Finally, the proper colouring of the above mentioned graph can be done by 4 colours using the vertex colouring algorithm which leads to the edge colouring of the bipartite multigraph G.

Map colouring and GSM mobile phone networks:

Groups Special Mobile (GSM) is a mobile phone network where the geographical area of this network is divided into hexagonal regions or cells. Each cell has a communication tower which connects with mobile phones within the cell. All mobile phones connect to the GSM network by searching for cells in the neighbours.
Since, GSM operate only in four different frequency ranges it is clear by the concept of graph theory that only four colours can be used to colour the cellular regions. These four different colours are used for proper colouring of the regions.
Therefore,
the vertex colouring algorithm may be used to assign at most four different frequencies for any GSM mobile phone network.

Conclusion

In this project, we gave a sufficient review of Graph Colouring and its applications. Vertex and edge colouring are promising tools for describing and studying complex networks and scheduling systems. Graph colouring can be used by both practitioners and theoreticians. Thus, they provide a powerful medium of communication between them: practitioners can learn from theoreticians how to make their models more methodical, and theoreticians can learn from practitioners how to make their models more realistic.
In conclusion, it is clear to see that graph colouring is a simple but effective method of analysing real-world systems. Its use is commonplace because of the simplicity in understanding the models as well as analysing them. They lend themselves to automation and numerous computer programs exist to create and analyse complex colouring patterns with great speed and ease. The use of vertex and edge colouring will surely lead to more efficient and more stable systems being implemented—such as in timetabling and frequency assignment—and therefore increasing the productivity and efficiency of modern computational methods.

2456-8686, x(ii), 2026: 102 - 130
https://doi.org/10.26524/cm241

References

[1] Anindya J.Pal, Samar S.Sarma, Biman Ray, “CCTP, Graph Colouring algorithms – Soft computing Solutions IEEE, 2007.

[2] Bondy, J. A. (1976). Diconnected orientations and a conjecture of Las Vergnas. J. London Math. Soc., to be published.

[3] Daniel Marx, “Graph Colouring problems and their applications in scheduling”.

[4] Gobel, F. and Jagers, A. (1976). On a conjecture of Tutte concerning minimal tree numbers. J. Combinatorial Theory (B), to be published” 36 and 37. Appel, K. and Haken, W.(1976). Every planar map is four colourable. Bull. Arner. Math. Soc., 82, 711-2.

[5] Kumar Bisen, Sanjay. ”Application of Graph Theory in Transportation Networks.” International Journal of Scientific Research and Management 5.7 (2017): 6197-6201.

[6] Narasingh Deo, “Graph theory with applications to engineering and computer science”, Prentice Hall of India, 1990.

[7] Patel, Pranav, and Chirag Patel. ”Various graphs and their applications in real world.” International Journal of Engineering Research and Technology 2.12 (2013): 1499-1504.

[8] Sarma, S. Venu Madhava. ”Applications of graph theory in human life.” International Journal of Computer Application 1.2 (2012): 21-30.

[9] Shariefuddin Pirzada and Ashay Dharwadker, “Journal of the Korean Society for Industrial and applied Mathematics.

[10] Vedavathi, N., and Dharmaiah Gurram. ”Applications on Graph Theory.”