

Application of Graph Colouring

Shashvat Kedia(1610110347) , Raghav Kirpekar(1610110271) , Raman Dutt(1610110357)

December 13, 2017

Abstract

A Graph G is a mathematical structure of consisting of two sets $V(G)$ (vertices of G) and $E(G)$ (edges of G). Proper colouring of a graph is an assignment of colours to the vertices such that the adjacent vertices are coloured differently. This paper discusses colouring on graphs with *Mathematica* and *WebMathematica*.

We draw any graph and also try to show whether it has a Eulerian and Hamiltonian cycles.

1 Introduction

Graph theory would not be what it is today if there had been no colouring problems. A Graph G is a mathematical structure of consisting of two sets $V(G)$ (vertices of G) and $E(G)$ (edges of G).

Vertex colouring is a hard combinatorial optimisation problem we apply several operations on graphs to give different graphs. In addition, we colour vertices to these obtained graphs properly. Many of these graphs are truly beautiful and they provide a wide range of structures to manipulate and study.

A complete graph is a simple graph such that every pair of vertices is joined by an edge. Some number of vertices connected in a close chain is called a cycle. A graph which is obtained by joining a new vertex to every vertex of a cycle is called a Wheel. A connected Acyclic graph is called a tree [?].

Graph Colouring Algorithms : There are many sequential techniques for colouring a graph. One of them is Greedy Graph Colouring. Greedy Colouring concentrates on carefully picking the next vertex to colour and the colour for the next vertex. There are two basic ordering techniques used First Fit and Degree Based Ordering.

First Fit : The First Fit colouring algorithm is given the set of vertices in an arbitrary order and the job of the algorithm is to assign lowest legal colour to each vertex. First fit is an $O(n)$ time algorithm.

Degree Based Ordering : Degree Based Ordering is considered a better strategy as compared to First Fit because it uses a certain criterion for choosing the vertex to be coloured from a set of uncoloured vertices rather than picking the next vertex from an arbitrary order. Some of the well known Degree based Ordering techniques are Largest Degree Based Ordering and Saturation Degree

Based Ordering.

Largest Degree Based Ordering : Ordering the vertices by decreasing degree proposed by C. Avanthay, A. Hertz, N. Zufferey [?]. it is considered one of the oldest ordering strategies. Suppose the vertices v_1, v_2, \dots, v_{i-1} have been chosen and coloured. Vertex v_i is chosen to be the vertex with the maximum degree among the set of uncoloured vertices. Largest Degree Ordering provides a better colouring than First Fit since during each iteration chooses a vertex with the highest number of neighbours which potentially produces the highest colour. Largest Degree Based Ordering Algorithms has a complexity of $O(n^2)$.

Saturation Degree Based Ordering : Saturated degree ordering was proposed by E. Falkenauer [?]. The saturation degree of a vertex is defined as the number of differently coloured vertices the vertex is adjacent to. Saturation Degree Based Ordering works as follows : Suppose that vertices v_1, v_2, \dots, v_{i-1} have been chosen and colored. Then at step i , vertex v_i with the maximum Saturation degree is selected. While choosing a vertex of maximum saturation degree, ties are broken in favor of the vertex with the largest degree.

Saturation Degree Based Ordering provides better colouring as compared Largest Degree Based Ordering since it first colours the vertices most constrained by previous colour choices. Saturation Degree Based Ordering algorithm has a complexity of $O(n^2)$.

2 Graph Colouring with WebMathematica

WebMathematica is a technology developed by Wolfram research which enables instructors to create websites that allow users to compute and visualize results directly from a web browser. It is based on a standard Java technology called servlets. It allows a site to deliver HTML(Hyper Text Markup Language) pages that are enhanced by the addition of Mathematica commands [?]. When a request is made for one these pages, the commands are evaluated and the computed result is placed in the page.

In this section we give applications to colour the vertices of the graphs with *WebMathematica*.

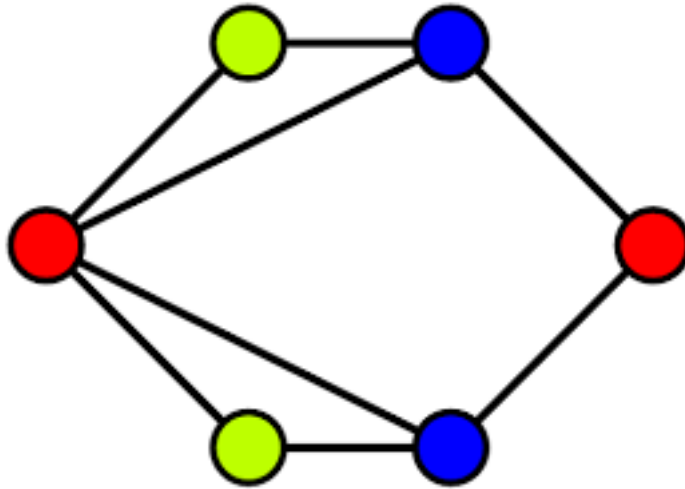
2.1 Vertex Colouring

The most applications involving vertex colouring are concerned with finding the minimum number of colours required so that two adjacent vertices cannot have the same colour. A proper vertex colouring of a Graph is an assignment from its vertex set to a coloured set that any two adjacent vertices are assigned two different colours.

The chromatic colour of a Graph G denoted by $\chi(G)$, is the minimum number of different colours required for a proper vertex colouring of G .

Applications of vertex colouring include timetable scheduling, assignment of radio frequencies and computer optimization. We use commands in the Combinatorica package with Mathematica to colour the vertices of the graph given

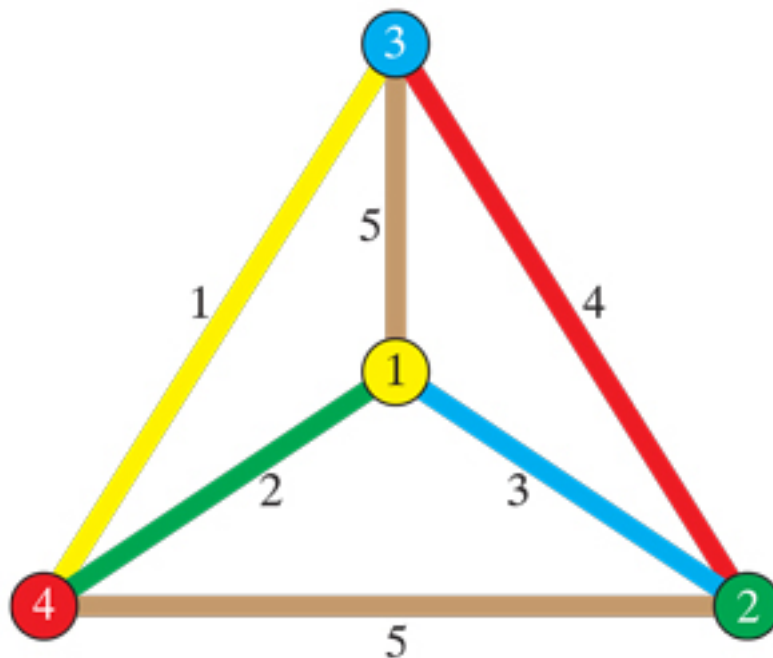
below[?].



2.2 Edge Colouring

Edge colouring is an optimization problem: An edge colouring of a graph G is an assignment of colours to the edges of G such that edges with a common endpoint have different colours. The problem to colour edges is an NP problem which means that the algorithm used for the solving th problem does not have a polynomial time complexity.

Commands in Mathematica can be used to colour edges of graphs and give web-based examples with *WebMathematica*[?].



If the user presses the Wheel button and enter the number of vertices, he/she can get the dge-coloured graph.

3 Generating Graphs with WebMathematica

Some of the most important operations on Graphs are sum, union, join and product of two graphs etc. These operations can be performed on common graphs, complete graphs, cycle, wheel and random tree using *Web Mathematica*.

Join of two graphs : The union of two graphs with addition of edges between all pairs of vertices from different graphs is known as the join of two graphs.

Union of two graphs : The union of two graphs is formed by taking the union of the vertices and edges of the graphs. Thus the union of graphs is always disconnected.

Sum of two graphs : The sum operation of two graphs is to take the edges of the second graph and add them to the first graph.

Product of two graphs : The product of two graphs $G \times H$ has a vertex set defined by the Cartesian product of the vertex sets of G and H . There is an edge between (u,v) and (s,t) if $u = s$ and v is adjacent to t in H or $v = t$ and u is adjacent to s in G .

4 Cycle Structure in Graphs with WebMathematica

A Cycle in a graph is a simple closed path. We will represent a cycle in Graph G as a list of vertices $C = v_1, v_2, \dots, v_1$ such that there is an edge of G from each vertex to the next in G .

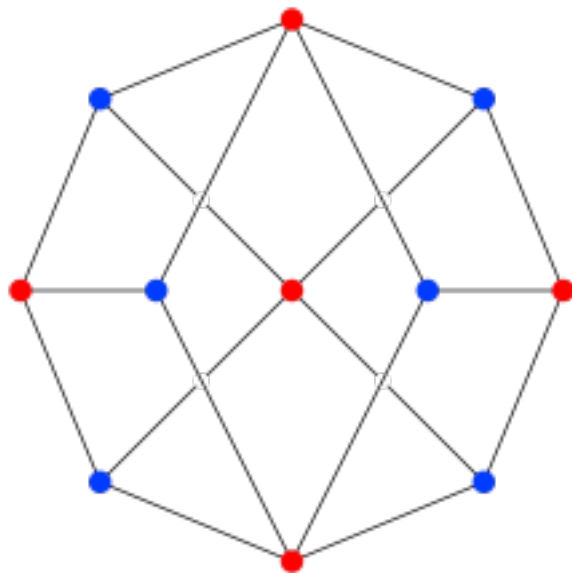
4.1 Eulerian Cycle

An Eulerian Cycle, also called a Eulerian Circuit is a path with starts and ends at the same vertex in graph G . In other words it is a graph cycle which uses each vertex edge exactly once.

Euler initiated the study of Graph theory in 1736 with the famous seven bridges of Königsberg Problem the town of Königsberg straddled the Pregel river with a total of seven bridges connecting the two shores and two islands. The townsfolk were interested in crossing every bridge exactly once and returning to the starting point.

An Eulerian cycle is a complete tour of all the edges of a graph. The term circuit is often used instead of cycle since each vertex can be visited more than once.

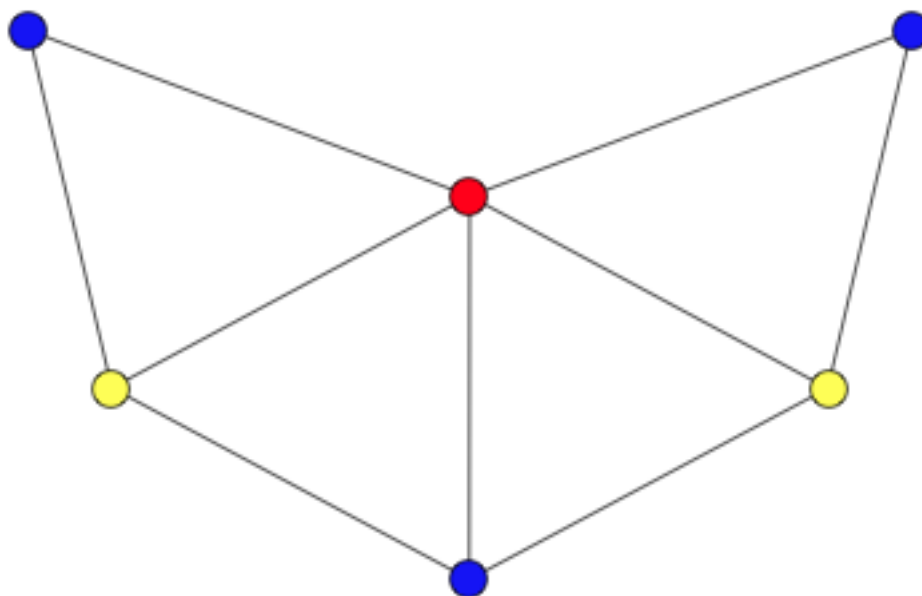
We use *WebMathematica* to find the Eulerian cycle and to give web based examples. If the number of the vertices is entered, it is possible to see the Eulerian cycle in that graph if there exists one [?].



4.2 Hamiltonian Cycle

Hamiltonian Cycle of a Graph G is a cycle which visits each vertex in G exactly once, as opposed to a Eulerian Cycle which visits each edge exactly once. A Hamiltonian Path is a path between two vertices of a graph that visits each vertex exactly once. The problem of computing a Hamiltonian Cycle is fundamentally different from the problem of computing a Eulerian Cycle because testing whether a graph is Hamiltonian is NP complete which means that the algorithm used for computing whether a graph contains a Hamiltonian circuit does not have a polynomial time complexity.

We use *WebMathematica* to find the Hamiltonian cycle and to give web based examples. If the number of the vertices is entered it is possible to see the Hamiltonian cycle in that graph if there exists one[?].



ations research, and complex analysis, real analysis, functional analysis, and topology, algebra, graph theory, and combinatorics, combinatorics, topology, and functional analysis, operations research, graph theory, and coding theory, operations research, graph theory, and number theory, algebra, number theory, and coding theory, algebra, operations research, and real analysis.

Let S be a set of students, $P = \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10\}$ be the set of examinations respectively algebra, real analysis, topology, operational research, complex analysis, functional analysis, graph theory, combinatorics, coding theory, and number theory. $S(p)$ be the set of students who will take the examination $p \in P$. Form a graph $G = G(P, E)$, where $a, b \in P$ are adjacent if and only if $S(a) \cap S(b) \neq \Phi$. Then each proper vertex colouring of G yields an examination schedule with the vertices in any color class representing the schedule on a particular day. Thus $\chi(G)$ gives the minimum number of days required for the examination schedule.

5 days are required and you can see below the lessons in the same parenthesis which are on the same day $\{\{1, 6\}, \{2, 8, 9\}, \{3, 4\}, \{5, 7\}, \{10\}\}$

References

- [1] Jonathan, G and Jay, Y : Graph theory and its application, CRC Press, (1999).
- [2] Pemmaraju, S. and Skiena, S. : Computational Discrete Mathematics, Cambridge University Press, (2003).
- [3] Ufuktepe, U. Bacak, G. and Beseri, T. : Graph colouring with *WebMathematica*, Springer-Verlag, (2003).
- [4] Google Images
- [5] C. Avanthay, A. Hertz, N. Zufferey, A variable neighborhood search for graph coloring, European Journal of Operational Research 151 (2) (2003) 379–388.
- [6] E. Falkenauer, A hybrid grouping genetic algorithm for bin packing, Journal of Heuristics 2 (1) (1996) 5–30.