Graph Coloring with web*Mathematica*

Ünal Ufuktepe, Goksen Bacak, and Tina Beseri

Izmir Institute of Technology, Department of Mathematics
Urla, Izmir, TURKEY
{unalufuktepe,goksenbacak,tinabeseri}@iyte.edu.tr

Abstract. Coloring of a graph is an assignment of colors either to the edges of the graph G, or to vertices, or to maps in such a way that adjacent edges/vertices/maps are colored differently. We consider the problem of coloring graphs by using web*Mathematica* which is the new web-based technology. In this paper, we describe some web-based interactive examples on graph coloring with web*Mathematica*.

1 Introduction

A graph G=(V,E) is a mathematical structure consisting of two sets V and E. The elements of V are called vertices and the elements of E are called edges.

web*Mathematica* is based on a standard Java technology called servlets. It allows a site to deliver HTML pages that are enhanced by the addition of *Mathematica* commands. When a request is made for one of these pages the *Mathematica* commands are evaluated and the computed result is placed in the page. This is done with the standard Java templating mechanism, Java Server Pages (JSPs), making use of a library of tag extensions called the MSP Taglib; examples of these for graph coloring are given in a later section [8]. We developed some modules and used Combinatorica package [5] to color the graphs by webMathematica.

1.1 Edge Coloring

Edge coloring is an optimization problem: Given a graph, how many colors are required to color its edges in such a way that no two edges which share an endpoint receive the same color? A k-edge coloring with k colors G is k-edge colorable if a k-edge colors exists. The smallest k for which G is k-edge colorable is called the edge-coloring number of G.

Definition 1. *The edge-coloring number of G is called the chromatic index of G and is denoted by $\chi'(G)$.*

$\Delta(G)$ is the maximum vertex degree in G. An obvious lower bound for $\chi'(G)$ is $\Delta(G)$ since the edges incident with one vertex must be differently colored. It follows that $\chi'(G) \geq \Delta(G)$. On the other hand, Vizing has proved in 1964 that any simple graph G has an edge coloring with at most $\Delta(G) + 1$ colors:

Proposition 1 (Vizing, 1964). *Every graph G satisfies,*

$$\Delta(G) \leq \chi'(G) \leq \Delta(G) + 1$$

1.2 Vertex Coloring

The most applications involving vertex-coloring are concerned with determining the minimum number of colors required under the condition that the end points of an edge can't have the same color. A proper vertex-coloring of a graph is an assignment from its vertex set to a color set such that the endpoints of each edge are assigned two different colors. The chromatic number of a graph G, denoted by $\chi(G)$, is the minimum number of different colors required for a proper vertex coloring of G. Thus, $\chi(G) = k$ if graph G is k-colorable but not $(k-1)$-colorable. A graph G has $\chi(G) = 1$ if and only if G has no edges. A graph G has $\chi(G) = 2$ if and only if it is bipartite. The study of vertex coloring of graphs is customarily restricted to simple graphs. A graph with a self-loop is regarded as uncolorable, since the endpoint of the self-loop is adjacent to itself. Moreover, a multiple adjacency has no more effect on the colors of its endpoints than a single adjacency. Although the chromatic number is one of the most studied parameters in graph theory, no formula exists for the chromatic number of an arbitrary graph. Thus, we must try to find bounds for the chromatic number.

1.3 Map Coloring

A map on a surface is an imbedding of a graph on that surface. An k-coloring of a map is a coloring with k-colors. A map coloring is proper if each $e \in E_G$, the regions that meet an edge e are colored differently. The chromatic number of a map is the minimum number of colors needed for a proper coloring.[4,7]

The chromatic number of a map equals the chromatic number of its dual graph.

2 Applications with web*Mathematica*

To understand what web*Mathematica* can do regarding coloring planner graph and maps we need to know Combinatorica, the standard package that has many functions for dealing graphs.

We mainly use this package by adding the following ColorVertices, ColorEdges, and DrawG modules:

```
ColorVertices[g_] := Module[{c, p, s},
c = VertexColoring[g];
p = Table[Flatten[Position[c, i]], {i, 1, Max[c]}];
s = ShowLabeledGraph[ Highlight[g, p]]]
```

The module ColorVertices colors vertices of the given graph g.

```
ColorEdges[g_] := Module[{k, e, m, kk, r, s},
k = EdgeColoring[g]; e = Edges[g];
m = Max[k]; kk = Table[i, {i, 1, 1000}];
r = For[i = 1, i <= m ,
For[j = 1, j <= M[g], If[k[[j]] == i,
kk[[j]] = Hue[i/m]]; j++]; i++] ;
s = ShowGraph[g, Table[{e[[i]], EdgeColor -> kk[[i]]}, {i, 1,
M[g]}],
VertexNumber -> On, EdgeStyle -> Thick]]
```

The module ColorEdges colors vertices of the given graph g.

```
DrawG[elist_]:=Module[{edgelist=elist,size,vertlist,vnum},
size=Length[edgelist];
vertlist=Union[Flatten[edgelist]];
vnum=Length[vertlist];
Do[edgelist[[i]]={edgelist[[i]]},{i,size}];
vertlist=CompleteGraph[vnum][[2]];
Graph[edgelist,vertlist]]
```

The module DrawG draw the simple graph without isolated points. DrawG takes as input the list of edges of a graph. The vertices of the graph of order n must be labeled consecutively $1, 2, 3, \ldots, n$. This module must be added to the package DiscreteMath`Combinatorica`.

The package Combinatorica must be loaded before running the program.

webMathematica allows the generation of dynamic web content with Mathematica. The following example draws the given graph and colors edges and vertices

```
<%@ page language="java" %>
<%@ taglib uri="/webMathematica-taglib" prefix="msp" %>
<html><head> <title>Graph Coloring </title>
</head> <body bgcolor="# ffffff">
<h1>Graph Coloring </h1>
<FORM ACTION="graphcolor.jsp" METHOD="POST">
Input the list of the edges in order as follows:
<msp:allocateKernel>
<INPUT type="text" name="v" ALIGN="LEFT" size="50"
Value ="<msp:evaluate> MSPValue[ $$v, "{{1,4},{2,3},
{2,4},{3,4}}"]</msp:evaluate>" />
<msp:evaluate> <<DiscreteMath`Combinatorica`
<<Graphics`Colors`
</msp:evaluate>
<h3>Vertex Coloring and Edge coloring are </h3><p>
```

```
<msp:evaluate> input=True;
If[MSPValueQ[$$v],
n=MSPToExpression[$$v]; g=DrawG[n];
MSPShow[ColorVertices[g]],input=False;]
</msp:evaluate>
<msp:evaluate>
If[MSPValueQ[$$v],
MSPShow[ColorEdges[g]],input=False;]
</msp:evaluate> </p> <p>

   The Chromatic Number of the given graph is
<msp:evaluate>
kn=ChromaticNumber[g]
</msp:evaluate></p>
<INPUT TYPE="Hidden" NAME="formNo" VALUE="1">
<INPUT TYPE="Submit"NAME="taskValue"
VALUE="Color the graph's edges and vertices" >
</msp:allocateKernel>
</form></body></html>
```

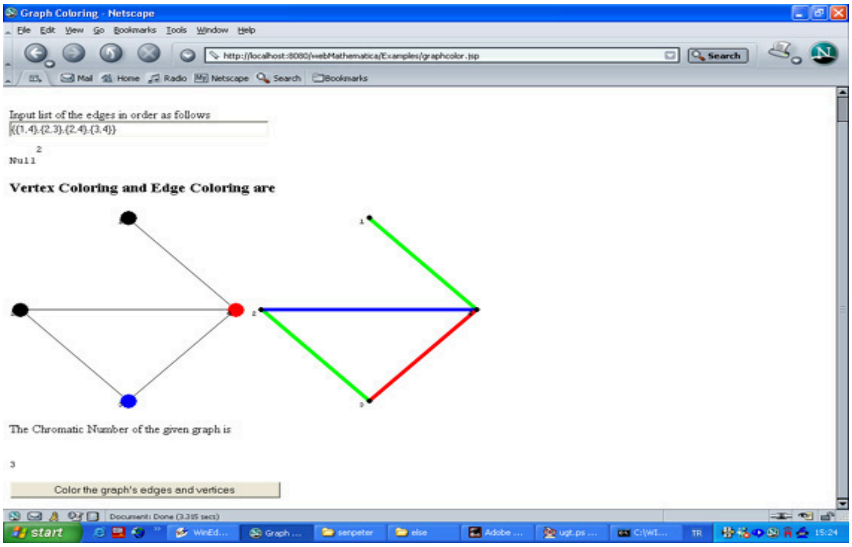
A form element is a block of HTML that may contain input elements. A form may be activated with an input of type submit. The action attribute refers to a URL that accessed when the form is activated. The method attribute tells the browser what HTTP method to use, in this case, a post method. This example has two input tags. The first allows the user of the page to enter the list of edges of the graph, and the second specifies a button that, when pressed, will submit the form. When the form is submitted, it will send information from input elements to the URL specified by the action attribute. This information is sent to a Mathematica kernel and assigned to a Mathematica symbol (see Fig.1). The name of the symbol is given by $$ to the value of the name attribute. When a value entered in the text field and the "Color the graph's edges and vertices" button pressed,the text is displayed. This example also shows the use of the MSP functions MSPShow, MSPValue, and MSPToExpression.

To color edges and vertices of the standard graphs (Complete graph, wheel, tree, cycle, and the others) we need more inputs. If the web user selects one of the standard graphs and its number of vertices then they can get easily colored graph and its chromatic numbers as follows:

```
   <FORM ACTION="familycolor.jsp" METHOD="POST">
<p>Please select one of the following graphs and
input into the box:(1,2,3,or 4) </p> <p>1-CompleteGraph,</p>
<p>2-RandomTree, </p> <p>3-Wheel,</p> <p>4-Cycle)</p>
<msp:allocateKernel>
```

```
<INPUT type="text" name="m" ALIGN="LEFT" size="6"
value="<msp:evaluate> MSPValue[$$m,"1"]</msp:evaluate>" />
Input the number of the vertices for the selected graph:
<INPUT type="text" name="n" ALIGN="LEFT" size="6"
value="<msp:evaluate> MSPValue[$$n,"5"]</msp:evaluate>" />
<msp:evaluate> <<DiscreteMath`Combinatorica`
<<Graphics`Colors`</msp:evaluate>
<h3>Vertex Coloring and Edge Coloring are </h3>
<msp:evaluate> MSPBlock[{$$m,$$n},
Which[$$m==1,MSPShow[ColorVertices[CompleteGraph[$$n]]],
$$m==2, MSPShow[ColorVertices[a=RandomTree[$$n]]],
$$m==3, MSPShow[ColorVertices[Wheel[$$n]]],
$$m==4, MSPShow[ColorVertices[Cycle[$$n]]]]]
</msp:evaluate>
<msp:evaluate> MSPBlock[{$$m,$$n},
Which[$$m==1, MSPShow[ColorEdges[CompleteGraph[$$n]]],
$$m==2, MSPShow[ColorEdges[a]],
$$m==3, MSPShow[ColorEdges[Wheel[$$n]]],
$$m==4, MSPShow[ColorEdges[Cycle[$$n]]]]]
</msp:evaluate>
The Chromatic Number of the selected graph is <msp:evaluate>
MSPBlock[{$$m,$$n},
Which[$$m==1, ChromaticNumber[CompleteGraph[$$n]],
$$m==2, ChromaticNumber[a],
$$m==3, ChromaticNumber[Wheel[$$n]],
$$m==4, ChromaticNumber[Cycle[$$n]]]]
</msp:evaluate>
<p> The Edge Chromatic Number of the selected graph is
<msp:evaluate>
MSPBlock[{$$m,$$n},
Which[$$m==1, EdgeChromaticNumber[CompleteGraph[$$n]],
$$m==2, EdgeChromaticNumber[a],
$$m==3, EdgeChromaticNumber[Wheel[$$n]],
$$m==4, EdgeChromaticNumber[Cycle[$$n]]]]
</msp:evaluate> </p> <input type="submit" name="button"
value="Graph Color"> </msp:allocateKernel>
```

We need one more input tag to color edges and vertices of the bipartite graph one in this script. Combinatorica package uses Brelaz's heuristic to find a good, but not necessarily minimal, edge coloring of graph G. Then when the web user enters the huge number of the vertices he/she might get interesting results or time out error for the edge coloring. We also tried to color maps by using GraphColoring package of Stan Wagon and J.P.Hutchinson with webMathematica but since there are some version problems we left it for the future works [4,7].


Graph Coloring - Netscape
File Edit View Go Bookmarks Tools Window Help
http://localhost:8080/webMathematica/Examples/graphcolor.jsp
Search
Mail Home Radio Netscape Search Bookmarks
Color the graph's edges and vertices
Document: Done (3.315 secs)
start
15:24

Input list of the edges in order as follows

{{1,4},{2,3},{2,4},{3,4}}

```
    2
Null
```

Vertex Coloring and Edge Coloring are

The Chromatic Number of the given graph is

3

Fig. 1. A view of graphcolor.jsp after information entered by the user

Please check
http://gauss.iyte.edu.tr:8080/webMathematica/Examples/graphcolor.jsp
http://gauss.iyte.edu.tr:8080/webMathematica/Examples/familycolor.jsp
for the details of these JSP scripts.

References

1. Appel,K.I., Haken,W., and Koch,J.: Every planar map is four colorable I: Discharing, Illinois J.Math, (1977)
2. Birkhoff,G.D. and Lewis,D.C.: Chromatic polinomials, Trans.Amer.Math.Soc., 60:355-451, (1946)
3. Gross,J. and Yellen,J.: Graph Theory and Its Applications, CRC Press, (1999)
4. Hutchinson,J.P. and Wagon,S.: The four-color theorem, Mathematica in Education and Research, 42-51, 6:1 (1997)
5. Skiena,S.: Implementing Discrete Mathematics-Combinatorics and Graph Theory with Mathematica, Addison-Wesley Publishing Company, (1990)
6. Soaty,T.L. and Kainen,P.C.: The four color problem, Dover, New York, (1986)
7. Wagon,S.: An April Fool's hoax, Mathematica in Education and Research, 46-52, 7:1 (1998)
8. Wickham,T.: webMathematica A User Guide, Wolfram Research, Inc., (2002)
9. Wolfram,S.: The Mathematica Book, Cambrigde Univ. Press, (1996)