

Visually Pleasing Graph Representation Using ‘GraphDraw’

December 23, 2015 · Mathematics

Sharan Maiya

Maiya S. Visually Pleasing Graph Representation Using ‘GraphDraw’. Young Scientists Journal. 2015 Dec 23 [last modified: 2016 Aug 16]. Edition 1.

Abstract

Introduction

In Graph Theory, a branch of Discrete Mathematics, a graph is defined as a collection of vertices, commonly drawn as circles, with some edges, drawn as lines, between them.

They can be used to show links between related entities in a way that is easy to understand and interpret. For example they are used to model situations such as traffic networks or chemical bonds in a compound. It often becomes necessary to develop a representation of a network which exhibits certain specific properties, such as finding graphs of a particular girth (the girth refers to the length of the shortest cycle within the graph). The project looked at graphs of girth 5 – graphs which include no triangles or squares – of order between 20 and 32. These graphs, which were previously unknown, are studied in a paper being written at The University of Glasgow¹. The program ‘GraphDraw’, which was developed at the university recently, uses symmetry to represent graphs². Graphs can be drawn in many ways and it was preferable to pick the most ‘visually pleasing’ representation of these graphs. When a graph is drawn in a visually pleasing way it is much easier to analyse and identify its structure³.

Aims

There were two main aims of this project. The first was to draw all of the graphs of girth 5 for the cases $20 \leq v \leq 32$ in the most visually pleasing way possible through the use of the program GraphDraw. The second aim of this project was to evaluate the software GraphDraw with respect to its effectiveness and scope, and to suggest any improvements for future iterations.

Applications

Graph theory has many intriguing applications in the modern world, particularly with regards to Computer Science. For example graph colouring can model problems where we have limited resources, and constraints on how they may be used together, such as in timetabling and job scheduling. They are also used when conducting studies on social networks, a very important topic in recent times. Other applications include designing efficient transport systems for future cities and routing traffic on the internet⁴.

Definitions

- A *graph* is a set of vertices with a set of edges connecting them. We write $G = (V, E)$ where V is the set of vertices and E is the set of edges between the vertices. Edges are denoted $(1, 2)$ which would represent an edge between the vertices 1 and 2.
- Graphs can be drawn in various different ways. In this project, they are drawn by representing the vertices with ovals and the edges as lines connecting the ovals. If there exists an edge between two vertices 1 and 2 then these two vertices are said to be *adjacent*.
- The *order* of a graph is the number of vertices.
- A path along some of the vertices of a graph that begins and ends on the same vertex is known as a *cycle*.
- The *girth* of a graph is defined as the length of the shortest cycle within the graph.
- A graph G can be represented by what is known as its *adjacency matrix*. This can make it easier for a computer to interpret. For a graph of order n , the adjacency matrix is an $n \times n$ matrix A . The entry for $(1, 2)$ on this matrix will be a ‘1’ if the vertices 1 and 2 are adjacent and a ‘0’ if they are not. For example, a simple graph and its adjacency matrix are given below:

Article Image

Image not found

<https://archive.archive.ysjournal.com/wp-content/uploads/2015/11/Article-Image-300x161.png>

- An *isomorphism* is a bijective mapping from V_1 to V_2 such that v_1 is adjacent to v_2 if and only if $f(v_1)$ is adjacent to $f(v_2)$ (Note: a function is described as bijective if it is both injective and surjective. That is, there is a one-to-one correspondence).

isomorphic

Image not found

<https://archive.archive.ysjournal.com/wp-content/uploads/2015/11/isomorphic-300x210.png>

- An isomorphism of G_1 to itself is known as an *automorphism* of G_1 . Automorphisms may be composed to form new automorphisms, and every automorphism has an inverse. The set of all automorphisms of G_1 equipped with composition and inverse, is the *automorphism group* of G_1 , denoted $\text{Aut}(G_1)$. There are several packages available which can be used to find the automorphism group of a graph e.g. NAUTY. In this project simple GAP programs were used (these program would call NAUTY) to find the automorphism group of some graphs.

Visually Pleasing Graphs

A drawing of a graph must be 'visually pleasing' in order to make analysing it a much simpler task. That is, the representation of the graph must show underlying structure. There are various factors which contribute to how attractive a drawing of a graph looks. Research has shown that symmetry is one of the most important properties of a graphical representation when trying to make it visually pleasing. Humans are drawn to symmetry in many different aspects of life, and a graph which shows at least partial symmetry can be easier to interpret and identify certain properties. In GraphDraw, graphs can be shown to have complete (or partial) symmetry by arranging the vertices in clusters based on part of the automorphism group.

Another important factor when creating a visually pleasing graph is to arrange the vertices on a known geometrical shape, such as a triangle. In GraphDraw, the chosen shape was a circle as it could allow for an undefined number of vertices to be placed on it relatively easily. There were also two secondary factors which were taken into account. They were the number of edge crossings, and the total length of all edges. When these factors are minimised, the graph is more visually pleasing.

Once all of the representations were created they were analysed based on the factors given above in order to determine whether they were visually pleasing. A comparison of two very different representations of the same graph is given below:

article image 4

Image not found

<https://archive.archive.ysjournal.com/wp-content/uploads/2015/11/article-image-4-300x122.png>

In the representation on the left, the graph has been drawn randomly; no method or algorithm has been used to make it more visually pleasing. As it is, the graph is clearly not very easy on the eyes. The vertices are all over the place making it seem jagged and messy and the placement of some of the vertices has resulted in it being extremely difficult to trace some of the edges, especially those nearer the centre of the graph as many overlap and cross over each other. Looking at the graph as a whole, it is very difficult to identify any type of structure or property.

The graph on the right is the same graph shown before, this time using various settings allowed within GraphDraw. Immediately it looks much better than its counterpart. The vertices are arranged on the shape of a circle, with groups of vertices in clusters based on the automorphism group. This causes the graph to show partial symmetry and means that its basic structure is easy to understand. The edges are easy to discern from one another as there are few confusing edge crossings. It should also be noted that most of the edges are of similar length; this also makes them easier to identify than if they were of varied lengths as in the previous graph.

A Selection of Results

article graph 1

article graph 2

Image not found

<https://archive.archive.ysjournal.com/wp-content/uploads/2015/11/article-graph-2-300x247.png>

article graph 3

Image not found

<https://archive.archive.ysjournal.com/wp-content/uploads/2015/11/article-graph-3-300x258.png>

article graph 4

Image not found

<https://archive.archive.ysjournal.com/wp-content/uploads/2015/11/article-graph-4-300x259.png>

article graph 5

Image not found

<https://archive.archive.ysjournal.com/wp-content/uploads/2015/11/article-graph-5-300x251.png>

article graph 6

Image not found

<https://archive.archive.ysjournal.com/wp-content/uploads/2015/11/article-graph-6-300x244.png>

article graph 7

Image not found

<https://archive.archive.ysjournal.com/wp-content/uploads/2015/11/article-graph-7-300x273.png>

article graph 8

Image not found

<https://archive.archive.ysjournal.com/wp-content/uploads/2015/11/article-graph-8-300x275.png>

article graph 9

Image not found

<https://archive.archive.ysjournal.com/wp-content/uploads/2015/11/article-graph-9-300x278.png>

article graph 10

Image not found

<https://archive.archive.ysjournal.com/wp-content/uploads/2015/11/article-graph-10-300x272.png>

Evaluation of GraphDraw

After the evaluation of GraphDraw was written, a list of possible improvements to the program was made. They included:

- Include a function to remove an edge or vertex.
- Allow for the ability to move a vertex.
- Allow for the ability to rotate a graph.
- Add an 'Undo' button.
- Allow the user to change the labelling of the vertices.
- Allow the user to export an image of a graph as an SVG image instead of a JPEG image (this feature was implemented during the project).

In the original version of GraphDraw the user has the ability to save a JPEG image of a graph meaning the computer stores the image as an array of pixels (a bitmap image). The resulting picture is *resolution dependent* increasing the size of the image can result in it appearing blurry and pixelated. An SVG (vector) image is stored when the computer records the key shapes in the picture. These could be the coordinates of a line or a circle for example. This is much more suitable for relatively simple images and means the picture will be *resolution independent*. By editing the source code to include a button to save the graphs as SVG files, the final images could be resized without any apparent loss of quality. Two sections of the same graph are shown below. The image on the right is from a JPEG file and its counterpart is from an SVG file:

article image 5

Image not found

<https://archive.archive.ysjournal.com/wp-content/uploads/2015/11/article-image-5-300x163.png>

Note: These advantages of storing images as vector graphics are only relevant as the images are relatively simple. For complex detailed images (e.g. a photograph taken on a digital camera) it may be more advantageous to use a bitmap image.

Conclusion

The first and main aim of the project was to use the program GraphDraw to create the most visually pleasing representations of all of the graphs of girth 5 for the cases $20 \leq v \leq 32$ to be included in a research paper in development. All of these graphs were drawn successfully and the aim was achieved. The secondary aim was to provide useful feedback on the usability of GraphDraw with a view to any future improvements which could be made. This aim was also achieved and a brief list of improvements were suggested for future iterations of the software. Many of them were beyond the initial scope of the project however one improvement was implemented during the course of the research.

It would be pleasing to see GraphDraw become a standard tool for Graph Theory papers, but it currently lacks some key features which would make it much more usable. It is suggested that these features be implemented in future work.

References

1. Mike Codish, Alice Miller, Peter Stuckey and Patrick Prosser. Graphs of girth 5 with orders between 20 and 32. In preparation.
2. Dragos Miron. Symmetrical Graph Drawing. Level 4 project, School of Computing Science, University of Glasgow, 2015.
3. Stephen M. Kosslyn. Graph Design for the Eye and Mind. Oxford University Press, 2006.
4. S.G.Shrinivas, S.Vetrivel and N.M.Elango. Applications of Graph Theory in Computer Science an Overview. International Journal of Engineering Science and Technology, Vol. 2(9), 4610-4621, 2010.