

Graph Plotter: a Software Tool for the Investigation of Fault-Tolerant Gossip Graphs

Vilyam, Hovnanyan

Institute for Informatics and
Automation Problems
NAS of Armenia, 0014 Yerevan,
Armenia

e-mail: william_93@mail.ru

Vahagn, Poghosyan

Institute for Informatics and
Automation Problems
NAS of Armenia, 0014 Yerevan,
Armenia

e-mail: povahagn@gmail.com

Suren, Poghosyan

Institute for Informatics and
Automation Problems
NAS of Armenia, 0014 Yerevan,
Armenia

e-mail: psuren55@yandex.ru

ABSTRACT

The gossip problem (telephone problem) is an information dissemination problem where each of n nodes of communication network has a unique piece of information that should be transmitted to all the other nodes using two-way communications (telephone calls) between the pairs of nodes. During a call between the given two nodes, they exchange the whole information known to them at that moment. The k -fault-tolerant gossip problem is a generalization of the gossip problem, where at most k arbitrary faults of calls are allowed. In this paper we present a graph plotter software tool, which uses the interactive interface and automated algorithms to construct arbitrary graphs, and has an ability to check the level of fault-tolerance and other characteristics of the graphs.

Keywords

Networks, telephone problem, gossip problem, fault-tolerant communication.

1. INTRODUCTION

In our studies we have used a Graph Plotter software tool. The graphical interface of the tool is made by WPF technology, which provides quicker and easier job especially with graphs with a large number of vertices and edges.

The main purpose of the tool is to check for a given input graph whether it satisfies the given level of fault-tolerance. In addition, it is possible to check whether the input graph is a NOHO (No One Hears Own) [[2]] or a NODUP (No Duplicates) [[3]] graph. The NOHO graphs are graphs, which do not contain any node that listens to its own information, or equivalently, do not contain a cycle. The NODUP graphs are defined as the graphs, whose nodes listen to each piece of information exactly once. It means that there is exactly one increasing path between two arbitrary vertices.

In addition, the system provides a convenient means of working with graphs, such as, for example, auto numbering of vertices and edges of the graph, in order to show the paths between two vertices it is possible to paint vertices and edges with any color, if necessary, add a new edge with proper weight or sort weights of all edges with integer values, etc.

In addition to all this, the system makes it possible to change the position of vertices adjacent to the current edge, also to permute all edges which are adjacent to the current edge and have bigger (or smaller) weights. The meaning of this action lies in the fact that if there has been a call between two arbitrary vertices at time t since then these two have exactly the same information, so nothing will be changed if from that moment on all nodes that will connect to these two will change their calls direction from one vertex to another, if the considered graph is symmetric. This function might be useful if you need to disperse or collect edges into one channel, since in some modifications of Gossip problem the number of edges in one channel is limited.

The system also has certain restrictions regarding the maximum number of vertices (999), which is connected to the measures of the vertices and time needed for calculations, which has $O(n^2)$ complexity.

It should be noted that the system does not have a sample or an analogue, at least in the field of universal access, and the main reason is that the system can be useful only in a narrow range of problems, so it can be used only for scientific research. Prior to this, exclusively a mathematical approach was used in solving problems of gossiping and the hardware model of calculations was not used.

Graph Plotter is an easily extensible tool. Since *Gossip test*, *NOHO* and *NODUP* modules are written in C++ language and are separate processes in relation to the main process, new modules can be integrated with this software tool very easily.

It is expected to involve such technologies of parallel computing as MPI, OpenMP and CUDA, since it will let us increase the performance of the tool, which is very important for the investigation of the graphs with relatively large number of vertices.

Instrument capabilities and user interface are presented below.

2. USER INTERFACE DESCRIPTION

The interface can be divided into three parts: Toolbar, which is placed at the top of the window and contains management buttons, Canvas, which is designed for painting an input graph and the information windows on the right side of the main window, which allow to enter and receive information concerning the graph.

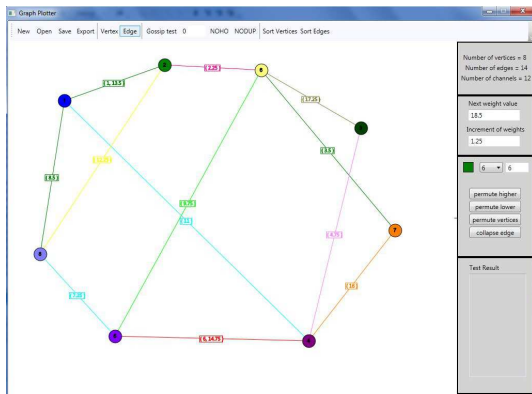


Figure 1. Graph Plotter - demonstration

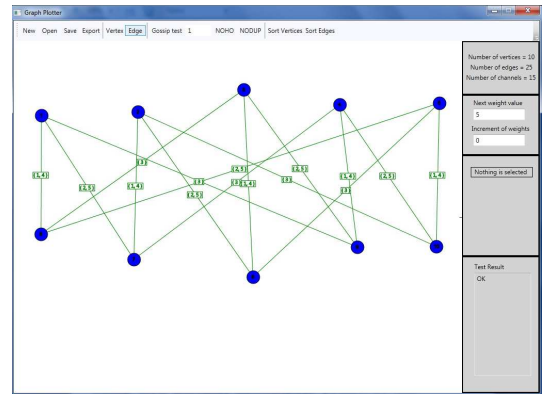


Figure 3. 1-fault-tolerant Knödel graph

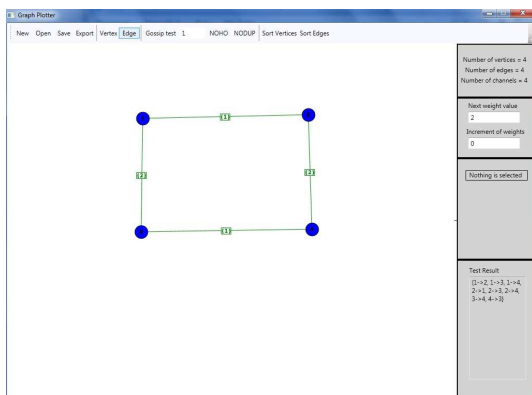


Figure 2. All bad paths of tested graph

The input graph is painted on the panel mounted on the left side of the window. Vertices are painted by clicking the left mouse button, if Toolbar's "Vertex" mode is activated. To make an edge we have to choose two vertices with the left mouse button when Toolbar's "Edge" mode is activated. To choose a vertex or an edge the mouse right button is used. By clicking on it (right mouse button) and pressing "Delete" we can remove the chosen vertex or edge. Vertex shift could be done by choosing the vertex with the right mouse button and holding the pressed state moving it.

Here is a more detailed description of all components. The first information window located on the left side of the main window shows the numbers of vertices, edges and channels. As a channel we mean the set of edges connecting the same vertices.

The next two methods are applied in order to further simplify the process of building a graph:

next weight value - the value of weight of the next edge, which can be changed and assigned integer and float values,

increment of weights - shows the size of the change of the previous parameter.

These two parameters could be changed at any moment. This can be used when we want to add a new edge with an intermediate weight value.

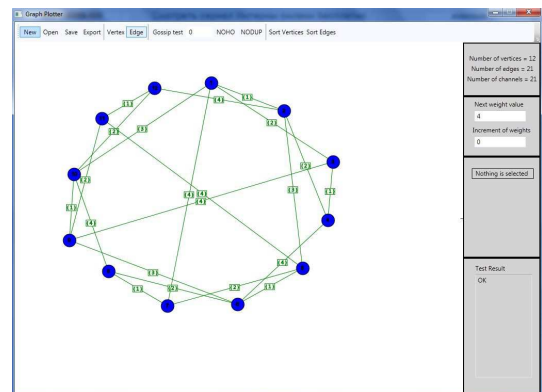


Figure 4. NODUP graph

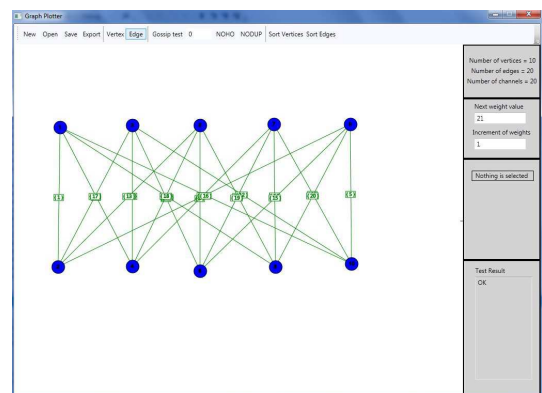


Figure 5. NOHO graph

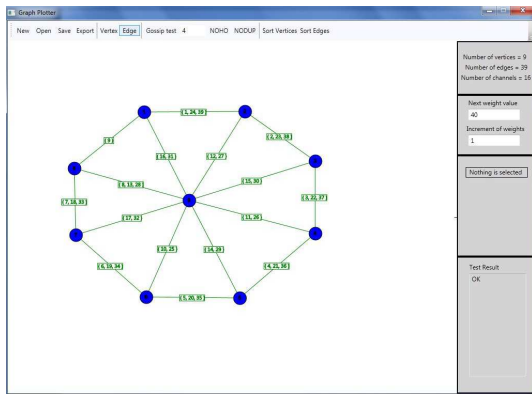


Figure 6. 4-fault-tolerant Combined graph

The next menu is called *Edge/Vertex Parameters* which describe the selected figure and let us change its color. In addition, here we have the following functionalities:

permute higher - permutes all edges adjacent to the selected ones and with higher weight values than the selected ones,

permute lower - permutes all edges adjacent to the selected ones and with lower weight values than the selected ones,

permute vertices - permutes vertices adjacent to the selected edge,

collapse edge - performs the compression of edge, i.e. two vertices adjacent to the selected edge transforms to one vertex with total information.

The last information menu is *Test Result*, which is intended to show information of various tests which can be done with the input graph. For example, in case of "Gossip test" it can be "OK", if the fault-tolerance level of the input graph is k , otherwise the list of all pairs of vertices that do not make k failure-save communication. The same thing can be said about NOHO and NODUP operations.

The buttons on Toolbar panel are divided with separators into four groups. The buttons of the first group are intended to save and load projects (gossip graphs). This group contains the following buttons:

New - cleans the panel which is intended for painting the input graph and prepares it for a new graph,

Open - this button is intended to load the project which already exists,

Save - saves the project (graph) which is painted on the corresponding panel, this file is saved in .mgr extension,

Export - saves painted graphs, but this time in .jpg format (export it as a picture).

The next group consists of two buttons, which are mutually exclusive:

Vertex - if this mode is activated then the vertex will be painted by pressing the mouse left button on the corresponding panel,

Edge - if this mode is activated we have to choose two vertices with the mouse left button and they will be connected with a new edge.

The next and the most important group is intended for studying the main properties of fault-tolerant gossip graphs. It involves the following buttons:

Gossip test - checks the level of fault-tolerance of the input graph, giving the researcher a chance to input and check an arbitrary number as needed fault-tolerance level,

NOHO - checks whether there are cycles in the graph, otherwise the graph is called NOHO graph,

NODUP - checks the presence of duplicate information, if there are not any duplicates then the graph is called NODUP graph.

Gossip test, *NOHO* and *NODUP* buttons are creating new processes that have been developed in C++ language and the main process is waiting until all calculations finish.

The last group contains two buttons:

Sort vertices - makes auto numbering of labels of vertices and is useful in case one or more vertices were removed,

Sort edges - makes auto numbering of edges started with 1 and using only integer values. This is useful in case the new edge was added or one of the edges of the graph was removed.

REFERENCES

- [1] V.H. Hovnanyan, H.E. Nahapetyan, V.S. Poghosyan and Su.S. Poghosyan, "Tighter Upper Bounds for the Minimum Number of Calls and Rigorous Minimal Time in Fault-Tolerant Gossip Schemes", *arXiv:1304.5633* (2013).
- [2] D.B. West, "Gossiping without duplicate transmissions", *SIAM J. Alg. Disc. Meth.* 3 (1982) 418-419.
- [3] Seress, "Quick Gossiping without Duplicate Transmissions" *Graphs and Combinatorics* 2 (1986) 363-381.
- [4] D.B. West, A class of solutions to the gossip problem, part I, *Disc. Math.* 39, 3 (1982) 307-326; part II, *Disc. Math.* 40, 1 (1982) 87-113; part III, *Disc. Math.* 40, 2-3 (1982) 285-310.