Volume

1

GRAPH GENERATOR SERIES

DEPARTMENT OF COMPUTER SCIENCE
UNIVERSITY OF CALIFORNIA RIVERSIDE
JANUARY 2002

AUTHOR

DEREK RANDOLPH DREIER

Manual of Operation:

Barabasí Graph Generator v1.0

UNIVERSITY OF CALIFORNIA RIVERSIDE

Graph Generator Series

Table of Contents

Introduction

Chapter

1

The Barabasí Model

Chapter

2

Operation of the Generator

REQUIREMENTS

This generator was developed using Perl 5 and Perl Tk 8 on a SuSE Linux Platform . The generator has been tested also on both Windows NT and Windows 2000 platforms running Active State Perl 5.22 and Tk 800.022. The generator has a GUI interface and requires the Tk library to operate. To obtain the latest copy of Perl and Tk go to the following sites: http://www.perl.com and http://www.activestate.com . The following minimum hardware requirements are recommended: 200MHz processor with 32MB RAM.

BASIC OPERATION

To begin operation of the generator, at your command prompt type: **perl barabasi.perl**. This will bring up generator as can be seen in Figure 1. There are a lot of features and controls available with this generator as outlined by Figure 1, so lets go through them one by one.

1. Iterations: This sets how many iterations the generator will perform.

2. Initial Nodes: This sets the initial number of nodes before the generator creates the graph.

3. Edges Added: This sets the number of edges added or the maximum number of edges added to a new node on each iteration.. If the check box is set to "Constant" then exactly N edges are added to the new node on each iteration. If the check box is set to "Random" then between 0 – N edges are randomly added on each iteration. N is assumed to be the number of edges entered in the "Edges added" entry box.

4. Probability Distribution: This controls how edges are assigned to existing nodes. A uniform distributions makes each existing node equally likely to have the edge

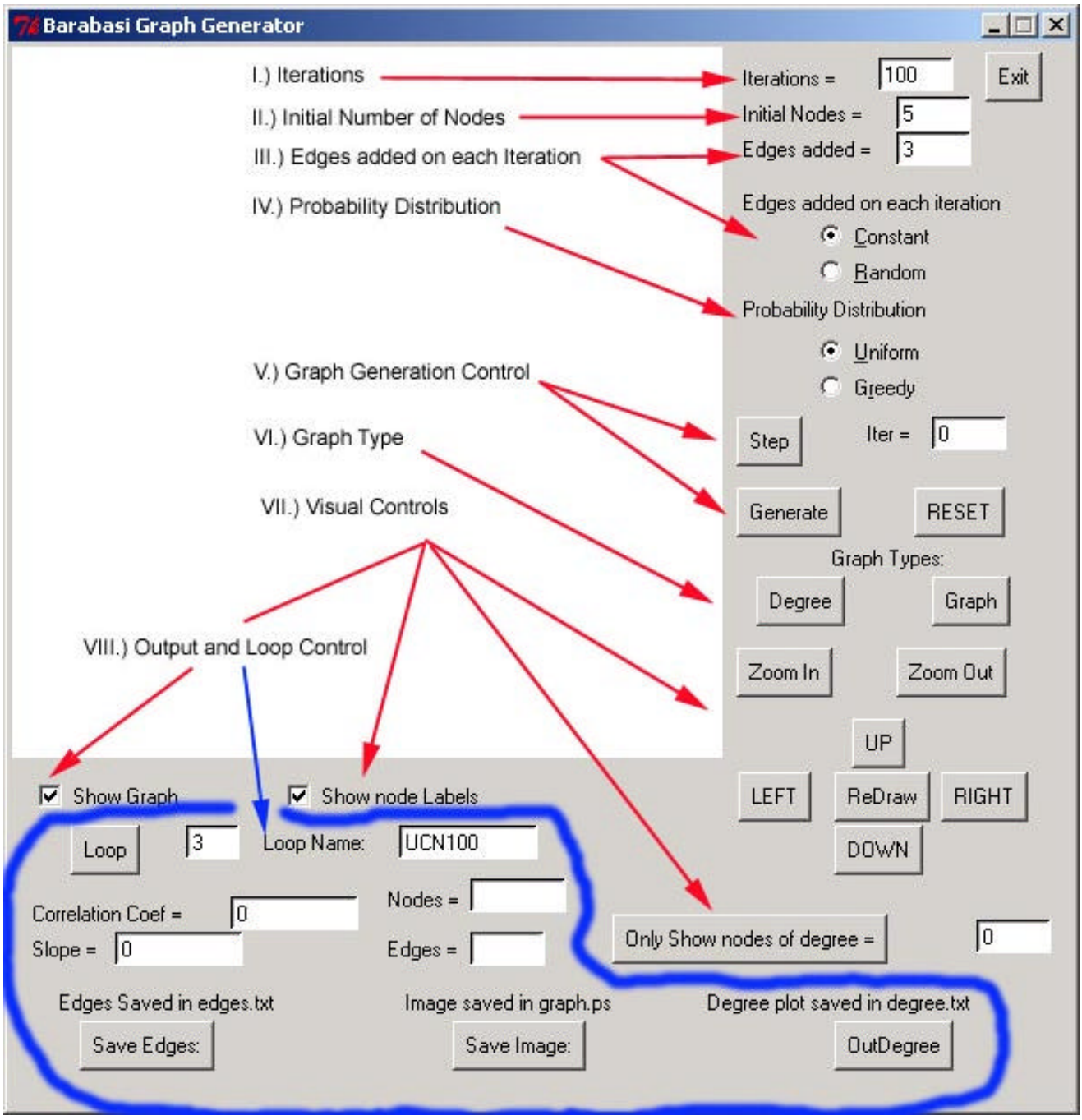

Figure 1: Overview of the Generator

5. Graph Generation Control: This is composed of three buttons. RESET reinitializes the generator and should be clicked each time a new graph is generated. STEP allows one to add one node at a time to the graph. GENERATE generates the graph desired and is controlled by the number of iterations that the user specifies.

6. Graph Type: These are the two possible views. Graph is default which is the actual nodes and edges which make up the graph. Degree shows the power law graph of the nodes of a particular degree versus their frequency.

7. Visual Controls: These are used only when viewing the generated graph. They allow one to zoom in, zoom out, move left, right, up and down. It also allows you to see a subsets of the graph by viewing nodes of only a specified degree.

8. Output and Loop Control: This controls the output generated and an optional control to loop through and create many graphs of a particular type. Correlation Coef and Slope refer to the line of best fit of the power law graph. Nodes and Edges are the number of nodes and edges in the resulting graph. Save Edges out puts the edges to the file edges.txt. Save Image outputs a postscript image to the file graph.ps. What ever is displayed on the canvas when this button is pressed is what gets stored to this file. OutDegree saves the points (x,y) values that make up the power law graph. The loop control has two input values, the number of graphs to generate and the name of the session(loop name). By pressing LOOP, the generator will create N graphs with names suffixed by Loop Name to textfiles, one for the edges, degree, and image. Additionally a file prefixed by "master" followed by Loop Name contains statistics of the graphs such as average correlation coefficient.

Operating the Generator in Increment Mode

This mode was mainly created to test the generator and to visually see the graph being constructed. To operate the generator in Increment Mode is very simple. For example, set the initial number of nodes to 15, Edges Added to 3 with "Constant" checked for Edges Added on each Iteration.. Set the probability distribution to "Uniform". Now click the RESET button to clear the generator. Now click STEP one time and the initial five nodes will appear. Click STEP again and a new node is added along with three new edges. Continue to press STEPto set the graph get generated.. Whenever desired you can press the DEGREE button to see the power law graph and then continue pressing STEP to continue the generation of the graph. The ZOOM IN, ZOOM OUT, UP, DOWN, LEFT, RIGHT, REDRAW can all be used to move around the graph to view different nodes. The ONLY SHOW NODES OF DEGREE= button can be used to just see the nodes with a specific degree. On each press of the STEP button, three values are updated: "Iter", "Correlation Coef", "Slope" , "Edges", "Nodes" boxes. "Iter" is the number of iterations so far, "Correlation Coef" is the correlation coefficient of the power law graph, "Slope" is the slope of the line of best fit of the power law graph, "Edges" is the number of edges in the graph, and "Nodes" is the number of nodes in the graph.

Operating the Generator in Generate Mode

This mode is used to create a large network graph by allowing the generator to iterate through the generation algorithm until a specific number of iterations is reached. For example, set "Iterations" to 1000, "Initial Nodes" to 5, "Edges Added" to 3 with Constant checked, "Probability Distribution" to Greedy. Now click the RESET button to clear the generator. Then click GENERATE. Click DEGREE to see the resulting power law graph.

Operating the Generator in Repeat Loop Mode

This mode is used to create a set of graphs of the same type in order to do statistical analysis like computing the average correlation coefficient of the set. The initial parameters are set and then the generator creates N of these graphs where N is the number of graphs that are desired. For example, set "Iterations" to 1000, "Initial Nodes" to 5, "Edges Added" to 3 with Constant checked, "Probability Distribution" to Greedy. Now click the RESET button to clear the generator. Next to the LOOP button is a box where you enter the number of graphs you wish to create. Set this to 5. Make up a name for the "Loop Name" box, this name will be appended to the output files. For now just call it UCN100. IMPORTANT: uncheck "Show Graph" because the generator runs very slow if it is required to draw the graph on each iteration. Now click LOOP, and the graphs will be generated. To see the results you must look at the output files that are created. The most important file will be "masterUCN100.txt" which will have the statistics for the generated graphs. IMPORTANT NOTE: as of this writing this mode is not working properly when run under Windows, there is a bug that needs to be fixed, however, under Linux there seems to be no problem

Recommendations

Generating a Barabasi Graph: The probability distribution must be set to "Greedy", and "Constant" must be checked for Edges Added on each Iteration. Typical values for this model are as follows: for initial number of nodes set 5 with 3 edges added on each iteration, with 2000 to 3000 iterations set. This requires about 10 seconds on a 1.2 GHz Amd processor running Windows 2000. Typical correlation coefficients are 0.97 for this setup.

Increment Mode is mainly to view the graph generation process and not used to create graphs. It also exposes some bugs still present in this generator which still need to be fixed. For example, if you set the initial number of nodes less than the number of edges added the generator will hang.

Generate Mode is used to create a representative graph of a particular type. If a graph of 3000 nodes is needed using the Barabasi model, then this mode should be used to create such a graph.

Repeat Loop Mode is used for statistical analysis. A large sample of graphs of the same type are created so that the average slope and correlation coefficient of this set can be analyzed.

License

The license for use and distribution of this graph generator is very simple. Anyone is free to download and use this generator at no cost and at his or her own risk. There is no warranty or guarantee attached to this software program. Anyone is free to alter the code and enhance or customize this program to their own design as long as the changes are well documented and proper credit is given to the author (Derek Randolph Dreier) if the modified program is further distributed. The original program should not be distributed except through download from http://www.cs.ucr.edu/~ddreier or by sending a request to Derek Randolph Dreier at ddreier@cs.ucr.edu or derek@ivic.net .

Index