BERGISCHE
UNIVERSITÄT
WUPPERTAL

Fakultät für Mathematik
und Naturwissenschaften

Lehrstuhl für Scientific Computing & High Performance Computing

Graph(X)

Documentation

by	Simon Gerten
student number	1658115
course of studies	Applied Science, Mathematics & Computer Science
date	September 19, 2023

Contents

1 Introduction

This is the documentation for the software Graph(X). The software is a tool for analyzing and visualizing the centrality of edges in a network. It offers various functions and options for determining and plotting edge centrality. Additionally, methods are available to compare different centrality measures in various ways.
The documentation describes the graphical user interface and all functions and settings of the software in detail. Overall, the documentation is intended to facilitate the use of the software and provide assistance in case of any occurring problems.

2 Graphical User Interface (GUI)

This section describes the graphical user interface (GUI) of Graph(X). The GUI mainly consists of two windows: the main window and the option window. There is a light and a dark color theme, which depends on the setting of the operating system. On Linux only the light color theme is available.

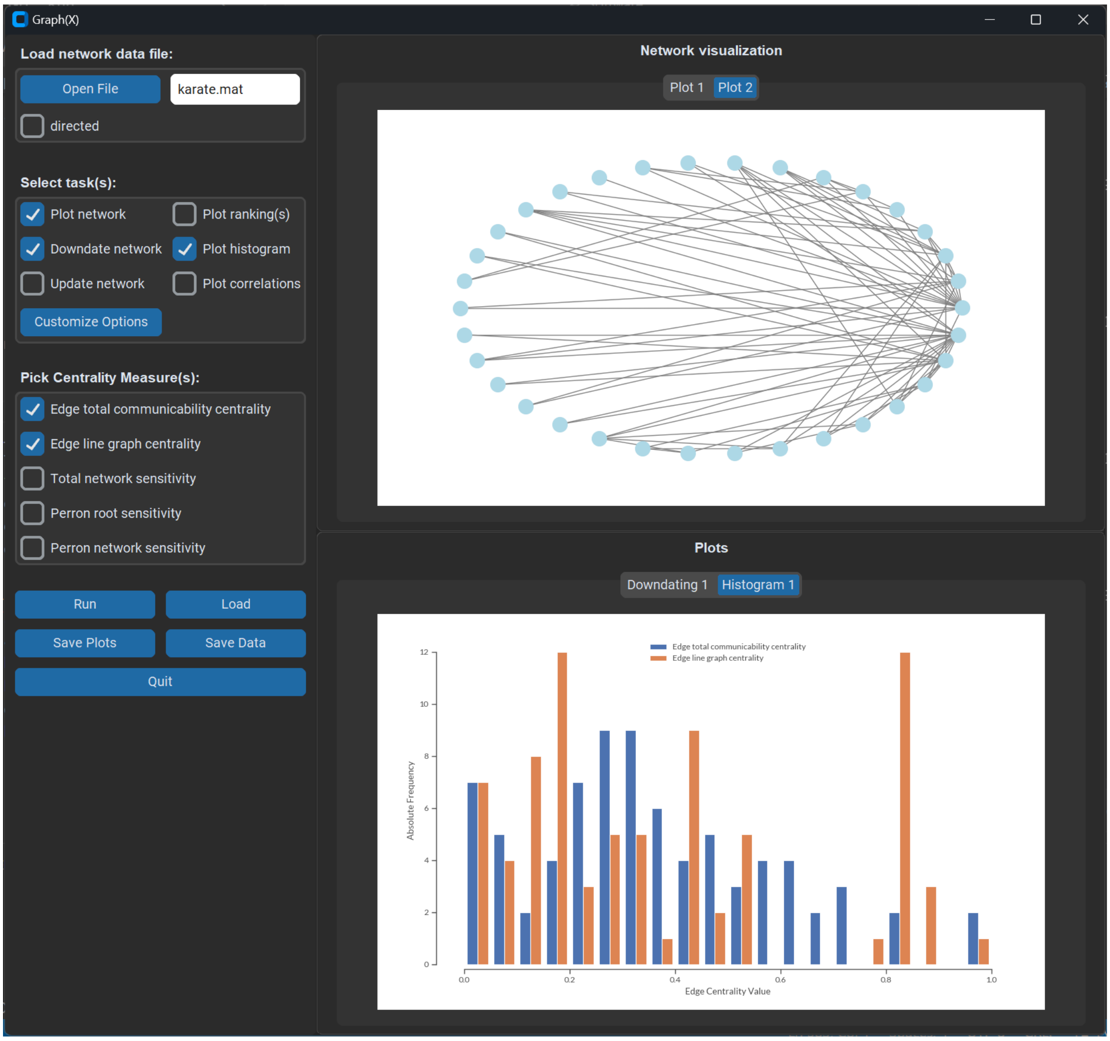
2.1 The main window

The main window of Graph(X) is divided into three sub-windows (see Figure 2.1):

1. On the left we have a panel with various buttons and checkboxes. Broadly speaking, these buttons and checkboxes control what to analyze and how to plot the results. They are arranged in four functional groups. The first group at the top deals with the loading of network data. In the second group underneath you can select the tasks of analyzation and comparison and specify how to plot the results using the customization options. The third group displays the edge centrality measures from which to choose. Lastly, we have the main execution buttons at the bottom of the panel. For a detailed description of the functionality for the buttons and checkboxes refer to section 3.

2. The frame **Network visualization** displays the plots of networks as a result to the *Plot Network* task. The different plots can be toggled through using the tabs.

3. The plots of all other tasks are being displayed in the frame **Plots**. Again, it is possible to toggle through the plots using the tabs.


Graph(X)

Load network data file:

Open File karate.mat

□ directed

Select task(s):

☑ Plot network
□ Plot ranking(s)
☑ Downdate network
☑ Plot histogram
□ Update network
□ Plot correlations

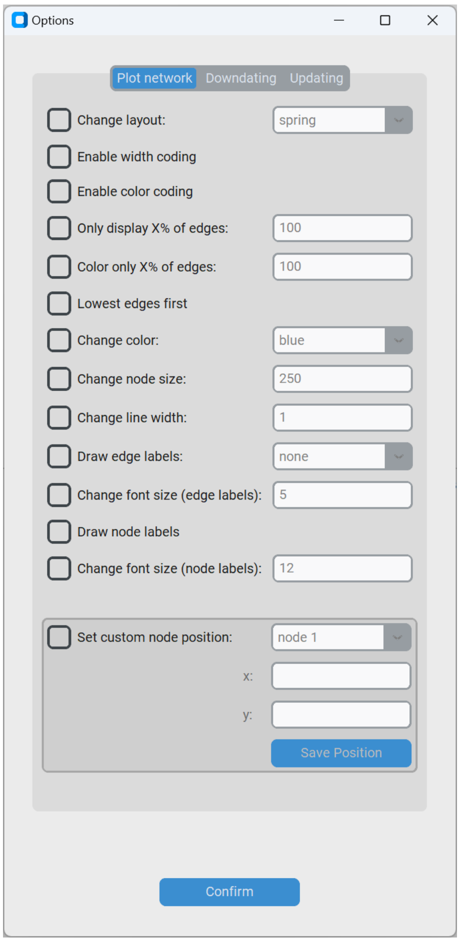
Pick Centrality Measure(s):

☑ Edge total communicability centrality
☑ Edge line graph centrality
□ Total network sensitivity
□ Perron root sensitivity
□ Perron network sensitivity

Figure 2.1 Main window of Graph(X) (dark color theme)

2.2 The option window

The option window (see Figure 2.2) can be opened by pressing the *Customize Options* button in the left panel. For now, customization options only exists for the *Plot network*, the *Downdating* and the *Updating* task. It is possible to toggle through the options for the different tasks using the tabs. By default, all options are inactive (grayed out). They can be activated by pressing the corresponding checkboxes which also allows the user to access the combo or input boxes in the corresponding cases. The combo and input boxes are set to default values. The value of a box is active even if the corresponding checkbox is not checked. At the bottom of the *Plot network* tab there is an option to set custom node positions. Every node of the network can be assigned custom x- and y-values which determines the position of the node when plotting the network. For a detailed description of all options refer to section 3.2.



□ Change layout: spring

□ Enable width coding

□ Enable color coding

□ Only display X% of edges: 100

□ Color only X% of edges: 100

□ Lowest edges first

□ Change color: blue

□ Change node size: 250

□ Change line width: 1

□ Draw edge labels: none

□ Change font size (edge labels): 5

□ Draw node labels

□ Change font size (node labels): 12

□ Set custom node position: node 1

x:

y:

Figure 2.2 Option window of Graph(X) (light color theme)

3 Functions and Settings

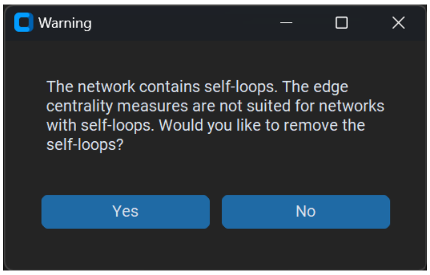
This section contains descriptions of the functions and settings of Graph(X).

3.1 Network data

Needless to say, a data file of the network is required for analyzing or visualizing the network. The data file can be loaded via the *Open File* button. By pressing the button a new window pops up, in which you can select the directory of the data file on your hard drive. Supported file types are ".mat" and ".npz" files. The data file must contain the adjacency matrix of the network in a matrix or sparse matrix format. The name of the key for the adjacency matrix in the Matlab file must be set to "A". If the data file is successfully loaded, the name of the file is displayed next to the *Open File* button.
During the loading process, Graph(X) examines various properties of the network:

- **directed or undirected:** If the adjacency matrix of the network is symmetric, Graph(X) assumes that the network is undirected. If, however, the network should be treated as a directed network nonetheless, the checkbox underneath the *Open File* button needs to be checked manually. In the case of an asymmetric adjacency matrix, the network has to be directed and the checkbox is checked automatically.

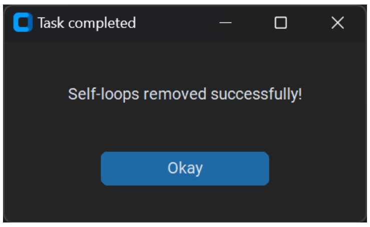
- **self-loops:** If the network contains self-loops, the following window will pop up asking to remove the self-loops:



Warning

The network contains self-loops. The edge centrality measures are not suited for networks with self-loops. Would you like to remove the self-loops?

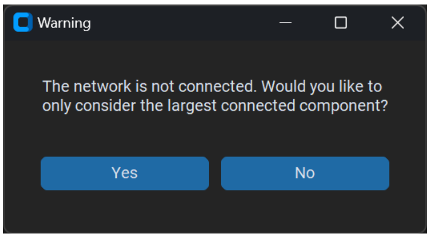
By affirming, Graph(X) will delete all self-loops of the network. A successful completion of this task will be confirmed via the following pop-up window:



Task completed

Self-loops removed successfully!

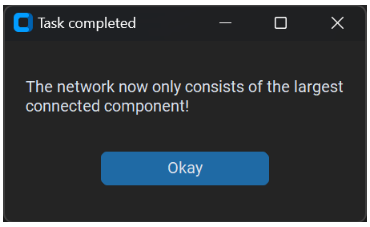
- **connectivity:** If the network is not (strongly) connected, the following window will pop up asking to only consider the largest connected component:



Warning

The network is not connected. Would you like to only consider the largest connected component?

By affirming, Graph(X) will only consider the largest connected component of the network, which will be confirmed via the following pop-up window:



Task completed

The network now only consists of the largest connected component!

- **weights:** If the graph is weighted, the following line will be printed to the background console: "Graph is weighted!".

3.2 Tasks

Currently, there are six different tasks available, which can be toggled on or off by selecting or deselecting the corresponding checkboxes in the left panel of the main window. It is possible to run multiple tasks simultaneously.

3.2.1 Plot network

The *Plot network* task will plot the graph of the current network and display the plot in the **Network visualization** frame. There are multiple options for customization:

- **Change layout:** The layout determines the shape of the network in which the nodes are positioned. The following layouts are available: spring, circular, spiral, spectral, shell, kamada kawai and random. It is set to "spring" by default. Refer to the documentation of the Python package NetworkX for a description of these layouts: https://networkx.org/documentation/stable/reference/drawing.html.

- **Enable width coding:** If enabled, the width of an edge depends on its centrality value. The higher the value the thicker the edge is drawn and vice versa. The centrality measure for calculating the centrality values can be selected in the combo box to the right (not currently implemented: For now, the first selected centrality measure provides the centrality values).

- **Enable color coding:** If enabled, the color of an edge depends on its centrality value. The higher the value the darker the color of the edge and vice versa. The centrality measure for calculating the centrality values can be selected in the combo box to the right (not currently implemented: For now, the first selected centrality measure provides the centrality values).

- **Only display X% of edges:** It can be specified how many edges of the network should be drawn depending on the edge centrality values. The input field takes a floating point number between 0 and 100. This number determines the X% of edges with highest centrality values. For example, a value of 20 will only draw the 20% of edges with highest centrality values. It is set to 100% by default.

- **Color only X% of edges:** It can be specified how many edges of the network should be drawn in color depending on the edge centrality values. The input field takes a floating point number between 0 and 100. This number determines the X% of edges with highest centrality values. For example, a value of 20 will only draw the 20% of edges in color with highest centrality values. It is set to 100% by default.

- **Lowest edges first:** This option only has an effect if one of the two options above is active. If enabled, the X% of edges with lowest centrality values are drawn (in color). It is disabled by default.

- **Change color:** Changes the color in which the edges of the network are drawn. Three colors are available: blue, red and gray. It is set to "blue" by default.

- **Change node size:** Changes the size of the nodes. It is set to 250 by default.

- **Change line width:** Changes the width of the edges. It is set to 1 by default.

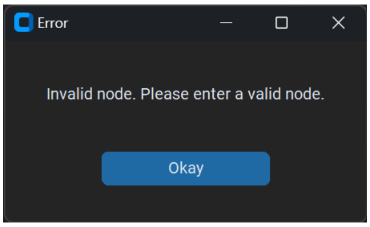
- **Draw edge labels:** Determines whether to draw edge labels. There are two different kinds of edge labels available. One displays the coordinates and the other one displays the number of the edge. It is set to "none" by default.

- **Change font size (edge labels):** Determines the font size of the edge labels. It is set to 5 by default.

- **Draw node labels:** Determines whether to draw node labels. If enabled, it displays the number of the node starting from 1. It is disabled by default.

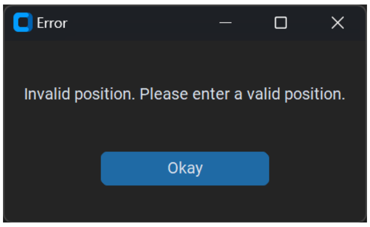
- **Change font size (node labels):** Determines the font size of the node labels. It is set to 12 by default.

- **Set custom node position:** It is possible to set a custom position for every node in the network. The node of which the position should get specified, has to be selected in the combo box. The combo box is editable, meaning it is possible to just input the specific node in the form "node + number of the node". Note that the numbering of the nodes starts at 1. If the form is incorrect or the node does not exists in the network the following error window will pop up, when trying to save the position:

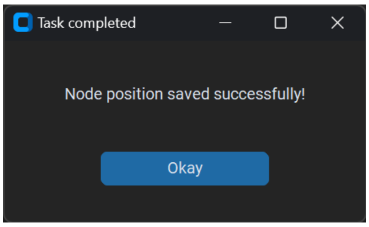
  Invalid node. Please enter a valid node.

  The x- and y-coordinate of the node position has to be specified in the input fields underneath the combo box. The node is positioned in a standard coordinate system, meaning that, e.g., a position of (1.5, 1.5) will position the node in the upper right corner and a position of (-1.5, -1.5) will position the node in the lower left corner. If the entered position is invalid, the following error window will pop up, when trying to save the position:


Error

Invalid position. Please enter a valid position.

The *Save Position* button saves the position of the node if the node and the position are valid. It is confirmed by the following pop-up window:


Task completed

Node position saved successfully!

3.2.2 Downdate network

The downdating of a network is an analytic process which analyzes the decline of the total network communicability when edges of the network are being removed successively. The order, in which to remove the edges, depends on the centrality values of the edges determined by an edge centrality measure. By this means, it is possible to analyze the performance of an edge centrality measure as the calculated centrality values should directly correlate to the total network communicability in a way that edges with high centrality contribute significantly to the overall communicability and edges with low centrality have a minimal impact on the communicability. It is possible to select multiple edge centrality measures and run the downdating process for each measure simultaneously. The results of the downdating process(es) will be displayed in the **Plots** frame. There are a few options available to customize the process:

- **Change number of iterations:** An iteration represents a step of the process in which an edge is being removed and the total network communicability of the new downdated graph is calculated. The number of iterations, i.e., the number of removed edges can be set in the input field. Note that for undirected networks the number of iterations will be cut in half, meaning that the number of iterations should be 20 if 10 undirected edges should be removed. If the number is set to 0, all edges will be removed. By default the number is set to 0.
- **Enable greedy mode:** The greedy mode considers the recalculation of the centrality measure(s) in every iteration, so to give the measure(s) the possibility to take the

removal of an edge into account. It is disabled by default.
→ computing intensive

- **Change order:** The order in which to downdate can be specified. If "lowest" is selected, the process will remove the edges in ascending order starting with the lowest centrality value. If "highest" is selected, the process will remove the edges in descending order starting with the highest centrality value. It is set to "lowest" by default.

The individual iteration steps, which specify the removed edges and the values for the total network communicability for all centrality measures, will be printed to the background console.

3.2.3 Update network

The updating of a network is an analytic process which analyzes the increase of the total network communicability when edges are being added successively to the network. Which edge to add depends on its centrality value determined by an edge centrality measure which calculates the centrality of edges not present in the network (not working for "Edge line graph centrality"). By this means, it is possible to analyze the performance of those edge centrality measures as the calculated centrality values should directly correlate to the total network communicability in a way that edges with high centrality contribute significantly to the overall communicability and edges with low centrality have a minimal impact on the communicability. It is possible to select multiple edge centrality measures and run the updating process for each measure simultaneously. The results of the updating process(es) will be displayed in the **Plots** frame. There are a few options available to customize the process:

- **Change number of iterations:** An iteration represents a step of the process in which an edge is being added and the total network communicability of the new updated graph is calculated. The number of iterations, i.e., the number of added edges can be set in the input field. Note that for undirected networks the number of iterations will be cut in half, meaning that the number of iterations should be 20, if 10 undirected edges should be added. If the number is set to 0, all edges not present in the network will be added. By default the number is set to 10.

- **Enable greedy mode:** The greedy mode considers the recalculation of the centrality measure(s) in every iteration, so to give the measure(s) the possibility to take the addition of an edge into account. It is disabled by default.
  → computing intensive

- **Change order:** The order in which to update can be specified. If "lowest" is selected, the process will add the edges in ascending order starting with the lowest centrality value. If "highest" is selected, the process will add the edges in descending order starting with the highest centrality value. It is set to "highest" by default.

The individual iteration steps, which specify the added edges and the values for the total network communicability for all centrality measures, will be printed to the background console.

3.2.4 Plot ranking(s)

This tasks will plot the ranking(s) of the selected centrality measure(s). A ranking is a list of the edges of a network with corresponding centrality values for all edges. The plot will be displayed in the **Plots** frame. The centrality values will be scaled to the interval from 0 to 1 for reasons of comparability.

3.2.5 Plot histogram

This task will plot the distribution of the centrality values of the selected centrality measure(s). The plot will be displayed in the **Plots** frame. The centrality values will be scaled to the interval from 0 to 1 for reasons of comparability.

3.2.6 Plot correlations (not yet implemented)

This task will plot the correlation of the rankings of two selected centrality measures. If more than two centrality measures are selected, it will plot the correlations of all possible pairs of distinct centrality measures. The plot(s) will be displayed in the **Plots** frame. The plot(s) also include the specification of the Kendall rank correlation coefficient τ.

3.3 Centrality Measures

There are five different edge centrality measures to choose from. For all tasks except the *Plot network* task it is required to select at least one centrality measure.

3.4 Main execution buttons

There are five different main execution buttons:

- **Run:** This button executes all selected tasks.
- **Load:** This button loads the "data.json" file in the current working directory and runs the tasks specified in the data. The data file should consist of the adjacency matrix, the name of the network file, the edge rankings including the centrality values of all previously selected centrality measures, all previously selected tasks and options as well as the specified node positions.
- **Save Plots:** This button will save all currently displayed plots of the **Network visualization** frame and the **Plots** frame to the "plots" folder in the current working directory. The filename of a plot matches the name of the corresponding tab in the corresponding frame.

- **Save Data:** Saves the following data to a "data.json" file in the current working directory: the adjacency matrix, the name of the network file, the edge rankings calculated at last including the centrality values of all selected centrality measures, all selected tasks and options as well as the specified node positions. The data can be loaded in a later instance of Graph(X) using the *Load* button.

- **Quit:** This button closes the current instance of Graph(X). All unsaved plots and data will be lost.