GUI Interface to Read Incidence Matrix and Adjacency Matrix

Ayonya Prabhakaran
Dept. of Information Technology,
Indian Institute of Information Technology, Allahabad

Ajay Saini
Dept. of Information Technology,
Indian Institute of Information Technology, Allahabad

Ankit Mund
Dept. of Information Technology,
Indian Institute of Information Technology, Allahabad

***Abstract*—The paper details the process of creation of the graphical user interface(GUI) in Java to take read an incidence and adjacency matrix from the user after verification. The input is then checked to check if it represents a valid graph.**

***Index Terms*—graphs, incidence matrix, adjacency matrix, graphical interface**

I. Introduction

A graph is an ordered pair G = (V, E) comprising a set V of vertices, nodes or points together with a set E of edges, arcs or lines, which are 2-element subsets of V. A graph is made up of vertices, nodes, or points which are connected by edges, arcs, or lines. A graph may be undirected, meaning that there is no distinction between the two vertices associated with each edge, or its edges may be directed from one vertex to another. The graph can be represent using an incidence matrix or an adjacency matrix. This paper will be breif on building a GUI to read the input matrices from the user and verify if the matrices represent a valid graph.

II. Methods/ Algorithm

A. User Verification

The input entered is verified by the user so as to make sure there are no wrong edges added to the graph. This is done by means of a pop-up box that asks the user to verify if he/she approves of the particular edge to be added to the network of vertices. In case the user wishes to delete a particular edges present, he/she can do that also.

B. Incidence Matrix

An incidence matrix is a matrix that shows the relationship between two classes of objects. If the first class is X and the second is Y, the matrix has one row for each element of X and one column for each element of Y. The entry in row x and column y is 1 if x and y are related and 0 if they are not. The incidence matrix of an undirected graph is a n x m matrix B, where n and m are the numbers of vertices and edges respectively, such that $B_{i,j}$ = 1 if the vertex v_i and edge e_j are incident and 0 otherwise, as shown in Fig. 1. To check if

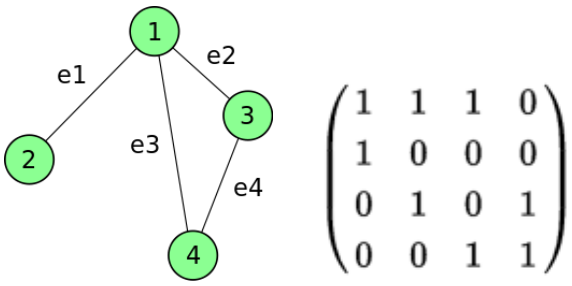


Fig. 1. An undirected graph containing 4 vertices and 4 edges and its corresponding incidence matrix.

the graph is valid or not the sum of each column is calculated as shown by the conditions in (1).

$$\{sum_j = 2 \ \forall \ j \ \in \ m\} \Rightarrow Valid\ Graph, \\ \{sum_j \neq 2 \ \forall \ j \ \in \ m\} \Rightarrow Invalid\ Graph \tag{1}$$

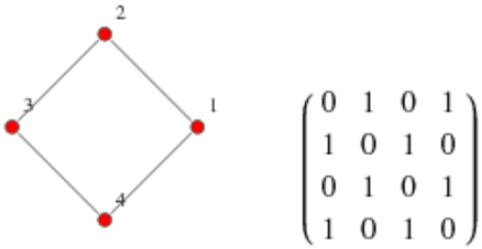
The incidence matrix of a directed graph is a n x m matrix B where n and m are the number of vertices and edges respectively, such that $B_{i,j}$ = 1 if the edge e_j leaves vertex v_i, 1 if it enters vertex v_i and 0 otherwise. In this case the conditions are given by (2) that give the validity of the graph.

$$\{sum_j = 0 \ \forall \ j \ \in \ m\} \Rightarrow Valid\ Graph, \\ \{sum_j \neq 0 \ \forall \ j \ \in \ m\} \Rightarrow Invalid\ Graph \tag{2}$$

C. Adjacency Matrix

An adjacency matrix is a square matrix used to represent a finite graph. The elements of the matrix indicate whether pairs of vertices are adjacent or not in the graph. If the graph is undirected, the adjacency matrix is symmetric. For a simple graph with vertex set V, the adjacency matrix is a square V x V matrix $\boldsymbol{A}$ such that its element $\boldsymbol{A}_{ij}$ is 1 when there is an edge from vertex i to vertex j, and 0 when there is no edge, as shown in Fig. 2. The diagonal elements of the matrix are all 0, since edges from a vertex to itself (loops) are not allowed. Thus, in the case of a adjacency matrix the sum of the diagonal elements of must be zero as represented by (3).

$$\{sum_{i,i} = 0 \ \forall \ i \ \in \ V\} \tag{3}$$



$$\begin{pmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \end{pmatrix}$$

Fig. 2. An undirected graph containing 4 vertices and 4 edges and its corresponding adjacency matrix.

Fig. 3. The welcome page of the GUI.

III. Implementation Details

The implementation was done in Java. The GUI was created with the help of the ***Java Swing Components*** which include the *JFrames, JButtons*, *JLabels* and *JTextFields*.

A. Welcome Page

The first page of the GUI is where the user can select to input an adjacency matrix or an incidence matrix and an exit button to exit the graphical interface, as shown in Fig. 3.

B. GUI for the Adjacency Matrix

This page contains the the buttons as listed below with corresponding functions and text boxes to enter the respective input:

- "Create Graph" button reads the number in entered by the user in the "Number of Vertices" text box and initializes a matrix of the corresponding size.
- Text boxes "Vertex_1" and "Vertex_2" take the input for an undirected edge to be added between the specified vertices.
- "Add Edge" button adds an edge between the specified vertices mentioned in the above text boxes after verifying the same with a pop-up window, as shown in Fig. 4. The user may choose "Yes" or "No" accordingly. Warnings for addition of self-loops and already added edges (repetition of edges) are displayed.
- "Delete Edge" button deletes an edge between the specified vertices mentioned in the above text boxes after verifying the same with a pop-up window in which user may choose "Yes" or "No" accordingly. Warnings for deletion of self-loops and empty edges (no edge between these pair of vertices) are displayed.
- "Display Adjacency Matrix" displays the matrix created, as shown in Fig. 5.
- "Go Back" button takes the user to the *Welcome Page* of the GUI.

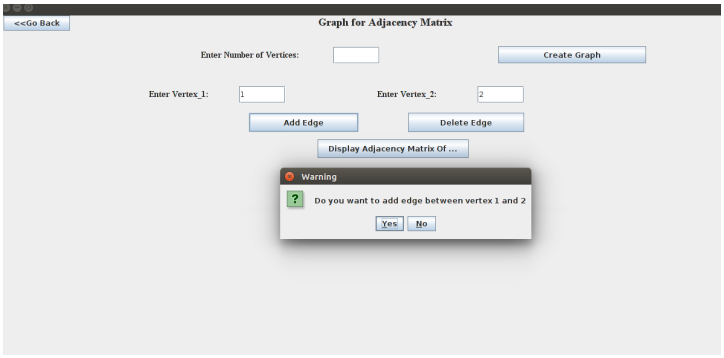


Fig. 4. The adjacency matrix page of the GUI with the pop-up box for user verification.

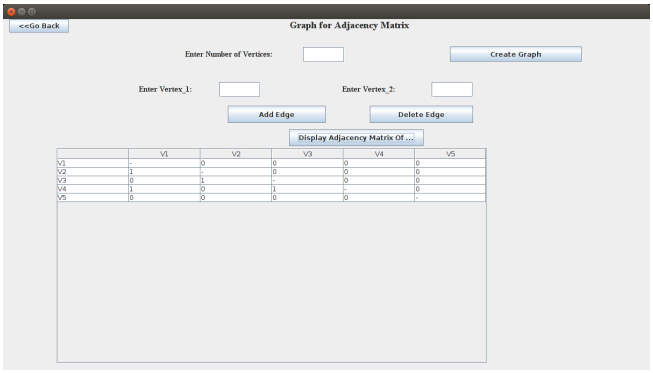


Fig. 5. The output after the "Display Adjacency Matrix" button in pressed.

C. GUI for the Incidence Matrix

This page contains the the buttons as listed below with corresponding functions and text boxes to enter the respective input:

- "Create Graph" button reads the number in entered by the user in the "Number of Vertices" text box and "Number of Edges" text box and initializes a matrix of the corresponding size.
- Text boxes "Vertex No" and "Edge No" take the input for an undirected edge to be added between the specified indices.
- "Add Entry" button makes the edge incident on the specified vertex mentioned in the above text boxes after verifying the same with a pop-up window, as shown in Fig. 6. The user may choose "Yes" or "No" accordingly. A warning is displayed in case the user makes an edge incident on more than two vertices which is found by the

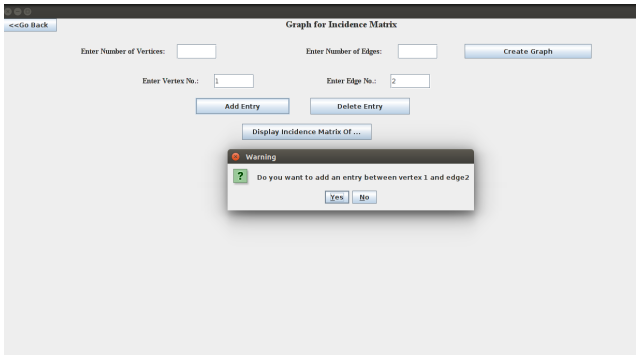


Fig. 6. The incidence matrix page of the GUI with the pop-up box for user verification.

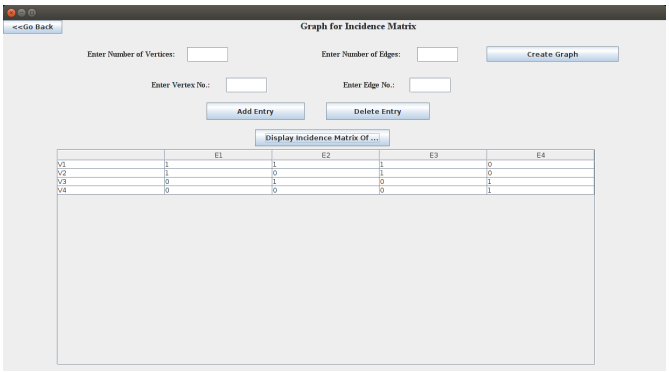


Fig. 7. The output after the "Display Incidence Matrix" button in pressed.

checking the sum of the columns as mentioned in Sect. II A.

- "Delete Entry" button deletes an edge from being incident on the specified vertex mentioned in the above text boxes after verifying the same with a pop-up window in which user may choose "Yes" or "No" accordingly. A warning for deletion of empty incidence (no such edge is incident on that vertex) is displayed.
- "Display Incidence Matrix" displays the matrix created, as shown in Fig. 7.
- "Go Back" button takes the user to the *Welcome Page* of the GUI.

IV. Conclusion

The above paper creates a user-friendly GUI which solves the problem of starting over while providing input if an erroneous input is provided due to the added functionality of verification of the edge before it is added and also the functionality of deletion of the same.

References

[1] Wikipedia (https://en.wikipedia.org/)
[2] Wolfram Mathworld (http://mathworld.wolfram.com/)
[3] Google Images (https://images.google.com/)
[4] Geeks For Geeks (http://www.geeksforgeeks.org/)