ARISTOTLE UNIVERSITY OF THESSALONIKI

Thesis Title

by

Iakovos Marios Tsouros

A thesis submitted in partial fulfillment for the
Graduate degree

in the
Faculty of Sciences
School of Physics

Supervising Professor: Dr. Panagiotis Argyrakis

Date

Declaration of Authorship

I, [author's name], declare that this thesis titled, [thesis title] and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a research degree at this University.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Signed: ______________________________

Date: ______________________________

Inspiring quote goes here (optional)

Quote's attribution

ARISTOTLE UNIVERSITY OF THESSALONIKI

Abstract

Faculty of Sciences
School of Physics

Graduate Degree

by Iakovos Marios Tsouros

Abstract goes here.

Acknowledgements

Acknowledgements go here.

Contents

List of Figures

List of Tables

Abbreviations

Acronym	**W**hat (it) **S**tands **F**or

Dedication (optional)

Chapter 1

Introduction

1.1 Graphs

In this subsection, the main aspects of graph theory are briefly presented.

1.1.1 Introduction

In the real world, many problems can be described by a diagram connecting a set of points with lines, joining pairs of these points, or even creating loops on a single point. A simple example of that would be a set of points representing people with lines connecting acquintances, or points representing atoms and lines representing chemical bonds, creating a representation of a molecule as a graph attribute. In the examples above, the only information contained is whether two points are associated, with the manner being disregarded. The concept of a graph consists of a mathematical abstraction of the above. [1]

Definition 1.1. Mathematically, in its simplest form, a **graph** is an ordered pair[1] $G = (V, E)$ of:

- **V**, a set of vertices (also known as nodes).
- $E \subseteq \{\{x, y\} | x, y \in V \; x \neq y\}$, which is the set of **edges** which consists of unordered pairs of vertices that connect two nodes.

This type of object is called an **undirected simple graph** to avoid confusion with other types.

[1]An ordered pair (a, b) is a pair of objects in which the order of appearance or insertion is significant; the ordered pair (a, b) is different than (b, a) unless $a = b$. An unordered pair is a set of the form a, b is a set having two elements with no relation between them and $a, b = b, a$.

Definition 1.2. A *graph* G is an ordered pair $(V(G), E(G))$ consisting of a set $V(G)$ of *vertices* (also called *nodes* or *points*) and a set $E(G)$, disjoint from $V(G)$ which consists of *edges* (also called *links* or *lines*) together with an incidence function ψ_G that associates with each edge of G an unordered pair of not necesserily distinct vertices of G. If e is an edge and u and v are vertices such that $\psi_G = u, v$ then e is said to *join* u and v, and the vertices u and v are called the *ends* of e. We denote the numbers of vertices and edges G by $u(G)$ and $e(G)$ which two parameters are called the *order* and *size* of G, respectively [1].

In short, we can define a **graph** as an ordered triple $G = (V, E, \phi_G)$ consisting of:

- V, a set of *vertices*
- E, a set of *edges*
- $\phi_G : E \to \{\{x, y\} | x, y \in V \ and \ x \neq y\}$ an *incidence function* mapping every edge to an unordered pair of vertices - an edge associated with two distinct vertices. The incidence function is a function of the edges.

This type of object is called an *undirected multigraph*, to avoid confusion. Note, that the above definition of the *incidence function* does not allow for *loops* (mappings of an edge on the same vertex).

A *loop* is a an edge that allows a connection of a vertex to itself and a graph can be defined to either allow or disallow the presence of loops. Some authors allow for loops to exist on *multigraphs* [2], while other consider these kind of graphs to exist in a different category, called *pseudographs* [3]. Allowing loops requires modifying the incidence function so they can be supported. The new incidence function can be written as:

$$\phi_G : E \to \{\{x, y\} | x, y \in V\} \tag{1.1}$$

The example presented below should better illustrate clarify the definition (of a pseudograph).

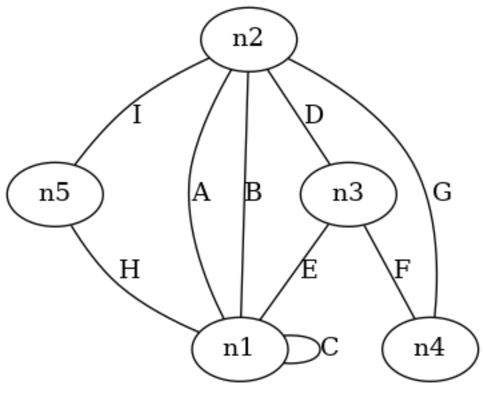


FIGURE 1.1: An undirected pseudograph with labeled nodes and edges.

Example 1.1.

For the graph presented in Figure 1.1 the following can be assumed:

$$G = (V(G), E(G))$$

and

$$V(G) = \{n_1, n_2, n_3, n_4, n_5\}$$
$$E(G) = \{A, B, C, D, E, F, G, H, I\}$$

and the incidence function is defined as:

$$\psi_G(A) = n_1n_2, \quad \psi_G(B) = n_1n_2, \quad \psi_G(C) = n_1n_1, \quad \psi_G(D) = n_2n_3,$$
$$\psi_G(E) = n_1n_3, \quad \psi_G(F) = n_3n_4, \quad \psi_G(G) = n_2n_4, \quad \psi_G(H) = n_1n_5,$$
$$\psi_G(I) = n_2n_5$$

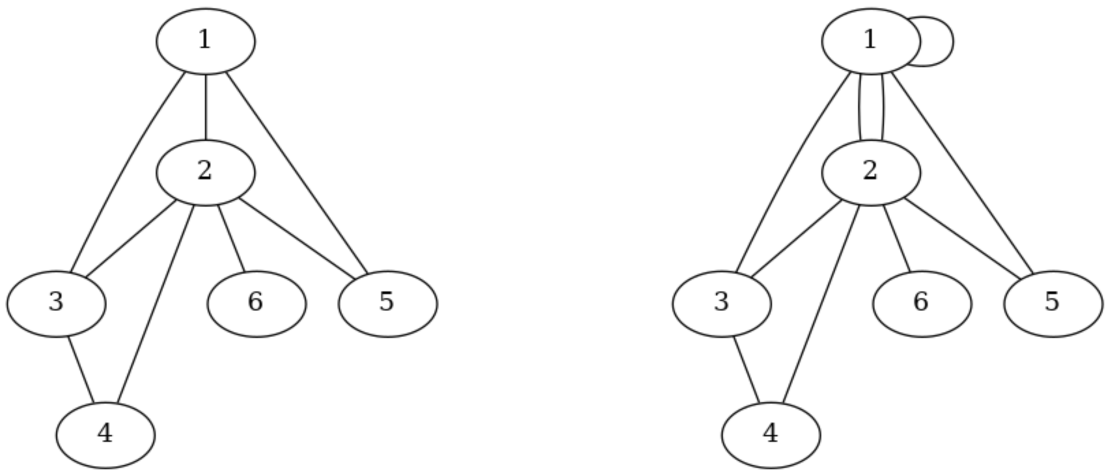
It should now be clear that with the newer definition of ϕ_G, self loops are now possible. Additionaly, even though this was not prohibited by the previous definition, it is worth noting that a node can be connected to another with multiple edges (or multiedges), or that it can have zero connections to other nodes. Generally, V is assumed to be a non-empty set, but E can be empty.

It is now possible to define some characteristic attributes of graphs:

- $|V|$: the **order** of a graph is the number of its vertices.
- $|E|$: the **size** of a graph is the number of its edges.
- The **degree** (or **valency**) of a single node is the number of edges connected to it. The **degree** of a graph is the maximum number of edges connected to a single vertex that belongs to it.
- The edges of create a *homogenous relation*[2] $\sim$ on the vertices of the graph that is called **adjacency relation**; for each edge (x, y), its endpoints *x, y* are said to be **adjacent** to each other, denoted by $x \sim y$. This property will be particulary useful when the adjacency matrix is defined in the following section.

It can be inferred from the above definitions and attributes that for an undirected graph of order *n*, the maximum *degree* of a node is $n - 1$ and and maximum *size* of a graph is $n(n - 1)/2$.

[2] A **homogenous relation** (or **endorelation**) over a set X is a set of assignments (binary relations) over X and itself; i.e. it is a subset of the cartesian product $X \times X$



(A) Multigraph with no loops and multiple edges.

(B) Mutligraph with loops and multiple edges.

FIGURE 1.2: Two undirected multigraphs.

In this section only *undirected* graphs were considered, which are graphs with edges with no orientation. A whole other class of graph objects with edges which have orientation exists, called *directed graphs*. These kind of graphs objects are out of scope for this thesis and will not be presented.

1.1.2 Adjacency Matrix

Definition 1.3. The *adjacency matrix* is the fundamental mathematical representation of a graph. It is a square matrix, the elements of which represent which pair of nodes are *adjacent* or not. Thus, the adjacency matrix $\mathbf{A}$ of a graph of order n is the $n \times n$ matrix with elements A_{ij} such that:

$$A_{ij} = \begin{cases} 1 & \text{if there exists at least one edge connecting } i \text{ and } j \\ 0 & \text{if there no edges connecting those edges directly.} \end{cases} \tag{1.2}$$

Considering the simple undirected graph of Figure 1.3a we can construct the following adjacency matrix:

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

TABLE 1.1: Adjacency matrix for Figure 1.3a

For this simple network, which has no loops and only one edge connect two nodes, the diagonal matrix elements are always zero and the matrix is symmetric, as for each edge connecting i and j there is a representation for the j to i connection as well.

In a more complex case, such as the one presented in Figure 1.2b where loops and multiedges are present an adjacency matrix can still be constructed. In this case, a multiedge is represented by setting the value of the corresponding A_{ij} value equal to the multiplicity of the edge. In this case, $A_{12} = A_{21} = 2$.

For loops, the most common representation in the case of undirected graphs is to still set the value of the A_{ii} element equal to 2 (i.e. $A_{11} = 2$ in the example presented). Essentially, an edge of a loop has two ends that connect to the same node, thus the result [4, p. 68]. Additionaly, defining the matrix in such manner, allows for better computations and is consistent with the definition of the representation of an edge connecting two nodes of an undirected graph [5, p. 108].

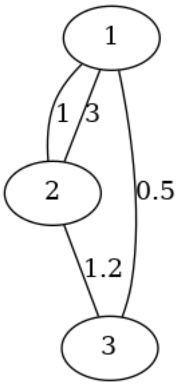
Thus, the adjacency matrix for the graph of Figure 1.2b is

$$A = \begin{pmatrix} 2 & 2 & 1 & 0 & 1 & 0 \\ 2 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \end{pmatrix}$$

TABLE 1.2: Adjacency matrix for Figure 1.2b

Remark 1.4. Note that the degree of a node can be easily found by summing the values of the column or row of the adjacency matrix that correspond to said node.

So far, while presenting edges, we have considered graphs where connections between nodes represented binary relations between them; they either existed or they did not. In many situations when studying graphs, it is useful to represent edges as connections which carry some kind of attribute or value, commonly called *weight*. This weight could be any real number that fits a particular example, such as the distance between two airfields on an airline network, the kinship of connections on a social network (negative values can represent animosity and vice versa) or any type of relational attribute that can be quantified and characterizes the connection between nodes that belong in the same network[5, p. 109]. A simple example is presented in the figure below.



(A) Multigraph with no loops and multiple edges.

$$A = \begin{pmatrix} 0 & 4 & 0.5 \\ 4 & 0 & 1.2 \\ 0.5 & 1.2 & 0 \end{pmatrix}$$

(B) Corresponding adjacency matrix.

FIGURE 1.3: Simple example of an unordered graph with weighted edges

Generally, edges and nodes can hold any type of variable as values, such as vectors, the usefulness of which will become apparent when computer algorithms for graph representations and graph neural networks are discussed in later sections and chapters.

Remark 1.5. An alternative to the adjacency matrix is the *adjacency list*. An adjacency list is a collection of lists, one for each node i. Each list contains the labels of the nodes that i is connected to, and its the most common method for storing networks on computers as it requires less space. It is also possible to represent edge attributes in an adjacency list by appending an extra column which holds these values. An example for the graph presented in Figure 1.2b with multiedges and loops:

Node	Neighbors
1	1,2,2,5,3
2	1,1,3,5,6,4
3	1,2,4
4	3,2
5	1,2
6	2

TABLE 1.3: Adjacency list for Figure 1.2b

Each edge of the network appears twice, thus for a network with m edges the size of the adjacency list would be $2m$, much smaller compared to the $n \times n$ matrix required to build an adjacency matrix. This is particularly useful in networks which are relatively *sparse*[3], but have a high order.

[3] *Sparse* networks are networks with a much lower number of edges than those possible.

Chapter 2

Literature Review

Chapter 3

Notation & Fundamentals

Chapter 4

Basic Principles and Implementation Framework for an [Problem to be Solved]

Chapter 5

Implementation and Core Components of [Platform Title]

Chapter 6

Experimentation & Validation

Chapter 7

Conclusions & Future Work

Bibliography

[1] U.S.R. Murty Adrian Bondy. *Graph theory*. Graduate texts in mathematics 244. Springer, 3rd corrected printing. edition, 2008. ISBN 1846289696; 9781846289699.

[2] Bela Bollobas and Endre Szemerédi. Girth of sparse graphs. *Journal of Graph Theory*, 39: 194 – 200, 01 2002. doi: 10.1002/jgt.10023.

[3] Ping Zhang Gary Chartrand. *A First Course in Graph Theory*. Dover Books on Mathematics. Dover Publications, 2012. ISBN 0486483681; 9780486483689.

[4] A. Ashikhmin and A. Barg. *Algebraic Coding Theory and Information Theory: DIMACS Workshop, Algebraic Coding Theory and Information Theory, December 15-18, 2003, Rutgers University, Piscataway, New Jersey*. DIMACS series in discrete mathematics and theoretical computer science. American Mathematical Soc. ISBN 9780821871102. URL `https://books.google.gr/books?id=wp7XsCAm9EC`.

[5] Mark Newman. *Networks*. Oxford University Press, 2 edition, 2018. ISBN 0198805098; 9780198805090.