Applications of Graph Theory in Daily Life

Nicholas Wijaya 13516121[1]
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
[1] *nic.nicholas1907@gmail.com*

***Abstract*—This paper aims to emphasize the applications of graph theory in several aspects in human's daily life (technologies, chemistry, network, daily jobs). This paper gives some examples in those several aspects, and also overviews and how the graph theory is applied in those applications.**

***Keywords*—graph, graph theory, Hamiltonian Graph, Eulerian Graph**

I. Introduction

A graph is a planned drawing, consisting of nodes and lines that shows the relation of the nodes. Graphing is used in every aspect of human's life. From creating a mind map to fulfill a student's duty, to creating an artificial intelligence. Using colors and a little imagination, a person is capable of making a professional-looking graph.

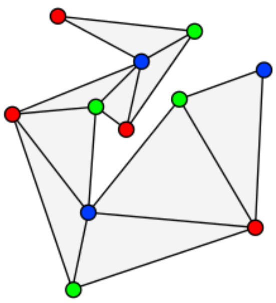

Figure 1.1 Colored Graph
source: https://upload.wikimedia.org/wikipedia/commons/thumb/c/c2/Triangulation_3-coloring.svg/228px-Triangulation_3-coloring.svg.png

Graph is a very powerful tool. It can be used in everyday life. Even when people do not realize, in fact they are using graph and they have gained a benefit of using it. For example, when a student is doing mind-mapping to help them with their memorizing, he/she is creating a graph. When people use a navigation application such as Google Maps or Waze, they are taking advantage of using a graph to find them the fastest routes to their destination.

II. Theory

A. History

Graph theory is a branch of discrete mathematics. It is a study of graphs which models relations between objects. Graph is used to represents discrete objects and relations between those objects [1]. Graph theory was originated by the Swiss mathematician Leonhard Euler who studied the "Königsberg bridge problem". It is a puzzle concerning the possibility of finding a path to cross all the bridges exactly once. Euler finally published a paper named "Seven Bridges of Königsberg" in 1736 [4]. It is considered as the first paper in the history of graph theory, and his graph later will be known as Eulerian Graph [3].

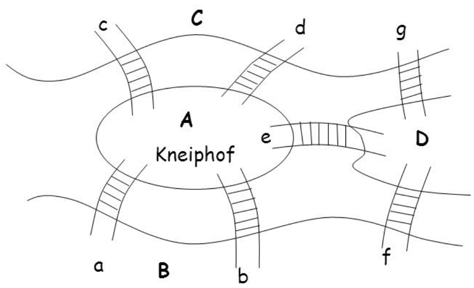

Figure 2.1 Königsberg bridge problem
source: http://slideplayer.com/slide/4035313/13/images/2/Konigsberg+Bridge+Problem+(Cont.).jpg

In 1857 William Rowan Hamilton, an Irish mathematician invented a puzzle (the Icosian Game) which involved finding a path that begins and ends at the same node, while passing through each nodes exactly once, which will be called as Hamiltonian circuit. The graph that involves the possibility of having Hamiltonian circuit is known as a Hamiltonian graph [3].

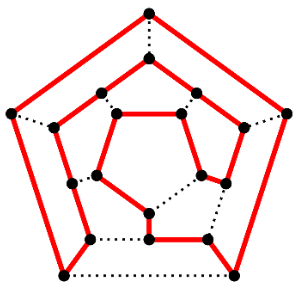

Figure 2.2 Hamilton circuit
source: https://en.wikipedia.org/wiki/Hamiltonian_path

B. Definition

Conceptually, a graph is formed by vertices and edges connecting the vertices [2]. Formally, a graph is a pair of sets vertices V and edges E. Mathematically, the notation of a graph is

$$G(V, E)$$

where V is the set of vertices and E is the set of edges.

Graphs are represented visually by dots (nodes) for vertices and lines for edges.

Example: Graph is formed by lines and nodes

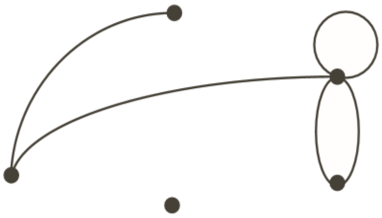

Figure 2.3

source: http://math.tut.fi/~ruohonen/GT_English.pdf

Often, the vertices are labeled with letters (for example: v_1, v_2, v_3, ... or *a, b, c, ...*), or numbers (1, 2, 3, …). Similarly, the edges are also often labeled with letters or numbers, but often it is not labeled (for example: *e1, e2, e3, ...; a, b, c, ...* or 1, 2, 3, …). Figure 2.4 will show the labeled vertices and edges graph [2].

Example: Vertices and edges labeling

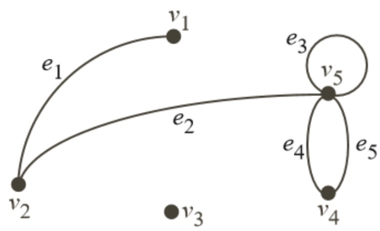

Figure 2.4

source: http://math.tut.fi/~ruohonen/GT_English.pdf

A graph can have loops and multiple edges. In Figure 2.4, the edge e_3 goes from and to the same vertex v_5. This kind of edge (v, v) is called a loop. The edge e_4 and e_5 connects the same two vertices (v_4, v_5). This is called multiple edges [2].

C. Types of Graph

Based on the existence of loops or multiple edges, a graph could be classified into:

1. Simple Graph

   A graph that doesn't consist of neither loops nor multiple edges is called a simple graph [1].
2. Non-simple Graph

   A graph that consist of either loops or multiple edges, or both of them is called a non-simple graph [1].

Based on the orientation of the edges, a graph could be classified into:

1. Undirected Graph

   A graph which doesn't have the directional character on its edges is called undirected graph. The relation between vertices in an undirected graph are symmetric [5].
2. Directed Graph

   A directed graph is a graph which its edges have direction properties that direct from one vertex to another [6]. The edges relation with a vertex in a directed graph can be classified into incoming edges (the edges which directing to the vertex) and outgoing edges (the edges which directing to other vertices from the vertex).

There are also some unique simple graphs that are often used:

1. Complete Graph

   A complete graph is a simple graph that every vertex has edges to all other vertices [1].
2. Circle Graph

   A circle graph is a simple graph with each vertices has a degree of two [1].
3. Regular Graph

   A regular graph is a simple graph with all the vertices has the same degree [1].
4. Bipartite Graph

   A bipartite graph is a simple graph whose set of vertices can be separated into two parts in a such way that every vertex on one part connects with one or more vertices on the other part [1].

D. Graph Basic Terminology

In graph theory, there are some terms that are often used, which are:

1. Adjacent

   Two vertices in an undirected graph are called adjacent if both of the vertices are directly connected by one or more edges [1].
2. Incident

   An edge $e = (v_1, v_2)$ are called to be incident with vertex v_1 and v_2 [1].
3. Degree

   In an undirected graph, degree of a vertex is total number of edges connected to the vertex.

   In a directed graph, degree is separated into indegree and outdegree. Indegree of a vertex is total number of incoming edges connected to the vertex. Outdegree of a vertex is total number of outgoing edges connected to the vertex [1].
4. Isolated Vertex

   A vertex is called an isolated vertex if it doesn't have any edges incident with it. The degree of an isolated vertex is 0 [1].
5. Null Graph or Empty Graph

   A null graph or empty graph consists of vertices with degree of 0, which means all of its vertices are isolated vertex [1].
6. Path

   A path between v_0 and v_n is the edges connecting v_0 to v_n in such a way that $e_1 = (v_0, v_1)$, $e_2 = (v_1, v_2)$, $e_3 = (v_2, v_3)$, …, $e_n = (v_{n-1}, v_n)$. The length of the path is total number of the edges in the path [1].
7. Cycle or Circuit

   A path which begins and ends at the same vertex is called a cycle or a circuit [1].
8. Connected

Two vertices are called connected with each other if one or more paths are connecting them.

A graph is called a connected graph if each pairs of vertices in the set of vertices in the graph are connected. A graph is called a disconnected graph if not all pairs of vertices are connected.

A directed graph is called a connected graph if its undirected graph is a connected graph. Two vertices *u* and *v* in a directed graph are called strongly connected if there are one or more directed paths from *u* to *v* and from *v* to *u*, else they are called weakly connected [1].

9. Subgraph and

   Assume $G = (V, E)$ is a graph. Then, $G_1 = (V_1, E_1)$ is a subgraph of graph G if $V_1 \subseteq V$ and $E_1 \subseteq E$ [1].

10. Spanning Subgraph

    A subgraph of a graph *G* which its vertices contains all vertices in graph *G* is called a spanning subgraph [1].

11. Cut-Set

    A cut-set of a graph is a set of edges which, when removed, cause the graph become disconnected graph. A cut-set cannot contain another cut-set [1].

12. Weighted Graph

    A weighted graph is a graph which each edges is given a numerical weight [1].

D. Eulerian Graph

1. Euler Path

   A path in a graph is called Euler Path if it traverses every edge on the graph exactly once.

2. Euler Circuit

   A circuit in a graph is called Euler Circuit if it traverses every edge on the graph exactly once.

3. Eulerian Graph

   A graph which contains Euler Path or Euler Circuit is called a Eulerian Graph.

Example: Eulerian Graph (*e.g.* path 1 0 3 4 2 1)

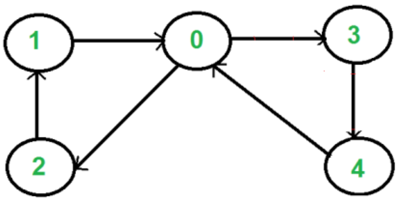

Figure 2.5

source: http://www.geeksforgeeks.org/wp-content/uploads/SCC1.png

E. Hamiltonian Graph

1. Hamilton Path

   A path in a connected graph is called Hamilton Path if it crosses every vertex on the graph exactly once.

2. Hamilton Circuit

   A circuit in a connected graph is called Hamilton Circuit if it crosses every vertex on the graph exactly once.

3. Hamiltonian Graph

   A graph which contains Hamilton Path or Hamilton Circuit is called a Hamiltonian Graph.

Example: Hamiltonian Graph

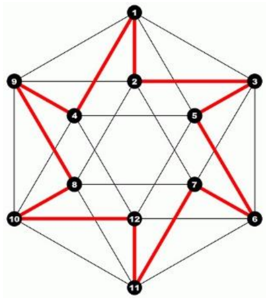

Figure 2.6

source: https://farm6.static.flickr.com/5288/5359590467_f0bf465586.jpg

III. Application of Graph Theory in Daily Life

A. Navigation Apps

Navigation application such as Google Maps, Waze, and Maps.me helps us to find the shortest and fastest route to our destination. In fact, those navigation apps consist of one huge graph, with lots of vertices and edges [7]. The graph that is used would be a weighted graph and it would look something like in Figure 3.1.

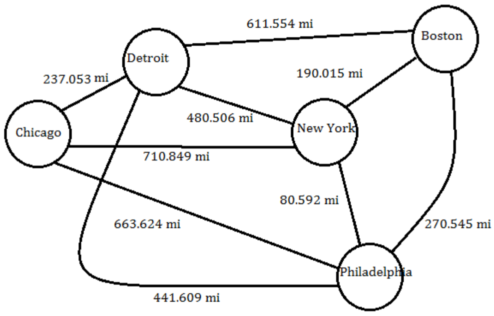

Figure 3.1 Graph representation in navigation apps

source: http://blogs.cornell.edu/info2040/2011/09/14/google-maps-its-just-one-big-graph/

There vertices in the apps can be a city, an intersection, or even a certain address such as a university, while the edges are the roads or maybe train railway [7]. The two-way roads are represented by non-directional or bidirectional edges, and the one-way roads are represented by the directional edges. The weight in each edge represents the distance of two vertices and the level of congestion in the road.

An "add stops" feature in some navigation apps uses the Hamilton Path. The stops that user added will be new vertices. It finds the shortest route to the destination while it needs to cross every stop that user have added.

Navigation apps can calculate the shortest route by calculating the minimum distance from the node that represents the start point, to the destination node. It would take a long time to calculate the shortest path moreover it needs to count the level of the congestion of the path. According to the [7], these calculations are done by using Dijkstra's algorithm.

B. Clearing Road Blockage

In a subtropical country, snow in the winter sometimes blocks the roads in the city. Putting salt on the roads is needed to melt the ice. The Euler Paths or Euler Circuits can be used to plan most efficient way to be traversed by the salt trucks.

C. Social Networks

Social networks (social media) such as *Line*, *WhatsApp*, and *Instagram* connects people with others by certain kind of relationship, commonly friends or followers. Those can be represented as a graph, with people as the vertices and their relationship as edges.

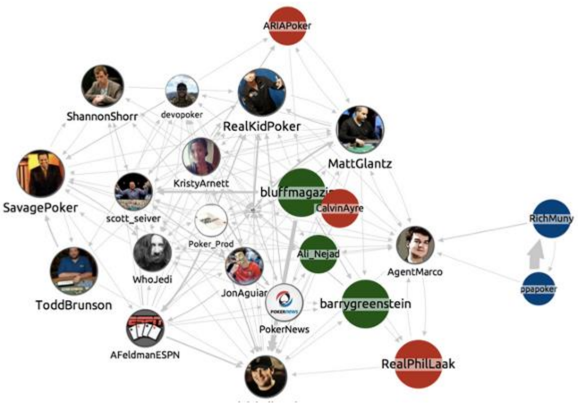

Figure 3.2 Graph representation of social network
source: https://cambridge-intelligence.com/wp-content/uploads/2013/11/poker-network.png

Social networks are not static [8]. Vertices and edges are formed and vanished as the time goes by. Some principles are used to maintain the appearing and disappearing vertices and edges, such as:

1. Triadic Closure
   If two people in a social network have a friend in common, then there is an increased likelihood that they will become friends [8].

Example: B and C have an incentive to become friends

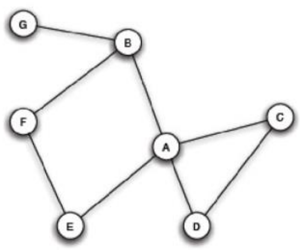

Figure 3.3
source:
http://eng.uok.ac.ir/esmaili/teaching/spring2012/sna/slides/Lecture2.pdf

2. The Clustering Coefficient
   The Clustering Coefficient is generalization of Triadic Closure. The Clustering Coefficient of a vertex A is a probability that two randomly selected friends of A are friends with each other [8].

D. Mobile Apps Designing

Mobile apps designing can be modeled as a graph, where its activities are represented by vertices and the links between activities are represented by edges in the graph. The edges are directional edges which usually begins from a button and ends at the other activity that it links with.

E. PageRank

The most commonly used search engine in the world, Google Search Engine is based on one algorithm called PageRank. Originally created by Larry Page and Sergey Brin in 2008, PageRank is an algorithm that based on a simple graph [9].

When someone searches a query, the search engine will parse the string and find the webpages with similar string. Finally, the PageRank will show the matches webpages from the best rank to the worst.

The PageRank graph is generated by representing all World Wide Web pages as vertices and hyperlinks on the pages as edges. The edges are weighted edges, whose weight represents the strength. Webpages that are linked by more credible source such as CNN or The Washington Post will have higher weightings for its edges [9]. Thus, the PageRank will give better rank for the page.

```
extern void page_rank(size_t size)
{
        matrix w;
        matrix_init(&w, size);
        gen_web_matrix(&w);

        matrix g;
        matrix_init(&g, size);
        gen_google_matrix(&g, &w);

        matrix_free(&w);

        vector p;
        vector_init(&p, size);

        matrix_transpose(&g);

        matrix_solve(&p, &g);

        vector_sort(&p);

        vector_save(&p);

        matrix_free(&g);
        vector_free(&p);
}
```

Figure 3.5 PageRank source code example in C++
source: https://github.com/nazgob/PageRank/blob/master/algorithm.c

F. Drug Designing

A drug consists of certain molecules. Each molecule consists of certain atoms and bonds between its elements. Each molecule can be represented as a graph with its elements as the vertices and the bonds between atoms as the edges.

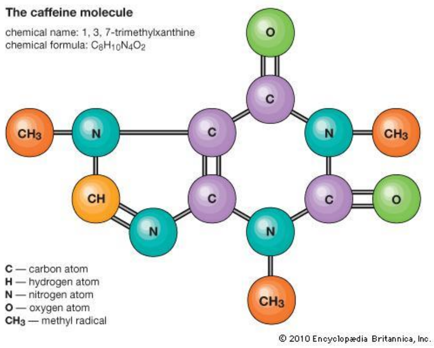

Figure 3.6 Caffeine molecule represented as a graph
source: https://media1.britannica.com/eb-media/28/105428-004-13BB2171.jpg

A certain study about graph in drug designing is molecular similarity. It is a measure of the degree overlap between a pair of molecules in some property space [10]. Molecular similarity searching is important for drug discovery. A reduced graph is used to do similarity searching [10].

G. Game Development

Creating a game needs a lot of graph. The most obvious thing in a game that uses a graph is the map. In a game map, certain location such as buildings, home, and markets are represented as the vertices of the graph, and the roads connecting it are represented as the edges.

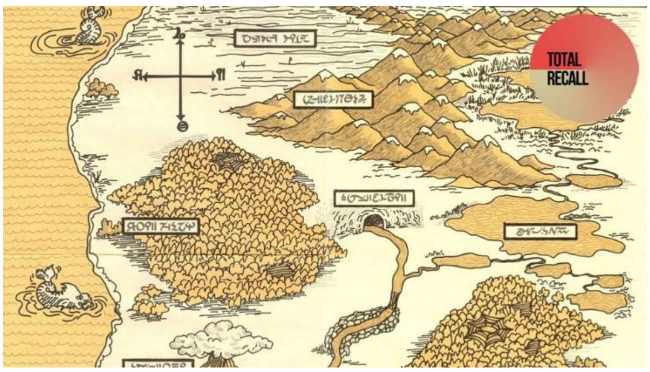

Figure 3.7 Map of a game Zork
source: http://cache.gawkerassets.com/assets/images/9/2011/09/zorkhead.jpg

Achievement system in a game is represented by a directed graph. The achievement system is like when the player has achieved this achievement, next achievements will be unlocked. The achievement is represented as the vertices and the edges represent the next-unlocked achievement.

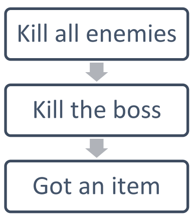

Figure 3.8 Achievement system example

Artificial Intelligence (AI) in a game also represented by graphs, usually directed graph. The enemy that can move and attack by itself, the vehicle on the street that can move well along the road, it is Artificial Intelligence in a game. Graph about Artificial Intelligence will be explained in the sub-section.

H. Artificial Intelligence (AI)

Artificial Intelligence (AI) is an area of computer science that emphasizes the creation of intelligent machines that work and react like humans [11]. An Artificial Intelligence in the game means the "brain" of enemies or companion. Artificial Intelligence includes:

1. Speech recognition
2. Problem solving
3. Knowing
4. Perception
5. Ability to manipulate and move objects

Finite-State Machine (FSM) is a model of computation based on a hypothetical machine made of one or more states [12]. Finite-State Machines are commonly used to organize and represent an execution flow, which is useful to implement AI in games [12]. In a Finite-State Machine, only a single state can be active at once. Then there must be a transition function to change from one state to another state. Every state in FSM represents an action, such as attack or evade.

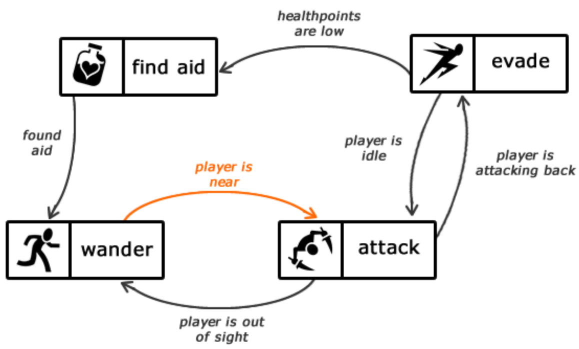

Figure 3.9
source: https://gamedevelopment.tutsplus.com/tutorials/finite-state-machines-theory-and-implementation--gamedev-11867

A Finite-State Machine can be represented by a directed graph where the vertices are the states and the edges are the transitions [12]. Each edge is labeled by certain condition, for example in Figure 3.8, the yellow edge is labeled "player is near", which indicates to change state from "wander" to "attack".

```
public class StackFSM {
    private var stack :Array;

    public function StackFSM() {
        this.stack = new Array();
    }

    public function update() :void {
        var currentStateFunction :Function = getCurrentState();

        if (currentStateFunction != null) {
            currentStateFunction();
        }
    }

    public function popState() :Function {
        return stack.pop();
    }

    public function pushState(state :Function) :void {
        if (getCurrentState() != state) {
            stack.push(state);
        }
    }

    public function getCurrentState() :Function {
        return stack.length > 0 ? stack[stack.length - 1] : null;
    }
}
```

Figure 3.10 Code implementation of stacked-FSM in Java

source: https://gamedevelopment.tutsplus.com/tutorials/finite-state-machines-theory-and-implementation--gamedev-11867

I. Flight Route

A flight company uses graph to represent their flight route. The airports are represented by vertices and the air routes that connects between a pair of airports are represented by edges in the graph.

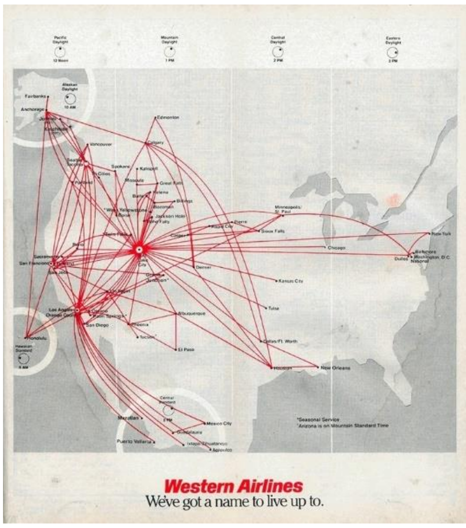

Figure 3.11 Airlines flight route

source: https://i.pinimg.com/736x/fb/c6/25/fbc6255889c628013c67f63028222bd3--commercial-aircraft-aviation-art.jpg

Airline company usually give some choices of flight route when their customer looks for a flight to a certain destination. There will be several choices which include a connecting (transit-included) flight and direct flight (non-transit flight). Its algorithm uses Dijkstra's Algorithm to find the shortest route to the destination.

IV. CONCLUSION

Graph theory is used in a lot of aspects in people's daily life. Lots of problem in human life can be solved more easily by model it into a graph. Lots of new technologies that are used daily are also invented by graphing. For example, Artificial Intelligence, social networks, and navigation apps. That's why people are suggested to learn about graph theory in order to solve their problem more easily.

V. ACKNOWLEDGMENT

The author would like to thank all family members and friends for their support throughout the making of this paper until it's finally done. Especially the author would like to thank Dr. Ir. Rinaldi Munir, M.T. for his passion through all his lectures throughout the semester.

REFERENCES

[1] http://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2015-2016/Graf%20(2015).pdf
[2] http://math.tut.fi/~ruohonen/GT_English.pdf
[3] https://www.britannica.com/topic/graph-theory
[4] Harary, Frank (1969), Graph Theory, Reading, MA: Addison-Wesley.
[5] http://mathworld.wolfram.com/UndirectedGraph.html
[6] http://mathinsight.org/definition/directed_graph
[7] http://blogs.cornell.edu/info2040/2011/09/14/google-maps-its-just-one-big-graph/
[8] http://eng.uok.ac.ir/esmaili/teaching/spring2012/sna/slides/Lecture2.pdf
[9] http://blogs.cornell.edu/info2040/2011/09/20/pagerank-backbone-of-google/
[10] https://www.slideshare.net/Reihaneh_SOH/application-of-graph-theory-in-drug-design
[11] https://www.techopedia.com/definition/190/artificial-intelligence-ai
[12] https://gamedevelopment.tutsplus.com/tutorials/finite-state-machines-theory-and-implementation--gamedev-11867
[13] https://www.math.cmu.edu/~af1p/Teaching/OR2/Projects/P6/PAPERPLANES.doc

PERNYATAAN

Dengan ini saya menyatakan bahwa makalah yang saya tulis ini adalah tulisan saya sendiri, bukan saduran, atau terjemahan dari makalah orang lain, dan bukan plagiasi.

Bandung, 3 Desember 2017

Nicholas Wijaya
13516121