March 2022

Using Graph Theory for Design and Safety in Railroad Systems

Kaleigh R. Poirier

USING GRAPH THEORY FOR DESIGN AND SAFETY IN RAILROAD SYSTEMS

By Kaleigh R. Poirier

A Field Project Submitted in Partial Fulfillment

of the Requirements for

the Honors Program

in the Department of Mathematical Sciences

The Faculty of Arts and Sciences

Rhode Island College

2021

Acknowledgements

A special thank you to Dr. Mark Medwid for assisting in this project.

Abstract

Trains, a method of transportation for people and/or goods, are different than other methods of transportation, such as boats or trucks. Trains can only travel in a railroad system, specifically upon tracks. If something is preventing the train from moving on the tracks, then that whole line of railroad is shut down. If boats and trucks cannot travel on a particular route, there are alternative routes that can be taken. The design of the railroad system may appear to be created by chance, in that no trains crash into each other or into stations, but there is an application to graph theory which can help determine safety and efficiency in the railroad system. Additionally, there are types of graphs and information that are used specifically for railroad systems in graph theory, which will be discussed briefly. In this paper, there is an introduction to graph theory, its properties, and a basic application of graph theory to the railroad system. Additionally, three train lines in the *London Underground* are transformed into graphs, each of which are then analyzed, in respect to one of three design and safety problems of railroad systems (the blocking problem, the yard location problem, and the train schedule problem). Overall, it was found that graph theory is effective in developing railroad systems and that the three design and safety problems of railroad systems can be analyzed by using graph theory.

Contents

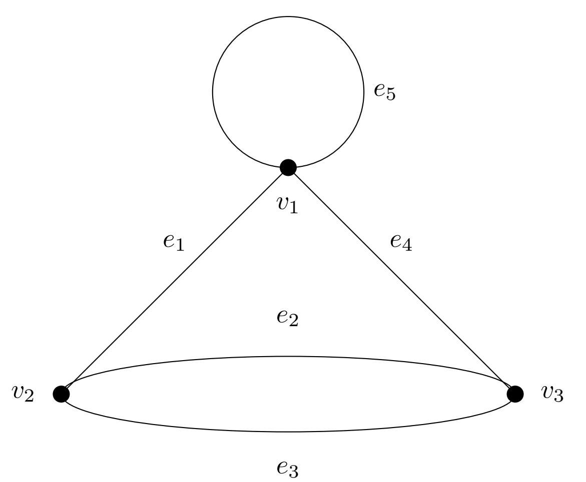
Chapter 1

Introduction

In order to relate graph theory to design and safety in the railroad system, we must define graphs. Graphs will be defined as:

> **Definition 1.1** A **graph** G consists of two finite sets: a nonempty set $V(G)$ of **vertices** and a set $E(G)$ of **edges**, where each edge is associated with a set of consisting of either one or two vertices called its **endpoints**. The correspondence from edges to endpoints is called the **edge-endpoint function**. [9]

Example 1.1 Consider a graph G that consists of vertices: v_1, v_2, and v_3 and edges e_1, e_2, e_3, e_4, and e_5.



From Example 1.1, the vertices and edges of the graph can be listed as:
$V(G) = \{v_1, v_2, v_3\}$ and $E(G) = \{e_1, e_2, e_3, e_4.e_5\}$. Additionally, the edge-endpoint function of G is as follows:

Edge	Endpoints
e_1	$\{v_1, v_2\}$
e_2	$\{v_2, v_3\}$
e_3	$\{v_2, v_3\}$
e_4	$\{v_1, v_3\}$
e_5	$\{v_1\}$

Different edges and vertices of the graph can also be described with other terminology of the graph, G.

Definition 1.2 An edge with just one endpoint is called a **loop**, and two or more distinct edges with the same set of endpoints are said to be **parallel**. An edge is said to **connect** its endpoints; two vertices that are connected by an edge are called **adjacent**; and a vertex that is an endpoint of a loop is said to be **adjacent to itself**. An edge is said to be **incident on** each of its endpoints, and two edges incident on the same endpoint are called **adjacent**. [9]

Referring back to Example 1.1 and using Definition 1.2, we can say that e_5 is a loop, e_2 and e_3 are parallel to each other, e_1 and e_4 are incident to v_1, v_2 and v_3 are adjacent to v_1, and e_1, e_2, and e_3 are adjacent to e_4.

Going forward, when we discuss the relationship between graph theory and the railroad system itself, we can let different aspects of graph theory represent the parts in the railroad system. We are going to let edges represent the railroad tracks and vertices represent train stations and train stops. Additionally, if we were to refer back to Example 1.1 and let that be a railroad system, we can say that a train can travel upon a route from station 1 (v_1) to station 2 (v_2), the train would travel upon e_1. A route, similar to the brief example just given can be explained in other terminology, such as walks, trails, paths, and circuits. These will be defined as follows:

Definition 1.3 Let G be a graph, and let v and w be vertices in G. A **walk from *v* to *w*** is a finite alternating sequence of adjacent vertices and edges of G. Thus a walk has the form:

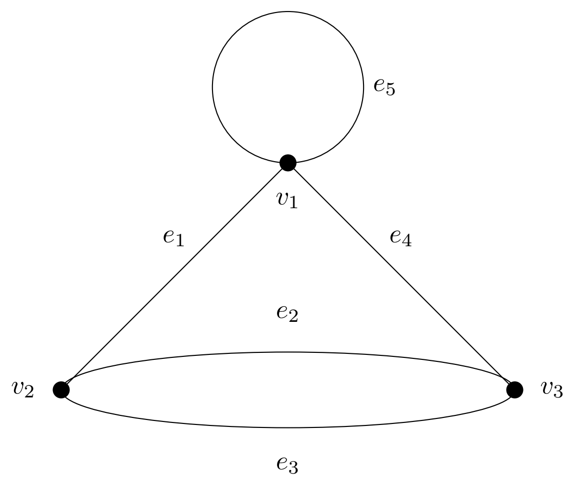
$$v_0 e_1 v_1 e_2 \dots v_{n-1} e_n v_n,$$

where the v's represent vertices, the e's represent edges, $v_0 = v$, $v_n = w$, and for each $i = 1, 2, \dots n, v_{i-1}$ and v_i are the endpoints of e_i. A **trail from *v* to *w*** is a walk from v to w that does not contain a repeated edge. A **path from *v* to *w*** is a trail that does not contain a repeated vertex. A **closed walk** is a walk that starts and ends at the same vertex. A **circuit** is a closed walk that contains at least one edge and does not contain a repeated edge. A **simple circuit** is a circuit that does not have any other repeated vertex except the first and last. [9]

Additionally, vertices in a graph can have an even or odd degree.

> **Definition 1.4**: Let G be a graph and v a vertex of G. **The degree of v**, denoted **deg(v)**, equals the number of edges that are incident on v, with an edge that is a loop is counted twice. A vertex that has an **even degree** is connected to $2n$ (where $n \in \mathbb{N}$ or $n = 0$) edges. A vertex that has an **odd degree** is connected to $2n + 1$ (where $n \in \mathbb{N}$ or $n = 0$) edges. [9]

Example 1.2 Refer back to graph G in Example 1.1.



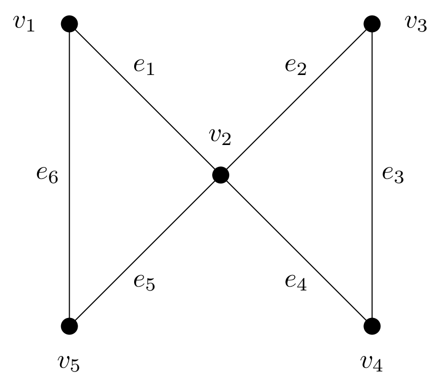
We can see that $deg(v_1) = 4$ (an even degree), since e_1 and e_4 are incident to v_1 and the loop e_5 is also incident to v_1. We can also see that $deg(v_2) = 3$ and that $deg(v_3) = 3$ (both of odd degrees), as e_1, e_2, and e_3 are incident to v_2 and e_2, e_3, and e_4 are incident to v_3.

There are different types of circuits, but specifically we will be focusing on two: Euler Circuits and Hamiltonian Circuits.

1.1 Euler Circuits

> **Definition 1.5:** Let G be a graph. An **Euler circuit** for G is a sequence of adjacent vertices and edges in G that has at least one edge, starts and ends at the same vertex, uses every vertex of G at least once, and uses every edge of G exactly once. [9]

Example 1.3 This is an example of an Euler circuit. Consider the following graph A.



An Euler circuit for A would be $v_1e_1v_2e_2v_3e_3v_4e_4v_2e_5v_5e_6v_1$. This is an Euler circuit because every edge is used exactly once and every vertex is used at least once.
Additionally, we can note that every vertex of A has an even degree. [13]

Theorem 1.1: If a graph G has an Euler circuit, then every vertex of the graph has a positive even degree.

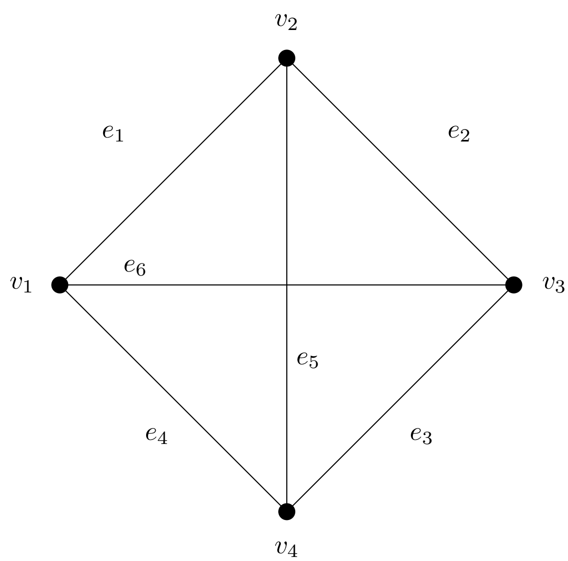
Proof: Let G be a graph that has an Euler circuit.
Let v be any vertex in G.
We want to show that every vertex of the graph has a positive even degree.
By the definition of an Euler circuit, we know that circuit contains every edge of G.
Therefore, we can say that the circuit contains all edges incident on v.
Even though graph edges do not have a "middle" or "halfway point," as they are noncontinuous objects, we can let an edge have a middle, such that after we have traveled from one vertex to the edge incident to it for a while, we have passed the "middle" of that particular edge.
Beginning at the "middle" of an edge adjacent to the starting of the Euler circuit, we can travel through the Euler circuit, until we reach the middle of the starting edge.
Every time we traveled along one edge, after moving from that edge, we went to another, new edge.
Therefore, as we "exited" one edge, we "entered" another.
We know that every edge in G is traveled over exactly once, and therefore, every edge incident on v is traveled through exactly once.
Therefore, the edges incident on v occur as "entry" and "exit" pairs.
Therefore, the degree of v must be $2n$, where $n \in \mathbb{N}$.
Thus, if a graph has an Euler circuit, then every vertex of the graph has a positive even degree. [16]

■

1.2 Hamiltonian Circuits

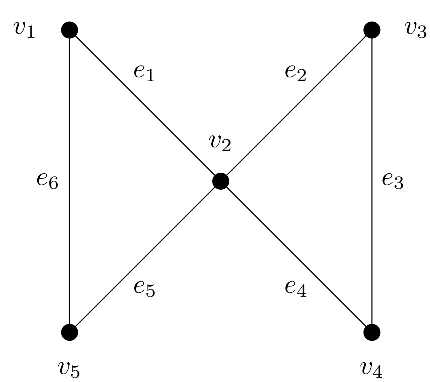
Definition 1.6: Given a graph G, a **Hamiltonian circuit** for G is a simple circuit that includes every vertex of G. That is, a Hamiltonian circuit for G is a sequence of adjacent vertices and distant edges in which every vertex of G appears exactly once, except for the first and the last, which are the same. [9]

Example 1.4 Consider the graph B below. Note that this is the same graph as in Example 2.1, except with different defined edges and vertices.



A Hamiltonian circuit would be $v_1e_1v_2e_2v_3e_3v_4e_4v_1$. Note that not all graphs have Hamiltonian circuits, though.

Example 1.5 Refer back to graph A from Example 2.2. A picture of Graph A is below.



In order for there to be a Hamiltonian circuit in A, we would need to travel along the graph twice through v_2. This cannot occur, as according to the definition of an Hamiltonian circuit, every vertex in A can only appear once in the sequence, with the exception of the starting vertex In this case v_2 would appear twice (more than two times if v_2 was the starting vertex). Therefore, A does not contain a Hamiltonian circuit. [9]

Chapter 2

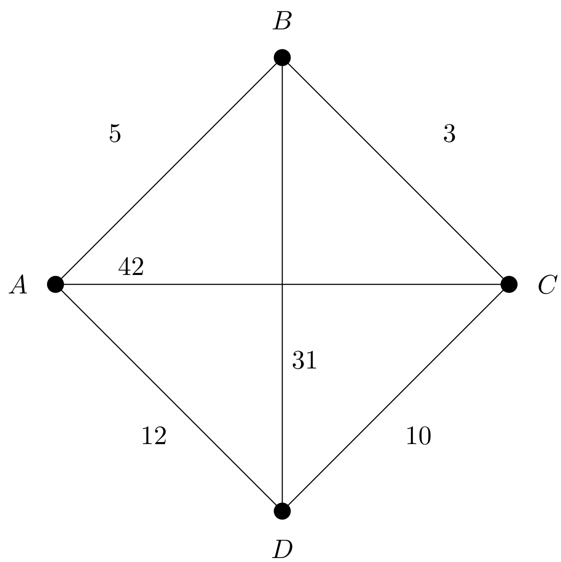
Types of Graphs

A graph without any other information, would be the type of object defined in Chapter 1 (specifically in Definition 1.1 and shown in Example 1.1). A graph with an adjective in front of the word "graph" would refer to a type of graph with a particular structure. In this chapter, we will be discussing three different types of graphs: weighted graphs, direct graphs, and double vertex graphs. First, we will discuss weighted graphs and its properties.

2.1 Weighted Graphs

Weighted graphs are similar to graphs (defined in Definition 1.1), with the exception that each edge, e, carries a "weight," such as the distance between two vertices. Weighted graphs can be used to determine the shortest and most efficient route of travel. Specifically, this is seen in two famous examples, the *Traveling Salesman Problem* and the *Chinese Postman Problem*. [9]

Example 2.1 This example will be a version of the *Traveling Salesman Problem*. A more general example of this problem can be found in [9].
Consider a graph H with the following vertices and edges (that have weights).



Imagine that H represents a route, where A, B, C, and D are different cities. Each edge connecting to cities represents the total length (in kilometers) between the two cities. For example, between City A and City B the travel between the two cities is 5 kilometers. A person has to travel from City A, visit every other city only once, and return back to City A. What is the shortest route this person can possibly take? There are six possible routes and the total distances below:

Route	Total Distance of Route (km)
$ABCDA$	30
$ABDCA$	88
$ACBDA$	88
$ACDBA$	88
$ADBCA$	88
$ADCBA$	30

From the table, we can see that the shortest routes to take are either $ABCDA$ or $ADCBA$, which are both 30 kilometers. Additionally, since this is the same graph structure as in Example 1.4, the circuits, $ABCDA$ and $ADCBA$, would also be considered Hamiltonian circuits. We can use information and graphs like this construct railroad systems, but there are more efficient graphs that take into perspective everything that is going on it a railroad system. Note that this version of the problem was inspired by two examples, found in [5] (*The Chinese Postman Problem*) and [9] (*The Traveling Salesman Problem*). Therefore, we can note that in some graphs, one can travel upon edges in both directions, meaning that the edges do not have any specified direction. An edge that can be travel upon in both directions is denoted pictorially as a single line in a graph.

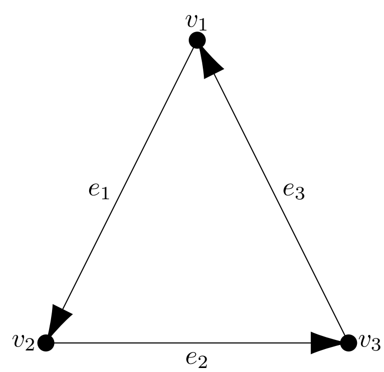
2.2 Directed Graphs

Directed Graphs, which are also known as "digraphs," will be defined as follows:

> **Definition 2.1** A **directed graph** D is a finite, nonempty set of vertices, together with some directed edges joining pairs of vertices. It can be said that the edge-endpoint function associates each edge, e to an ordered pair of vertices, (v, w), where v is the initial vertex ("starting" vertex) of edge e and w is the terminal vertex ("ending" vertex) of edge e. Directed graphs are held to additional restriction: that the initial and terminal vertices cannot be the same. [14]

In a pictorial representation of a graph, an edge that can be traveled upon in only one direction is drawn as a single line with an arrow.

Example 2.2 Consider graph D below. Graph D is a directed graph.

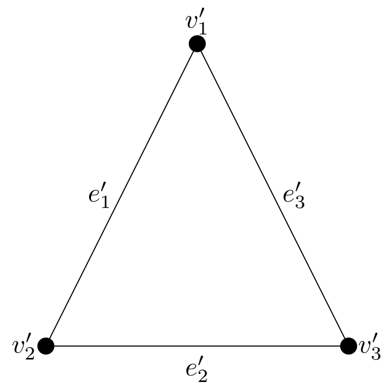


In order to complete a circuit that starts and ends at v_1, we can only perform the following circuit:

$$v_1 e_1 v_2 e_2 v_3 e_3 v_1$$

This is the only circuit, starting and ending at v_1 because of the directed edges.

Consider graph D'.



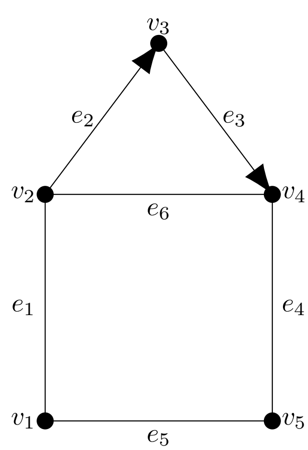
If we want to complete a circuit, with it starting and ending at v_1', we will have the following two circuits:

$$\text{Circuit 1} : v_1'e_1'v_2'e_2'v_3'e_3'v_1'$$
$$\text{Circuit 2} : v_1'e_3'v_3'e_2'v_2'e_1'v_1'$$

Therefore, since there are no directed edges, we can make a circuit by starting from v_1' to e_1' or from v_1' to e_3'.

Note that in a variations of directed graphs, there are some edges that are directed and some that are not. Directed edges can only be traveled upon in one direction and will always be denoted with an arrow. Edges that are not denoted with an arrow can be assumed to be undirected. An example is below.

Example 2.3 Consider graph H.



We can see that there are edges that can be traveled upon in both directions (e_1, e_4, e_5 and e_6) and edges that can only be traveled upon in one direction

(e_2 and e_3). Therefore, when applied to the railroad system, we can have train tracks that are only traveled upon in one direction, while there are other tracks that can be traveled upon in both directions.

2.3 Double Vertex Graphs and Time Table Data

The special type of graph used to design a railroad system is called "double vertex graphs." Unlike graphs (defined in Definition 1.1), weighted graphs, or directed graphs, pictorially, double vertex graphs look very different (see Figure 2.1). Additionally, these graphs hold different information. In weighted graphs, we saw that graph edges can have specific lengths, as we seen in Example 2.1, where we were trying to determine the shortest route possible for a person to travel. Additionally, in directed graphs, we saw that graph edges can be denoted with arrows and can only be traveled upon in one direction, as seen in Example 2.2. In a railroad system, not only do we need to know the length of the track itself or the direction that the train travels on that track, but we also need to take into consideration other aspects of the railroad system, such as speed of a train on a particular track.

> **Definition 2.2**: Let a **double vertex graph** T be a finite, nonempty set of vertices, $V(G)$, and there be be a finite set of edges, $E(G)$, between the vertices, where none of the edges are loops or that there are multiple edges between a pair of vertices. [7] Additionally, these graphs are specifically for railroad systems and every edge holds different pieces of information for the railroad, such as track length, maximum speed, etc. [5]

A double vertex graph holds specific, physical information about a railroad system that a graph, weighted graph, or directed graph, alone, of the railroad system would not have.

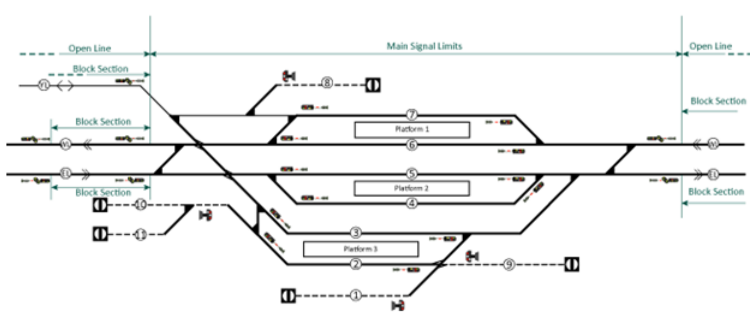


Figure 2.1 [5]

Additionally, for a double vertex graph there is "time table data." Time table data holds all information for each train at particular stations. Some examples of information that time table data includes are train connections, arrival time, departure time, and minimal stop time. This helps to calculate a train's movement. Additionally, in order to fully evaluate the safety and design of the whole railroad system, in time table data, the train stores its own information, such as position, power, speed, etc. Time table data can be evaluated in a simulation as shown in Figure 2.2. [5]

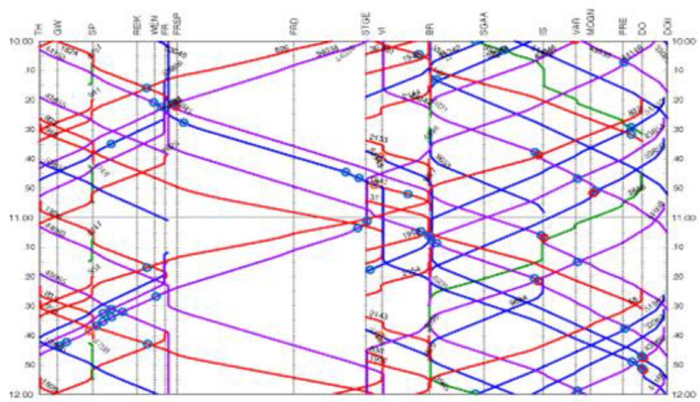


Figure 2.2 [5]

For example, if one was to consider two railroad systems that had equivalent graphs, yet the two railroad systems had different speeds of travel on their tracks, the two railroad systems cannot be designed in a similar manner. Different safety measures would need to be taken to reflect the specific needs of each railroad system, which would be done through the use of double vertex graphs to ensure safety. The use of double vertex graphs can truly ensure safety for travel in the railway system, as double vertex graphs provide specific design information, regarding the railway itself, rather than just depending alone on the "set up" of train tracks (edges) and train stations and stops (vertices). For the scope and purpose of this paper though, please refer to the article, *Graph Theory use in Transportation Problems and Railway Networks*, where the basis of information regarding double vertex graphs and time table information in this paper was obtained [5], or Section 3.4.2 in the article, *Algorithmic decision support for train scheduling in a large and highly utilised railway network* by Gabrio C. Caimi, for more information about double vertex graphs. [7]

Chapter 3

Design and Safety Problems in the Railroad System

When designing railroad systems, there are various safety and design problems that need to be considered. Railroad systems need to be designed very carefully, as some trains can only travel in one direction, but travel on a two-way rail. Therefore, one needs to prevent any train crashes from happening, so that if two trains are traveling upon the same rail, at the same time, and nothing is designed accordingly, it is likely that the two trains will crash into each other.

Even though there are multiple safety and design problems in relation to graph theory, for the scope of this paper, we will be focusing on three: the *blocking problem*, the *yard location problem*, and the *train schedule problem*. Please refer to the article, *Network Models in Railroad Planning and Scheduling* by Ravindra K. Ahuja, Cladio B. Cunha and Giivenç Şahin, to learn more about the other railroad design and safety problems. [18]

3.1 The Blocking Problem

One of the most important railway design and safety problems is the "blocking problem." The blocking problem "arises in the context that a railroad carries millions of shipments from their origins to their perspective destinations." [18] The goal of the problem is to minimize the total travel distance for the train (such as station changes or track merges), and therefore minimize the cost. Specifically, in this problem, we will be focusing on minimizing the total travel distance for trains.

A "block" is the system of edge(s) that the train travels on. In a block, each edge between two stations is called an arc. When a train is placed in a block, it will not be categorized until it reaches it destination. A "yard" is where the train travels to from its starting point, which we will call an origin, to its destination. Therefore, it can be said that a yard consists of the vertices that represent stations that are not the origin, or the destination station. From here, a blocking network can be created. [18]

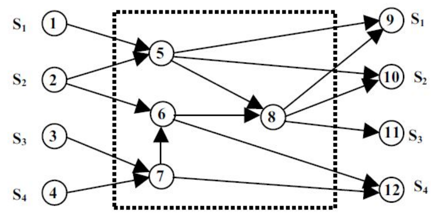


Figure 3.1 [18]

Figure 3.1 would be considered an example of a blocking network, where S_1, S_2, S_3, and S_4 can be considered our trains (or shipments), with the origins: nodes $1, 2, 3$, and 4, respectively. Note that nodes are similar to vertices, except that they are drawn with a circle and have number or letter inside the circle. The yards consist of the nodes 5, 6, 7, and 8, and the destinations are nodes 9, 10, 11, and 12. For example, S_4, which starts at node 4 has two blocking paths (ways that that the train can travel) to get to node 12, its destination. The two blocking paths are:

$$4 - 7 - 6 - 12$$
$$4 - 7 - 12$$

Therefore, in the first blocking pattern, the train flows on arcs (4,7); (7,6); and (6,12). In the second blocking pattern, the train flows on arcs (4,7) and (7,12). By creating these blocking patterns, we can determine the total minimized distance for the train to travel.

3.2 The Yard Location Problem

Another important problem that relates to the "blocking problem" in the design and safety of railroad systems is the "yard location problem." We try "...to find the best network configuration in terms of number and location of yards where cars can be reclassified into new blocks and switch trains." [18] We previously said that a "yard" is where the train travels to from its starting point, which we will call an origin, to its destination. Additionally, yards are also where trains from other train lines in the system are reclassified to travel to another destination. There are three types of yards: local yards, system yards, and regional yards. Local yards are when there are trains coming in from stations that are not in yards and are organized into blocks. System yards and regional yards are considered as "hub yards," as both deal with bigger amounts of trains and shipments. Local yards are on a smaller scale than system yards and regional yards. [18] Specifically, we will be focusing on local yards.

Additionally, the number of yards and the location of the yards, play an important role in this problem. Also, the focus of the yard location problem (like the blocking problem) is to minimize the total distance of the trains' travel. Without taking into consideration the number of yards and the location of yards, we cannot accurately minimize the route of the train, even if the blocking network is minimized, due to the increasing traffic on railroads and the addition of more trains. By considering both the blocking network, and the yards in a railroad system, we can determine a more accurate minimized route for a train to travel from its origin to its destination. [18]

3.3 The Train Schedule Problem

Lastly, another important problem in the design and safety of the railroad systems is making a train schedule. In this problem, we try to develop a schedule, such that we know the train routes, times of operation, and also determine the most efficient route; to make it the quickest, most cost-effective, and most time efficient. This can include looking at the number of days the train line operates, the total area of coverage of the train, and the cost per day it costs to operate the train, etc. [18] Additionally, the train route problem involves various decisions that have to be made, such as what the route of the train(s) is, what the times that the train arrives and leaves a station are, etc. [18] Specifically, this problem is similar to the *Traveling Salesman Problem* [9], which we gave a similar example in Chapter 2 (Example 2.1). Our goal in that problem was to determine the quickest route (in distance), such that the person visited every city. It is similar in the railroad system, our goal is to determine the quickest route, such that every station is visited at least once. Specifically, we will be looking at time and distance in the train schedule problem.

Chapter 4

Application to the London Underground

To apply graph theory to the aspects of design and safety in railroad systems, we will use the London Underground, which is commonly referred to as "The Tube." The London Underground contains eleven train lines [11] and covers a total of 250 miles. [4] Specifically, we are going to look at three lines: the Central Line (colored red), the Piccadilly Line (colored dark blue), and the Circle Line (colored yellow), and relate each line to one of three design and safety problems of the railroad system: the blocking problem, the yard location problem, and the train route problem, respectively.

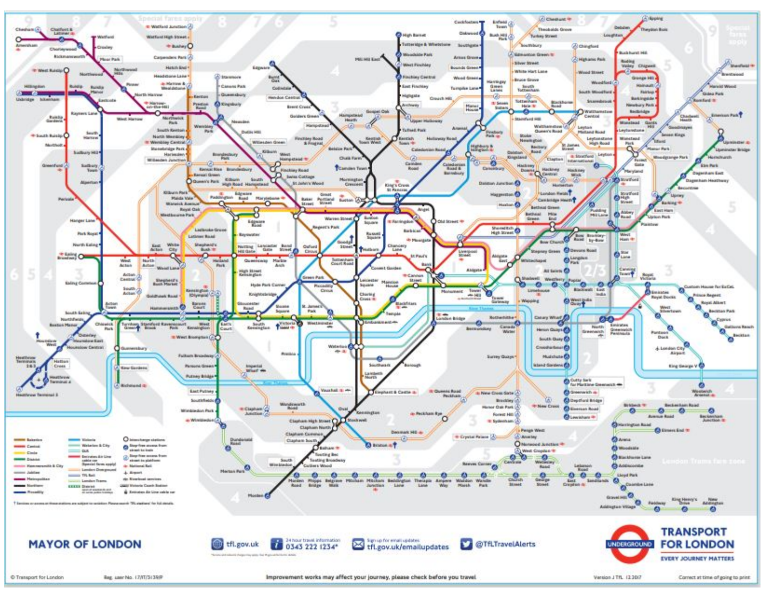


Figure 4.1 [12]

4.1 Central Line

First, we will look at the Central Line and consider the blocking problem. The Central Line is a total of 46 miles and has 49 stations. [1] Please refer to Figure 4.2 for a graphical representation of the Circle Line. Note that in Figure 4.2, there are a total of 52 vertices, where three of them are colored red. Every vertex that is black represents a station, and every vertex that is red represents what we will call a "merging point." A merging point means that these areas on the railroad are not stations, but rather places where the train joins new rail, or makes a turning point. Additionally, we are going to consider that there are two trains traveling on this line at once, as there are two "starting points" v_1 and v_8. The trains are also traveling at the same speed. We will say that Train 1 begins at v_1 and Train 2 begins at v_8 and that both trains depart from their first station at the same time. Note that though, in order to visit $v_{42}, v_{41}, v_{40}, v_{39}, v_{38}, v_{37}, v_{36}, v_{35}, v_{34}$, or v_{33}, there is a separate train that visits those stations and then rejoins the rest of the train line. [10] We will only be focusing on the two trains, where Train 1 starts at v_1 and ends at v_{52} and Train 2 starts at v_8 and ends at v_{42}, where we will change trains at v_{32}.

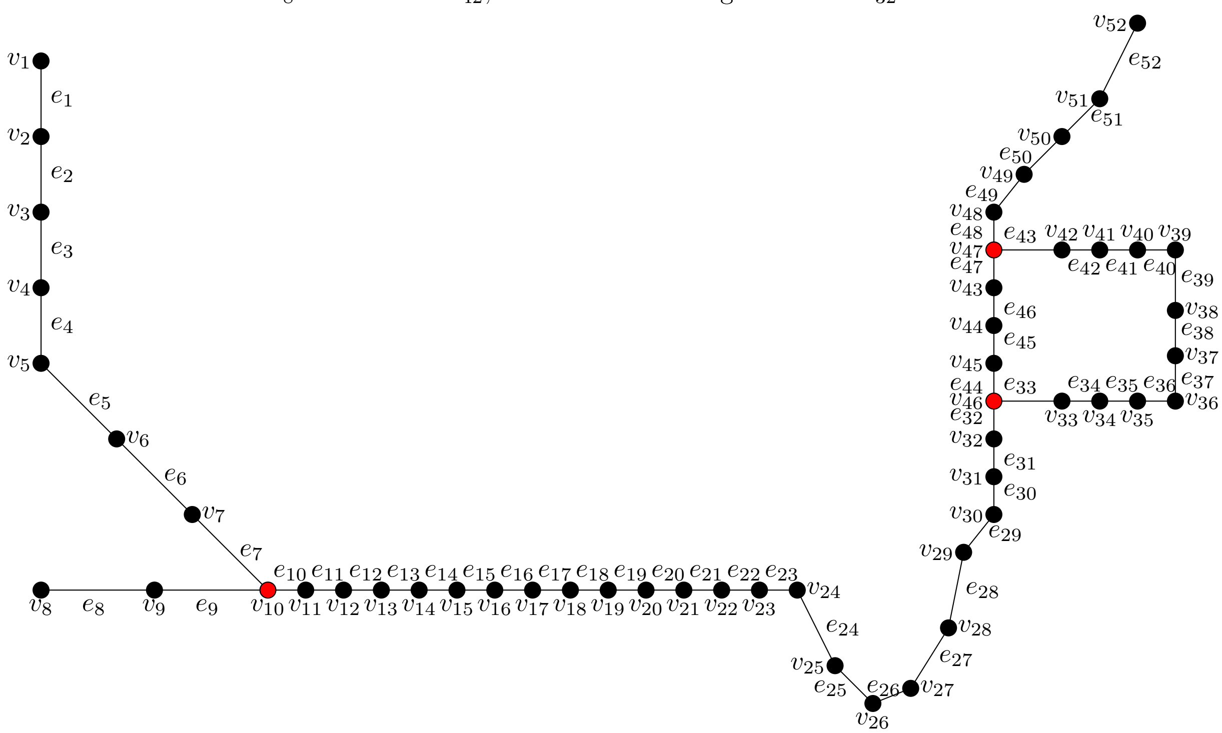


Figure 4.2

From a general graph theory explanation, our trains have two walks:

Train 1 : $v_1 e_1 v_2 e_2 \ldots v_{10} e_{10} \ldots v_{32} e_{32} v_{46} e_{44} \ldots v_{50} e_{51} v_{51} e_{52} v_{52}$

Train 2 : $v_8 e_8 v_9 e_9 v_{10} e_{10} v_{11} e_{11} \ldots v_{32} e_{32} v_{46} e_{33} v_{33} e_{34} \ldots v_{40} e_{41} v_{41} e_{42} v_{42}$

Since the walks of both trains can be very complex, we can create a blocking network which arises from the blocking problem. The blocking network can help to show a more simplistic view of the train route and rail changes. Figure 4.3 is an example of a blocking network of Trains 1 and 2 traveling to their previously specified destinations, v_{52} and v_{42}, respectively. Note that the dotted box the yard nodes specifically. Additionally, note that v_{32} is where Train 2 changes trains so that it can get to v_{42}.

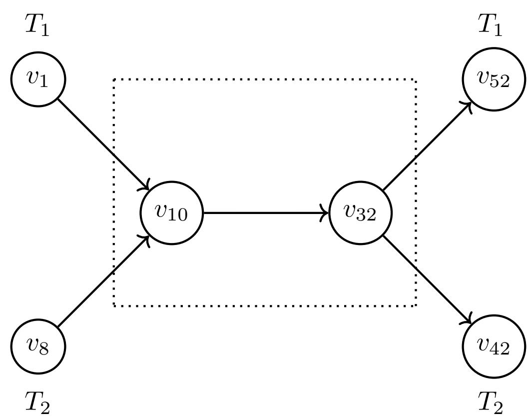


Figure 4.3

Therefore, it can be determined that we have two blocking patterns for Train 1 and Train 2, respectively:

$$v_1 - v_{10} - v_{32} - v_{52}$$
$$v_8 - v_{10} - v_{32} - v_{42}$$

The arcs that the train flows upon can also be determined for Train 1 and Train 2, respectively:

$$(v_1, v_{10}); (v_{10}, v_{32}); \text{ and } (v_{32}, v_{52})$$
$$(v_8, v_{10}); (v_{10}, v_{32}); \text{ and } (v_{32}, v_{42})$$

Additionally, from the blocking network we can see that Train 1 and Train 2 both travel to v_{10} and v_{32}, in order to get to their final destinations. Note that v_{10} is not a station, though, rather it is a place where the track merges together. Even though in the network it looks like the rails are the same length, if we refer back to Figure 4.2, we can see that v_8 to v_{10} is much shorter than v_1 to v_{10}, as there are more tracks to be traveled upon and more stations from v_1 to v_{10} that need to be visited. By having both the blocking network and a graphical representation of the Central Line, we can see that both trains will not crash into each other, as Train 2 will arrive at the merging point, v_{10}, much sooner than Train 1 will, and therefore Train 2 will arrive at v_{32} before Train 1. This will prevent any crashes between the two trains on the railroad. Lastly, the the blocking problem can give an overview of a railroad system, to see if the amount of trains being used and distance traveled is cost-effective.

4.2 Piccadilly Line

Now, we are going to consider the Piccadilly Line and the yard location problem. The Piccadilly Line consists of 53 stations and covers approximately 44 miles. [17] Refer to Figure 4.4 for a graphical representation of the Piccadilly Line.

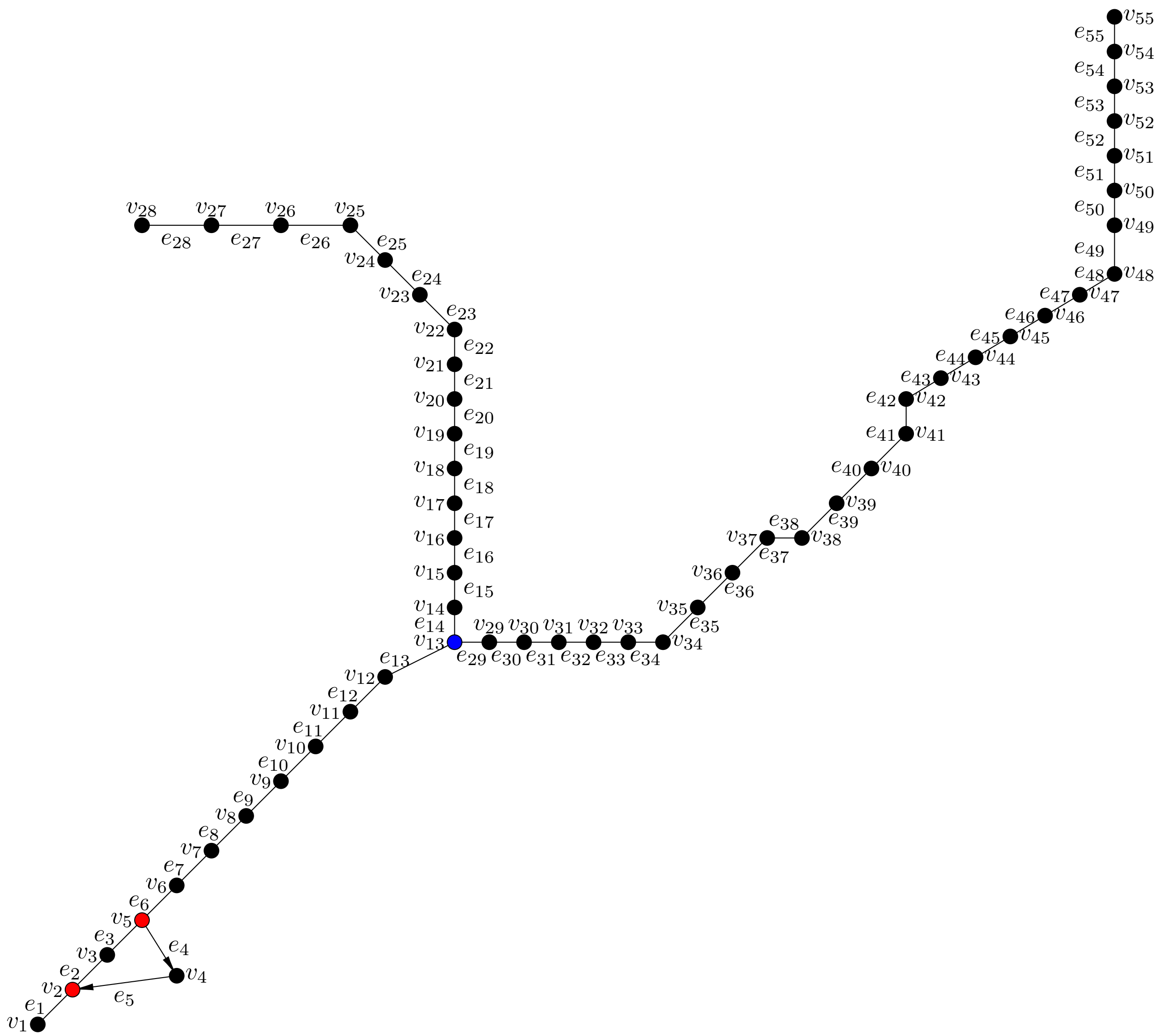


Figure 4.4

Note that there are 55 vertices, in which 53 are stations (colored black, except for v_{13}, which is colored blue, as v_{13} is a "merging point" and a station, where the trains join a the new track to get to the final destination, v_{55}) and two "merging points" (colored red). Note that these two merging points create a directed graph portion in the overall graph. Therefore, for the directed graph portion there is a walk:

$$v_1e_1v_2e_2v_3e_3v_5e_4v_4e_5v_2$$

As of 2021, though, this portion of the train station is closed and not accessible. [10] Again, we are going to be considering that we have two trains: Train 1, which starts at v_{28} and ends at v_{55} and Train 2, which starts at v_1 and ends at v_{55}. Note that both Train 1 and Train 2 are traveling at the same speed. Therefore, we can determine the following walks:

$$\text{Train 1}: v_{28}e_{28}v_{27}e_{27}\dots v_{13}e_{29}\dots v_{53}e_{54}v_{54}e_{55}v_{55}$$
$$\text{Train 2}: v_1e_1v_2e_2v_3e_3v_5e_6\dots v_{13}e_{29}\dots v_{53}e_{54}v_{54}e_{55}v_{55}$$

Additionally, in order to view the Piccadilly Line as a blocking network system. Figure 4.5 is a blocking network system of the Piccadilly Line.

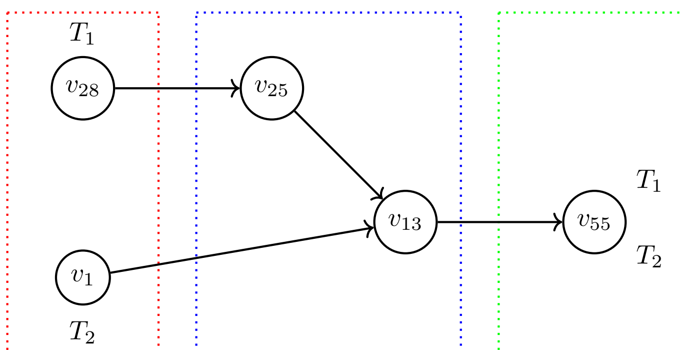


Figure 4.5

In Figure 4.5, the red dotted box highlights the origin nodes, v_{28} and v_1, that the trains leave from, the blue dotted box highlights the yard nodes, v_{25} and v_{13}, that both trains travel to before its final destination or the station where Train 1 turns (v_{25}), and the green dotted box highlights the destination nodes of both trains. Overall, this shows a local yard. [18] If we were to consider the whole London Underground, that would be a system yard. [18] Note that another train line in the London Underground, the District Line, also visits v_{13}. [11] If we were to refer back to Figure 4.5 and add that connection from the District Line, it would look like Figure 4.6.

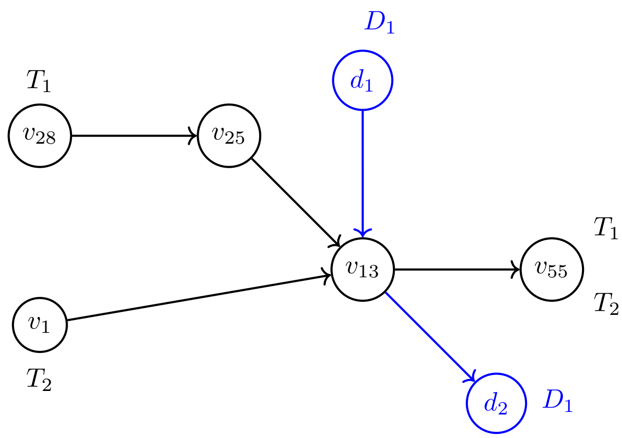


Figure 4.6

D_1 represents the train traveling along the District Line that has an origin node, d_1, and destination node, d_2. Note that these were drawn in blue to distinguish between the two different blocking systems: the Piccadilly Line blocking system and the District Line blocking system. By looking at the blocking network, we can see that three trains pass through v_{13}. It is important to look at the overlapping blocking systems, and specifically the yard nodes in the blocking systems, as all of the trains need to move at a pace, such that they do not crash into each other, or if they are going to arrive at the same station at the same time, they know how far in advance they have to stop. In order for Train 1 (T_1) to travel from v_1 to v_{55}, it takes approximately 1 hour and 41 minutes, and specifically it takes 37 minutes to travel from v_{28} to v_{13}. In order for Train 2 (T_2) travel from v_1 to v_{55}, it takes 1 hour and 33 minutes, and specifically it takes 29 minutes to travel from v_1 to v_{13}. Lastly, it takes D_1 6 minutes to travel from d_1 to v_{13}. [10] By just looking graphically at Figure 4.4, we can only see the Piccadilly Line, and by just looking at Figure 4.5, we can only see the Piccadilly Line blocking network. By combining some of the District Line blocking system with the Piccadilly Line blocking system (especially its yard nodes), along with time information, we can see overall that it unlikely that the three trains will crash into each other, as there is less distance and stations between d_1 and v_{13}. Even though arrival and departure times can be fairly accurate, trains can get stuck or breakdown at a station. Therefore, we need to analyze blocking networks of not only one single train line, but also look at the yards of other train lines in the blocking network that might be affected.

4.3 Circle Line

Lastly, we will look at the Circle Line and consider the train schedule problem. Particularly, we are going to focus on the train route and see if the Circle Line was designed such that it is the quickest route it could be (distance wise) and the most time efficient. Figure 4.7 is a graph of the Circle Line, in which there are 36 stations (36 vertices to represent each one) and 36 tracks (36 edges to represent each one), in which the tracks can be traveled upon in either direction. Additionally, the Circle Line is a total of 17 miles. [2]

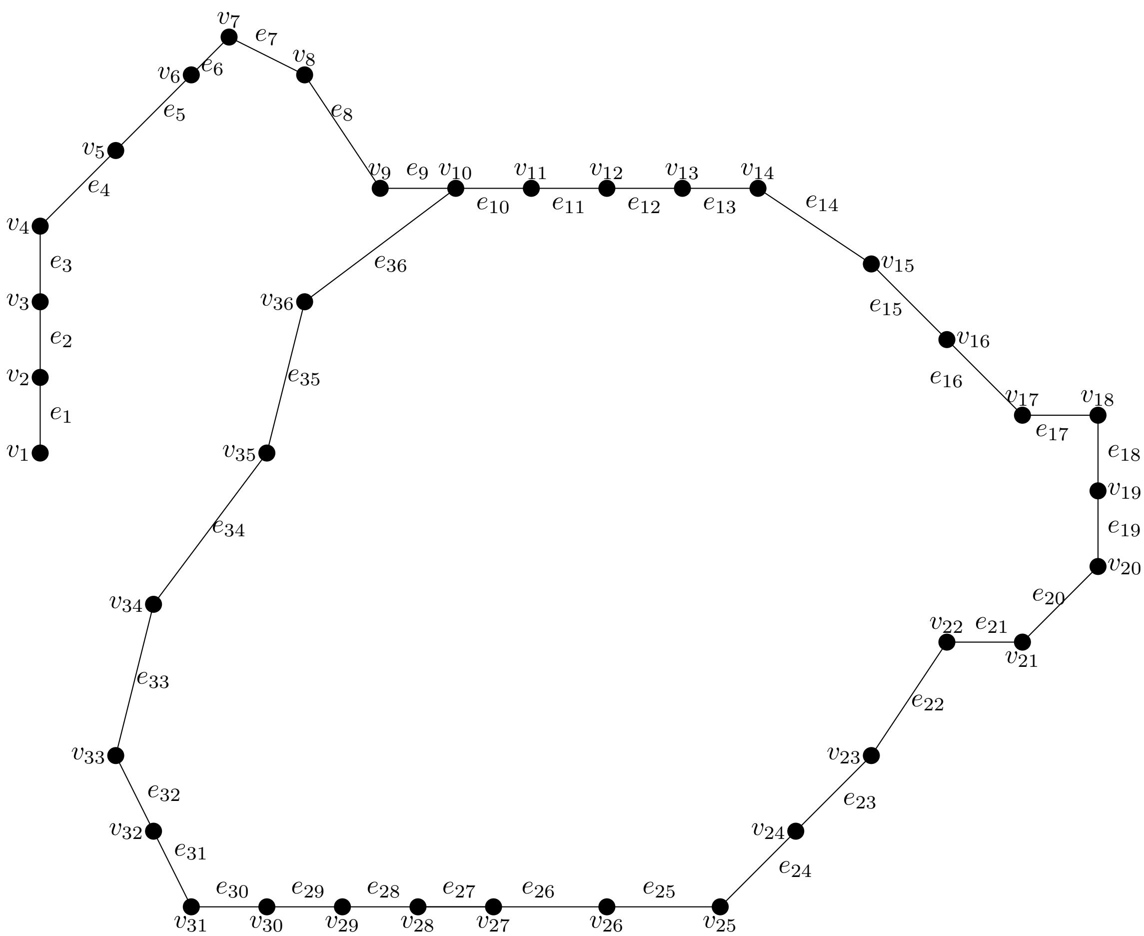


Figure 4.7

In order to visit every station at least once, the train travels in a "spiral shape," in which the train travels from v_1 to v_{10}, where every other station (v_2, v_3, ..., v_{35}, v_{36}) is visited. Note that in order for this to occur, the train travels to v_{10} twice. Therefore, we can see that there is a walk:

$$v_1 e_1 v_2 e_2 v_3 e_3 v_4 e_4 v_5 e_5 v_6 e_6 v_7 e_7 v_8 e_8 v_9 e_9 v_{10} e_{10} \ldots v_{35} e_{35} v_{36} e_{36} v_{10}$$

Prior to 2009, though, trains in the Circle Line traveled in a circle (hence, the name). The railroad system was then redesigned to add another line of stops prior to Edgeware Road. This addition added about 3 miles of additional railroad. [3] Additionally, the Circle Line traveled in both directions (there are multiple trains that travel on the line), but for the purpose of this paper, we will only be focusing on the clockwise direction (from Edgeware Road towards Paddington, where we end at Edgeware Road). Please refer to Figure 4.8 to see a map of what the line originally looked like and the stations.

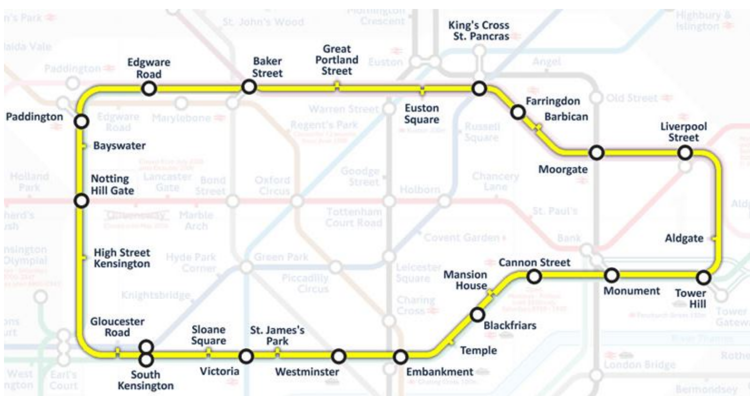


Figure 4.8 [6]

If we refer back to Figure 4.7, the original Circle Line would have only included vertices v_{10} to v_{36} and edges, e_{10} to e_{36}, therefore making the whole train route a circuit, specifically an Euler circuit, as every station was visited at least once and every edge was visited exactly once. The Euler circuit in the original Circle Line was as follows:

$$v_{10} e_{10} v_{11} e_{11} v_{12} e_{12} \ldots v_{34} e_{34} v_{35} e_{35} v_{36} e_{36} v_{10}$$

Looking at the train schedule problem, was the addition of 9 stations and 9 lines of rail, a good decision, such that the time and distance traveled is better and faster than what it was before? It might seem better, as if someone wanted to travel from the Hammersmith station (v_1 in Figure 4.7), she could easily travel to the Bayswater station (v_{34} in Figure 4.7), a route which was impossible to travel on the previous design of the Circle Line. But, in terms of time and distance, the answer is no. In the current design of the Circle, it takes approximately 1 hour and 11 minutes to travel to every station at least once, where we start at v_1 and end at v_{10}. [10] Specifically, it takes 15 minutes to travel the following walk:

$$v_1 e_1 v_2 e_2 v_3 e_3 v_4 e_4 v_5 e_5 v_6 e_6 v_7 e_7 v_8 e_8 v_9 e_9 v_{10},$$

which was the addition to railroad system. Fifteen minutes does not seem like a whole lot in an hour and 11 minutes of a whole train route, but this new addition makes the route much longer, since the train now travels in a "spiral," which graphically is a walk, compared to the original "circle" route, which was a whole circuit. By keeping the original "circle" design in the Circle Line, the line would be more efficient, and it would not take as long to travel to other stations. Yes, if one is traveling from $v_1, v_2, v_3, v_4, v_5, v_6, v_7, v_8$, or v_9 to anywhere else in the original "circle" route, it may seem like a good addition, but the main source of traffic on this line comes from the original Circle Line. Also, if one wanted to travel from v_7 to v_1, she would have to ride the whole train route (the spiral), which would waste more time. The whole new addition is also only about 3 miles, where in total the Circle Line is 17 miles (including the addition). Therefore, the original line was only about 13 miles, and the addition of 3 miles can be considered minuscule, as it does not make the railroad system any faster; it only makes it slower. Thus, the original Circle Line, without the addition, was a better design than the current design.

Chapter 5

Conclusion

Applying graph theory to the railroad system is an efficient tool when designing railroad system, especially in terms of safety. By using the basic properties of graph theory, we were able to apply these properties and create three graphs of train lines in the London Underground, specifically the Central Line, Piccadilly Line, and Circle Line. In these three lines, we were then able to apply three of the design and safety problems (the blocking problem, the yard location problem, and the train schedule problem). Then, we used more advance techniques, such as creating blocking networks of the trains and other train yards that connect into the train lines to analyze possible safety issues, such as train crashes and stopping. Additionally, we were able to analyze updated train designs and see if they were actually more efficient than they previously were. In our case of analyzing the Circle Line, we saw that the addition of 9 stations, and therefore a new train route, did not in fact make the train route any more efficient, and in some respect it can be said to be much worse. Overall, the basic and complex properties in graph theory can be utilized and analyzed for design and safety purposes in the railroad system.

References

[1] Central line overview. https://www.london-tube-map.info/central-line/.

[2] Circle line overview. https://www.london-tube-map.info/circle-line/.

[3] How to get from hammersmith to edgware road underground station. https://www.rome2rio.com/s/Hammersmith/Edgware-Road-Underground-Station.

[4] Jolyon Attwooll. 150 london underground facts. https://www.telegraph.co.uk/travel/destinations/europe/united-kingdom/england/london/articles/London-Underground-150-fascinating-Tube-facts, January 2017.

[5] Sanjay Kumar Bisen. Graph theory use in transportation problems and railway networks. *International Journal of Science and Research*, 6(5):1764–1768, 2017.

[6] BritEvents. A day out on the circle line. http://www.britevents.com/features/a-day-out-on-the-circle-line/.

[7] Gabril C. Caimi. Algorithmic decision support for train scheduling in a large and highly utilised railway network. *ETH Zürich: Research Collections*, 2009.

[8] Sybil Derrible and Christopher Kennedy. Applications of graph theory and network sciences to transit network design. *Routledge*, 2011.

[9] Susanna S. Epp. *Discrete Mathematics with Applications*. Cengage, Boston, 2020.

[10] Transport for London. Timetables. https://tfl.gov.uk/travel-information/timetables/.

[11] Transport for London. Tube map, December 2014.

[12] Transport for London. Visit london: Official visitor guide. https://www.visitlondon.com/traveller-information/getting-around-london/london-maps-and-guides/free-london-travel-maps, December 2017.

[13] Terry A. Loring. How to find an euler circuit.

[14] Joseph Malkevitch and Walter Meyer. *Graphs, Models, and Finite Mathematics*. Prentice Hall, Englewood Cliffs, 1974.

[15] Pablo Arboleya Manuel Coto and Cristina González-Morán. On the use of graph theory for railway power supply systems characterization.

[16] UCB Mathematics. *Chapter 10 Section 5*. Berkeley.

[17] London Transport Museum. The picadilly line. https://www.ltmuseum.co.uk/collections/stories/transport/piccadilly-line.

[18] Cladio B. Cunha Ravindra K. Ahuja and Giivenç Şahin. Network models in railroad planning and scheduling. *Informs*, 2005.