Gotta Graph 'Em All: Graph Theory and Roaming Pokemon in *Pokemon SoulSilver*

Stella Lei, Lisa Lin

December 12, 2022

Cover Letter

Math 22A Final Project

We worked together on the project, with both of us working through the core concepts separately to understand graph theory in our own ways to craft an understandable explanation for a varied audience. When writing the project, Stella worked on fleshing out the mathematical proofs while Lisa focused on structure, readability, and ensuring that the topic was approachable from any angle. We couldn't have done this project without the advice of Eunice Sukarto, our TF, and from our peer reviewers who provided insight on how a 22A student may understand our project. The majority of our peer reviewers urged us to provide further explanations on our proofs, especially Theorem 1.2.1, clearer transitions, and intuitive explanations. To correct this, we added examples and more direct relationships between the proofs and why we need them to solve the problem.

While writing our definitions section and brainstorming relevant examples, we ran into an issue: there's no way we could dive fully into the electrical engineering concepts that come with EE. The effective research papers were focused on timing for different electrical circuit graphs, which we found was similar to the way that our map was structured. The methodology would take around 5-6 proofs to lay down the core concepts and they mostly rely on concepts that may be unfamiliar to 22A students who haven't taken electricity and magnetism. On Eunice's advice, we attempted to make it as digestible as possible and view things through the lens of linear algebra. We would like to emphasize to the reader that it's acceptable to look up more confusing theorems that may not necessarily relate to graph theory, our main focus.

Another problem we faced was how neither of us were too familiar with LaTeX. Eunice and peer reviewers pointed out some corrections that we could make to our notation that we did our best to implement. We also decided to make this topic a more approachable one with formal definitions and explanations for the more mathematical concepts while tying it in with the lightheartedness that Pokémon brings for us. Plus, working with one of our favorite games on a unique project that's applicable to Computer Science, something we're both interested in concentrating in, was a great experience!

Thank you so much for taking the time to read our paper, and we hope that a mathematical explanation of a childhood game brings the joy to others that it did to us. It expanded our view on seemingly basic concepts and ingrained a deeper appreciation on the thought placed into the games! Please let either of us know if there are any questions.

1 Introduction

Pokémon HeartGold and SoulSilver, initially released in Japan in 2009, was the highly anticipated Generation IV remake of its Generation II counterpart: *Pokémon Gold and Silver.* Conveniently, it's also our favorite Pokémon game, so we spent time discussing the game and its features when compared with Generation II. While we discovered that there aren't many major plot differences between the game besides the HeartGold/SoulSilver series including the Kanto region for postgame exploration, these games kept a core unique concept that players have been attempting to crack: roaming Pokémon.

Typically, there is a list of all species you can find per route depending on the time of day, use of items, and more. For instance, on Route 29 in *Pokémon SoulSilver*, you can find Pidgey, Sentret, and Rattata in the morning, but only Hoothoot and Rattata at night. Around 99% of Pokémon follow this pattern, with the only exceptions being static legendary Pokémon, events, headbutt (where you use the move 'Headbutt' on a tree to shake out a Pokémon), and traded ones. The night and day cycle follows for all of other routes in the game except for the rare legendary roaming Pokémon: Raikou and Entei. These legendary dogs wander around every Route, and you can track them using the Pokégear, a multi-function tool your mother provides at the start of your journey.

The process of capturing a roaming Pokémon is an arduous one: you chase them across the map, slowly applying status conditions such as burn or poison to weaken them, lowering their health, and throwing Pokéballs over and over again until it finally stays in the ball. We will let this mechanic be our guiding question in our paper: **Is there an optimal strategy to catch the roaming Pokémon in *SoulSilver*?** Since both *HeartGold* and *SoulSilver* have the same mechanic, we decided to consolidate our project down to only *SoulSilver* since we both played that version. We will first elaborate on primary topics necessary to understand our methodologies, including a basic introduction to graph theory (adjacency and Laplacian matrices, random walks and Markov Chains), and effective resistance, a topic found in electrical engineering. The core concepts of graph theory, namely Markov Chains, relate back to 22A topics such as eigenvectors and eigenvalues, and a general understanding of linear algebra conepts is essential. We have provided relevant examples and proofs to ensure that a Math 22A student can follow, but the reader may find it useful to research into the electrical engineering concepts when tackling effective resistance. Next, we elaborate on how we conducted our calculations utilizing these concepts and their potential applications to other Pokémon / pursuit-evasion games. Thank you for reading our paper, and let's get started!

2 Graph Theory and Linear Algebra

Before investigating the video game Pokémon SoulSilver in terms of graph theory, we will provide some basic definitions and consider relevant ways in which graph theory relates to linear algebra.

The first question that arrives when approaching this problem may be this: **How do we represent the game's route system?** The answer is pretty obvious when viewing the 20 different routes and how they're connected in terms of only lines and points. This condenses down the city names and flashy graphics to something similar to a series of relationships between routes, including intersections. This brings us to the study of Graph Theory: the study of structures, or graphs, that model pairwise relationships between objects. We want to utilize this to strip down the Johto map from a mess of cities, towns, and routes to a simpler graph and utilize math to see the dogs' paths.

2.1 Graph Theory Overview and Definitions

A **graph** is an ordered pair $G = (V, E)$ of sets such that $E \subseteq V^2$. Within this graph, elements of V are called **vertices**, and elements of E are called **edges**. The **order** of G is defined by $|V(G)|$, that is, the number of vertices, whereas the **size** is defined by $|E(G)|$, that is, the number of edges. The degree $d_G(v) = d(v)$ of a given vertex v is the number of edges $|E(v)|$ through v (Chartrand 1985, 25).

We can visualize a graph by representing vertices with dots and edges as lines, as depicted in Figure 2.1.1, located on the next page. From this figure, we can see that graph G is a graph on $V = 1, ..., 6$

with edge set $E = (1,2),(1,3),(3,4),(4,5)$. The order of G is 6, and its size is 4. Additionally, we see that the degree $d(1)$ of vertex 1 is 2, $d(2) = 1$, $d(3) = 2$, $d(4) = 1$, $d(5) = 1$, and $d(6) = 0$.

Two vertices v_1, v_2 of a given graph G are *adjacent* if v_1 to v_2 is an edge of G. Similarly, two edges $e \neq f$ are adjacent if they have a vertex in common. A graph G is considered **complete** if all its vertices are pairwise adjacent. Conversely, a set of vertices or edges is **independent** if no two of its elements are adjacent (Chartrand 1985, 26).

We can define a **path** through G as a non-empty graph $G = (V, E)$ of the form

$$V = v_1, v_2, ...v_k, E = e_1e_2, e_2e_3, ..., e_{k-1}e_k$$

A graph G is **connected** if it is non-empty and any two of its vertices are linked by a path (Chartrand 1985, 53).

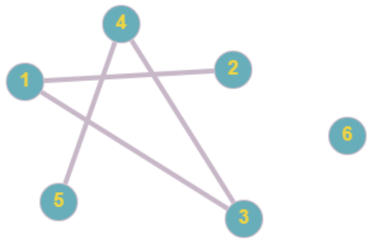


Figure 2.1.1: Graph G

Knowing this, we can simplify the map of the game so that cities and towns become vertices and the numbered routes connecting them become edges:

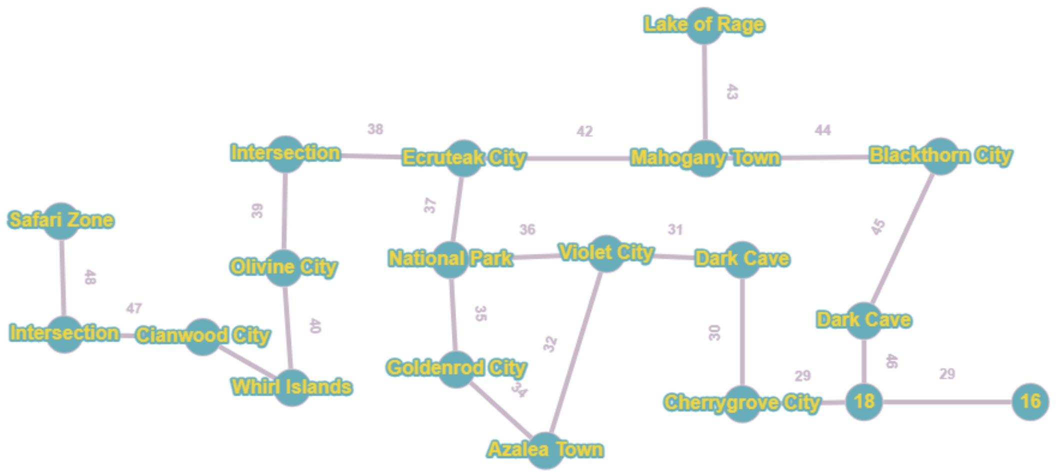


Figure 2.1.2: Graphical Representation of the Johto Region

On the edges of the graph, wild Pokémon appear specifically on tiles designated as tall grass, water, or caves. Every time the player moves, there is a chance of the game to randomly generate a Pokémon depending on time of day, items, and other factors. There is a random set of Pokémon for each route on each tile, each with their own percentage chance of appearing. Most Pokémon appear on a specific set of edges at a specific frequency. For example, in Johto, Abra has a 1/10 chance of appearing when traversing through the grass on Routes 34 and 35.

However, roaming Pokémon are distinguished by their ability to move around the entire graph instead of staying limited to specific edges. In *SoulSilver*, Raikou and Entei appear in the Johto region, while Latios and Latias in the Kanto region. All of these Pokémon follow the same movement rules (Bulbapedia):

- The Pokémon will attempt to move to a different route every time the player changes "map," or moves to a new area.
- There is a 1/(8N) chance that it will move to a completely random area of the map, excluding the one the player is on, where N is the number of adjacent routes to the one the Pokémon is at. Otherwise, it will move to an adjacent route.
- There is a 1/16 change that currently roaming Pokémon will move after each battle

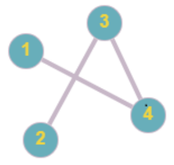
Because there is always a 1/(8N) chance that the Pokémon will move to a random location, we can disregard this in our calculations, as it is something that is unchangeable. Instead, we'll focus on the other section, where there is a chance that the Pokémon will move to an adjacent route. In addition, we won't consider the chance for it to move after battling, since typically, if a player is only hunting for the roaming Pokémon, they won't attempt to catch other ones between the routes for the sake of time. This brings us to our next question: **What route is the best, given these rules, to maximize the chance we encounter either Raikou or Entei?**

In the next two sections 2.2 and 2.3, we'll discuss two important concepts to understand before diving into our calculations. Then ,we'll introduce how to analyze these in the lens of randomness in 2.4, and with number of steps in 2.5.

2.2 Adjacency Matrices

One of the ways graph theory connects to linear algebra is through the ability to represent graphs and specific aspects of graphs as matrices.

If we let G be a graph of order n with vertices $v_1, v_2, ..., v_p$, then the adjacency matrix $A = A(G) = [a_{ij}]$ is the $n \times n$ matrix in which $a_ij = 1$ if v_i and v_j are adjacent and $a_{ij} = 0$ otherwise (Chartrand 1985, 228). An example of a graph and its adjacency matrix are given in Figure 2.2.1.



$$\mathrm{A} = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 \end{bmatrix}$$

Figure 2.2.1: Graph G and Adjacency Matrix A

Looking at the adjacency matrix A, we can observe that each row sum is equivalent to the degree of the corresponding vertex that the row represents.

Furthermore, we can define the **length** of a path as the number of occurrences of edges in it. Two paths v_1 to v_2 of the same length are equivalent if their edges occur in the exact same order. The **distance** $d(i, j)$ between vertices v_1 and v_2 of a graph G is the path of minimum length between the vertices (Chartrand 1985, 229). From these definitions, we can observe the following:

Theorem 2.2.1. Let G be a graph with adjacency matrix A and let k be a positive integer. Then the matrix power A^k gives the matrix where A_{ij} counts the the number of paths of length k between vertices v_i and v_j (Chartrand 1985, 230).

Proof: We prove this theorem by induction on k. We know from the explanation above that the number of walks from i to j of length 1 is equal to 1 if and only if $i, j \in E$. If not, then the number of walks is 0. Thus, A_{ij} is equal to the number of walks of length l from i to j.

Inductive Step: We know that for where A is a $n \times n$ matrix:

$$\sum_{p=1}^{n} A_{ip}^{k-1} A_{pg} = A_{ij}^{k}$$

By examining the summation visually, we can see that every path of length k from i to j will cross the point adjacent to k after traversing $k-1$ number of edges. Thus, every walk of length is equivalent to a walk of length $k-1$ from i to some neighbor p of j when along the edge (p, j) By the same logic, every walk of $k-1$ length from i to p can be extended to a walk of length k from i to j by adding the edge (p, j).

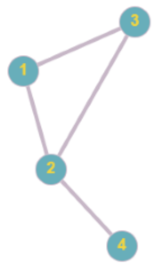
By the inductive hypothesis, A_{ip}^{k-1} and the equation above, we can infer that A_{ip}^{k} will count the number of walks from i to p of length $k-1$ for all p such that $(p, j) \in E$, which is the same as the number of walks from i to j of length k. Thus, the theorem holds. □

Adjacency matrices are important to have a mathematical representation of the layout of Johto. However,in roaming mechanics, the dogs' paths can be affected by the number of routes connecting the route they're on. This is where Laplacian matrices come in.

2.3 Laplacian Matrices

While adjacency matrices can be useful for converting more simple representations of graphs to matrices, Johto is extremely complex. To account for this, we must utilize Laplacian matrices, since they unlike Adjacency matrices, they can represent the relationship and connections between vertexes. It contains more information about the structure of the graph; it includes information on both edge weights and vertices.

If we let the d_i denote the degree of a vertex v_i, the Laplacian Matrix of a graph G, denoted $L(G) = [l_{ij}]$, is the $n \times n$ matrix in which $p_{ij} = -1$ if v_i is adjacent to v_j, $p_{ij} = 0$ if v_i is not adjacent to v_j, and $p_{ij} = d_i$ if $i = j$ (Johnson 2017). An example of a graph G and its Laplacian matrix are given in Figure 2.3.1. Looking at this matrix, we observe that the degrees of the graph's vertices appear along the diagonal.



$$L = \begin{bmatrix} 2 & -1 & -1 & 0 \\ -1 & 3 & -1 & -1 \\ -1 & -1 & 2 & 0 \\ 0 & 0 & -1 & 1 \end{bmatrix}$$

Figure 2.3.1: Graph G and Laplacian Matrix L

Furthermore, if we define $D(G)$ to be a diagonal matrix whose entries are the degrees of each vertex, then the Laplacian Matrix of G can be equivalently defined as $L(G) = D(G) - A(G)$

2.4 Random Walks and Markov Chains

Now that we have an understanding of paths and how to analyze our map using matrices, we can extend these ideas to the randomness of roaming Pokémon. There is always a random chance for the dogs to move – this can be represented by a random walk - a concept that can be thought as an person taking a sequence of steps, with the direction of each step chosen randomly.

A **random walk** on a graph consists of a series of vertices generated from a starting vertex by selecting an edge, traversing the edge to a new vertex, and repeating the process. For undirected graphs specifically, these random walks have connections to electrical networks. Each edge has a **conductance**, parallel to an electrical conductance. If the walk begins at a vertex u, it selects an edge from all edges incident to u to walk to the next vertex with a probability proportional to that edge's conductance. A **simple random walk** is a random walk in which from any vertex, there is an equal probability of moving to an adjacent vertex (Hopcroft and Kannan 2014, 153).

We can visualize this concept with the easiest example: a 1-D line. Suppose Raikou was on this line graph at 0. On each step, Raikou has a 50% chance of taking a step right (increasing the coordinate by 1) and a 50% chance of taking a step left (decreasing the coordinate by 1).

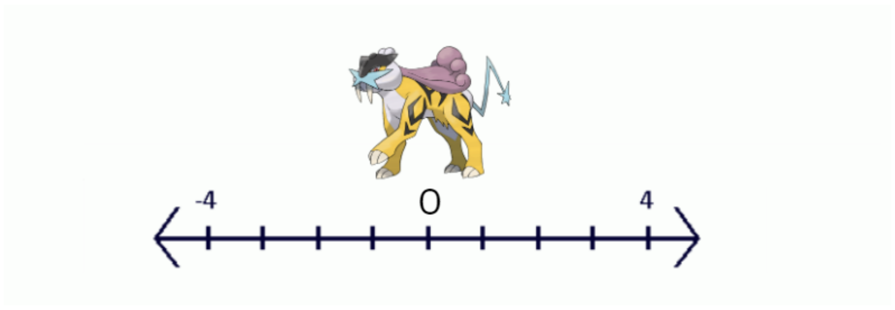


Figure 2.4.1: Random Walk on 1-D Line

After one step, Raikou would either be at -1 or 1. After two steps, Raikou could be at -2, 0, or 1, all of which with equal probabilities. Continued infinitely, the path taken by Raikou is our random walk!

An equivalent concept to random walks is called a **Markov chain**, which starts at a state x, and for each time step, selects the next state y with a probability p_{xy}. The matrix P consisting of these p_{xy} is called the **transition probability matrix** of the chain. A Markov chain is **connected** if its underlying graph has a directed path from every vertex to every other vertex (Hopcroft and Kannan 2014, 154).

Interestingly, this intersection with Markov Chains applies to the topics we learned towards to the end of 22A: eigenvectors and eigenvalues. To demonstrate this, we'll use a simpler example with the roaming Pokémon: Raikou. (Normalized Nerd, 2020)

Suppose in our hypothetical example, Raikou can only travel between three routes. It will only move to the new route solely based on its present position, and the probability of all of its actions are shown in Figure 2.4.1.

Each arrow shows a transition from one state to another, and the future state depends on the previous state. This is called the **Markov Property**. This can be demonstrated as a equation: $P(X_{n+1} = X|X_n = x_n$. We can utilize this property to find the overall probability of Raikou going to every route if it took an infinite number of walks. Let's start with a smaller scale example:

Let's say Raikou takes 10 walks through Route 1, 2, and 3, denoted as R_1, R_2, and R_3 respectively.

$$R_1 \rightarrow R_2 \rightarrow R_1 \rightarrow R_3 \rightarrow R_1 \rightarrow R_3 \rightarrow R_3 \rightarrow R_3 \rightarrow R_1 \rightarrow R_2$$

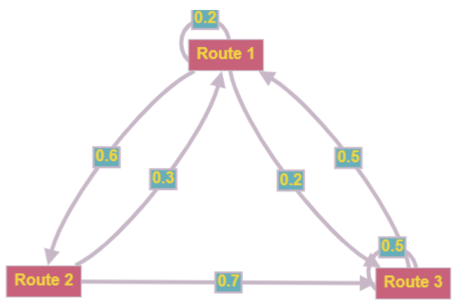


Figure 2.4.2: Theoretical Rakiou Roaming Route

To find the probability of R_1, R_2, and R_3, we can simply use fractions: $P(R_1) = \frac{4}{10}, P(R_2) = \frac{2}{10}, P(R_3) = \frac{4}{10}$

While this may give us a general understanding of the probability of Raikou appearing on every route, we want to do this an infinite number of times. Theoretically, after running an infinite number of steps, the probability will reach an equilibrium state, called the **stationary distribution** that won't change with time. Luckily, we can find this utilizing what we've learned in 22A: matrices!

First, we'll make a **directed graph**, a matrix where the elements denote the weight connecting to the corresponding vertices. For instance, the element on row 1, column denotes the probability of Raikou returning to R_1 after making a walk. If the element is 0, then there is no edge. We'll call this **transition matrix** A.

$$A = \begin{bmatrix} 0.1 & 0.6 & 0.2 \\ 0.3 & 0 & 0.7 \\ 0.5 & 0 & 0.5 \end{bmatrix}$$

Our goal now is to find the probability of each state occurring in a matrix form. This vector is denoted as π. Suppose Raikou is currently located on R_2. This means $\pi_0 = \begin{bmatrix} 0 & 1 & 0 \end{bmatrix}$. By repeatedly multiplying this row vector by the transition matrix A, we can find the $P(R_2)$.

$$\pi_0 A = \begin{bmatrix} 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} 0.1 & 0.6 & 0.2 \\ 0.3 & 0 & 0.7 \\ 0.5 & 0 & 0.5 \end{bmatrix} = \begin{bmatrix} 0.3 & 0 & 0.7 \end{bmatrix}$$

Then, we can take the output and replace π_0 with it:

$$\pi_0 A = \begin{bmatrix} 0.3 & 0 & 0.7 \end{bmatrix} \begin{bmatrix} 0.1 & 0.6 & 0.2 \\ 0.3 & 0 & 0.7 \\ 0.5 & 0 & 0.5 \end{bmatrix} = \begin{bmatrix} 0.41 & 0.18 & 0.41 \end{bmatrix}$$

We won't go through the remainder of the calculations, but the process is the same. At a certain point, if there exists a stationary state, the output vector will be equal to the input vector. This vector is denoted as π. The equation, called the **Fundamental Theorem of Markov Chains**, and proved later in this paper, is extremely reminiscent of our eigenvector equation:

$$\pi A = \pi$$

$$Av = \lambda v$$

Because of this, we can imagine π as a left eigenvector of matrix A with eigenvalues equal to 1. Recall once again that the elements of π must sum to 1 because of the probability distribution. With this information in mind, we can solve the problem:

$$\pi[1] + \pi[2] + \pi[3] = 1$$

$$\pi = \begin{bmatrix} \frac{25}{71} & \frac{15}{71} & \frac{31}{71} \end{bmatrix}$$

Now, we know that in our hypothetical, Raikou has an approximately 35% chance of appearing on Route 1, 21% on Route 2, and 44% on Route 3. While our actual problem is much more complex, this is a good starting point to begin our calculations.

Before proving the Fundamental Theorem of Markov Chains, we must prove a technical lemma (Hopcroft and Kannan 2014, 157):

Lemma 2.4.1. Let P be the transition probability matrix for a connected Markov chain. The $n \times (n+1)$ matrix A $= [P - I, 1]$, obtained by augmenting the matrix $P - I$ with an additional column of 1s, has a rank of n.

Proof by Contradiction: Given the rank of A being equal to $[P - I, 1]$, if this rank was less than n, there would exist 2 linearly independent solutions to $Ax = 0$. Every row in matrix P will sum to 1, thus, every row in $P - I$ will have a sum of 0. Thus, $x = (1, 0)$, where all elements except for the last coordinate of x is 1. This is one solution to $Ax = 0$.

Assume there exists a second solution (x, b) perpendicular to (1,0). This means that $(P - I)x + b = 0$. From this equation, we obtain the summation equation:

$$\sum_j p_{ij} x_j = x_i$$

From this, we can see that every value x_i is a convex combination of some $x_j + b$. In other words, the linear combination of the points are all non-negative and have a sum of 1. Let S be the set of i where x_i attains its maximum possible value. $\overline{S}$ is not empty since we defined earlier that x is perpendicular to 1. Thus:

$$\sum_j x_j = 0$$

Since they are all connected, it implies that part of x_k is adjacent to some x_l of a lower value. Therefore,

$$\sum_j p_{kj} x_j < x_k$$

Then, $b > 0$ when b is added to the summation and set equal to x_k. A symmetric argument with T, the set of i with x_i with its minimum value implies b ¡ 0, producing a contradiction. This proves the lemma. □

Now, we have the technical tools to prove the Fundamental Theorem:

Theorem 2.4.2. (Fundamental Theorem of Markov Chains) For a connected Markov chain, there exists a unique probability vector π satisfying $\pi P = \pi$. Moreover, for any starting distribution, $\lim_{x \to \infty} a^t$ exists and equals π (Hopcroft and Kannan 2014, 158).

Proof: Since A^t has non-negative components and sum to 1, it is a probability vector. Now, we run one step of the Markov chain with this distribution. Afterwards, the distribution will follow $a^t P$.

$$\begin{aligned} a^t P - a^t &= \frac{1}{t}[p^0 P + p^1 P + ... + p^{(t-1)} - \frac{1}{t}[p^0 + p^t + ... + p^{(t-1)} \\ &= \frac{1}{t}[p^1 + p^2 + ... + p^t] - \frac{1}{t}[p^0 + p^1 + ... p^{(t-1)}] \\ &= \frac{1}{t}(p^t - p^0). \end{aligned}$$

Thus, $b^t = a^t P - a^t$ satisfies $|b^t| \leq \frac{2}{t} \to 0$, as $t \to \infty$. By the Lemma 2.4.1. Given an $n \times n$ matrix A with rank n, the $n \times n$ sub-matrix B of A consisting of all columns except the first columns is inevitable. Let c^t be obtained from from b^t by removing the first entry. Then, $a^t B = [c^t, 1]$ and so $a^t = [c^t, 1]B^{-1} \to [0, 1]B^{-1}$. Thus, we have the theorem with $\pi = [0, 1]B^{-1}$ (Chekuri 2020). □

2.5 Effective Resistance

To determine the most efficient path, we've chosen to minimize the number of walking steps the player needs to take to encounter the roaming Pokémon. To do this, we will utilize an idea from electrical engineering called **effective resistance**. In electrical engineering, effective resistance describes the resistance between two points in an electrical network. In terms of graph theory, the effective resistance between two vertices x and y, R_{xy}, is the expected number of traversals out of x along any specific edge in a random walk starting at x and ending at y (Hopcroft and Kannan 2014, 159).

We can use Laplacian matrices to calculate the value of effective resistance. The Laplacian matrix L for a connected graph G has all positive eigenvalues except one that is 0. The eigenvector ϕ associated with the 0-eigenvalue has all its components the same, which we can take to be 1. Thus, L has a generalized inverse Γ that can be calculated as $\Gamma = L + U/n$ where U is the unit $n \times n$ matrix (Klein and Radic 2002).

Using this generated inverse, effective resistance can be calculated as follows:

$$R_x y - \Gamma_x x - \Gamma_{xy} - \Gamma_{yx} + \Gamma_{yy}$$

This formula can be formally derived using an electrical engineering concept known as Kirchoff's laws and is long-known in electrical engineering literature. Thus, for the purposes of this paper, we can accept this formula as a definition of effective resistance.

For this problem, we want to use effective resistance to determine $E[H_{xy}]$, the expected transit time from vertex x to y. To do so, we will use the following theorem (Tetali 1991).

Theorem 2.5.1. $E[H_{xy}] = (1/2)\sum_z d(x)[R_{xy} + R_{yz} - R_{xz}]$

Proof: The proof of this theorem requires several other theorems, which will be described below but not proved within this paper for the sake of space. To see the full proofs of these theorems, please refer to (Tetali 1991).

Theorem 2.5.2. $R_x y$ is the expected number of traversals out of w ($\neq x, y$) along any specific edge (w, z) in a simple random walk from x to y and then back to x.

Theorem 2.5.3. The expected number of traversals out of w along a specific edge during a random walk from x to y is the same as the expected number of traversals out of x along a specific edge in a random walk from w to y.

From Theorems 2.5.2 and 2.5.3, we obtain Theorem 2.5.4.

Theorem 2.5.4. (The Triangle Inequality for Effective Resistance) Any three vertices, x, y, z of a network satisfy

$$R_{xz} + R_{zy} - R_{xy} = \frac{U_x^{yz}}{d(x)} + \frac{U_y^{xz}}{d(y)}$$

Now, we are ready to prove the main theorem:

Proof: By Theorem 2.5.4, we have

$$R_{xy} + R_{yz} - R_{xz} = 2\frac{U_z^{xy}}{d(z)}$$

This then can be expressed as

$$U_z^{xy} = (1/2)d(z)[R_{xy} + R_{yz} - R_{xz}]$$

Summing over z then gives

$$\sum_z U_z^{xy} = (1/2)\sum_z d(z)[R_{xy} + R_{yz} - R_{xz}]$$

Then we have $E[H_{xy}] = (1/2)\sum_z d(x)[R_{xy} + R_{yz} - R_{xz}]$, and the theorem is proved. $\square$

3 Tracking the Legendary Dogs

Now that we have a general understanding of the rules Raikou and Entei follow, a mathematical representation of the region's map, and an introduction the mathematical tools used to calculate their path, it's time to put them into action. The methodology of this section is based on the work of Ross Churchley.

We understand that in addition to their regular movement, the roaming Pokémon have a 1/(8N) chance of escaping and moving anywhere on the graph. Thus, determining the best player movement to catch them is less so about finding the vertex that the roaming Pokémon return to fastest, and moreso about trying to minimize

$$\frac{1}{|V(G)|}\sum_u E[H_{xy}]$$

where $E[H_{xy}]$ is the expected transit time from x to y, or the expected time for a random walk starting at x to first reach the vertex y. Dividing this value by the number of vertices in the graph allows the formula to take an average of how much time it would take to reach the vertex y from any vertex on the graph.

We can compute $E[H_{xy}]$ with the formula Tetali provides in Theorem 2.5.1:

$$E[H_{xy}] = (1/2)\sum_z d(x)[R_{xy} + R_{yz} - R_{xz}]$$

The effective resistance can be solved using the formula $R_x y - \Gamma_x x - \Gamma_{xy} - \Gamma_{yx} + \Gamma_{yy}$

Crucially, we notice that the roaming Pokémon cannot appear at vertices of the graph, only on their edges. Thus, to minimize the time it takes to encounter a roaming Pokémon, it should be fastest to move between two edges, where both steps provide a chance for an encounter, instead of between an edge and a vertex. There are four pairs of edges in the Johto region where this type of movement is possible: 30-31, 35-36, 36-37, 45-46. Thus, when applying our calculations, we want to merge the vertices $(x, 0)$ and $(y, 1)$ corresponding to the player's location after an even or odd number of steps respectively.

First, we express the graph of the Johto region as the following Laplacian matrix L:

$$L = \begin{bmatrix}
1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
-1 & 3 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & -1 & 2 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & -1 & 2 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & -1 & 2 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & -1 & 2 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & -1 & 2 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 3 & -1 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 2 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 2 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & -1 & 3 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 3 & -1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 2 & -1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 2 & -1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 2 & -1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 2 & -1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 2 & -1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 1
\end{bmatrix}$$

We can then use this matrix to complete the effective resistance and expected transit time calculations for the listed edges, yielding the following result:

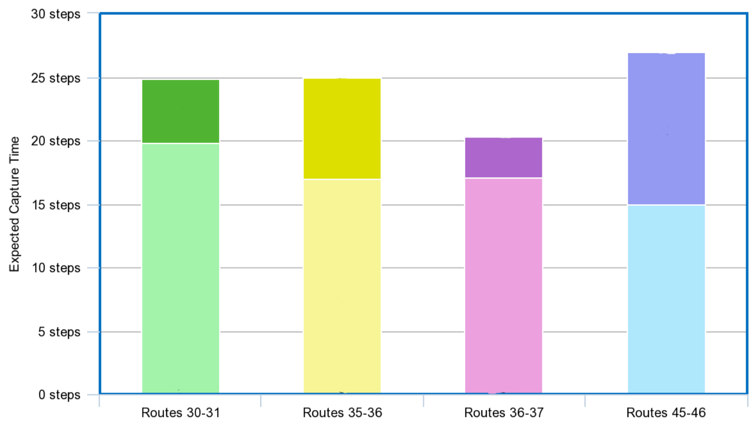


Figure 3.1: Expected capture time when moving between adjacent edges. Each pair of edges has two expected times, shown in different shades, depending on which edge is the starting point

We see that moving between Routes 36 and 37, starting on Route 37, leads to the encounter time with the fewest number of steps. Thus, the player should use this strategy to minimize the amount of time it takes for the user to encounter the roaming Pokémon in *SoulSilver*.

4 Conclusion

Finally, we've reached the answer! Through utilizing graph theory concepts and effective resistance, we've concluded that there is an optimal strategy to catching Raikou and Entei in *Pokémon SoulSilver*. The player should move back and forth between the edges corresponding to Routes 36 and 37, ideally starting on Route 37. To find this, asked ourselves questions while researching:

- **Main Question: Is there an optimal strategy to catch the roaming Legendary Dogs in *SoulSilver*? If so, what is it?**
- How do we create a graphical representation of the Johto map? (2.1, 2.2, 2.3)

- How do we account for random movement? (2.4)
- How do we measure time and efficiency through steps? (2.5)

After breaking down the problem into parts, we found it much easier to consolidate our research. Afterwards, we expressed the map of the game as a graph and subsequently as a Laplacian matrix, from which we computed effective resistances and expected transit times for relevant routes. This methodology should theoretically work for other graph-pursuit games with similar goals pursuer-evader setups, such as the other Pokémon franchise games with similar randomness mechanics (*Pokémon Gold and Silver, Pokemon Black and White*, etc). Once again, thank you for reading our paper, and good luck catching 'em all!

References

Chartrand, G. (1985). Introductory graph theory. New York: Dover.

Chekuri, C. Topics in Graph Algorithms. 2020.

Hopcroft, J., &amp; Kannan, R. (2014).

Johnson, D. (2017).

Klein, D. J., &amp; Randic, M. (1993). Resistance Distance. Journal of Mathematical Chemistry, 12.

Markov Chains Clearly Explained. (2020). YouTube. https://www.youtube.com/watch?v=i3AkTO9HLXo

Roaming pokémon. Roaming Pokémon - Bulbapedia, the community-driven Pokémon encyclopedia. https://bulbapedia.bulbagarden.net/wiki/Roaming˙Pok%C3%A9mon. Accessed 12 December 2022

Ross Churchley. How to catch legendary pokémon fast. How to catch legendary Pokémon fast. https://rosschurchley.com/blog/how-to-catch-legendary-pokemon-with-electrical-networks/. Accessed 12 December 2022

Tetali, P. (1991). Random walks and the effective resistance of networks. Journal of Theoretical Probability, 4(1), 101–109. doi:10.1007/bf01046996