**2 Introduction:**

The Culver Academies is a private boarding school with a long history, established in 1894. Since then, the school has developed and expanded gradually, with new demands and constructions. The road network that has developed in these years is a complex transportation system that is not necessarily developed to fit the needs of students today.

Most of the student body at the Culver Academies had probably been shouted at by adults for crossing the grass rather than taking a detour on the sidewalks. This is a result of frustration caused by the designs of the pathways on campus, feeling the irritation of seeing the destination drifting further away as we walked on the sidewalks. The existing pathway system is not planned for the continued growth of future transportation needs.

       To make everyday path planning more convenient for students, this paper will explore the various methods to generate and optimize pathway networks in the Culver Academies. Designing an efficient path network has been a challenge due to the interdependent system consisting of discrete nodes. The goal of the designer is to construct a pathway system with minimum cost while satisfying traffic needs for members of the Culver Academies.

       The problem of designing a network to meet certain constraints while optimizing specific characteristics draws similarities to many real-life applications in communication networks, water systems, and many other problems that could be modeled by the interactions between discrete individual nodes. This wide range of applications encouraged the development of many approaches. In this paper, I will be investigating the approaches of several agent-based and heuristic algorithms and applying them to the setting of Culver Academies.

**3. Background Research**

3.1 Network Planning

Designing a network is a complicated optimization problem typically formulated as an Integer Linear Programming (ILP) problem with considerations of performance, reliability, and cost. The discrete nature of the problem makes it a challenging problem to solve, often requiring human experts to set certain standards (Zhu et al, 2021). Many of the design processes and algorithms in network planning are developed to other fields of study, primarily in telecommunications.

This type of optimization falls under combinatorial optimization, which is finding an optimal subset of the given set of objects. In this case, the problem is to find the best combination of sidewalks among all possible paths between buildings on Culver Academies by deciding which paths to keep or remove. Non-heuristic optimizers such as linear programming or gradient descent struggle with this discrete nature. Exact methods are also difficult due to their computational complexity. For a set of buildings, the number of possible subsets of paths is , making the problem NP-hard (Schrijver, 2003).

In telecommunications, designing a network starts with designing the topologies with regard to cost. This step outputs the physical network and the locations of intermediate nodes. Then, the traffic will be evaluated to formulate the logical links between nodes (Penttinen, 1999). In this study, there is no difference between the logical and physical, but the process of topological design before optimizing based on traffic can be applied to this problem.

3.2 Graph Theory

A road network is often represented by a graph, with destinations and intersections as vertices, and roads as edges, denoted as . *V* denotes the finite collection of vertices and *E* denotes the finite collection of edges. An edge connecting two vertices could be directed, represented by the ordered pair , or can be non-directed, where it can be represented by a set . While a sidewalk seems to be intuitively non-directed as one can walk in both ways, a directed edge is also useful to represent direction of traffic at certain times.

To better model the geographical distance and traffic capacity of the road network, a weighted graph is used where denotes the graph representing the road network and maps every element in to a real number. In other words, it simply assigns one or more numbers to each edge.

The relationship between nodes can be represented by an adjacency matrix. The adjacency matrix for a graph is an matrix. For the vertices , if , meaning that there exist an edge for and , then , else . For a weighted graph, could also be equal to a weight value. (Griffin, 2023).

3.3 Problem Description:

Given the locations of 30nodes which represents all buildings on campus and the traffic requirements between them, the problem is the determine which of the possible edges to construct, which means choosing from possible topologies (Corne et al., 2000).

The aim of the design should be to satisfy the traffic demands of students and the cost demands presented to the school.

To better fit the problem into a graph theory model, several assumptions are made regarding the graph representation of the path network:

1. The path network is a 2-dimensional model, ignoring all differences in attitude.
2. All paths have the same traffic capacity.
3. Only buildings enclosed by Academy Road, State Road 10, and East Shore Drive will be included in this model.

3.4 Data Collection

The data used in this study is obtained from geojson.io by pinpointing geographical locations of buildings on the Culver campus manually. The result is converted by the website into a geojson file which can be processed in Python. The data contains a total of 30 buildings on the Culver campus. For each building, the converted file contains an index, name, longitude, and latitude that corresponds to it.

图表, 散点图

描述已自动生成

Figure 3.4.1. This figure shows the geographical location (longitude and latitude) of the buildings on Culver Academies, labeled with their names.

Lake Maxincuckee is intentionally ignored in this study as it is a non-hole region in the sample space shown above. This makes the geographical space of Culver Academies homeomorphic with the rectangular space shown above. Since no buildings reside on the Lake, the graph structures are maintained when the Lake is ignored.

**4. Methodology:**

4.1. Slime Mold Algorithm (SMA)

The slime mold algorithm draws inspiration from *Physarum polycephalum*, a type of slime mold that can form biological networks as a part of their foraging strategy to discover and exploit food sources (Tero et al., 2010). These biological networks have been optimized through many cycles of evolution, making it a reasonable approach to design a transportation network.

|  |  |
| --- | --- |
| **Symbol** | **Description** |
|  | Current iteration |
|  | Maximum Pheromone Density for a cell |
|  | Decay parameter |
|  | Current position of a slime cell |
|  | Position of target food |
| MOVE\_TH | Moving threshold for primary diffusion |
| DIFF\_TH | Diffusion threshold for secondary diffusion |
|  | Pheromone level of cell on at iteration |
|  | Direction diffusion |
|  | Diffusion decay rate |
|  | Pheromone level when a slime cell reaches food |

**Table 4.1.1** Notation used in the slime mold algorithm

The slime mold algorithm can effectively model the pathway network by mimicking how slime mold forms efficient nutrient pathways between food sources. In this study, each building, or node, is represented as a food source that attracts slime cells. The direction-based exploration mechanism simulates the natural preference of shorter, direct paths by penalizing longer or redundant paths with pheromone decay. The reinforcement of the paths optimizes the cost of the pathway system.

The slime mold simulation on the Culver building complex will be implemented using Python, using the model described in Zhang, 2022. The simulated slime moldmodel, an lattice graph , where is the set of cells for the lattice. In the simulation, the dimension of the lattice graph is determined by the maximum horizontal and vertical distance between buildings on Culver Academies.

An individual cell contains its location index , its current state, and its pheromone density. Each cell can have one of the following states at a given iteration , represented as .

* 0 - Empty: Cells without any food or slime.
* 1 - Slime: Cells that are considered part of the slime mold, which would represent the paths on the pathway network.
* 2- Food: Cells containing food source, representing the nodes of the pathway network.

In the simulation, an initial slime is placed onto one arbitrary food cell to start. It will then start searching for nearby unconnected food sources. Each slime cell, although viewed together as a single organism, is modeled as an individual agent, making its own decisions based on local inputs. Each slime cell will first gather the shortest path to the nearest food source, and will diffuse, by replicating into neighboring cells, following these four features (Zhang, 2022):

1. Diffusion into another cell decreases that cell’s pheromone density.
2. The slime will prioritize diffusing to a neighboring cell closest to shortest path to the food source.
3. The slime will not diffuse if it is too far from a connected food source, or if the target cell’s pheromone level is too low.
4. The slime’s decision will also be affected by the state of the neighboring cell: food, empty, or slime.

The diffusing mechanism can be categorized into primary and secondary diffusion, where primary diffusion occurs along the shortest path towards the food source, and secondary diffusion occurs when primary diffusion is blocked.

Before diffusion happens, from all 8 possible directions (positive and negative directions along the horizontal, vertical, or diagonal), one or more direction indices is selected if the direction points to a food source. These indices are set as the primary direction for a slime cell.

Diffusion occurs when the pheromone level in the slime cell exceeds the moving threshold ( MOVE\_TH), the slime cell attempts to initiate primary diffusion to the neighboring cell denoted by . If the cell lies within the lattice grid and is empty, the diffusion will begin by setting to a slime cell.

Secondary diffusion occurs when the slime cell cannot diffuse along the primary direction. Secondary diffusion is random and occurs when the distance if its distance to the nearest food is smaller than the diffusion threshold (DIFF\_TH) (Cai et al, 2020). This process is much slower to reach a food source in order to favor primary diffusion, which uses the shortest distance.

For every iteration during diffusion, an amount of pheromone is lost in a slime cell. The pheromone level of the starting cell is updated by this exponential decay function, where is the diffusion decay rate and is the decay parameter:

By diffusing into the neighboring cell , the slime mold deposits an amount of pheromone to the new slime cell since an empty cell has no pheromone. The pheromone level is updated as follows to ensure that the level of pheromone does not exceed the maximum level (Zhang, 2022):

This same function is used when a slime cell has less pheromone than its neighbor to simulate the transfer of nutrients from a food source.

If any slime cell becomes adjacent to a food cell during an iteration, its pheromone level is instantly set to a fixed value . This represents a reward for finding food.

The processes described above will continue until it reaches the maximum iteration set at the beginning of the simulation.

During diffusion, the slime mold system will decay simultaneously based on its distance to a connected food source and pheromone density of the cell it is on. The diffused slimes will deliver nutrients to other food sources and increase pheromone levels in this process. If a slime cell has a pheromone level below a certain point, it will decay into an empty cell. This process ensures that only the most efficient nutrient pathways are kept (Li, et al, 2020).

To model this, the activity of the slime cell is directly tied to its pheromone level so that if no exterior pheromone from the food sources is provided, it cannot move to a new location through diffusion and becoming inactive. Therefore, to determine the best path generated by the slime mold algorithm, I would only have to look at if any two food sources is connected by a continuous track of slime cells with high levels of pheromone.

The slime mold simulation is visualized step by step. Figure 4.1.1 shows the slime mold’s behavior during exploration. Figure 4.1.1b and 4.1.1c shows the diffusion-decay process of SMA.

图表, 散点图

描述已自动生成

(a) 1 step (b) 20 steps (c) 50 steps

Figure 4.1.1. First 50 steps of the SMA on the Culver geographic dataset. Darker orange represents a higher concentration of pheromone.

In the simulation, a trial of pheromone concentrated cells between two food sources will be recognized as edges to the network connecting two nodes.

4.2 Union of Rings Algorithm

Besides the agent-based SMA model, another approach is through heuristic algorithms, such as the Union of Rings Algorithm. This method is described first by Frank and Frisch (1971) and optimized by Blessings (1999) to construct the Minimum Weight Flow-Equivalent Graph (MWFEG). The significance of the MWFEG is that it satisfies the flow requirements under minimum cost, meaning that it is optimal under cost conditions. Another important property of the MWFEG is that it is a biconnected graph, meaning that if any vertex is removed, the graph is still connected (Blessings, 1999).

The algorithm was intended to be applied to telecommunications networks, which prioritizes reliability and puts less importance on geographical proximity. In this study on a transportation network, the algorithm will be adjusted to prioritize distance constraints over reliability.

The union of rings algorithm follows these procedures to construct the MWFEG outlined by Blessings (1999):

1. Draw the requirement matrix that represents the graph as a complete graph where every vertex is connected, and each edge is labeled with the minimum traffic requirement between nodes and . In this study, geographical distance is combined with traffic requirement to form a combined matrix which will be used instead.
2. Construct the maximum spanning tree . A spanning tree is a graph that connects all nodes without cycles. For a maximum spanning tree, the total edge weights (from the combined matrix) are maximized, meaning that paths with the highest traffic demands are prioritized.
3. Convert into a linear flow equivalent graph . The graph is flow-equivalent to the maximum spanning tree , meaning that satisfy the properties that has in terms of traffic requirements while reducing the cost and complexity of the network.
4. Factor into several uniform capacity rings (subgraphs or cycles with equal flow capacity). This step is designed to distribute traffic uniformly within rings, balancing the traffic load across multiple paths.
5. Combine the rings from the previous step with the linear flow graphto create the network topology . The generated topology will be biconnected, meaning that every node in the network is connected by at least two independent paths, providing robustness to the network.
6. Optimize the resulting graph manually to remove insufficient edges or add additional shortcuts.

Step 3 follows these steps to convert the maximum spanning tree to the flow equivalent graph (Blessings, 1999):

* 1. Initialize to be an empty graph and set all nodes in as unmarked.
  2. Select an arbitrary node in , mark it, and add it to . This will be the starting node for .
  3. While contains unmarked nodes:
     1. Identify all edges in that connect a marked node to an unmarked node.
     2. Select the edge with the maximum weight.
     3. Add the selected edge to .
     4. Mark the unmarked node at the other end of the edge and add it to .

The resulting Minimum Weight Flow-Equivalent Graph represents a pathway system in Culver Academies. Due to the properties of the MWFEG, the resulting pathway network will have balanced cost, robustness, and traffic capacity.

The requirement matrix used in this study is a weighted sum of the distance matrix and the requirement matrix . The distance matrix should award higher weight for shorter distance. Therefore, the value on the th row and th column of is calculated by the inverse of the Euclidean distance between the th and th node and multiplied by 100 to have similar degree to the requirement values:

In this study, I will be assuming the requirement matrix which includes the minimum traffic requirements between buildings on campus. The assumption will simply assign traffic requirements by assigning each connection between buildings values 1, 3, 5, 10, 15, and 20, with higher weight meaning more traffic demand. The discrete numbering used to assign traffic requirement is due to the fact that a uniform capacity ring can only be constructed on edges with equal weight.

Given matrices and , the adjacency matrix that will be used is calculated as follows:

Where are corresponding weights to and and is the resulting combined matrix.

Below shows the detailed process of the algorithm when applied to the dataset:

A diagram of a constellation

Description automatically generated

The maximum spanning tree shown above is generated from a combined adjacency matrix acquired by a weighted sum of the distance and requirement matrix (weights = 1, 0.25 correspondingly).



The linear flow graph is then calculated, it is a linear graph that maintains all flow properties of the maximum spanning tree. The above figure shows the linear flow graph labeled with the index of the node.

The uniform capacity rings will be extracted from the linear flow graph, shown below in Figure ().

These rings are imposed onto the maximum spanning tree. Creating a combined graph shown in Figure ().

The uniform capacity rings added to the maximum spanning tree added edges to the linear flow graph. These paths balances traffic and provide shortcuts.

Blessings (1999) mentioned to manually optimize the graph by removing insufficient edges. This procedure is implemented in this study through an autonomous process instead, where edges are removed and modified under certain constraints. An edge is removed if its length exceeds 700 units. Then if the distance of an edge to a node which it does not connect is less than 45 units, the edge replaced by edges and . The processes described above might break the biconnectivity constraint, but since reliability is not one of the key evaluation criteria, it will be placed of a lesser importance.

4.3 Comparison between the two algorithms

Both the SMA and Union of Rings Algorithm provide a topology of the pathway system for Culver Academies that has a balanced cost and flow. Since the network planning problem has multiple criteria, neither algorithm can guarantee a globally optimal solution. The SMA offers a decentralized, self-optimized approach where the algorithm “learns” the optimal paths over time. Meanwhile, the Union of Rings Algorithm requires a predetermined traffic requirement, but its optimization process is more transparent since it is based on mathematical standards. The Union of Rings Algorithm and the SMA provides a comparison between a globally optimized method and a more locally optimized method (Blessings, 1999).

Since both algorithms are heuristic, it allows for improved flexibility during optimization, making it capable of handling non-linear and discrete constraints in this problem.

4.4 Intersections

Currently, the dataset only includes locations of buildings as nodes. In application, intersections between roads are made to shorten travel distance, distribute traffic and reduce overall cost. Intersections can also be represented by nodes, but its location must be calculated rather than collected. These points that are added to optimize computations are known as Steiner points.

In this study, the intersections in the pathway system are calculated using a Voronoi diagram. The Voronoi diagram partitions a plane containing points into polygons such that each polygon contains one point and any point on the polygon is closer to the point that it contains than to any other (Weisstein). This makes the edges and vertices of the polygons a reasonable location to place intersection nodes. In this study, the intersection nodes will be placed on vertices of Voronoi polygons as it would have equal distance to at least 3 other nodes.

Shown in Figure 5.4.1, the Voronoi diagram divides the campus into zones of influence for each building.

图表, 雷达图

描述已自动生成

To avoid redundancy, the intersection node will only be placed when there is no other intersection node in its proximity. The intersection nodes will also not be placed when its distance to the nearest building node exceeds a threshold.

图表, 雷达图

描述已自动生成

The resulting graph representation with intersections added is shown above. This model will be used to generate paths in both the SMA and Union of Rings Algorithm.

4.5 Evaluation methods

Several criteria can be used to evaluate the effectiveness of a transportation network. This study will focus on the quantitative statistics of the graphs, particularly their clustering coefficients, average path length, and total cost of the pathway system. An ideal graph model should satisfy the “small world property”, which includes two criteria to evaluate a graph model: clustering and path length. Models that can produce graphs with higher clustering and shorter path lengths are better (Downey, 2018). In this case, higher clustering means that buildings are connected more directly, while a shorter path length allows people to take less time to reach their destinations. Clustering coefficients and mean path lengths are determined using the open-source Python module NetworkX.

The clustering coefficient is a measure of how interconnected the nodes in the graph are; it measures the likelihood that its neighbors are also connected to each other. The clustering coefficient of a node with neighbors is calculated by:

is a value between 0 and 1, with 0 meaning that none of the neighbors of are connected, and 1 meaning that all neighbors of are connected. In this study, I will look at the average clustering coefficient for all nodes in the graph. In a graph with nodes, the average clustering coefficient is given by:

To evaluate the efficiency of the pathway network, the mean shortest path length can be used. It measures the average shortest path in the pathway network between 2 arbitrary buildings. The formula of mean distance , given the network’s graph model , is:

Where is the number of nodes in the graph and is the shortest path distance between nodes and .

Another way of evaluating a graph is analyzing the probability distribution of the degrees of nodes (how many edges a node is connected to). A PMF (Probability Mass Function) to degree chart will be created to reveal node connectivity patterns. For instance, a scale-free network, where connections are concentrated on a few nodes, follows a power-law distribution. Meanwhile a random network, which is the case for most transportation networks, follows a Poisson distribution (Nykamp).

图表, 折线图

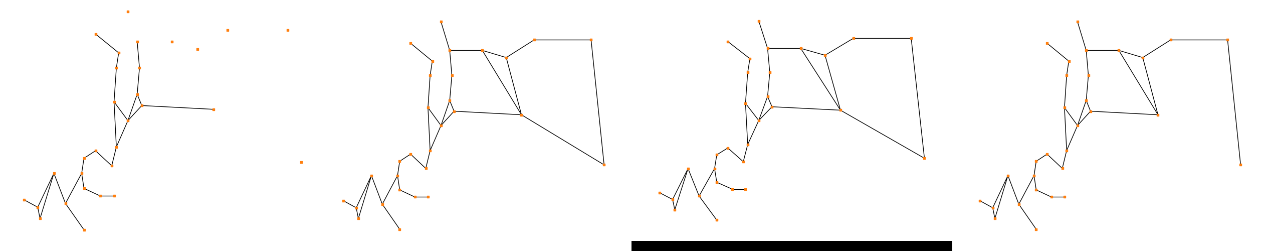
描述已自动生成

Figure 5.5.1. An example of Poisson distribution and power-law distribution in the context of this study.

1. **Results**

5.1 Slime Mold Algorithm

The slime mold simulation used in this study produces step-by-step simulations. This simulation scaled the dataset to a grid where each unit corresponds to 10-4 degrees in both latitude and longitude on a under decay()=0.15. Figure 6.1.1. depicts 4 screenshots of the simulation after 50, 150, 250, and 350 steps, as well as their corresponding graph network. 图片包含 图示

描述已自动生成

a) 50 steps b) 150 steps c) 250 steps d) 350 steps

Figure 6.1.1. Visualization of the slime mold simulation.

|  |  |  |
| --- | --- | --- |
|  | Original Graph | Slime Mold Graph |
| Clustering Coefficient | 0.294047 | 0.130434 |
| Avg. Path Length | 4.193103 | 6.744827 |
| No. of edges | 45 | 33 |
| Total Path Length (Cost) | 525.316170 | 343.959840 |

Table 6.1.1. Comparison of evaluation criteria between the original and resulted graph.

Table 6.1.1 shows the comparison between the graph connected by the slime mold and the original graph from Culver Academies’ existing pathway system. We can observe that the generated graph resulted in a much different pathway network compared to the original system. Overall, the original system is a better model with regards to the “small world properties” but had significantly more cost.

图表, 折线图

描述已自动生成

Figure 6.1.2. PMF in linear scale

Figure 6.1.2 shows that the original system contained nodes with higher degrees, which indicates that the original system traded cost for better access and robustness. Both the original and generated graph follow Poisson distribution, indicating that both graphs are random graphs.

To test SMA’s adaptability to different cost-efficiency constraints, a parametric study is conducted on different decay values. The values tested are 0.075, 0.1, 0.15, 0.2, and 0.25.

|  |  |  |  |
| --- | --- | --- | --- |
| **Decay** | **Clustering Coefficient** | **Path Length** | **Edges** |
| 0.075 | 0.106667 | 7.022989 | 32 |
| 0.100 | 0.173333 | 6.898851 | 33 |
| 0.150 | 0.073333 | 6.347126 | 33 |
| 0.200 | 0.108696 | 6.310345 | 33 |
| 0.250 | 0.036232 | 6.425287 | 33 |

Table 6.1.2. Statistics of the parametric study

Table 6.1.2. shows a comparison of the three decay values under the evaluation criteria. Based on the table, no linear trend can be concluded about decay value and network efficiency. Little improvement is made from these intervals of decay values.

Other statistics are collected during network generation, shown in Figures 6.1.3 to 6.1.5.

A blue and white bar graph

Description automatically generated

A graph of different colored lines

Description automatically generated

A graph of different colored lines

Description automatically generated

Figures 6.1.3 to 6.1.5 all suggest that as decay values increase, the number of steps required to connect all food sources decreases.

6.2 Union of Rings Algorithm

The Union of Rings Algorithm is applied to the geographic dataset scaled identically with that in the SMA. The generated graph is shown in Figure ().

A map of the constellation

Description automatically generated

The statistics of the network produced is found in Table ().

|  |  |  |
| --- | --- | --- |
|  | Original Graph | Graph produced by Union of Rings |
| Clustering Coefficient | 0.294047 | 0.130434 (change to actual) |
| Avg. Path Length | 4.193103 | 3.687356 |
| No. of edges | 45 |  |
| Total Path Length (Cost) | 525.316170 |  |

The statistic of the resulted graph indicates that it had improvements in both total and average path length, creating a sidewalk network with faster access between buildings and lower cost.

A graph with a line

Description automatically generated

Based on Figure(), the Union of Rings Algorithm created a network with more even degree distribution.

* 1. Graph Generation with Added Intersections

1. **Conclusion**
2. **Future Works**

This study, focused on planning sidewalks in Culver Academies, had provided insights towards network planning. Since traffic is largely ignored in this paper, further studies can investigate the dynamic traffic flows in Culver Academies throughout the day in peak and off-peak hours, such as lunch time or class commutes. Another way to incorporate traffic is to build intersections using a weighted Voronoi diagram, where buildings can be prioritized based on traffic requirements.

The study focused solely on heuristic algorithms. Future studies could explore non-heuristic algorithms particularly those used in objective optimization, where target functions modeling cost and accessibility can be used to determine a global best solution. However, it might be challenging to incorporate the optimizers into a discrete setting.

Other methods of creating Steiner points can also be explored. The Voronoi diagram is one of many conventional methods to create intersection points in graph problems. Other methods, such as Delaunay triangulation, focus on different aspects of the graph than the Voronoi diagram, which might produce drastically different results.

Works Cited

Cai, Zhengying, et al. "A Node Selecting Approach for Traffic Network Based on Artificial Slime Mold." *IEEE Access*, vol. 8, 2020, pp. 8436-48, https://doi.org/10.1109/access.2020.2964002.

Corne, David, et al. *Telecommunications Optimization : Heuristics and Adaptive Techniques*. Wiley, 2000.

Griffin, Christopher. *Applied Graph Theory : an Introduction with Graph Optimization and Algebraic Graph Theory*. World Scientific, 2023.

Lee, Jason. *NETWORK OPTIMIZATION USING LINEAR PROGRAMMING AND REGRESSION*. 2016. U Oregon, Bachelor's thesis. *University of Oregon*, scholarsbank.uoregon.edu/server/api/core/bitstreams/cff1b048-02b6-442a-9eae-3c2a7b170ead/content. Accessed 1 Oct. 2024.

Li, Shimin, et al. "Slime Mould Algorithm: A New Method for Stochastic Optimization." *Future Generation Computer Systems*, vol. 111, Oct. 2020, pp. 300-23. *ScienceDirect*, https://doi.org/10.1016/j.future.2020.03.055. Accessed 15 Oct. 2024.

Penttinen, Aleksi. Lecture. *Helsinki University of Technology*, 1999, www.netlab.tkk.fi/opetus/s38145/s99/lectures/lect10.pdf. Accessed 6 Oct. 2024.

Tero, Atsushi, et al. "Rules for Biologically Inspired Adaptive Network Design." *Science*, vol. 327, no. 5964, 2010, pp. 439–42. *JSTOR*, http://www.jstor.org/stable/40508592. Accessed 15 Oct. 2024.

Zhang, Wenxiao. "CITS4403 Project Report." U of Western Australia, 17 Oct. 2022. *Github*, github.com/MoeBuTa/SlimeMould/blob/master/report/report.pdf. Accessed 23 Oct. 2024.

Zhu, Hang, et al. "Network planning with deep reinforcement learning." *SIGCOMM '21: Proceedings of the 2021 ACM SIGCOMM 2021 Conference*. *ACM Digital Library*, dl.acm.org/doi/10.1145/3452296.3472902. Accessed 6 Oct. 2024.