

Carinthia II

Part 3

Nature Tech

Published since 1811
215th Year of Carinthia



Volume 2

Issue 2

2025 Verlag des Naturwissenschaftlichen
Vereins für Kärnten

Applying a slime mold-inspired algorithm to network planning challenges in the Carinthian region

Kristina Wogatai, Emir Sinanović, Wilfried Elmenreich

ABSTRACT

This article presents the application of the slime mold-based algorithm *Simulation of Slime Molds* (SISMO) using case studies from the Carinthian region, demonstrating its applicability to various network structures. SISMO is inspired by the ability of the myxomycete *Physarum polycephalum* to process information and solve optimization tasks. The effectiveness of SISMO in network formation and finding shortest paths is first evaluated using parts of Carinthia's bus and railway infrastructure. Our results show that SISMO can create networks similar to the existing transportation network, consider obstacles, and replicate existing connections. These findings allow conclusions for the future optimization of further networks using bio-inspired approaches. Based on these insights, this paper presents an approach that uses SISMO to create a repair plan for the Carinthian power transmission network after a simulated electromagnetic pulse attack. The algorithm is applied to a corresponding graph model to identify the most critical areas to be repaired, analogous to the nutrient supply of the slime mold between food sources. SISMO was adapted for this grid planning task and fed with the positions of relevant power plants and substations of the Carinthian electricity transmission grid. This approach has significant potential for diverse applications, especially those where the algorithms are not limited by the physical or infrastructural constraints that shaped the original network topology. Further exploration of this approach could yield significant insights into various fields. Overall, the paper provides insights into the potential applications of bio-inspired algorithms, such as slime mold simulation, for solving network planning tasks and presents concrete case studies from the Carinthian region.

Anwendung eines von Schleimpilzen inspirierten Algorithmus für Netzwerkplanungs-herausforderungen in der Region Kärnten

ZUSAMMENFASSUNG

Dieser Artikel präsentiert die Anwendung des auf Schleimpilzen basierenden Algorithmus SISMO (*Simulation of Slime Molds*) anhand von Fallstudien aus der Kärntner Region und demonstriert dessen Anwendbarkeit auf verschiedene Netzwerkstrukturen. SISMO ist inspiriert von der Fähigkeit des Myxomyceten *Physarum polycephalum*, Informationen zu verarbeiten und Optimierungsaufgaben zu lösen. Die Effektivität von SISMO bei der Netzbildung und der Ermittlung kürzester Wege wird zunächst anhand von Teilen der Bus- und Bahn-Infrastruktur Kärntens bewertet. Unsere Ergebnisse zeigen, dass SISMO in der Lage ist, Netzwerke zu erzeugen, die dem bestehenden Verkehrsnetz ähneln, Hindernisse zu berücksichtigen und bestehende Verbindungen nachzubilden. Diese Erkenntnisse ermöglichen Rückschlüsse auf die zukünftige Optimierung weiterer Netzwerke durch bioinspirierte Ansätze. Aufbauend auf diesen Einsichten stellt diese Arbeit einen Ansatz vor, bei dem SISMO zur Erstellung eines Reparaturplans für das Kärntner Stromübertragungsnetz nach einem simulierten elektromagnetischen-Impuls-Angriff (EMP) eingesetzt wird. Der Algorithmus wird auf ein entsprechendes Graphmodell angewandt, um die kritischsten Bereiche für Reparaturen zu identifizieren – analog zur Nährstoffversorgung des Schleimpilzes zwischen Nahrungsquellen. Für diese Aufgabe der Netzplanung wurde SISMO angepasst und mit den Standorten relevanter Kraftwerke und Umspannwerke des Kärntner Stromübertragungsnetzes gespeist. Dieser Ansatz birgt vielversprechendes Potenzial für eine Vielzahl von Anwendungen, bei denen Algorithmen nicht durch physische oder infrastrukturelle Einschränkungen limitiert sind, die das ursprüngliche Netzwerk beeinflussen haben könnten. Eine weiterführende Untersuchung dieses Ansatzes könnte wertvolle Erkenntnisse in unterschiedlichen Anwendungsfeldern liefern. Insgesamt bietet der Artikel Einblicke in das Potenzial bioinspirierter Algorithmen wie der Schleimpilz-Simulation zur Lösung von Aufgaben der Netzplanung und stellt konkrete Fallstudien aus der Region Kärnten vor.

INTRODUCTION

Motivation

Biological principles are increasingly being used in various disciplines to develop and optimize technical systems and products. One prominent example of these nature-inspired optimization algorithms is the Ant Colony Optimization algorithm (ACO), which is based on the foraging behavior of ants. Dorigo et al. first introduced the ACO algorithms in the early 1990s [1], [2], [3], and they have since been applied in various domains, including manufacturing and intralogistics route optimization, communication channel switching,

KEYWORDS

- > bio-inspired algorithm
- > slime molds
- > network planning
- > traffic networks
- > transmission networks
- > self-organization
- > multi-agent simulation
- > NetLogo
- > Carinthian use-cases

and transportation management. Similar research has been performed based on the myxomycete *Physarum polycephalum* (PP), with Nakagaki et al. demonstrating that these organisms are even capable of information processing [4], [5], [6]. PP is a unicellular organism that, during one phase of its unique life cycle, typically forms a multinucleated protoplasmic mass without internal cell walls, known as a plasmodium [7]. It consists of a single cell with billions of nuclei that can stimulate the formation of pseudopodia (false limbs) to facilitate movement. In addition to their mobility, these slime molds are particularly interesting in different research fields due to their chemosensory perception and spatial memory [8]. Their behavior can be leveraged to develop optimization algorithms.

Objectives

In this article, our objective is to answer the following questions:

- “To what extent does the SISMO algorithm replicate the network formation behavior of real slime molds under controlled conditions, including the presence of obstacles?”
We show that our algorithm can replicate the behavior of slime molds and thus can be assigned to network formation tasks.
- “How accurately can SISMO model and reconstruct existing transportation and power transmission networks based on geographic and infrastructure data?”
Our results indicate that the algorithm would be able to partially reconstruct the Carinthian transmission network in the event of an electromagnetic pulse (EMP) attack.
- *We further show that using layout preprocessing with Gephi’s [9] layout algorithms, such as ForceAtlas, ForceAtlas2, or Yifan Hu, can improve the accuracy of SISMO in network environments where geographical proximity does not correspond to actual topological connectivity.*

This study does not include a comparison between SISMO and conventional planning methods, nor does it perform functional simulations (e.g., electrical simulations) of the resulting networks. The focus is on presenting a novel bio-inspired modeling approach. Our team has developed the slime mold-inspired SISMO algorithm and evaluated its effectiveness in forming networks and determining the shortest paths between individual points within them. To test its applicability, sections of the Carinthian traffic network were chosen as examples. Initially, in vitro experiments were performed to determine whether slime molds cultivated in Petri dishes could build a network similar to the existing traffic network. Subsequently, the SISMO algorithm was tested on the same sections. The first experiment was to determine whether both the slime molds and SISMO could build a network similar to the existing traffic network. The second experiment evaluated whether SISMO mimicked the behavior of the natural model. This study aimed to demonstrate the potential of SISMO for application to all types of networks that can be modeled using slime mold-based planning. In a third, separate experiment, it was investigated how the algorithm behaved when obstacles were present. This third experiment was conducted using SISMO in NetLogo programming language. The results demonstrated that SISMO could achieve levels of accuracy similar to those of real slime molds when obstacles were incorporated into the simulation. In this context, accuracy refers to the number of pre-existing connections replicated by the algorithm. Furthermore, a performance update was conducted, revealing that the simulation approach, despite its computational intensity, exhibits significantly faster results compared to the in vitro experiments. By modeling natural barriers as obstacles within the simulation, a significant improvement in the accuracy of the results was observed. In addition, the scalability of the SISMO algorithm was evaluated. The scalability measurements indicate that the simulation time is mainly influenced by the number of network nodes present within the slime mold

network rather than by the number of points to find (food sources). As the effectiveness of SISMO has been successfully established, it can be expected to be applied in optimizing other networks in practical applications, such as power supply networks, communication networks, and supply chain networks.

After demonstrating that SISMO can achieve satisfactory results in the planning and reconstruction of transportation networks, the algorithm was tested for application to transmission networks. This work builds on Himmelstein's proposal [10] by applying a slime mold-inspired algorithm to restore a power transmission network after an EMP attack. The network is modeled as a graph, with substations as nodes and transmission lines as edges. SISMO identifies critical repair areas, enabling faster and more resource-efficient recovery. The goal of this work was to develop an approach for reconstructing the power grid to ensure the protection of critical infrastructure.

The Carinthian transmission network was originally designed with geomorphological and infrastructural conditions in mind. In such networks, it often occurs that the geographical location of the nodes does not correspond to their actual connections. In the first and second experiments, this discrepancy often led SISMO to create edges between geographically close points that do not exist. We attempted to solve this problem by using the layout algorithms in the software Gephi [9] to adjust the network layout based on the existing connections. Through this preprocessing, we aim to improve the accuracy and usability of the simulation results.

BACKGROUND

Slime Molds

Slime molds show characteristics of animals and fungi, but do not belong to either of these groups. Rather, they form their own taxon and have over 1000 different species distributed all over the world, roughly divided into the three groups Myxomycetes, Dictyostelia, and Protostelia. PP belongs to the Myxomycetes, characterized by the formation of plasmodia, a multinucleated plasma mass in one of its unique life phases. The cell nuclei can stimulate the formation of pseudopodia, which serve the organism for locomotion. PP is mainly found on rotting wood and on fruiting bodies of fungi, but occasionally also grows on living plants (Figure 1). Due to its easy handling, cultivability and cell size, it is often used as a model organism in research. SISMO simulates the behavior of the PP in the plasmodium stage. In this stage, the slime mold grows as long as it has food available. It doubles the number of its cell nuclei approximately every eight hours. A plasmodium feeds on organic material such as bacteria, algae, fungal spores or fungi. While feeding, the plasmodial mass moves around the food source, ingesting digestible material and excreting indigestible material. In doing so, the slime mold leaves a trail which, together with a kind of chemosensory perception, is used in the search for food [11].



Fig. 1

Figure 1:
Plasmodium of *Fuligo septica* found in the Carinthian Woods
(Photo by W. Marcher).
Source: Figure 2.2, [16]

Abbildung 1:
Plasmodium von *Fuligo septica*, gefunden in den Kärntner Wäldern
(Foto: W. Marcher).
Quelle: Abbildung 2.2, [16]

The locomotion of the slime mold is made possible by a cytoskeleton of actin filaments and the protein myosin. Contractions in a slime mold follow the same principle as muscle contractions. The two protein filament structures are also responsible for the plasma currents that promote locomotion. Organisms capable of locomotion generally use sensory mechanisms that allow them to sense changes in environmental conditions and move toward better conditions. The plasmodium of PP uses specific signal transduction mechanisms to respond to changes in the chemoreceptor conformation with motor chemotaxis responses. Chemotaxis is the movement of cells toward attracting substances (positive chemotaxis) or away from repelling substances (negative chemotaxis) [11].

PP is not only capable of movement and sensing chemical stimuli, but also has a remarkable ability to remember the location of food for future use. Although spatial memory is typically associated with the brain of an organism, PP, like all slime mold species, does not have a brain [8]. However, it is able to avoid previously explored areas while foraging, enabling it to solve complex optimization problems. This ability cannot be solely attributed to its chemosensory perception, indicating that slime molds possess a type of memory. Memory refers to the retention and retrieval of information related to previous events [12]. Some ant species employ external memory by leaving pheromone trails to mark the route between the nest and the food source. This enables indirect route communication without individual memory retention [8]. The plasmodium of PP consists of many small oscillating units that oscillate at a frequency determined by their environment and interaction with neighboring oscillators [13]. Positive chemotaxis increases the oscillatory frequency near the food source, causing plasma to flow in that direction. In contrast, negative chemotaxis decreases the oscillatory frequency [14]. The collective behavior of the many oscillating units leads to the movement of the organism. As it moves, the plasmodium leaves behind a specific protein-based extracellular slime (Figure 2). In their research, Reid et al. [8] suggest that the plasmodium avoids areas with this slime when foraging, but this behavior ceases when there are no unexplored areas left. Based on this observation, Reid et al. inferred the existence of an externalized spatial memory system in the plasmodium of PP.



Figure 2:
Plasmodium
of *Physarum*
polycephalum
cultivated in a Petri
dish. Source: own
image

Abbildung 2:
Plasmodium
von *Physarum*
polycephalum, kultiviert
in einer Petrischale.
Quelle: eigenes Bild

Fig. 2

SISMO

The first version of SISMO was developed as part of two master's theses at the University of Klagenfurt [15], [16]. NetLogo was used to implement the SISMO algorithm. Developed by Uri Wilensky in 1999, NetLogo is a free, open-source modelling environment for simulating complex natural and social systems. It enables independent mobile agents, known as *turtles* and stationary *patches* to interact and form networks or graphs, making it ideal for multi-agent simulations. It is well-suited to model slime mold behavior, effectively capturing movement and chemical responses. Advantages include scalability, flexibility and ease of programming, enabling the detailed study of decentralized, emergent behaviors in complex systems. Figure 3 illustrates how our algorithm operates within the NetLogo framework. It details how the slime mold network locates, searches for, and spreads to food sources, and how it forms the shortest possible path. SISMO is primarily based on the first three phases of the slime mold algorithm (SMA) [17] foraging process, namely searching food, approaching food, and wrapping food. In the SMA, pseudopods balance exploration and exploitation by first randomly searching for food sources and then reinforcing efficient paths once found. This ensures the algorithm avoids local optima while gradually identifying the shortest or most effective routes. SISMO applies this principle to optimize network restoration by focusing on critical connections without overlooking alternative paths. The SMA uses parameters LB and UB, where LB and UB denote the lower and upper limits, to set the search range bounds that define the area in which the slime mold can move. In contrast, the simulation of SISMO in NetLogo defines these boundaries by the size of the world model settings. Once a pseudopod finds a food source, it moves toward it during the foraging phase, following the NetLogo propagation pattern, known as the wiggle function, which is different from the arbitrary

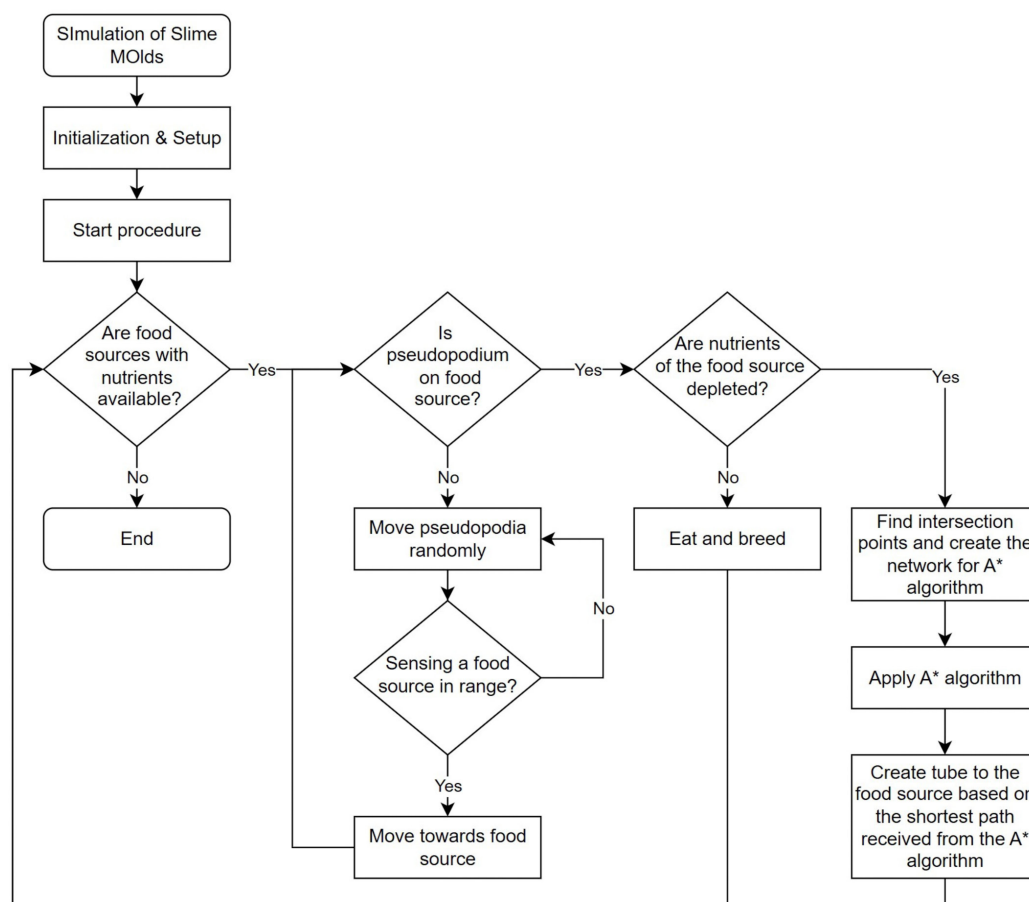


Figure 3:
NetLogo SISMO
algorithm workflow.
Source: own image

Abbildung 3:
Arbeitsablauf des
NetLogo-SISMO-
Algorithmus. Quelle:
eigenes Bild

Fig. 3

pseudopodia alternating positions in SMA. The wiggle command causes a turtle, in this case a pseudopodium, to perform a random rotation at a specified angle. The pseudopod considers the food source as a point during the food approach phase [15], [16]. The simulation terminates when there are no more food sources for the simulated slime mold to explore. This condition is based on global knowledge that would not be available for a natural slime mold and is not part of the core algorithm.

SISMO is available as open source on GitHub under a Creative Commons license [18]. To run simulations with SISMO, NetLogo version 6.2.2 is recommended. Clicking on the SETUP button will create the plasmodium, the selected number of food sources, and the selected number of initial pseudopodia. The slider AMOUNT-PSEUDOPODIA can be used to set the number of initial pseudopodia. The AMOUNT-FOODSOURCES slider sets the number of food sources. The SHOW-NUTRIENT-VALUE switch determines whether or not to display the number of nutrients from the food sources. The SHOW-NETWORK switch determines whether the created network needed for the A* algorithm should be displayed or not. The SHOW-INTERSECTION-POINTS switch defines whether the calculated intersection points should be displayed or not. The simulation is started with the GO button.

Prior to initiating SISMO, various simulation parameters can be established and configured using the NetLogo graphical user interface (GUI).

- AMOUNT-PSEUDOPODIA Defines the number of pseudopodia.
- AMOUNT-FOODSOURCES Defines the number of food sources.
- SHOW-NUTRIENT-VALUE Defines whether the nutrient value of the food sources should be displayed.
- SHOW-NETWORK Defines whether the network for the A* algorithm should be displayed.
- SHOW-INTERSECTION-POINTS Defines whether the intersection points should be displayed.

During initialization, all agent types (breeds) are created with their required properties and values. Users can choose certain values, such as the number of pseudopods and food sources, through the GUI.

The simulation uses six different breeds:

- Plasmodium
The plasmodium forms the core of the slime mold. From it, the pseudopodia spread out. In addition, the plasmodium serves as a starting point for calculating the shortest path using the A* algorithm.
- Pseudopodia
The pseudopodia are responsible for foraging. They spread randomly and form new pseudopodia when a food source is found.
- Tubes
Tubes are responsible for the visualization of the shortest path. The pseudopodia also form tubes, but in the separately created breed, the shortest path is thicker and visualized differently.
- Foods
This breed acts as food sources for the slime mold.
- Networkpoints
This breed is used for the network points in the A* algorithm.
- Searchers
In the A* algorithm searchers are used to calculate the shortest path.

During the setup phase, all pre-defined breeds are generated along with their respective attributes. The procedure involves creating each breed, defining their properties, assigning values to their traits, and ultimately displaying the plasmodium, pseudopodia, and food sources.

The SISMO algorithm's primary element is the simulation process, which comprises four main components: movement, feeding and breeding, network construction, and pathfinding. Agents (pseudopodia) move left or right at random and then move forward one step. If this is not possible, the agent rotates 180 degrees. During the movement, the agents cannot leave the specified map boundaries and bounce off the edge if encountered. After performing the movement, the x and y coordinates of the new position of the pseudopodium are stored in a list. When a pseudopodium reaches a food source during its propagation, it picks up nutrients from that source as long as they are present. A random number (ξ) between zero and one is generated. This random number determines whether a plasmodium will divide or not. The value of 15% for the hatching probability was chosen after several test runs, as it gives the best results in terms of performance and visualization. If this random number is less than or equal to 0.15, a new pseudopodium is generated. It inherits all the properties of its parent except its ID. A new path list is created for the newly created pseudopodium, with the values for ξ set to 1. This means that the entries in the new path list are a copy of the entries in the path list of the respective parent. For further movement, ξ is set to 0, since the new plasmodium will move independently of the parent from this point. In the construction of the network, all paths traveled by all pseudopodia must be considered. These paths are stored in the path lists of each pseudopod. As each path list is compared to itself and the others, the system searches for intersection points. The position of an intersection point found is entered into the path list. If no intersection point is found, a node is created. The graphical output of a

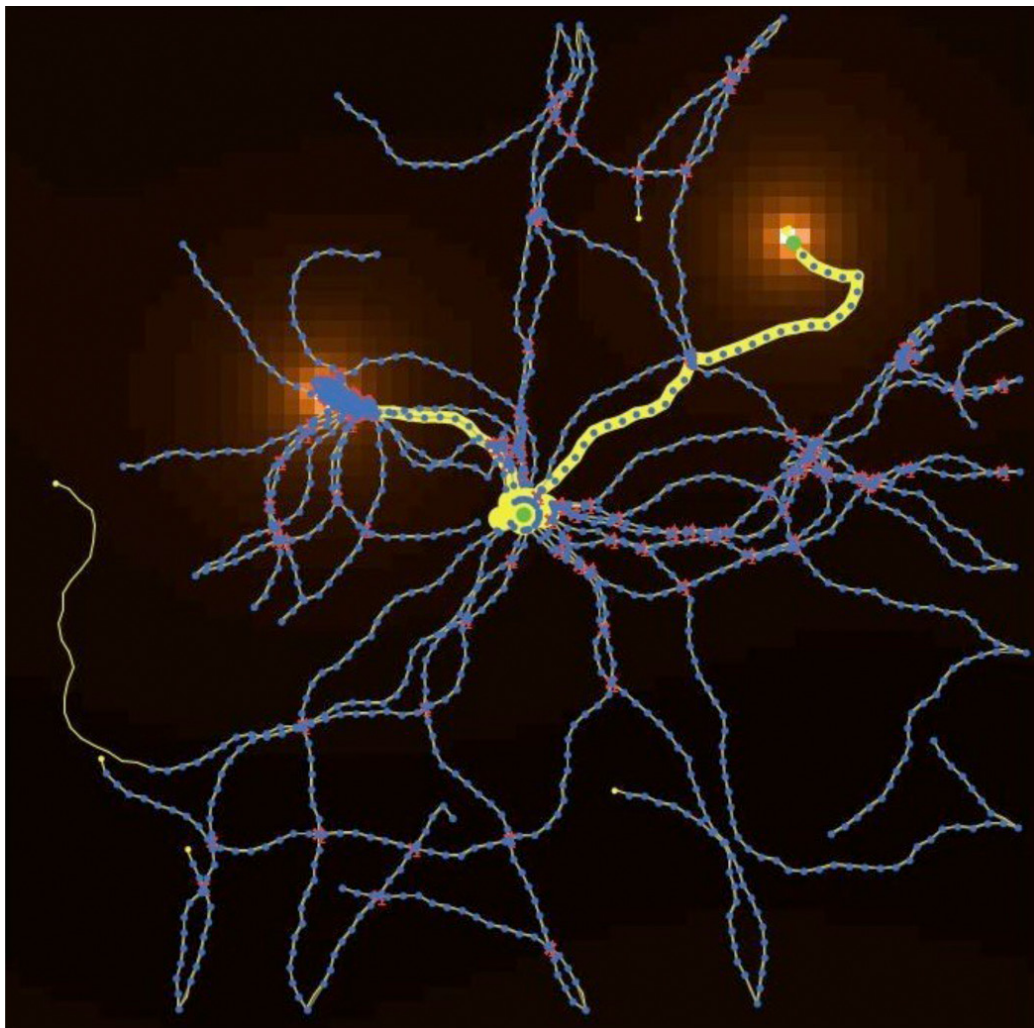


Figure 4:
SISMO Algorithm:
Simulation with
Shortest Path, Network
Nodes, and Intersection
Points. Source: Figure
5.12, [16]

Abbildung 4:
SISMO-Algorithmus:
Simulation mit
kürzesten Wegen,
Netzwerkknoten und
Schnittpunkten. Quelle:
Abbildung 5.12, [16]

SISMO simulation in NetLogo is depicted in Figure 4. The blue points represent the created network nodes. The crossing points are represented as red crosses. The A* algorithm is used to find the shortest path within the network. The thick yellow path represents the shortest routes between the core and the food sources.

To emulate the behavior of the slime mold in finding the shortest path in the network on which the mold is located [19], the simulation implements an A* algorithm to find the shortest path. To build the network for the A* algorithm, all pseudopodia paths must be considered and stored in a list of path lists. When there are multiple pseudopodia, the first path list is compared to the other path lists, including itself to prevent overlapping. The first element in the path list is compared with all other elements. Subsequently, the second element of the path list is compared to all others except the first to avoid unnecessary iterations. Checking for overlap is critical in constructing a network similar to that of a real slime mold. To achieve this, intersection points in the paths are identified except in specific situations where no intersection check is necessary. After constructing the mesh, the A* algorithm is utilized to calculate the shortest path. The implementation of the A* algorithm used for this project is built upon Caparrini's *General A* Solver in NetLogo* [20]. However, it has been modified to include a graphical display of the shortest path. Once the A* algorithm is applied, a new path is created using the coordinates of the shortest path, which is then represented as a thicker yellow line. This approach is inspired by the behavior of slime molds in nature, where paths through which the most nutrients flow are more pronounced. The thicker yellow line serves to highlight the most optimal path found by the algorithm and provides a visual representation of the efficiency of the A* algorithm in finding the shortest path.

METHODS

This section presents several tests performed to verify and assess the functionality of SISMO. When developing SISMO, it was important for the algorithm to behave as similarly as possible to its model in nature. The first two experiments, therefore, aimed to test how closely the results of simulations with SISMO resemble the spread of slime molds in Petri dishes. For this purpose, the plasmodium in the Petri dish and the simulation tool were provided with the same information. Another series of experiments considered obstacles such as lakes and mountains in the planning of the traffic network. SISMO showed promising results in all the transportation network planning experiments conducted. With some modifications, SISMO can also be applied to various other types of networks. A series of tests on the planning and reconstruction of the Carinthian distribution network is presented here. In the first use case, SISMO was applied to the original network of KNG-Kärnten Netz GmbH. Then, SISMO was applied on different networks in which the layout was adjusted based on the existing connections.

Evaluation Metrics

- 1) *Accuracy*: The primary metric used to evaluate the performance of network reconstruction methods is accuracy, defined as the ratio of the number of connections identified by the method—whether simulated or biologically grown—to the total number of actual existing connections in the built network. This metric quantifies how closely the reconstructed network aligns with real-world infrastructure.
- 2) *Performance Metrics for SISMO*: The effectiveness of the SISMO algorithm is assessed using several performance metrics. Simulation time measures the computational effort required to find food sources or complete the network reconstruction, including scalability relative to the number of nodes.

To evaluate the simulation's ability to identify connections, the following terms are defined: True positives (TP) represent connections correctly predicted by the simulation that exist in the real network. False positives (FP) are connections predicted by the simulation that do not exist in reality, while false negatives (FN) are actual connections in the real network that were missed by the simulation. True negatives (TN), though less relevant in this context, refer to non-existent connections correctly not predicted by the simulation.

Precision measures the proportion of correctly predicted connections among all predicted connections:

$$Precision = \frac{TP}{TP + FP}$$

Recall quantifies the proportion of correctly identified connections among all actual connections:

$$Recall = \frac{TP}{TP + FN}$$

The F1 score combines precision and recall as a harmonic mean, providing a balanced assessment of the simulation's performance:

$$F1\ Score = 2 \times \frac{Precision \cdot Recall}{Precision + Recall}$$

The F1 score ranges between 0 and 1, where a value of 1 denotes an ideal classifier with perfect precision and recall, and a value of 0 indicates a classifier with no predictive capability.

These metrics collectively enable a comprehensive evaluation of the SISMO algorithm's ability to accurately reconstruct and optimize network structures.

Experimental Analysis of Living Slime Molds

This set of experiments was part of a master's thesis at the University of Klagenfurt in 2022 [16]. The slime molds for the experiments were ordered partly via Amazon from the company KREZUS in France and partly via the online store Pro Formica (<https://pro-formica.de/>).

A hygiene protocol was followed to prevent mold growth on the test objects. Before use, the Petri dishes and all other utensils (pipettes, tweezers) were sterilized by treatment with boiling water. Agar-agar was mixed and distributed in the Petri dishes as a culture medium before the slime mold, in its dried form of the sclerotium, was placed in it. In this dry form, the plasmodium survives even if the necessary conditions (temperature, humidity, pH value, etc.) are not given. To return the plasmodium to its viscous form, a few drops of distilled water were added.

Experiment 1: ÖBB Railroad Network Carinthia

To prepare for the in vitro experiment, the chosen map section was printed, resized, and affixed to the bottom of the Petri dish. The Klagenfurt main station (point one) was chosen as the starting point for the experiment. At this point, the sclerotium of PP was placed in the Petri dish. From there, the slime mold spread across the map. Both the starting point and the selected stations were marked with oatmeal flakes, which served as food sources for the slime mold. An overview of the selected stations is available in Supplementary Table 1. The setup of this experiment and the setup of the simulation is shown in Figure 5.

Experiment 2: KMG Bus Line to University Klagenfurt

For the second experiment, bus line C of Klagenfurt Mobil GmbH (KMG) was chosen. Line C connects the Klagenfurt main station with the University of Klagenfurt and Lakeside

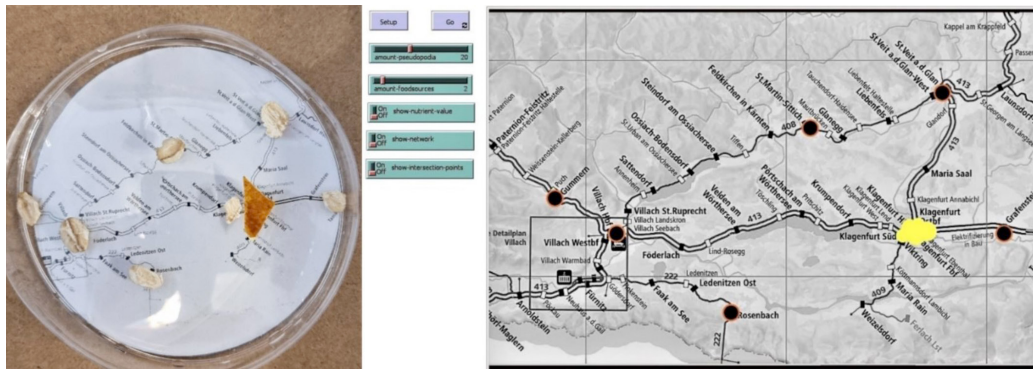


Figure 5:
Setup ÖBB Experiment
1. Source: adapted from [16]

Abbildung 5:
Aufbau des ÖBB-
Experiments 1. Quelle:
angepasst nach [16]

Park every ten minutes. Here, too, the slime mold in the Petri dish and the slime mold simulated with SISMO was tested on the same route section. The selected stations are listed in Supplementary Table 2. The experimental setup of the in vitro experiment and the setup of the simulation are shown in Figure 6.



Figure 6:
Setup KMG Experiment
2. Source: adapted from [16]

Abbildung 6:
Aufbau des KMG-
Experiments 2.
Quelle: angepasst
nach [16]

In the first experiment, the slime mold mimicked part of Carinthia's rail network. After 16 days, it created a structure largely similar to the real network, especially from St. Veit an der Glan. Deviations may have been due to too few or suboptimal stations or deformation of the Petri dish. In the second experiment, the slime mold replicated parts of Klagenfurt's bus line C within nine days. Several sections closely matched the actual route, while others differed, likely for the same reasons as in the first experiment.

Simulations of Experiments 1 and 2: Verifying the Behavior of SISMO

To test whether the behavior of the simulated slime mold in SISMO corresponded to the natural behavior, experiments 1 and 2 were carried out using in vitro experiments in parallel to the simulation tool. For the simulation with SISMO in NetLogo, the same background as printed and affixed on the Petri dishes was inserted as an image, and the same points were selected for the starting point and the food sources. The setup is shown together with the experimental design of the in vitro experiments in Figure 5 and Figure 6. In the first experiment, the slime mold in the Petri dish and SISMO had the task to plan a part of the railroad network of Carinthia based on some given stops. Both were able to plan a network that at least in places corresponded to the real network. The slime mold in the Petri dish took 16 days to solve this task. The simulation with SISMO needed six days for the same task.

In the second experiment, the in-vitro slime mold and SISMO had the task to plan a part of the KMG Klagenfurt bus network. For this purpose, a part of bus line C was chosen, covering the route from Heiligengeistplatz to the University of Klagenfurt and Lakeside Park. Also in this experiment, both the simulation and the natural slime mold were able to

plan a network that at least in places matched the real network. Figure 8 shows the results of the second in-vitro experiment and its simulation with SISMO. For the comparison, the eighth and thus penultimate day of the experiment with the slime mold in the Petri dish was used, since the tubes formed can be seen best here. In this experiment, the slime mold took nine days to reach all the desired stops in the map section and establish a network between them. The SISMO algorithm took four days to simulate the same scenario, considerably faster than the in-vitro experiment.

SISMO simulations showed greater deviations from real networks than the in-vitro slime mold experiments. These differences may result from limited food sources, historical network evolution, or ignored real-world constraints. In both experiments, the in vitro slime mold was slightly more accurate, though SISMO was faster in the second case. Better stop selection could improve both methods.

Table 1 presents a comparison between in vitro experiments and SISMO simulations, the accuracy value reflecting the degree to which the planned network sections correspond to the actual bus and rail networks. Concretely, the first value indicates the number of connections found by the method, while the second value represents the total number of actual existing connections.

Tab. 1		
ÖBB Experiment	Duration	Accuracy
Slime Mold	16 days	4/6
SISMO	16 days	2/6
KMG Experiment	Duration	Accuracy
Slime Mold	9 days	4/7
SISMO	4 days	3/7

After the two series of experiments showed promising results with regard to the similarity of the behavior of the simulation and the slime mold in the Petri dishes, further series of simulations were carried out with SISMO. A comparison of the results of the in-vitro tests and the simulations is shown in Figure 7 and Figure 8.

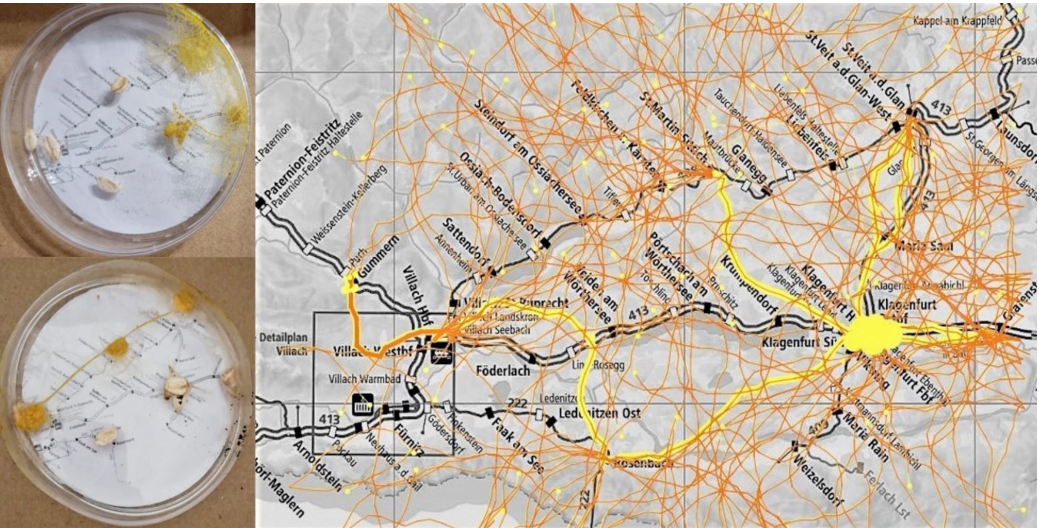


Fig. 7

Table 1:
Comparison of real
slime mold and
SISMO [16]

Tabelle 1:
Vergleich von echtem
Schleimpilz und SISMO

Figure 7:
Comparison of the ÖBB
in-vitro experiment 1 with
SISMO. Source: adapted
from [16]

Abbildung 7:
Vergleich des ÖBB-
In-vitro-Experiments
1 mit SISMO. Quelle:
angepasst nach [16]

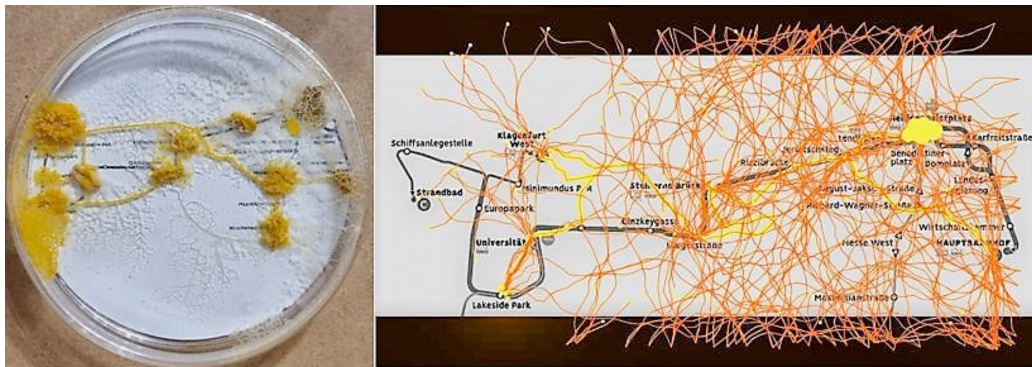


Figure 8:
Comparison of the KMG
in-vitro experiment 2
with SISMO. Source:
adapted from [16]

Abbildung 8:
Vergleich des KMG-
In-vitro-Experiments
2 mit SISMO. Quelle:
angepasst nach [16]

Fig. 8

Experiment 3: Inclusion of Obstacles and Performance Improvement

To simulate obstacles, the SISMO code was not only adapted, but also optimized for performance, which is also important for scalability. In this simulation, the background image used as the map was adjusted. Non-accessible areas were colored green and blue, with blue representing lakes. Specifically, the simulation included the Ossiacher See to the north of the green area and the Wörthersee to the south. The green area itself represented insurmountable obstacles, such as mountains or areas that are legally required to stay free of transportation lines, such as protected areas.

To enhance performance, code optimization was implemented in two areas. The first optimization pertained to the mesh creation for the A* algorithm. Previously, the mesh created during the first food source search was retained, resulting in all coordinates in a large list being checked for existing mesh points during future mesh calculations and extensions. To reduce runtime, the entire mesh was discarded and recreated after each A* algorithm run, for instance, by omitting the search for existing entries in a list containing several thousand items. For the first pseudopodium coordinate list, the runtime was significantly reduced. The second optimization related to the A* algorithm itself. Previously, the algorithm didn't consider whether a searcher was already on a network point, which led to multiple unnecessary runs for already checked routes with multiple searchers on the same point. By avoiding unnecessary redundancy in this way, performance was enhanced.

For implementation of the simulation considering obstacles, the task of the first in vitro experiment (replication of the network of ÖBB) was used. The same route section was simulated with the obstacles using the optimized implementation in NetLogo [15].

Figure 9 presents a comparison between two simulation results for the ÖBB network map. The left side of the figure displays the simulation result without obstacles, while the right side shows the result with obstacles. The simulations were evaluated based on their ability to replicate existing connections between various stops. The comparison reveals that both simulations performed well in replicating the connection between Klagenfurt and Grafenstein. The connection from Klagenfurt to St. Veit also showed similar results in both simulations, with the first difference appearing in the connection to St. Martin Sittich. In the simulation without obstacles, the connection was created cross-country, while the other result showed a much better approximation to the existing route of the ÖBB due to the obstacle. The connection from Klagenfurt to Rosenbach showed similar results in the south of the map in both simulations, with neither simulation creating anything close to a similar connection to the existing network. However, the simulation with obstacles approximated the existing connection between Klagenfurt and Villach much better than the simulation without obstacles, which had a deviation from the existing network. The connection from

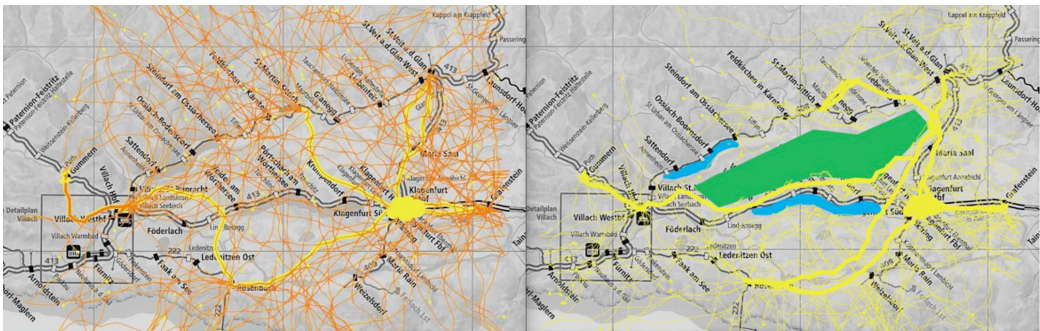


Figure 9:
ÖBB experiment 1
SISMO Simulation
Comparison. Source:
Fig. 6.48 [15]

Abbildung 9:
Vergleich des ÖBB-
Experiments 1 mit der
SISMO-Simulation.

Fig. 9

Villach to Gummern was well approximated by the simulation with obstacles, but not by the other simulation. Notably, neither simulation created the connection from St. Martin Sittich to Villach. Overall, the simulation with obstacles achieved better results by more accurately approximating the network map of the ÖBB.

Scalability

The results presented in this section were measured with the optimized code. In Table 2, an overview of various measurements conducted with SISMO is presented. The table compares the time taken by SISMO to locate a specific number of food sources, where *FS* in the table denotes the amount of food sources. The time is measured in minutes. It should be noted that some food sources may have a higher nutritional value than others, giving the pseudopodia more time to search for additional food sources. Meanwhile, if other pseudopodia have already found a food source, the calculation of the next shortest path is likely to occur almost immediately. Additionally, if fewer pseudopodia have hatched, fewer paths need to be examined for reticulation. The results show that the calculation for the last food source always takes the most time. As the experiment with only two food sources shows, the calculation for the second food source takes considerably longer than the calculation for the second food source in the following experiments with more than two food sources, which is shown in the second column of Table 2. As previously mentioned, the last iteration always takes the longest due to the behavior of the SISMO algorithm. The food sources have a certain number of nutrients, and only when the total number of nutrients is exhausted, the A* algorithm calculates the shortest path. While the nutrients are being depleted, other pseudopodia can continue to move and new pseudopodia can be hatched, resulting in the final iteration taking more time [15].

Tab. 2

# FS	1 found	2 found	3 found	4 found	5 found	6 found	7 found	8 found
1	0.35	-	-	-	-	-	-	-
2	0.08	2.95	-	-	-	-	-	-
3	0.05	1.40	4.19	-	-	-	-	-
4	0.04	0.70	4.44	33.83	-	-	-	-
5	0.05	0.93	3.06	7.86	51.90	-	-	-
6	0.05	0.09	0.67	4.68	22.55	146.97	-	-
7	0.07	0.23	1.50	4.51	20.46	46.94	131.02	-
8	0.04	0.14	0.24	0.44	4.47	20.30	120.83	236.6
Average	0.09	0.92	2.35	10.26	24.85	71.40	125.93	236.6

Table 2:
Number of food sources
(FS) to be found and time
taken (minutes) to find all
of them [15]

Tabelle 2:
Anzahl der zu findenden
Nahrungsquellen (FS)
und dafür benötigte Zeit
(Minuten) [15]

To optimize the simulation of shortest paths in a network, it is helpful to take into account the number of network nodes in each simulation step. Table 3 shows the time SISMO takes per node depending on the number of food sources. In this table, the time required to calculate the shortest path to the food source is indicated by the abbreviation STIS (Simulation Time In Seconds). The value represents the simulation time required for the number of nodes specified in NON (Number Of Nodes). Therefore, STIS/NON (S/N) indicates the value of how many seconds per node are needed. As the network size increases, this value increases as well. This is due to the fact that the list of nodes in the network becomes larger during creation, so it takes longer to check whether a node already exists to avoid creating unnecessary nodes.

Tab. 3

# FS	STIS	NON	S/N	S/N	S/N	S/N	S/N	S/N	S/N	S/N	S/N	S/N	S/N	S/N	S/N
1	STIS	21	0.02	-	-	-	-	-	-	-	-	-	-	-	-
	NON	1041	-	-	-	-	-	-	-	-	-	-	-	-	-
2	STIS	4.8	0.01	177	0.09	-	-	-	-	-	-	-	-	-	-
	NON	622	-	1915	-	-	-	-	-	-	-	-	-	-	-
3	STIS	3	0.01	84	0.06	251	0.16	-	-	-	-	-	-	-	-
	NON	522	-	1427	-	1609	-	-	-	-	-	-	-	-	-
4	STIS	2.4	0.01	42	0.04	266	0.14	2030	0.54	-	-	-	-	-	-
	NON	463	-	1162	-	1902	-	3741	-	-	-	-	-	-	-
5	STIS	3	0.01	55.8	0.04	184	0.12	472	0.23	3114	0.64	-	-	-	-
	NON	534	-	1250	-	1507	-	2063	-	4888	-	-	-	-	-
6	STIS	3	0.01	5.4	0.01	40.2	0.03	281	0.12	1353	0.4	8818	1.52	-	-
	NON	475	-	490	-	1257	-	2359	-	3419	-	5793	-	-	-
7	STIS	4.2	0.01	13.8	0.02	90	0.06	271	0.16	1228	0.41	2816	0.88	7861	1.57
	NON	586	-	616	-	1419	-	1700	-	2967	-	3216	-	5016	-
8	STIS	2.4	0.005	8.4	0.01	14.4	0.02	26.4	0.03	268	0.12	1218	0.34	7250	1.25
	NON	498	-	675	-	676	-	860	-	2276	-	3570	-	5801	14196 6629
Sum			0.06		0.28		0.53		1.08		1.56		2.74		2.14
Average			0.01		0.04		0.09		0.22		0.39		0.91		1.41

Table 3:
Time for SISMO needed
per network point [15]

Tabelle 3:
Für SISMO
benötigte Zeit pro
Netzwerkknoten [15]

Use Case: Repair Plan for the Carinthian Transmission Grid

In [21], we have presented an approach in which a repair plan for the Carinthian power transmission network after an EMP attack is developed using SISMO. In summary, SISMO is applied to a graph model to identify critical areas for repair. For this purpose, the algorithm was adapted and provided with the positions of the relevant power plants and substations in the Carinthian power transmission network. The locations of the 63 considered power plants and substations were taken from an overview by the *Landesrechnungshof Kärnten* (Supplementary Table 3) [22], and their coordinates were converted into a format compatible with NetLogo using a custom-developed program. Since the simulation results sometimes contained disconnected networks or introduced non-existent connections, graph visualization algorithms were proposed for preprocessing. Specifically, the layout was improved using Force Atlas, Force Atlas2 and Yifan Hu Proportional algorithms provided within the Gephi software [9]. Graph-layout-algorithms rearrange the nodes in a network without altering the nodes or edges themselves. Some of these algorithms align node positions based on their connectivity, placing connected nodes closer together and unconnected nodes further apart. This preprocessing step improves the accuracy and significance of the simulation results.

A total of eight simulations were performed using the original network before rearranging it with graph layout algorithms. The average duration of the simulations was 50 minutes and 5 seconds. Figure 10 (upper left) shows the result of one simulation based on Carinthia’s power transmission network. The plasmodium is visualized as a yellow cloud in the center, while food sources, placed at the coordinates of substations and power plants are marked as black dots with orange frames.

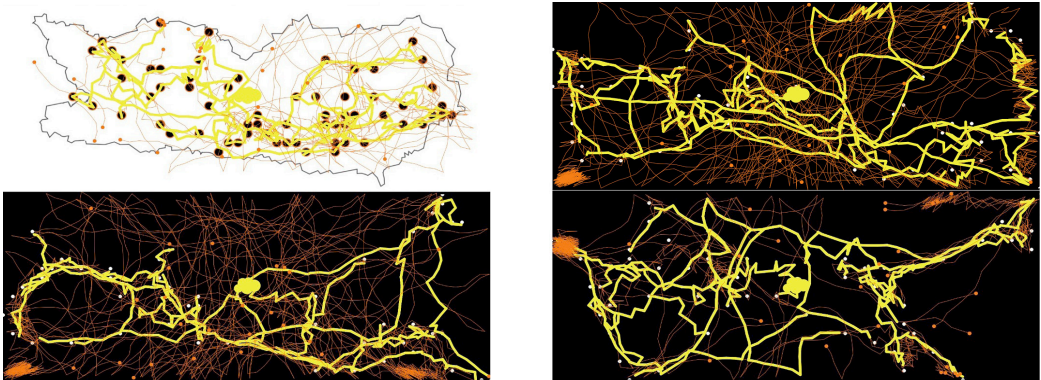


Figure 10:
Simulation Results:
Original Model (upper left), Force Atlas (upper right), Force Atlas2 (lower left) and Yifan Hu (lower right).
Source: own image

Abbildung 10:
Simulationsergebnisse:
Originalmodell (oben links), Force Atlas (oben rechts), Force Atlas2 (unten links) und Yifan Hu (unten rechts).
Quelle: eigenes Bild

Fig. 10

Orange lines represent the paths taken by the pseudopodia, and the shortest connections are highlighted with thicker yellow lines. In terms of performance metrics, the simulation achieved 29 TP, 1821 TN, 60 FP, and 43 FN. The precision was 0.33, the recall was 0.4, and the F1 score was 0.36. These results indicate that while the simulation successfully identified some actual connections, there is still room for improvement in accuracy.

A setup with Force Atlas node layout was simulated using the same settings, with an average runtime of 7 hours and 18 minutes each. The simulation achieved 34 TP, 1821 TN, 60 FP, and 38 FN, resulting in a precision of 0.36, recall of 0.47, and an F1 score of 0.41.

Ten simulations of the optimized network using Force Atlas2 were conducted with the same settings and initial values, averaging 1 hour and 42 minutes each. The simulations resulted in 33 TP, 1815 TN, 66 FP, and 39 FN, yielding a precision of 0.33, a recall of 0.46, and an F1 score of 0.39.

The Yifan Hu Proportional algorithm was used to optimize the network layout, keeping all other settings consistent with previous simulations. Ten simulations were conducted, averaging 5 hours, 14 minutes, and 13 seconds each. The simulations achieved 37 TP, 1818 TN, 63 FP, and 35 FN. The precision was 0.37, the recall 0.51, and the F1 score 0.43. While a higher F1 score is generally better, some FN can be acceptable as they indicate network thinning, potentially reducing repair efforts. Notably, the original network layout performed worse than the optimized models in this regard.

Table 4 provides an overview of the values for TP, TN, FP, FN, as well as the derived metrics precision, recall, and F1 score of all simulations conducted. A comparison of the F1 scores indicates that all simulations based on the rearranged graph models outperformed the simulation using the original network. Notably, the Yifan Hu model achieved the highest F1 score of 0.43, reflecting the most balanced trade-off between precision and recall.

Tab. 4

	Avg. Duration	TP	TN	FP	FN	Precision	Recall	F1 Score
Original network	00:50:05	29	1821	60	43	0.33	0.4	0.36
Yifan Hu	05:14:13	37	1818	63	35	0.37	0.51	0.43
Force Atlas 1	07:18:46	34	1821	60	38	0.36	0.47	0.41
Force Atlas 2	01:42:04	33	1815	66	39	0.33	0.46	0.39

Table 4:
Comparison of the
simulation results [22]

Tabelle 4:
Vergleich der Simulation-
ergebnisse [22]

Overall, this series of simulations used SISMO to implement Himmelstein's slime mold-inspired method [10] for energy network repair. However, the results often produced fragmented or unrealistic connections, limiting its practical use. To improve outcomes, we applied graph layout algorithms (Force Atlas, Force Atlas2, Yifan Hu Proportional), which reduced false links by reorganizing the network structure—though at the cost of longer computation times. While slime mold approaches may benefit complex network design, simpler methods like minimum spanning trees may be more suitable for targeted repair tasks. Our study is, to the best of our knowledge, the first to combine slime mold simulations with graph layouts, opening new possibilities for future research.

RESULTS

The accuracy of network reconstruction was assessed using precision, recall, and the F1 score. Among the tested models, the Yifan Hu layout achieved the highest F1 score of 0.43, followed by ForceAtlas with 0.41 and ForceAtlas2 with 0.39. The original network performed the worst, reaching only 0.36. Incorporating obstacles such as lakes and mountains improved the realism of the reconstructed networks. For example, the connection between Klagenfurt and Villach was more accurately represented in the model that included obstacles compared to the one without them. The scalability of the SISMO algorithm was evaluated by analyzing simulation times. Results showed that runtime increased with the number of food sources.

SISMO was able to identify critical areas for repair following a simulated EMP attack. The integration of graph layout algorithms, such as ForceAtlas and Yifan Hu, improved the accuracy of the repair planning, although this came at the expense of longer runtimes. Among the tested methods, the Yifan Hu model achieved the best balance, with a precision of 0.37 and a recall of 0.51. The performance metrics further illustrate the differences between models. The original network achieved 29 TP, 1821 TN, 60 FP, and 43 FN. The Yifan Hu model performed best, with 37 TP, 1818 TN, 63 FP, and 35 FN. ForceAtlas yielded 34 TP, 1821 TN, 60 FP, and 38 FN, while ForceAtlas2 resulted in 33 TP, 1815 TN, 66 FP, and 39 FN. Compared to the in vitro slime mold experiments, SISMO proved to be considerably faster—for instance, requiring only four days instead of nine days for reconstructing the KMG bus line—but was less accurate. The application of graph layout algorithms improved accuracy but also increased computational effort. Overall, SISMO demonstrates potential for network optimization and repair planning, particularly when natural obstacles are considered.

DISCUSSION

We have shown through experiments that the slime mold *P. polycephalum* and our simulation implementation SISMO can form and reconstruct networks by placing food sources as network nodes. Both the in vitro and simulation experiments validated this approach using Carinthian electricity transmission networks as well as public transportation systems in Carinthia and its regional cities. Results demonstrated that SISMO achieves comparable accuracy in transportation network reconstruction while providing significantly faster computational performance than in vitro experiments. The application of SISMO to Carinthian railway and bus networks highlights its potential for bio-inspired network optimization, particularly when incorporating natural obstacles such as lakes and mountains to improve path accuracy. Furthermore, our findings show that preprocessing network layouts with graph visualization algorithms such as ForceAtlas2 and Yifan Hu Proportional substantially improves SISMO's performance in network

recreation, especially in power transmission scenarios, with the Yifan Hu model achieving the highest F1 score of 0.43. The successful adaptation of SISMO for power grid repair planning after EMP attacks underlines the algorithm's versatility beyond transportation networks, offering a novel approach to critical infrastructure restoration that accounts for both geographical and topological constraints. While in some cases SISMO shows greater deviations from real networks than natural slime molds, its ability to account for obstacles and generate alternative configurations provides valuable insights for optimization scenarios unconstrained by existing infrastructure limitations. Integrating NetLogo's multi-agent simulation environment with bio-inspired algorithms thus proves effective for network planning research, enabling rapid prototyping and testing of diverse network scenarios with customizable parameters.

In comparison to other *Physarum*-inspired approaches, such as the work of Tero et al. [23] on the Tokyo railway system, SISMO offers a more flexible and computationally efficient solution. Tero demonstrated through biological experiments that *P. polycephalum* can reconstruct efficient, fault-tolerant networks and complemented these findings with a mathematical flow-based model, in which tubular conductance is reinforced or weakened depending on flux to achieve robust structures [23]. SISMO builds on these principles but translates them into a multi-agent simulation framework that operates significantly faster, allows stochastic exploration by independent agents, and provides greater adaptability. Unlike biological and purely mathematical approaches, SISMO can explicitly account for geographical obstacles and topological constraints, improve accuracy through network layout preprocessing, and extend beyond transportation to applications such as power grid restoration. Thus, SISMO represents a practical advancement over existing *Physarum*-inspired methods, providing concrete tools for network planning, optimization, and recovery.

FUTURE WORK

To expand the applicability of SISMO beyond network formation, future work will investigate how information could be transmitted within the simulated networks. This includes modeling communication or energy flow and analyzing how the network topology affects the efficiency and resilience of information propagation. Although SISMO has demonstrated promising results, further optimization of the algorithm is necessary to improve performance and scalability. This includes enhancing computational efficiency and reducing simulation time. To facilitate agent-based experimentation and integration with other models, the implementation of SISMO within the Python-based MESA framework is planned. This will allow for greater flexibility in testing, improved visualization capabilities, and easier comparison with other agent-based algorithms. At the application level, it is planned to test the SISMO algorithm as part of a method for Maximum Power Point Tracking, known as MPPT, in photovoltaic systems. The aim is to evaluate whether bio-inspired network formation can contribute to optimizing energy yield, particularly under dynamic environmental conditions.

REFERENCES

- [1] M. Dorigo, "Optimization, Learning and Natural Algorithms," Dissertation. Politecnico di Milano, Milan, 1992.
- [2] M. Dorigo, V. Maniezzo, and A. Colorni, "The ant system: an autocatalytic optimizing process," Milan: Politecnico di Milano, Milan, 1991.
- [3] M. Dorigo, V. Maniezzo, and A. Colorni, "Ant system: optimization by a colony of cooperating agents," In *IEEE transactions on systems, man, and cybernetics. Part B, Cybernetics: a publication of the IEEE Systems, Man, and Cybernetics Society* 26 (1), pp. 29–41, 1996, doi: <https://doi.org/10.1109/3477.484436>.

- [4] T. Nakagaki, "Smart behavior of true slime mold in a labyrinth," *Res. Microbiol.* 152 (9), pp. 767–770, 2001, doi: [https://doi.org/10.1016/S0923-2508\(01\)01259-1](https://doi.org/10.1016/S0923-2508(01)01259-1).
- [5] T. Nakagaki, R. Kobayashi, Y. Nishiura, and T. Ueda, "Obtaining multiple separate food sources: Behavioural intelligence in the *Physarum plasmodium*," *Proc. Biol. Sci.* 271 (1554), pp. 2305–2310, 2004, doi: <https://doi.org/10.1098/rspb.2004.2856>.
- [6] T. Nakagaki, H. Yamada, and A. Tóth, "Maze-solving by an amoeboid organism," *Nature* 407 (6803), p. 470, 2000, doi: <https://doi.org/10.1038/35035159>.
- [7] C. Gao, C. Liu, D. Schenz, X. Li, Z. Zhang, M. Jusup, et al., "Does being multi-headed make you better at solving problems? A survey of *Physarum*-based models and computations," *Phys. Life Rev.* 29, pp. 1–26, 2019, doi: <https://doi.org/10.1016/j.plrev.2018.05.002>.
- [8] C. R. Reid, T. Latty, A. Dussutour, and M. Beekman, "Slime mold uses an externalized spatial "memory" to navigate in complex environments," *Proc. Natl. Acad. Sci. USA*, 109 (43), pp. 17490–17494, 2012, doi: <https://doi.org/10.1073/pnas.1215037109>.
- [9] M. Bastian, S. Heymann, and M. Jacomy, "Gephi: An Open Source Software for Exploring and Manipulating Networks," *ICWSM* 3 (1), pp. 361–362, 2009, doi: <https://doi.org/10.1609/icwsml.v3i1.13937>.
- [10] D. Himmelstein, "Biologically inspired designs for restoring power transmission after an electromagnetic pulse attack," Online: https://figshare.com/articles/journal_contribution/Biologically_inspired_designs_for_restoring_power_transmission_after_an_electromagnetic_pulseattack/1488841?file=2182796
- [11] A. Awad, W. Pang, D. Lusseau, and G. M. Coghill, "A survey on physarum polycephalum intelligent foraging behaviour and bio-inspired applications," *Artif. Intell. Rev.* 56 (1), pp. 1–26, 2023, doi: <https://doi.org/10.1007/s10462-021-10112-1>.
- [12] J. D. Sweatt, Mechanisms of memory, 2nd ed. Amsterdam: Academic Press, 2010, Online: <https://ebookcentral.proquest.com/lib/kxp/detail.action?docID=583431>.
- [13] C. Durham and E. B. Ridgway, "Control of chemotaxis in *Physarum polycephalum*," *J. Cell Biol.* 69 (1), pp. 218–223, 1976, doi: <https://doi.org/10.1083/jcb.69.1.218>.
- [14] T. Latty and M. Beekman, "Irrational decision-making in an amoeboid organism: transitivity and context-dependent preferences," *Proc. Biol. Sci.* 278 (1703), pp. 307–312, 2011, doi: <https://doi.org/10.1098/rspb.2010.1045>.
- [15] E. Sinanović, "Simulation and mobility planning with a slime mold based algorithm," Master Thesis. Alpen Adria University of Klagenfurt, Klagenfurt am Wörthersee, 2022, Online: <https://netlibrary.aau.at/obvuklhs/download/pdf/8504854>, checked on 10/9/2025.
- [16] K. Wogatai, "SISMO (Simulation of Slime MOlds): a slime mold based algorithm and its application in traffic management," Master Thesis. Alpen Adria University of Klagenfurt, Klagenfurt am Wörthersee, 2022, Online: <https://netlibrary.aau.at/obvuklhs/download/pdf/8440422>, checked on 10/9/2025.
- [17] S. Li, H. Chen, M. Wang, A. A. Heidari, and S. Mirjalili, "Slime mould algorithm: A new method for stochastic optimization," *Future Gener. Comput. Syst.* 111, pp. 300–323, 2020, doi: <https://doi.org/10.1016/j.future.2020.03.055>.
- [18] E. Sinanović and K. Wogatai, "SISMO," Klagenfurt am Wörthersee: GitHub, Online: <https://github.com/barlum15/SISMO>, checked on 10/9/2025.
- [19] V. Bonifaci, K. Mehlhorn, and G. Varma, "Physarum can compute shortest paths," *J. Theor. Biol.* 309, pp. 121–133, 2012, doi: <https://doi.org/10.1016/j.jtbi.2012.06.017>.
- [20] F. S. Caparrini, "A General A* Solver in NetLogo," University of Seville, Seville, 2018, Online: https://www.cs.us.es/~fsancho/Blog/posts/General_A_Solver_NetLogo.md.html, checked on 10/9/2025.
- [21] K. Wogatai, J. Winkler, and W. Elmenreich, "A Graph-Based Approach for Applying Biologically-Inspired Slime Mold Algorithms for Repairing a Power Transmission Network after an Electromagnetic Pulse Attack," In: *2023 2nd International Conference on Power Systems and Electrical Technology (PSET)*, Milan, Italy, 2023, pp. 163–172. doi: <https://doi.org/10.1109/PSET59452.2023.10346595>.
- [22] Landes Rechnungshof Kärnten, "Kärnten Netz - Stromnetztarife und Herausforderungen. Landes Rechnungshof Kärnten," Klagenfurt am Wörthersee, 2020, Online: <https://lrh-ktn.at/berichte/kaernten-netz/>, checked on 10/9/2025.
- [23] Tero, S. Takagi, T. Saigusa, K. Ito, D. P. Bebber, M. D. Fricker, et al., "Rules for biologically inspired adaptive network design," *Science* 327 (5964), pp. 439–442, 2022, doi: <https://doi.org/10.1126/science.1177894>.

ABOUT THE AUTHORS

Kristina Wogatai
Networked and
Embedded Systems
University of Klagenfurt,
Austria kristina.wogatai@aau.at

Emir Sinanović
Networked and
Embedded Systems
University of Klagenfurt
Austria

Wilfried Elmenreich
Networked and
Embedded Systems
University of Klagenfurt
Austria



BACK TO CONTENT

SUPPLEMENTARY MATERIAL

Supplementary Tab. 1	
No.	Location
1	Klagenfurt main station
2	Grafenstein
3	St. Veit a.d. Glan
4	St. Martin-Sittich
5	Gummern
6	Villach main station
7	Rosenbach

Supplementary Tab. 3			
No.	Location	No.	Location
1	KUSW Außerfragant	33	USW Kirchengasse Klagenfurt
2	KUSW Feistritz	34	USW Klagenfurt Nord
3	KUSW Freibach	35	USW Klagenfurt Ost
4	KUSW Innerfragant	36	USW Klagenfurt West
5	KUSW Koralpe	37	USW Kleinkirchheim
6	KUSW Lavamünd	38	USW Landskron
7	KUSW Malta Hauptstufe	39	USW Lassendorf
8	KUSW Malta Unterstufe	40	USW Lienz
9	KUSW Schwabeck	41	USW Lieserhofen
10	KUSW Zirknitz	42	USW Oberdrauburg
11	KW Annabrücke	43	USW Obersielach
12	KW Edling	44	USW Radenthein
13	KW Feldsee	45	USW Rennweg
14	KW Forstsee	46	USW Seebach
15	KW Malta Oberstufe	47	USW Spittal
16	KW Reißeck	48	USW St. Andrä
17	KW Rosegg	49	USW St. Leonhard
18	KW Wölla	50	USW St. Margarethen
19	USW Auen	51	USW St. Martin
20	USW Bleiburg	52	USW St. Veit
21	USW Brückl	53	USW Steinfeld
22	USW Ettendorf	54	USW Treibach
23	USW Feldkirchen	55	USW Tröpolach
24	USW Ferlach	56	USW Villach Süd
25	USW Fürnitz	57	USW Vorderberg
26	USW Gallitz	58	USW Völkermarkt
27	USW Gmünd	59	USW Warmbad
28	USW Greuth	60	USW Wietersdorf
29	USW Gummern	61	USW Windischbach
30	USW Gurk	62	USW Wolfsberg
31	USW Hermagor	63	USW Würlach
32	USW Kamering		

Supplementary Tab. 2	
No.	Location
1	Heiligengeistplatz
2	Hauptbahnhof
3	Jergitschsteg
4	Steinerne Brücke
5	Luegerstraße
6	Universität Klagenfurt
7	Lakeside Park
8	Klagenfurt West

SUPPLEMENTARY TABLE 1
List of selected Stations of ÖBB.
Source: adapted from Table 6.1 [16]

Ergänzende Tabelle 1:
Liste der ausgewählten Stationen der ÖBB.
Quelle: angepasst nach Tabelle 6.1 [16]

SUPPLEMENTARY TABLE 2
List of selected Stations of KMG Bus Line
Source: adapted from Table 6.2 [16]

Ergänzende Tabelle 2:
Liste der ausgewählten Stationen der KMG-Buslinie C.
Quelle: angepasst nach Tabelle 6.2 [16]

SUPPLEMENTARY TABLE 3
List of relevant Power Plants and Substations of the Carinthian Transmission Network.
Adapted from [22]

Ergänzende Tabelle 3:
Liste der relevanten Kraftwerke und Umspannwerke des Kärntner Übertragungsnetzes.
Angepasst nach [22]