

Similarity of Path Networks constructed by Pedestrians and Physarum Polycephalum

M. Burghoorn
4113489

7,5 ECTS Bachelor Thesis Artificial Intelligence

Utrecht University

Supervisor: dr. G.A.W. Vreeswijk

Second assessor: dr. ir. D. Thierens

January 28, 2017

Abstract

The Physarum Polycephalum organism and the collective behavior of pedestrians are both able to construct minimal path networks. Models have been created of both concepts to simulate their individual network constructing behavior. This paper presents an implementation of a combined model, which simulates both the particle-based slime mold model and the Paths model. A qualitative analysis is conducted, which visually compares the networks constructed by the slime mold and the pedestrians. A measurement called “path coincidence” is introduced to measure the similarity of the constructed networks. This measurement is used to statistically support the results of the qualitative analysis. The results study the influences of four specific parameters. The visual comparison suggested that some of the parameters greatly affect the similarity in network construction. These results are supported by the statistical analysis, which found the same level of influence for each parameter. The experiment may provide some insight in the circumstances that affect the similarity of the path networks constructed by the particle-based Physarum Polycephalum model and the Paths model.

Table of Contents

1	Introduction.....	1
2	Models.....	3
2.1	Agents	4
2.2	Environment management	6
2.3	Nodes	7
3	Methods.....	9
3.1	Combined model.....	9
3.2	Path coincidence	10
3.3	Experiment setup and output.....	11
3.4	Statistical analysis	11
4	Results.....	12
4.1	withNodes	12
4.2	suckPheromone	14
4.3	Sensor- and rotation angle.....	16
5	Discussion and further research	20
6	References.....	21
	Appendix.....	22

1 Introduction

The Physarum Polycephalum is a single-celled myxomycete organism. Myxomycetes are a grouping of slime molds which develop a plasmodium during the vegetative stage. This plasmodium is capable of exploring nutrient-rich environments by growing exploratory branches. It can absorb the nutrients which enable it to grow and construct a transport network. The branches try to cover all available nutrient sources while minimizing the transportation cost of the nutrients through the network. Recently, these transport networks have been subject to research, because they exhibit complex patterning and adaptive behavior.

Examples of such researches cover maze solving, road planning and network optimization. In general, the Physarum Polycephalum organism is often used to tackle NP-hard problems. For example, Adamatzky has shown that the slime mold is capable of constructing spanning trees and other types of proximity graphs (Adamatzky, 2009). Nakagaki et al. have shown evidence of slime molds finding the shortest path between two selected points in a labyrinth (Nakagaki et al, 2001). The plasmodial branches of the organism in the dead ends of the maze are shrunk, which then eventually results in a minimum-length path between the selected points. Other research has shown that the plasmodium can reconstruct man-made road networks (Adamatzky and Jones, 2009). An interesting property of these approximated road networks is that they succeed in minimizing the nutrient transportation cost. In other words, they can find the most optimized network.

This network minimizing behavior is not unique to the Physarum Polycephalum. Recently, researchers have suggested that so-called “desire paths” might be a solution to walkways construction. Sivers stated that decisions should be made when the most information is available (Sivers, 2009). This was based on the idea to not make any walkways in advance, but to analyze the movement of pedestrians and pave those paths instead. The paths created by pedestrians have recently been researched by Nichols, who calls them “social desire paths”. These paths are often formed when pedestrians want better and more expedient routes (Nichols, 2014) and usually represent the shortest routes between two points.

The plasmodium created by the Physarum Polycephalum organism and the path network created by pedestrians seem to exhibit similar properties. Both are capable of constructing networks among two or multiple points. For the plasmodium, it has been suggested that it succeeds in building a minimal network. The pedestrian network will in most cases be minimal, because desire paths on flat surfaces usually serve as shortcuts. However, no research has been conducted on comparing these two concepts so far. It is difficult to compare these two concepts because their construction level is not the same. The Physarum Polycephalum is a tiny organism, whereas the pedestrian network is man-made. An unbiased

comparison could only be done when both networks are constructed in the same environment, under the same circumstances.

This unbiased environment can be realized by creating a computational model of the concepts. Computational models of the Physarum Polycephalum already exists and has been researched several times. The specific model addressed in this paper is a multi-agent system. This is a concept often used in the Artificial Intelligence field. The computerized system consists of multiple intelligent agents. The agents in the Physarum Polycephalum model are called particles and together they represent the slime mold. Research has been conducted using this particle-based model, to study the emergent properties of the slime mold. For example, Jones has shown that the construction of the network was strongly influenced by relative node position (Jones, 2011). Nodes being the points around which the network is constructed. However, this research does not include a comparison with the pedestrian model. This is not due to the lack of computational models for the walkways. In fact, a Paths model has recently been introduced by Grider (2015). This model is a multi-agent system as well, with pedestrians instead of particles. The collective behavior of the pedestrians is responsible for creating a walkway network.

As stated before, no research has been done comparing the two models or concepts. The question arises as to how similar these models are. Do the paths constructed by the Paths model overlap with the slime mold network? If so, what circumstances influence this overlap?

This paper presents a computational model, which compares the virtual plasmodium of the Physarum Polycephalum with the walkways constructed by the walkers. This is done by combining the implementation of the slime mold model as described by Jones (2011), with the implementation of the Paths model as presented by Grider (2015). This paper will show a qualitative and statistical analysis of the similarity between the models. A measure called “path coincidence” will be introduced to conduct this statistical analysis. The focus of this paper lies on four specific parameters, which are shown to influence the similarity of the networks.

Both models contain certain terminology which have to be explained to avoid ambiguity. This will be done in Section 2. The setup and details on the experiment will be presented in Section 3. This includes the details on how the two models were combined to allow comparison. Section 4 will present the qualitative and statistical results of the experiment described in the Section 3. This paper will be completed with a discussion. This includes a summary of the results and will present subjects for further research.

2 Models

This section discusses the Models and terminology used in this paper. Two existing models will be discussed, that were used to conduct the experiment described in Section 3. The first model is the Paths model as presented by Grider in (Grider, Netlogo Paths model, 2015). The second model is based on the particle-based Physarum Polycephalum model as described in (Jones, 2011).

First, a brief explanation of the main concept of both models is given. After that, the models are defined in more detail in several sections. Each section will describe the situation for both the Paths model and the Physarum model. This approach was chosen because this paper focusses on the similarity between the path networks constructed by the two models. To proof that this comparison is logical and viable, this approach will show that the models are built in the same unbiased environment. Moreover, it will show that the mechanisms of the models are very similar.

Section 2.1 will elaborate on the creation and behavior logic of the agents. Section 2.2 will explain how the simulation environment is managed. Finally, Section 2.3 will introduce the concept of nodes.

The Paths model is inspired on a phenomenon called “desire paths”. This term is used by landscape architects to describe dirt paths that develop overtime (Nichols, 2014). These dirt paths are created when travelers bypass sidewalks and prefer to take a shortcut. These paths are then positively reinforced as people tend to takes routes other travelers have taken before them.

The model of the Physarum Polycephalum is modelled as a multi-agent system, consisting of a large population of particles. All particles follow the same simple rules. These simple rules will be defined in the next sections. Together these particles create a network of chemoattractant paths, without a central control structure dictating how individual agents should behave.

Both models are run in a simulation environment. This environment consists of a two-dimensional plane. This 2D plane is a grid consisting of cells, which are called patches. These patches have multiple standard properties such as a color, but can also have model specific properties – see Section 2.2. The environment can be imagined as a torus, meaning that the edges of the surface are connected to each other. The simulation is run in two steps. First the simulation is initialized. During this initialization stage the patches and the agents are created and their properties are set to their base values. This is followed by the step stage of the simulation. During the first step the agents will follow the simple rules assigned to them. The observer asks the agents in a random order to process these rules, so no agents move simultaneously. Each step in the simulation repeats these simple rules, causing both models to eventually create paths. The details of these rules and path creation are given in the next few subsections.

2.1 Agents

This section will elaborate more on the behavior and movement of the agents used in both models. To differentiate between the agents used in the Paths model and the agents used in the Physarum model, they are given different names. The Paths model agents will be called walkers and the Physarum model agents will be called ants. If the word agent is used, this refers to both the walkers and the ants.

In both models one agent occupies only a single patch in the grid. This is the current location of the agent, specified in an x- and an y-coordinate. During the initialization stage, all agents are given a random location in the grid, as well as a random direction. The ants are colored orange and the walkers are colored blue. Once initialization is complete the simulation can be run and the first step is started. Every simulation step the same rules are processed. Because these rules differ per model, they will be addressed separately in the next few paragraphs. After all agents have processed and executed the rules, the next step is started. The simulation must be stopped manually, e.g. after the paths of both models have converged.

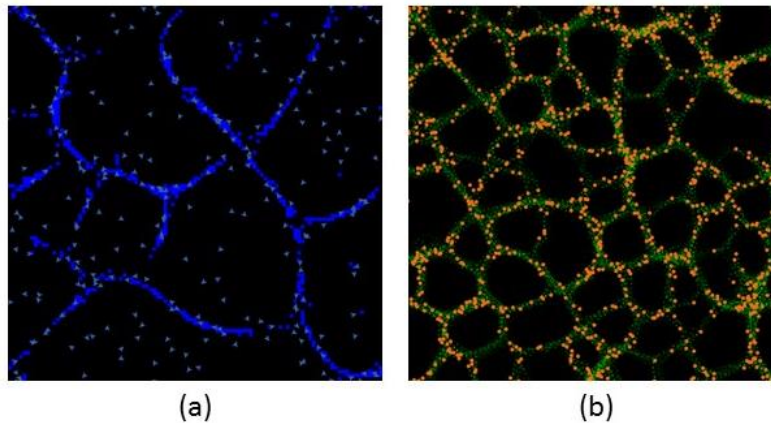


Figure 1: (a) Paths created by walkers, (b) Paths created by ants

The walkers each select a destination at random. This goal is one of the patches in the grid, unless there are nodes (so called buildings) present in the environment. Nodes will be discussed in Section 2.3 and are not considered in this subsection. On each step in the simulation the walker moves closer to its goal by a predefined step size (in pixels). However, if the walker detects a blue patch ahead on its way towards the goal, it will move there instead. It only detects patches in its current direction. The detection distance depends on the walker-vision-distance parameter. A blue patch represents an established path, which means that it is a commonly visited patch. Every patch, as explained before, has properties. In the Paths model, all patches have the variable popularity. This value indicates how popular a given patch is.

Every time a patch is visited by a walker, the popularity of the patch is increased by a constant amount. If this amount is above a certain threshold, the patch is turned blue. This means that established paths will be positively reinforced. However, if a patch is not visited for a while the popularity of this patch will decay, causing the patch to lose its blue color.

The rules that determine the behavior of the ants are a little more complicated than the walker rules. The patches in this model have a property called “pheromone”. This so-called pheromone is similar to the popularity property of the Paths model. The pheromone value on a patch represents a chemoattractant that attracts ants but is also created by ants. The ants leave a chemoattract trail as they walk, positively reinforcing the paths. This eventually creates a virtual plasmodium network that models the organic plasmodium very well (Jones, 2011).

The movement of the ant can be separated into two different stages, the sensory stage and the motor stage. During the sensory stage the ants sense the amount of pheromone on the patches in front of them. The ant uses three sensors of a specified length (sensor offset, SO). One sensor is placed directly in front of the ant. The two other sensors are placed to the left and right of the first sensor given a specified angle (sensor angle, SA). The ant senses the pheromone on all patches at the end of the sensors, within a specified range (sensor width, SW). The sensor (left, right or front) that detects the highest amount of pheromone affects the direction of the ant. If the most pheromone is detected by the front sensor, the heading of the ant does not change. If the left or the right sensor senses the most instead, the direction of the ant is rotated by a positive or negative rotation angle (RA) respectively.

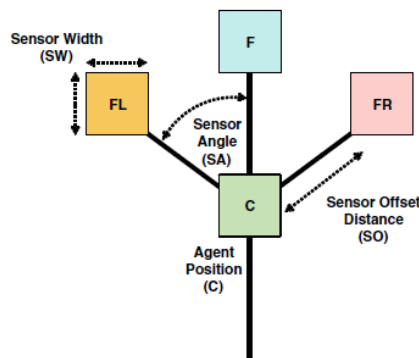


Figure 2: Sensors of the ants

After the sense stage is complete, the motor stage is executed. During this stage the ants attempt to move forward by the specified step-size (in pixels). Moving forward is only successful if the new patch is not occupied by another ant. If moving fails, the ant remains on the same patch and is given a new random heading. But if moving succeeds, pheromone is dropped on the new patch and the total amount of

pheromone on the patch is updated. The ants are addressed in a random order during both stages, to avoid possible collisions among ants. The pheromone on the patches is made visible by scaling a color to the amount of pheromone. This makes sure that ants leave a trail of slime as they are walking, creating the virtual plasmodium.

It should also be noted that during the sensing stage, all ants have a very small probability to be removed from the simulation. At that start of the simulation the removal rate will be high, because the ant population is big. But as the simulation continues the ant population shrinks and so does that rate of removal. This causes the virtual plasmodium to minimize. Walkers do not have this removal probability and collision among walkers is also allowed.

2.2 Environment management

After the agent updates are complete, the simulation will continue with the environment updates. These updates address the patches. As explained in Section 2.1, patches have two properties unique to the models, popularity and pheromone. These values are updated during this stage. When a path in Paths model becomes less visited, it will eventually disappear. This effect is created by diminishing the popularity value on the patches. This is done by asking the patches whether there are currently any walkers visiting and if not, to decrease the popularity value by the popularity-decay-rate. If a patch is blue and its popularity rate is less than 1, the patch is recolored to black.

The pheromone value on the patches is adjusted in two phases. First, the pheromone is diffused. This means that all patches drop some of their own pheromone value to their eight neighboring patches. This causes some paths to fade, but can also minimize the width of a path. The second phase asks the patches to evaporate a percentage of their pheromone. This pheromone is not added to any neighboring patches, but simply disappears. After this, the color of the virtual plasmodium is updated again using a color scale.

The *Physarum Polycephalum* model has a parameter which prohibits the border patches to have a positive amount of pheromone. This prevents the slime mold paths to bypass the grid walls. This is shown in Figure 3. If this parameter is FALSE, all patches are allowed to have a positive amount of pheromone. However, if this parameter is TRUE the pheromone is removed from the border. Specifically, during the environment updates the patches in this border set are asked to reset their pheromone value to 0. After the environment updates are complete, the simulation continues to the next step and repeats all steps described in Section 2.1 and 2.2.

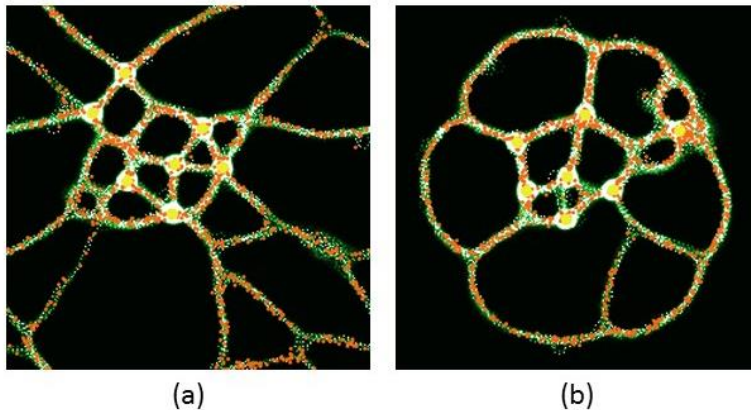


Figure 3: (a) suckPheromone False, (b) suckPheromone True

2.3 Nodes

One important parameter in both models is the presence or absence of nodes. Nodes can be defined as a point in the network at which paths intersect. There are two different types of nodes in both models: nodes created at random during the simulation and preplaced nodes. These preplaced nodes are placed at the setup of the simulation. When the term “nodes” is used in this paper, its referring to preplaced nodes.

In the Physarum model a total of seven nodes is placed at setup. The amount is usually variable, but in the conducted experiment the number of nodes will be fixed. The nodes in this model are called food sources or food. The food sources are bigger than the ants and cover multiple patches. The nodes are objects on the grid, shaped as circles. On creation, the simulation creates and places the food sources close to each other. It is, however, not possible for two food sources to intersect. After the food has been positioned, all patches in its radius are given a positive amount of pheromone. This makes the food attractive to ants. When ants pass a food source, they become blessed. Only when an ant is blessed, it is allowed to drop pheromone. Section 2.1 explained that ants drop pheromone when moving forward. However, this only happens if the ants are blessed. If an ant has not passed a food source yet, it will not drop pheromone on the patches it comes across. The amount pheromone that an ant drops on a food source patch is higher than the amount dropped on a normal patch. This amount is based on the regular pheromone-deposit-rate of the ant multiplied by the pheromone-factor-at-food. Because the amount of pheromone present on food patches is higher than on regular patches, the virtual plasmodium paths are created around these nodes. Therefore, nodes are an important influence in path creation. When preplaced nodes are absent in the model, the ants are pre-blessed. This allows ants to drop pheromone from the start of the simulation. It should also be noted that the ants and the food sources both emit the same

chemoattractant. This is a signification simplification with respect to the organic plasmodium (Jones, 2011, p. 1350).

It is also possible to add nodes to the Paths model simulation. These nodes are called “buildings”. The buildings are placed on the exact same spots as the food sources. The reason behind this is explained in Section 3.1. A building covers only one patch and is visualized by coloring the corresponding patch red. Food works differently. Food objects cover multiple patches and have a visible shape on the grid. When two or multiple buildings are present in the Paths model, the behavior of the walker changes. In a simulation without nodes – as explained in Section 2.1 – the walkers select a random patch as their goal. If nodes are present, the walkers select one of the buildings as their goal instead. The difference in path creation is clearly visible, see Figure 4. The blue patches, which indicate an established path, are still preferred over black patches. Therefore, they still affect the positively reinforce the path creation.

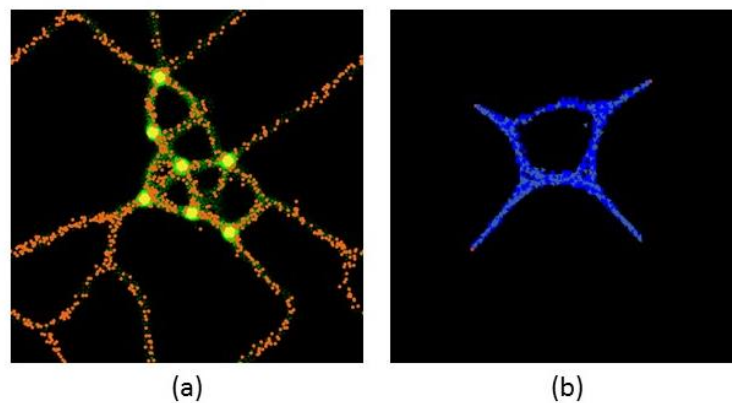


Figure 4: (a): *Physarum Polycephalum* model with nodes (b): *Paths* model with nodes

3 Methods

As stated in the introduction, the goal of this paper is compare the paths created by the Physarum Polycephalum model and the paths created by the Paths model. Specifically, the goal is to analyze what parameters influence the path overlap between the models. The two models have been combined and their overlap has been tested using the path coincidence measurement. This measurement will be explained in Section 3.2. How the models were combined is elaborated in Section 3.1. Using this combined model and the new measurement an experiment was executed. The details on this experiment are described in Section 3.3. A statistical analysis was conducted to support the results from this experiment. The details of this analysis are given in Section 3.4. The answers to the research question will be given and interpreted in Section 4, “Results”.

3.1 Combined model

This paper focusses on comparing the Physarum Polycephalum model with the Paths model. The best way to this visually, is to combine the two models. The combined model that was created, runs both models in the same simulation environment. It simulates the construction of the pathways created by walkers and the construction of the virtual plasmodium created by ants. Both individual models use the color property of the patches, as explained in the previous section. Due to this, a few changes had to be made to the original coloration of the models. This does not affect any of the rules or agent behavior in either of the models.

The patches in the simulation environment have been given three new properties, an R, G and B value. These values correspond to the red, green and blue factor in a specific color. All values range from 0 to 255. If a patch is given the values [R: 0, G: 255, B: 0], its colors will be green. If a patch is given the color [R: 0, G: 255, B: 255], its color will be cyan. These RGB values were used to draw the models on top of each other. The virtual plasmodium network created by the ants is mapped on the G-value. This G-value is scaled to the amount of pheromone on a patch. The network created by the walkers is mapped on the B-value. If the amount of popularity on a patch is above the threshold, the B-value of this patch is set to 255. Otherwise the B-value is equal to 0. Using these new properties, the slime mold paths will be green and pathways will be blue. If the paths from both models intersect, the patches in this intersection will color cyan. An example of this is shown in Figure 5.

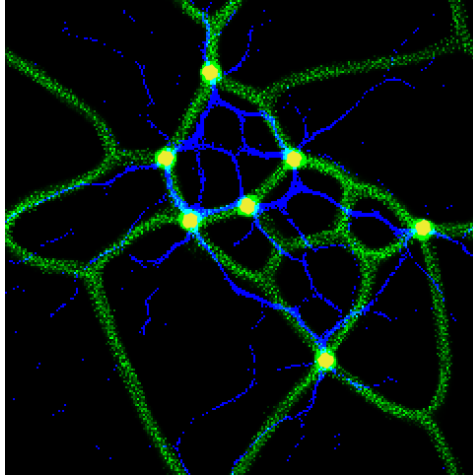


Figure 5: Green paths: Slime mold, Blue paths: Pathways, Cyan paths: Both models

Besides the coloration change in this combined model, a change to nodes positioning was made as well., It is important that the nodes of both models are in the same spots to be able to qualitatively compare the coincidence of the models. During the initialization stage, the food sources are added before the buildings. After the food sources have been placed, the buildings are added on the exact same positions as the food nodes. The buildings are, however, not visible in the combined simulation because they are visualized using the R-value of the patches. The food sources cover these red patches, because they are drawn as objects on top of the grid. This does not cause any complications, because the locations of the buildings are known due to food source positioning.

3.2 Path coincidence

The coincidence of the paths created by the models, will be measured by a value called “path coincidence”. The Models section explained that the patches own two path specific values, called “popularity” and “pheromone”. These two properties are used to calculate the path coincidence of one simulation run. Each patch in the two-dimensional plane is asked to multiply its amount of pheromone with its amount of popularity. After that, all multiplied patch values are summed up. This total is divided by the number of patches in the simulation. The final outcome represents the path coincidence.

The path coincidence value will be high for simulations with a great overlap between the paths of both models. For example, if one patch has a high amount of pheromone and a high amount of popularity, multiplying these results in a high path coincidence value. However, if one of the values, pheromone or popularity, is absent or very small, the path coincidence will not be high either. For this reason, path coincidence seems to be a great measurement for calculating the overlap between paths.

3.3 Experiment setup and output

Section 3.1 explained how the models were combined, but not how the experiment was setup. Interesting parameters in the simulation were varied during the experiment while others were kept constant. The Physarum Polycephalum model has a lot of optional parameters, which are not addressed in this paper. But for reference, all parameter values can be found in appendix A. There are a few important variable parameters in this experiment. Each of them will be briefly explained.

withNodes – This parameter indicates whether nodes are present in the simulation or not. The parameter can have two possible values, True and False. If the parameter is set to True, nodes (food sources and buildings) are placed in the simulation during the initialization stage. If the parameter is set to False, no nodes will be placed in the simulation. This is the only parameter that also effects the Paths model.

suckPheromone – As stated in Section 2, this parameter either allows or prohibits border patches to have a positive amount of pheromone. This parameter is also a boolean, meaning it has two possible values, True and False. If set to True, the pheromone is removed from the border and slime mold paths cannot bypass the edges of the grid. Otherwise, the slime mold model has no restrictions.

SA and RA – Also known as sensor angle and rotation angle respectively. These parameters have already been explained in Section 2.1. During the experiment, the parameters were given three possible values: 20, 45 and 60. They vary independently, for example SA can be set to 20 while RA is set to 45.

The experiment consisted of several runs, varying these parameters. All possible parameter combinations were tried, meaning there is total of 36 different runs ($2 \times 2 \times 3 \times 3 = 36$). Each run was stopped after 1000 steps, as the paths seemed to have converged after this amount of time for most parameter settings. Every combination was repeated 10 times, resulting in 360 different outputs. The output consists of several categorical and numerical values, including the parameter settings. An important and notable output value is the path coincidence as explained in the section above. Besides a table containing all values, a snapshot was taken after every run. This snapshot provides a clear picture of what the networks of both models looked like at the end of the simulation run. These snapshots are used to qualitatively analyze the overlap of the paths. The path coincidence outputs were used for the statistical analysis, which is addressed in the next subsection.

3.4 Statistical analysis

The 360 different numerical results are difficult to interpret when viewed as a table. Therefore, a statistical analysis was done. This analysis summarizes the results and can visualize the effects of all parameters. Besides visualizing the data, a statistical analysis is also necessary to confirm that the qualitative analysis of the snapshots is interpreted correctly. It also ensures that the apparent influences

are significant. Furthermore, as specified a set of 360 snapshots was used to conduct the qualitative analysis. This set is only a sample of the infinite set of snapshots. Therefore, statistical analysis is necessary to estimate the influences on the entire set, using this sample set. In other words, it is able to obtain a best estimate of the influences within the infinite set of runs.

The best approach to find the most influencing parameters is by creating a predictive model. The varying parameters, `withNodes`, `suckPheromone`, `SA` and `RA`, are used as the variables of this model. The path coincidence is used as the predicting value of this model. Because there are both categorical and numerical variables and since the output is numerical, the most suitable method is multiple regression. The model output includes coefficients for all parameters. These coefficients will be used to support the qualitative snapshot analysis in the results. Other than coefficients, the model was visualized to show the effect of different parameters of the path coincidence. These visualizations will also be used as supporting evidence for the observation in the Results section.

4 Results

In this section the results of the experiment, defined in the previous section, will be presented. The results have been divided in multiple sections. Each section will address observations based on one specific parameter, with the exception of Section 4.3. This section is instead based on two parameters, `SA` and `RA`, because they seem to influence each other. Section 4.1 will address the influences of the `withNodes` parameter and Section 4.2 will discuss the effect of the `suckPheromone` parameter. Every observation is based on multiple visual snapshots obtained by the experiment. The snapshots have chosen manually. The suggested observations are then supported by the statistical analysis on the path coincidence.

4.1 `withNodes`

The most notable result is the effect of the `withNodes` parameter. It is clearly visible that the paths overlap the most when the `withNodes` parameter is set to `True`, meaning that nodes are present in the simulation. This effect is clearly visualized in Figure 6. As shown, the walker paths lie on top of the slime mold paths in Figure 6 (a) and (b). The opposite is true for Figure 6 (c) and (d), where the walker paths seem disconnected and chaotic. The overlap between the models in the last two figures is presumably accidental.

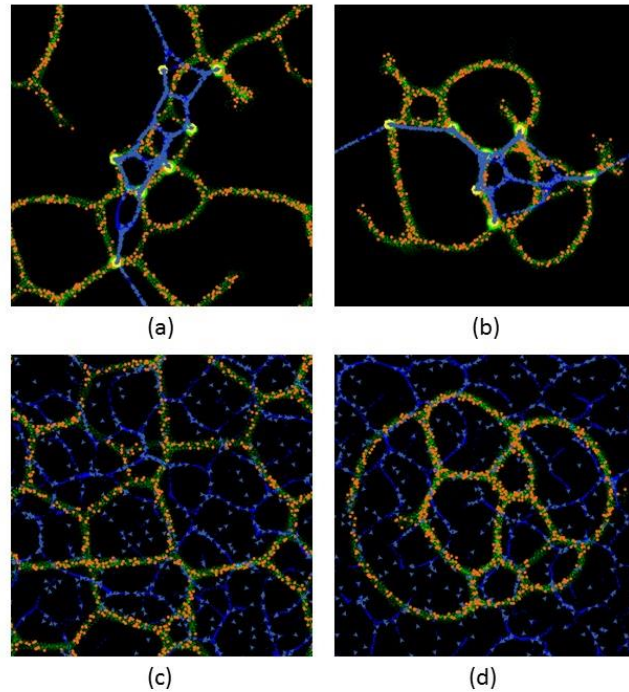


Figure 6: Snapshots of the influence of the `withNodes` parameter
 (a): `withNodes True, suckPheromone False` (b) `withNodes True, suckPheromone True`
 (c): `withNodes False, suckPheromone False` (d): `withNodes False, suckPheromone True`

The large influence of the `withNodes` parameter could have been expected by analyzing the algorithms of both models. As stated in the Models section, the walkers choose a random building as their goal when nodes are present. This causes the walker network to be constructed around the nodes. Nodes are placed in the center of the simulation grid and close to each other, which causes the walker network to only cover a small area of the grid. The ants are also heavily influenced by the presence of the nodes, because they contain a high level of pheromone. Therefore, the paths created by the ants are also constructed around these nodes. This is in contrast with the networks shown in Figure 6 (c) and (d). Both the ants and walkers have created paths covering the whole grid. The models cannot communicate with each other, so the location of their networks is independent of each other. This causes the overlap of the paths to be purely random and accidental.

The `withNodes` parameter seems to be independent of the other parameters, `suckPheromone`, `SA` and `RA`. In other words, in all snapshots the simulations with nodes visually overlap more than the simulations without nodes. Figure 6 also shows this specifically for the `suckPheromone` parameter. Figure (a) and (c) have `suckPheromone` turned off, whereas Figure (b) and d have pheromone removed from the border. In both cases, the paths clearly overlap more when `withNodes` is set to `True`. This phenomenon was found in all snapshot results, the pictures in Figure 6 were just selected because they illustrated this

effect the clearest. This result suggests that the withNodes is the most influential parameter compared to the other three.

Statistical evidence will be provided, to support the observation made above. As stated before, multiple regression was used, based on four parameters. The output of this regression model consists of coefficients. A large positive coefficient was found for the withNodes parameter. This implies that the withNodes parameter has a significant effect on the path coincidence value. The coefficient was also the largest compared to the coefficients of the other parameters. Therefore, it is suggested that the withNodes parameter is most influential parameter. This agrees with the visual observations stated before. All coefficients of the multiple regression function can be found in appendix B.

The influence of withNodes is visualized in Figure 7. The figure shows the RA plotted against the path coincidence value. The influence of RA will be discussed in a later subsection. The most important part in this figure is the difference between the blue and the red line. These lines are the different levels of the withNodes parameter, True and False. It is clearly visible that the blue line rises way above the red line. This suggests that the path coincidence of withNodes is True, is much greater than the path coincidence of withNodes is False. Moreover, the blue and red area around the lines, show the 95%-confidence interval of the path coincidence. Both bands are rather small, meaning that the lines are significant and a good predictor for the path coincidence based on the withNodes parameter. Therefore, this statistical analysis also implies that the withNodes parameter is the most influencing factor.

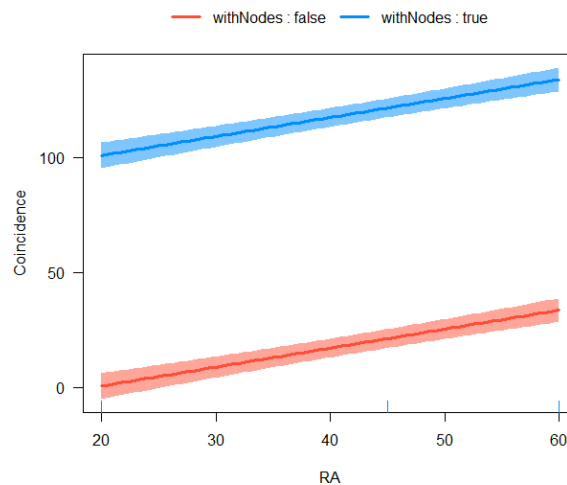


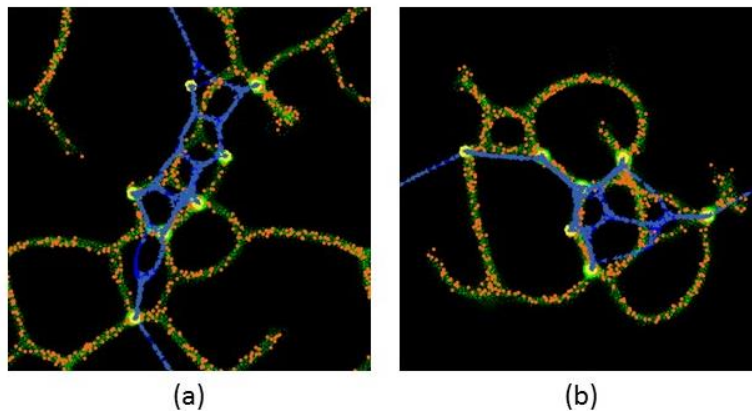
Figure 7: Path coincidence plotted against RA with confidence bands of 95% with different withNodes values

4.2 suckPheromone

The previous section stated that the influence of the withNodes parameter is independent of the other parameters. However, this does not suggest that the other parameters do not influence the path coincidence value at all. This subsection will discuss the influence of the suckPheromone parameter. For

clearer visualization purposes, only snapshots of simulations with nodes were selected. The statistical analysis is still based on all data.

The snapshots in Figure 8 suggest that the `suckPheromone` parameter has no significant influence on the path coincidence. In Figure (a) and (b), the pathways seem to lie on top of the virtual plasmodium network. Visually, the simulations seem to overlap equally. As stated before, the walkers of the Paths model tend to construct a minimal network around the nodes. Since the nodes are always located close to the center of the two-dimensional grid, this network is also formed in the center of the grid. The `suckPheromone` parameter prevents pheromone from stacking on the border of the grid, but it has no effect on the pheromone in the center. Therefore, the virtual plasmodium network is still created around the nodes located in the middle. Because both networks are created around these nodes, independent of the `suckPheromone` parameter, the paths of both models will visually overlap each other in the same degree. This suggests that the `suckPheromone` parameter has no influence on the path coincidence value.



*Figure 8: Snapshots of the influence of the `suckPheromone` parameter
(a) `suckPheromone` False (b): `suckPheromone` True*

The statistical analysis also suggests that the `suckPheromone` parameter has no significant influence on the path coincidence. Multiple regression found a relatively small and negative coefficient for the parameter. The regression model has been visualized in the Figure 9, to show that this coefficient is indeed insignificant. Again, the RA parameter has been used in the plot, but it will not be considered yet. The colored lines illustrate the different levels of the `suckPheromone` parameter. The red line appears above the blue line in the grid, implying that the `suckPheromone` is False has a positive effect. However, the figure also shows the 95%-confidence interval of both lines, illustrated with the colored areas. Both areas are relatively large and cover the line of opposite color. This implies that the influence of the `suckPheromone` parameter is indeed insignificant and negligible.

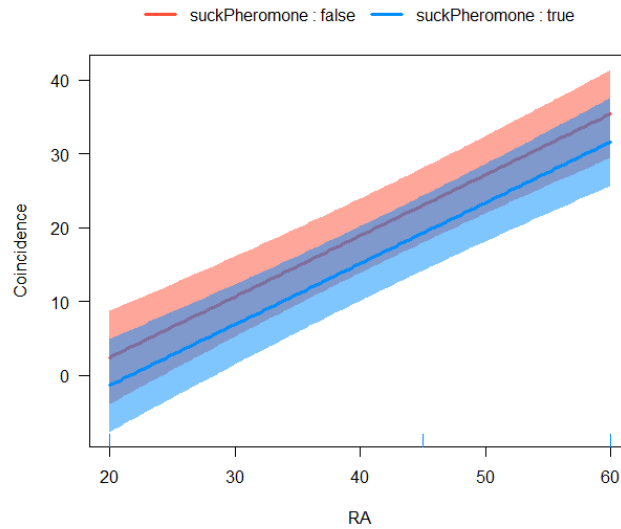


Figure 9: Path coincidence plotted against RA with confidence bands of 95% with different suckPheromone values

4.3 Sensor- and rotation angle

The remaining two parameters, SA and RA, will not be considered individually. As explained in section 2, the SA is used to sense pheromone in angled directions. If the most pheromone is detected to the left or right of the ant, the heading of the ant is then rotated by the RA. This implies that SA and RA are connected. Therefore, the observations made with a varying RA might also depend on the current SA or their ratio. The results presented in this section will vary the SA or RA while the other parameter is kept constant. The interval in which the parameters can vary is [20, 45, 60]. The constant parameter is kept at 45. First, the results for a varying SA parameter will be presented. Followed by the results for a varying RA parameter. This subsection will be completed with an overall observation of the RA parameter.

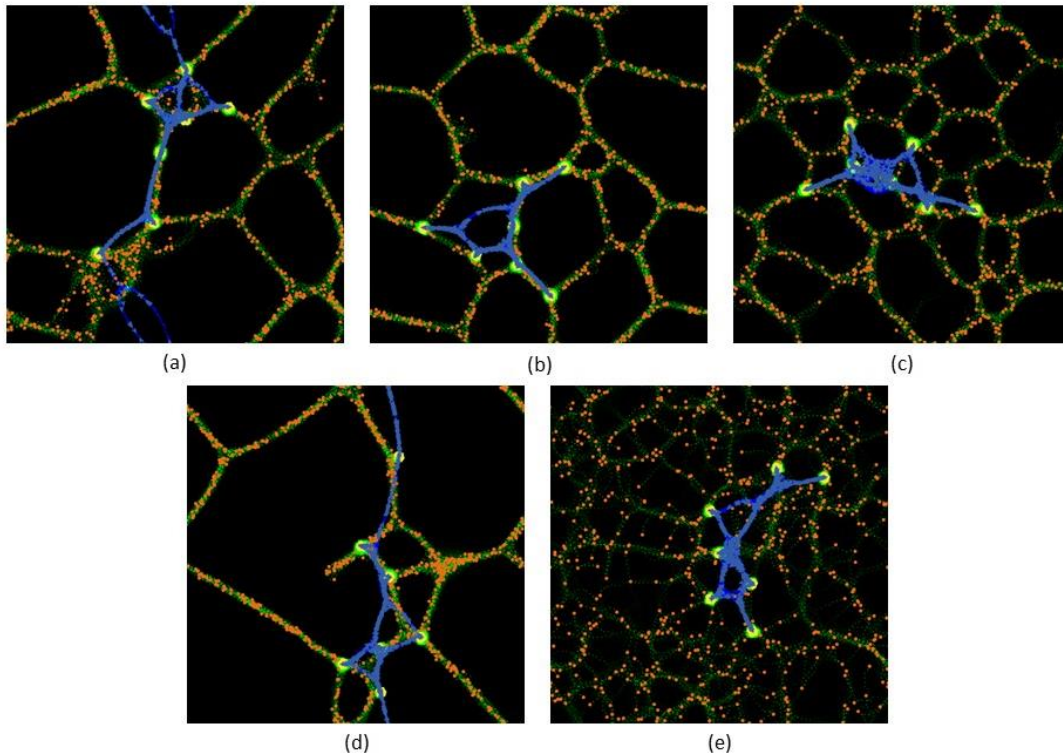


Figure 10: Path Coincidence with varying SA and RA
 (a) SA 20, RA 45. (b) SA 45, RA 45. (c) SA 60, RA 45. (d) SA 45, RA 60. (e) SA 45, RA 20.

Figure 10 shows all combinations for SA and RA using the interval [20, 45, 60] and a constant of 45. First, the top row will be discussed. In this row of pictures the RA is kept constant at 45, while the SA varies. Figure (a) has the largest lacunae in the virtual plasmodium network compared to Figure (b) and (c). In fact, the lacunae seem to become smaller as the SA rises. Simulations with larger lacunae in the virtual plasmodium seem to have a greater path coincidence compared to simulations with smaller lacunae. These small lacunae in the virtual plasmodium are associated with multiple shorter paths which result in a less minimized network. This is the exact opposite of the network created by walkers, which is extremely minimal. This is also shown in Figure (a) to (c), the paths created by the Paths model lie perfectly on of the virtual plasmodium, with the exception of a few path branches. Whereas, Figure (c) shows lots of smaller lacunae which are not covered by the pathways. These observations suggest that a smaller SA with respect to RA achieves the best results.

The next step is to vary the RA parameter. Figures (d) and (e) have a constant SA of 45 and a RA of 60 and 20 respectively. The third case of both angles equaling 45, was already shown in Figure (b). Figure (e) is in agreement with the previous observation. Its RA is smaller than its SA and this seems to cause further disorientation in the network. The lacunae have become so small that it is difficult to

separate established paths. The virtual plasmodium shows very slight shades of green and looks chaotic. The paths of both models are completely different in construction, which results in a very low path coincidence. In Figure (d), the opposite is true. The lacunae in the virtual plasmodium have increased in size. The slime mold network seems to have minimized compared to Figure (a). In Figure (d), the two networks seem to share a lot of their paths, except for the path that bypasses the wall. The walkers network also has a path in the middle that is not covered by the virtual plasmodium. These observations combined suggest that a simulation with $SA < RA$ seems to create the greatest path coincidence.

Through snapshots it is difficult to see the difference in path coincidence between Figure (a) and (d). Therefore, the suggestions made, will have to be confirmed by statistics. Another notable observation that statistics might confirm is if a greater RA parameter results in better path coincidence overall. Visually, the best presenting networks seem to be Figure (a) and (d). These networks have a large RA compared to its SA. The networks that seem to achieve the least overlap are Figure (c) and (e), which both have a small RA compared to their SA. Figure (b) seems to be right in the middle, which could indicate that RA has a linear positive effect on the path coincidence.

A plot has been created (see Figure 11), to confirm whether RA and path coincidence are linearly connected. It shows RA plotted against the path coincidence value. This plot is based on the same multiple linear regression model mentioned in the other results. Figure 11 shows non-horizontal line, confirming a positive effect of RA on path coincidence. In other words, a higher RA will result in a higher path coincidence value. It should be noted that the grey area around the blue line visualizes the 95%-confidence interval. The grey area is relatively large, meaning RA may not always affect the coincidence as much. This is probably caused by the settings of the other parameters. Because the large confidence bands may be caused by the SA parameter, an extra small statistical test was run. An extra column was added to the data which stated if the SA was great than (GT), equal to (EQ) or less than (LT) the RA parameter. This column was then added to the multiple regression function as an extra parameter and plotted. Figure 12 shows the results. Again, the RA was plotted against the path coincidence. However, in this plot several levels were used: EQ, GT and LT. As shown in the figure, the lines lie rather close to each other. Moreover, the confidence bands of all lines are enormous and they overlap each other greatly. This suggests that ratio of the SA and RA parameter do not seem to influence the path coincidence significantly. The RA parameter does, however, still show a positive effect in all three cases.

It should also be noted that the RA (and SA) have only been tested for an interval of [20, 45, 60]. Meaning, that the linear effect of RA on path coincidence may only exists within this interval. No statements can be made about the effect of RA (or SA) outside this area. This is subject for further research.

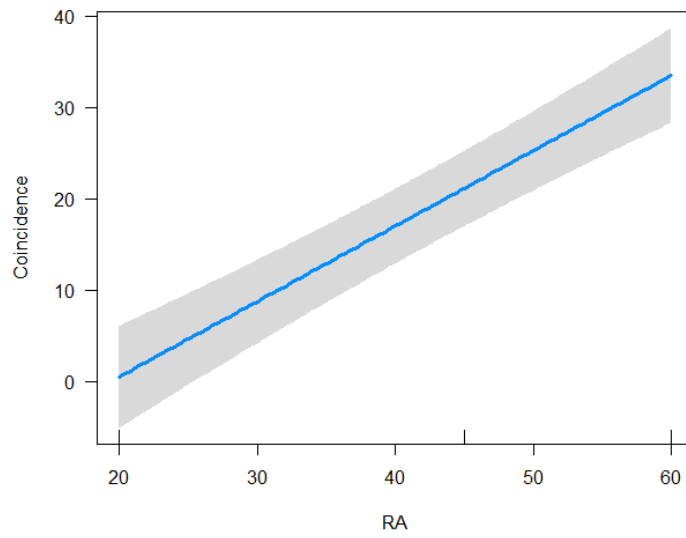


Figure 11: Path coincidence plotted against RA with a confidence band of 95%

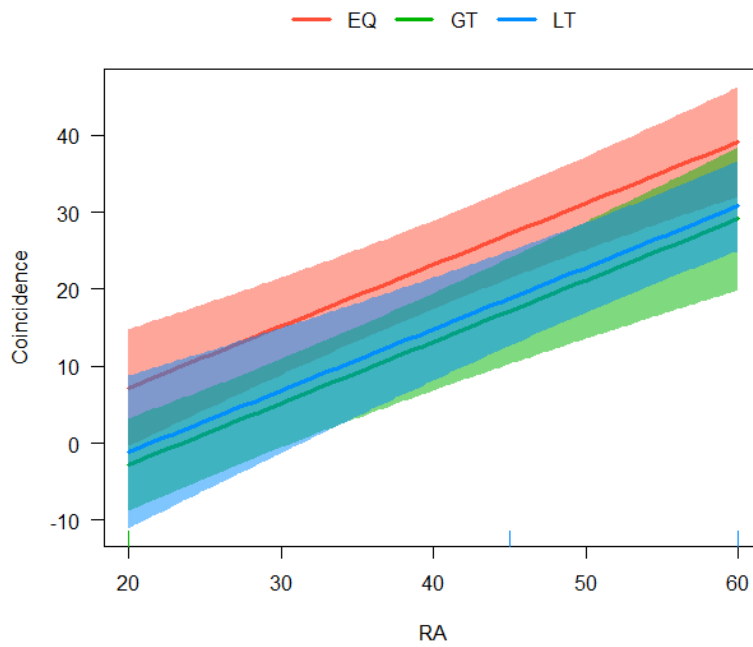


Figure 12: Path coincidence plotted against RA with confidence bands of 95% with different SA/RA-comparisons

5 Discussion and further research

Recently, the Physarum Polycephalum organism has been subject to research because of its unique capabilities. Its emergent properties allow the organism to construct minimal transportation networks. A particle-based model based on the plasmodium, also shows these properties. Research on this model suggested that network construction is influenced by relative node position. Constructing minimal networks is not unique to the slime mold. It is suggested that the collective behavior of pedestrians can also construct this. A model simulating this collective behavior has already been created.

In this paper the particle-based Physarum Polycephalum model and the model simulating pedestrian movement were combined. This combined model allowed the two similar networks to be compared in an unbiased environment. This thesis presented a qualitative analysis consisting of snapshots, visually comparing the two similar network constructions. Besides, a qualitative analysis, a statistical analysis was conducted using a self-defined measurement called “path coincidence”. The presented results focused on the influences of four specific parameters: withNodes, suckPheromone, SA and RA.

Results of the experiment suggested that the withNodes parameter is the most influencing parameter. When nodes were present in the simulation, the path coincidence of the networks was drastically increased. The influence of the suckPheromone parameter seemed to be insignificant and negligible. This implies that removing pheromone from the border does neither increase or decrease the path coincidence. The results also suggested that the ratio between SA and RA parameters does not affect the similarity of the networks. However, a positive linear effect was found when analyzing the RA parameter separately.

It should be noted that only four parameters were used. The found results can only suggest the influence of these four parameters on the path coincidence. To find what other circumstances optimize the similarity between the networks, more research has to be conducted. This is subject for further research. Another subject for research is the interval of the SA and RA parameters. In the conducted experiment the parameters were only varied in a finite interval of [20, 45, 60]. Therefore, the suggested influences of these parameters are only applicable for this interval. No implications can be made about the influence of the SA and RA parameters outside this interval. Future work could research whether the found effects also exist in wider angle ranges.

6 References

Adamatzky, A. (2009). Developing proximity graphs by Physarum Polycephalum: Does the plasmodium follow the Toussaint Hierarchy? *Parallel Processing Letters*, 19, pp. 105-127.

Adamatzky, A. (2010). Physarum Machines, Computers from Slime Mould. *World Scientific Publishing Company*, 1st ed., volume 74.

Adamatzky, A. & Jones, J. (2009). Road planning with slime mould: If Physarum built motorways it would route M4/M76 through Newcastle.

Grider, R. (2015). NetLogo Paths model. Center for Connected Learning and Computer-Based Modeling, Northwestern University, Evanston, IL.

Available at: <http://ccl.northwestern.edu/netlogo/models/Paths>.

Jones, J. (2010). Characteristics of pattern formation and evolution in approximations of physarum transport networks. *Artificial life*, 16 (2). p. 127-153.

Jones, J. (2011). Influences on the formation and evolution of Physarum polycephalum inspired emergent transport networks. *Natural Computing*, 10 (4). p. 1345-1369.

Jones, J. (2008). The Emergence and Dynamical Evolution of Complex Transport Networks from Simple Low-Level Behaviours.

Kohlstedt, K. (2016). Least Resistance: How Desire Paths Can Lead to Better Design, *99%Invisible*

Nakagaki T. et al., (2001). Path finding by tube morphogenesis in an amoeboid organism. *Biophysical Chemistry*, 92, p. 47-52.

Netlogo Slime mould (Physarum Polycephalum) model as described in (Jones, 2011).

Nichols, L. (2014). Social Desire Paths: An Applied Sociology of Interests. *Social Currents*, Vol. 1(2) 166-172.

Sivers, D. (2009). Let pedestrians define the walkways.

Available at: <https://sivers.org/walkways>

Wilensky, U. (1999). NetLogo. Center for Connected Learning and Computer-Based Modeling, Northwestern University, Evanston, IL.

Available at: <http://ccl.northwestern.edu/netlogo/>

Burghoorn, M. (2017). *Thesis model of combination Physarum Polycephalum and Paths*

Available at: <https://github.com/MaaikeBurghoorn/Path-Coincidence-Thesis>

Appendix

A. Parameter settings experiment

Parameter Name	Description	Experiment Value
Nr-of-ants	Size of ant population	1000
Probability-of-death	Probability of an ant dying each step	4.1E-4
Ant-step-size	Step size of the ants	3.00
Sensor-offset	Length of the sensors	10
Co-location-allowed	Allows ants on the same patch	False
Start-dispersed?	Ants start at random locations	True
Nr-of-food	Amount of food sources	7
Radius-of-food	Size of food sources	4.0
Pheromone-deposit-rate	Amount of pheromone dropped per ant step	50
Evaporation-rate	Amount of pheromone evaporated per step	0.90
Diffusion-rate	Amount of pheromone diffused per step	0.05
Pheromone-factor-at-food	Multiplied with pheromone-dropped	100
Popularity-decay-rate	Amount of pheromone decayed per step	4
Popularity-per-step	Popularity added to a patch per walker step	20
Minimum-route-popularity	Popularity coloration threshold	50
Walker-count	Size of walker population	500
Walker-vision-dist	Distance walker senses popularity ahead	10

B. Multiple regression details

Parameter	Coefficient
Intercept	-14,279
withNodesTRUE	100,357
SA	0,004
RA	0,825
suckPheromoneTRUE	-3,777

To carry out the multiple regression, a linear function was fitted as follows:

$$\text{path coincidence} = \beta_0 + \text{withNodesTRUE} * \beta_1 + \text{SA} * \beta_2 + \text{RA} * \beta_3 + \text{suckPheromoneTrue} * \beta_4$$

$\beta_1 - \beta_4$ are the coefficients. The values found through multiple regression are given in the table above.

For further details on the multiple regression, see <https://github.com/MaaikBurghoorn/Path-Coincidence-Thesis>