www.itcon.org - Journal of Information Technology in Construction - ISSN 1874-4753

EVALUATION OF METRO LINES WITH SWARM INTELLIGENCE APPROACH

SUBMITTED: January 2021
REVISED: December 2021
PUBLISHED: August 2022
EDITOR: Žiga Turk
DOI: 10.36680/j.itcon.2022.039

Sena Kaynarkaya, Research Asistant,
Istanbul Technical University;
kurkcuoglus@itu.edu.tr

Gülen Çağdaş Professor Doctor,
Istanbul Technical University;
cagdas@itu.edu.tr
ORCID: 0000-0001-8853-4207

SUMMARY: *Metro lines, which have become one of the most important transportation routes for today's cities, require long design and implementation processes. Their design load is mostly due to detailed engineering calculations, have to expand to meet the needs of the ever-increasing population. The behavior of slime moulds to find the shortest path they use to reach the food source has inspired models created specifically to test the accuracy of highway routes. The hypothesis of this study is that the swarm behavior of slime moulds and the routes they follow may produce the same or similar results as the existing metro lines. In this research, an answer was sought to the question of whether a metro line designed with an approach based on swarm intelligence could be compatible with the existing line.*

The paper aims to develop a model that guides the design processes of metro lines by using swarm intelligence and shortest path finding strategies of slime moulds in a computational model. In the model, slime moulds were represented by multi-agent systems. The metro route was produced by the model using the station locations on an existing metro line selected as the study area. The agent-based simulation model was developed in the Grasshopper Physarealm plug-in environment. In the first stage of the model, a numerical model was created using the data and parameters of an existing metro line. In the second stage, the simulation was carried out to create a new route by preserving only the station locations of the existing metro line. Finally, the existing metro route and the route developed with the simulation model based on the intelligent agent behaviour in the digital environment were compared by overlapping. The results show that the route created by slime moulds only to reach the food is almost the same as the currently designed metro line. Topography data is ignored in the model. It is seen that the model developed by using metro design criteria is an important decision support aid for designers in determining metro routes.

KEYWORDS*: Swarm Intelligence, Slime Mould Behaviour, Agent-based Modelling, Metro Route Design.*

REFERENCE: *Sena Kaynarkaya, Gülen Çağdaş (2022). Evaluation of Metro Lines with Swarm Intelligence Approach. Journal of Information Technology in Construction (ITcon), Vol. 27, pg. 802-826, DOI: 10.36680/j.itcon.2022.039*

1 INTRODUCTION

The field of architecture has always been inspired by biology; It is possible to see that information and communication technologies are inspired by the life cycles and behaviours of living organisms in the approaches and models used in architectural and urban design fields. When compared on both formal and relational basis, it is seen that architecture and biology are closely related to morphology. In addition, both disciplines exist from individual elements working as a whole and there is a continuous interaction between these elements. Besides all these similar features, biological processes are more mobile and productive. Based on the statement "Sometimes you have to move to a completely different area to solve a problem in a particular discipline," which Gödel (1931) used in his Incompleteness Theory; Cooperation between disciplines is often inevitable.

The representation of the behaviours of the creatures in nature, especially birds, fish and ants, which exhibit swarm behaviour, has been an important source of inspiration in solving many urban and architectural design problems with computer models (Andrasek, 2015).

This article aims developing a computational model based on swarm intelligence for designing metro lines and comparing the solution with an existing metro line. The most important factor in design process of metro routes is the determination of the shortest route. The food finding strategies of some living creatures in nature were used to determine highway and subway routes in literature. However, most of these studies are experimental studies. In this article, swarm behaviour of slime moulds is used as an approach for the computational model. In the model slime moulds were represented by multi-agent systems.

In this study first of all, studies simulating swarm behaviours in physical environments, and agent-based systems were reviewed from the literature. Then, a computational model was developed by analysing the life cycle and emerging swarm behaviour of slime moulds. Rhinoceros / Grasshopper and Python were used in the development of the model. The model was run using the data of an existing metro line, and the results were compared and evaluated.

1.1 Literature Review

1.1.1 Swarm Intelligence

Swarm intelligence is a behaviour that occurs as a result of the joint actions of many factors. Swarm intelligence was first used in robotic systems created in 1898 by Gerardo Beni, Susan Hackwood, and Jing Wang to describe an emerging phenomenon (Beni, Wang, 1990).

Swarm intelligence consists of factors that are aware of their environment, do not isolate themselves, are autonomous or semi-autonomous, and that can produce global behaviour (Hight, 2006; Wiesenhuetter et al., 2016). With this behaviour, it is possible to solve the problem defined by the interaction of factors, without requiring a central authority. Swarm intelligence is related to the concepts of self-organization. Stigmergy is the whole of the interactions that enable the individuals in the swarm to communicate while forming the infrastructure of the life cycles of ant colonies and slime moulds (Holland, 2006) (Figure 1). Distribution of information is essential in the concept of stigmergy. Thanks to the trace left by the influence of the changing environmental conditions, the whole system becomes aware of this effect and reorganizes itself. The pheromone released by ants is one of the examples of stigmergy. Self-organization is the process of adapting to the environment for the development of a certain function of the internal order of a system without any external control and direction (Hensel, et al., 2004). Development in self-organized systems happens through randomness, feedback, and interactions (Holland and Melhuish, 1999). Stigmergy, which has indirect interaction, is a concept that feeds self-organization. There is no need for central authority, with the swarm working by organizing itself. Organized interactions of local factors have global consequences while at the same time keeping the system balanced. Each of the individuals that make up the swarm is identical (Beni and Wang, 1990; Heylighen, 1989). Having limited autonomous behaviours and being able to serve the target as a group reveals global behaviour (Fig.1).

With the increase in swarm intelligence and the use of algorithms based on this intelligence in urbanism, planning based on the will of the designer has been replaced by the interaction of individuals. In this context, the aim of the Swarm Urbanism project, designed by Roland Snooks and Robert Stuart-Smith in Melbourne, is to produce algorithms that will help design transportation and infrastructure networks, which are indispensable elements of master plans. Rather than mapping the design with the study, it is desired to transform the collective design obtained by factors into a system (Fig. 2).

Fig.1: Self-Organization: Swarm of birds (Url-1).

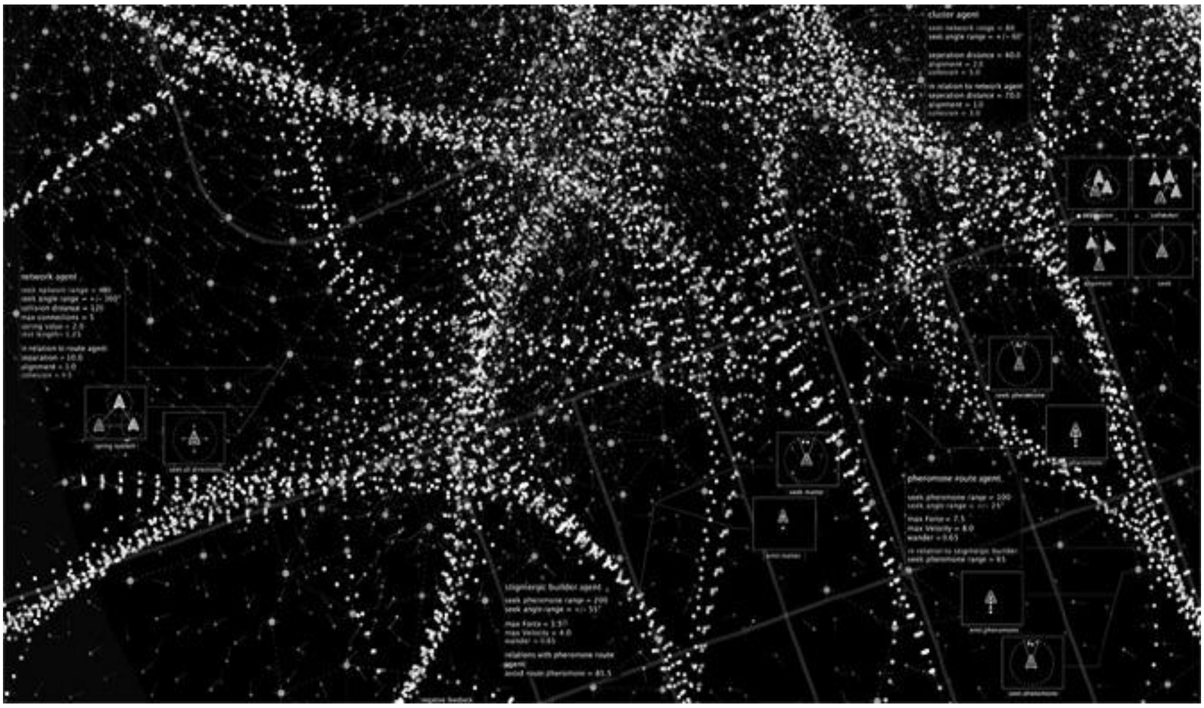


Fig. 2: Swarm Urbanism (Snooks, 2014).

The generic agent type was not used to organize the system. Instead, ecology has been created in which global characteristics can be obtained with factors. In addition to the characteristics of the factors, the hierarchical structure provided by their densities also played an active role in the design process. The high number of factors, which is one of the most critical points of swarm intelligence, makes it easy to explain the effects of many elements of urban systems on the shaping of the city and building. One of the projects that aims to obtain algorithms that form the basis for 2D and 3D projects by combining the operating principles of swarm intelligence with computational methodologies is the 'Swarm Intelligence project' belonging to Tyler Julian Johnson (Fig 3).

Fig 3: Swarm Intelligence (Url-2)

1.1.2 The Behaviour of Slime Mould and Related Studies

In nature, many living creatures seek food to continue their life activities. While doing this, one of the living species that interact with their environment by developing growth strategies that adapt to their environment, creating more global systems with simple life cycles, is "Slime Mould", a species of "Physarum polycephalum". Moulds have a network line that provides direct connections and additional connections between nutrients to facilitate the transmission between dispersed food sources. The web structure developed for nutrient transfer in slime moulds consists of the most efficient tubes in terms of the amount of nutrients transported and transport performance (Schelling, 1978). Besides efficiency, the most important feature of these connections is that they establish the shortest distance between food sources (Sanders, 2010).

Slime moulds, which are living organisms that do not have any nervous system. Although they do not have a central direction and brain function, they have been the subject of many studies with their potential to learn and transfer the information they have learned to other cells as soon as they are conjugated. After the success of slime mould cells in the maze solving experiment conducted by Nagakaki and Toth (2000), a group of researchers started a study on the Tokyo subway (American Association, 2010). These experimental studies have been carried out by wondering whether slime mould colonies can produce these network maps, which engineers have designed with long shifts. Slime mould spread was observed on the map of Tokyo and surrounding cities (Fig. 4).

It has been observed that the web produced by slime moulds is formed quite like the Tokyo subway. The similarity between the subway network and the slime mould network has made us think that the slime mould behaviour contains a mathematical model. Continuing the investigations on the Tokyo subway line, it was observed that the mould cells sent their networks everywhere in the first place. Over time, the other tubes were withdrawn, preserving the strongest and shortest of the networks between the food sources (Fig. 5).

As a result of the studies, the fact that these simple organisms formed a line produced with long calculations in a very short time provides an infrastructure that can quickly develop an alternative in the search for different networks in the future in situations such as natural disasters or war (Adamatzky, Prokopenkob, 2011). In addition, the system that develops with a decentralized will be able to provide an algorithmic base for technologies such as artificial intelligence.

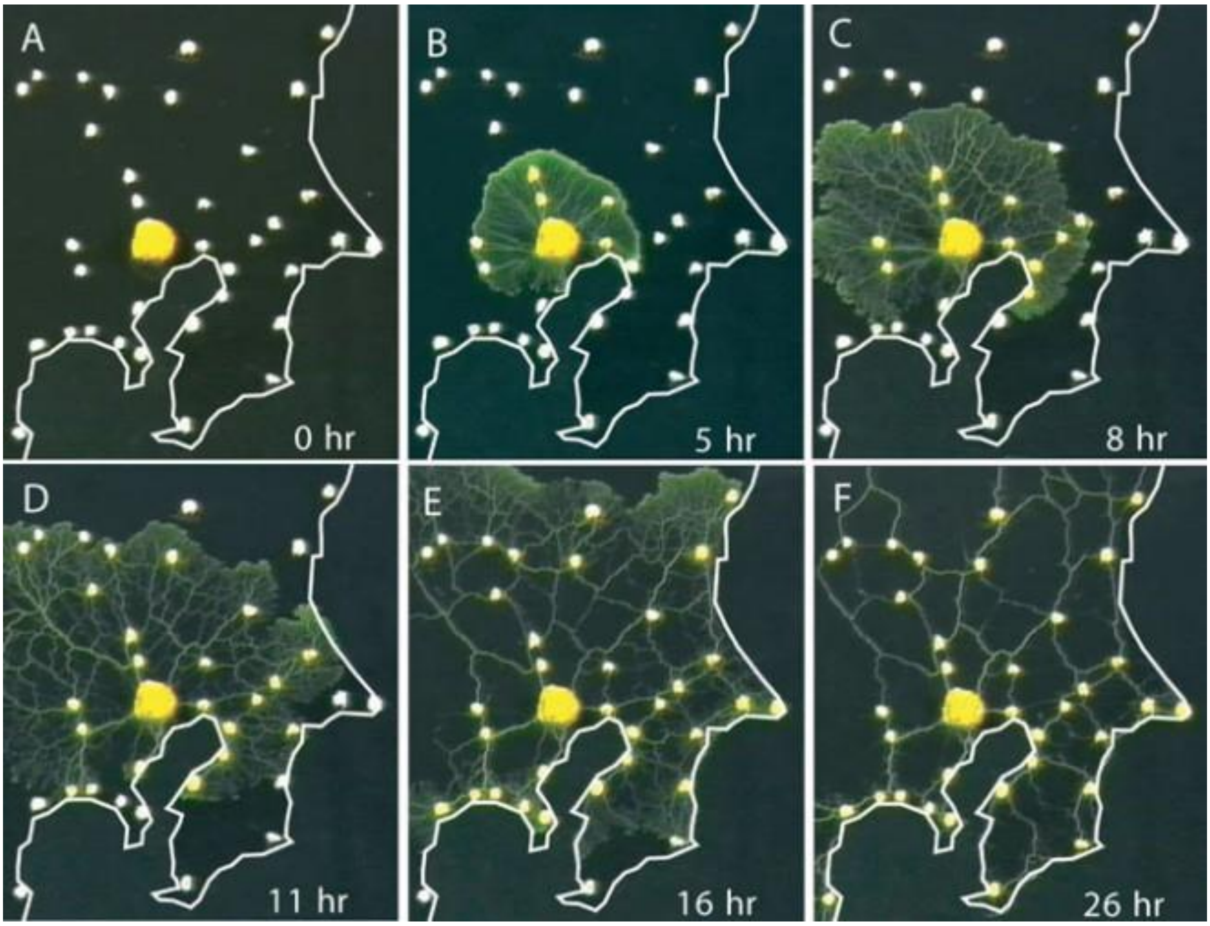


Fig. 4: Time-dependent slime mould growth on the Tokyo subway (Url-3).

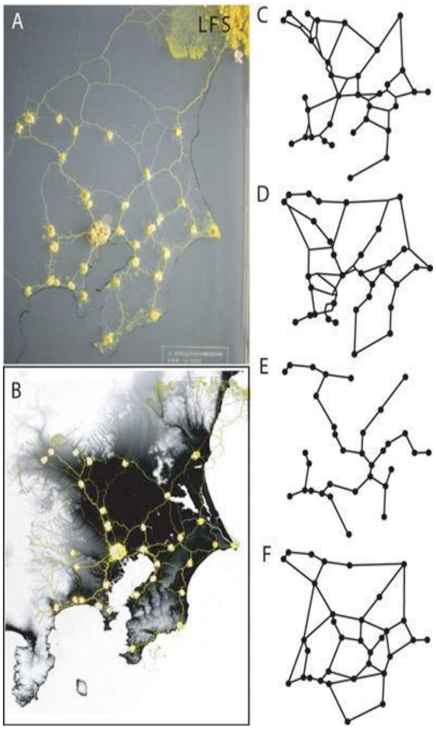


Fig. 5: Time-dependent change of plasmodium tubes (Url-4).

Adamatzky has conducted many studies evaluating the formation of highway networks of various countries around the world through slime moulds. Densely populated cities with high economic potential and critical transportation points were selected in the countries identified in the studies, and the slime mould and the highway networks of these countries were compared. Before starting the experiment to compare highway networks, a suitable environment was prepared for the development of slime moulds. Paper towels sprinkled with drinking water were placed in square polystyrene petri dishes with a size of 120x120 mm or a diameter of 90 mm. Slime moulds were placed on the oat flakes for growing. The map of the area to be tested was cut into agar plates by reducing it to a certain scale (Fig. 6). Oat flakes were placed in the cities where the line passes. Except when the experiment is visualized, the experiment environment is dark, and its temperature is 22-25 ° C. The experiment was repeated an average of 20-25 times according to the results. Test containers were scanned in a scanner and visualized by increasing the saturation level (Adamatzky and Akl, 2011).

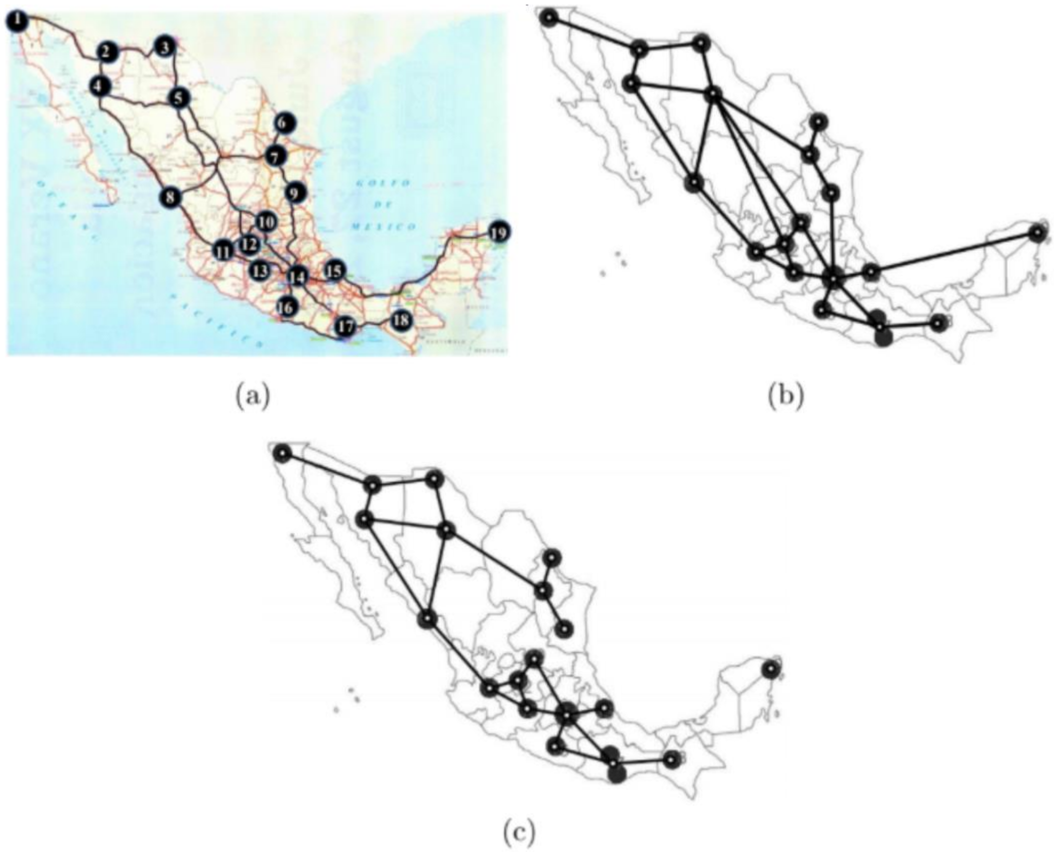


Fig. 6: Mexico's Network map (Adamatzky et al, 2010).

The experiment was started by inoculating slime mould on oats in the selected starting city on the cut agar plates. In addition to oats, which trigger the agglomeration action, crystal salt, which has the effect of repelling slime moulds, is used to prevent fungi from passing through the obstructed areas. The experiment was started by accepting the current state of the highway line as the H graph and the following assumptions were made while defining the slime mould graph:

U: Urban area set.

1. E: Set edge.
2. P: Slime mould graph
3. W: E [0,1]: Possible weight of the sides. Each side of E is associated with a probability.
4. P(Q): Threshold value. It is the thickness of the tubes formed between the points in the U region. It is proportional to tube thickness and weight. When the Q value changes, a change is observed in the way the points are connected.

1. Graph H is in raw state when ○ P (Q) = 0.

T (E) = When e and E: w (e)> Q, it removes all edges that have weight equal to or less than Q.

The slime mould chart is as follows:

P(Q) = (U, T(E), w)

The results obtained from the slime mould graph are compared with the Planar Proximity Graphs (Gabriel Graph, Relative Neighbourhood Graph) used in highway constructions (Adamatzky and Prokopenkob, 2011). These studies have been repeated for the highways of countries like England (Adamatzkya and Jones, 2009), (Fig 7), Mexico (Adamatzky, 2010), Canada (Adamatzky and Akl, 2011), Australia (Adamatzky and Prokopenkob, 2011), Netherlands (Adamatzky, et al., 2012), China (Adamatzky, et al., 2013), and it has been stated that there are large similarities.

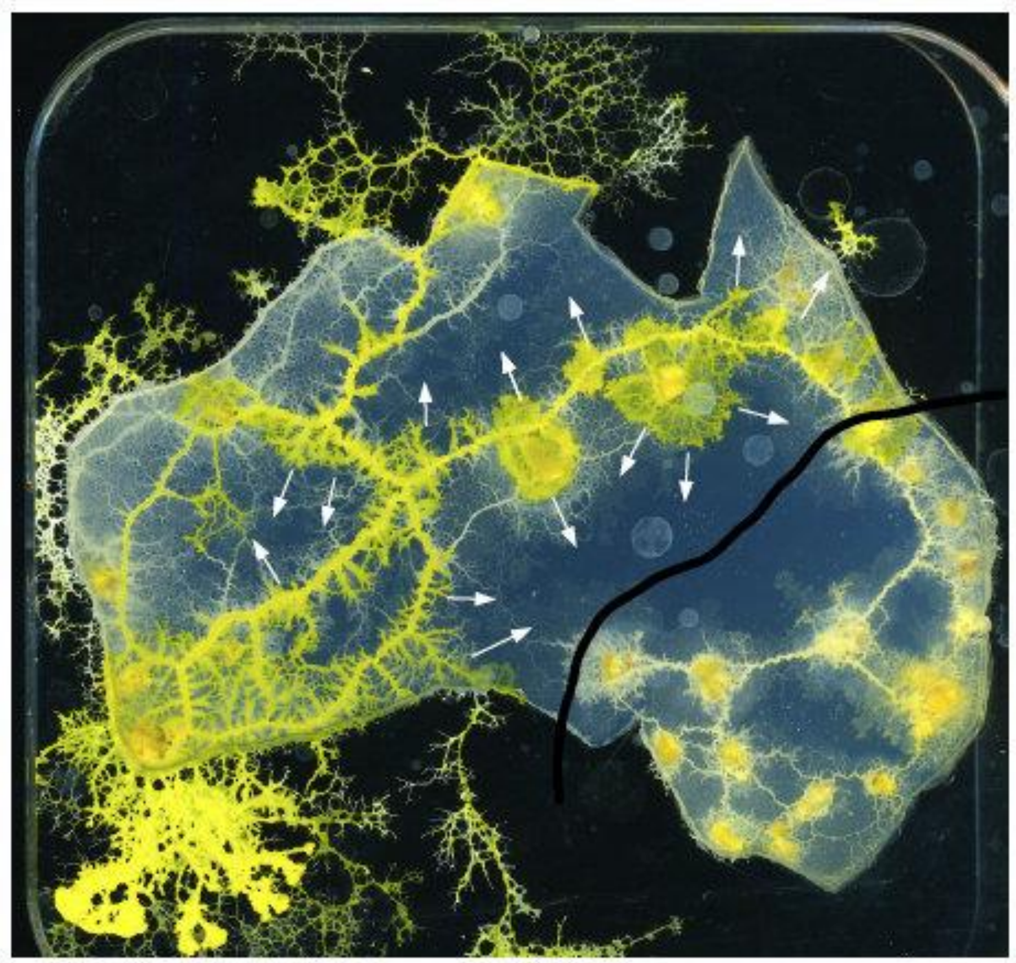


Fig. 7: Scanned image of experimental Petri dishes (Adamatzky et al, 2010).

1.1.3 Agent-based Systems

Agents are used to simulate intelligent behavior in an environment containing objects. While the agents represent each individual whose movement will be simulated; Objects represent passive components in the environment. The environment, on the other hand, is the topological space in which the model is created, the agents move, and the objects are in. The agents transform the situation they perceive from the environment into action according to the situation-action rules. The conditional action rule system is expressed by “if-then” rules (Russell, Norvig, 1995).

An agent-based system is a unified network of agents that act flexibly and autonomously in a given environment to achieve their goals. The agent properties can be summarized as autonomy, social ability, responsiveness and having memory. Autonomy can be defined as decision-making without any direct influence. Social ability of the agents is the ability to belong to a community and communicate with other agents (Padgham, Winikoff, 2005). This property provide to create collective behaviours while allowing complex and nonlinear interactions. The term behaviour is defined as the complex actions of people or animals; the improvisational moves of autonomous agents are also determined as behaviour.

Agent-based systems constitute the base of the simulation models. Agents move within a certain environment to achieve the targets defined to it. The outputs of the system, agent networks are the traces left by agents during the movement. Data acquired from the coordinate points which the agents pass during the movement are updated and saved by the time. The line between those points is the representation of agent network. These traces are dynamic and continue to take shape throughout the movement of the agent.

Multi-agent systems have many advantages, such as involving a large number of simulation techniques based on mathematical and heuristic models. First, multi-agent simulations provide a visual interface where the user (the designer of the model) can observe the movements of the agents. As a result of the interaction of the factors, the user of the model can watch how movements appear on a large scale.

2 A MODEL PROPOSAL BASED ON SWARM INTELLIGENCE FOR DESIGNING METRO LINES

This paper aims to develop a model that guides the design processes of metro lines by using swarm intelligence and shortest path finding strategies of slime moulds in a computational model. The most critical point of metro designs, consisting of detailed engineering calculations, is that the routes between the departure and arrival points are as short as possible. Therefore, the model proposed in the study is based on the behaviour of slime moulds to reach food sources.

Since the model's development process is entirely related to the feedback-based behaviours of the agents, the working process is represented by the diagram given in Figure 8. Like the physical experimental environments, the slime mould population represented by multi-agents systems in the model was ensured to reach a sufficient number. Appropriate parameters were obtained for directing them to station points representing the food source. Agents were allowed to spread out and create a route between stations to reach the food sources located at the stations.

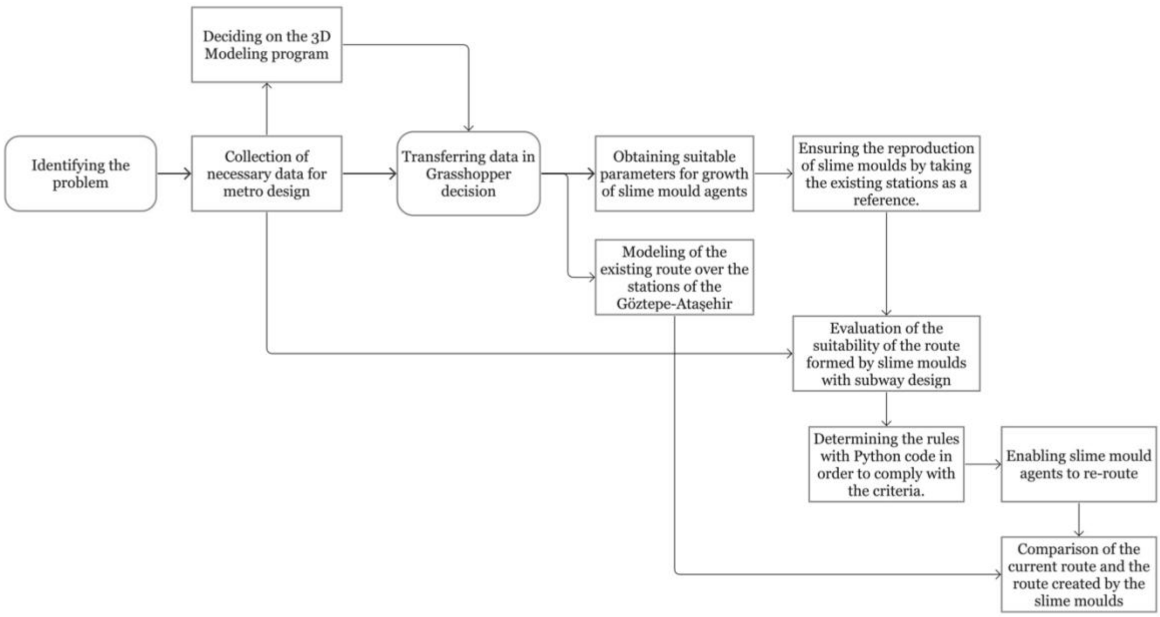
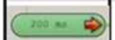
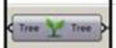
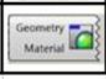


Fig. 8: Study Flowchart.

The agent-based simulation model was developed in the Grasshopper/Physarealm plug-in environment. In the first stage of the model, a numerical model was created using the data and parameters of an existing metro line. In the second stage, the simulation was carried out to create a new route by preserving only the station locations of the metro line. Finally, the existing metro route and the route developed with the simulation model based on the intelligent agent behaviour in the digital environment were compared by overlapping.

The fact that the factors show different breeding patterns and birth rates according to the distance between them, the angles between their locations and the intensity of the trace amount requires feedback in the program to be used. For this reason, Rhinoceros-Grasshopper, which offers simultaneous printing and parametric modelling, was preferred as an environment for developing the model of the study. Grasshopper is a modelling tool integrated into Rhinoceros. Grasshopper, developed by David Rutten (2007), provides the opportunity to constantly update parameters and instantly view changes that occur as a result of updating. Thanks to the simultaneous feedback provided by the program, the changes made to the geometry are permanently updated thanks to the sliders. The algorithm sets are connected to each other step by step, enabling the model to work. Modelling can be done in Rhinoceros or directly on Grasshopper (Table 1).

Table 1: Grasshopper components.

	COMPONENT	IMAGE	NAME	ID	DESCRIPTION	LOCATION
GRASSHOPPER	PARAMS		POINT	Pt	This component represents point or points.	Geometry
			SURFACE	Srf	This component represents surfaces or surfaces.	
			PANEL	-	It evaluates the components and lists this evaluation in order of binding.	Input
			NUMBER SILIDER	-	It is the slider that changes the number value.	Primitive
			BOOLEAN TOGGLE	Toggle	It allows you to quickly switch between a single true and false value.	
			COLOUR SWATCH	Swatch	Color chart.	
			TIMER	-	You can change the parameters at some intervals and see the result like an animated movie.	Util
	MATHS		LARGER THAN	Larger	The Greater than component takes two lists of data and determines whether the first item of List A is greater than the first item of List B. The two outputs let you decide whether you want to evaluate the two lists based on a larger value.	Operators
			SMALLER THAN	Smaller	The Less than component performs the reverse action of the Greater than component. The less-than component determines whether list A is less than list B and returns a list of boolean values.	
	SETS		CULL PATTERN	Cull	Faces to be taken from a cage.	Sequence
			GRAFT TREE	Graft	Creates a data tree by adding an extra branch to each item.	Tree
	VECTOR		CLOSEST POINTS	CPs	Finds the closest points in a point collection.	Point
	SURFACE		SPHERE	Sph	Create a spherical surface.	Primitive
			BREP CLOSEST POINT	BrepCP	Finds the closest point on a brep.	Analysis
	DISPLAY		CUSTOM PREVIEW	Preview	It provides the created shape to be colored.	Preview

The Physarealm plug-in, which is embedded in the Grasshopper and produced for Slime moulds behaviours was used for developing of the model. Physarealm is a plugin produced with agent-based algorithms. It is software used to simulate the experiments in the laboratory environment using the rules explaining the life cycle of slime moulds in the digital environment. It consists of six component groups: Analysis, Core, Emitter, Environment, Food, Settings (Fig.9).

In Physarealm, "Surface Environment" was selected as the breeding environment in terms of similarity to the materials used in physical experimental environments and suitability for the representation of the metro line, and "Point Food and Point Emitter" was chosen to show the vaccination and food sources. The remaining components were determined in accordance with breeding rules, media sizes and distance between stations. Grasshopper commands were used to arrange the surface where the model will work, the points where the factors will move, and the relationships (Table 1).

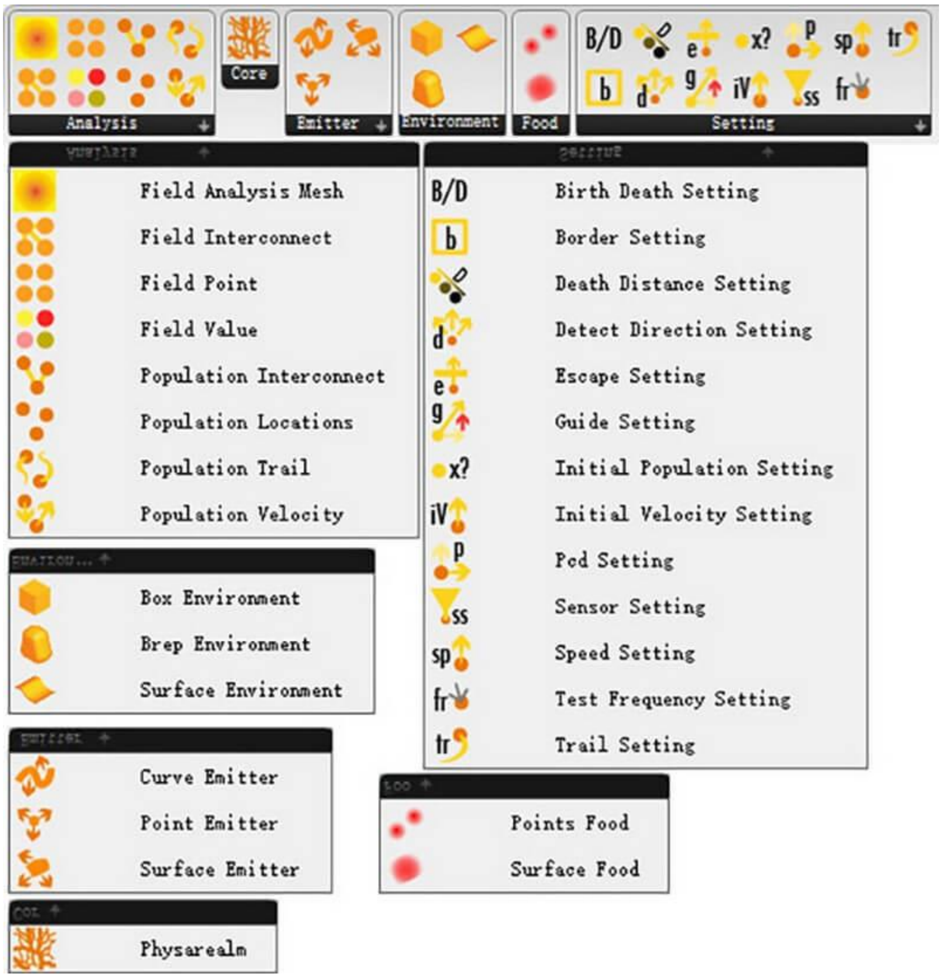


Fig. 9: Physarealm Components (Url-6).

Environment component, which is one of the components that make up the Physarealm plug-in, defines the environment where the factors will grow, while the Emitter component represents the first point where the colony is released to the environment and the starting station of the metro line. While the Emitter component, which can be represented by a box, B-rep or point, controls other food sources to which the mushrooms will be directed, the Food component, which represents the other station points of the metro line, connects to the Core node and enables the system to work.

Settings component contains subcomponents to which we can add parameters controlling mould behaviour. Optimum initial values were assigned to the parameters in the setting component, taking as reference the studies of Jeff Jones (2010), who formulated slime mould behaviour. Several behavioural models form the basis of the Physarealm behavioural models (Fig. 10).

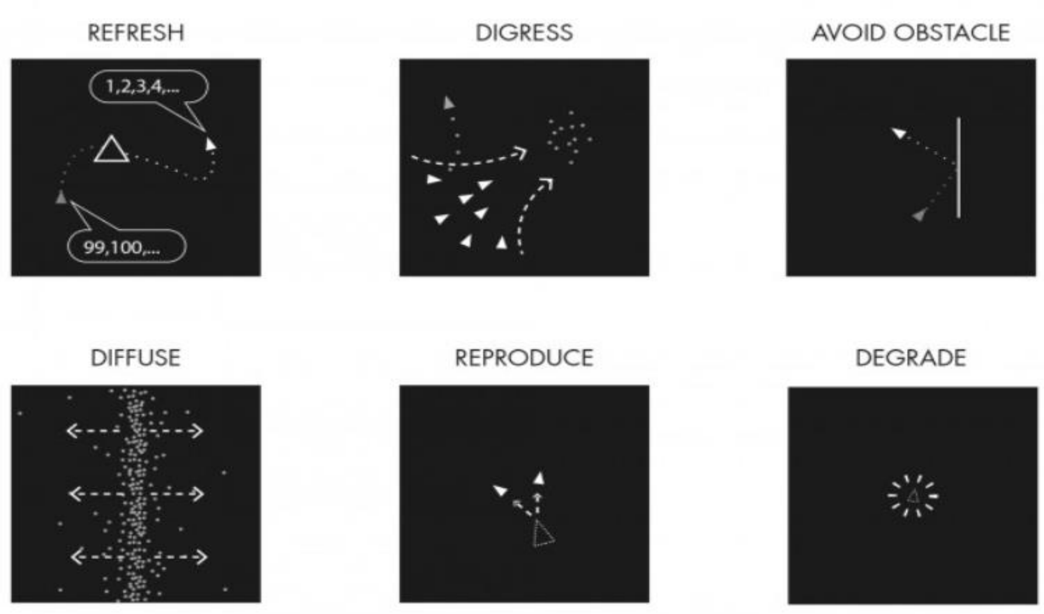


Fig. 10: Physarealm behavioural models (Url-7).

Although it is inspired by Jeff Jones' (2010) model, the two models are quite different from each other as a result. The breeding environment of the factors that make up the swarm is a cube divided into cages (Fig. 11). The behavioural patterns of the agents are shaped within this cube. Their positions relative to each other and the angles between these positions form the basis of the parameters in the Physarealm extension. The agents have sensors that enable them to detect the chemo-attractant substance secreted by each other. The behaviour component is composed of four parameters: a coordinate plane in which the agent moves in the Z axis, Sensing Balance (SB) at radial distance to each sensor, Detection Angle (DA) at the maximum pole angle, and Direction Sensing (NPhy), which consists of the number of sensors that enable to detect factors located in the same Z coordinate.

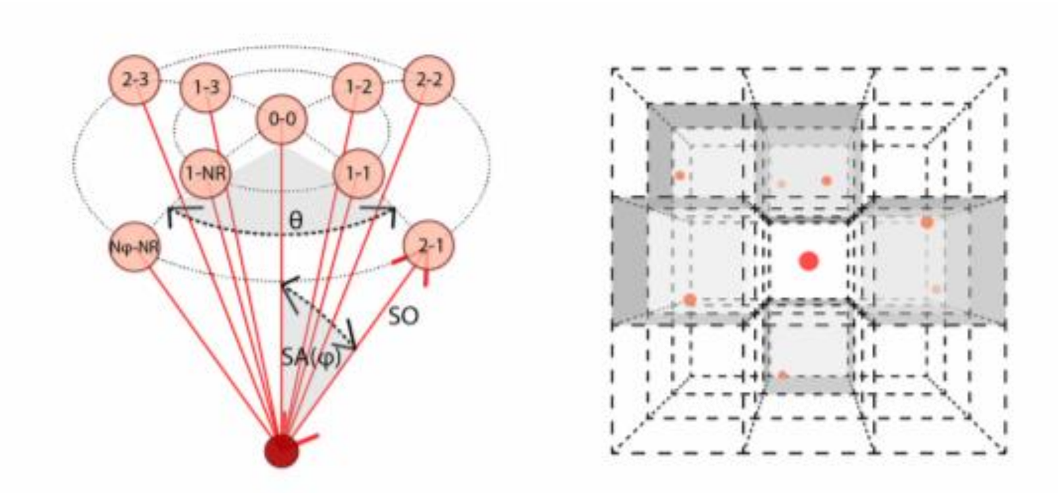


Fig. 11: The environment and behaviour angles (Url-8).

When the number of factors in birth and death behaviour (Adaptation) becomes too high, some of the factors in the environment die. The algorithm updates the behaviour based on the new number of agents. The rules work according to factors within a close radius (Table 2).

The radius for birth is DvR and for death it is DeR

The radius of the area is (2 * DvR + 1) and (2 * DeR + 1.

Therefore, the control area is 3 * 3 * 3 = 27 if radius = 1.

The number of neighbouring factors is N.

If DvMin> 0 and DvMax < $(2 * DvR + 1)^3$ then growth will not occur.

If DeMin> 0 and DeMax < $(2 * DeR + 1)^3$ the number of factors decreases.

For example; The algorithm counts 26 neighbouring cells and checks if any other factors are occupying them when DeR = 1. If the number of neighbouring cells is between DvMin and DvMax, the agent replicates itself. The cages forming the cube in the chemoattractant trace behaviour are (3 * 3 * 3) in size. The TrRat parameter is the amount of trace left by the agent passing through the food. The behaviour that emerges according to the amount of trace and the resulting algorithm also changes.

Apart from the main behaviour parameters, there are a few other parameters that change swarm behaviour:

Vertical Guide Factor (VGF): It is the parameter that affects the movement of the agent. With a high VGF, agents tend to move more vertically; this is useful for finding forms such as columns.

Initial Velocity (IV): This parameter is a factor that has a greater effect in multi-agent systems such as swarming. Although slime moulds are not flocked, this behaviour is important for initiating directional factors.

Probability of Escape (EscP): A large parameter value causes factors to escape from the specified limits or enter the determined obstacles.

Table 2: B / D Setting Parameter Definitions.

Speed Setting
Max Speed = Fastest speed at which agents will move.
Death Distance Setting
Death Distance = Death distance in the positive value
Sensor Setting
Sensor Angle (SA) = It is the maximum pole angle perceived by agents.
Rotate Angle (RA) = The agent goes to the direction of the angle of rotation, not where there is the highest amount of chemoattractor.
Sensor Offset (SO) = It is the highest value of the range detected by the sensor.
Detect Direction Setting
DDirR = It is the point amount at which the lower circle of the perceived conical structure will be divided.
DDirP = It is the angle at which the lower circle of the perceived conical structure will be divided.

For the study to be conducted within the scope of the article, Göztepe-Ümraniye metro line was determined as a pilot line due to the up-to-dateness of the data that can be obtained. From the Istanbul metro design criteria and standards booklet, data on the issues to be considered while designing the metro line has been obtained. In line with the Istanbul Metropolitan Area Urban Transport Master Plan; the line with its first stop, last stop and number of stops is shaped according to these criteria. The general rules applied for each line such as the maximum length of the distance between two stops, the highest inclination value that the vehicle can climb, and the radius of the bend it can turn, and the state of the environment where the line passes are the factors to be considered when determining the station points.

With the comparisons made between two preliminary projects and final projects obtained from Metro A.Ş Survey and Projects Directorate, the criteria taken into consideration when determining the station points were prioritized. The criteria determined when designing the Göztepe-Ümraniye line are: Integration, Density (School, shopping mall, hospital), and Zoning status (Table 3).

Table 3: Göztepe-Ümraniye metro line design criteria.

	Stations	Criteria
1	Göztepe 60. Yıl Parkı	Caddebostan beach vehicle road integration
2	Göztepe	Marmaray integration
3	Sahrayıcedid	Minibüs street vehicle road integration
4	Yeni Sahra	Kadıköy-Tavşantepe Subway integration
5	Ataşehir	Maximum distance between two stations and zoning status
6	Finans Merkezi	Attraction point
7	Soyak Yenişehir	Maximum distance between two stations and zoning status
8	Atakent	Maximum distance between two stations and zoning status
9	Çarşı	Üsküdar-Ümraniye Subway integration
10	Hastane	Ümraniye training and research hospital
11	Kazım Karabekir	Last station

In this study, the main basis of the behaviours of slime moulds, which are determined as reference creatures, are neighbourhood relations and interactions arising from this relationship. The reproduction status of the factors representing slime moulds in the environment, their deaths due to the highest number they can reach and their distance from each other is directly related to the neighbourhood principle observed in emerging phenomena. The breeding environment was chosen as the surface since the model developed within the scope of the paper was built on metro routes. The study was carried out ignoring the slope the medium size determined as pixels in Jeff Jones (2010) study and other applications is given in centimetres in this study. Since the parameter and equation calculations are based on pixels, some parameters did not fit exactly in this study. The environment in which the model was developed was generated by reducing the dimensions of 90x90cm in 1 / 10.000 scale. Metro station locations have been placed in accordance with the original in line with the data received from Metro AŞ (Fig. 12). Station and route representations are as follows:

Current Metro Route

Current Station Points

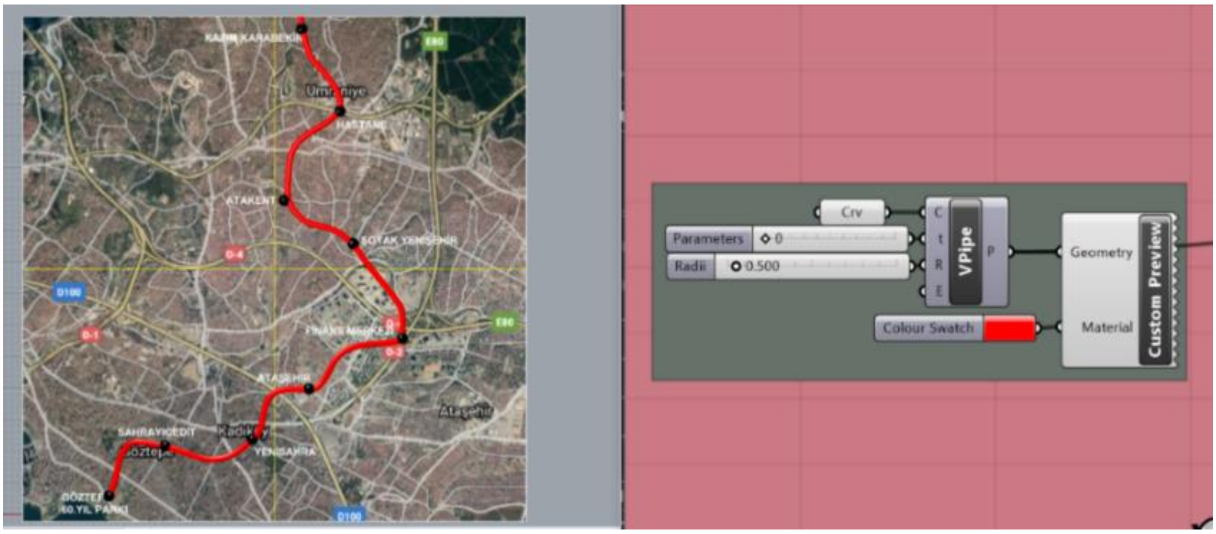


Fig. 12: Representation of the current line in Grasshopper.

AutoCad drawings of the project obtained from Metro AŞ were transferred to the Rhinoceros environment, making the station points and line visually more readable. The curve forming the line with the "Pipe" component in the Grasshopper is represented by a tube over the AutoCad drawing. The similarity of the line formed by the swarm with the current line and the differences between it with the existing line were observed through the digitization. First, the existing line and the model created in Grasshopper were compared without any intervention other than the starting parameters of the swarm. Then, the development of the line was observed depending on the orientation of the swarm with the obstacles placed at certain points. There are five main setting components used for the "Surface Environment" component in Physarealm plugin. Each setting component contains many parameters that affect the reproduction. These parameters were created for the chemoattractant trace to be perceived by other factors and to change their behaviour. (Table 3, Table 4). The number of factors determined at the beginning of the simulation process is 70. Since the swarm population behaviour affects, behaviours were observed over the values obtained by assigning different values to the parameters.

Table 4: Parametric definitions of Speed Setting, DDis. Setting, Sensor Setting and Detect D. Setting.

B/D Setting
DvR = Birth distance radius
DvMin = The shortest distance between the factors in the DvR
DvMax = Long distance of factors in the DvR with each other
DeR = Death distance radius
DeMin = The shortest distance of factors in DeR to each other
DeMax = Longest distance of factors in DeR to each other

The model began to be developed based on the initial data provided by the plugin developers. The size of the medium dramatically changes the number of factors and parameters used. In B/D Setting, components such as Dvmin and DvMax are called "parameters," and the effect numbers of these components are called "factors." Reproduction is carried out by configuring the factor changes in the parameters according to the size of the environment. The most suitable speed at which the simulation can be followed quickly was selected. The Sensor Adjustment values are close to the values recommended by the developers (Table 5). Since the number of agents is enormous, each sphere form in the simulation represents a set of agents.

Table 5: Factors of simulation-1.

B/D Setting	Speed Setting	Death Distance Setting	Sensor Setting
DvR=1 DvMin=2 DvMax=25 DeR=3 DeMin=1 DeMax=10	Max Speed=3	Death Distance=9	SA=22.5 RA=45 SO=10

When the values are defined, and the simulation is started, the factors began to increase from the starting station to the other stations (Fig. 13). Although they tended to reproduce a little in the first working phase of the model, their number decreased, and all factors died before reaching the fourth station (Fig. 14). The high DvMax parameter caused the factors to reproduce at long distances, uncontrolled swarm growth, and die suddenly.

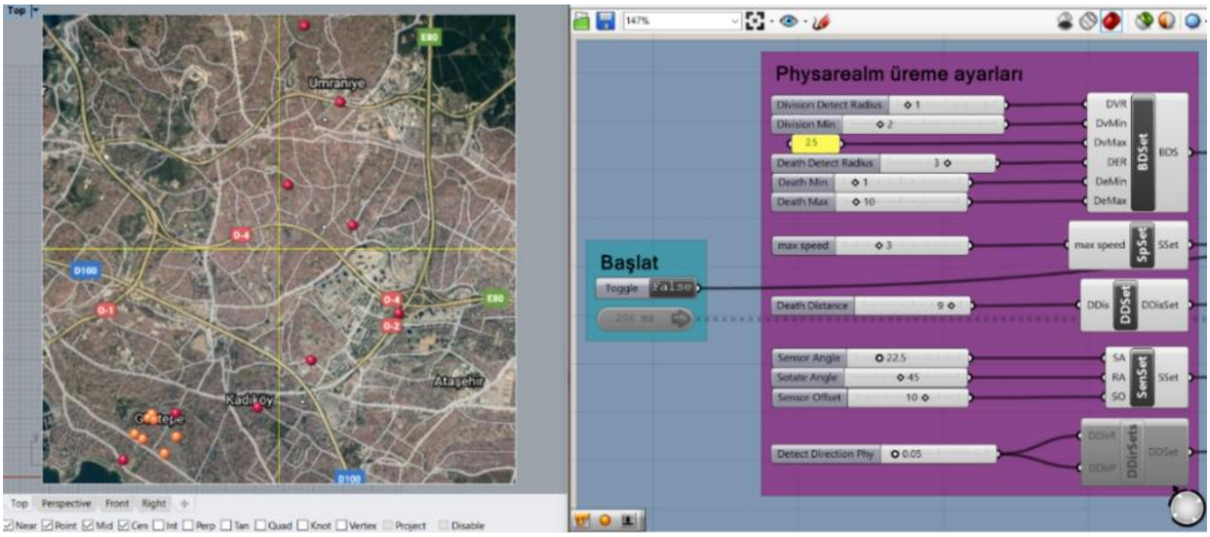


Fig. 13: Simulation 1-a.

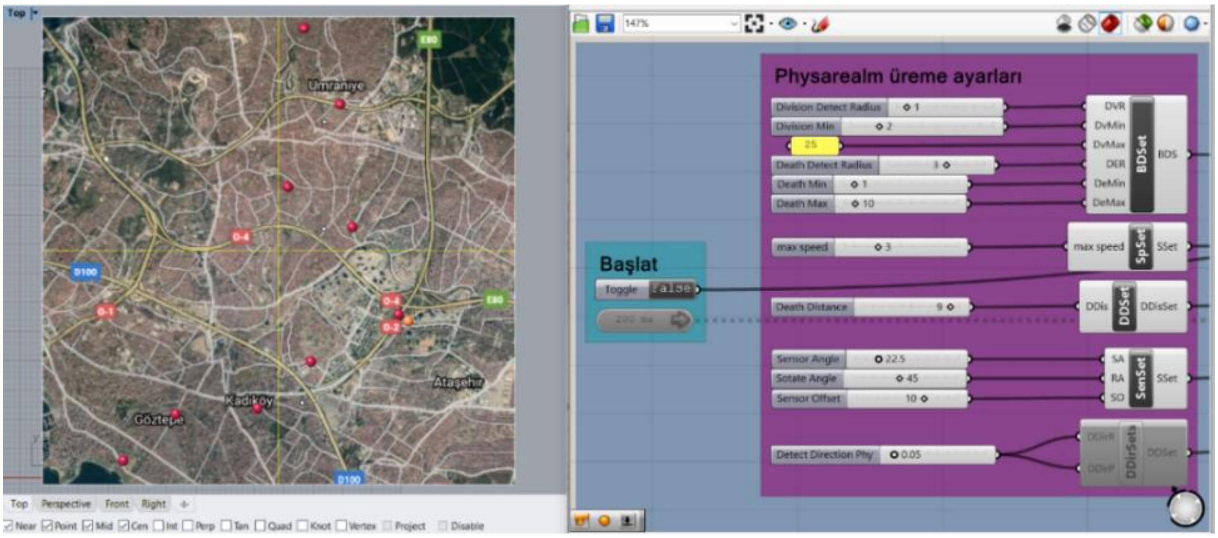


Fig. 14: Simulation Process 1-b.

During the operation of the model, some of the parameters were kept constant and the simulation results were followed by changing the values of some; seven simulations were carried out in this process. The value of DvMin, DeMin and SA parameters, which are the main parameters of the model, is very important for the factors to follow the chemo attractor trace. Therefore, the DvMin, DeMin and SA parameters were changed in the seventh simulation and the behaviour of the agents was repeated (Table 6). With the last parameters defined, it was observed that the factors continued to reproduce regularly and reached the last station (Fig. 15).

Table 6: Factors of simulation 7.

B/D Setting	Speed Setting	Death Distance Setting	Sensor Setting	Detect Direction Setting
DvR=1 DvMin=0 DvMax=1 DeR=3 DeMin=1 DeMax=10	Max Speed=3	Death Distance=9	SA=22.5 RA=45 SO=10	DDirR=0.05 DDirP=0.05

It is important that there is not too much difference between the B / D Setting parameters to avoid the dispersion and therefore the destruction of the swarm. Because when the difference between the minimum and maximum value of the birth radius increased, the factors moved away from each other and caused the traces to be not perceived. However, the high amount of chemo attractor alone is not the most important parameter for the movement of the factors. Therefore, the SA parameter directs the growth of the agents by working together with the trace amount.

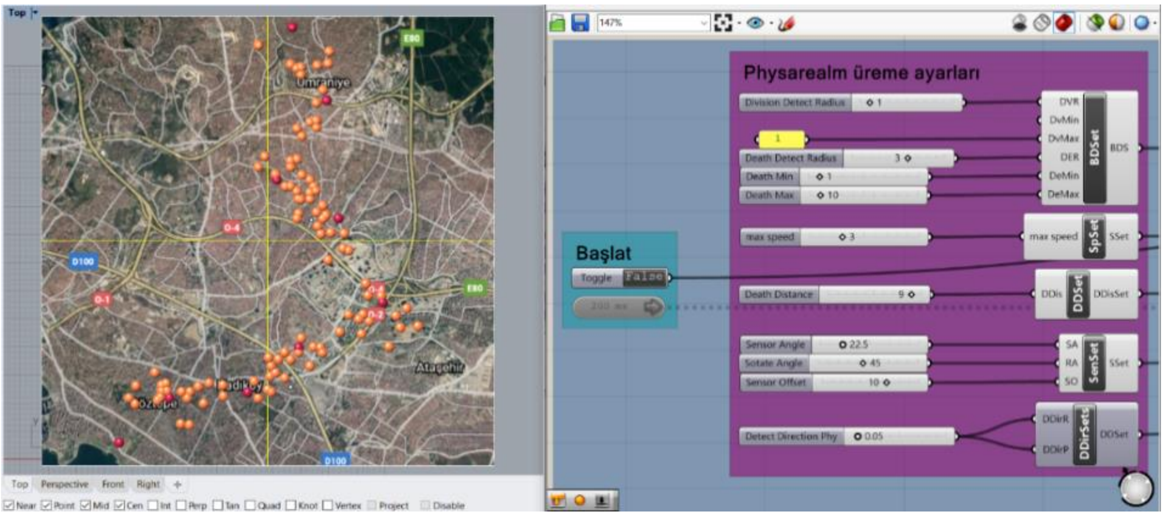


Fig. 15: Simulation 7.

After the simulation started, some deficiencies were found in the operation of the model. At the same time, dispersions started to occur as all factors stopped by the stations in turn (Fig. 16)

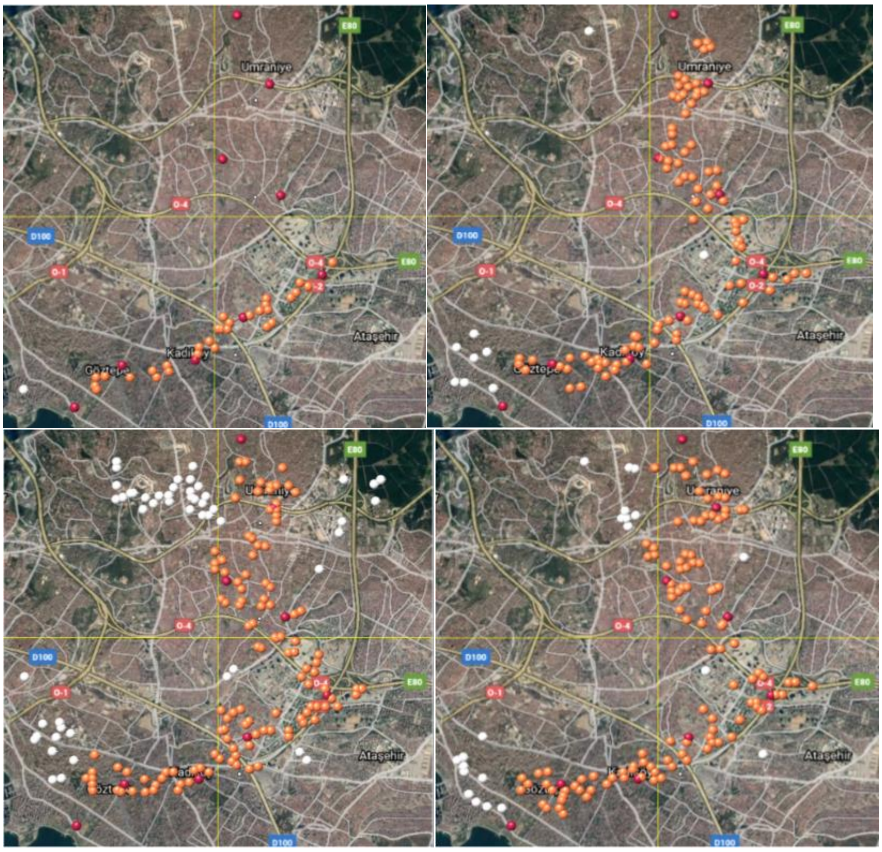


Fig. 16: Spreading process of slime mould agents.

While increasing the number of factors, attention was paid to the distance at which the spreading tendency of the agents occurred. First of all, the density of the factors forming the swarm was reduced and groupings were made with the help of the operators in the Grasshopper. The Closest Points parameter creates a group within a point cloud with elements close to a certain point. The number of elements in the group is optional. The swarm was simplified with the Closest Points component, and then the remaining factors were regrouped according to the distance between them and the Smaller Than component. When factors with less than 4.00 units are left in the simulation, it has been observed that there are no dispersions (Fig. 17).

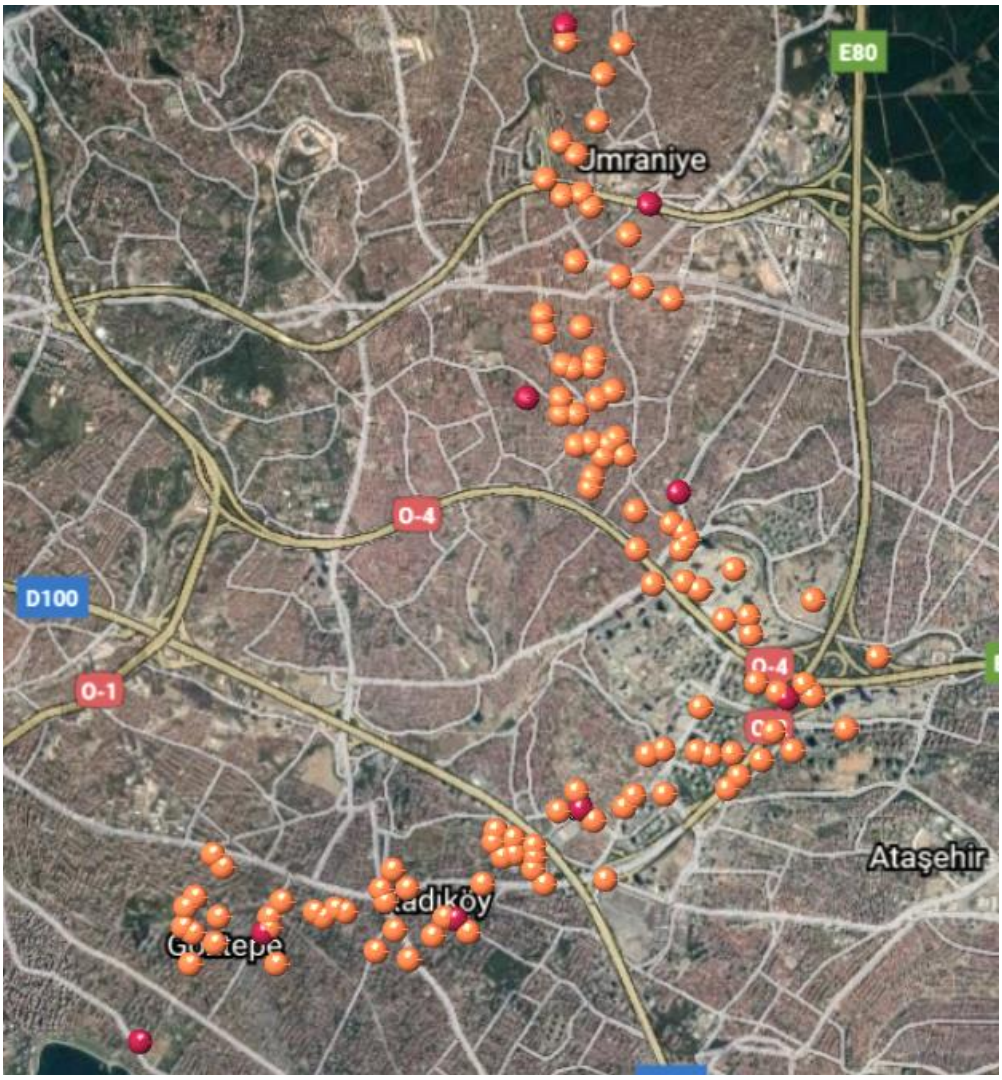


Fig. 17: Disrubition of grouped agents.

Considering that the spread is investigated through the metro route, its representation has been changed to make it easier to understand. To find too many factors and to make a route analysis, the swarm elements were further simplified and the slime mould swarm was drawn again with the determined parameters. The points coming from the Smaller Than component was grouped first, their list was made, and the centre points of the polygons were selected by creating a polygon from the points in this list. Then the centre points were converted back into a list and it was provided to output as points. Less point factors obtained were connected to each other with Interpolate Curve and thickness was given with the Pipe command, and the subway route formed by the behaviour of slime moulds between the stop points was revealed (Fig. 18).

The criteria affecting the design process of the existing metro line are defined on the model, such as food or threat. According to the characteristics of these data, Python codes were written to enable agents to move towards food or escape from the threat. In the last case, a metro route validating the existing metro line was obtained from the simulation. As a result, when all kinds of design criteria and environmental data are defined in the model, it will be possible to design new metro lines by using computational models in the future.

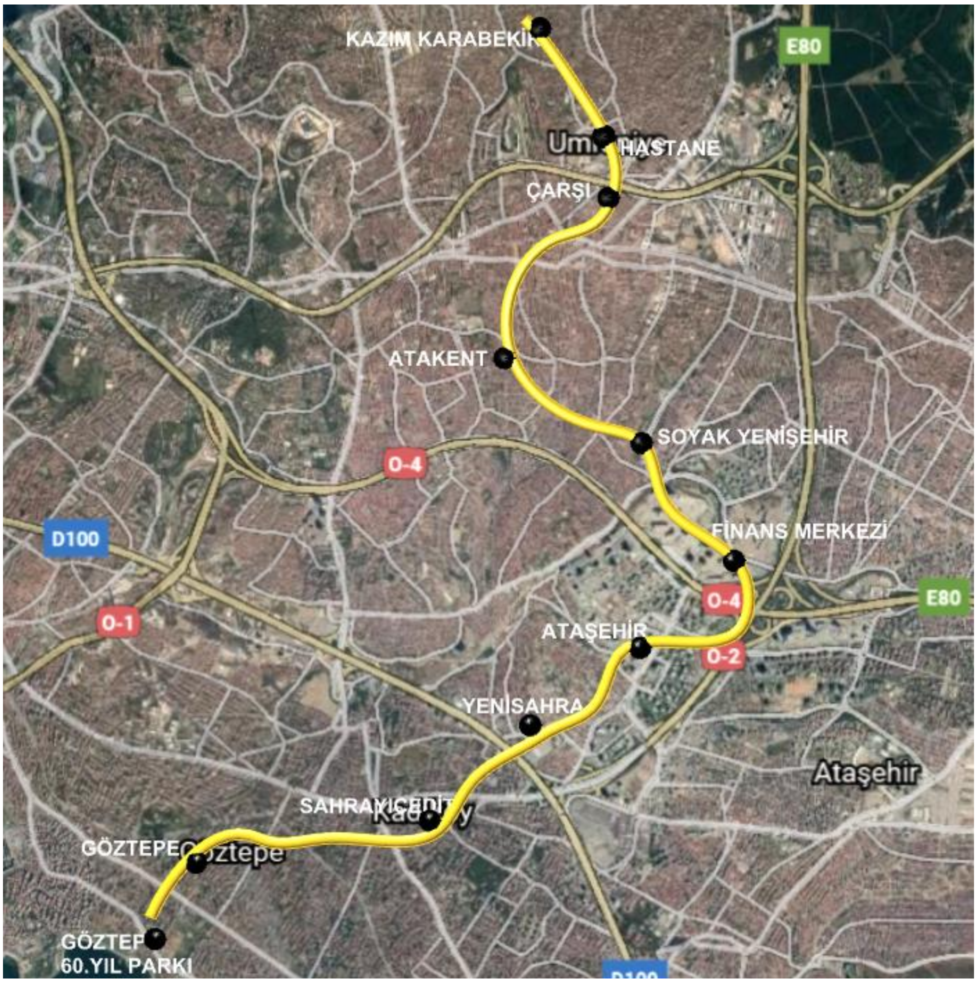


Fig. 18: The produce metro route in model.

Considering the regions where the route created by the factors in the model passes, when the station exits, subway route, rail system design criteria are evaluated together, it is revealed that the factors in the model must be intervened for three stations:

1. Göztepe Station: Since metro exits are close to public spaces, the metro route must be shifted towards Göztepe Park.
2. Sahrayı Cedit Station: Since the principle of integration is one of the main principles of metro design, it is inevitable that the tire vehicle road on the minibus street and the metro route are integrated.
3. Yeni Sahra Station: When evaluated in terms of cost, metro stations are built in regions with less expropriation. For this reason, passing through the location of the Metro Market Chain building will increase the cost, so the route should be shifted to another direction.

For the interventions to be made in the determined stations to be in one step, the GhPyhton component in the Grasshopper is used. The GhPyhton component, which allows adding Python code into the model, has two input and two output points that can be increased or decreased as desired. Entry and exit points are named according to the code piece to be written. Instead of creating unnecessary node load, the inputs connected to the GhPyhton component and the output nodes of the component have been obtained easier results (Fig. 19).

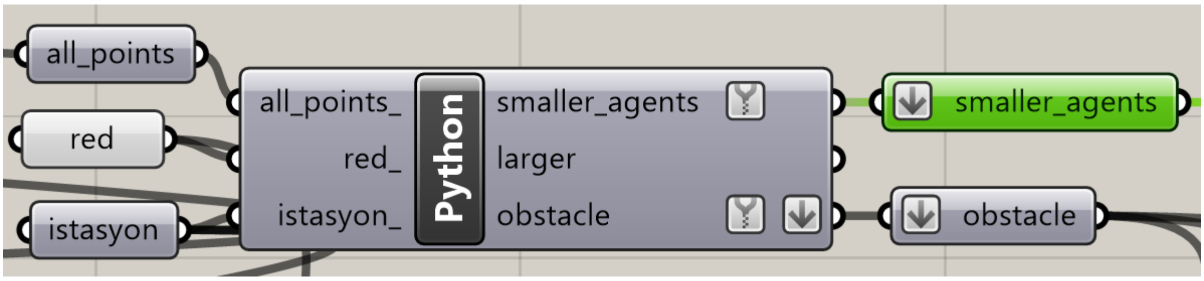


Fig. 19: Python components, inputs and outputs.

In this study, the rules required by the working logic of swarm intelligence were needed in order to obtain the return of the efficiency obtained in the laboratory experiments of slime moulds in the computer environment when the current route was taken. Some rules had to be specified in order to prevent the factors from dispersing, to keep them away from obstacles and to apply orientation force. For this reason, the rules determined with the Python component were included in the model and the model was visualized on the outputs obtained. First of all, the code written in the GhPyhton component works the same as the Smaller Than component used when determining the distance between the swarm elements; It enabled the transformation of factors from "point to list, list to point" and comparison of distances. After grouping the factors with the code in the component according to their distances to the stations, it was decided what kind of interventions would be made at the stations of Yeni Sahra, Göztepe and Sahrayı Cedit. Since it is thought that passing the line through Göztepe Park and Minibus Street will be more suitable for the design criteria, factors should be directed to these areas. In addition, since it would be more appropriate that the route passing through the private property zone does not pass through that point at the Yeni Sahra station; the line should be removed from there. For this reason, after determining the areas to be directed and distanced from, point groups called "red and green" were placed in the regions.

The rule set for barrier points was created by first connecting the red points to the component by connecting them from the nodes of the code written above. The code in the component was run with all factors, and outputs separated in two groups as obstacle and smaller_agents. The obstacle output was obtained by calculating the distance of the red point component that would be an obstacle to all factors. The points in the obstacle component were re-included in the simulation by changing their directions randomly when their distance to the red points was less than 4.00 units. As long as the simulation runs, the rule set is activated again and again, calculating the distances for each factor and changing the direction of the close ones every time. In order to change the direction of the points, the X, Y and Z coordinates of the points are extracted first. Later, the vector was created from the coordinates. Random component is connected to direction node by connecting vectors to *Rotate* component. Vectors whose directions are changed are divided into their coordinates and transformed into points (Fig. 20). After changing their directions, the factors continued to work with the Physarealm rules and proceed to the target stations. Except for the factors that move away from the obstacle, rules are also applied for the factors that arise each time.

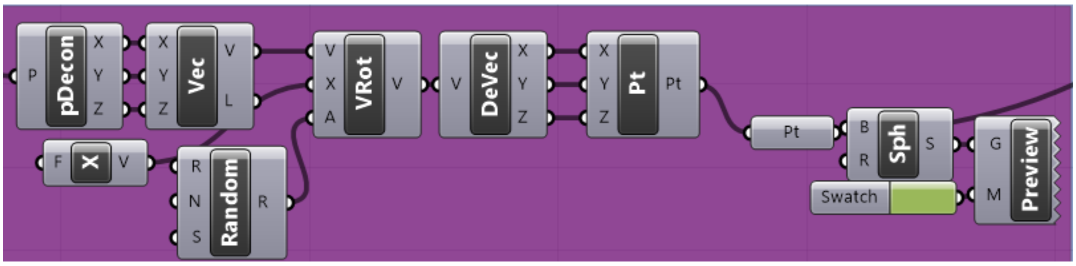


Fig. 20: Changing the direction of the points in the obstacle component.

When the location of the obstacle points connected to the red component was changed, the distances of all the factors to the obstacles were calculated each time, and the directions of the factors with a distance of less than 4 units were changed randomly (Fig. 21). Factors with changing direction are represented by the colour green, while the colour of the factors before the change is represented by dark blue. Dark blue factors were kept as a separate group in order to be able to determine their directions more easily.

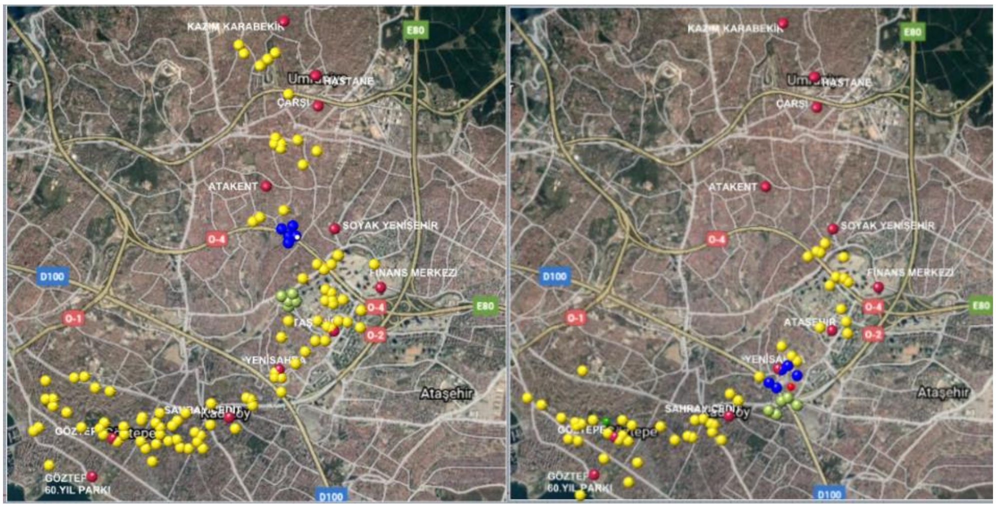


Fig. 21: Distribution of factors with changing directions.

The green component, which contains the points to be oriented, is connected to a new GhPython component with smaller_agents, which is the output of the GhPyhton code (Fig. 22).

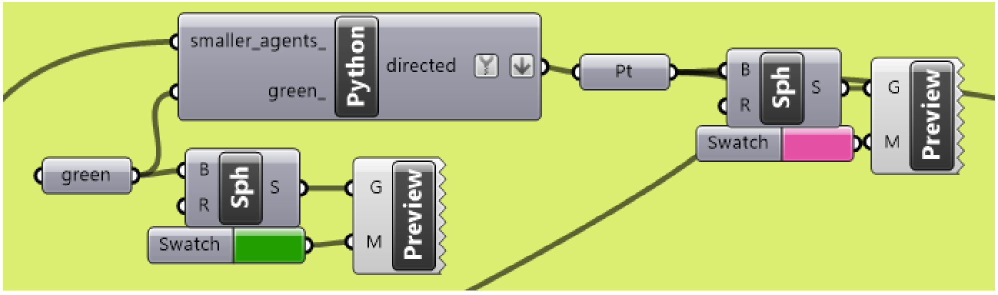


Fig. 22: Node sequence created for orientation.

The code and function of the component is as follows:

1. The library to be studied is called.
2. Entries connected from node points are introduced as points.

After grouping the factors according to their distance from the stations, the distances to the points to be directed to the obstacle are measured.

"If the distances of the factors are less than 8.00 units, write them in the directed list, if not, return and unlist." The factors with distances smaller than 8.00 are listed, vectors are created between green points and smaller_agents points, and smaller_agents points are moved to green points through these vectors. The distance of each factor approaching the green points was calculated, and those who were less than 8.00 were brought closer to the green points each time. As long as the simulation was running, the rules were applied over and over again for every agent born. When the Directed output is punctuated and visually connected to spheres and the simulation is started, the Python code ensures that these factors are moved to the points defined in the green component when the distance of smaller_agents to the green points defined in the green component is less than 8.00 (Fig. 23).

When it approaches the defined points in the green component, the yellow-coloured spheres turn pink. All of the factors that need to change their direction, the factors that change their direction, the factors to be oriented and the oriented factors are all worked in the last swarm. However, in this last case, the factors in the obstacle and directed components should be separated from the whole agent swarm, since the factors will be run twice and cause a malfunction. For this reason, the whole swarm was considered as a "tree" and repeating branches were removed from this tree (Fig. 24).

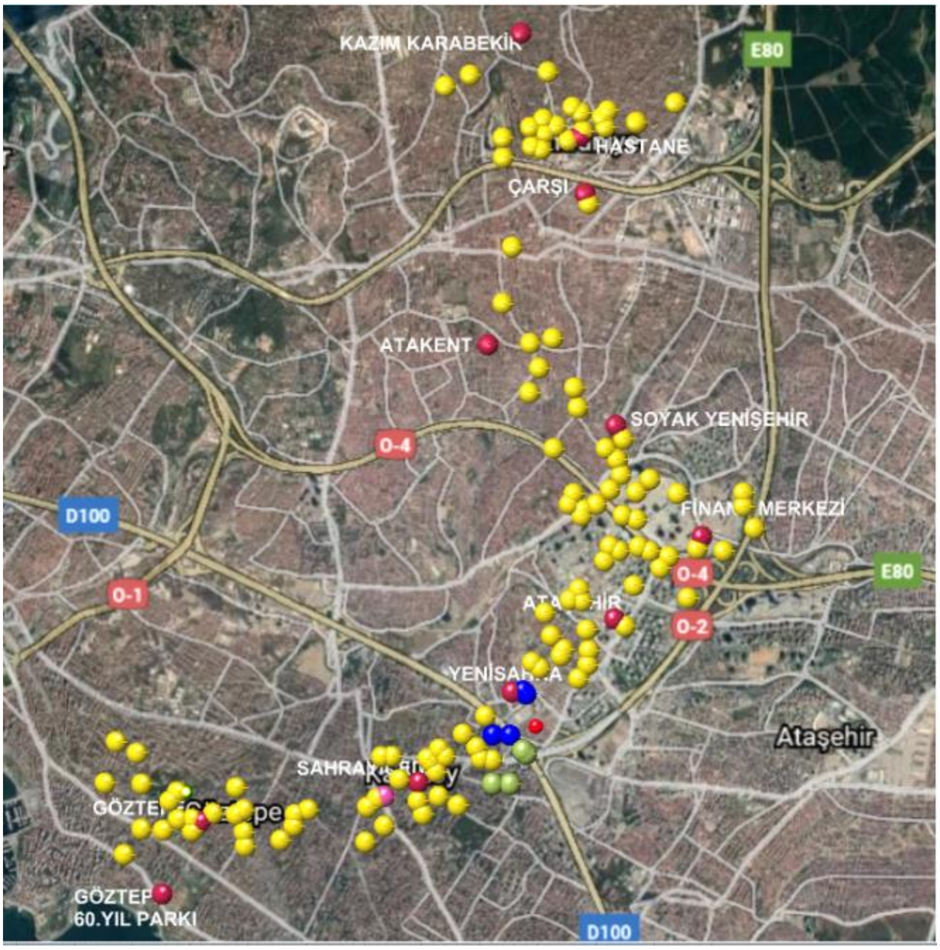


Fig. 23: Distribution of oriented factors.

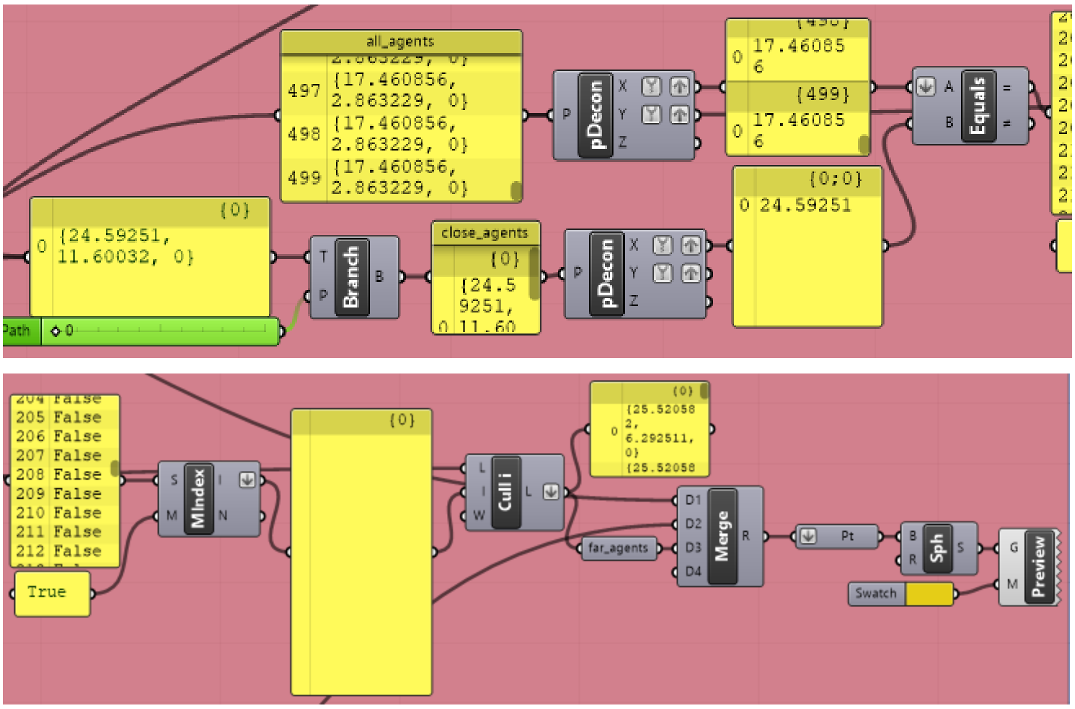


Fig. 24: Decomposition of the Agent Lists.

After removing the list of the factors that make up the slime mould simulation, whose direction was changed and directed to a specific coordinate, all the factors were collected in a single list with the Merge command. The list output has been taken point by point. In the last case, since there are many factors in the swarm, the curvilinear expression of the points is not possible in the Grasshopper environment, so the factors were taken into the Rhinoceros environment and the points were transformed into a single curve.

The model created with the factors in the Grasshopper has been simplified and expressed as a diagram (Fig. 25). Care has been taken to be similar to the expressions in the Grasshopper environment in order to minimize the complexity caused by the use of too many components and the excessive nodes of the components.

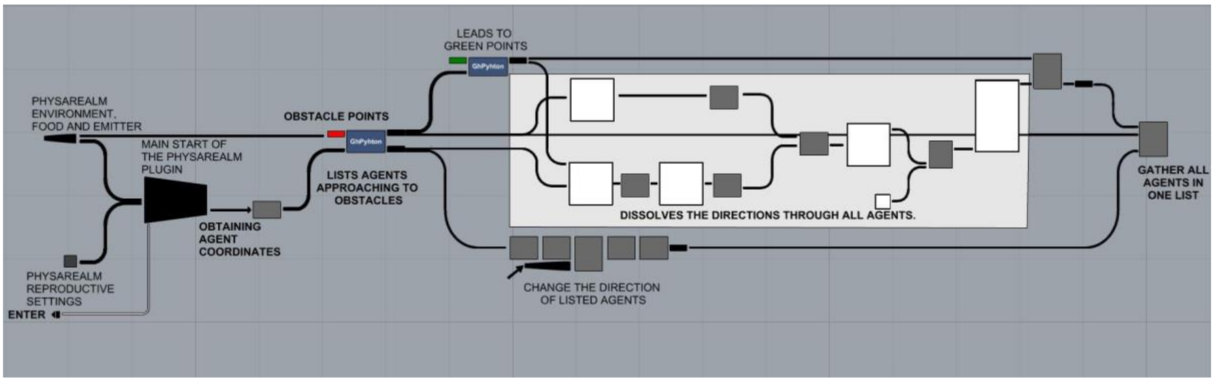


Fig. 25: Graphic representation of the model developed in grasshopper

It was observed that the factors carried out the metro route design without intervention after the reproduction parameters were determined (Fig. 26).

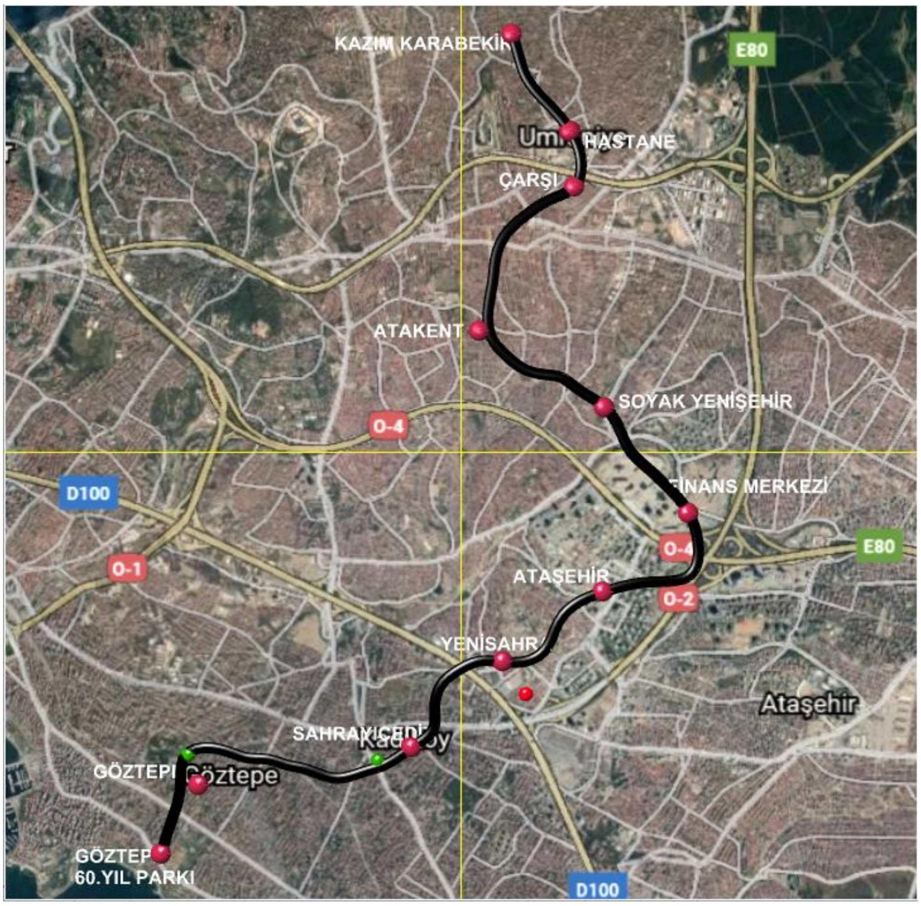


Fig. 26: The metro line obtained as a result of simulation.

In the light of the life cycles shaped by the emergence of living organisms, the metro route was drawn using a model, independent of the existing metro line, depending on the regions where the station locations of an existing metro line will be located. Considering the determined parameters and metro design constraints, it was thought that the resulting route should be intervened in three station regions. While they were removed from the area due to the inconvenience of the points, they passed through the Yeni Sahra station, they were directed to other regions by taking into account the integration at the Sahrayı Cedit and Göztepe stations. It has been observed that there is a great similarity between the metro route created by the agents as a result of these interventions and the Göztepe-Ümraniye metro route, which is currently under construction (Fig. 27).

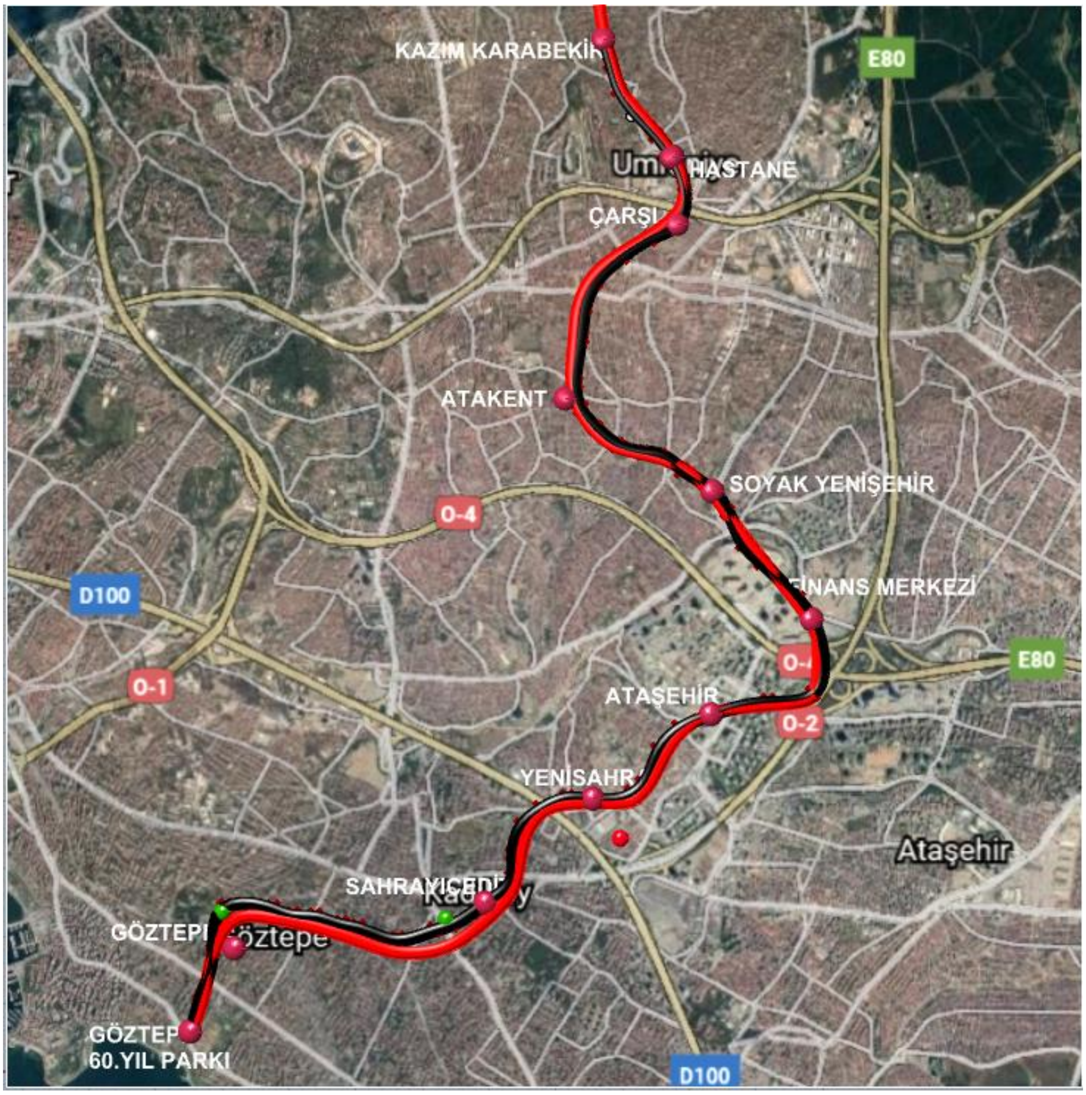


Fig. 27: Overlapping the existing route with the model (Existing line: red, Model line: black).

In the literature review conducted within the scope of this study, no computational model were found based on the life cycles of living organisms. In this context, the designs of metro routes provide more accurate and reliable results with modeling tools. Transportation networks, which are one of the important actors of urban design, can be completed in very laborious and long processes. It is an important issue that the processes involving many mathematical operations and computational burden are short, both in the design of a new metro line or in cases where the existing metro line will be extended. Especially when working on existing lines, faster but at the same time reliable results are needed to ensure the efficiency and continuity of transportation in a more integrated way with other transportation networks of local regions. When the results obtained in this study are examined, it is thought that the simulations made with computational models while designing metro routes will provide many gains in terms of time and money.

3 RESULTS

Swarm behaviour, in which high-level behaviours emerge, consists of interactions based on the cooperative behaviour of its constituents. The multiplicity of interacting units is just as important as the interactions. The collective orientation of slime moulds to food sources they cannot reach alone by communicating with the chemo-attractive trace among themselves is highly affected by environmental factors. This situation is similar to the design ecosystem. Although the metro routes are planned, they are among the essential components of the cities that produce macro-level results as long as they contain the human element.

In this context, this study proposes a metro design model that provides an advantage to urban design based on feedback-based swarm intelligence. In this model, which is based on the behaviour of slime moulds when reaching for food, it was first tried to find parameters suitable for the reproduction of the flock. It has been observed that DvMax, which is one of the parameters representing the birth and death rate of slime mould factors, should be greater than one (1) value. Since the distance between the factors is inversely proportional to the chemoattractive trace, it is clear that a range of 0 to 1 ensures the proliferation of the slime moulds. In addition, Sensor Setting parameters, which enable an agent to detect nutrients and other factors in the environment, have an independent effect from other parameters. To increase intelligibility in the slime mould population, the factors were grouped and represented by spheres. In cases where the distance between them is less than 4.00 units, the slime mould swarm tends to stations represented by food sources without dispersal. The metro design criteria determined the regions on the map where the metro route should not pass. After placing the spheres representing the substances that will repel the swarm in the determined areas, the slime fungus swarm was made to show an orientation again. Finally, it is seen that the existing metro route and the route created on the model by using only the station locations are highly similar to each other.

This study reveals that metro lines with specific design criteria, such as station locations and route lengths, can be designed quickly with the parameters specified in the model. This study, which ignored topography data, revealed the potential of designing transportation networks through digital models. Future studies aim to obtain more precise results by repeating the simulation considering the altitude data.

REFERENCES

Adamatzky, A., Martinez, G. J., Chapa-Vergara, S. V., Asomoza-Palacio, R., Stephens, C. R., (2010). Approximating Mexican highways with slime mould. Advances in Complex Systems, Vol. 10, No. 3, (pp. 1195-1214).

Adamatzky, A., Akl, S. G., (2011). Trans-canada Slimeways: Slime Mould Imitates the Canadian Transport Network. ArXiv.

Adamatzky, A., Yang, X., Zhao, Y., (2013). Slime mould imitates transport networks in China. International Journal of Intelligent Computing and Cybernetics Vol. 6 No. 3, (pp. 232-251)

Adamatzkya, A., Jones, J., (2009). Road planning with slime mould: If Physarum built motorways it would route M6/M74 through Newcastle, Vol. 20, No. 10, (pp. 3065-3084).

Adamatzky, A., Martinez, G., Lees, M., Sloot, P. M. A. (2012). Bio-development of Motorway Networks In The Netherlands: A Slime Mould Approach. Advances in Complex Systems Vol. 16, No. 02n03, 1250034.

Adamatzky, A., Prokopenkob, M. (2011). Slime mould evaluation of Australian motorways. International Journal of Parallel Emergent and Distributed Systems Vol.4, (pp. 275-295)

American Association for the Advancement of Science. (2010). Slime design mimics Tokyo's rail system: Efficient methods of a slime mould could inform human engineers. ScienceDaily.

Beni, G., Wang, J. (1990). Self-Organizing Sensor Systems. NATO ASI Series F: Computer and Systems Sciences. Vol. 58. (pp. 251-261).

Chang, W. (2016). Influence on Architectural Geometry by Emergent Design https://www.academia.edu/34757943/Influence_on_Architectural_Geometry_by_Emergent_Design 10.10.2020

Evangelidis, V., Jones, J., Dourvas, N., Tsompanas, M. A., Sirakoulis, G. Ch., Adamatzkya, A. (2017). Physarum machines imitating a Roman road network: the 3D approach. Scientific Reports.

Hensel, M., Menges, A., Weinstock, M. (2004). Emergence: Morphogenetic Design Strategies. Argentina, Wiley.

Hight, C. (2006). Collective intelligence in design. London: Wiley-Academy.

Hofstadter, D. (1979). Godel, Escher, Bach: An Eternal Golden Braid. New York. Basic Books.

Heylighen, F. (1989). Self-Organization, Emergence and the Architecture of Complexity. Proceedings of the 1st European Conference on System Science, (AFCET, Paris), (pp. 23-32).

Holland, O. & Melhuish, C. (1999). Stigmergy, self-organization, and sorting in collective robotics. Artif. Life, 5, 173--202. doi: 10.1162/106454699568737.

Holland, J. H. (2006). Studying Complex Adaptive Systems. Journal of Systems Science and Complexity Vol.19(1). (pp.1-8). http://hdl.handle.net/2027.42/41486. 25.05.2020.

Jones, J. (2010). Characteristics of Pattern Formation and Evolution in Approximations of Physarum Transport Networks. Artificial Life Vol.16, (pp. 127-153).

Johnson, S. (2001). Emergence. The New York Times. Retrieved from http://www.nytimes.com

Keller, E. F. (1983). The Force of the "Pacemaker" Concept in Theories of Aggregation in Cellular Slime Mould. Perspectives in Biology and Medicine. 26, No. 4, 515521.

Leach, N. (2009). Swarm Urbanism. Architectural Design Vo.:79, (pp. 56-63).

Padgham, L., & Winikoff, M. (2005). Developing intelligent agent systems: A practical guide (Vol. 13). John Wiley & Sons.

Russell, S. ve Norvig, P. (1995). Artificial Intelligence: A Modern Approach. Prentice Hall, Upper Saddle River, N.J.

Rutten, D. (2007). Rhinoscript10, Robert McNeel & Associates. (pp.4).

Sander, L. (2010). Slime Mould Grows Network Just Like Tokyo Rail System. Wired, from https://www.wired.com/2010/01/slime-mold-grows- network-just-like-tokyo-rail-system/. 02.06.2020.

Schelling, T. (1978). Micromotives and Macrobehaviour. N.Norton. London.

Snooks, R. (2014). Behavioral Formation Multi-agent Algorithmic Design Strategies. School of Architecture and Design College of Design and Social Context, RMIT University.

Stevens, R. Sosic, R. (1995). Emergent behaviour in slime mould environments, Proceedings of 1995 IEEE International Conference on Evolutionary Computation, Perth, WA, Australia.

Toht, A., Nakagaki, T. (2000). Intelligence: Maze- Solving by an Amoeboid Organism. Nature: Vol. 407. (p. 470).

Wiesenhuetter, S., Wilde, A., Noennig, J. R. (2016). Swarm intelligence in architectural design. Lecture Notes in Computer Science, (pp. 3-13).

Url-1<http://archieboytom.blogspot.com>, 15.02.2020

Url-2 < http://www.tyler-johnson.com/Swarm-Intelligence>, 09.07.2021

Url-3< https://www.wired.com/2010/01/slime-mold-grows-network-just-like-tokyo-rail-system/>, 22.03.2020

Url-4< https://www.wired.com/2010/01/slime-mold-grows-network-just-like-tokyo-rail-system/>, 23.03.2020

Url-5< http://ma-yidong.com/2016/09/13/physarealm/>, 17.03.2020

Url-6< http://ma-yidong.com/2016/09/13/physarealm/>, 02.03.2020

Url-7< http://ma-yidong.com/2016/09/13/physarealm/>, 02.03.2020

Url-8< http://ma-yidong.com/2016/09/13/physarealm/>, 02.03.2020