IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received December 17, 2019, accepted December 31, 2019, date of publication January 6, 2020, date of current version January 15, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2964002

A Node Selecting Approach for Traffic Network Based on Artificial Slime Mold

ZHENGYING CAI, ZEPING XIONG, KUNPENG WAN, YAQI XU, AND FAN XU
College of Computer and Information Technology, China Three Gorges University, Yichang 443002, China

Corresponding author: Zhengying Cai (master_cai@163.com)

This work was supported in part by the National Natural Science Foundation of China under Grant 71471102, in part by the Science and Technology Research Program of Hubei Provincial Department of Education in China under Grant D20101203, and in part by the Yichang University Applied Basic Research Project in China under Grant A17-302-a13.

ABSTRACT The node selecting problem of traffic network is a significant issue and is difficult to be solved. In this paper, an artificial slime mold method is proposed to help us solve the problem. First, the chief components of an artificial slime mold are introduced to simulate the foraging behavior of a true slime mold, including external food sources, plasmodium, myxamoeba, nucleus, and nutrients. Then the learning mechanism of nutrient concentration for the artificial slime mold is illustrated, though there is no brain or neuron in its body. After that, the node selecting approach is described according to the propagation capabilities of nodes. Second, the algorithm flow is designed to show how to solve this kind of complex selecting problem. The algorithm flow to select important traffic nodes by artificial slime mold is composed of 4 main steps, including initialization, food searching, feeding, and selecting for output. Third, a comprehensive example is designed and derived from references to certificate that the proposed artificial slime mold can help us select important traffic nodes by their generated traffic topologies. The contributions of this paper are important both for traffic node selecting and artificial learning mechanism in theoretical and practical aspects.

INDEX TERMS Traffic network, node selecting, artificial intelligence, slime mold, foraging behaviour.

I. INTRODUCTION

Node selecting is often used in all kinds of networks, including traffic networks, to help us select the most important nodes or edges. As we all know, the important nodes or edges in a traffic network play a significant role in network operation. Reference [1] studied the node importance evaluation of the high-speed passenger traffic complex network based on the Structural Hole Theory. Congestion or disturbance taking place in the important nodes or edges will easily soon be propagated to other parts of the traffic network, and the traffic network efficiency will also be reduced. The importance of nodes or edges is strongly related to their positions in the network, i.e., [2] designed a location-aware and node ranking value-assisted embedding algorithm for one-stage embedding in multiple distributed virtual network embedding. Hence, the propagation effect is decided by the network topology and the importance of a node selecting, namely the positions of different nodes or edges. To solve this problem, [3] used deep learning to research incorporating network structure with node contents for community detection on large networks.

These researches verified that the node selecting of the traffic network should consider their importance, and the selecting problem is, in essence, a learning problem [3]. For a traffic network, node importance is connected with its propagation capability under all kinds of disturbances. Reference [4] put forward a DeepRank to improve unsupervised node ranking via link discovery. Hence, to evaluate the propagation capability of a traffic node may help us analyze the spreading dynamics and reconfigurable topology. In early 1995, [5] surveyed basic routines for the rank-2k update with 2D torus vs reconfigurable network. Spreading dynamics and reconfigurable topology of traffic networks are so pervasive that this aspect of researches might shed their light on most networks in the real world. Reference [6] gave some node centrality indices in food webs for rank orders versus distribution, and [7] described a virtual network embedding through topology-aware node ranking.

To rank the node importance, there are many traditional measures, including degree [1], location-aware [2], node content [3], link discovery [4], node centrality [6],

The associate editor coordinating the review of this manuscript and approving it for publication was Keivan Navaie.

topology-aware [7] and many other topology-based methods [8]. For example, [8] showed the impact of rank attack on network topology of routing protocol for low-power and lossy networks. Different methods can get different evaluations, bringing great difficulty to the optimizing and scheduling of traffic networks. Fortunately, in the past decade, some researchers used a strange living thing, slime mold, to solve the traffic network optimization problems. In 2010, Atsushi, et al., published a paper in *Science* about the rules for biologically inspired adaptive network design, where a slime mold is applied to successfully solve the traffic planning problem of the Tokyo railway [9]. In the same year, Adamatzky et al. also used a physarum to build motorways to route M6/M74 through Newcastle [10]. In 2011, Adamatzky et al. again depicted approximating Mexican highways with slime molds [11].

The contributions in this paper are as follows. First, a node selecting model based on the propagation topology of the traffic network is proposed after the background analysis. Second, an artificial slime mold is proposed to rank the network nodes by their propagation capabilities and topology importance. The proposed artificial slime mold includes the main structural organizations of a true slime mold and can simulate its expansion and contraction behavior. Third, a comprehensive example is presented to test the proposed model, and the results reveal that the artificial slime mold can help us rank the traffic network nodes. Finally, the paper is summarized and future research directions are pointed out. The contributions of this paper are of both theoretical and practical importance for traffic network optimization and artificial intelligence theory.

II. RELEVANT WORK

In the past two decades, there are numerous researches and investments in the traffic network and node selecting. Some of them have been deployed in practice like railway, highway, and road systems [1]–[11]. Well-designed traffic systems provide passengers with the least time consumption on traveling within destination areas through variable route selections. Reference [12] stated a super edge rank algorithm and its application in identifying opinion leaders of online public opinion supernetwork, and [13] concerned a new mutually reinforcing network node and link ranking algorithm. They applied ranking algorithms mainly on the important nodes and areas to optimize the traffic topology according to the total traffic volume. Other researches typically used spreading ability or destructiveness to rank nodes which have been proven in practice to effectively utilize traffic space and decrease traveling time. Reference [14] ranked the spreading ability of nodes in the network core, and [15] maximized the destructiveness of node capture attack in wireless sensor networks.

Most of the research work above focused on how to detect the node importance to rank node [1]–[15]. However, those methods still not solved some problems. The selecting of traffic nodes is decided by the whole traffic network where unreasonable topology will easily lead to higher traffic congestion while disturbance or congestion. Also, this may lead to the most known phenomenon of network congestion or unbalanced traffic flows. Reference [16] proposed a GEVD-based low-rank approximation for distributed adaptive node-specific signal estimation in wireless sensor networks, and [17] introduced FRANK as a fast node ranking approach in large-scale networks. Some researchers found it is important to collect the data on the traffic network and adopt a learning mechanism to realize topology development [3], [4]. [18] introduced an efficient mapping algorithm with a novel node-ranking approach for embedding virtual networks, and [19] used a spectral learning algorithm to reveal the propagation capability of complex networks. [20] described an inverse-square law to identify influential nodes in complex networks, and [21] employed the evidence theory to identify node importance.

But it is more difficult to efficiently evaluate the propagation capabilities by network data, i.e., [22] described an automated optimization of intersections using a genetic algorithm. The researches above applied a lot of complex artificial intelligence algorithms, such as deep learning [3], [4], spectral learning [19], genetic algorithm(GA) [22]. In 2010, a novel slime mold method was used for traffic network optimization [9]–[11]. After that, more researchers began to apply it to solve the optimization problems of traffic networks. In [23], Andrew Adamatzky's research team again applied an improved physarum polycephalum algorithm for the shortest path problem. Reference [24] designed an efficient physarum algorithm for solving the bicriteria traffic assignment problem, and [25] considered physarum machines to imitate a Roman road network with a 3D approach.

Different from most living things using a brain or neurons to learn about the environment, the slime mold can only learn by a single-cell structure without any brain or neurons. The above learning algorithms [3], [4], [19], [22] are different from the single-cell slime mold, and cannot illustrate its operation mechanism. More importantly, the described slime mold in our paper is an artificial intelligence algorithm to solve the planning problems of traffic networks by simulating the foraging behavior of a real slime mold, rather than a true slime mold in [9]–[11], [23]–[25].

Although [9]–[11], [23]–[25] got great advantages in solving the network planning problem by a real slime mold, there are still some apparent shortcomings. First, it will cost a lot of time for a real slime mold to find a feasible solution (such as 26 hours in [9], 70 hours in [11], and 96 hours in [23]). Second, solving precision is too low where the petri dish is used as a traffic map and the oats are applied as traffic nodes. Third, it is very difficult to directly operate real slime mold to solve these problems and the whole experimental process needs professional biological skills and expensive equipment. Fourth, a lot of space, time, and materials should be spent in other activities not directly related to the experiment, such as the cultivating of slime molds, living environment control, and mass data processing. Additionally, until now, the researches on both the node selecting of traffic networks and slime mold's foraging behavior are still very few.

Hence, this paper gives us a new method to solve the node selecting problems of traffic networks, and the proposed artificial intelligence algorithm will help us in similar problem solving and decision making.

III. MODEL DESCRIPTION

A. CHIEF COMPONENTS OF ARTIFICIAL SLIME MOLD

The proposed artificial slime mold is based on references [9]–[11], [23]–[25] and simulates the foraging behavior of a true slime mold to solve the node selecting problem. In references [9]–[11], [23]–[25], an experimental system of slime mold includes dish, map, external food sources, and a slime mold, where a slime mold composes of a plasmodium, multiple myxamoebas, one or more nucleus, and nutrients, as shown in Figure 1.

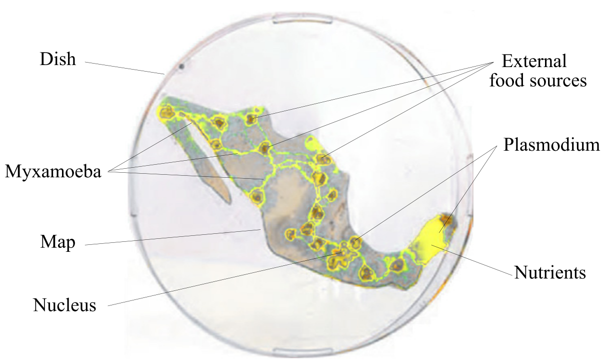

FIGURE 1. An experimental system of slime mold [11].

In our artificial slime mold, the chief components are as follows.

i) External food sources. They are around the artificial slime mold and act as the searched goals of traffic network nodes. The slime mold will look for all the external food sources to get the nutrients to support its survival. The found external food sources will be digested to be nutrients in the body of an artificial slime mold.

ii) Plasmodium and myxamoebas. They are the main tools for problem solving, and the main operations are expansion and contraction. A single-cell plasmodium without fixed size or shape can freely move on the food surfaces. The tentacle-shaped myxamoebas are the deformed structure of the plasmodium, and multiple myxamoebas can continuously search the external food sources in parallel. The plasmodium and myxamoebas will digest the found food sources and transport them into its body through many myxamoebas by a parallel probability searching algorithm.

In food searching, the longer and bolder myxamoebas are very helpful to increase transporting capacity and the searching probability to find more food sources but will cost more energy and nutrients at the same time. On the contrary, shorter and shiner myxamoebas will consume less but can only find less food. After food searching, the plasmodium and myxamoebas will tell us the importance of nodes or edges in the traffic network.

iii) Nucleus. It acts as the center of artificial slime mold and the starting point of food searching, and a slime mold can only move or feed around the nucleus or it will not survive without the nucleus. The nucleus is often acting as the origin node of a traffic route or the most important node in the traffic network.

iv) Nutrients. They provide the most significant sources of energy and materials for the survival of slime mold and play an important role in the learning mechanism of artificial slime mold or information transmission between different myxamoebas. The nutrients all over the slime mold's body come from the external food sources by feeding behavior and will be continuously consumed in almost all life activities of slime mold, such as food searching, moving, the expanding and contracting of myxamoebas, feeding and digesting, etc. The nutrients act as the information media for its parallel computing.

Each component above of the artificial slime mold will collaboratively work in problem solving, and the whole solving process for the traffic network includes two main stages, namely the food searching stage and feeding stage. The first food searching stage makes myxamoebas continuously expand around the nucleus to search external food resources, and the expanding behavior is constrained around the nucleus. The second feeding stage makes myxamoebas continuously contract to form an optimized traffic network to transport the absorbed nutrients from the food source nodes into its body.

The area of a slime mold will continuously change from small to big in the food searching stage. The artificial slime mold can sense external nodes in its myxamoebas' expanding and interact with the surrounding environment, to simulate the biochemical reactions between myxamoebas and environment to identify food sources or non-food sources.

Conversely, the area of a slime mold will continuously contract from big to small in the feeding stage. Similar to a true slime mold digesting the external food sources and transporting the digested nutrients into its body, the artificial slime mold can also form an optimal traffic network by a probability search algorithm and parallel computing mechanism.

There is a self-learning mechanism in the artificial slime mold to search multiple food sources and transport the nutrients from multiple food sources into its body at the same time. The slime mold will deform its plasmodium and myxamoebas to form an optimal network structure according to the nutrient concentration, that is, the routes with higher nutrient concentration are where the plasmodium and myxamoebas are. After a lot of computing iterations of computing for expansion and contraction, the artificial slime mold will form an optimal topology as a feasible solution with high efficiency and robustness.

B. LEARNING MECHANISM OF SLIME MOLD

A traffic network to be solved can be described as a directed graph $G = (N, E)$ with n nodes and multiple edges, there is

$$G = (N, E) \tag{1}$$

The $N = [(x_i, y_i)]_n$ is a node matrix composing of n traffic nodes; the $E = \{e_{ij}|i, j \in N\}$ is an edge matrix describing the relationship between the nodes. For all known traffic nodes, the distance matrix of edges can be calculated as:

$$D = [d_{ij}]_{n\times n} = [|e_{ij}|]_{n\times n} = [\sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}]_{n\times n} \tag{2}$$

i) Parallel computing. Multiple myxamoebas search traffic nodes by parallel expanding in multiple directions, and then transport the nutrients into its body by parallel contracting after the finding of traffic nodes. Similar to a true slime mold, multiple myxamoebas in an artificial slime mold try to find the traffic nodes in parallel as many as possible. Multiple myxamoebas will collaborate to search all external traffic nodes in all directions, and share information of all traffic nodes and myxamoeba routes.

As a time-varying structure, the traffic network G can be noted as a $n \times n$ matrix at the time t, there is

$$\mu = [\mu_{ij}(t)]_{n\times n} \tag{3}$$

where

$$\mu_{ij} = \begin{cases} 1, & \text{there is a direct edge from node } i \text{ to } j \\ 0, & \text{there is no direct edge from node } i \text{ to } j \end{cases}$$

Assuming w_{ij} is used to describe the flow volume of an edge between node i and j at the time t, the traffic flow in every edge e_{ij} will also continuously change to form an $n \times n$ weight matrix, there is $w = [w_{ij}(t)]_{n\times n}$.

ii) Iterative computing. The myxamoebas will continuously expand to find traffic nodes and contract into optimal transporting routes to connect all nodes with time-varying. After many iterations of expanding and contracting, it is possible to form an optimal traffic network. For all myxamoebas, it is not compulsory to repeatedly pass through all nodes they have passed. At the end of iteration computing, the error varying and topology changing will become less and less, then the solution can be output as a feasible answer.

The nutrient concentration matrix on the edge $V = \{e_{ij}|i, j \in N\}$ at time t is

$$\tau(t) = [\tau_{ij}(t)]_{n\times n} \tag{4}$$

In every iterative computing, there are two ways to adjust the nutrient concentration, namely enhancing operation and decreasing operation.

In the enhancing operation, the nutrient concentration $\tau_{ij}(t)$ on the corresponding traffic route e_{ij} will be enhanced to simulate nutrient transportation on the shared traffic routes by multiple myxamoebas in a true slime mold. Then the nutrient concentration $\tau_{ij}(t)$ on the route will be higher. There is

$$\tau_{ij}(t) = \tau_{ij}(t-1) + \Delta_{ij}(t), \quad \Delta_{ij}(t) \geq 0 \tag{5}$$

In the decreasing operation, the nutrient concentration $\tau_{ij}(t)$ on a traffic route v_{ij} will be reduced to simulate the nutrient consumption and energy expenditure in the life activities of a true slime mold. There is

$$\tau_{ij}(t) = \tau_{ij}(t-1) - \sigma_{ij}(t), \quad \sigma_{ij}(t) \geq 0 \tag{6}$$

After iterative computing, the routes can be easily ranked by the values of nutrient concentration, which will help us rank traffic nodes.

iii) Shared computing. Although there is no brain or neuron in its body, a single-cell slime mold can still transmit information to learn the environment. The nutrient concentration in artificial slime mold can timely provide information about traffic nodes and transporting routes to other myxamoebas. If there is no traffic node in expanding areas, the myxamoebas will only contract as a true slime mold used up all the nutrients and energy on this area. Additionally, when a traffic network node is found the myxamoebas will tend to use the already formed transporting routes so as to further strengthen the nutrient concentrations on the shared routes.

It is assumed that there are m myxamoebas traverse all traffic nodes or edges, and the $k(\leqslant n \leqslant m)$ th myxamoeba has traveled c_k nodes. $i_1 \sim i_{c_k}$ are the nodes it passed through, and the edge set of the $k(\leqslant n \leqslant m)$ th myxamoeba to form its route L_k can be expressed as:

$$L_k = \{e^k_{12}, e^k_{23}, \ldots, e^k_{l-1,l}, \ldots, e^k_{c_k-1,c_k}\} \tag{7}$$

where L_k is an arrangement of a subset in the edge set $E = \{e_{ij}|i, j \in N\}$, and the last node i_{c_k} is the present node arrived. The whole route length of myxamoeba $k(\leqslant n \leqslant m)$ will be calculated by formula (2), there is:

$$\begin{aligned} D_k &= \sum_{l=2}^{c_k} d^k_{l-1,l} = \sum_{l=2}^{c_k} |e^k_{l-1,l}| \\ &= \sum_{l=2}^{n} \sqrt{(x^k_{l-1} - x^k_l)^2 + (y^k_{l-1} - y^k_l)^2} \end{aligned} \tag{8}$$

All the m myxamoebas can be calculated in the order of node number $1 \leq k \leq m$, and each node will be solved every time. All myxamoeba traffic routes can be summed up to form the whole traffic network, that is:

$$\begin{aligned} L(t) &= [L_k(t)]_m \\ &= [\{e^k_{12}(t), e^k_{23}(t), \ldots, e^k_{l-1,l}(t), \ldots, e^k_{c_k-1,c_k}(t)\}]_m \end{aligned} \tag{9}$$

All myxamoebas can learn the route selection information of the whole traffic network by the selection matrix, there is:

$$\mu = [\mu_k]_m = [[\mu^k_{ij}(t)]_{n\times n}]_m \tag{10}$$

iv) Probability computing. The food searching process of an artificial slime mold is by probable expanding, and the feeding process to find an efficient traffic network is also by probable routing. Hence, the myxamoebas can try different topologies each time to expand the unexplored areas or to contract to search for more efficient traffic networks.

The myxamoeba learns two kinds of experience for route searching. On the one hand, it learns its own experience by a certain probability $\alpha_{ij}^k(t) \in [0, 1]$, namely self-learning experience. On the other hand, it learns the experience from the neighbor's myxamoebas by another probability $\beta_{ij}^k(t) \in [0, 1]$, namely the neighbor-learning experience. So all myxamoebas will learn from their own and their neighbors to collaboratively build a global optimal route $L^* = [L_k^*]_m = [\{e_{12}^k, e_{23}^k, \ldots, e_{l-1,l}^k, \ldots, e_{c_k-1,c_k}^k\}]_m$, and the next transiting probability $p_{ij}^k(t) \in [0, 1]$ of each myxamoeba k can be expressed as a sum of two kinds of experience. There is,

$$p_{ij}^k(t) = \alpha_{ij}^k(t) + \beta_{ij}^k(t) \tag{11}$$

$$\alpha_{ij}^k(t) \in [0, 1] \tag{12}$$

$$\beta_{ij}^k(t) = \frac{\tau_{ij}^k(t)}{\sum\limits_{i,j=1}^{n} \tau_{ij}^k(t)} \tag{13}$$

where $\beta_{ij}^k(t)$ is related to the nutrient concentration $\tau_{ij}^k(t)$, so each myxamoeba can easily learn the routes with higher nutrient concentrations; but $\alpha_{ij}^k(t)$ is used to avoid prematurely trapping in the local optimal solutions and improve the global searching ability.

v) Artificial computing. The so-called learning mechanism of artificial slime mold can imitate the true slime mold to identify and learn food sources and other non-food objects. After a very long evolutionary process, learning mechanism has been built in slime molds, that is why [9]–[11], [23]–[25] used them to solve traffic planning problem. Similarly, by parallel computing, iterative computing, shared computing, and probability computing, the proposed artificial slime mold can learn and identify the characteristics of food and non-food through affinity and biochemical reactions and allows different myxamoebas be able to learn the experience of its own and its neighbors to form an optimal traffic network topology. The more important thing is a generated optimal network topology reflects the propagation capability of the node, so it is believable to apply artificial slime mold to solve the node selecting problem of traffic network.

C. NODE SELECTING BY PROPAGATION CAPABILITY

The node selecting problem of traffic network is, in essence, to identify the importance of nodes or edges on a network topology. According to formulas (1)~(3), the importance of traffic network nodes or edges depends on their positions in the existing topologies, and some indexes can be derived from the position of nodes and edges.

First, node degree can directly give a reference for node importance [3]. The greater the traffic volume or the more the edges pass through a node i, the more important the node is. Its definition is as follows:

$$Dg_i(t) = \sum\nolimits_j \mu_{ij}(t) \tag{14}$$

where $Dg_i(t)$ is the degree of traffic node i. To apply it in networks with different sizes, $Dg_i(t)$ needs to be further transformed into a ratio of the network size. Because in a network with n nodes, at most $n-1$ neighbor nodes may connect to the node i where $\max\{Dg_i(t)\} = n - 1$, the importance degree index of a node i can be given as:

$$\overline{Dg}_i(t) = \frac{Dg_i(t)}{\max\{Dg_i(t)\}} = \frac{Dg_i(t)}{n-1} \tag{15}$$

Second, besides the number of connecting edges, the flow volume is also an important attribute of an edge, namely the closeness index for node importance [4]. Let $w_{ij}(t)$ be the flow volume of station node i, describing the load of the traffic network. Hence, the flow volume of a node composes of all flows running through it, that is

$$w_i(t) = \sum_{j=1}^{n} w_{ij}(t) \tag{16}$$

where $w_{ij}(t)$ includes the flow volume of in and out traffic on edge e_{ij}. So the closeness index can be defined as follows:

$$\overline{Cl}_i(t) = \frac{w_i(t)}{\sum_{i=1}^{n} w_i(t)} \tag{17}$$

Third, the betweenness index notices that the shortest routes often have an important influence in the traffic network [6]. The more the shortest routes connecting to a node i, the more important the node i is. Let $\lambda_{jk}(t)$ be the number of the shortest routes from node j to k passing through a node i at the time t. The betweenness index $Bt_i(t)$ of a traffic node i can be calculated as

$$Bt_i(t) = \sum_{j=1}^{n} \sum_{k=1}^{n} \lambda_{jk}(t) \tag{18}$$

Furthermore, in a network with n nodes, at most $n - 1$ neighbor nodes may connect to a node i, and the betweenness degree index of a traffic node i can be given as follows:

$$\overline{Bt}_i(t) = \frac{Bt_i(t)}{\sum_{i=1}^{n} Bt_i(t)} \tag{19}$$

In a complex network, the importance of fringe nodes and central nodes is mostly not equal, and the topology connectivity can be used to evaluate the node's importance. If the average degree of a network topology is relatively higher, the network is more complex.

Forth, different from the indexes above depending on their relative positions in the existing topologies, we try to select the important nodes by a slime mold according to the propagation capabilities of nodes. In our opinion, an important node easily leads to a simpler topology connecting all nodes, which reflects its propagation capability or spreading ability. After several iterations, all nodes may be linked in a newly generated network by multiple myxamoebas. When the stopping condition of the solving process is satisfied, an optimal topology connecting all the nodes will be output. Now, the total degree of the traffic network can be used to evaluate the complexity of the generated network of the node i:

$$Dg_i(t) = \sum\nolimits_j Dg_j(t) \tag{20}$$

Then the number of all edges on the generated network of the node i can also be gotten

$$E_i(t) = \sum\nolimits_j \sum\nolimits_k \mu_{jk}(t) \tag{21}$$

Then the total distance of all myxamoebas will also be gotten according to formula (2), and the degree index in formula (15), closeness index in formula (17), and betweenness index in formula (19) can all be gotten to evaluate the generated networks.

According to the index (20) above gotten by artificial slime mold, a useful new index can be gotten, that is the average degree of the network originated from node i, called as redundancy rate index here:

$$Rr_i = \Sigma_j Dg_j(t)/n \tag{22}$$

Different from the degree index, closeness index and betweenness index of nodes above, the proposed redundancy rate index depicts the propagation capability or spreading ability of a node i by its originated traffic network. The redundancy rate index includes the total degree of all nodes and the total number of nodes in its generated network, which can help us rank all nodes by the calculating of an artificial slime mold.

D. OBJECTIVE FUNCTION OF NODE SELECTING

The whole route of the traffic network includes the total length of all m myxamoebas and subtracts the length of overlapping paths of the myxamoebas.

$$D_{\Sigma m} = \sum_{k=1}^{m} D_k - \sum_{i,j=1}^{n} d_{ij}|_{overlap} \tag{23}$$

After a lot of computing iterations, each node can be originated to form an optimal traffic network with different importance degree. In each case, the m myxamoebas can obtain the shortest routes traversing all nodes:

$$L^* = [L_k^*]_m = [\{e_{12}^k, e_{23}^k, \ldots, e_{l-1,l}^k, \ldots, e_{c_k-1,c_k}^k\}]_m \tag{24}$$

By the optimal routes L^*, it is easy to get the importance index of each network topology as the objective function. This selecting method can evaluate the important capability of each node to form an optimal network with the shortest routes. When one or more failure takes place in a traffic network, different nodes can form different topologies to connect all nodes with the shortest distance instead of only depending on their relative positions in existing topologies. There is:

$$\begin{aligned} obj : & Rank\{\max(\overline{Dg}_i), \max(\overline{Cl}_i), \max(\overline{Bt}_i), \\ & \min(Rr_i), \min(D_{i\Sigma m}^*)\} \\ & st : i, j \in (1, n), \quad k \in (1, m), \ L \in [0, L_{\max}] \end{aligned} \tag{25}$$

Unlike the traditional node selecting methods with a single importance index [1]–[8], [12]–[21], the objective function is a multi-objective function with different objectives and constraints. The multi objectives can help us rank different nodes by their positions and propagation capabilities according to their originated optimal network topologies with shortest routes traversing all nodes. This novel selecting method can tell us which node is the most suitable for building an optimal network around itself. When an attack or disturbance takes place on any node or edge, the proposed selecting method can help us learn the propagation mechanism and find which node is subject to the attack or disturbance. At the same time, the new selecting method can also help us build a more robust traffic network by the iterative calculating of an artificial slime mold.

IV. ALGORITHM DESIGN OF ASM

A. STEP 1: INITIALIZATION STAGE

Initialization is the first step of the ASM algorithm. It is to initialize the iterative counter of our algorithm, the parameters of the traffic network $G = (N, E)$ to be solved, including the positions of all n nodes for selecting and the distance matrix $D = \{|e_{ij}||i, j \in N\}$ between all nodes. At the start time $t = 0$, the node sets which m myxamoebas have passed through are set to be empty, there is $L = [L_k]_m = [\{\}]_m$, and the length of every myxamoeba is $D_k = 0$. The nutrient concentration on each edge is set as $\tau_{ij}(0) = 0$, meaning no nutrient on transporting route at the start.

Set the affinity parameter $\xi_i \geqslant 0$ of the food source node i, and the affinity matrix is initialized as $\xi = [\xi_i]_n$. The more nutrient the traffic node i includes, the larger the ξ_i is; on the contrary, the less nutrient the traffic node i includes, the smaller the ξ_i is.

Set the consumption parameter $\zeta_{ij} \geqslant 0$ on the edge e_{ij} and the matrix of nutrient consumption rate is initialized to be $\zeta = [\zeta_{ij}]_{n\times n}$. The greater the nutrient consumption on the traffic route v_{ij} is, the bigger the ζ_{ij} is; otherwise, the less the nutrient consumption on the traffic route v_{ij} is, the smaller the ζ_{ij} is.

In this stage, the expanding speed and contracting speed of the myxamoebas are set to be v^+ and v^- respectively. There are $v^+ \geqslant v^-$ when the myxamoebas expand, and $v^+ < v^-$ when the myxamoebas contract.

B. STEP 2: FOOD SEARCHING STAGE

The food searching stage is the second step of the ASM algorithm where the myxamoeba will expand. Selecting the nodes to be ranked as the iteration counter, there are n iteration circulations. Set $v^+ \geqslant v^-$, the m myxamoebas will expand from the nodes to be ranked to other nodes, simulating the food searching behavior of a true slime mold. Moreover, the expansion operation of the plasmodium is constrained by the originated nodes and the capacity of nutrients.

At the time of t, when a node is selected by a myxamoeba into its route $L_k(t) = \{e_{12}^k(t), e_{23}^k(t), \ldots, e_{l-1,l}^k(t), \ldots, e_{c_k-1,c_k}^k(t)\}$, the selection parameter will be $\mu_{ij}^k(t) = 1$

meaning that the edge $e_{ij} \in E$ is selected into the route L_k; otherwise, there is $\mu_{ij}^k(t) = 0$ meaning that the edge $e_{ij} \in E$ is not selected into the route L_k. The selecting matrix of myxamoeba L_k can be gotten as $\mu_k = [\mu_{ij}^k(t)]_{n\times n}$. When $\mu_{ij}^k(t) = 1$, the nutrient concentration on the edge $e_{ij} \in E$ of the myxamoeba k will be increased at a speed $\Delta_{ij}^k(t) > 0$ according to formula (5), meaning more nutrients flowing on the edge e_{ij}. Conversely, no matter $\mu_{ij}^k(t) = 1$ or $\mu_{ij}^k(t) = 0$, the nutrients will be continuously consumed at a speed $\sigma_{ij}^k(t)$ by formula (6), often $\sigma_{ij}^k(t) \leq \Delta_{ij}^k(t)$. The nutrient concentration on each edge will be updated.

$$\tau_{ij}(t) = \tau_{ij}(t-1) + \sum_{k=1}^{m} \mu_{ij}^k(t)\Delta_{ij}^k(t) - \sum_{k=1}^{m} \sigma_{ij}^k(t) \quad (26)$$

As we can see, m myxamoebas will connect all nodes, and the nutrient concentration on the shared transporting routes by multiple myxamoebas will be strengthened by $\Delta_{ij}^k(t) - \sigma_{ij}^k(t) \geq 0$. Different myxamoebas can learn the node information and route experience of e_{ij}, by which the connections between these shared routes will be enhanced. Generally, $\Delta_{ij}^k(t)$ and $\sigma_{ij}^k(t)$ are positively correlated with the affinity ξ_i of traffic nodes, and negatively correlated with the distance d_{ij} of edges. Hence, the richer the affinity ξ_i is and the shorter the distance d_{ij} on the edge e_{ij} is, the higher the nutrient concentration τ_{ij} is.

After the expansion operation of myxamoebas, the origin node to be ranked can connect all traffic nodes by an artificial slime mold, and the shortest searching route will be gotten to prepare the next feeding stage. Some indexes of the traffic network will be calculated, and different nodes can get different network topologies with different indexes according to formulas (15), (17), (19), and (22).

C. STEP 3: FEEDING STAGE

After finding all traffic nodes connecting the originated nodes to be ranked, the artificial slime mold will further search the optimal network topologies for each originated node and evaluate the solution to get an optimal traffic network for selecting. In the third step, each node to be ranked can be taken as a calculating iteration of an outer loop, and a calculation error can be predefined as an end condition for the feeding stage. The myxamoebas will start from the selected nodes to be ranked to connect all nodes and try to find an optimal topology. If the calculation is not finished, the myxamoeba $k(1 \leqslant k \leqslant m)$ begins to form a route from the original node, and all myxamoeba routes form a traffic network $L(t) = [L_k(t)]_m$. At the beginning of the feeding stage, the routes of myxamoeba $k(1 \leqslant k \leqslant m)$ will be set as an empty set $L(0) = [L_k(0)]_m = [\{\}]_m$.

The most important thing in this step is myxamoeba contraction, where the total number m of myxamoebas constitutes a calculating loop in the middle layer, trying to search for an optimal solution for each node to be ranked. According to the learning mechanism, the myxamoebas will consume the nutrients $\tau_{ij}(t)$ in the whole process and begin to form an optimal traffic network to connect all nodes to the original node. The nutrient concentration of myxamoebas will indicate the myxamoebas how to learn the experience of its own by formula (12) or the experience of neighbors by formula (13).

The nutrient concentrations on the optimal routes will be higher and higher, or vise versa. Then, the traffic network will continue to be optimized according to the learning mechanism. After several iterations of computing, each myxamoeba learns and contracts by the two probabilities $\alpha_{ij}^k(t) \in [0, 1]$ and $\beta_{ij}^k(t) \in [0, 1]$, and then form an optimal solution for each node to be ranked. The shortest distance and the objective function can be calculated by formulas (23) and (25).

D. STEP 4: SELECTING FOR OUTPUT

After a lot of iterative computing, the solutions produced by the learning mechanism will be evaluated for ranking and output. Different application cases can select different indexes as the evaluation function, such as degree index in formula (15), closeness index in formula (17), betweenness index in formula (19), and redundancy rate index in formula (22), or others. Those traffic topologies with low values of objective function to connect the origin nodes will be shifted out, and each node can select an optimal topology as its solution for selecting.

According to the learning experience and solution evaluation, the optimal route L^* of each myxamoeba will be gotten by formula (24) to connect all nodes with an optimal evaluation value. The finishing conditions will be judged whether the predefined error is fit or the number of iterations is reached. If not, the algorithm will return to repeat the computing of step 2 and step 3. Finally, all nodes will get an optimal traffic network originated from themselves and related indexes of their optimal topologies can also be gotten. Then, the node selecting can be output according to the indexes of their optimal traffic networks.

A parallel and probabilistic searching method is employed to solve the selecting problem of traffic network nodes preventing itself from falling into a local optimum. In every iterative computing, all myxamoebas will collaboratively get an optimized solution to connect all nodes around the node to be ranked. When the computing error between the latest iterations is less than a preset threshold, the optimal solution will be output to calculate its performance for node selecting. Or the algorithm will return to the previous steps to continue its calculating until the finishing conditions are met.

E. ALGORITHM FLOW

The algorithm flow to select traffic nodes by artificial slime mold composes of 4 main steps, including initialization, food searching, feeding, selecting for output. The myxamoebas can randomly expand around the nodes to be ranked and search all other traffic nodes in the food searching stage. Then the myxamoebas will continuously contract around the nodes to be ranked to optimize the traffic network in the feeding stage. The algorithm flow simulates the foraging behavior of a slime mold in figure 1 and produces an optimal solution for each node to be ranked which provides us an effective reference on the propagation capability or spreading ability of the selected node. The whole algorithm flow is shown in Figure 2.

Node Selecting Algorithm of ASM
Step 1: Initialization Stage
  set the initial traffic network $G = (N, E)$
  set the initial nutrient concentration $\tau_{ij}(0) = 0$
  set the learning probability $\alpha_{ij}^{k}(t)$ and $\beta_{ij}^{k}(t)$
Step 2: Food Searching Stage
  for selecting node i=1 to n
    calculate the food searching networks
  end for
Step 3: Feeding Stage
  for each myxamoeba k =1 to m
    learn routes with probability $\alpha_{ij}^{k}(t)$ and $\beta_{ij}^{k}(t)$
    update the nutrient concentration $\tau(t) = [\tau_{ij}(t)]_{n \times n}$
    contracting and optimizing traffic route $L_k(t)$
  end for
  evaluate the optimal traffic network L^*
Step 4: Selecting for Output
  for selecting node i=1 to n
    compare different topologies for node selecting
    print selecting results of traffic nodes
  end for
end procedure

FIGURE 2. The algorithm flow of the artificial slime mold.

The proposed selecting method is different from traditional selecting algorithms [1]–[8], [12]–[22] or other artificial intelligence algorithms [4], [19], [22]. First, it comprises the main components of a slime mold, such as external food sources, plasmodium, myxamoebas, and nutrients, etc., as shown in figure 1. Second, it can simulate the basic expansion, contraction, and learning operations of a true slime mold interacting with the outer environment. Third, the artificial slime mold can use multiple myxamoebas to implement parallel, probabilistic, and distributed computing to search traffic nodes and form optimal traffic networks. Fourth, the proposed myxamoebas can learn the experience from itself and its neighbors to form an optimal traffic network for node selecting. Fifth, it evaluates the performance of the optimal traffic networks originated from every node to rank these nodes by the propagation capability or spreading ability.

V. EXPERIMENT ANALYSIS

A. EXPERIMENT RESULTS

Here an experiment is designed to verify the proposed selecting algorithm, related data are selected according to the Mexico highways in [11]. In 2011, Adamatzky et al. used 19 oat flakes as 19 geographical locations of traffic nodes, then used several true slime molds to draw approximating Mexican highways [11]. Although the slime mold is so primitive that there is no brain or neuron in its body, the experimental results reveal that the true slime mole can successfully depict a map of Mexico in about 70 hours, as shown in figure 1. Mexico comprises 31 states and one government center in Federal District, and there is a fast route linking main highways from each city to Mexico City. People cost numerous years to construct such a highway system, as shown in figure 3, but a slime mold can draw an approximating Mexican highway map in dozens of hours.

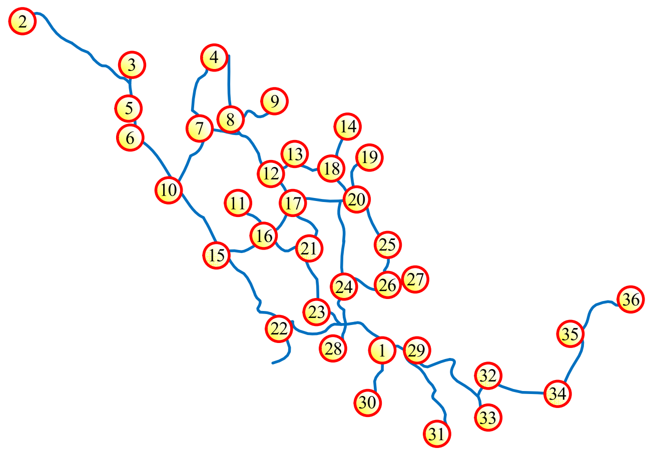

FIGURE 3. The experimental map of Mexico highway [11].

In our experiment, 36 cities in Mexico are selected to test the artificial slime mold, 17 nodes more than the node number 19 in [11], where the node No.1 is Mexico City, as shown in figure 3. To simplify the experimental analysis, some assumptions are made as follows.

i) Assuming that the positions and parameters of the Mexican highway system follow the data in reference [11];

ii) 16 nodes are randomly selected from the 36 cities of the Mexican highway system as an example for node selecting;

iii) For impartial comparison, the topology complexity is required to be the same in each comparison, such as the shortest route connecting all nodes without a return circuit.

iv) The number of myxamoebas is the same in each comparison, and all myxamoebas can learn environmental information by the self-learning possibility 0.3 and neighbor-learning possibility 0.4.

v) The end condition includes no more than 100 iterations and the trialed error less than 0.0001.

At first, 3 myxamoebas are selected to generate a Mexican highway system, and our artificial slime mold spends less time(in about 10 seconds) in getting every optimal solution similar to the results in [11], as shown in figures 4 (a)∼(l). Our results in figure 4 are almost the same as the experimental results in reference [11] (figure 1) and the true Mexican highway network (figure 3) by human of trial-and-error and reconstruction for decades. The experimental results in figure 4 present different results by 3 myxamoebas originated from 16 nodes to be ranked, and each node can generate an optimal traffic network with different shortest total distances. These may help us rank these nodes by the propagation capability

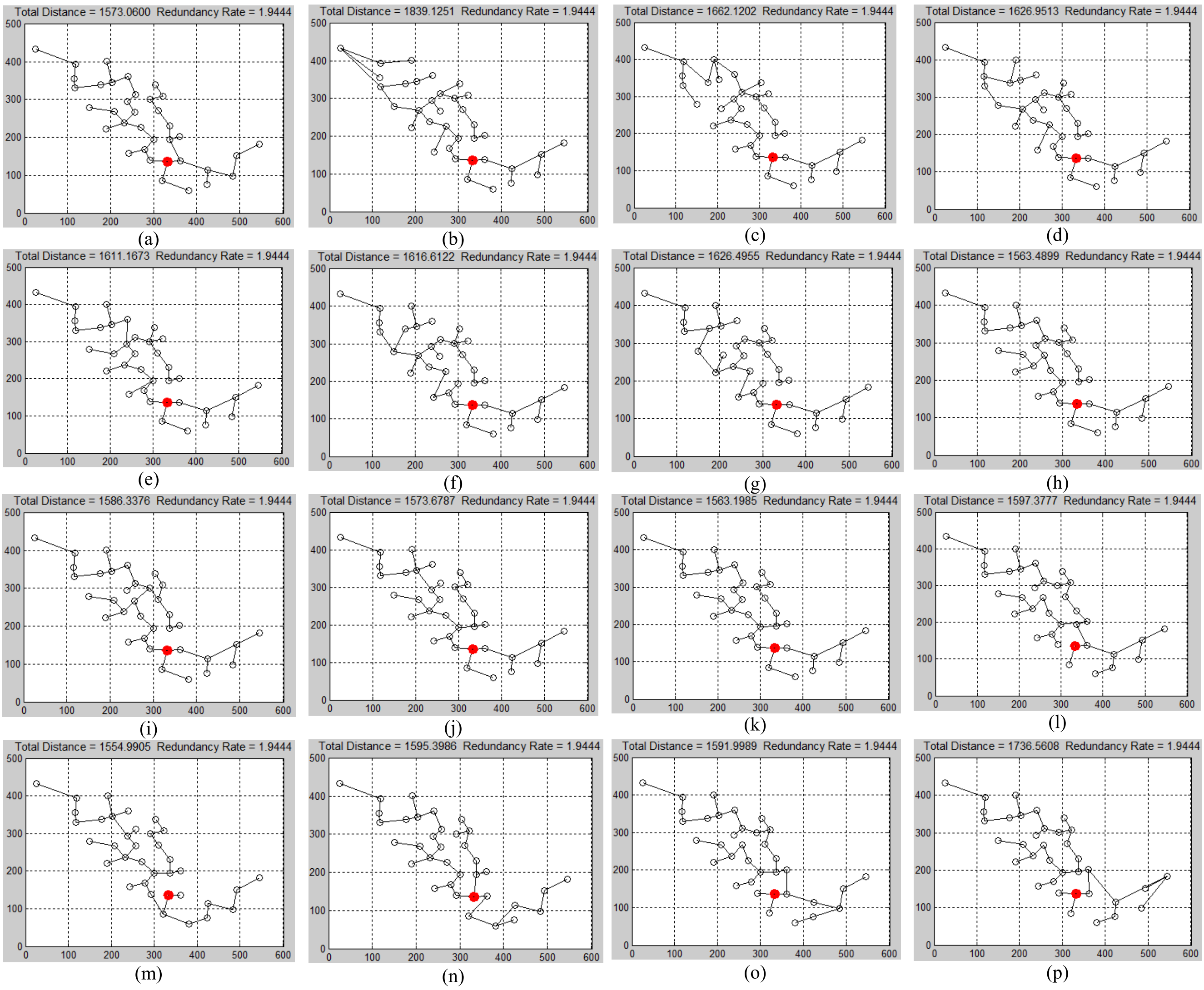

FIGURE 4. Simulated Mexican highway maps with 3 myxamoebas. (a) No.1; (b) No.2; (c) No.4; (d) No.5; (e) No.8; (f) No.10; (g) No.15; (h) No.17; (i) No.20; (j) No.23; (k) No.24; (l) No.27; (m) No.30; (n) No.31; (o) No.34; (p) No.36.

or spreading ability according to their generated topologies, and extend our mind to think about the learning mechanism of non-nervous living things.

Our simulation results of the Mexican highway network running on a personal computer are also based on the basic principle of the slime mold foraging behavior in references [9]–[11], [23]–[25]. Our results in figure 4 and the solutions of [11] in figure 1 are so similar that the effectiveness of the proposed artificial algorithm can be verified and our artificial slime mold can successfully simulate the foraging behavior of a true slime mold in [9]–[11], [23]–[25].

Now, the generated traffic networks can be used for node selecting. The largest point in the center of figure 4 is marked as No.1 of Mexico City, and the rests are other traffic nodes in the Mexican highway system. All the subfigures in figure 4 use 3 myxamoebas and get the same redundancy rate 1.9444 from formula (22) for fair comparison. Hence, these nodes will be ranked by their generated topologies with similar topology complexity.

The proposed artificial slime mold can feed itself around the selected node by its propagation capability or spreading ability, such as Mexico City in the center of the traffic map in figure 4(a), and can form different topology connections in figures 4 (a)∼(p) around these highway nodes to be ranked. However, the total length of the highway network is increasing from 1554.9905 to 1839.1251 in different topologies of figure 4. According to the total distance in figure 4, the most important nodes are in turn No.30 with total distance 1554.9905 (in figure 4 (m)), No.24 with total distance 1563.1985 (in figure 4 (k)), No.17 with total distance 1563.4889 (in figure 4 (h)), and No.1 with total distance 1573.0600 (in figure 4 (a)), verifying the node No.1 (Mexico City) is an important node in the highway system. Node No.30, No.24, and No.17 are also important nodes in the

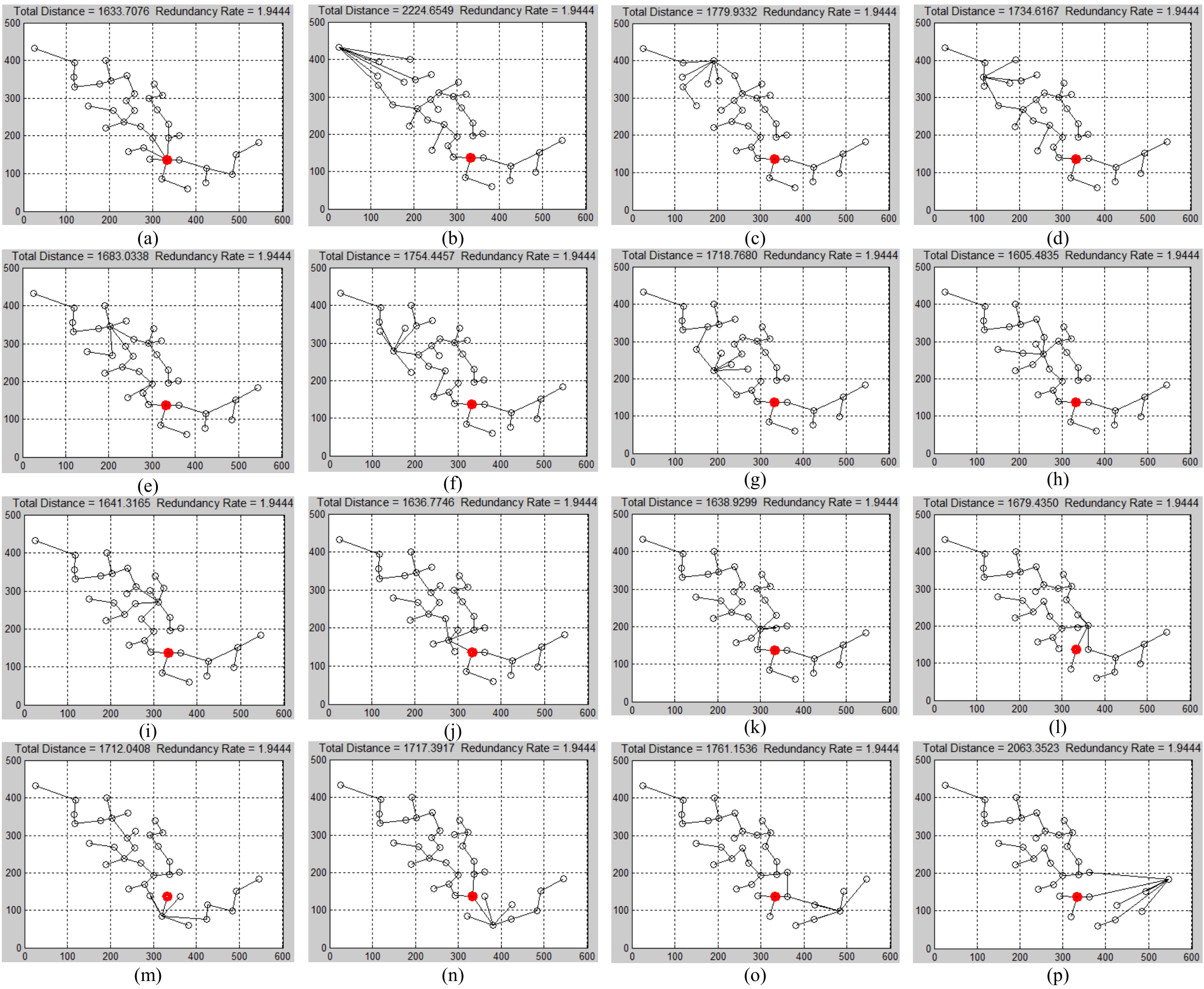

FIGURE 5. Simulated Mexican highway maps with 6 myxamoebas. (a) No.1; (b) No.2; (c) No.4; (d) No.5; (e) No.8; (f) No.10; (g) No.15; (h) No.17; (i) No.20; (j) No.23; (k) No.24; (l) No.27; (m) No.30; (n) No.31; (o) No.34; (p) No.36.

highway system by their propagation capabilities or spreading abilities, and they can be selected as the most important highway nodes from all traffic nodes according to the total distances of their generated topologies, i.e., No.30, No.24, No.17, No.1.

Then, more trial-error results with 6 myxamoebas are shown in figures 5 (a)∼(p), with different total distances and the same redundancy rate. Compared with the results in figure 4, the number of myxamoebs around the nodes to be ranked in figure 5 increases from 3 to 6, and more myxamoebas can help us find more routes with the same redundancy rate. Compared with the true slime mold in [9]–[11], [23]–[25], it is easier to adjust the experimental parameters in our artificial slime mold algorithm to get more solution results.

In figure 5, it is apparent that the traffic network topologies formed each time are slightly different because of the propagation capability or spreading ability of different nodes to be ranked, but the artificial system can find the optimal solution with the same redundancy rate 1.9444 from formula (22) after several computing iterations. The total length of the highway network is increasing from 1605.4835 to 2224.6549 in different topologies. According to the total distance in figure 5, the most important nodes are in turn No.17 with total distance 1605.4835 (in figure 5 (h)), No.1 with total distance 1633.7076 (in figure 5 (a)), No.23 with total distance 1636.7746 (in figure 5 (j)), and No.24 with total distance 1638.9299 (in figure 5 (k)), verifying the node No.1(Mexico City) is one of the most important nodes in the highway system. Node No.17, No.23, and No.24 are still important nodes in the highway system by their propagation capabilities or spreading abilities, they can be selected as the most important highway nodes by the total distances of their generated topologies, i.e., No.17, No.1, No.23, No.24.

Interestingly, the nodes No.17, No.1, and No.24 are selected in two cases(3 myxamoebas, and 6 myxamoebas), and the node No.17 is ahead of No. 1 in both experiments, which may mean the node No.17 is very important.

Summing up the results in figure 4 and figure 5, the proposed artificial slime mold is verified to be able to connect all the nodes in Mexican highway system after the continuous learning and optimization of the myxamoebas, and get similar results as [11] but with shorter calculating time(70 hours is cost to get a solution in reference [11]). According to the optimized traffic networks with the same redundancy rate, the artificial slime mold can help us rank these nodes by the propagation capability or spreading ability.

B. ANALYSIS AND DISCUSSION

The section of analysis and discussion considers the selecting comparisons with different myxamoebas, as shown in figure 6 and figure 7, where figure 6 shows the total distances of different network topologies originated from all 36 nodes with myxamoeba number $m = 3, 6$; figure 7 presents the selecting results of all 36 nodes with myxamoeba number $m = 3,6$. Compared with the results in figures 1, 3, 4, 5, 6 and 7, the proposed artificial slime mold can accurately simulate the foraging behavior of the real slime mold in [9]–[11], [23]–[25], and takes advantages over those results in figure 1 [11] in solving speed and accuracy. The artificial slime mold can help us build different Mexico highway networks with less cost, and the complex biological operations are all deleted from our method, such as the cultivation of slime molds, experimental design, professional biological operation steps, and data analysis for biologists in [9]–[11], [23]–[25].

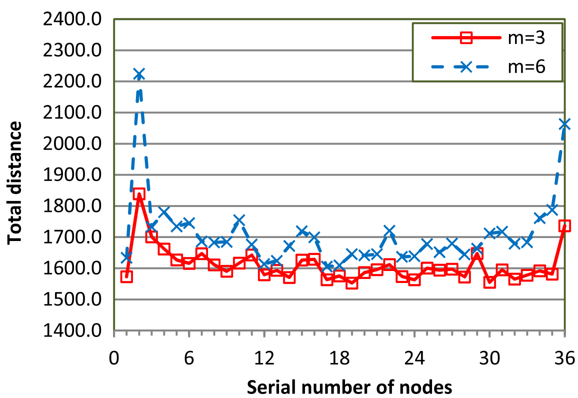

FIGURE 6. **Total distance with different myxamoebas.**

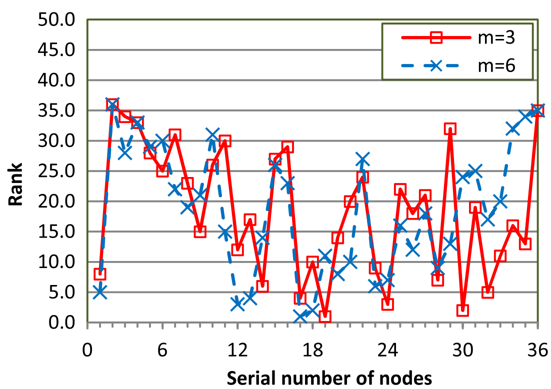

FIGURE 7. **Selecting results with different myxamoebas.**

Even compared with traditional learning algorithms, such as deep learning [3], [4], spectral learning [19], genetic algorithm(GA) [20], the proposed artificial slime mold will help us easily try more traffic topologies with better performance, and at the same time with lower computational errors and shorter solving period.

The most important thing, by the great advantages in topology exploration, the artificial slime mold can help us select important nodes by their propagation capabilities or spreading abilities according to their originated network topologies. Different indexes can be chosen as evaluation functions to meet different requirements, such as degree index in formula (15), closeness index in formula (17), betweenness index in formula (19), and redundancy rate index in formula (22), or others.

As we can see from figures 6 and 7, the proposed artificial slime mold can get consistent selecting results in the traffic networks with different myxamoeba numbers ($m = 3, 6$). The proposed selecting method is different from traditional selecting algorithms [1]–[8], [12]–[22], but our results will help us to build the most efficient traffic routes in case of traffic disturbances or great attacks. The selecting results are based on the network topologies generated by the ranked nodes themselves (figures 4, 5, 6, 7), and will provide more significant information about the importance of a node in a network topology. In our experiment, the importance of fringe nodes(No.2, No.3, No.4, No.6, No.10, No.15, No.22, No.36) is lower than that of central nodes (No.1, No.12, No.13, No.17, No.18, No.24). Among them, No.17 seems to be more important than No.1 with a better capability of topology propagation in both cases($m = 3, 6$), which can also be seen in figure 4(h) and figure 5(h). Network manager can use the tool to search the important nodes which are often neglected, or the attackers will select these important but neglected nodes as their first attacking objectives. The comparison of algorithm performance is shown in Table 1.

TABLE 1. **Comparison of algorithm performance.**

Sources	Number of nodes	Solving speed	Solving accuracy
[9]	36	26 hours	millimeter-scale
[11]	19	70 hours	millimeter-scale
[23]	7	96 hours	millimeter-scale
ASM	36	10 seconds	2^{64}(By computer)

Traditional selecting methods and optimization algorithms for traffic networks often cost people a lot of time in repetitive planning, construction, and reconstruction. However, the proposed artificial slime mold can help us easily try many different traffic topologies with the shortest total distances and lowest redundancy rates. Because of its advantages in topology exploration, it can also be used to solve the more complex TSP problems, computer networks or routing protocols, and multi-objective optimization problems.

VI. CONCLUSION

In this paper, a novel artificial slime mold is proposed to simulate the foraging behavior of a true slime mold and extends our minds about the learning mechanism of single-cell living things. Its biological behaviors are illustrated here, and it is employed to solve the node selecting problem of traffic network by the expansion and contraction mechanisms similar to a true slime mold without a brain or neutron. The mathematical model of artificial slime mold for node selecting is built, and is verified by an experiment of Mexico highways from [11]. The proposed artificial slime mold is parallel, probabilistic, iterative, and distributed and is fully different from the traditional artificial intelligence methods or machine learning algorithms. After continuously expanding and contracting of multiple myxamoebas, the proposed artificial slime mold will help us solve the node-selecting problem and give us an important reference to the propagation capability or spreading ability of every node.

However, the proposed method still has some shortcomings, i.e., the parallel computing on a computer is not completely the same as a true slime mold, and the amount of computing will increase with the number of computing nodes. In future directions, a computer biomimetic optimization method for multiple slime molds may be interesting, the optimization of a 3D network is also provocative, and an engineering application will also be built to put the proposed algorithm into practice. Furthermore, the proposed method may have wider application prospects in decision-supporting, social network, public opinion, mass emergency response, fire/flood/earthquake/disaster escaping, and so on.

ACKNOWLEDGMENT

The authors would like to thank the hard work of all anonymous reviewers to improve the quality of this submission.

REFERENCES

[1] X. Zhang and B. Chen, "Study on node importance evaluation of the high-speed passenger traffic complex network based on the structural hole theory," *Open Phys.*, vol. 15, no. 1, pp. 1–11, Mar. 2017.

[2] H. Cao, Y. Guo, Y. Hu, S. Wu, H. Zhu, and L. Yang, "Location aware and node ranking value-assisted embedding algorithm for one-stage embedding in multiple distributed virtual network embedding," *IEEE Access*, vol. 6, pp. 78425–78436, 2018.

[3] J. Cao, D. Jin, L. Yang, and J. Dang, "Incorporating network structure with node contents for community detection on large networks using deep learning," *Neurocomputing*, vol. 297, pp. 71–81, Jul. 2018.

[4] Y.-A. Lai, C.-C. Hsu, W.-H. Chen, M.-Y. Yeh, and S.-D. Lin, "DeepRank: Improving unsupervised node ranking via link discovery," *Data Mining Knowl. Discovery*, vol. 33, no. 2, pp. 474–498, Mar. 2019.

[5] F. Desprez and B. Tourancheau, "Basic routines for the rank-2K update: 2D torus vs reconfigurable network," *Parallel Comput.*, vol. 21, no. 3, pp. 353–372, Mar. 1995.

[6] B. Bauer, F. Jordán, and J. Podani, "Node centrality indices in food Webs: Rank orders versus distributions," *Ecolog. Complex.*, vol. 7, no. 4, pp. 471–477, Dec. 2010.

[7] X. Cheng, S. Su, Z. Zhang, H. Wang, F. Yang, Y. Luo, and J. Wang, "Virtual network embedding through topology-aware node ranking," *SIGCOMM Comput. Commun. Rev.*, vol. 41, no. 2, p. 38, Apr. 2011.

[8] A. Le, J. Loo, A. Lasebae, A. Vinel, Y. Chen, and M. Chai, "The impact of rank attack on network topology of routing protocol for low-power and lossy networks," *IEEE Sensors J.*, vol. 13, no. 10, pp. 3685–3692, Oct. 2013.

[9] A. Tero, S. Takagi, T. Saigusa, K. Ito, D. P. Bebber, M. D. Fricker, K. Yumiki, R. Kobayashi, and T. Nakagaki, "Rules for biologically inspired adaptive network design," *Science*, vol. 327, no. 5964, pp. 439–442, Jan. 2010.

[10] A. Adamatzky and J. Jones, "Road planning with slime mould: If physarum built motorways it would route M6/M74 through newcastle," *Int. J. Bifurcation Chaos*, vol. 20, no. 10, pp. 3065–3084, Oct. 2010.

[11] A. Adamatzky, G. J. Martínez, S. V. Chapa-Vergara, R. Asomoza-Palacio, and C. R. Stephens, "Approximating Mexican highways with slime mould," *Natural Comput.*, vol. 10, no. 3, pp. 1195–1214, Sep. 2011.

[12] N. Ma and Y. Liu, "SuperedgeRank algorithm and its application in identifying opinion leader of online public opinion supernetwork," *Expert Syst. Appl.*, vol. 41, no. 4, pp. 1357–1368, Mar. 2014.

[13] Z. Wang, L. Dueñas-Osorio, and J. E. Padgett, "A new mutually reinforcing network node and link ranking algorithm," *Sci. Rep.*, vol. 5, Oct. 2015, Art. no. 15141.

[14] X.-L. Tong, J.-G. Liu, J.-P. Wang, Q. Guo, and J. Ni, "Ranking the spreading ability of nodes in network core," *Int. J. Mod. Phys. C*, vol. 26, no. 5, May 2015, Art. no. 1550059.

[15] C. Lin, G. Wu, C. W. Yu, and L. Yao, "Maximizing destructiveness of node capture attack in wireless sensor networks," *J. Supercomput.*, vol. 71, no. 8, pp. 3181–3212, Aug. 2015.

[16] A. Hassani, A. Bertrand, and M. Moonen, "GEVD-based low-rank approximation for distributed adaptive node-specific signal estimation in wireless sensor networks," *IEEE Trans. Signal Process.*, vol. 64, no. 10, pp. 2557–2572, May 2016.

[17] Y. Zhang, L. Gu, X. Liao, H. Jin, D. Zeng, and B. B. Zhou, "FRANK: A fast node ranking approach in large-scale networks," *IEEE Netw.*, vol. 31, no. 1, pp. 36–43, Jan. 2017.

[18] H. Cao, Y. Zhu, L. Yang, and G. Zheng, "A efficient mapping algorithm with novel node-ranking approach for embedding virtual networks," *IEEE Access*, vol. 5, pp. 22054–22066, 2017.

[19] S. Xu, P. Wang, C.-X. Zhang, and J. Lu, "Spectral learning algorithm reveals propagation capability of complex networks," *IEEE Trans. Cybern.*, vol. 49, no. 12, pp. 4253–4261, Dec. 2019.

[20] L. Fei, Q. Zhang, and Y. Deng, "Identifying influential nodes in complex networks based on the inverse-square law," *Phys. A, Stat. Mech. Appl.*, vol. 512, pp. 1044–1059, Dec. 2018.

[21] H. Mo and Y. Deng, "Identifying node importance based on evidence theory in complex networks," *Phys. A, Stat. Mech. Appl.*, vol. 529, Sep. 2019, Art. no. 121538.

[22] L. Cruz-Piris, M. A. Lopez-Carmona, and I. Marsa-Maestre, "Automated optimization of intersections using a genetic algorithm," *IEEE Access*, vol. 7, pp. 15452–15468, 2019.

[23] X. Zhang, Q. Wang, A. Adamatzky, T. S. F. Chan, S. Mahadevan, and Y. Deng, "An improved physarum polycephalum algorithm for the shortest path problem," *Sci. World J.*, vol. 2014, Mar. 2014, Art. no. 487069.

[24] B. Y. Chen, W. H. Lam, A. Sumalee, and H. Shao, "An efficient solution algorithm for solving multi-class reliability-based traffic assignment problem," *Math. Comput. Model.*, vol. 54, nos. 5–6, pp. 1428–1439, Sep. 2011.

[25] V. Evangelidis, J. Jones, N. Dourvas, and M.-A. Tsompanas, "Physarum machines imitating a Roman road network: The 3D approach," *Sci. Rep.*, vol. 7, Aug. 2017, Art. no. 7010.

ZHENGYING CAI received the bachelor's, master's, and Ph.D. degrees from the Huazhong University of Science and Technology, China. He is currently a Professor with the College of Computer and Information Technology, China Three Gorges University. He has published more than 80 academic articles in conferences and journals. He holds more than 40 patents. His research interests include artificial intelligence, cloud manufacturing, and quantum computing.

ZEPING XIONG received the bachelor's degree from Hankou University, China. He is currently pursuing the master's degree in computer technology with China Three Gorges University. His research interests include artificial intelligence and quantum communication.

KUNPENG WAN received the bachelor's degree from China Three Gorges University, where he is currently pursuing the master's degree with the College of Computer and Information Technology.

YAQI XU received the bachelor's degree from China Three Gorges University, where she is currently pursuing the master's degree with the College of Computer and Information Technology.

FAN XU received the bachelor's degree from China Three Gorges University, where he is currently pursuing the master's degree with the College of Computer and Information Technology.