Extended Physarum Solver for Capacity-Constraint Routing

Kohsuke Sonoda*, Hiroshi Katada†, Shigeru Shimamoto†, Taku Yamazaki* and Takumi Miyoshi*
*College of Systems Engineering and Science, Shibaura Institute of Technology, Saitama, Japan
Email: {bp20129, taku, miyoshi}@shibaura-it.ac.jp
†Department of Communications and Computer Engineering, Waseda University, Tokyo, Japan
Email: hkatada@suou.waseda.jp, shima@waseda.jp

***Abstract*—Physarum solver (PS), which is inspired by the foraging behavior of Physarum polycephalum, has been attracting attention as a metaheuristic for solving optimization problems such as finding the shortest path. However, it could not solve optimization problems such as routing with link capacity constraint due to the lack of the link capacity concept in the PS. To address this problem, this paper introduces an extended Physarum solver (EPS) that can recognize and maintain link bandwidth resources. The EPS makes minimal changes to the existing PS adaptive equations but enhances the PS to solve optimization problems involving capacity-constraint routing by introducing the concept of link capacity. Experiments with various traffic scenarios show that the EPS achieves the optimal routing and further extends the link capacity of the PS. It was also confirmed that the EPS is consistent with the existing research. This suggests that metaheuristics based on the foraging behavior of Physarum polycephalum can be promising solutions to optimization problems in communication networks.**

***Index Terms*—routing, engineering neo-biomimetics, slime mold, Physarum solver, optimization problem, metaheuristic**

I. INTRODUCTION

In recent years, metaheuristics inspired by natural phenomena have been studied as a method for solving complex optimization problems [1]–[5]. In particular, metaheuristics based on the foraging behavior of Physarum polycephalum (hereafter referred to as a slime mold), a type of deformable fungus, have recently been attracting much attention. Slime molds are large, multinucleate, and unicellular organisms that, when they find food, take it in by extending their tubes and moving their protoplasm to cover the food. Nakagaki et al. observed the behavior of slime molds by placing two foods and showed that slime molds connect the foods by the shortest pathway [6]. Building upon this, Tero et al. mathematically modeled the path construction of slime molds and proposed the Physarum solver (PS) [7].

The PS has well solved several types of optimal problems, including the pathfinding problem [8], [9]. However, the PS does not support the concept of link capacity in the graph theory and cannot solve capacity-constraint optimization problems, e.g., a routing problem with link capacity constraint. Although some previous studies have attempted to adapt the notion of capacity to the PS [10], [11], they encountered challenges in terms of complexity due to additional algorithms, delayed speed of convergence, and limited applicability in optimization problems.

In this paper, we propose an extended Physarum solver (EPS) to address the above-mentioned issues. The EPS introduces the concept of graph-theoretic link capacity into the PS with minimal changes to the basic adaptive formulae of the PS. Various use cases envisaged show that the capacity-constraint routing problem can be solved by the EPS. The EPS significantly broadens the scope of the application of the PS, making it possible to apply it to practical optimization problems such as network routing. It is also expected to make it easier to extend the concept of link capacity in previous studies. In this paper, the applicability and convergence of EPS are verified by solving assumed routing problems in different scenarios of traffic patterns.

The contributions of this study can be summarized in the following three points. Firstly, we propose an EPS, which extends the concept of link capacity to the conventional PS. The EPS enables to solve capacity-constrained optimization problems while the conventional PS cannot. Secondly, by examining the EPS in various network scenarios, we can conclude that the EPS can determine the optimal routing path while accounting for link capacity constraints. The results indicate that the EPS is useful for solving optimization problems in real-world communication networks. Finally, we show that the EPS is consistent with previous research, indicating that the EPS successfully extends the PS adaptation formulae with minimal modifications while inheriting previous research. This facilitates the extension of the framework of existing research. These contributions suggest that the EPS is an important method for communication network optimization.

II. RELATED WORK

A. Physarum Solver

PS is a metaheuristic inspired by the foraging behavior of slime molds [7]. Slime molds construct pathways according to the arrangement of food in the maze. This is illustrated in Fig. 1. Fig. 1 (a) shows the initial state with food sources (FS) at two locations, in which all pathways including paths α_1 and α_2 that are the routes between the center point and the upper-right food source are filled by the slime mold. Fig. 1 (b) shows the changes in the maze after a certain time from state (a). It can be seen that the slime molds are retreating from the pathways that do not reach the food source. Fig. 1 (c) shows a further time after state (b), where the slime molds

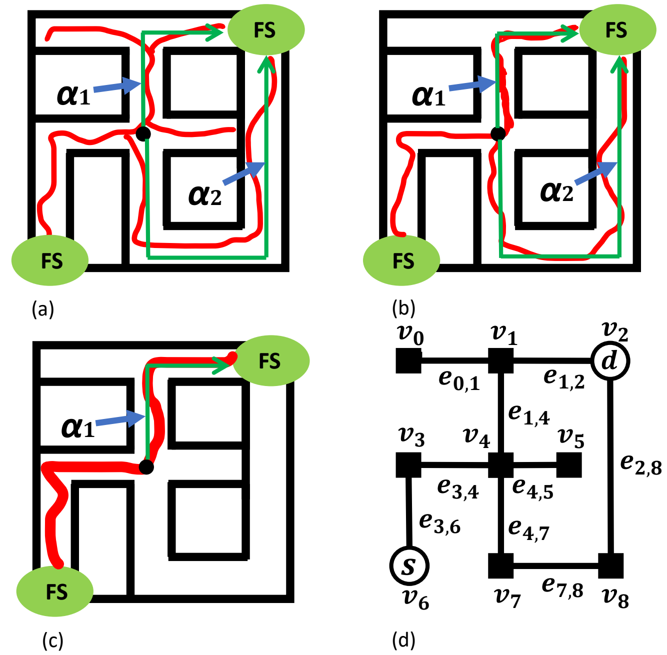

Fig. 1. Figures (a) – (c) represent slime molds solving a maze: (a) initial state, (b) intermediate state, and (c) final state. Figure (d) is graph-theoretic representation of a maze.

have formed the shortest path between the two food sources via α_1. Fig. 1 (d) shows the maze applied to graph theory. The intersections and dead ends in the maze are designated as node $v_i \in \mathbb{V}$, and the slime tube between the nodes v_i and v_j is denoted as link $e_{ij} \in \mathbb{E}$. One of the feeds is designated as the source node s, and the other as the destination node d.

The following is a mathematical model of how a pathway is constructed by slime mold between two nodes. Let $Q_{ij}(t)$ be the flow rate of link e_{ij} between nodes v_i and v_j at time t. This flow rate is based on the Poiseuille flow and expressed as follows:

$$Q_{ij}(t) = \frac{\pi(a_{ij}(t))^4}{8\kappa}\frac{p_i(t) - p_j(t)}{L_{ij}}, \tag{1}$$

where $a_{ij}(t)$ is the radius of the tube at time t, κ is the viscosity coefficient of sol, $p_i(t)$ is the pressure at the contact point on node v_i at time t, and L_{ij} is the length of the tube between nodes v_i and v_j. Eq. (1) can be changed to the following equation using the conductivity $D_{ij}(t) = \pi(a_{ij}(t))^4/8\kappa$,

$$Q_{ij}(t) = \frac{D_{ij}(t)}{L_{ij}}(p_i(t) - p_j(t)), \tag{2}$$

assuming $D_{ij}(t)$ is closely related to the thickness of the tube. The source node s and the destination node d represent the inflow and outflow points in the entire network, while $Q_{ij}(t)$ balances the inflow at the other nodes. Summarizing the equations, $Q_{ij}(t)$ can be expressed as follows:

$$\sum_i Q_{ij}(t) = \begin{cases} -I_0 & (j = s) \\ I_0 & (j = d) \\ 0 & (otherwise) \end{cases}. \tag{3}$$

Note that I_0 is the amount of flow from the source node to the destination node and given as a constant. Experimental observations show that the tube conductivity $D_{ij}(t)$ tends to increase or decrease with time according to the flow rate through the tube. The following equation is a mathematical expression of this adaptive property of slime molds with respect to time:

$$\frac{d}{dt}D_{ij}(t) = f(|Q_{ij}(t)|) - rD_{ij}(t). \tag{4}$$

An increasing function satisfying $f(0) = 0$ is usually used for $f(|Q_{ij}(t)|)$, and the convergence of the final path changes depending on the function used. For example, if $f(|Q_{ij}(t)|) = |Q_{ij}(t)|$, the path converges to the shortest path, and if $f(|Q_{ij}(t)|) = |Q_{ij}(t)|^{\mu}/(1 + |Q_{ij}(t)|^{\mu})$, $\mu > 1$, it converges to multiple paths. r is a parameter that controls the convergence speed of the tubes and $r = 1$ is commonly used. Discretizing Eq. (4) to solve it yields the following equation,

$$\frac{D_{ij}(t + \Delta t) - D_{ij}(t)}{\Delta t} = f(|Q_{ij}(t)|) - rD_{ij}(t), \tag{5}$$

where $0 < \Delta t < 1$ denotes the time interval. This equation can be expressed as

$$D_{ij}(t + \Delta t) = D_{ij}(t) + \Delta t\{f(|Q_{ij}(t)|) - rD_{ij}(t)\}. \tag{6}$$

By calculating Eqs. (2), (3), and (6), $Q_{ij}(t)$ is tentatively obtained, and by repeated calculations, the value of Q_{ij} converges, and then the path is finally determined. The PS is described as follows:

1) Calculate the pressure loss $p_i - p_j$ generated in each tube, using Eq. (2) with the old flow rate Q_{ij}, and Eq. (3).
2) Substitute the calculated pressure loss $p_i - p_j$ occurring in the tube into Eq. (2), and calculate the new flow rate Q_{ij}.
3) Substitute the derived new flow rate Q_{ij} into Eq. (6), and update the tube conductivity D_{ij}.
4) Feedback the tube conductivity D_{ij} to Eq. (2).
5) Repeat the above procedures until Q_{ij} converges.

B. Previous Study about Physarum Solver

There has been a wide range of research on PS, including studies on speeding up solution convergence [12], [13] and solving real-world problems and optimization problems [14], [15]. However, these previous studies on PS have focused on its application to path-finding problems. The reason is that while PS is a good adaptive metaheuristic for solving routing problems, it does not include the concept of link capacity in graph theory at the same time. Therefore, flow network optimization problems such as capacity-constraint routing could not be solved. Wang et al. incorporated the concept of capacity and solved the maximum flow problem, which is one of the flow network problems [10]. This method introduced the concept of energy when solving the problem. This concept, however, complicates the PS calculations and may even have a negative impact on convergence. On the other hand, Huang et al. introduced the concept of capacity without adding spare nodes and solved flow network problems such as the maximum flow problem [11]. However, this approach limits the situations in which the concept of capacity can be incorporated into the PS, and the scope of application of the PS is restricted to the conditions of a specific formula. This drawback is not in line with the idea of a metaheuristic that is not concerned with a specific problem.

C. Measurement of Convergence

In this paper, the relative gap (RGAP) is used to measure convergence:

$$\mathrm{RGAP} = \frac{|f(O) - f(P)|}{|f(O)|}, \tag{7}$$

where $f(O)$ and $f(P)$ denote the value of the theoretical objective function of the optimization and that of the solution obtained by the proposed algorithm, respectively. RGAP is taken as an absolute value to indicate the difference between the proposed and the optimal solutions. The closer Eq. (7) is to 0, the closer the proposed solution is to the optimal solution and the more the convergence is obtained.

III. Proposed Method

This section describes our proposed method, EPS, which extends the capacity to the traditional PS framework and can be used to solve optimization problems such as capacity-constraint routing problems. The proposed method introduces new elements to the conventional PS. Meanwhile, the modification should be minimized to maintain the original characteristics of the PS. In the EPS, therefore, we keep to use Eqs. (2) and (3) and modify only Eq. (6) as follows:

$$\begin{aligned} D_{ij}(t+\Delta t) &= D_{ij}(t) \\ &\quad + y_{ij}(t)\Delta t\{f(|Q_{ij}(t)|) - rD_{ij}(t)\}, \end{aligned} \tag{8}$$

$$y_{ij}(t) = \tanh(C_{ij} - |Q_{ij}(t)|). \tag{9}$$

The calculation procedure for the EPS is the same as for the conventional PS, where Eqs. (2), (3), and (8) are computed iteratively. A particularly important element of the proposed method is C_{ij}, which represents the capacity of each link, added to Eq. (9). In the conventional PS, the flow through each link can exceed the capacity of the link C_{ij} because this factor is not taken into account. In the proposed method, the concept of link capacity is introduced into Eq. (6), which models the adaptive characteristics of slime molds, so that the flow Q_{ij} flowing through each link between a node pair v_i and v_j is controlled to be not more than its capacity C_{ij}, which is reflected in Eq. (9). Specifically, y_{ij} introduces the hyperbolic tangent function. As a result, the EPS considers the current flow rate Q_{ij} and capacity C_{ij} between nodes v_i and v_j and behaves as follows:

- If Q_{ij} is less than C_{ij}: y_{ij} functions like a normal PS and flow is not restricted.
- If Q_{ij} exceeds C_{ij}: y_{ij} works to shrink the conductivity D_{ij} to decrease flow and make Q_{ij} converge to C_{ij}.
- If Q_{ij} reaches or approaches C_{ij}: y_{ij} restricts the change in D_{ij} and adjusts the flow according to the difference $C_{ij} - |Q_{ij}|$ so that it is stable and within the capacity.

The function y_{ij} acts as a regulating valve to prevent the slime mold protoplasm from clogging the pathways. With these improvements, the EPS extends the applicability of the PS to practical flow network optimization problems, in particular network resource-aware routing. As the proposed method does not significantly modify the adaptation formulae of the PS, it is so highly scalable as to be applied to both flow network and routing problems, and also can be integrated into the conventional PS research without difficulty.

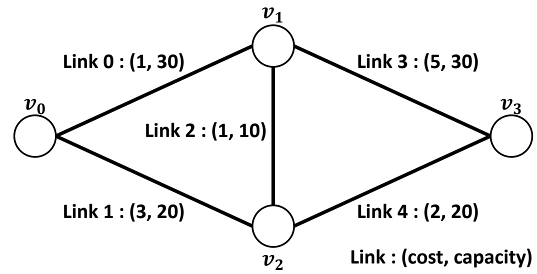

Fig. 2. Simulation topology.

IV. Performance Evaluation

In this section, the performance of the proposed EPS is evaluated. For the evaluation, three different network scenarios are considered, and the EPS is applied to verify its effectiveness and convergence. The routers deployed in the network are considered as nodes and the link capacity extended by the EPS is regarded as the maximum bandwidth between routers. For the convergence, the EPS is considered to be converged when the RGAP value expressed by Eq. (7) is less than 1×10^{-5}. The simulation software was implemented using C++.

1) Scenario 1: Single source – single destination with small amount of traffic data: In this scenario, we evaluate a case in which a small amount of traffic is transferred from a single source node to a single destination node using the EPS. The small amount here refers to the condition that the offered traffic does not exceed the maximum capacity of any link on the path. The experiment utilizes a topology shown in Fig. 2 and transfers 5 Mbps of data from node v_0 to node v_3. For the function in Eq. (8), we use the one commonly used to find the shortest path for the conventional PS, i.e., $f(|Q_{ij}(t)|) = |Q_{ij}(t)|$, and $\Delta t = 0.01$. While we assume the undirected weighted graph shown in Fig. 2 and do not consider the link direction theoretically, the calculation procedure treats flows as signed values. The direction of link 2 between nodes v_1 and v_2 is assumed to be positive from node v_1 to node v_2. Dijkstra's algorithm is used in Open Shortest Path First, which is a protocol for determining the path in the Internet routing protocols. The optimal solution obtained by Dijkstra's algorithm is link 0 $\rightarrow$ link 2 $\rightarrow$ link 4.

Fig. 3 shows the link bandwidth assigned by the EPS. Fig. 3 shows that, before the number of calculation iterations reaches

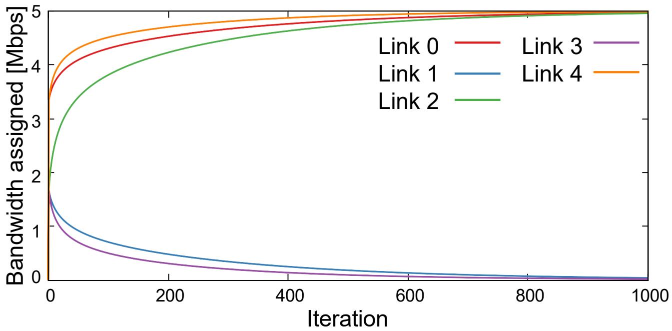

Fig. 3. The results of the EPS in scenario 1: single source - single destination with small amount of traffic.

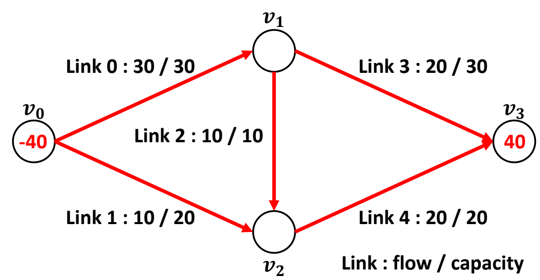

Fig. 4. The optimal solution of NetworkX in scenario 2: single source - single destination with large amount of traffic.

1000, the assigned bandwidth on link 1 and link 5 drops to 0 Mbps, while that on link 0, link 2, and link 4 reaches 5 Mbps. In other words, the EPS found the forwarding path link 0 $\rightarrow$ link 2 $\rightarrow$ link 4. The RGAP was 9.99983×10^{-6} at the 2557th calculation, indicating that the EPS has converged at this point. Comparing the solution of the EPS with that of Dijkstra's algorithm, the EPS can precisely find the shortest path to forward traffic. While the previous study [8] showed that the PS can solve the shortest path problem, this result suggests that the EPS can provide a solution that is consistent with the previous study while considering the concept of link capacity. The computational complexity is $O(n^3)$ as in the original PS, but the application of the previous studies on improving the convergence speed [12], [13] to the EPS suggests the possibility of speeding up the EPS.

2) Scenario 2: Single source – single destination with large amount of traffic data: In this scenario, we evaluate a case in which a large amount of traffic is transferred from a single source node to a single destination node using the EPS. The large amount here refers to the condition that the offered traffic exceeds the maximum capacity of a particular link on the path: multiple paths are thus required to transfer all the data considering the link cost that should be minimized. As in scenario 1, the topology in Fig. 2 is used for the experiments, and 40 Mbps data are transferred from node v_0 to node v_3. To validate the calculation results from the PS and EPS, we utilized NetworkX [16], a Python library that can solve the optimization problem of sending flows with minimum cost in a directed graph. The solution obtained by NetworkX is shown in Fig. 4.

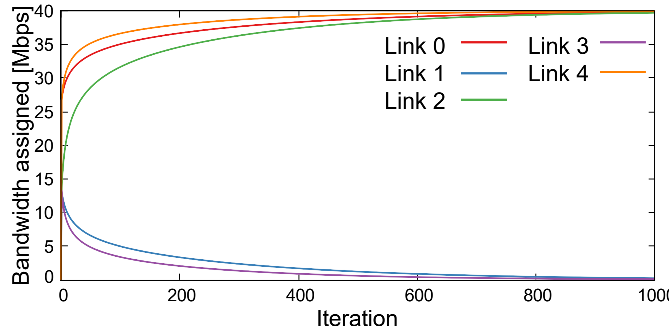

Fig. 5. The results of the PS in scenario 2: single source - single destination with large amount of traffic.

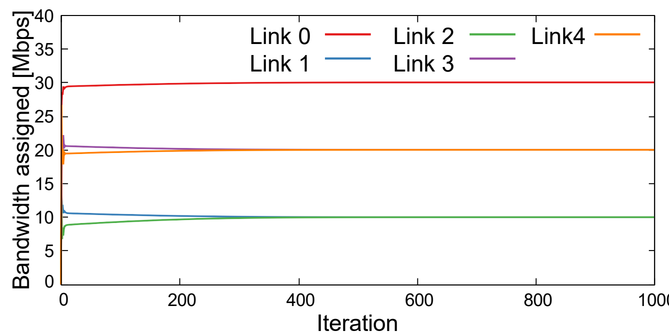

Fig. 6. The results of the EPS in scenario 2: single source - single destination with large amount of traffic.

First, we evaluate the results of solving the problem with the conventional PS, as shown in Fig. 5. The bandwidth assigned on link 0, link 2, and link 4 converges to 40 Mbps, which significantly exceeds the link capacity, 30 Mbps, 10 Mbps, and 20 Mbps, respectively. These results mean that the conventional PS cannot precisely solve capacity-constraint routing problems. Next, the solution solved by EPS is evaluated. The function in Eq. (8) used to find the solution is the same as in scenario 1, i.e., $f(|Q_{ij}(t)|) = |Q_{ij}(t)|$, and $\Delta t = 0.01$. Fig. 6 shows the convergences of the link bandwidth assigned by the EPS. Although the bandwidth on some links is not stable in the initial calculation stage, that converges stably by about 100 calculation iterations, and after about 400 iterations, the bandwidth on link 0 converges to 30 Mbps, that on link 1 and link 2 to 10 Mbps, and that on link 3 and link 4 to 20 Mbps. The RGAP reached 9.89628×10^{-6} at the 708th iteration, confirming that the EPS calculations have converged. Comparing Figs. 4 and 6, it can be seen that the solution of the EPS is the same as that of NetworkX. Therefore, it can be confirmed that the EPS can successfully find the optimal solution to transfer the amount of data to the destination node with the minimum cost.

3) Scenario 3: Multiple sources – single destination with large amount of traffic data: In this scenario, we evaluate a case where a large amount of traffic is transferred from multiple source nodes to a single destination node using the

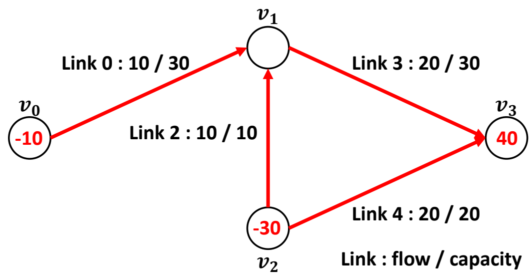

Fig. 7. The optimal solution of NetworkX in scenario 3: multiple sources - single destination with large amount of traffic.

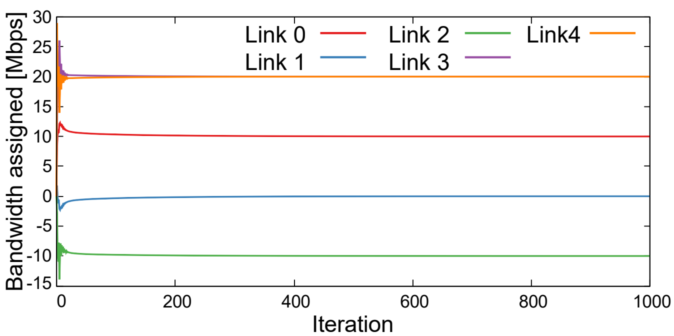

Fig. 8. The results of the EPS in scenario 3: multiple sources - single destination with large amount of traffic.

EPS. As in scenario 1, the topology shown in Fig. 2 is used for the experiment. Two traffic flows are assumed: 10 Mbps traffic from node v_0 and 30 Mbps traffic from node v_2 flowing to node v_3. The function in Eq. (8) used to find the solution is the same as in scenarios 1 and 2, and the direction of link 2 is positive from node v_1 to node v_2. NetworkX was also utilized to obtain the optimal solution, which is shown in Fig. 7.

Fig. 8 shows the results obtained by the EPS. We can see that the assigned bandwidth on link 3 and link 4 is 20 Mbps, that on link 0 is 10 Mbps, that on link 1 is 0 Mbps, and that on link 2 is -10 Mbps. Since the direction of link 2 is considered positive from node v_1 to node v_2, this result indicates that 10 Mbps flows from node v_2 to node v_1 is assigned. Unlike scenario 2, no bandwidth was assigned on link 1. This is because the EPS has appropriately avoided selecting a costly link. The RGAP converged to 9.95641×10^{-6} at the 1076th iteration. The assigned bandwidth on each link is completely equal to the solution using the Python library NetworkX, which indicates that slime mold can solve this problem appropriately.

The above experiments suggest that the proposed EPS can solve optimization problems such as capacity-constraint routing, and that the EPS can be applied directly to the previous studies.

V. Conclusion

This paper proposed an extended Physarum solver to appropriately solve capacity-constraint network routing problems. We introduced the concept of link capacity into the conventional PS, which did not take this factor into account. By evaluating the proposed EPS in three typical network scenarios, we can conclude that the EPS successfully extends the PS adaptation formulae with minimal modifications and precisely solves capacity-constraint routing problems. Furthermore, it was shown that the EPS can be applied in a way that is consistent with existing research.

This study is a step forward in showing that a metaheuristic based on the foraging behavior of Physarum polycephalum is a promising solution for capacity-constraint optimization problems. Future work includes investigating the relative performance and benefits of the EPS through comparisons with algorithms already in use in the real world. In addition, conducting intricate simulations is necessary to adapt the EPS to very large or dense networks. Measures to deal with the resulting increase in computational complexity also need to be addressed.

References

[1] A. Darwish, "Bio-inspired computing: algorithms review, deep analysis, and the scope of applications," Future Comput. Inform. J., vol. 3, no. 2, pp. 231–246, Jun. 2018.
[2] M. Dorigo, M. Birattari, and T. Stutzle, "Ant colony optimization," IEEE Comput. Intell. Mag., vol. 1, no. 4, pp. 28–39, Nov. 2006.
[3] X.-F. Liu, Z.-H. Zhan, J.D. Deng, Y. Li, T. Gu, and J. Zhang, "An energy efficient ant colony system for virtual machine placement in cloud computing," IEEE Trans. Evol. Comput., vol. 22, no. 1, pp. 113–128, Feb. 2018.
[4] S. Das and P.N. Suganthan, "Differential evolution: a survey of the state-of-the-art," IEEE Trans. Evol. Comput., vol. 15, no. 1, pp. 4–31, Feb. 2011.
[5] F. Aminifar, C. Lucas, A. Khodaei, and M. Fotuhi-Firuzabad, "Optimal placement of phasor measurement unit using immunity genetic algorithm," IEEE Trans. Power Del., vol. 24, no. 3, pp. 1014–1020, Jul. 2009.
[6] T. Nakagaki, H. Yamada, and Á. Tóth, "Maze-solving by an amoeboid organism," Nature, vol. 407, no. 6803, p. 470, Sep. 2000.
[7] A. Tero, R. Kobayashi, and T. Nakagaki, "A mathematical model for adaptive transport network in path finding by true slime mold," J. Theor. Biol., vol. 244, no. 4, pp. 553–564, Jul. 2007.
[8] H. Katada, T. Yamazaki, and T. Miyoshi, "Adaptive load balancing ad hoc routing scheme inspired by true slime mold," J. Telecommun. & Inform. Technol., no. 1, pp. 14–22, Mar. 2019.
[9] D. Straszak and N.K. Vishnoi, "On a natural dynamics for linear programming," ACM Conf. Innov. Theory Comput. Sci., p. 291, Jun. 2016.
[10] Z. Wang and D. Wei, "Solving the maximum flow problem by a modified adaptive amoeba algorithm," IAENG Int. J. Comput. Sci., vol. 45, no. 1, pp. 130–134, Feb. 2018.
[11] Y. Huang, D. Chu, Y. Deng, and K.H. Cheong, "The capacity constraint physarum solver," J. Comput. Sci., vol. 62, paper no. 101725, May 2022.
[12] C. Gao, X. Zhang, Z. Yue, and D. Wei, "An accelerated physarum solver for network optimization," IEEE Trans. Cybern., vol. 50, no. 2, pp. 765–776, Feb. 2020.
[13] Q. Wang, X. Lu, X. Zhang, Y. Deng, and C. Xiao, "An anticipation mechanism for the shortest path problem based on physarum polycephalum," Int. J. General Syst., vol. 44, no. 3, pp. 326–340, Mar. 2015.
[14] M.-A.I. Tsompanas, G.C. Sirakoulis, and A.I. Adamatzky, "Evolving transport networks with cellular automata models inspired by slime mold," IEEE Trans. Cybern., vol. 45, no. 9, pp. 1887–1899, Sep. 2015.
[15] L. Liu, Y. Song, H. Zhang, H. Ma, and A.V. Vasilakos, "Physarum optimization: a biology-inspired algorithm for the steiner tree problem in networks," IEEE Trans. Comput., vol. 64, no. 3, pp. 818–831, Mar. 2015.
[16] NetworkX Developers, "Networkx: network analysis in python," [Online]. Available: https://www.networkx.org/ [Access: 20 Sep. 2023].