A Biologically Inspired Network Design Model (Supplementary Information)

Xiaoge Zhang\textsuperscript{a,d}, Andrew Adamatzky\textsuperscript{b}, Felix T.S. Chan\textsuperscript{c}, Yong Deng\textsuperscript{a,d,\*}, Hai Yang\textsuperscript{e}, Xin-She Yang\textsuperscript{f}, Michail-Antisthenis I. Tsompanas\textsuperscript{g}, Georgios Ch. Sirakoulis\textsuperscript{g}, Sankaran Mahadevan\textsuperscript{d,\*}

\textsuperscript{a}School of Computer and Information Science, Southwest University, Chongqing 400715, China; \textsuperscript{b}Unconventional Computing Center, University of the West of England, Bristol BS16 1QY, UK; \textsuperscript{c}Department of Industrial and Systems Engineering, The Hong Kong Polytechnic University, Hung Hum, Kowloon, Hong Kong; \textsuperscript{d}School of Engineering, Vanderbilt University, Nashville, 37235, USA; \textsuperscript{e}Department of Civil and Environmental Engineering, The Hong Kong University of Science and Technology, Clear Water Bay, Kowloon, Hong Kong; \textsuperscript{f}School of Science and Technology, Middlesex University, London NW4 4BT, UK; \textsuperscript{g}Department of Electrical and Computer Engineering, Democritus University of Thrace, Xanthi 67100, Greece;

\*Correspondence and requests for materials should be addressed to Y. D. (prof.deng@hotmail.com) and S. M. (sankaran.mahadevan@vanderbilt.edu)
Algorithm 1 Physarum Polycephalum Algorithm for Network Design Problem \((F, L)\)

1: // \(N\) is a number of nodes, \(\delta\) has a threshold value.
2: // \(L\) is a \(n \times n\) matrix, \(L_{ij}\) denotes the length between node \(i\) and node \(j\)
3: // \(F\) is a \(n \times n\) matrix, \(F_{ij}\) represents a traffic flow between node \(i\) and node \(j\)
4: Construct a traffic flow \(F\) according to Eq. (8)

\[
F_{ij} = G \frac{M^{x_1} M^{x_2}}{D^{x_3}}
\]

5: \(\text{previous} \, D_{ij} \leftarrow 0 \) (\(\forall i, j = 1, 2, \ldots, N\))
6: \(\text{original} \, D_{ij} \leftarrow 0 \) (\(\forall i, j = 1, 2, \ldots, N\))
7: while \(\sum_{i=1}^{N} \sum_{j=1}^{N} |\text{current} \, D_{ij} - \text{original} \, D_{ij}| \geq \delta\) do
8: \(\text{previous} \, D_{ij} \leftarrow \text{current} \, D_{ij}\)
9: \(\text{original} \, D_{ij} \leftarrow \text{current} \, D_{ij}\)
10: \(\text{current} \, D_{ij} \leftarrow 0\)
11: for \(i = 1 : N\) do
12: for \(j = 1 : N\) do
13: \(\text{current} \, D_{ij} = \text{current} \, D_{ij} + P_A(i, j, L, \text{previous} \, D, F_{ij})\)
14: end for
15: end for
16: According to Eq. (11), the conductivity is normalised.

\[
D_k(i, j) = \frac{D_k(i, j)}{\max(D_k)} \quad (i = 1, \cdots, N; j = 1, \cdots, N)
\]

18: end while

Require: \(s, e, L, D, f\)

Ensure: \(D\)

19: function \text{Physarum Algorithm} \(PA(s, e, L, D, f)\)
20: // \(s\) is the starting node, \(e\) is the ending node, \(f\) is the traffic flow starting from node \(s\) to node \(e\)
21: // \(D\) is an initialised conductivity matrix
22: \(\text{iteration} = 0\)
23: while \(\text{iteration} < 1\) do
24: Calculate a pressure according to Eq. (9)
25: \(\sum_i D_{ij} (p_i - p_j) = \begin{cases} +f & \text{for } i = s, \\ -f & \text{for } i = e, \\ 0 & \text{otherwise.} \end{cases}\)
26: \(Q_{ij} \leftarrow D_{ij} \times (p_i - p_j) / L_{ij} \) // Using Eq. (1)
27: \(D_{ij} \leftarrow Q_{ij} + D_{ij} \) // Using Eq. (7)
28: \(\text{iteration} \leftarrow \text{iteration} + 1\)
29: end while
30: return \(D\)
31: end function