INTERFACE

royalsocietypublishing.org/journal/rsif

Research

Cite this article: Hsu S, Schaposnik LP. 2022 Cell fusion through slime mould network dynamics. *J. R. Soc. Interface* **19**: 20220054. https://doi.org/10.1098/rsif.2022.0054

Received: 18 January 2022
Accepted: 29 March 2022

Subject Category:
Life Sciences–Mathematics interface

Subject Areas:
biomathematics, bioinformatics

Keywords:
cell fusion, network dynamics, slime mould

Author for correspondence:
Laura P. Schaposnik
e-mail: schapos@uic.edu

Cell fusion through slime mould network dynamics

Sheryl Hsu[1] and Laura P. Schaposnik[2]

[1]Valley Christian High School, San Jose, CA, USA
[2]University of Illinois, Chicago, IL, USA

LPS, 0000-0003-1417-2201

Physarum polycephalum is a unicellular slime mould that has been intensely studied owing to its ability to solve mazes, find shortest paths, generate Steiner trees, share knowledge and remember past events and the implied applications to unconventional computing. The CELL model is a cellular automaton introduced in Gunji *et al*. (Gunji *et al*. 2008 *J. Theor. Biol*. **253**, 659–667 (doi:10.1016/j.jtbi.2008.04.017)) that models *Physarum*'s amoeboid motion, tentacle formation, maze solving and network creation. In the present paper, we extend the CELL model by spawning multiple CELLs, allowing us to understand the interactions between multiple cells and, in particular, their mobility, merge speed and cytoplasm mixing. We conclude the paper with some notes about applications of our work to modelling the rise of present-day civilization from the early nomadic humans and the spread of trends and information around the world. Our study of the interactions of this unicellular organism should further the understanding of how *P. polycephalum* communicates and shares information.

1. Introduction

Physarum polycephalum is a unicellular slime mould. The main portion of *Physarum's* life cycle is spent in the plasmodium stage, where *Physarum* forms a network of tubes. According to [1], small *Physarum* fragments (microplasmodia) fuse to form macroplasmodia, forming one giant plasmodium connected by tubes. These tubes contain protoplasm, which flows between these tubes through rhythmic contractions. As protoplasm flows, these tubes can grow or shrink, causing the plasmodium to reshape itself to find food or move [2]. Recent research [3] has shown that the growth and shrinkage of tubes is caused by a softening agent that serves as a way to encode memory.

Experiments have shown that *P. polycephalum* in this plasmodium state is able to solve mazes, find the shortest path, build high-quality networks between multiple points, adapt and respond to stimuli and remember past events [4–6]. *Physarum* has also been shown to be able to share information and fuse with other *Physarum* organisms [7]. Recently, several mathematical models inspired by *Physarum* have been developed to approach problems such as the shortest path or Steiner tree problems [8–11].

The behaviour of *P. polycephalum* can be studied, in particular, using cellular automata, as done in [8], through what the authors called *the CELL model*. In this model, a *Physarum* cell is modelled on a grid with cytoplasm and cytoskeleton. At each iteration, the cytoplasm and cytoskeleton is slightly reshaped with the introduction of an outside bubble. As shown in [8], this model can be used to simulate amoeboid motion with tentacle formation, maze solving, shortest path finding and network creation.

Inspired by [8], we take a novel approach which allows us to discern several interesting patterns. When considering cells, one expects to observe certain behaviours. In particular

1. the time it takes for two cells to merge should increase directly with the distance between them;

THE ROYAL SOCIETY PUBLISHING

2. it should be fastest and most likely for cells to merge with the cell closest to them;
3. the smaller a cell is, the more mobile it should be;
4. the smaller cells are, the better they should fuse in terms of cytoplasm mixing.

We shall begin this paper by introducing in §2 some of the different models used in the literature to study *P. polycephalum*, paying particular attention to the *flow-conductivity model* [10], the *cellular model* [8], the *multi-agent model* [9] and the *shuttle-streaming model* [11].

After introducing these models, we dedicate §3 to the main findings of our work. By considering cells at different distances apart and studying the number of iterations it took for them to merge, we could see that

— the relationship between distance and iterations is linear, as shown in §3.1.

By considering three cells and adjusting the distance between them, we measured which cells merge first and the number of iterations it took them to do so, allowing us to see that

— the two closest cells are most likely to merge together, followed by the next two closest cells, followed by the two furthest cells;
— the two closest cells take the least number of iterations to merge, followed by the next two closest cells, followed by the two furthest cells.

We also considered cells of different sizes and measured mobility using the average distance from the spawn point, the maximum distance from the spawn point and speed, leading to the following results:

— the relationship between average distance and size is exponentially decreasing;
— the relationship between maximum distance and size is exponentially decreasing;
— the relationship between speed and size is also exponentially decreasing.

Finally, by considering the fusion of two cells of varying sizes, we measured cytoplasm mixing using Lacy's mixing index [12], leading to the following findings:

— there is a decreasing exponential relationship between the mean mixing index and cell area;
— there is a decreasing logistical relationship between the maximum mixing index and cell size.

We conclude this paper by expanding on the analysis and applications of the above findings in §4. In particular, we explore the applications of our work to modelling the rise of present-day civilization from the early nomadic humans and the spread of trends and information around the world.

2. Modelling *Physarum polycephalum*

We shall dedicate this section to reviewing four models inspired by *P. polycephalum*, which we shall consider in the present paper when building a new model to study cell fusion between multiple cells. We shall begin by reviewing a flow-conductivity model in §2.1, a cellular model in §2.2 and a multi-agent model in §2.3. Finally, we shall conclude the section by introducing a shuttle-streaming model in §2.4. A good overview of the current literature on *P. polycephalum* can be found in [13].

2.1. Flow-conductivity model

Inspired by the idea that within *P. polycephalum* tubes of the plasmodium grow thicker as more protoplasm flows through that tube, the authors of [10] introduced a flow-conductivity model which we shall review below. In this setting, two further observations about *Physarum*'s behaviour come into play

— when two tubes connect to the same food source, the longer one usually disappears;
— open-ended tubes are likely to disappear.

In what follows, we shall describe this model following the ideas of [10]. Consider nodes N_1, $N_2 \ldots$, and let N_1 be the source node (start) and N_2 be the sink node (end). The tube/edge connecting nodes N_i and N_j is denoted as M_{ij}. Let Q_{ij} be the flux through M_{ij}, let p_i be the pressure at node i, let L_{ij} be the length of M_{ij} and let D_{ij} be the conductivity of M_{ij}. Then, we have

$$Q_{ij} = \frac{D_{ij}}{L_{ij}}(p_i - p_j). \tag{2.1}$$

By Kirchhoff's law,

$$\sum_i Q_{ij} = 0 \quad (j \neq 1, 2). \tag{2.2}$$

Since N_1 and N_2 are the start and end nodes, respectively,

$$\sum_i Q_{i1} + I_0 = 0 \tag{2.3}$$

and

$$\sum_i Q_{i2} - I_0 = 0. \tag{2.4}$$

The adaptation of the tube thickness can be modelled using the following equation:

$$\frac{\mathrm{d}}{\mathrm{d}t} D_{ij} = f(|Q_{ij}|) - rD_{ij}, \tag{2.5}$$

where $f(Q)$ must be an increasing function with $f(0) = 0$, and thus for the purposes of this model one may consider $f(Q) = \alpha|Q|$. Setting $p_2 = 0$, one can compute all the p_i's using equations (2.2)–(2.4) and therefore all Q_{ij}. The flow-conductivity model has been used to find Steiner trees [14], solve the travelling salesman problem [15] and design fault-tolerant networks [16]. It has been applied to creating a shortest path navigation system across the US interstate highway [10], designing railroad networks similar to the Tokyo railroad system [17], designing transportation networks with changing traffic distributions [18], identifying focal nodes for disease spread in epidemiological networks [19] and solving supply chain network design problems [20].

Example 2.1. In order to illustrate how the *flow-conductivity model* can be used, consider the graph in figure 1, where 1 is the source node and 6 is the sink node.

Starting with a flux of $I_0 = 10$ into node 1, and assuming all tubes start off with the same conductivity $D_{ij} = 2$, one

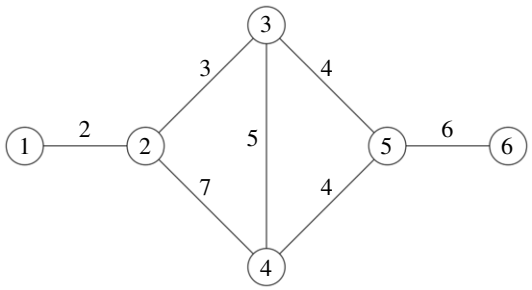

Figure 1. Node 1 is the source node and node 6 is the sink node. The number on each edge represents the length of that edge.

has the following system of equations:

$$\frac{2}{2}(p_2 - p_1) + 10 = 0,$$
$$\frac{2}{2}(p_1 - p_2) + \frac{2}{3}(p_3 - p_2) + \frac{2}{7}(p_4 - p_2) = 0,$$
$$\frac{2}{3}(p_2 - p_3) + \frac{2}{5}(p_4 - p_3) + \frac{2}{4}(p_5 - p_3) = 0,$$
$$\frac{2}{7}(p_2 - p_4) + \frac{2}{5}(p_3 - p_4) + \frac{2}{1}(p_5 - p_4) = 0,$$
$$\frac{2}{4}(p_3 - p_5) + \frac{2}{1}(p_4 - p_5) + \frac{2}{6}(p_6 - p_5) = 0$$

and $$\frac{2}{6}(p_5 - p_6) - 10 = 0.$$

Setting $p_6 = 0$ and solving this system leads to the values $p_1 = 57$; $p_2 = 47$; $p_3 = 38$; $p_4 = 33$; and $p_5 = 30$. Then, from equation (2.1) one has $Q_{12} = 10$; $Q_{23} = 6$; $Q_{24} = 4$; $Q_{34} = 2$; $Q_{35} = 4$; $Q_{45} = 6$; and $Q_{56} = 10$. Updating D_{ij} using $\Delta D_{ij} = \alpha |Q_{ij}| - r\, D_{ij}$, and taking $\alpha = 1$ and $r = 1$, one has that

$$\Delta D_{12} = 8; \quad \Delta D_{23} = 4; \quad \Delta D_{24} = 2$$

and

$$\Delta D_{34} = 0; \quad \Delta D_{35} = 2; \quad \Delta D_{45} = 4; \quad \Delta_{56} = 8.$$

The updated values of D_{ij} then are

$$D_{12} = 10; \quad D_{23} = 6; \quad D_{24} = 4$$

and

$$D_{34} = 2; \quad D_{35} = 4; \quad D_{45} = 6; \quad D_{56} = 10.$$

After repeating this process several times, a path appears when a lot of the D_{ij} are close to zero and then the rest are much greater.

2.2. Cellular model

In the cellular model as described in [8], or *CELL model*, a cell is a mass of cytoplasm surrounded by a membrane, which is placed on a lattice grid where every square has an assigned number/state. State 2 represents cytoskeleton, state 1 represents cytoplasm and state 0 represents squares that are not part of the CELL. In biological *Physarum* organisms, protoplasmic tubes consist of an outer layer of actin–myosin cytoskeleton with cytoplasm inside. The cytoskeleton periodically contracts, moving cytoskeleton and allowing *Physarum* to respond to a stimulus [2,21]. With this in mind, the CELL model is based on the following assumptions:

1. the membrane is the part of the cell where the cytoskeleton is concentrated and hardened;
2. when a cytoskeleton assembly is taken apart and the membrane softens, the cytoplasm is distributed to other areas of the cell;
3. cytoplasmic flow is accompanied by transportation of the cytoskeleton and distribution of cytoskeleton assemblies.

The CELL model has two phases: *development* and *foraging*. In the *development phase*, the cell grows, forming a diamond-shaped model with cytoskeleton around the edges and cytoplasm inside. In the *foraging stage*, the model 'eats' zero, which leads to a redistribution of cytoplasm and cytoskeleton/membrane. The algorithm is as follows:

1. Chose a site with state 2, the *stimulus point*.
2. Randomly choose a neighbour of the stimulus point in state 0 and replace the state of the stimulus point with the state of the chosen neighbour, so zero invades the cell. We call the zero a *bubble*.
3. Replace state 1 with 2 so all non-zero cells are 2. Set number of moves to zero.
4. Mark the site with the bubble.
5. Decide whether s sites of the bubble's neighbours are in state 0 or not. If yes, go to 8 otherwise go to 6.
6. Decide whether the number of moves exceeds threshold n. If yes, go to 8 otherwise go to 7.
7. Randomly choose one of the bubble's non-marked neighbours which is in state 2. Replace the state of the bubble with the state of the chosen neighbour. Add 1 to the number of moves, go to 4.
8. Reorganize the boundary and inside using the algorithm described above. For example, if the site with state 2 is surrounded by only neighbours with state 2, its state is changed to 1. Return to 1.

It should be noted that, depending on the location of the stimulation, the model behaves differently. If the stimulus point is chosen randomly, the CELL moves like an amoeba. If the stimulus is chosen from several active zones, an adaptive network with tentacles forms.

Example 2.2. In order to illustrate the model, we shall consider the CELL model in a 5×5 grid. After the initial growth stage, the CELL model is seen in figure 2*a*. By randomly choosing a cell (seen in grey) to be the stimulus point, and choosing a neighbour and swapping, one can see the following step in figure 2*b*.

In the following steps, the bubble is randomly swapped with neighbours more times. By repeating this step multiple times, one obtains figure 3*a*. Finally, one can reassign states to the cells, leading to figure 3*b*.

A typical CELL has bubbles within its cytoplasm. If enough of these bubbles group together, a chamber can form within the CELL. When this chamber bursts, tentacles are formed, as illustrated in figure 4.

This model can also be used to form networks and find the shortest path. This is mainly done by creating active zones in which the stimulus point is always chosen from. For example, in figure 5, active zones are created in three regions of the cell, leading to the cell forming a tree between the three active zones. Finally, this model has been previously

(a)

0	0	2	0	0
0	2	1	2	0
2	1	1	1	2
0	2	1	2	0
0	0	2	0	0

(b)

0	0	2	0	0
0	2	2	2	0
2	2	2	2	2
0	2	2	0	2
0	0	2	0	0

Figure 2. The first two steps in a CELL model on a 5×5 grid.

(a)

0	0	2	0	0
0	2	0	2	0
2	2	2	2	2
0	2	2	2	2
0	0	2	0	0

(b)

0	0	2	0	0
0	2	0	2	0
2	1	2	1	2
0	2	1	2	2
0	0	2	0	0

Figure 3. The last two steps in a CELL model on a 5×5 grid.

Figure 4. An example image from the implementation of the described CELL algorithm.

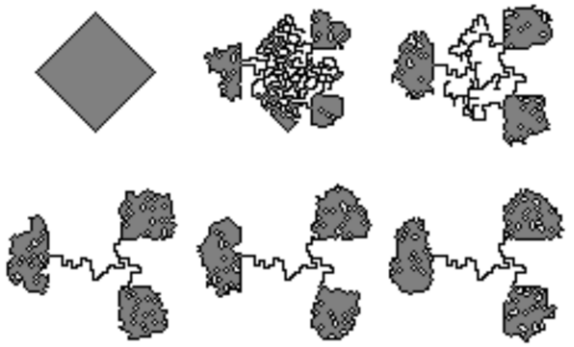

Figure 5. An example image from the implementation of the CELL algorithm with active zones for tree formation.

used to solve mazes, generate spanning trees [8] and model crowd evacuation [22].

2.3. Multi-agent model

Inspired by the many different behaviours *Physarum* demonstrates, the authors of [9] introduced a multi-agent model where *Physarum* is made out of agents that travel throughout *Physarum*. As these agents move, they make their decisions based on a chemoattractant map and also deposit chemicals onto this map. The multi-agent model also exhibits multiple properties consistent with those observed in actual *Physarum* such as the disappearance of open-ended tubes, minimization of path length, movement of agents similar to that in shuttle streaming and amoeboid movement [23].

We shall represent the model as a large grid similar to the CELL model, and consider an agent to fill a cell of the grid. Each agent also has three sensors which sample values a sensor offset (SO) distance away. The agents then use the readings from the three sensors and orient themselves towards the strongest chemoattractant reading from the sensors.

There are two important parameters which need to be considered in this set-up:

— there is sensor angle (SA), which is the angle between two of the three sensors;
— there is the rotation angle (RA), which is the amount the agent rotates to the strongest chemoattractant reading.

For example, if RA < SA, contraction behaviour is increased, but if RA > SA, spontaneous branching happens.

The agent first reads from each of its three sensors. It turns by the RA in the direction of the greatest sensor. After the agent has rotated accordingly, the agent attempts to move forwards. If it can move forwards (the cell is not occupied), chemoattractant is deposited (and the grid is updated) and the agent moves. Agents are randomly chosen to rotate and then move. In addition, a 3×3 mean filter is applied on the grid to simulate diffusion. A damping factor is usually added to this mean filter to create steeper gradients from food sources. Depending on where one initializes agents, and what the SA and RA are, one gets different behaviours for the system, which can be summarized as follows:

— *Filamentous condensation method*. Initialize a few agents at random locations and orientations. Then, a network forms. Add food sources which emit chemoattractants and the network will form a Steiner tree.
— *Filamentous foraging method*. Initialize agents at food sources.
— *Plasmodial shrinkage method*. Randomly remove agents; other agents will move and form a minimum spanning tree.

Example 2.3. To understand the multi-agent model described above, we shall consider the chemoattractant map shown in figure 6*a*, and assume that one has an agent with SA and RA = 45 at the shaded square in figure 6*a*. The rightmost sensor will then sense the highest chemoattractant (29) and the agent will rotate 45° to the right. The sensor then moves forwards if possible and deposits chemoattractant on the map (we shall assume it always deposits 5), and thus the next step in the model can be seen in figure 6*b*.

To simulate diffusion, we then apply a 3×3 mean filter to the map. Each cell of the map is assigned a value of 0.9 (0.1 dampening factor) times the average of the 3×3 surrounding box, leading to the chemoattractant map shown in figure 7.

Finally, it should be noted that through this model one can see that the agents gradually move from being randomly distributed to forming a network, as shown in figure 8.

(*a*)

10	16	15	22	21	12
11	15	7	29	19	13
21	23	26	11	17	39
16	10	18	10	1	2
19	13	6	9	11	32
12	23	13	18	17	28

(*b*)

10	16	15	22	21	12
11	15	7	29	19	13
21	23	26	11	17	39
16	10	18	10	1	2
19	13	6	9	11	32
12	23	13	18	17	28

Figure 6. The first two steps in a multi-agent model on a 6×6 chemoattractant map. The agent at the grey box in (*a*) moves forwards and deposits 5 on the chemoattractant map.

11.7	11.1	15.6	16.95	17.4	14.625
14.4	14.9	16.9	17.2	18.3	18.15
14.4	15.2	15.4	14.3	14.1	13.65
15.3	15.7	13.1	11.4	13.2	15.3
13.95	13	12	10.3	12.8	13.65
15.075	12.9	12.3	11.1	17.25	19.8

Figure 7. The chemoattractant map after a 3×3 mean filter with a dampening factor is applied to the map shown in figure 6*b*.

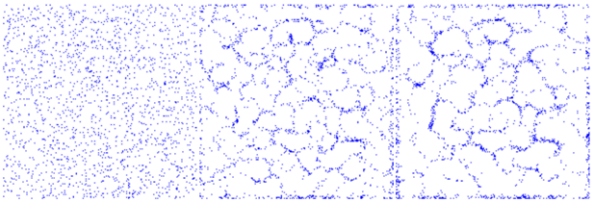

Figure 8. Images from a multi-agent model implementation.

2.4. Shuttle-streaming model

We shall next describe the shuttle-streaming model following [11]. This model is based on three biological observations about *P. polycephalum*:

— open-ended tubes gradually disappear;
— when two tubes connect the same food sources, the longer one disappears;
— outside changes such as the addition of nutrients can cause changes to the rhythmic contractions and shuttle streaming of *Physarum*.

Shuttle streaming is the flow of protoplasm through *Physarum*'s tubes and plays an important role in chemical signalling. When the organism contracts periodically (1–3 min), the direction of the shuttle streaming changes as hydrostatic pressure is produced.

In order to find the shortest path from node v_S to node v_F, which are two food sources, one considers two different protoplasmic flows, originating at each of the two nodes v_S and v_F.

2.4.1. Forward flow (v_S to v_F)

The amount $\mathrm{Re}_f((v_i, v_j), t)$ of nutrients received from node v_i through edge (v_i, v_j) during forward flow is given by

$$\mathrm{Re}_f((v_i, v_j), t) = \mathrm{Se}_f((v_i, v_j), t) - \eta L_{ij}, \tag{2.6}$$

where $\mathrm{Se}((v_i, v_j), t)$ is the amount of nutrients sent through edge (v_i, v_j) at node v_i, the value L_{ij} is the length of the edge and η is the absorption rate per unit length of the tube wall. Additionally, $\mathrm{totRe}(v_j, t)$ is the total amount of nutrients that node v_j receives.

The node v_j distributes the total amount of nutrients received in forward flow proportionally to the amount received in previous backward flow. Hence, the amount of nutrients sent out through edge (v_j, v_k) by node v_j is given by

$$\mathrm{Se}_f((v_j, v_k), t) = \mathrm{totRe}_f(v_j, t) \times \frac{\mathrm{Re}_b((v_k, v_j), t - \delta t)}{\mathrm{totRe}_b(v_j, t - \delta t)},$$

$$\mathrm{totRe}_f(v_j, t)v = \sum_{\forall v_i \in (v_i, v_j)} \mathrm{Re}_f((v_i, v_j), t)$$

and

$$\mathrm{totRe}_b(v_j, t - \delta t) = \sum_{\forall v_k \in (v_k, v_j)} \mathrm{Re}_b((v_k, v_j), t - \delta t).$$

2.4.2. Backward flow at time $t + \delta t$

Similar to the forward flow, one has that

$$\mathrm{Re}_b((v_p, v_q), t + \delta t) = \mathrm{Se}_b((v_p, v_q), t + \delta t) - \eta L_{pq}.$$

The corresponding equations for

$$\phi := \frac{\mathrm{Re}_f((v_r, v_q), t)}{\mathrm{totRe}_f(v_q, t)}$$

are given by

$$\mathrm{Se}_b((v_q, v_r), t + \delta t) = \mathrm{totRe}_b(v_q, t + \delta t) \times \phi,$$

$$\mathrm{totRe}_b(v_q, t + \delta t) = \sum_{\forall v_p \in (v_p, v_q)} \mathrm{Re}_b((v_p, v_q), t + \delta t)$$

and

$$\mathrm{totRe}_f(v_q, t) = \sum_{\forall v_r \in (v_r, v_q)} \mathrm{Re}_f((v_r, v_q), t).$$

The two nodes v_S and v_F act as both source and sink nodes depending on the direction of the protoplasmic flow. Assuming that the amount of nutrients the tubular network absorbs per one forward or backward flow session is N_{in},

the amount of nutrients sent and received by nodes v_s, v_F can be computed in the following fashion.

At $t = 0$, one can compute the number of nutrients sent from node v_j as follows:

$$\mathrm{Se}_f((v_j, v_k), T = 0) = \frac{\mathrm{totRe}_f(v_j, t = 0)}{\sum_{\forall v_k \in (v_j, v_k)} 1}. \tag{2.7}$$

This can be thought of as the total amount of nutrients divided by the number of edges. When the amount of nutrients carried by an edge is zero or negative, we can discard the edge since biologically it would die without nutrients. One continues to do this until the end of the optimization process is reached, at which point each node only has one edge that nutrients flow in from and one edge that nutrients flow out of.

Example 2.4. To illustrate this model, consider a similar graph to the flow-conductivity model of figure 1, as shown in figure 9.

Let $N_{\mathrm{in}} = 10$, which means that one puts in 10 nutrients every time. The starting values can then be computed using equation (2.7). Let $\eta = 0.1$. Then we have that

$$\begin{aligned}
&\mathrm{Se}_f((1, 2), 0) = 10,\\
&\mathrm{Re}_f((1, 2), 0) = 9.8,\\
&\mathrm{Se}_f((2, 3), 0) = 4.9,\\
&\mathrm{Re}_f((2, 3), 0) = 4.6,\\
&\mathrm{Se}_f((2, 4), 0) = 4.9,\\
&\mathrm{Re}_f((2, 4), 0) = 4.2,\\
&\mathrm{Se}_f((3, 5), 0) = 4.6,\\
&\mathrm{Re}_f((3, 5), 0) = 4.2,\\
&\mathrm{Se}_f((4, 5), 0) = 4.2,\\
&\mathrm{Re}_f((4, 5), 0) = 4.1,\\
&\mathrm{Se}_f((5, 6), 0) = 8.3\\
\text{and}\quad &\mathrm{Re}_f(5, 6) = 7.7.
\end{aligned}$$

The values of $\mathrm{totRe}_f(v, 0)$ can be computed in a similar manner, and are given by

$$\begin{aligned}
&\mathrm{totRe}_f(1, 0) = 10,\\
&\mathrm{totRe}_f(2, 0) = 9.8,\\
&\mathrm{totRe}_f(3, 0) = 4.6,\\
&\mathrm{totRe}_f(4, 0) = 4.2,\\
&\mathrm{totRe}_f(5, 0) = 8.3\\
\text{and}\quad &\mathrm{totRe}_f(6, 0) = 7.7.
\end{aligned}$$

Now it is time for backward flow. Starting from the opposite end, one sets $\mathrm{Se}_b((6, 5), 1) = 10$, and so $\mathrm{Re}_b((6, 5), 1) = 9.4$. Then, one has that

$$\begin{aligned}
&\mathrm{Se}_b((5, 3), 1) = 4.7566267,\\
&\mathrm{Re}_b((5, 3), 1) = 4.3566267,\\
&\mathrm{Se}_b((5, 4), 1) = 4.64337,\\
&\mathrm{Re}_b((5, 4), 1) = 4.54337,\\
&\mathrm{Se}_b((3, 2), 1) = 4.3566267,\\
&\mathrm{Re}_b((3, 2), 1) = 4.0566267,\\
&\mathrm{Se}_b((4, 2), 1) = 4.54337,\\
&\mathrm{Re}_b((4, 2), 1) = 3.84337,\\
&\mathrm{Se}_b((2, 1), 1) = 7.9\\
\text{and}\quad &\mathrm{Re}_b((2, 1), 1) = 7.7.
\end{aligned}$$

As done with forward flow, one can compute totRe_b and then repeat the same procedure with forward flow. After repeating the procedure several times, the flow through certain edges will eventually become very small, at which point one can remove such edges until the shortest path is obtained.

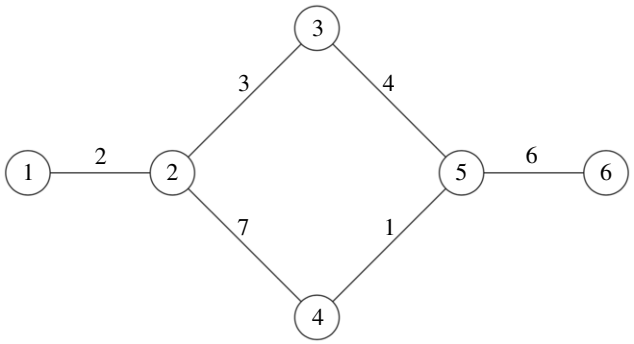

Figure 9. Graph as in figure 1 but without the edge between nodes 3 and 4. The numbers on each edge represent the length of that edge.

Since the model is a shortest path algorithm, it can be thought of as being very similar to flow conductivity, so any of the shortest path applications such as traffic routing should be possible, although this model has been shown to not always give the shortest path.

3. Exploring interactions between multiple cells

In what follows, we shall introduce a modified version of the CELL model described theoretically in [8] with memorized flow. After creating code to model our new algorithm (source code and a video demonstration are available at [24,25]), we shall investigate the interactions between multiple cells. To do this, our new model incorporates a unique ID corresponding to a colour for each cell, so that each cell is spawned with a unique colour, allowing us to distinguish between the cytoplasm of each cell.

Example 3.1. An example of our model is shown in figure 10, where one can see two CELLs initially spawned in a diamond shape. After a few iterations, one can see that the cells lose their diamond shape and become more circular, and there are multiple chambers within the cells.

By the third image of figure 10, the cells have made contact. In the fourth image, cytoplasm continues to flow between the two newly connected cells. In the fifth and sixth images, cytoplasm continues to flow further between the two cells. Finally, in the seventh image, the cytoplasm appears to start becoming well mixed; in the eighth image, the cell seems to have taken on a more circular or compact shape after initially being elongated owing to two cells combining. In the ninth image, the cell seems circular and the cytoplasm is well dispersed. In the following sections, we shall see further analysis of how cells fuse through our model.

3.1. Time to cell fusion in relation to spawn distance

We shall dedicate this section to understanding how the distance that two CELLs are spawned apart affects the number of iterations it takes for the cells to contact. Across this section, we use the following parameters: $n = 1000$, $s = 3$, and define two CELLs contacting as a cytoplasm of one ID being adjacent to a cytoplasm of another CELL. To illustrate this definition, figure 11 is an example of the first place that two CELLs contact.

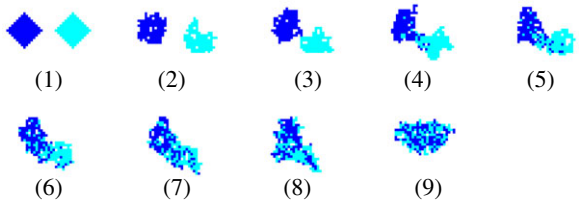

Figure 10. The spawn of two CELLs at (15, 20) and (35, 20), with parameters: size = 15, $s = 3$, $n = 1000$.

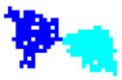

Figure 11. In this image, the two cells have just contacted for the first time.

To understand the behaviour of two cells through our model, we run 1025 trials of each spawn distance from 16 to 34. We graph the average iterations of each trial versus the spawn distance in figures 12 and 13. These plots show that there are extreme upper outliers, which is expected given the random nature of the cell model. The boxplot also shows an overall increasing trend in the number of iterations as the spawn distance increases.

In order to understand the results inferred from our experiments, we fit our findings to curves of linear, exponential and logistical functions

$$y = mx + b, \tag{3.1}$$

$$y = a \cdot b^x + c \tag{3.2}$$

and

$$y = \frac{L}{1 + e^{-k(x - x_0)}} + b. \tag{3.3}$$

The 10% trimmed mean of each spawn distance is given in figure 14, where one can see that these data appear to be linear (equation (3.1)) with values of $m = 18\,873.73644734$ and $b = -291\,495.3914131$. It can be seen that the linear function gives an r^2 value of 0.9707 and appears to fit the data, allowing us to conclude that the relationship between time to merge and the distance apart is linear.

3.2. Analysis of cell fusion with three cells

The premise of this experiment is that we spawn three CELLs in a straight line. We adjust the distances between the cells in order to see which cells fuse first as well as gather data on the number of iterations taken for the first cell fusion event to occur. We try different distances between the three CELLs ranging from 16 to 29 inclusive. We keep track of how many iterations it takes for the first two CELLs to merge and which two cells end up merging. We perform 350 trials. In general, the distance between cells 1 and 2, or *distance 1*, will be greater than or equal to the distance between cells 2 and 3, or *distance 2*. We always spawn cell 2 in the centre of the image and then spawn cells 1 and 3 the set distance away from cell 2. For example, in figure 15, the cells are from left to right 1, 2 and 3.

In order to visualize the data from our experiments, we use a three-dimensional scatterplot in figure 16, through which we see that there are extreme outliers when both distances are very large.

Moreover, one can see that, for every set of distances, there are three different possibilities: cells 1 and 2 join, cells 2 and 3 join or cells 1 and 3 join. For each of these possibilities, we compute the mean number of iterations it took for the cells to fuse. This is displayed in figure 17. One sees that, as the distances increase, so does the number of iterations. In general, it appears that the number of iterations for cells 1 and 3 (green bar) to join is overall much larger than the number of iterations for 1 and 2 or 2 and 3.

In order to understand how and when two cells fuse together, we compute the probability that a certain two cells will join, depicted in figure 18. Overall, one sees that the probability that cells 1 and 3 join is the lowest, which is expected since cell 2 is between cells 1 and 3. Cells 2 and 3 have the highest probability of joining, which is consistent with the fact that they are closer.

The number of iterations it takes for two out of the three cells to fuse steeply increases as the distance between the cells does, and this is shown in figure 19. It is important to note that distance 1 was always greater than distance 2, so the upper triangle of this plot had no data.

3.3. The impact of cell size on movement

In what follows, we shall show the importance of the CELL's size in its ability to move. In this section, we measure mobility by computing three values: the average distance from the spawn point, the furthest distance from the spawn point and the total distance travelled. After every iteration of the cell algorithm, we compute the centre of mass of the cell by iterating through all pieces of cytoplasm and computing the average x and y values. We then compute the distance from the spawn point using the distance formula with the cell's current centre of mass and the spawn point (49, 49). Finally, we compute the total distance travelled by summing the distance between the current centre of mass and the centre of mass an iteration before. For each calculation, we have 2000 trials of each cell size.

We shall begin by analysing the average distance from the spawn point. To do this, we compute the 10% trimmed mean of the average distance for each given cell size, and then fit an exponential curve to these data. The results are shown in figure 20. The exponential curve (equation (3.2)) $a = 8.0305104$, $b = 0.91459648$ and $c = 0.36077371$ appears to fit the data points very well, suggesting that the relationship between the average distance and size resembles that of exponential decay.

In what follows, we shall consider the largest distance from the spawn point: this value represents the furthest point that a cell has travelled to. We first compute the 10% trimmed mean of the average distance for each given cell size. We then fit the exponential curve (equation (3.2) with values $a = 15.91026266$, $b = 0.91514874$ and $c = 0.71778398$, as shown in figure 21, allowing us to conclude that the relationship between the largest distance and area resembles that of exponential decay.

Finally, we shall analyse the speed of the fusion: at each iteration of our modified CELL model, we compute the distance of the centre of mass from the previous centre of mass. Averaging all the distances to find the average speed of the cell, we see that the data fit an exponential curve, as shown in figure 22.

3.4. Cytoplasm mixing in cell fusion

In this section, we shall further consider how the sizes of cells can impact their fusion. To do so, we spawn two cells of varying size and measure how well the cytoplasms of each cell mix together. We use Lacy's mixing index M, described in

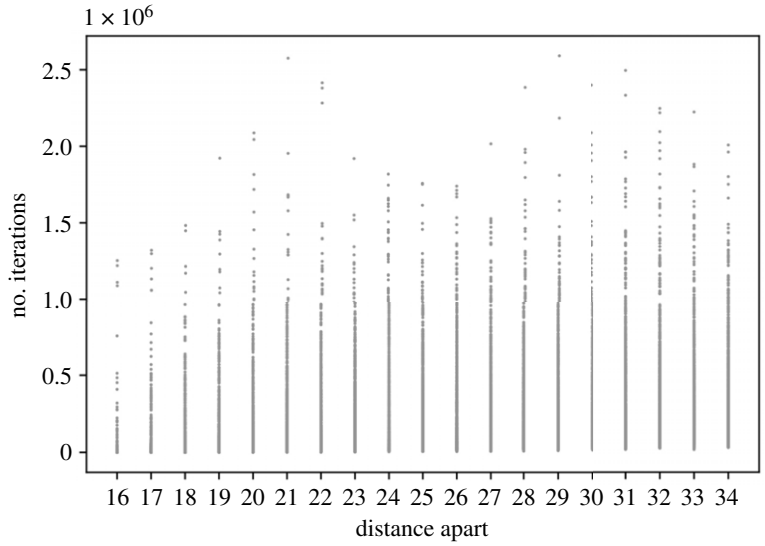

Figure 12. Graph of the 1025 trials of each distance: the distance between the centre of the spawn points is on the *x*-axis, and the number of iterations until they merge is on the *y*-axis.

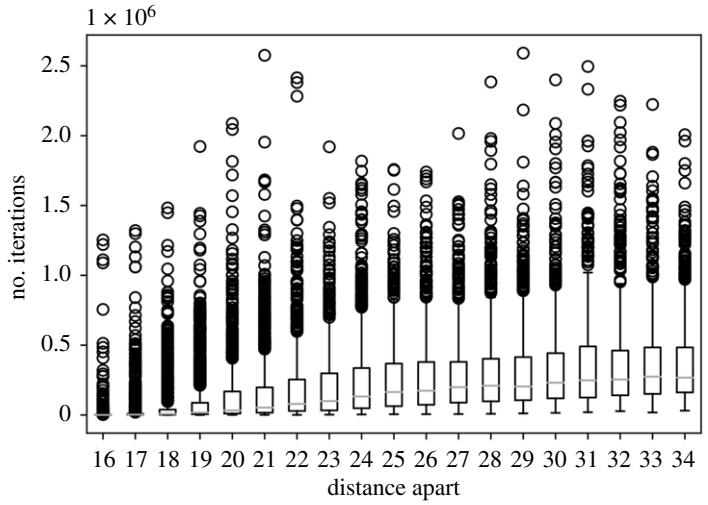

Figure 13. The 1025 trials of each distance: the distance between the centre of the spawn points is on the *x*-axis and the number of iterations until they merge is on the *y*-axis.

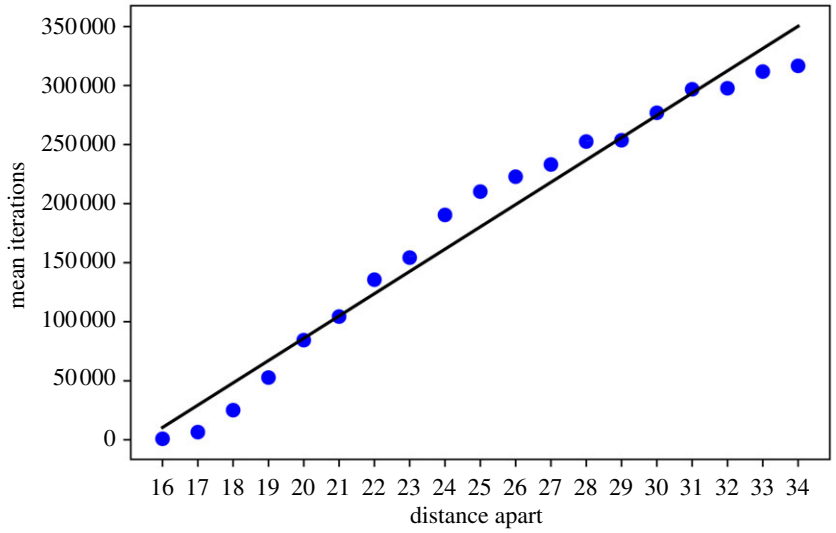

Figure 14. To deal with the heavily skewed data as evidenced by the scatter and box plots, we trim 10% from both the upper and lower ends of the data. We then compute the trimmed mean. We use the scipy.optimize.curve_fit function to fit these data to a linear function (equation (3.1)) with the values $m =$ 21 892.74674907 and $b = -354\,967.6942326$. This gives an r^2-value of 0.970675949.

Figure 15. Three cells of size 15, labelled from left to right as 1, 2 and 3. The distance between cells 1 and 2 (*distance 1*) is 27 and between cells 2 and 3 (*distance 2*) is 23.

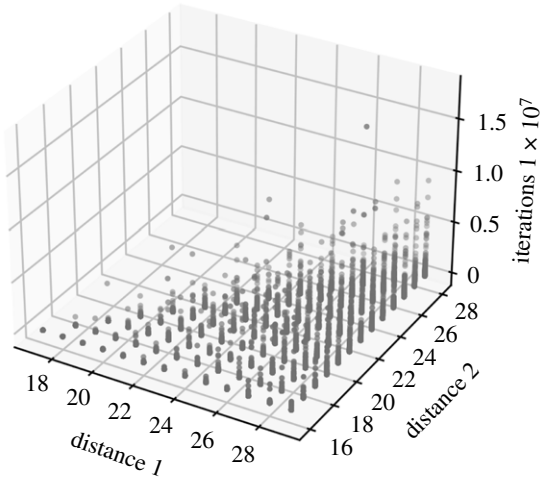

Figure 16. Using the Python package matplotlib, we create a three-dimensional scatterplot. On the *xy* plane are the distances between the three cells and on the *z*-axis is the number of iterations it takes for two cells to combine.

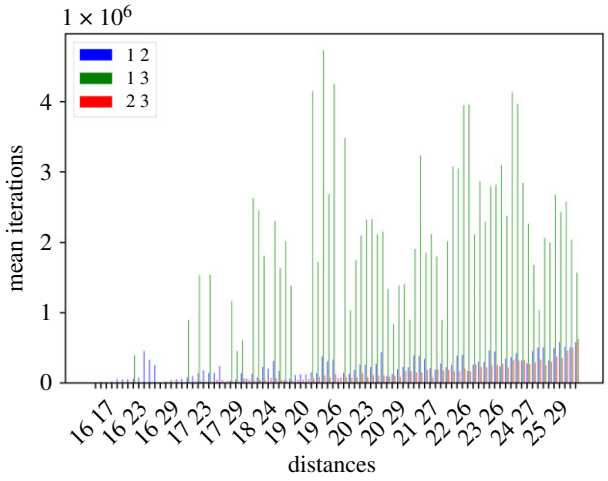

Figure 17. Using the Python package matplotlib, we compute the mean of each set of distances. For each distance, we have three bars: one for each of the possible two CELLs combining.

[12], to measure the mixing of cytoplasm. Here, the variable M ranges from 0 to 1, where 0 is not mixed at all and 1 is completely mixed, and is defined in equation (3.4)

$$M = \frac{S_0^2 - S^2}{S_0^2 - S_R^2}, \tag{3.4}$$

where S_0 and S_R are defined in (3.5) and (3.6), respectively;

$$S_0^2 = qp \tag{3.5}$$

and

$$S_R^2 = \frac{qp}{N}, \tag{3.6}$$

for q the proportion of the mixture that is of the first substance, p the proportion of the mixture that is of the second substance and N the number of particles in the samples.

The standard deviation over n samples is shown in (3.7), where we define x_i to be the proportion of the sample that is the first substance, and set $\bar{x}$ to equal p,

$$S^2 = \frac{1}{n}\sum_{i=1}^{i=n}(x_i - \bar{x})^2. \tag{3.7}$$

As a point of reference, figure 23 has a mixing index of 0.726.

Considering a starting cell of size 15, we then spawn another cell of varying size from 15 to 49 (odd sizes only), and run 1000 trials for each of the sizes. Each trial consists of 10 000 iterations of our modified CELL model. The cells are spawned so that they are initially barely touching, as shown in figure 24. In each iteration, we compute the mixing index as discussed above. We sample the current state of the CELLs by taking all 3×3 squares in the grid. If more than half of the 3×3 square is not cytoplasm, we disregard the sample. For each of the trials, we compute the mean value of the mixing index as well as the largest mixing index.

We shall first analyse the mean mixing index M with respect to area, which is proportional to size, and which we have been analysing in the previous sections, squared. For each area size, we trim the lower and upper 10% of the dataset and graph the 10% trimmed mean in figure 25. These data are clearly best fitted by an exponential function.

We now analyse the maximum mixing index, or the highest amount of mixing achieved in the 10 000 iterations. We see in figure 26 that the maximum mixing index decreases as size increases and has more lower outliers than upper outliers.

Finally, we compute the 10% trimmed mean of the maximum mixing index, and fit these data to a logistical function in figure 27.

4. Concluding remarks

In the present paper, we have modified the original CELL model [8] in order to be able to study interactions between multiple CELLs of varying sizes and at different distances. Four different experiments were run. The first of these experiments measured the time it took for two cells to come into contact based on the distance between them. The second of these experiments tested which two cells out of three would merge first. The third of these experiments tested the mobility of cells in relation to cell size. The fourth of these experiments measured the mixing of cytoplasm of two cells in relation to cell size.

4.1. Summary of results

Our observed results corroborate multiple expected *Physarum* behaviours: merging time increases with distance, cells merge with closer cells, cells are more mobile and smaller cells fuse better. This further validates the CELL model as well as the modified CELL model with multiple CELLs of varying sizes and at varying distances. Furthermore, we have been able to identify the relationship between important characteristics such as speed versus distance (experimental decay) which would have been difficult to determine by experimentally growing *Physarum*.

From the first experiment, we were able to identify that the correlation between the time to contact and the

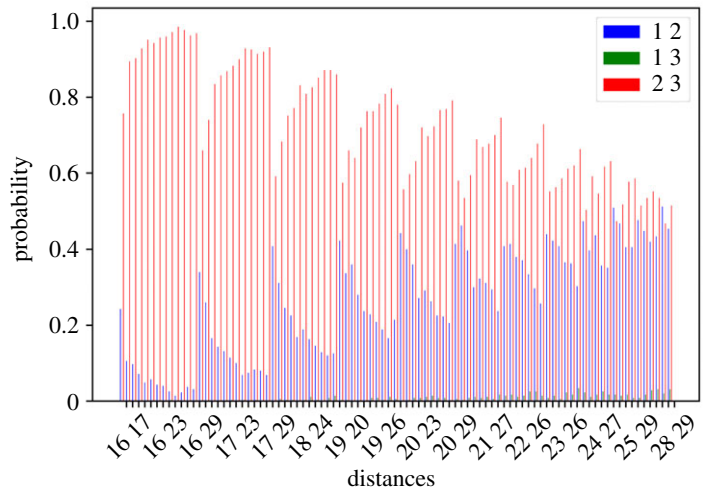

Figure 18. Using the Python package matplotlib, we compute the probability that, for a given set of distances between CELLs, each combination of two cells (1 2, 1 3, 2 3) will be the ones that combine first.

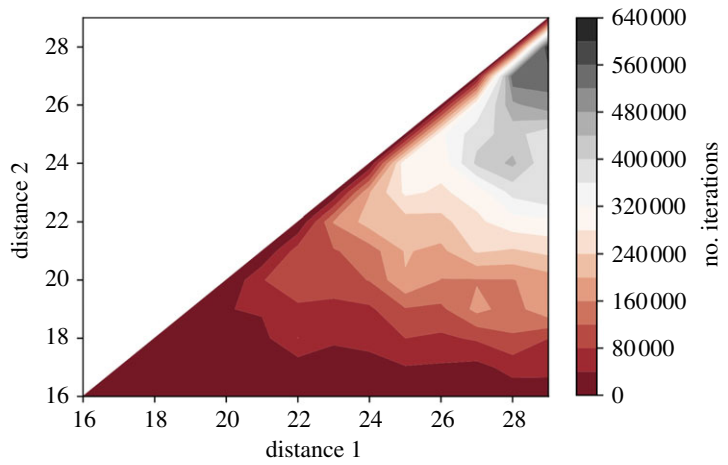

Figure 19. Using the Python package matplotlib, we create a contour plot. On the *xy* plane are the distances between the three cells; the colour gradient represents the number of iterations it takes for two cells to combine.

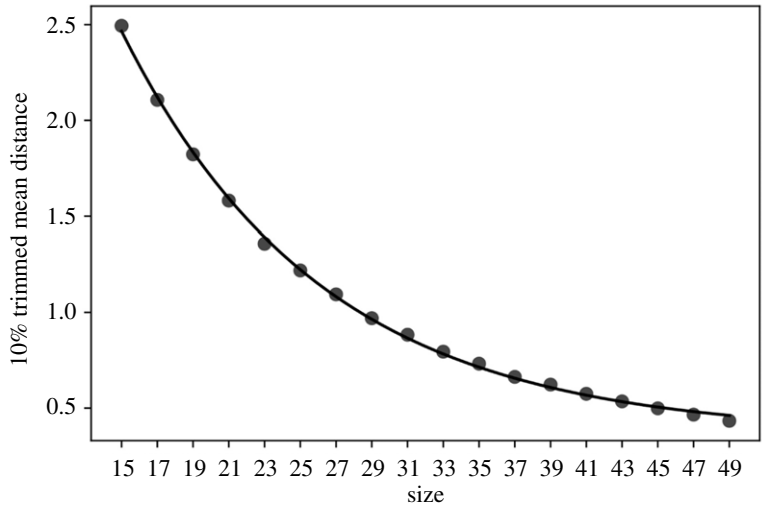

Figure 20. We compute the 10% trimmed mean of each cell size. We then fit an exponential curve to the data.

distance between CELLs is linear. In the second experiment, we determined that CELLs most often merge with the CELLs closest to them and do so quicker than merging with other CELLs. In the third experiment, we determined that there is an exponential decay relationship between measures of mobility (speed, distance from start) and size. In the fourth experiment, we identified the relationship between the mean mixing index and area as exponential decay and between the maximum mixing index and size as logistical decay.

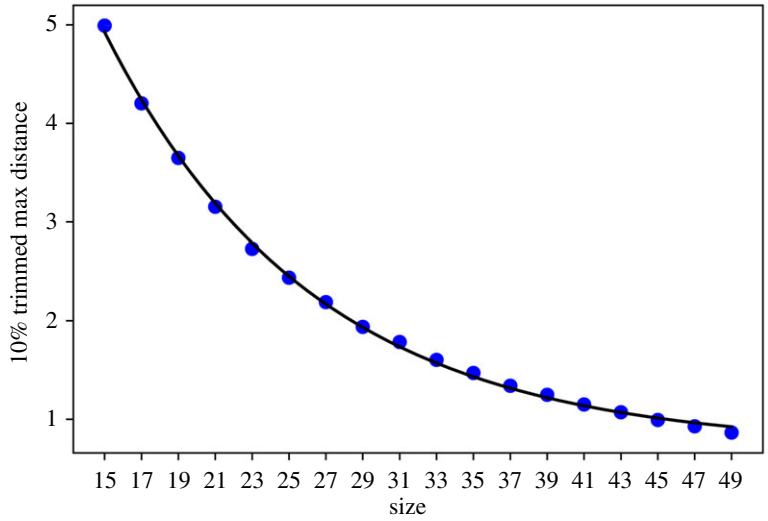

Figure 21. We compute the 10% trimmed mean of each cell size. We then fit an exponential curve to the data.

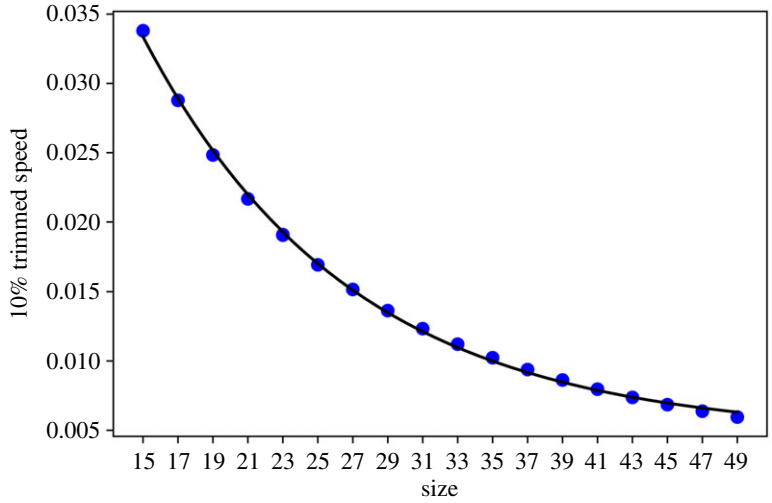

Figure 22. We compute the 10% trimmed mean of the 1000 trials. We then attempt to fit the data to the exponential curve (equation (3.2)) with values $a = 0.10144046$, $b = 0.9192343$ and $c = 0.0046737$.

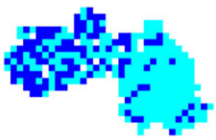

Figure 23. Two cells of size 15 and 20, respectively. After 5000 iterations, the mixing index is 0.726373742358510.

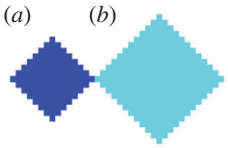

Figure 24. Here, we have two CELLs of size 15 (*a*) and 23 (*b*). The CELLs have just been spawned so that they are barely touching each other at the centre of the image.

4.2. Applications

We foresee many applications of our work. In particular, our modified CELL model can be used in the following set-ups:

1. Fusion between multiple CELLs can also be used to model the rise of present-day civilization from the early nomadic humans [26]. Each tribe or group is modelled as a cell. Smaller groups or tribes are able to travel faster. As the size grows, they become less and less mobile until permanent settlement occurs. As the groups travel around, they meet other groups where they could merge. As with the results described in this paper, groups closer together have a closer connection than groups further apart.
2. Multiple CELLs can be used to model the spread of knowledge among groups of people. Following the ideas of [27], certain groups would begin with certain knowledge; when they encounter another group, they would combine and both share the knowledge, which is possibly modified. In such a way, one could potentially understand the spread and deformation of ideas and information.

In addition, there is more biological work that can be done to verify these results and further validate the CELL model [8] as well as our generalized CELL model. In particular, each of the experiments done with the CELL model can

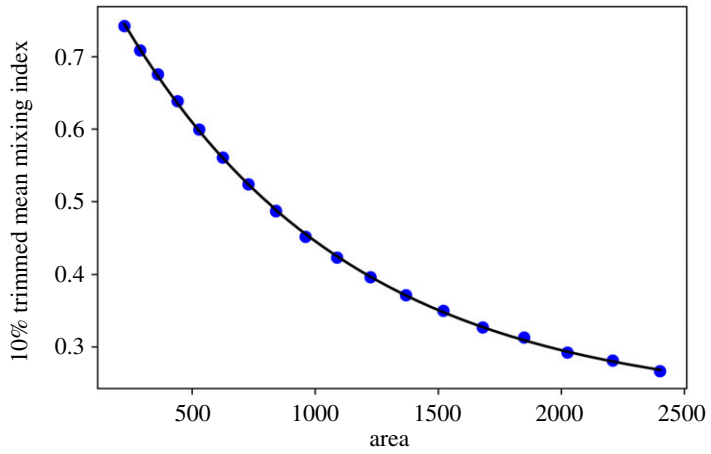

Figure 25. We compute the 10% trimmed mean of the mean mixing index. We fit these data to an exponential function (equation (3.2)) pictured in blue with values $a = 0.67188919$, $b = 0.99891735$ and $c = 0.21859742$.

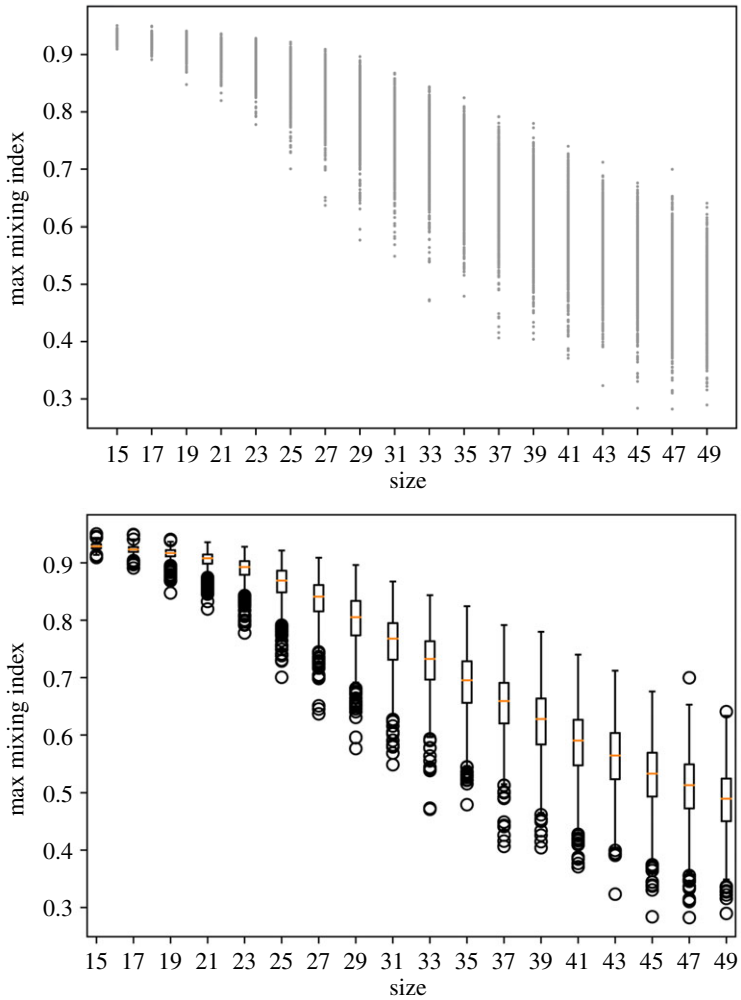

Figure 26. We graph the data from our 1000 trials of each size ranging from 15 to 49 (odd only) using a scatterplot and boxplot.

be done with physical *P. polycephalum*. In addition, multiple *Physarum* organisms can share information and communicate prior to protoplasm fusion through the mucus that surrounds *Physarum* [28]. New models can be developed or existing models can be modified to represent this additional aspect of multiple *Physarum* fusing. The fusion of *Physarum* is very complex, and the model presented in this paper can be further developed and tweaked to better represent it.

There is also potential to use multiple *Physarum* organisms to solve problems that a single *Physarum* has already solved, such as maze solving and tree formation, and this is currently being studied in [29].

4.3. Final comments

Exploring the interaction between multiple *P. polycephalum* organisms is essential to discovering more about *Physarum*'s

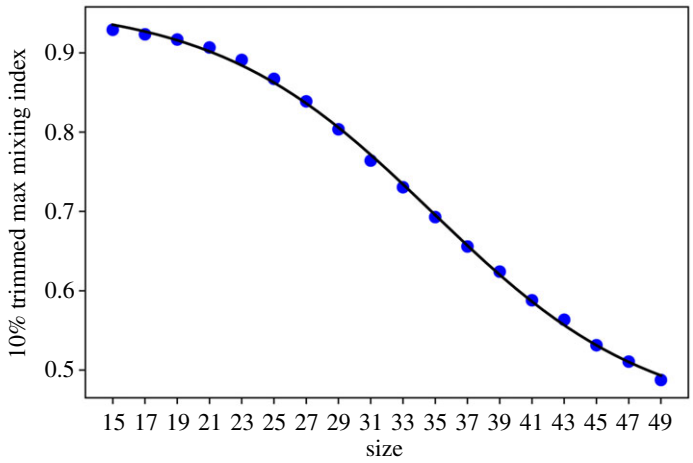

Figure 27. We compute the 10% trimmed mean of the mean mixing index. We fit these data to a logistical function (equation (3.3)) pictured in blue. The parameters are $L = -0.52639764$, $k = 0.52639764$, $x_0 = 34.83393314$ and $b = 0.96196691$.

unique ability as a unicellular organism to communicate with other cells and remember past events. More research into *Physarum*'s unique learning, communication and memory capabilities will allow us to discover more about possible unconventional solutions to the world's largest problems.

Data accessibility. This article has no additional data.

Authors' contributions. S.H.: writing—original draft; L.P.S.: writing—original draft.

Both authors gave final approval for publication and agreed to be held accountable for the work performed herein.

Conflict of interest declaration. We declare we have no competing interests.

Funding. The work of L.P.S. is partially supported through NSF grant no. CAREER DMS 1749013.

Acknowledgements. The authors are thankful to MIT PRIMES-USA for the opportunity to conduct this research together, and in particular to Fidel A. Schaposnik for bringing *Physarum polycephalum* to our attention after it started growing in his boiler.

References

1. Fessel A, Oettmeier C, Bernitt E, Gauthier NC, Döbereiner H-G. 2012 Physarum polycephalum percolation as a paradigm for topological phase transitions in transportation networks. *Phys. Rev. Lett.* **109**, 078103. (doi:10.1103/PhysRevLett.109.078103)
2. Alim K, Andrew N, Pringle A, Brenner MP. 2017 Mechanism of signal propagation in *Physarum polycephalum*. *Proc. Natl Acad. Sci. USA* **114**, 5136–5141. (doi:10.1073/pnas.1618114114)
3. Kramar M, Alim K. 2021 Encoding memory in tube diameter hierarchy of living flow network. *Proc. Natl Acad. Sci. USA* **118**, e2007815118. (doi:10.1073/pnas.2007815118)
4. Nakagaki T, Yamada H, Tóth Á. 2000 Maze-solving by an amoeboid organism. *Nature* **407**, 470–470. (doi:10.1038/35035159)
5. Tero A, Takagi S, Saigusa T, Ito K, Bebber D, Fricker M, Yumiki K, Kobayashi R, Nakagaki T. 2010 Rules for biologically inspired adaptive network design. *Science* **327**, 439–442. (doi:10.1126/science.1177894)
6. Boisseau RP, Vogel D, Dussutour A. 2016 Habituation in non-neural organisms: evidence from slime moulds. *Proc. R. Soc. B* **283**, 20160446. (doi:10.1098/rspb.2016.0446)
7. Vogel D, Dussutour A. 2016 Direct transfer of learned behaviour via cell fusion in non-neural organisms. *Proc. R. Soc. B* **283**, 20162382. (doi:10.1098/rspb.2016.2382)
8. Gunji Y, Gunji Y-P, Shirakawa T, Niizato T, Haruna T. 2008 Minimal model of a cell connecting amoebic motion and adaptive transport networks. *J. Theor. Biol.* **253**, 659–667. (doi:10.1016/j.jtbi.2008.04.017)
9. Jones J. 2009 Approximating the behaviours of *Physarum polycephalum* for the construction and minimisation of synthetic transport networks. In *Unconventional computation* (eds CS Calude, JF Costa, N Dershowitz, E Freire, G Rozenberg), pp. 191–208. Berlin, Germany: Springer.
10. Tero A, Kobayashi R, Nakagaki T. 2006 Physarum solver: a biologically inspired method of road-network navigation. *Physica A* **363**, 115–119. (doi:10.1016/j.physa.2006.01.053)
11. Siriwardana J, Halgamuge S. 2012 Fast shortest path optimization inspired by shuttle streaming of *Physarum polycephalum*. In *2012 IEEE congress on evolutionary computation*, pp. 1–8. (doi:10.1109/CEC.2012.6252956)
12. Lacey PMC. 1954 Developments in the theory of particle mixing. *J. Appl. Chem.* **4**, 257–268. (doi:10.1002/jctb.5010040504)
13. Sun Y. 2019 Physarum-inspired network optimization: a review. (https://arxiv.org/abs/1712.02910)
14. Liu L, Song Y, Zhang H, Ma H, Vasilakos AV. 2015 Physarum optimization: a biology-inspired algorithm for the Steiner tree problem in networks. *IEEE Trans. Comput.* **64**, 818–831. (doi:10.1109/TC.2013.229)
15. Qian T, Zhang Z, Gao C, Wu Y, Liu Y. 2013 An ant colony system based on the Physarum network. In *Advances in swarm intelligence* (eds Y Tan, Y Shi, H Mo), pp. 297–305. Berlin, Germany: Springer.
16. Houbraken M, Demeyer S, Staessens D, Audenaert P, Colle D, Pickavet M. 2013 Fault tolerant network design inspired by *Physarum polycephalum*. *Nat. Comput.* **12**, 277–289. (doi:10.1007/s11047-012-9344-7)
17. Tero A, Takagi S, Saigusa T, Ito K, Bebber DP, Fricker MD, Yumiki K, Kobayashi R, Nakagaki T. 2010 Rules for biologically inspired adaptive network design. *Science* **327**, 439–442. (doi:10.1126/science.1177894)
18. Watanabe S, Takamatsu A. 2014 Transportation network with fluctuating input/output designed by the bio-inspired Physarum algorithm. *PLoS ONE* **9**, 1–11. (doi:10.1371/annotation/99b0e7bf-5878-42de-bcce-719ceea15466)
19. Liu Y, Deng Y, Jusup M, Wang Z. 2016 A biologically inspired immunization strategy for network epidemiology. *J. Theor. Biol.* **400**, 92–102. (doi:10.1016/j.jtbi.2016.04.018)
20. Zhang X, Chan FT, Adamatzky A, Mahadevan S, Yang H, Zhang Z, Deng Y. 2017 An intelligent Physarum solver for supply chain network design under profit maximization and oligopolistic competition. *Int. J. Prod. Res.* **55**, 244–263. (doi:10.1080/00207543.2016.1203075)
21. Wohlfarth-Bottermann KE. 1977 Oscillating contractions in protoplasmic strands of Physarum: simultaneous tensiometry of longitudinal and radial rhythms, periodicity analysis and temperature

dependence. *J. Exp. Biol.* **67**, 49–59. (doi:10.1242/jeb.67.1.49)

22. Kalogeiton V, Papadopoulos D, Georgilas I, Sirakoulis G, Adamatzky A. 2015 Cellular automaton model of crowd evacuation inspired by slime mould. *Int. J. Gen. Syst.* **44**, 354–391. (doi:10.1080/03081079.2014.997527)
23. Jones J. 2015 *From pattern formation to material computation: multi-agent modelling of Physarum polycephalum*, vol. 15. Berlin, Germany: Springer.
24. Hsu S. Model of multiple cells. See https://github.com/sher222/model-of-multiple-cells.
25. Hsu S. Cell fusion through slime mold dynamics. See https://youtu.be/JCLBW9uzqfY.
26. McNeill WH. 1984 Human migration in historical perspective. *Popul. Dev. Rev.* **10**, 1–18. (doi:10.2307/1973159)
27. Bhansali R, Schaposnik LP. 2020 A trust model for spreading gossip in social networks: a multi-type bootstrap percolation model. *Proc. R. Soc. A* **476**, 20190826. (doi:10.1098/rspa.2019.0826)
28. Ray SK, Valentini G, Shah P, Haque A, Reid CR, Weber GF, Garnier S. 2019 Information transfer during food choice in the slime mold *Physarum polycephalum*. *Front. Ecol. Evol.* **7**, 67. (doi:10.3389/fevo.2019.00067)
29. Hsu S, Schaposnik LP. The power of many: a Physarum swarm Steiner tree algorithm. See https://arxiv.org/abs/2110.08233.