

Programmable reconfiguration of *Physarum* machines

Andrew Adamatzky · Jeff Jones

Published online: 13 August 2009
© Springer Science+Business Media B.V. 2009

Abstract Plasmodium of *Physarum polycephalum* is a large cell capable of solving graph-theoretic, optimization and computational geometry problems due to its unique foraging behavior. Also the plasmodium is a unique biological substrate that mimics universal storage modification machines, namely the Kolmogorov–Uspensky machine. In the plasmodium implementation of the storage modification machine data are represented by sources of nutrients and memory structure by protoplasmic tubes connecting the sources. In laboratory experiments and simulation we demonstrate how the plasmodium-based storage modification machine can be programmed. We show execution of the following operations with the active zone (where computation occurs): merge two active zones, multiply active zone, translate active zone from one data site to another, direct active zone. Results of the paper bear two-fold value: they provide a basis for programming unconventional devices based on biological substrates and also shed light on behavioral patterns of the plasmodium.

Keywords *Physarum polycephalum* · Kolmogorov–Uspensky machine · Pattern formation · Morphogenesis · Graph theory

1 Introduction

Physarum polycephalum,¹ commonly known as a true or multi-headed slime mold, is—at one stage of its complicated life cycle—a single cell with many diploid nuclei (a syncytium). This mass of cytoplasm containing a great many nuclei and enclosed in a membrane, but no internal cell boundaries, is called plasmodium. It looks like an amorphous yellowish

¹ Species of order *Physarales*, subclass *Myxogastromycetidae*, class *Myxomycetes*, division *Myxozelida*.

A. Adamatzky (✉) · J. Jones
University of the West of England, Bristol BS16 1QY, UK
e-mail: andrew.adamatzky@uwe.ac.uk

J. Jones
e-mail: jeff.jones@uwe.ac.uk

mass. The plasmodium behaves and moves as a giant amoeba. It feeds on bacteria, spores and other microbial creatures. When foraging for its food the plasmodium propagates towards sources of food particles, surrounds them, secretes enzymes and digests the food. Typically the plasmodium forms a congregation of protoplasm in a food source it occupies. When several sources of nutrients are scattered in the plasmodium's range,² the plasmodium forms a network of protoplasmic tubes connecting the masses of protoplasm at the food sources. When we think of food sources as nodes and protoplasmic tubes as edges, we say the plasmodium develops a planar graph (all our experiments are done on a planar substrate, if the plasmodium did occupy a three-dimensional porous matrix, where it can propagate, then graphs would not be planar).

Nakagaki et al. (2000, 2001, 2001) showed that the topology of the plasmodium's protoplasmic network optimizes the plasmodium's harvesting on distributed sources of nutrients and makes more efficient flow and transport of intra-cellular components. Therefore the plasmodium is considered as a parallel computing substrate complementary (Adamatzky 2007b) to existing massive-parallel reaction-diffusion chemical processors (Adamatzky et al. 2005). Experimental observations suggest that during development of its protoplasmic network the plasmodium undergoes transitions between various classes of proximity graphs (Adamatzky 2008b). It starts with spanning trees (Adamatzky 2008a) and may complete its protoplasmic network development as a Delaunay triangulation (Shirakawa et al. 2008).

Implementation of a general purpose computing machine is the most remarkable feature of the plasmodium of *Physarum polycephalum*. In (Adamatzky 2007a) we experimentally demonstrated that the plasmodium can implement the Kolmogorov–Uspensky (KUM) machine (Kolmogorov 1953, Uspensky 1992), a mathematical machine in which the storage structure is an irregular graph. The KUM is a forerunner and direct 'ancestor' of Knuth's linking automata (Knuth 1968), Tarjan's reference machine (Tarjan 1977), and Schönhage's storage modification machines (Schönhage 1973, 1980]. The storage modification machines are basic architectures for random access machines, which are the basic architecture of modern-day computers. The plasmodium-based implementation of KUM (Adamatzky 2007a) provides a first-ever biological prototype of a general purpose computer.

The key component of the KUM is an active zone (Kolmogorov 1953, Uspensky 1992), which may be seen as a computational equivalent to the head in a Turing machine. Physical control of the active zone is of utmost importance because it determines functionality of the biological storage modification machine.

In the present paper we show—in laboratory and computer experiments with *Physarum polycephalum*—how basic operations ADD NODE, ADD EDGE, REMOVE EDGE are implemented in the *Physarum* machine. We also provide unique results on controlling the movement of an active zone. The paper is structured as follows. We provide a very brief introduction to Kolmogorov–Uspensky machine in Sect. 2. Section 3 describes a simple experimental setup for study of plasmodium. A particle-based model of the plasmodium is presented in Sect. 4. In Sect. 5 we discuss results of laboratory experiments and computer simulation on the reconfiguration of basic protoplasmic graphs. An overview of the results and proposed future experiments are outlined in Sect. 6.

² Experimentally we found that bacteria populating oat flakes are sensed by the plasmodium at a distance of up 2–3 cm.

2 *Physarum* machine

Kolmogorov–Uspensky Machine (KUM) is defined on a labeled undirected graph (storage structure) with bounded degrees of nodes and bounded number of labels (Kolmogorov 1953, Uspensky 1992). KUM executes the following operations on its storage structure: select an active node in the storage graph; specify local active zone, i.e. the node's neighborhood; modify the active zone by adding a new node with the pair of edges, connecting the new node with the active node; delete a node with a pair of incident edges; add/delete the edge between the nodes. A program for KUM specifies how to replace the neighborhood of an active node (i.e. occupied by an active zone) with a new neighborhood, depending on the labels of edges connected to the active node and the labels of the nodes in proximity of the active node (Blass and Gurevich 2003).

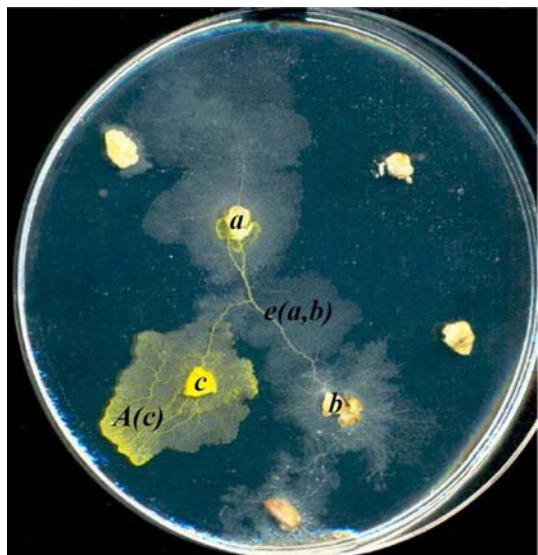
Physarum machine is a biological implementation of KUM, where a node of the storage structure is represented by a source of nutrients (e.g. an oat flake); an edge connecting two nodes is a protoplasmic tube linking two sources of nutrients corresponding to the nodes; and, an active zone is domain of space (which may include food sources) occupied by a propagating pseudopodium. An sample architecture of *Physarum* machine is illustrated in Fig. 1. Some schematic representations of active zones are provided in Fig. 2.

In *Physarum* machine the computation is implemented by the active zone, or several active zones. To make the computation process programmable one needs to find ways of sensible and purposeful manipulation with the active zones. Several operations, manipulation procedures, are discussed in the present paper.

3 Experimental method

The plasmodia of *Physarum polycephalum* were cultured on wet paper towels, fed with oat flakes, and moistened regularly. We subcultured the plasmodium every 5–7 days. Experiments on growing spanning trees were undertaken in standard Petri dishes, 9 cm in diameter. The substrate was wet filter paper. We preferred the filter paper not 2% agar gel,

Fig. 1 An exemplar snapshot of *Physarum* machine. Protoplasmic tube connecting flakes *a* and *b* represents an edge $e(a,b)$ of KUM. Active zone $A(c)$ emerges in the node *c*



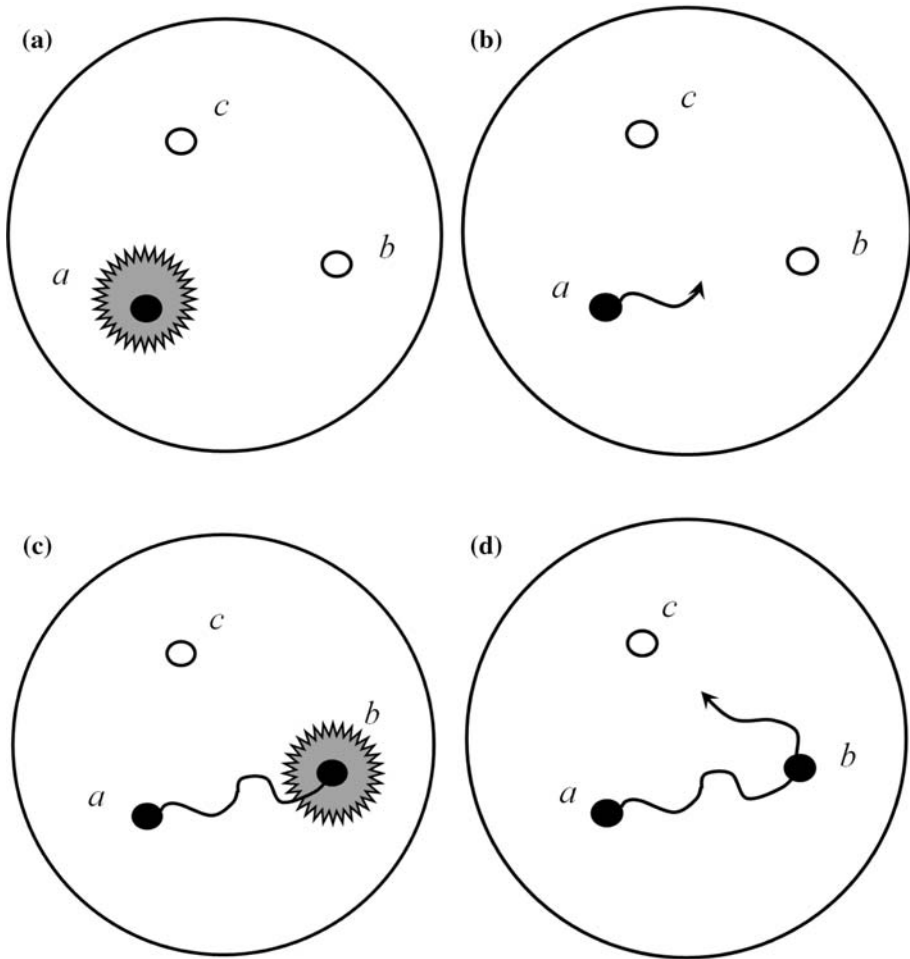


Fig. 2 Schematic dynamics of active zones: **a** active zone occupies node *a*, **b** active zone propagates towards node *b*, **c** active zone occupies node *b*, **d** active zone propagates towards node *c*. Oat flakes populated by plasmodium are shown by *solid discs*

because the paper offers less favoured conditions for the plasmodium growth, and thus less branching of the propagating pseudopodia is observed (see details in Adamatzky et al. 2008, Takamatsu 2007). The Petri dishes with plasmodia were kept in darkness and only exposed to light during observation and recording of images.

Data points, to be connected by protoplasmic graphs, were represented by oat flakes. Photographs of the protoplasmic networks, developed by the plasmodium, were made using the digital camera FinePix S6500. Colours of original images were enhanced by increasing contrast to 100 and decreasing gamma correction to 0.3.

4 Particle-based model approximating the behaviour of *Physarum* plasmodium

The computational model takes a multi-agent (particle) approach to generate emergent plasmodium-like behaviours. The model is based on the simple particle behaviours that

were used in (Jones 2008) to generate emergent transport networks. The model is computationally very simple, using distributed local sensory behaviours, but approximates some of the complex phenomena observed in *Physarum*: foraging for food stimuli, spatially distributed oscillations, oscillation phase shifting, shuttle streaming, amoebic movement, spontaneous network formation, surface area minimisation, network adaptation, and network minimisation behaviours. A full description of the model is in preparation, but an overview specifically in relation to plasmodium behaviour is given below.

A single particle, and an aggregation of particles, is related to the *Physarum* plasmodium in the following way: We conceptualise and simplify the plasmodium syncytium as an aggregate of identical component parts (formed by repeated nuclei division). Each ‘part’ represents a hypothetical unit of gel/sol interaction. Gel refers to the relatively stiff sponge-like matrix composed of actin/myosin fibres and sol refers to the protoplasmic solution which flows within the matrix. In *Physarum* the resistance provided by the gel matrix to the flow of sol, and the spontaneous contraction of the actin–myosin fibres, results in changes in thickness of the plasmodium due to hydrostatic pressure and a rhythmic streaming flow of sol throughout the matrix. Ultimately a complex set of distributed oscillations (of plasmodium thickness) is observed. When coupled to the effect of nutrient gradients at the periphery of the plasmodium, there is thought to be a temporary ‘softening’ of the gel structure in the presence of nutrients and the resulting surge of protoplasm generates an extending pseudopod process streaming towards the nutrients.

In the particle model the innate flow of sol is provided by the movement of the agents and the resistance of the gel matrix is provided by the agent–agent collisions (one can imagine a tightly crowded population within a room, with restricted movements). The cohesion of this aggregate ‘crowd’ is ensured by the fact that there is mutual attraction to the stimuli deposited by the agent population (described below). Collective movement and internal oscillations of the plasmodium are emergent properties of the aggregate behaviour of the agent population. Directional orientation and movement of the plasmodium active front is generated by coupling the emergent mass behaviours with chemoattraction to local food source stimuli.

To provide the computer model approximation of *Physarum* plasmodium active zone computation, the laboratory experiment configurations were used to map synthetic environments for the model organism. A population of mobile agents, adopting simple stimulus-response behaviours is created. The population adopts autocrine chemotaxis behaviours—agents both secrete and sense approximations of chemoattractants, so the actions of the agents can also affect their behaviour. The population is coupled to a two-dimensional discrete map representing the problem configuration—the ‘data’ map, Fig. 3.

Food source stimuli whose positions are stored in the data map are projected to another coupled map (the ‘chemoattractant’ map) at every step of the scheduler. The strength of the projected food source stimuli (in arbitrary units) can be adjusted with a weighting parameter. When a real plasmodium engulfs a food source the stimulus for diffusion is reduced by the encapsulation. This is approximated in the model by damping the food stimulus projection values if the site is occupied by an agent. The food stimuli are diffused by means of a simple 3×3 mean filter kernel. The combined diffusion stimuli and agent trails stored in the chemoattractant map are degraded at every scheduler step by 0.01 to maintain a relatively steep diffusion gradient away from the food source. The diffusion gradient corresponds to the quality of the nutrient and substrate of the plasmodium’s environment (for example the different growth patterns seen in soaked filter paper and agar substrates), and differences in the stimulus strength, stimulus area, affect both the steepness, and propagation distance of the diffusion gradient and affect the growth patterns of the synthetic plasmodium.

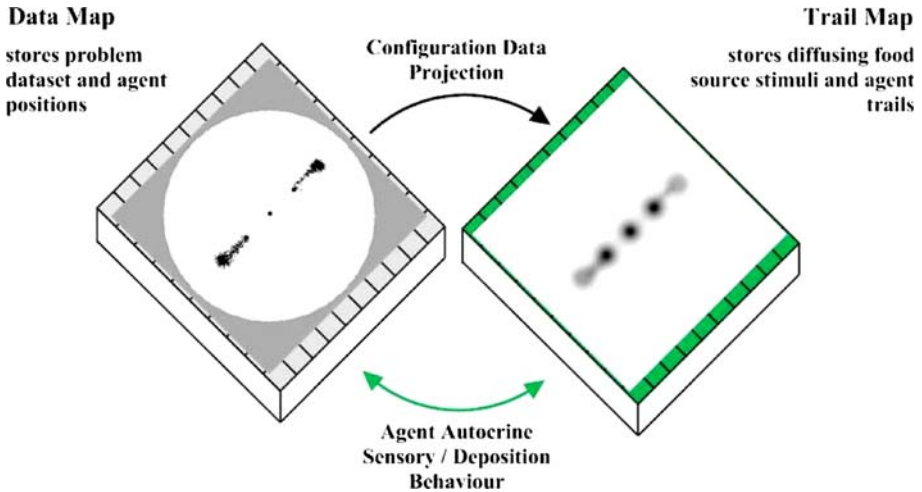


Fig. 3 Problem data projected to coupled chemoattractant map storing diffusing food source stimuli and agent movement trails

The agent population (size 2000 agents for most experiments, in the last case we employed 3200 agents) is initialised with random agent positions (on a lattice of 200×200 sites) and random agent orientations. To reproduce the ‘wet’ *Physarum* experiments agents were inoculated at particular locations (for example food sources) and new agents are initialised at the front of plasmodium growth according to local occupancy measures. These measures are based on a local window surrounding each agent (θ_w), a minimum occupancy threshold (θ_{min}), and a maximum occupancy threshold (θ_{max}). In these results $\theta_w = 3 \times 3$ window, $\theta_{min} > 0$ and $\theta_{max} < 4$. In the current version of the model the maximum population size is manually assigned, based on a proportion of the size of the problem configuration area. Die back of population size is not incorporated into the current version of the model (in the real organism the plasmodium grows, subject to resource availability, to encompass the substrate). Work is in progress to automatically increase and decrease the population size dynamically in relation to resource availability.

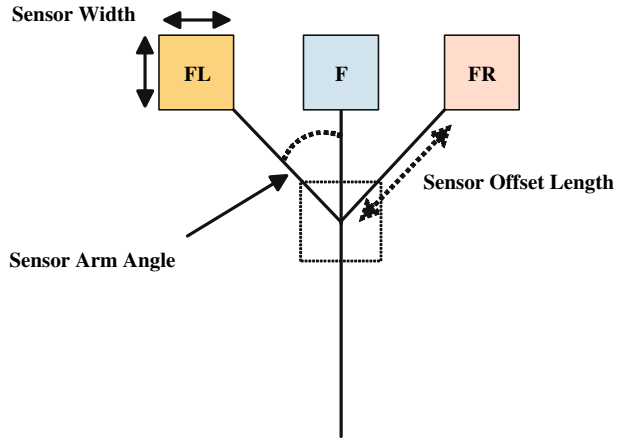
The morphology of each agent is shown in Fig. 4. Let l, s, r be concentrations of chemoattractants as measured by left (FL in Fig. 4), forward (F in Fig. 4, and right (FR in Fig. 4) sensors. Each agent can be characterised at time step t by its current position x^t and the angle α^t the agent faces at this time step. At each step of the simulation the angle of the agent α is calculated as follows:

$$\alpha^{t+1} = \alpha^t + \begin{cases} 0, & \text{if } s^t > l^t \text{ and } s^t > r^t \\ -\psi, & \text{if } l^t > r^t \text{ and } l^t > s^t \\ \psi, & \text{if } r^t > l^t \text{ and } r^t > s^t \\ \text{random}(-\alpha, \alpha), & \text{otherwise} \end{cases}$$

Agent’s position is updated as follows:

$$x^{t+1} = \begin{cases} \mathbf{R}(\alpha) \circ \mathbf{T}(\delta_x), & \text{if } \gamma(\mathbf{R}(\alpha) \circ \mathbf{T}(\delta_x)) = 0 \\ \mathbf{R}(\text{random}(360)), & \text{otherwise} \end{cases}$$

Fig. 4 Morphology of agent particle



where $\mathbf{R}(\alpha)$ is rotation on angle α operation and $\mathbf{T}(\delta_x)$ is a translation on distance δ_x , and for any Euclidean point y $\gamma(y) = 1$ if there is an agent residing at site y ; $\gamma(y) = 0$ otherwise.

The agents sense the concentration of stimuli in the chemoattractant map from three forward oriented sensors. At each scheduler step the agents orient themselves towards the strongest chemoattractant source by rotating left or right, depending on the source of the stimuli.

There is significant interplay between the sensor arm angle (β) and the rotation angle (ψ) parameters. When the two parameters are equal (for example, both 45°), the effect on network formation is to contract the initial network that emerges. If $\psi < \beta$ the contraction behaviour is increased. If, however, $\psi > \beta$ then spontaneous branching will appear during the network formation. The branching occurs because the larger rotation angle places an agent’s sensory apparatus outside the zone of the chemoattractant trail that the agent was following. The ψ/β parameters can be used to tune the network behaviour towards minimisation or towards high connectivity. For these simulations sensor width is one pixel, sensor arm angle $\beta = 30^\circ$ (Fig. 4), $\psi = 45^\circ$, sensor offset distance is nine pixels, and the distance moved per step, δ_x is one pixel. After the sensory stage every agent attempts to move forward in its current direction (represented by an internal state from 0° to 360°). If the new site is not occupied, the agent moves to the new site and deposits chemoattractant trail onto the chemoattractant map at the new location (deposition value 5 units). If the agent cannot move forwards successfully then no deposition is made to the chemoattractant map.

Inertial movement for each agent is provided by maintaining a floating point representation of the current position, as well as the discrete position corresponding to the image structure. This effectively allows the agents to ‘slide past’ one another even when the next cell is occupied (the occupation can only actually happen when a cell becomes free). The inertial behaviour results in the emergence of surging movements in the population and corresponds to the spatial oscillations seen in *Physarum*. The strength of the oscillations can be reduced by firing a ‘change direction’ event with probability 0.05 (in these results). When the event is triggered for an agent, the floating point position is restored to the discrete position and a new direction is randomly selected. The effect is to dampen the surge of movement caused by the inertial behaviour. The agent population is updated in a random order for both the sensory and movement stages to avoid any influence from sequential positional updates.

The experiments on the model plasmodium were designed and configured to reflect the ‘wet’ experiments reported in this report. The results images show the experimental field within the confines of the ‘dish’. The food sources correspond to the oat flake positions used in the ‘wet’ experiments and are indicated as small dark spots. The position of the plasmodium is indicated as the rough textured masses. Evolution of the model for each experiments proceeds from left to right.

5 Results: manipulating active zones

We present experimental operations with active zones. Schematic examples of fusion and multiplication operations are shown in Fig. 5. When two active zones A_1 and A_2 fuse they can either produce new active zone A_3 (Fig. 5a) or just protoplasmic tube (Fig. 5b). Multiplication of active zones is another basic operations: the active zone A_1 splits into two independent active zones A_2 and A_3 propagating along their own trajectories (Fig. 5c). In some cases active zones can be generated from “empty active zones”, oat flakes populated by plasmodium (Fig. 5d).

The operation $FUSE(A_1, A_2)$, fusing of two active zones A_1 and A_2 can be implemented via collision of the active zones. An example is demonstrated in Fig. 6.

Five oat flakes were arranged in a line on wet filter paper. Two pieces of plasmodium were placed near the extreme flakes, one piece near South-West flake, another piece near North-East. The active zones are formed as pseudopodia and propagate towards the center of the chain (Fig. 6a, b). When the active zones A_1 and A_2 collide they fuse and annihilate, $FUSE(A_1, A_2) = \emptyset$ (Fig. 6c). Depending on the particular circumstances the new active zone (the result of fusing) may become inactive, transform to protoplasmic tubes, or remain active, $FUSE(A_1, A_2) = A$, and continue propagation in a new direction.

The model output (Fig. 7) shows the inoculation and multiplication of the plasmodium at the outer nodes. The growing plasmodium is attracted to the inner food sources and the propagation continues inwards from each direction. The plasmodium fuses and maintains its surface area spanning the array of nodes. The model plasmodium was seen to periodically oscillate its position over the array whilst maintaining coverage.

Multiplication of an active zone, the operation $MULT(A) = \{A_1, A_2\}$, can be implemented by placing sources of nutrients nearby the protoplasmic tubes, or inactive zones (Fig. 8). An example is shown in Fig. 8. A chain of oat flakes is connected by protoplasmic veins, we add two new oat flakes to evoke new active zones (Fig. 8a). Ten hours later two active zones A_1 and A_2 are formed, each pseudopodium travels to its unique oat flake (Fig. 8b).

The model output initially recreates the spanning tree from the outer nodes, using the method in (Fig. 7). Two more food sources are added to either side of the array and the plasmodium sends two streams outwards to engulf the sources (Fig. 9). When the food sources have been engulfed, the plasmodium shifts in position by redistributing its component parts to cover the (diamond shaped) area created by the addition of the two new nodes.

In some cases we may need to translate the active zone not to another graph node (source of nutrients) but to a domain of an active space. Moreover, it may be necessary to provide the active zone A with certain initial velocity vector \mathbf{v} to the zone continues its propagation in the predetermined direction. Such operation $DIRECT(A, \mathbf{v})$ is executed using additional sources of nutrients (Fig. 10). Given a chain of oat flakes connected by protoplasmic tubes we added two new flakes on the west of the chain (Fig. 10a). An active

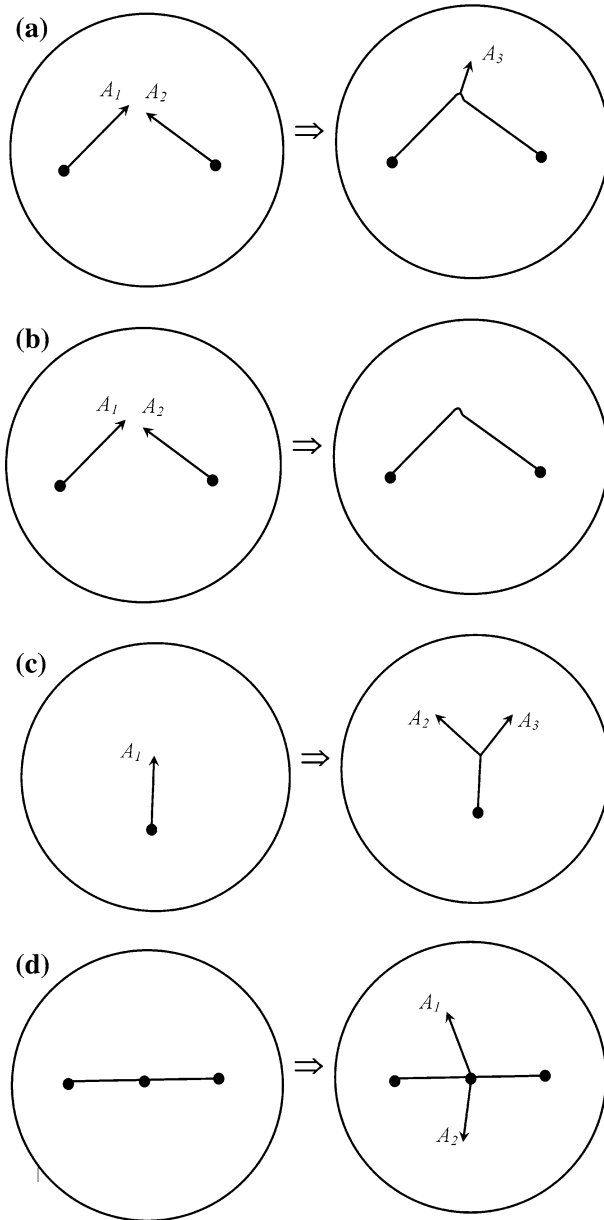


Fig. 5 Schematics of operation with active zones: **a** $FUSE(A_1, A_2) = \emptyset$, **b** $FUSE(A_1, A_2) = A_3$, **c** $MULT(A_1) = \{A_2, A_3\}$, **d** $MULT(\emptyset) = \{A_1, A_2\}$

pseudopodium, representing the active zone A sprouted from the ‘old’ flake lying between projections of the new flakes. It continues propagating along the bisector separating two new flakes. The active zone continues its propagation towards West till collide with the wall of the Petri dish. Meantime, two more active zones are formed to connect new oat flakes to the existing protoplasmic network (Fig. 10b).

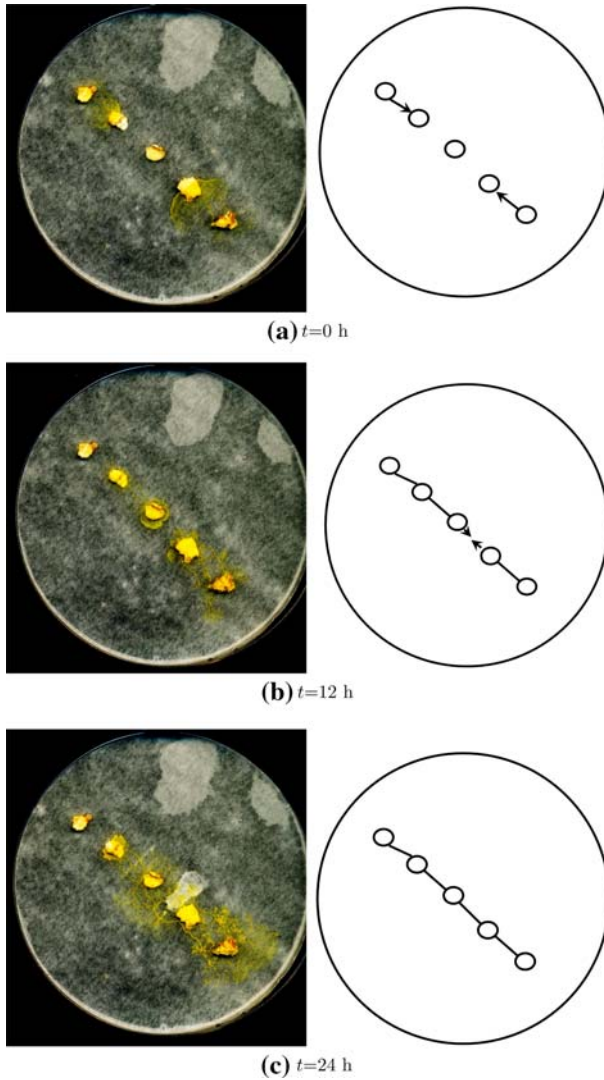


Fig. 6 Fusing of active zones. Protoplasmic tubes are shown as *lines* and active zones as *arrows* in the accompanying diagrams

The model was not able replicate the experimental results of Fig. 10. As soon as a food stimulus is removed from the environment the model plasmodium begins to shrink. We speculate that the agent population is responding too quickly to the changing environment, and so the inertia of the agent population (in the current version of the model) is not as strong as observed in the real plasmodium. The model of *Physarum* machine requires some direct stimulus to move the plasmodium active zone. If it is not possible to use food source stimuli to move the active zone then it may be possible to use a synthetic hazard (such as the response of real plasmodia to certain wavelengths of visible light) to drive the active front to a new position by repulsion.

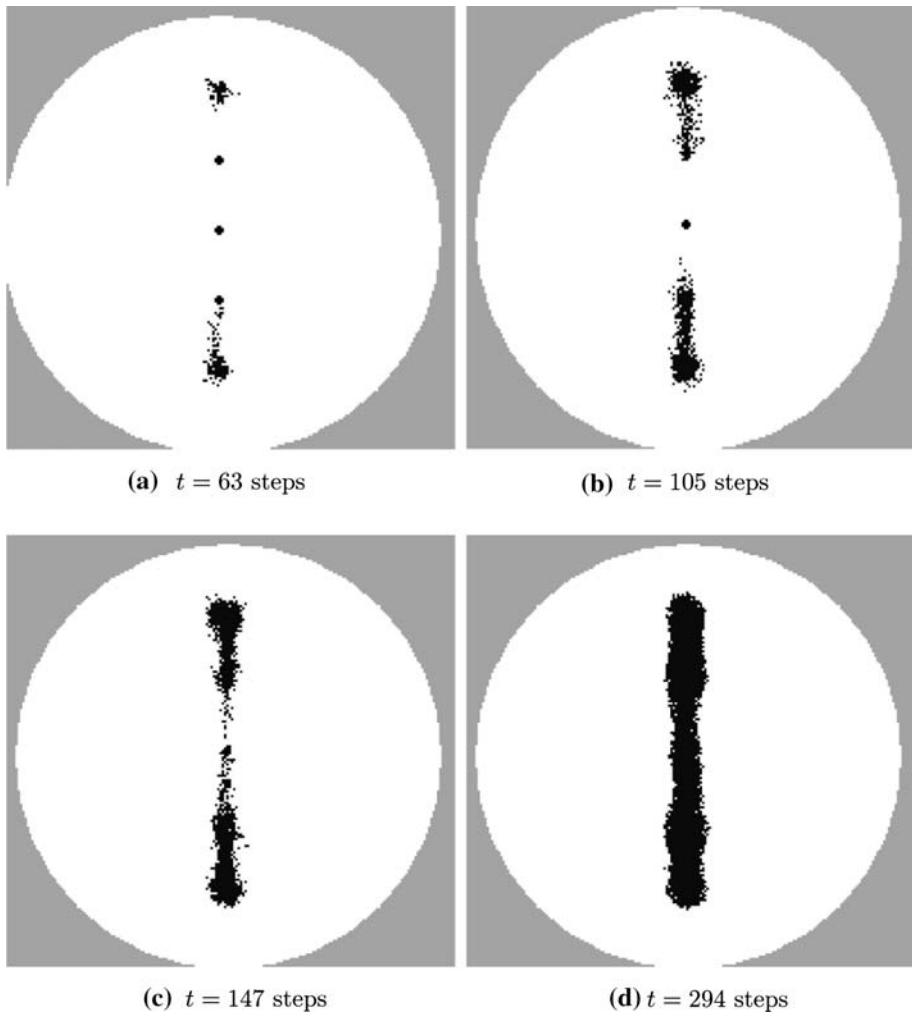


Fig. 7 Evolution of model, simulating fusing of active zones

Translation of active zone may lead to, or used for, undertaking of the major restructuring of the data storage graph. Thus, a parallel partial shift of the graph chain is demonstrated in Fig. 11. Three new oat flakes are places alongside the chain of oat flakes already connected by protoplasmic tubes (Fig. 11a). In few hours the plasmodium occupies new oat flakes. Moreover, part of the protoplasmic graph coaligned with new oat flakes is shifted to new oat flakes (Fig. 11b). Abandoned protoplasmic veins, former edges of the shift part, are visible as white tubes.

Such a major restructuring of the graph was caused by relocation of the active zone. At first the active zone travelled North-East (Fig. 11a). Addition of new oat flakes caused the zone to switch to the new location and move South-West (Fig. 11b).

A major restructuring was also observed in experiments with the particle model. Note that the particle model exhibits distinct differences to the actual organism. The graph restructuring with the real plasmodium is affected by the remnants of old protoplasmic

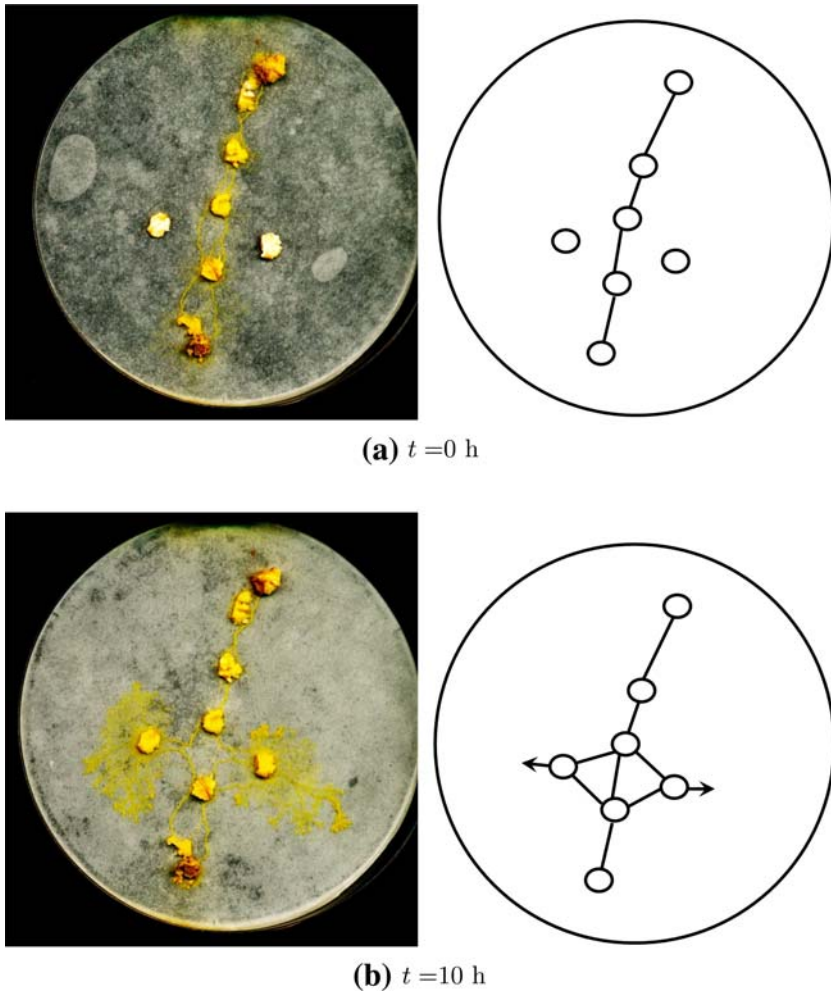


Fig. 8 Multiplication of active zone. Protoplasmic tubes are shown as *lines* and active zones as *arrows* in the accompanying diagrams

veins. Such debris is not present in the model (primarily since the ‘veins’ in the model merely consist of protoplasmic flow itself and have no structural features). The shift in active front position is dramatically illustrated in Fig. 12a where the front moves from its central position to engulf the three new nodes on the left. Figure 12b continues the same experiment, deleting the three leftmost nodes and adding three new right-side nodes, resulting in a significant surge of the active front to the right side of the ‘dish’.

In many examples above we observed the formation of protoplasmic tubes, connecting two geographically closest sources of nutrients. Given a chain of oat flakes, an additional oat flake is added to the experimental container. What is the exact mechanism of inclusion the new flake in the graph? Do all nodes of the graph develop active zones, which travel to the flake while competing with each other for the flake?

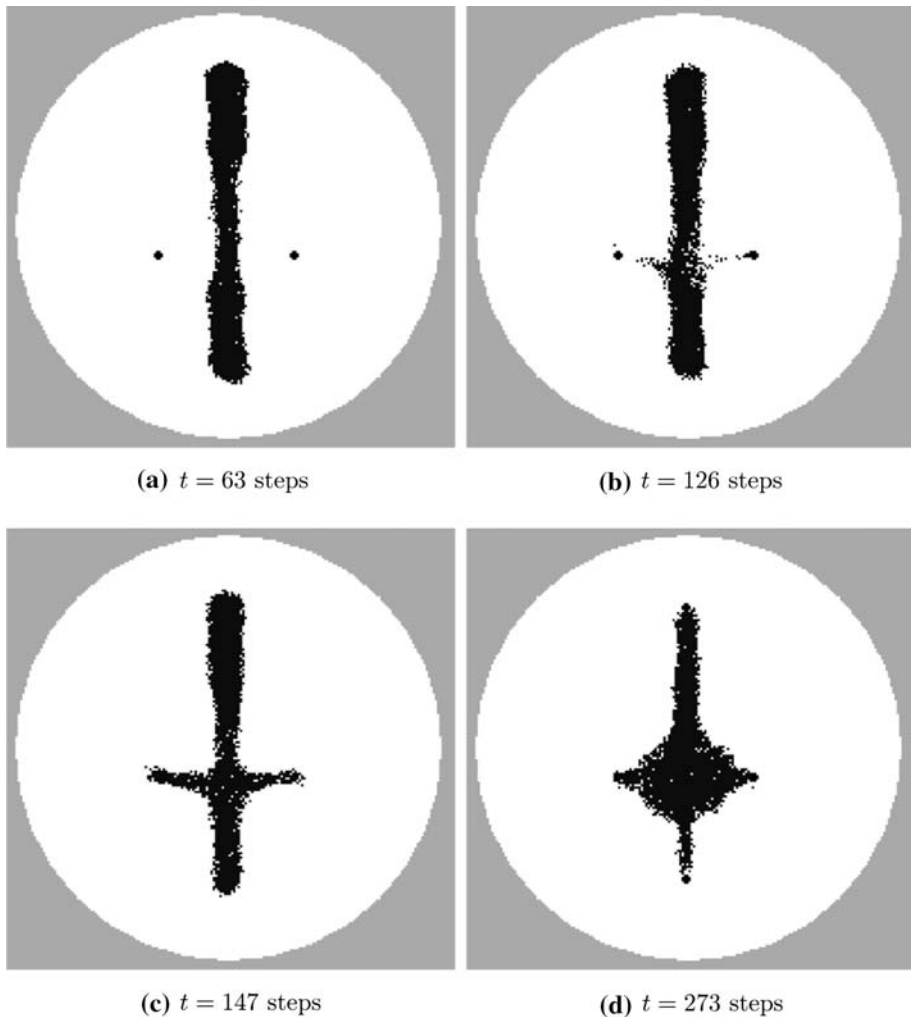


Fig. 9 Evolution of model, simulating multiplication of active zones. New active zones A_1 and A_2 are visible as patterns propagating East and West

Our experimental observations show that if the graph remains connected then only one active zone, heading for the new oat flake, emerges. This happens due to synchronization of activity in the whole protoplasmic network.

This may indicate that plasmodium first ‘decides’ which part of the protoplasmic graph is closest to the recently added source of nutrients and only then generates an active zone. All parts of the *Physarum* machine sense the chemo-attractants coming from the new source of nutrients, however the node closest to the new source somehow suppresses, or inhibits, activity of other nodes. Existence of connections between nodes of the *Physarum* machine is a pre-requisite for the inhibition.

To prove the point we arranged oat flakes in a chain and allowed the chain to be spanned by the plasmodium (Fig. 13a). When the flakes are connected by protoplasmic tubes we add a new oat flake, the eastmost flake on the picture (Fig. 13a). To break communication

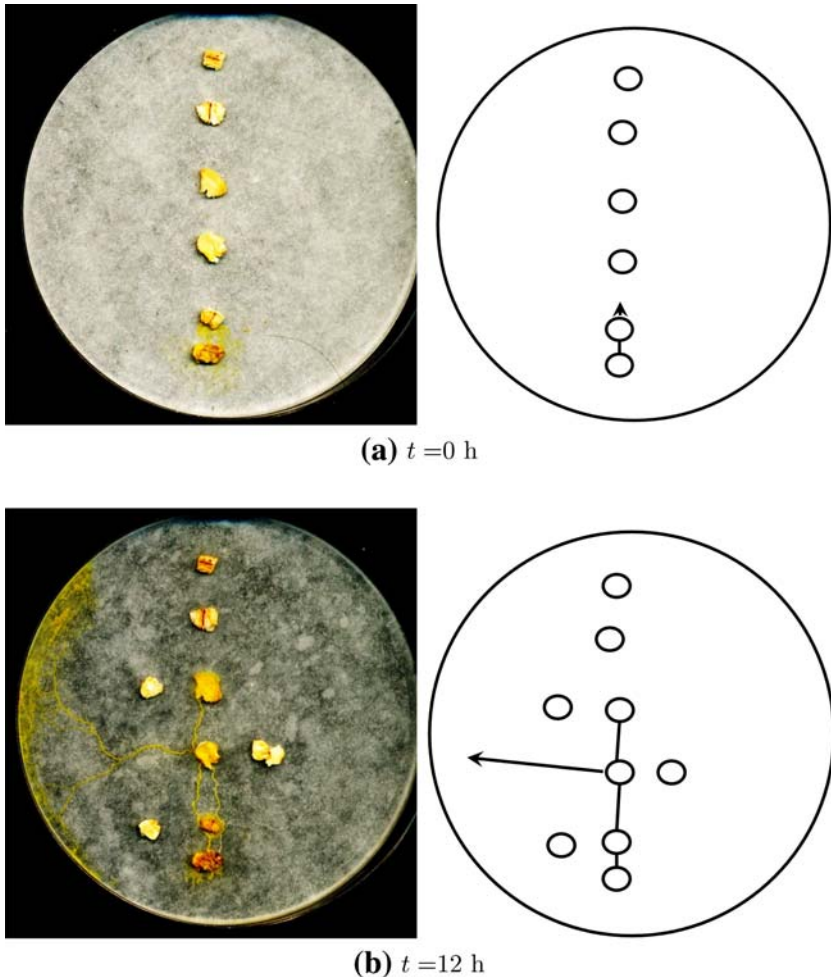


Fig. 10 Illustration of how active zone can be relocated to a domain without food sources, operation DIRECT(A, \mathbf{v}). Protoplasmic tubes are shown as *lines* and active zones as *arrows* in the accompanying diagrams. Vector \mathbf{v} is shown by *Westward pointing arrow* in the diagram **b**

between nodes of the *Physarum* machine we cut through protoplasmic tubes connecting three Northern nodes, (cut places are shown by lines in Fig. 13a). Due to breakup in communication all nodes of the plasmodium react, almost simultaneously, to the addition of new source of nutrients by sprouting pseudopodia (Fig. 13b). In a few hours cut tubes are restored (Fig. 13c). This reinstates communication between all parts of the *Physarum* machine. The distant, from the new flake, node of the *Physarum* machine de-activate their active zones (cease propagation of pseudopodia). In the result of restored communication only one active zone remains, and the new node is connected to the storage graph by a single edge.

Just cutting protoplasmic tubes does not lead to formation of new pseudopodia (Fig. 14). When tubes are cut they do usually fuse back again in few hours without any adverse effect to the plasmodium behaviour.

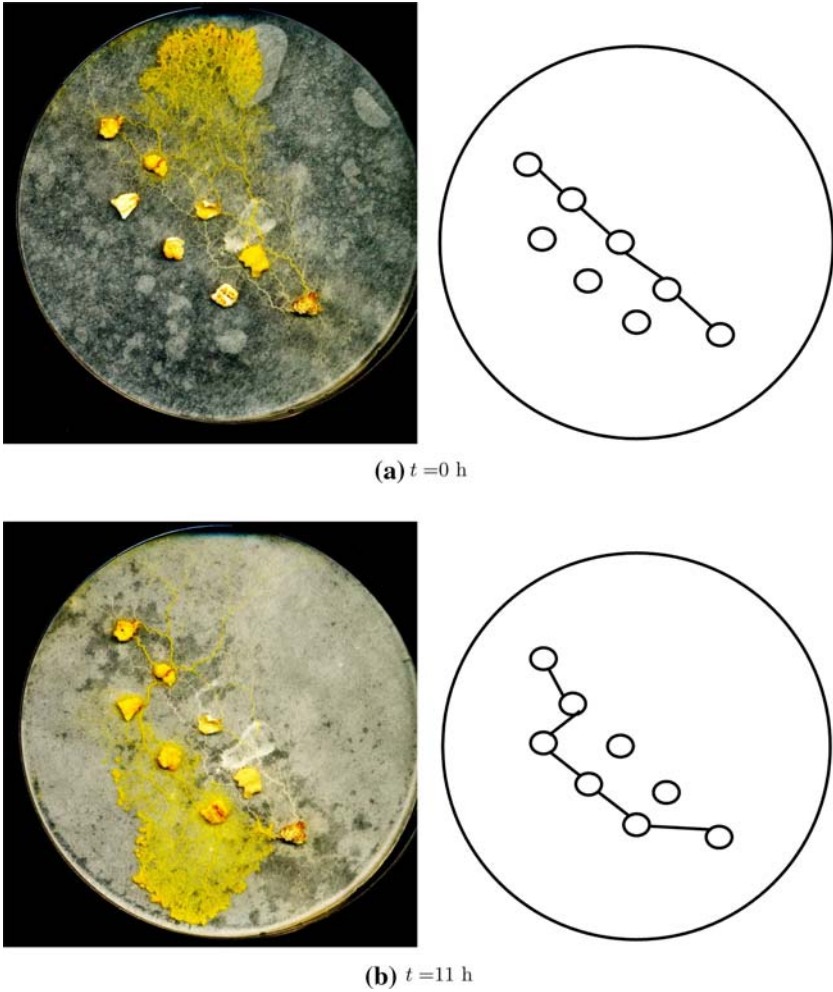


Fig. 11 Major restructuring of the graph. Protoplasmic tubes are shown as *lines* and active zones as *arrows* in the accompanying diagrams

The model plasmodium was not able to replicate the experiment of Fig. 13, once again suggesting that the model is currently lacking in the mechanism which couples the plasmodium behaviour across food sources.

6 Summary

We studied the *Physarum* machine, an implementation of general-purpose Kolmogorov–Uspensky machine (KUM) in vegetative stage, plasmodium, of *Physarum polycephalum*. Our present paper is mainly concerned with manipulating active zones (actively growing pseudopodia of the plasmodium) because they are the main computational units of the storage modification machine, implemented in the plasmodium. In laboratory experiments

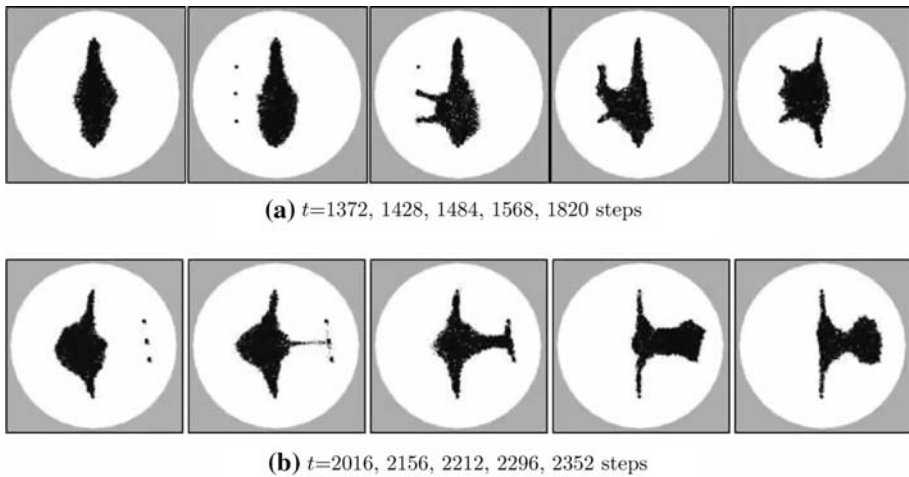


Fig. 12 Major restructuring of the graph in the particle model. **a** Active front shifts towards new stimuli on the left. **b** Active front shifts towards new stimuli on the right

and computer simulation of the plasmodium we executed basic operations with active zones. We shown how merge two active zones, to multiply an active zone, to translate an active zone to a new node of the storage structure, to direct an active zone in the space not occupied by data nodes. We envisage our experimental and theoretical findings will be employed in future programming of spatially distributed biological computers.

As with any new computing substrate, the reliability of the low-level implementations must first be ensured so that they may be used to deploy higher level functionality. The notion of reliability depends on both correctness and repeatability—does the organism perform correctly, and does it do so every time? The problem with applying the correctness and repeatability criteria to a living organism is that a surviving plasmodium always manages to forage and minimise the transport between available food sources—its criteria is survival rather than the arbitrarily and externally applied task. However in the example case of construction of efficient transport networks the final ‘performance’ metric appears to be a trade-off between minimal distance and tolerance of network disconnections (see Nakagaki et al. 2004, for example), and the actual visible result (the morphology of the plasmodium) may visually take on different forms. Experiments with the simple model discussed in this paper also suggest that even when the emergent behavioral patterns do converge on identical solutions, they may do so through very different trajectories. It is clear that the external computational task assigned to the biological substrate should be well matched to the natural modes of the organism.

Assuming that a good computational match is made between the biological substrate (*Physarum*) and chosen task (mechanisms of KUM), the most critical part of reliably implementing the low level commands is an accurate control of the propagation of the active zone. Since *Physarum* is a living organism, it is not possible to force the computing substrate to extremes of activity, as is with the case for silicon substrates (where the logical digital signals are derived from the extremes of activity of essentially analogue transistor behaviour). Indeed forcing extreme external conditions upon *Physarum* results in at worst death, or at best uncontrolled growth or reversion to the dormant sclerotium stage. The spatially extended nature of the KUM also appears to rule out the fixed encoding of

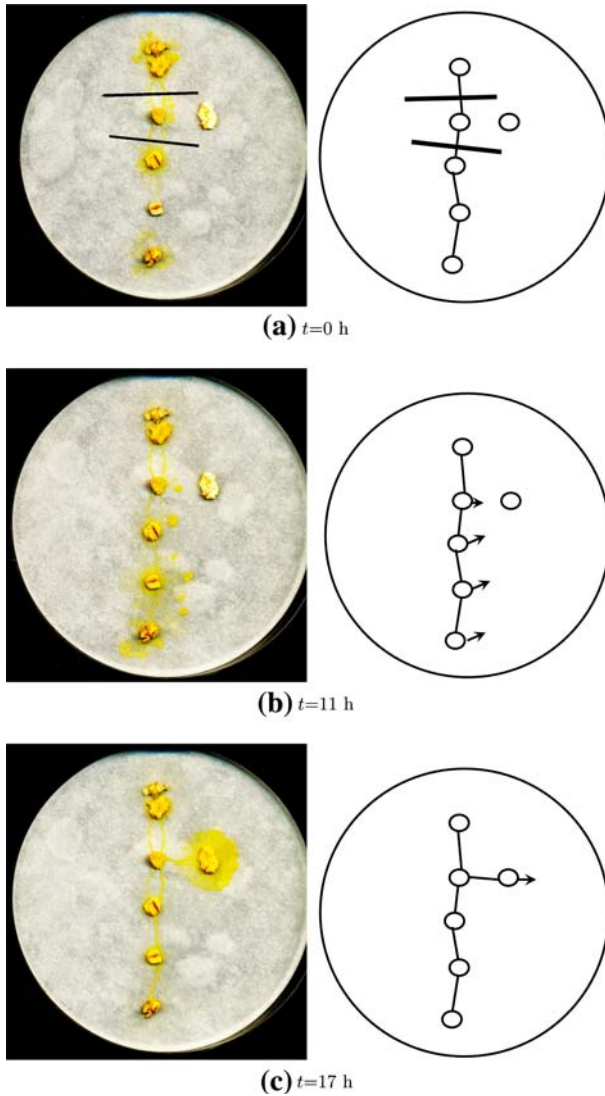


Fig. 13 Cutting edge leads up to de-synchronization of the *Physarum* machine. Protoplasmic tubes are shown as *lines* and active zones as *arrows* in the accompanying diagrams

plasmodium signal propagation channels, such as those employed by Tsuda et al. in (2004).

If it is not possible to force or channel the plasmodium growth, experimental results do suggest that *Physarum* can be coaxed into desired morphological configurations by means of less extreme positive and negative stimuli. Positive stimuli (stimuli which stimulate growth or movement) include chemotactic food gradients or warmth gradients and may be said to act as the presence or absence of a pull impetus (+ or 0) towards the stimulus. Negative stimuli (inhibiting growth or movement) include exposure to certain wavelengths of visible light (Nakagaki et al. 1999, 2007) and some simple chemical compounds

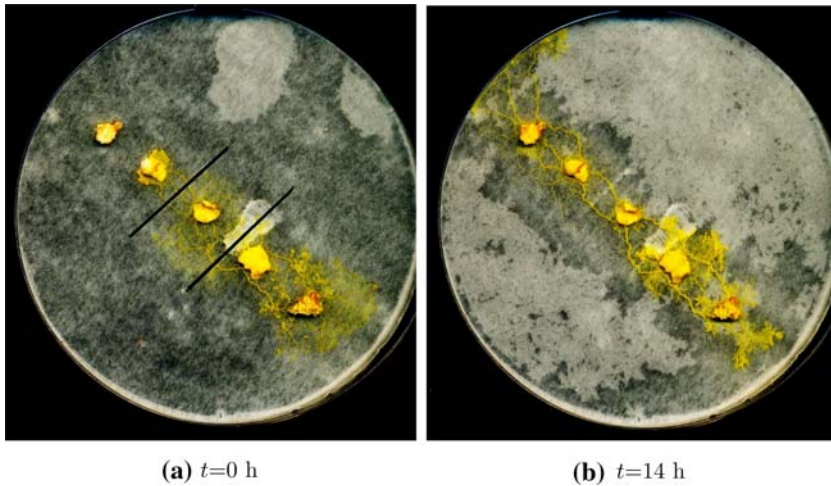


Fig. 14 Absence of adverse reaction to tube cutting

(Shirakawa and Gunji 2007), acting as the presence or absence of a push impetus ($-$ or 0) away from the stimulus. Since the KUM disallows fixed encoding of signal channels the most suitable candidates for reliable control of the low-level commands would appear to be those that can be temporarily applied.

Temporary application of positive and negative guiding stimuli suggests a means of sensory feedback to govern the application of the stimuli, such as the light masking employed by Aono and Hara in (Aono and Hara 2007). It is envisaged that such mechanisms of gentle guiding and coaxing to control the plasmodium would avoid driving the organism beyond the extremes of its operation, yet provide reliable methods of control of the active front. The application of such fine control of the active front is beyond the scope of this proof-of-concept investigation but suggests possible mechanisms for future implementations of *Physarum* machines.

References

- Adamatzky A (2007a) *Physarum* machine: implementation of a Kolmogorov–Uspensky machine on a biological substrate. *Parallel Process Lett* 17:455–467
- Adamatzky A (2007b) *Physarum* machines: encapsulating reaction-diffusion to compute spanning tree. *Naturwissenschaften* 94:975–980
- Adamatzky A (2008a) Growing spanning trees in plasmodium machines. *Kybernetes: Int J Syst Cybern* 37: 258–264
- Adamatzky A (2008b) Developing proximity graphs by *Physarum polycephalum*: does the plasmodium follow the Toussaint hierarchy? *Parallel Process Lett* 19:105–127
- Adamatzky A, De Lacy Costello B, Asai T (2005) *Reaction-diffusion computers*. Elsevier, Amsterdam
- Adamatzky A, De Lacy Costello B, Shirakawa T (2008) Universal computation with limited resources: Belousov-Zhabotinsky and *Physarum* computers. *Int J Bifurc Chaos* 18:2373–2389
- Aono M, Hara M (2007) Amoeba-based nonequilibrium neurocomputer utilizing fluctuations and instability. In: *Lecture Notes in Computer Science* 4618, p 41
- Blass A, Gurevich Y (2003) Algorithms: a quest for absolute definitions. *Bull Eur Assoc TCS* 81:195–225
- Jones J (2008, in press) The emergence and dynamical evolution of complex transport networks from simple low-level behaviours. *Int J Unconv Comput*

- Knuth DE (1968) The art of computer programming, vol. 1: fundamental algorithms. Addison-Wesley, Reading
- Kolmogorov AN (1953) On the concept of algorithm. *Uspekhi Mat Nauk* 8(4):175–176
- Nakagaki T (2001) Smart behavior of true slime mold in a labyrinth. *Res Microbiol* 152:767–770
- Nakagaki T, Yamada H, Ueda T (1999) Modulation of cellular rhythm and photoavoidance by oscillatory irradiation in the *Physarum* plasmodium. *Biophys Chem* 82:23–28
- Nakagaki T, Yamada H, Ueda T (2000) Interaction between cell shape and contraction pattern in the *Physarum plasmodium*. *Biophys Chem* 84:195–204
- Nakagaki T, Yamada H, Toth A (2001) Path finding by tube morphogenesis in an amoeboid organism. *Biophys Chem* 92: 47–52
- Nakagaki T, Kobayashi R, Nishiura Y, Ueda T (2004) Obtaining multiple separate food sources: behavioural intelligence in the *Physarum plasmodium*. *Proc Roy Soc B* 271:2305–2310
- Nakagaki T, Makoto I, Ueda, T, Nishiura T, Saigusa T, Tero A, Kobayashi R, Showalter K (2007) Minimum-risk path finding by an adaptive amoebal network. *Phys Rev Lett* 99:68104
- Schönhage A (1973) Real-time simulation of multi-dimensional Turing machines by storage modification machines. Project MAC Technical Memorandum 37. MIT
- Schönhage A (1980) Storage modification machines. *SIAM J. Comput* 9:490–508
- Shirakawa T, Gunji Y (2007) Emergence of morphological order in the network formation of *Physarum polycephalum*. *Biophys Chem* 128:253–260
- Shirakawa T, Adamatzky A, Gunji Y-P, Miyake Y (2008, submitted) On simultaneous construction of Voronoi diagram and Delaunay triangulation by *Physarum polycephalum*
- Takamatsu A (2007) Mobiligence in an amoeboid cell, plasmodium of *Physarum polycephalum*. In: Second international symposium on mobiligence. Awaji, Japan, pp 48–51
- Tarjan RE (1977) Reference machines require non-linear time to maintain disjoint sets. STAN-CS-77-603, March 1977
- Tsuda S, Aono M, Gunji Y (2004) Robust and emergent *Physarum* logical-computing. *BioSystems* 73: 45–55
- Uspensky VA (1992) Kolmogorov and mathematical logic. *J Symb Logic* 57: 385–412