Naturwissenschaften (2007) 94:975–980
DOI 10.1007/s00114-007-0276-5

ORIGINAL PAPER

Physarum machines: encapsulating reaction–diffusion to compute spanning tree

Andrew Adamatzky

Received: 25 January 2007 / Revised: 23 April 2007 / Accepted: 21 May 2007 / Published online: 29 June 2007

Abstract The Physarum machine is a biological computing device, which employs plasmodium of *Physarum polycephalum* as an unconventional computing substrate. A reaction–diffusion computer is a chemical computing device that computes by propagating diffusive or excitation wave fronts. Reaction–diffusion computers, despite being computationally universal machines, are unable to construct certain classes of proximity graphs without the assistance of an external computing device. I demonstrate that the problem can be solved if the reaction–diffusion system is enclosed in a membrane with few 'growth points', sites guiding the pattern propagation. Experimental approximation of spanning trees by *P. polycephalum* slime mold demonstrates the feasibility of the approach. Findings provided advance theory of reaction–diffusion computation by enriching it with ideas of slime mold computation.

Keywords Cellular automata · Slime mold · Proximity graphs · Non-classical computation

Introduction

In reaction–diffusion computers (Adamatzky 2001; Adamatzky et al. 2005), data are presented by initial concentration profile or configuration of disturbance (e.g. sites of stimulation of excitable media), information is transfered by spreading wave patterns, computation is implemented in collisions of wave-fronts, and final concentration profile represents results of the computation. Reaction–diffusion computers are theoretically and experimentally proven to be capable for quite sophisticated computational tasks, including image processing, computational geometry, logics, and arithmetics, and robot control (Adamatzky et al. 2005). However, there still remains a range of problems where chemical processors cannot cope without external support. Shortest path is one such problem. One can use excitable medium to outline a set of all collision-free paths in a space with obstacles (Adamatzky and De Lacy Costello 2002), but to select and visualise the shortest path amongst all those that are possible, one needs to use an external cellular-automaton processor or conceptually supply the excitable chemical medium with some kind of field of local pointers (Adamatzky and De Lacy Costello 2002). Experimental setups (Agladze et al. 1997; Steinbock et al. 1995) which claim to directly compute a shortest path in chemical media are indeed employing external computing resources to store time-lapsed snapshots of propagating wave-fronts and to calculate the intersection of wave fronts. Such usage of external resources dramatically reduces the fundamental values of the computing with propagating patterns.

Graph-theoretical computations pose even more difficulties for spatially extended non-linear computers. For example, one can compute a Voronoi diagram of a planar set but cannot invert this diagram (Adamatzky et al. 2005). Let us consider a spanning tree, the most

A. Adamatzky (✉)
Faculty of Computing, Engineering and Mathematical Sciences, University of the West of England,
Bristol, BS16 1QY, UK
e-mail: andrew.adamatzky@uwe.ac.uk

famous of classical proximity graphs. Given a set of planar points, one wants to connect the points with edges such that the resultant graph has no cycles and there is a path between any two points of the set. So far, no algorithms of spanning tree construction have been experimentally implemented in spatially extended nonlinear systems. This is caused mainly by uniformity of spreading wave-fronts, their inability to sharply select directions toward locations of data points, and the fact that excitable systems usually do not form stationary structures or standing waves.

The present paper recurs to the author's early algorithm of a computing spanning tree by an abstract growing neuroblast (Adamatzky 1991) and speculates on how such an algorithm can be implemented in excitable systems with retained excitation rules

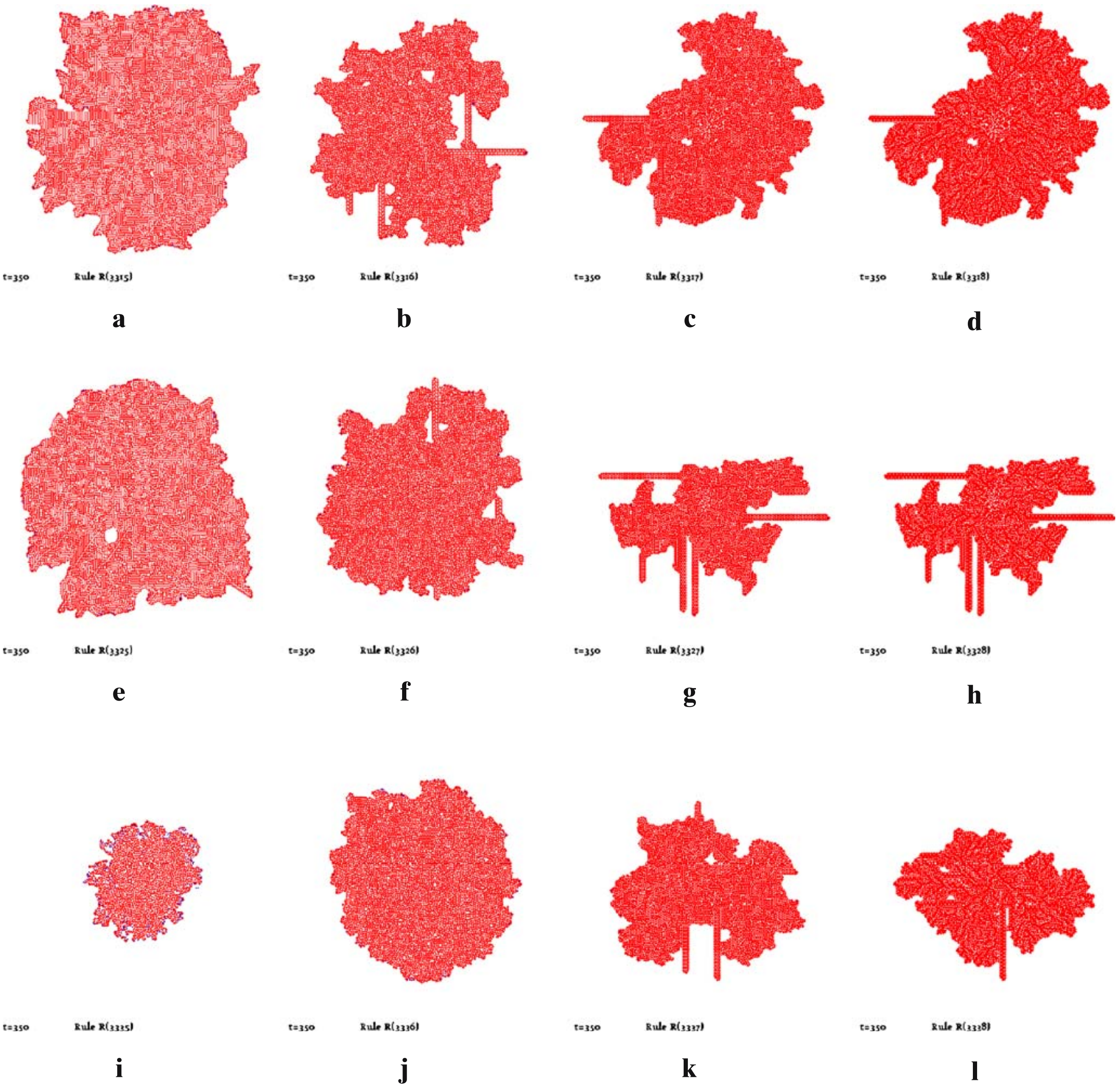

Fig. 1 Snapshots of excitation dynamics of cellular automaton with retained excitation, 300 × 300 cells, taken at the 350th step of development. A locus of initial stimulation, shown in **a**, is a disc radius with 20 cells, each cell inside the disc were assigned refractory state or excited state with probability 0.16 each. Snapshots are provided for each rule of the domain: **a** $R(3315)$, **b** $R(3316)$, **c** $R(3317)$, **d** $R(3318)$, **e** $R(3325)$, **f** $R(3326)$, **g** $R(3327)$, **h** $R(3328)$, **i** $R(3335)$, **j** $R(3336)$, **k** $R(3337)$, **l** $R(3338)$

(Sect. Amoeboid patterns in excitable mediu). I demonstrate the experimental computing of a spanning tree by *Physarum polycephalum* slime mold (Sect. Physarum machines), which is a biological analog of a reaction–diffusion system enclosed in a membrane. Further directions of research are discussed in Sect. Conclusion.

Amoeboid patterns in excitable mediu

Is it possible to imitate bounded and non-uniformly propagating patterns similar to that exhibited by amoeboid creatures? Let us consider the model of retained excitation (Adamatzky 2007). Every cell of a two-dimensional array has eight neighbours and takes three states: resting, excited, and refractory. All cells update their states simultaneously in discrete time and depending on the states of their closest neighbours. A resting cell excites if a number of excited neighbours belong to excitation interval $[\theta_1, \theta_2]$. An excited cell remains excited if a number of excited neighbours belong to excitation retention interval $[\delta_1, \delta_2]$. Each rule therefore can be described as a tuple $R(\theta_1\theta_2\delta_1\delta_2)$.

In exhaustive analysis of excitation rules (Adamatzky 2007), I located a sub-domain of rule space, $R(33\delta_1\delta_2)$, $\delta_1 \in [1, 3]$ and $\delta_2 \in [5, 8]$, the elements of which are responsible for non-uniform or directed growth of the excitation patterns. As shown in Fig. 1, initial circular domains of stimulation are transformed to unevenly growing shapes. Wave-fronts are split into pseudopodia-like growing domains and also more straightforwardly growing domains (as the ones in Fig. 1h,i); all propagate with different speeds.

In many rules, directed, or almost directed, growth of pseudopodia is guided by small tips consisting of excited and refractory states [analogs of growth cones, which were elementary processors in algorithm (Adamatzky 1991)]. We can see that the growth phenomena have an underlying tree-like structure, where branching structure depends on the exact rules governing the cellular automata. Increase of the upper boundary, δ_2, of excitation retention interval reduces

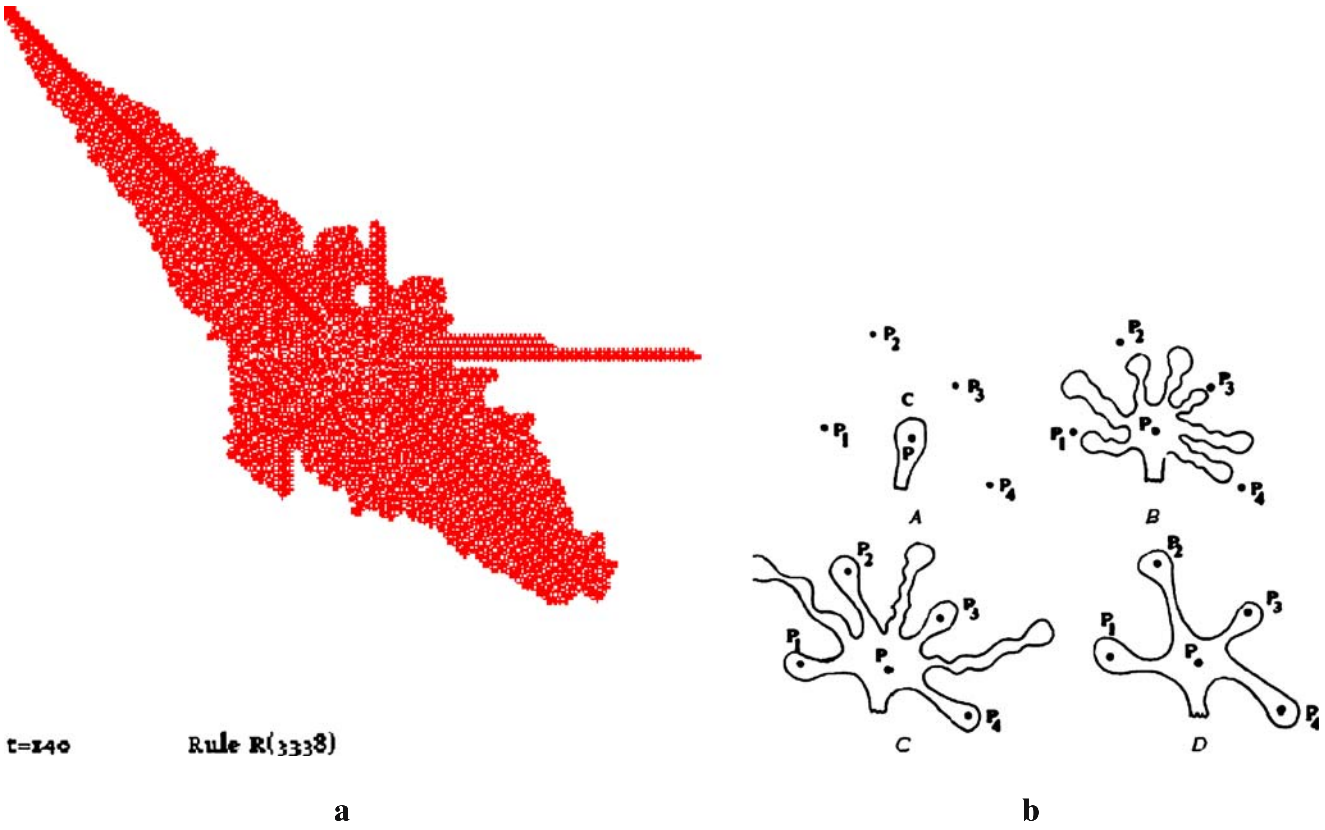

Fig. 2 Guided growth of amoeboid patterns. **a** Amoeboid excitation pattern in discrete model of retained excitation. Cells lying at the centre to north-west, centre to east and centre to south-east diagonals are always kept in excited states. Initial stimulation is a centered disc radius of five cells, where every cell takes a non-resting state with probability 0.3. Wave-fronts propagate with higher speeds along excited segments. **b** Scheme of original idea on spanning tree computation posed in 1991 (Adamatzky 1991)

branching, while increase of the lower boundary, δ_1, makes growing patterns more compact. In cases of very wide interval of excitation retention, $\delta_2 = 8$, the role of refractory states diminishes.

Containing and controlling excitation waves

I have demonstrated in the previous section that even extremely simple rules of local dynamics in discrete lattice can generate complex non-uniform slowly and selectively growing patterns. These models were rather analogous representations of amoeboid-like propagation of an excitable spatially extended system.

Can the excitation amoeboid pattern be guided to calculate a spanning tree, or at least, starting at some given point, to span neighbouring planar points in a manner outlined in Fig. 2b? Despite their sophisticated space-time dynamics, amoeboid patterns are barely controllable. Exact shape of the pseudopodia is determined by initial configuration of excited and refractory states in stimulation disc. We can indeed guide propagation of the wave fronts, as demonstrated in

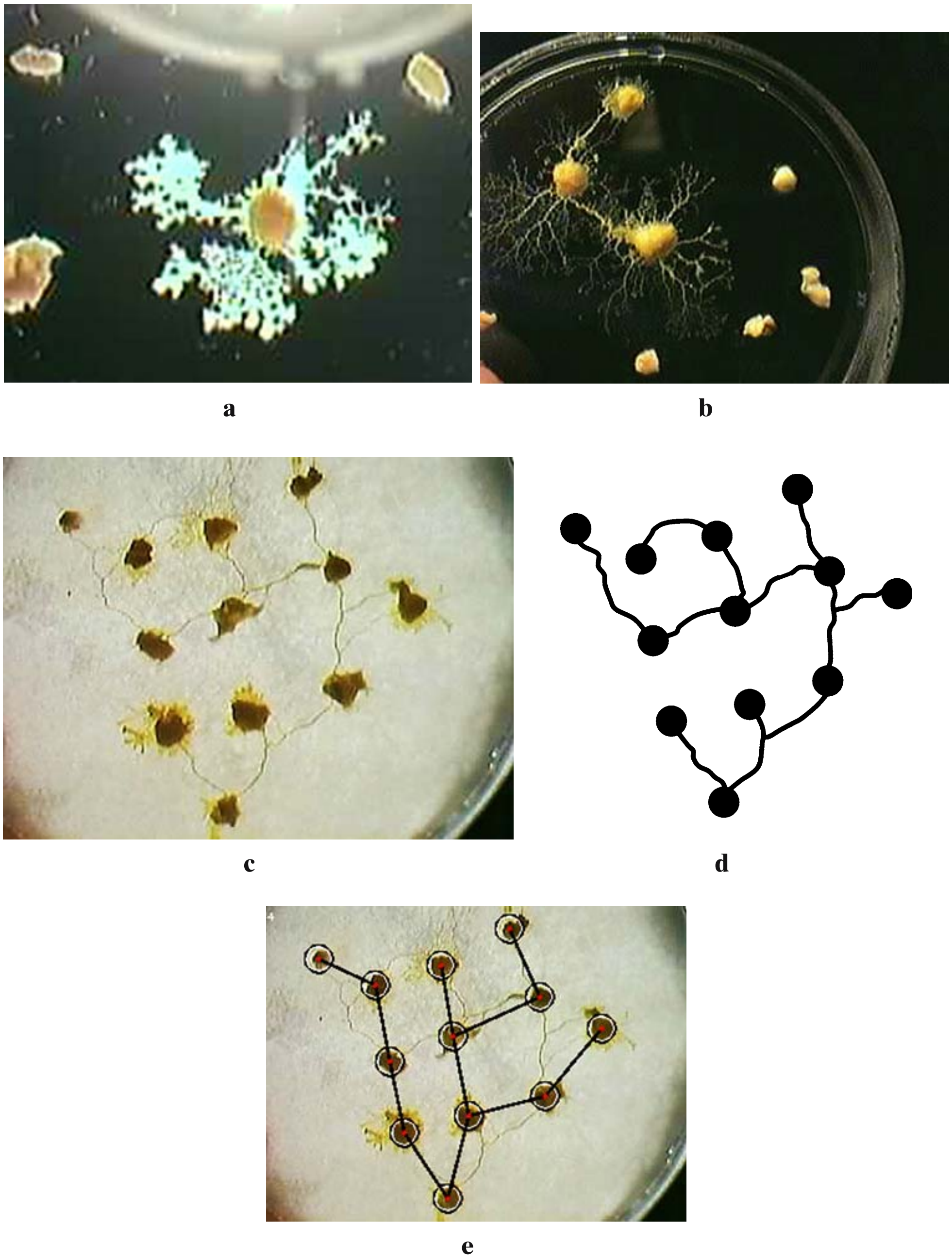

Fig. 3 Approximation of spanning trees by Physarum slime mold; data points are represented by oat flakes: **a** exploratory phase, first day of the experiment; **b** partly constructed tree, second day of the experiment; **c** photograph of slim mold constructed tree of 12 points, tree was constructed on the third day of the experiment; **d** highlighted tree; **e** spanning tree constructed by classical algorithms. Snapshots **a** and **b** show plasmodium grown on a bare container, no substrate, Petri dish, and **c** paper moistened with water. Snapshots have actual views

Fig. 2; however, it seems to be impossible at the present stage to span an arbitrary planar set of points with any kind of graph. This is because excitation patterns do not shrink: they can indeed cover a set of data points they do contain but do not display structure of the proximity graph.

The diffusive component of the growing patterns must be somehow inhibited. Encapsulating the reaction–diffusion system in a membrane would be a solution. As far as I am aware, there are no theoretical models nor experimental prototypes of chemical reaction–diffusion systems contained in controllable and growing membrane.

The ultimate goal of our studies in unconventional computer architectures is to build working real-world prototypes of non-classical computers working on chemical or biological substrates. To make a shortcut toward the goal, I have not been designing a computer model of an excitable reaction–diffusion system enclosed in membrane but instead found quite a close biological analog of such a system. This is a plasmodium of true slime mold *P. polycephalum*. Plasmodium is *P. polycephalum*, which is a single cell with many nuclei which behaves like an amoeba.

I believe *Physarum* plasmodium is an analog of an excitable reaction–diffusion system enclosed in a membrane. This proposition is based on the following findings: Growing and feeding plasmodium exhibit characteristic rhythmic contractions with articulated sources. The contraction waves are associated with waves of potential change, and the waves observed in plasmodium (Matsumoto et al. 1986, 1988; Yamada et al. 2006) are similar to those found in excitable chemical systems, like Belousov–Zhabotinsky medium. The following wave phenomena were discovered experimentally (Yamada et al. 2006): undisturbed propagation of contraction wave inside the cell body, collision and annihilation of contraction waves, splitting of the waves by inhomogeneity, and formation of spiral waves of contraction (see Fig. 6c–f in (Yamada et al. 2006)). These are closely matching dynamics of pattern propagation in excitable reaction–diffusion chemical systems.

Yamada et al. (Yamada et al. 2006) indicate the possibility for interaction between the contraction force generating system with oscillating chemical reactions of calcium, ATP and associated pH (Nakamura et al. 1982; Ogihara 1982; Ridgway and Durham 1976). Moreover, there are evidences that chemical oscillations are primary oscillations (i.e. contraction waves are guided by chemical oscillation waves) because chemical oscillations can be recorded in the absence of contractions (Nakagaki et al. 1999; Nakamura and Kamiya 1985; Oster and Odel 1984).

The plasmodium has already been proven to be a unique fruitful object to design various schemes of non-classical computation (Aono and Gunji 2004a,b; Tsuda et al. 2004), including shortest path (Nakagaki 2001; Nakagaki et al. 2001) and even design of controllers for robots (Tsuda et al. 2006).

Physarum machines

The scoping experiments were designed as follows: I either covered the container's bottom with a piece of wet filter paper and placed a piece of living plasmodium on it or just planted plasmodium on the bottom of a bare container and fixed wet paper on the container's cover to keep humidity high. Oat flakes were placed at the positions of given planar points to be spanned by a tree. Plasmodium was kept fed before experiments. Experiments took place at room temperature, circa 25^oC. The containers were stored in the dark except during periods of observation.

Once placed in the container and recovered, the plasmodium starts to explore the surrounding space. Numerous pseudopodia emerge, frequently branch, and proceed (Fig. 3a). The plasmodium growth from its initial position by protoplasmic pseudopodia detecting, by chemotaxis, relative locations of the closest sources of nutrients. When another source of nutrients, element of the given planar set, is reached, the relevant part of the plasmodium reshapes and shrinks to a protoplasmic strand or a tube. This tube connects initial and newly acquired sites. This protoplasmic strand represents an edge of the computed spanning tree (Fig. 3b).

Planar points distributed in a Petri dish are usually spanned by a slime mold tree in 2–3 days (Fig. 3c,d). The trees computed by *P. polycephalum* are remarkably close, in number of edges and total sum of edge lengths, to spanning trees computed by classical algorithms (Fig. 3e).

Conclusion

I adopted a paradigm of slime mold computing (Nakagaki 2001; Nakagaki et al. 2000) to experimentally imitate algorithms of spanning tree approximation in reaction–diffusion computers (Adamatzky et al. 2005) encapsulated in membranes. Experiments with *P. polycephalum* proved that the theoretical framework bears a huge potential for designing non-classical computing devices, and that the experimental approach is feasible.

By encapsulating reaction–diffusion systems in membrane, we did not speed up the computation. In general, plasmodium propagates with a speed similar to that of diffusing chemical species. Therefore, Physarum machines and reaction–diffusion computers are the same class of very slow natural computing devices. Speed of computation was not a goal of the present study. By containing a non-linear, spatially extended system in the membrane, we made the system capable of directional selectivity, controllable motion and, ultimately, building optimal graph structures.

Plasmodium can live on a wide variety of substrates, including bare glass and metal. Comparing to reaction–diffusion computers (which are mainly liquid phase or gel-based), Physarum machines are more stable, are more tolerant to unfriendly environments and can perform their tasks in a wider range of specialised computing devices.

Despite the success of the experimental implementation of the Physarum machine, I understand that criticism is coming and, to be precise, would like to make several clarifications.

First, slime mold does not compute a *minimal* spanning tree. To compute a minimal tree of *n* planar points, one should undertake at least *n* experiments, planting the initial piece of plasmodium at different points of the given set and then comparing total lengths of the trees calculated (however, even in such a situation, one can achieve reasonable complexity boundary $O(nD(n))$, where $D(n)$ is a diameter of the given planar set). In my experiments, a minimal – plasmodium converts distance to time – tree for a *given start point* is computed. Second, on many occasions, slime mold calculates a Steiner tree because sometimes protoplasmic strands branch at the points not belonging to the given planar set. Third, slime mold computation is rather interactive because once the spanning tree is computed, the computation does not stop, and more strands connecting oat flakes are formed, cycles in the calculated graph emerge and the proximity graph computed resembles relative neighbourhood or Gabriel graphs. This particular feature will be the subject of future studies.

Acknowledgements I am unlimitedly grateful to Soichiro Tsuda and Tomohiro Shirakawa for igniting my interest in slime mold computing, providing me with plasmodium culture and instructing me in full details on how to care for the plasmodium. Soichiro and Tomohiro, without your help and support, this paper would never have been written!

References

Adamatzky A (1991) Neural algorithm for constructing minimal spanning tree. Neural Netw World 6:335–339

Adamatzky A (2001) Computing in non-linear media and automata collectives. IoP, Bristol

Adamatzky A (2007) Phenomenology of retained excitation. Int J Bifurc Chaos (in press)

Adamatzky A, De Lacy Costello BPJ (2002) Collision-free path planning in the Belousov-Zhabotinsky medium assisted by a cellular automaton. Naturwissenschaften 89:474–478

Adamatzky A, De Lacy Costello B, Asai T (2005) Reaction–diffusion computers. Elsevier, Amsterdam

Agladze K, Magome N, Aliev R, Yamaguchi T, Yoshikawa K (1997) Finding the optimal path with the aid of chemical wave. Physica D 106:247–254

Aono M, Gunji Y-P (2001) Resolution of infinite-loop in hyperincursive and nonlocal cellular automata: introduction to slime mold computing. Computing Anticiaptory Systems, AIP Conference Proceedings, vol 718. AIP, Melville, pp 177–187

Aono M, Gunji Y-P (2004) Material implementation of hyperincursive field on slime mold computer. Computing Anticiaptory Systems, AIP Conference Proceedings, vol 718. AIP, Melville, pp 188–203

Matsumoto K, Ueda T, Kobatake Y (1986) Propagation of phase wave in relation to tactic responses by the plasmodium of Physarum polycephalum. J Theor Biol 122:339–345

Matsumoto K, Ueda T, Kobatake Y (1988) Reversal of thermotaxis with oscillatory stimulation in the plasmodium of Physarum polycephalum. J Theor Biol 131:175–182

Nakagaki T, Yamada H, Ito M (1999) Reaction–diffusion advection model for pattern formation of rhythmic contraction in a giant amoeboid cell of the Physarum plasmodium. J Theor Biol 197:497–506

Nakagaki T (2001) Smart behavior of true slime mold in a labyrinth. Res Microbiol 152:767–770

Nakagaki T, Yamada H, Toth A (2000) Maze-solving by an amoeboid organism. Nature 407:470–470

Nakagaki T, Yamada H, Toth A (2001) Path finding by tube morphogenesis in an amoeboid organism. Biophys Chemist 92:47–52

Nakamura S, Yoshimoto Y, Kamiya N (1982) Oscillation in surface pH of the physarum plasmodium. Proc Jpn Acad 58:270–273

Nakamura S, Kamiya N (1985) Regional difference in oscillatory characteristics of physarum plasmodium as revealed by surface pH. Cell Struct Funct 10:173–176

Ogihara S (1982) Calcium and ATP regulation of the oscillatory torsional movement in a triton model of physarum plasmodial strands. Exp Cell Res 138:377–384

Oster GF, Odel GM (1984) Mechanics of cytogels I: oscillations in physarum. Cell Motil 4:469–503

Ridgway EB, Durham ACH (1976) Oscillations of calcium ion concentration in Physarum plasmodia. Protoplasma 100:167–177

Steinbock O, Tóth A, Showalter K (1995) Navigating complex labyrinths: optimal paths from chemical waves. Science 267:868–871

Tsuda S, Aono M, Gunji Y-P (2004) Robust and emergent Physarum-computing. BioSystems 73:45–55

Tsuda S, Zauner KP, Gunji YP (2006) Robot control: from silicon circuitry to cells. In: Ijspeert AJ, Masuzawa T, Kusumoto S (Eds) Biologically inspired approaches to advanced information technology. Springer, Berlin Heidelberg New York, pp 20–32

Yamada H, Nakagaki T, Baker RE, Maini PK (2007) Dispersion relation in oscillatory reaction–diffusion systems with self-consistent flow in true slime mold. J Math Biol (in press)