Int. Journ. of Unconventional Computing, Vol. 0, pp. 1–25 Reprints available directly from the publisher Photocopying permitted by license only

Towards Programmable Smart Materials: Dynamical Reconfiguration of Emergent Transport Networks

JEFF JONES

Centre for Unconventional Computing, University of the West of England, Coldharbour Lane, Bristol, BS16 1QY, UK E-mail: jeff.jones@uwe.ac.uk

Received: January 4, 2011. Accepted: March 30, 2011.

Smart materials promise adaptive morphology and functionality of materials, however, controlling the desired pattern formation using simple and local bottom-up interactions is a difficult task, but one which living organisms appear to manage effortlessly. We have previously demonstrated a virtual material inspired by the slime mould *Physarum polycephalum* which, from simple interactions within a swarm based particle collective, forms complex emergent transport networks. One desired characteristic of smart materials is that they should be programmable, adapting their structure in response to external stimuli. As a step towards this aim we suggest a prototype method to dynamically reconfigure emergent transport networks, based on real-time network analysis of the current configuration and feedback via dynamic adjustment of network node weights. The analysis method utilises a novel collective memory of previous network history which is used to provide connectivity information to control a feedback method to the network nodes. Although simple in operation, the feedback method utilises complex neural network-like control including excitation, inhibition and refractory dynamics. The transitions of the reconfiguration method are analysed and high level motifs and transitions are described. We suggest how the dynamical reconfiguration method may be used as a spatially represented unconventional computing method for combinatorial optimisation problems including the Euclidean Travelling Salesman Problem. We conclude by discussing limitations and possible improvements to the dynamical reconfiguration method and exploring the potential advantages of exploring low-level and indirect methods of influence on smart materials.

Keywords: Smart materials, collective behaviour, transport networks, *physarum polycephalum*, combinatorial optimisation.

1 INTRODUCTION

Smart materials, currently the subject of much recent research, are materials which can change their structural and/or functional properties in response to external stimuli [15]. These may produce a change in morphology. For example, a deformed substance made from a shape memory metal alloy [27] will return to a previously 'programmed' shape when heated. Alternately, the external stimuli may result in a change of the function of the substrate. Examples include piezoelectric materials [12] which can be used as both sensors (responding to force by producing electrical charge) and actuators (responding to electrical current by producing a mechanical force). Smart materials may also refer to the self-assembly of structures at very small scales, as seen in biological systems [31]. In this method physical phenomena which exhibit self-organisation at small scales are harnessed to provide regularised assembly and patterning of small structures. Examples include the assembly of bulk materials and thin layers [10], with regularly spaced domains [13], or structures which exhibit or utilise complex macroscopic properties, such as hydrophobic patterning [11].

Smart materials suggest novel and advantageous methods of product assembly, device control, and even self-repair yet the technology faces significant difficulties. Aside from the obvious challenges in materials science, one important issue is how to control the behaviour of large aggregates of smart material substrate to conform them to the desired structural and functional behaviour? Yet natural systems represent an enviable solution to many of the problems faced by smart materials: For example, from a single specifying unit - a fertilised cell - and given the requisite energy and nutrient supply, a fully functioning 'device' ultimately emerges from a combination of lowlevel physical, chemical and genetic interactions (a tiny 'window' on these interactions is given in [16]). From a relatively small pool of materials emerges an organism which is capable of growth, movement, sensory functions, environmental interactions, learning, maintenance, and self repair. The goal or biologically inspired smart materials is to harness some of the mechanisms used by natural systems, particularly self assembly and the coupling of low level physical forces to develop materials, and ultimately entire devices, whose form and function are reconfigurable.

One the most striking and minimal examples of self-organising pattern formation for useful behaviour is the giant amoeboid organism *Physarum polycephalum*, or true slime mould. The plasmodium stage of this organism's complex life cycle exhibits extremely complex adaptive pattern formation behaviour. In this stage the organism is a single giant membrane bound cell where repeated nuclear division results in a large amorphous mass ranging from microscopic to over 1m². The plasmodium is able to sense nearby levels of chemoattractant and stream towards them, extending pseudopodia and engulfing and consuming bacteria on the substrate. The internal mechanism

generating the movement is the hydrostatic force generated by emergent synchronised contraction of a meshwork of actin-myosin filaments. The contraction forces induce changes in the structure of the plasmodium and changes from an immobile gel phase to watery sol phase are observed. As the plasmodium grows a protoplasmic transport network is formed, distributing nutrients between food sources engulfed by the plasmodium in a characteristic to-and-fro manner called shuttle streaming. The growth patterns are dependent to a large extent on environmental conditions, for example radial growth in nutrient rich environments and dendritic pseudopod-like growth in nutrient poor conditions (Figure 1). The organism is thus able to adapt its internal network pattern, and gross external morphology, quickly in response to changing environmental conditions.

The rapid adaptation of the organism to its environment, and the fact that the complex behaviour is distributed within the plasmodium among identical simple components, has stimulated recent interest in the possible computational behaviours of *Physarum*. Since the work of Nakagaki et al, who demonstrated the ability of the plasmodium to solve simple maze puzzles [26], there has been intense interest in exploring the computational behaviours of the organism. *Physarum* has been shown to be capable in solving graph construction problems [2,3], plane division problems [30], path planning problems [32,7,34] implementing classical logic gates and adder circuits [35,21], spatially represented universal computing machinery [1,6], combinatorial optimisation [8], robotic control systems [37] and robotic micro manipulation [5]. For a historical summary, and a survey of some recent examples from the field, see [25].

Physarum plasmodium may be considered as a living spatially represented smart material because it is composed of relatively simple parts yet exhibits complex self assembly and controllable pattern formation. The plasmodium



FIGURE 1

Growth patterns of Physarum polycephalum.

Left: Radial growth in nutrient rich oatmeal agar environment showing formation of protoplasmic transport network behind the active growth front.

Right: dendritic growth on nutrient poor damp filter paper showing the plasmodial transport network efficiently connecting food sources.

material can act as sensor *and* actuator [37,22] and the mechanisms controlling both properties arise from simple interactions and are distributed within unspecialised tissue [36] resulting in fault tolerance and self-repair (for example a plasmodium may be split into two parts, yielding two functional plasmodia, or alternately two plasmodia may fuse to form a single plasmodium). Control of the plasmodium smart material is by attractants (nutrient sources, warmth) to which the plasmodium moves towards, and repulsive sources (sodium chloride, light irradiation) which the plasmodium tries to avoid [4]. It is possible to control the spatial patterns of the plasmodium for the direct spatial encoding of problems. Gross control of plasmodium morphology was used in [6] and [4] by manually adding and removing nutrient sources or hazards but can the desired spatial configurations be achieved automatically?

An interesting hybrid spatial/logical approach to controlling *Physarum* was used by Aono et al. [9] in which a *Physarum* plasmodium was constrained by patterning it as the hub of a circular chamber surrounded by 16 stellate projections. Each projection corresponded to a possible link between nodes in a four node network. By applying an image analysis system and a neural-type feedback mechanism similar to Hopfield-Tank networks [17] they controlled the natural extension of pseudopodia to the projections surrounding the arena by light irradiation. Each arm represented a possible connection between cities in a Travelling Salesman Problem tour and the neural feedback mechanism was used to enforce the avoidance of invalid tours. The method was capable of finding good tours to the problem and the plasmodium spontaneously adapted its morphology between good nearby suboptimal tours, thus avoiding being trapped in local minima conditions.

The paper by Aono et al. shows that by constraining the natural spatial behaviour of the *Physarum* plasmodium it is possible to control its peripheral morphology to represent abstract problem representations. More recently, the same experimental approach, and similar results, was performed with a different material substrate composed of a population of the protist *Euglena gracilis* experimentally and in simulation [28]. The fundamental difference in the nature of the computing substrates (single-celled vs multi-cellular) suggests that distributed computation may be possible with a wide range of relatively simple materials.

Although the Aono et al. experiments demonstrated the morphological control of the periphery of the plasmodium, the full range of patterning of the migrating and adapting plasmodium was not exploited for computational purposes. Instead, the problem was encoded to fit the constraints of the experimental cell. In this report we investigate possible mechanisms for the dynamic spatial reconfiguration of smart materials, with the ultimate aim of controlling the behaviour of the material for *spatially represented* computation. We use a virtual material whose properties emerge from the interactions of a simple particle population. This 'material' is inspired by the amoeboid

organism *Physarum polycephalum* and has previously been shown to exhibit complex self-organised pattern formation behaviour under a number of parametric conditions which render it a suitable candidate for investigating the directed low-level assembly and reconfiguration of programmable matter. After introducing the collective and summarising some of its adaptive pattern formation characteristics we describe a novel real time network analysis method which is coupled to a neural inspired feedback mechanism whereby the spatial configuration of the material can be dynamically altered by the dynamic adjustment of network node weights to achieve pre-desired network characteristics. We present some results which illustrate the complex network transitions of the material in response to excitation, inhibition and refractory conditions generated by the feedback system, suggesting that the method and materials may be useful for spatially represented classical unconventional computation approaches to combinatorial optimisation tasks, including the Euclidean Travelling Salesman Problem and the Hamiltonian Path Problem. We conclude by discussing some advantages and limitations of the current approach and suggestions for future research.

2 METHODS

Emergent transport networks were introduced in [19] in an attempt to model the network adaptation behaviour of *P. polycephalum* using low-level, emergent and so-called 'bottom-up' methods. The approach is based on a population of mobile, particle-like agents residing in a 2D diffusive lattice. The agent particles execute simple sensory and motor instructions (Figure 2). During the sensory stage the particles sense the concentration of a generic 'chemotaxis' substance in the lattice from three forward positioned sensors and orient towards the strongest of the sensors. In the motor stage each particle attempts to move forwards in the lattice in its current orientation (an angle between 0 and 360°). If the movement is successful (i.e. the next cell is unoccupied, only one particle may occupy a lattice site) the particle moves to the new cell and deposits chemoattractant into the cell. If the move is not suc-



FIGURE 2 Particle morphology and sensory behaviour algorithm.

IJUC_ST(02)_JONES.indd 5

cessful the particle selects a new orientation randomly, stays in the current cell and does not deposit chemoattractant. The model is an example of LALI (Local Activation and Long-range Inhibition) reaction-diffusion pattern formation. Local activation is generated by the auto-catalytic nature of the particle behaviour: particles both sense and deposit the same chemoattractant substance. The long range inhibition component is generated by depletion of chemoattractant - particles are not attracted to regions of low chemoattractant concentration and the lack of occupation in these regions further reduces their attractiveness.

The population is initialised at random positions on the lattice and with random orientations. The population self-organises to form a dynamical network pattern. The network patterns are composed of the positions of the particles and the network undergoes complex morphological changes over time. The network exhibits quasi-physical behaviour, showing apparent tension in the network which causes the closure of network lacunae and the minimisation of paths at junctions. In this regard the network corresponds to soap film evolution (three branched junctions with Plateau angles of 120°), and the evolution of lipid nanotube networks in [24]. The default behaviour of the material is influenced by the particle sensory parameters and the environmental boundary and diffusion parameters. The full range of patterning behaviour of the material were explored in [18] and a brief example of characteristic pattern formation and evolution is shown in Figure 3.

Influences on the Behaviour of Emergent Transport Networks

The basic patterns are influenced by changes in particle sensory parameters (RA/SA for pattern type and SO for pattern scale) and correspond to classical



FIGURE 3

Formation and evolution of dynamical emergent transport networks.

Inverted network trails shown, darker regions indicate greater chemoattractant flux. a) Periodic boundary, SA 22.5°, RA 45° generates dynamical branching network b) Fixed boundary, SA 45°, RA45° generates sheet-like material Turing-type reticulated, labyrinthine and spotted patterns. The evolution (minimisation) of the transport networks can be influenced by projecting fixed point stimuli into the diffusion map at every scheduler step. Since the chemoattractant produced by the particles is also located in the diffusion map, the diffusion from projection sources is also attractive to the particle collective - the projection stimuli are analogous to food sources. When the network encounters a projection source the network is 'snagged' at that location by its attraction to the node and subsequent minimisation is constrained by the arrangement and position of the projection stimuli. When two or more paths at the same node come into contact with each other, the paths merge and form a single path from the node and the two separate paths are 'zipped' into a single path. The network evolution continues to close cycles and minimise path lengths until relaxation of the network paths is complete and the final result is a stable network pattern, albeit one whose components are in continual flux (Figure 4a). As the network reduces in size, the paths become thicker (since the population remains at the same size). To maintain thin network paths and, more importantly, allow for adaptive growth and reduction of the population in response to changing environments we employ the same particle algorithm with the modification to the population scheduler described in [7]. This modification spawns a new particle – provided there is sufficient free space around a particle - when stimulated by local movement and external chemoattractant stimuli (growth), and removes particles when the local particle density - in a local window centred around each particle - is too high (shrinkage).



FIGURE 4

Evolution of transport network and the effect of increasing 'nutrient' node weight on network connectivity.

a) Evolution of emergent transport network after 124, 1176, 2646 and 10001 steps

b) Left to right: Separate experiments at node projection weight 0.01, 0.1 and 1

It has been shown that, as with *P. polycephalum*, the emergent transport networks approximate the Toussaint hierarchy of proximity graphs [3, 20]. The evolution of the network is strongly affected by the projection strength of the nutrient nodes. Lower strength nodes exert less of an influence ('pulling' force) on the networks. Steiner nodes are formed within the network paths (Figure 4b, left) and the networks approximate lower regions of the proximity graph hierarchy. At higher node strengths the paths are attracted more strongly to the nodes and less closure of network cycles is seen (Figure 4b, middle). At very high node strengths there is very little opportunity for cycle closure (Figure 4b, right) and nodes are able to retain two separate paths emanating from the node at relatively acute angles, thus generating networks whose pattern and degree of connectivity resembles those at the upper range of the proximity graph hierarchy.

3 DYNAMICAL RECONFIGURATION METHOD

The results shown in Figure 4 demonstrate how different node projection weights affect the evolution of the network in separate runs. In [20] it was found that by systematically alternating periods of very high node weights followed by very low node weights, it was possible to gradually reduce the degree of connectivity of the network. The adjustment of node weights was performed by adjusting all nodes simultaneously. This, however, raises the possibility of a more dynamic control of network evolution within a single run: By using a suitable feedback mechanism is it possible to dynamically adjust individual node projection weights so that a 'preferred' configuration can be generated? By 'preferred' we mean a configuration that satisfies some pre-desired connectivity criterion, such as mean degree of connectivity per node. Both node weight adjustments and preferred configuration need only be specified in a local fashion. To feed back information to adjust node strengths we require some analysis of the current state of the transport network so we can provide local measures of the network configuration. The method given here uses information about the degree of node connectivity (i.e. for each node: how many other nodes is it connected to in total, and to which nodes exactly?).

3.1 Real-time Analysis of Transport Network Configuration

Analysing the state and connectivity of a spatially represented network at regular intervals is a computationally demanding task when performed at regular intervals. This workload is in addition to the computation required to generate the transport network behaviour of the virtual material and also the feedback mechanism. This raises an important distinction between the approach used by Aono et al. (which coupled the physical system of the slime mould to a computer controlled sensory/feedback mechanism) and the approach in this paper where the 'physical system' is itself the emergent product of a low-level simulation. The simulation of the physical system therefore demands particularly low cost methods of analysing the current state of the system. Using classical image processing methods requires sampling of current spatial state of the system. This involves smoothing the image, thresholding the image, binarisation, skeletonisation and finally establishing connectivity by tracking from node to node for each node at every network analysis step. To make the approach more computationally tractable we sought to develop a method whereby the network connectivity could be generated from the 'bottom-up' by the collective actions of the particles which comprise the transport network, yet still provide an accurate representation of network connectivity. The analysis which provides information about node connectivity is generated in the following way. Before an experiment is started the start positions of the nodes are automatically detected once by standard image processing methods and stored in an array, giving the number of network nodes and their (x, y) co-ordinates. This information is used to construct a two-dimensional array, the node connectivity table. Each entry in the table stores a relative strength between the two nodes referenced at the node table row and column positions. At the start of an experiment all connectivity is reset to zero.

During an experimental run each particle, initialised at a random position, receives a 'source_node' identifier when it passes the vicinity (specified by node_radius parameter, value of 11) of the first node it encounters. When the particle moves about the lattice its position is checked to see if it is in the vicinity of a new node. When the particle encounters a new node the new node is set as the 'destination node'. The node connectivity table at the two positions (source_node and destination_node) is incremented by a value (node inc), set to 1 (Figure 5, top row). Finally the particle takes the destination node value as the new source node and clears the destination node value until another new node is encountered. This process is repeated for all of the particles in the lattice and, over time, a collective 'picture' of node connectivity emerges as strength weights increase in the node connectivity table. The method of using the particles themselves to build up a measure of connectivity requires some additional complexity in the behaviour of the particles but is efficient in terms of storage (each particle needs only to store the source and destination node) and computational time since typically the population size quickly settles to <1000 particles. Although a very rough approximation, each collective update of the node table thus takes ~ 1000 simple computations (entries to the table) whereas classical image analysis methods are limited mostly in performance by screen area (typically 200×200 for these experiments) and thus each step in the image processing pipeline would require a minimum of 40,000 calculations, and some parts of the pipeline particularly connectivity tracking - require much greater cost. The collective representation of connectivity has another advantage over classical methods

9



FIGURE 5

Schematic illustration of real-time analysis of emergent transport networks.

Top Left: Node positions (1-9) are stored by automatic detection before experiment begins. Particle source node (s) and destination node (d) are indicated.

Top Right: Node connectivity table is updated whenever a particle arrives at a different node. Bottom Left: Actual spatial configuration of emergent transport network.

Bottom Right: Topological connectivity of emergent transport network. Crosses indicate node positions and circles indicate radius of the particle node detection vicinity. Thicker lines between indicate stronger connections. See text for explanation of cross lines between nodes.

in that the table also contains implicit information about the relative strength of path weights (i.e. how 'busy' each path connection is) which would not be achievable using classical tracking methods. This information is utilised during the feedback method and significantly affects the evolution of the network. One objection to this method of connectivity analysis is that it requires more complexity than would be seen in simple particles. This is indeed a valid criticism when applied to material properties of simple molecules and the method used does take the liberty of arbitrarily extending such limitations in the interests of computational tractability. It is of note, however, that similar distributed representations of collective connectivity have been observed in simple insects such as bee [23] and ant [14] colonies.

The collective representation of connectivity established by the particles may be subsequently used by the control system to apply feedback to the population. The degree of connectivity of a node to all other nodes can be assessed by checking the connectivity strengths along its row in the node table (any nodes with a value greater than zero indicates a connection to this node). The node connectivity table is effectively an emergent history, or collective memory, of the connectivity of the emergent transport network. However because the network is highly dynamical with constant changes in connectivity, the information in the table is soon outdated and irrelevant. To reduce the memory time of the node table all of the nodes in the table are decremented at every scheduler step by a value node_decay. This ensures that the node table can keep up to date with the changing network topology. Note that the node_decay parameter is critical in setting the memory time of the network. If the value is too low the connectivity table information used to control the feedback system will be out of date (e.g. referring to connections which no longer exist). However it is necessary to enforce some slight lag in order for the feedback method to have some tangible effect on the network structure. The effects of this parameter will be discussed in more detail in the description of the feedback mechanism.

The result of the dynamical network analysis is a transformation of the actual transport network flux into a real-time dynamic topological representation of network connectivity (Figure 5, bottom row).

Although the actual network and topology are similar, the topological representation shows extra connections (shown as the thin lines between nodes on Figure 5, bottom right). This is because at low node projection weights (where node strength is similar to network path strength) the transport network is partially detached from the nodes to adopt the minimal network shape (the Steiner tree, which has additional junctions in addition to the original node array). In doing so, some of the particles in the network path between three nodes do not actually visit the fixed middle node, instead passing through the Steiner point between the nodes. The topological representation illustrates that a fixed node may be indirectly connected, via a Steiner node, to a nearby node. The relative differences between node connections can be visualised by the differences in thickness of the dynamical representation of network topology.

3.2 Feedback Mechanism

The connectivity information in the node table is used during the feedback process to dynamically adjust node weights. The adjustment of node weights depends on certain rules. Different sets of rules may be specified to achieve different connectivity effects. The effect we are seeking in this prototype example is to attain a degree of connectivity for all nodes of exactly 2, thus the aim is to generate a complete network with only one cycle where all nodes have connectivity of 2. Such a path is a possible tour of the Travelling Salesman Problem (TSP, although not necessarily an optimal tour). The feedback rules are summarised in Table 1.

The weights of the nodes are initially set to a predefined fixed value 'start_ weight' at the start of an experiment because at the early stages of an experimental run a record of the network connectivity is yet to emerge in the node connectivity table. At regular intervals specified by the 'feedback_interval' parameter (every 10 scheduler steps) the degree of connectivity of each node is analysed and is updated according to the rules in Table 1. If the degree is

Degree of Connectivity	Node Status	Desired Action	Mechanism
0	Disconnected from network.	Reconnect node to network.	Increase node weight. Seed new particles.
1	Connected to only one other node.	Connect to another node.	Increase node weight. Seed new particles towards other single nodes.
2	Connected to two nodes in chain.	Reinforce Connectivity.	Increase node weight.
3	Directly or indirectly connected to three nodes.	Reduce Connectivity.	Decrease node weight.
>3	Directly or indirectly connected to more than three nodes.	Strongly Reduce Connectivity.	Strongly decrease node weight.

TABLE 1

Feedback mechanism in response to connectivity information

<2 then node weights are increased by the 'weight_inc' parameter (set to 0.001). If the degree is <1 (i.e. no network path passes through the node) the node weight is incremented by weight_inc * 2. The node weight is subject to a maximum value, 'weight_max', typically set to 10. Increasing the node weight attracts nearby network paths to the node due to the increased distance of diffusion of attractant from the node and nearby paths are 'snagged' by the node. If a network path is already connected to a node then further increasing the node strength strengthens the attraction of the path to the node (effectively 'tightening' the path and shortening the path distance between nodes).

If the degree of connectivity at a node is >2 then inhibition of the node is achieved by decreasing the node weight by the 'weight_dec' parameter (set to 0.0001). If the degree is >3 then the node weight is decreased by weight_dec * 2. Decreasing the node weight reduces the attraction of network paths to a node. The minimum value of node weight is zero. At low node weights the network detaches from the node as its attraction to the network is less than the flow within the network path. To further reduce the attraction to an overpopulated node, network trails nodes in such nodes are decremented by 'path_ decrease' which is calculated by the number of particles within a 5×5 window of the node position / 10.

The advantage of decreasing network paths from nodes is that much stronger changes in network connectivity can be obtained by temporarily decoupling paths from nodes. However, an issue which arises is how to reestablish contact with a node which has little contact with the remainder of the network (degree 1, i.e. the node is at the end of a path of connected nodes), or has been completely detached from the network due to inhibition by low node weight. To re-establish contact and increase connectivity to the node new network particles are seeded around the isolated node, preferentially in the direction of other isolated nodes but at a random angle otherwise. The distance at which new particles were seeded was set to a random value between zero pixels and the 'scatter distance' parameter (typically 10-20 pixels). The re-seeding mechanism would be very difficult (if not impossible) to implement with molecular materials but it may be possible to implement such a feature in other methods of physical computation such as gas discharge plasmas used in [29] for optimal path planning. It could also be argued that the re-seeding method also adds a mode of global control to an otherwise bottom-up approach, however, the reattachment of nodes is merely used to satisfy minimal constraints of network connectivity (one could interpret this merely as a mechanism to "keep the drivebelt on the engine"). A similar method was applied (but under more strict constraining conditions) in the original Hopfield-Tank approach to enforce a correct solution "syntax" [17], but although the re-seeding mechanism does allow large changes in network connectivity, it is not part of a global method to enforce solutions and thus cannot be said to detract from the bottom-up mode of network evolution.

The changes in node strength (increase or decrease in node projection strength) combined with the limited memory storage of the connection weight array (caused by a lag due to the decay factor) combine to generate a neural network-like feedback behaviour. The increase in node weights and subsequent attraction of the network paths corresponds to excitation. Conversely the reduction in node weights and the subsequent reduction in influence of the nodes on network paths correspond to inhibition. The neural network analogy can be extended if we consider the arrival of particles at the nodes as spiking stimuli which are integrated at regular intervals as feedback excitation or feedback inhibition. The persistence of the collective memory of network structure generates a lag period during which node weights at inhibited nodes are not increased. The lag (between actual network configuration and the collective memory of configuration) and low node weight at inhibited nodes allows the network structure to adapt and reconfigure. When inhibition is present (due to degree >2 connectivity), the inhibition combined with the delay lag can be considered as a refractory period. The refractory period is not programmed into the feedback mechanism but is an emergent property of the feedback lag and can be utilised to ensure that network adaptation can occur, without the feedback network responding too quickly to changes in topology. The cycle of feedback behaviours can be summarised by Figure 6 showing the relationship between actual network connectivity, the collective memory of logical connectivity, the feedback mechanism, and the subsequent effect on node weights.



FIGURE 6

Relationship between network behaviour, analysis and feedback mechanisms in adaptive reconfiguration of emergent transport networks.

4 RESULTS

We began with simple images with regularly spaced node arrangements to examine the behaviour of the network in response to dynamic adjustment of node weights. Figure 7 shows a simple image with six nodes and the results of five separate experiments at different fixed node projection weights. At the highest setting (5) there is little reduction in network length as the nodes exert a very strong attraction to the network, preventing the network paths from direct interaction. As the node weights decrease the influences of the nodes decreases and network paths come into contact with other paths resulting in minimisation until, at the lowest weight, the network evolves to its minimum shape to form the Steiner tree. The varying network patterns at different node weights suggest that dynamical reconfiguration of node weights (i.e. dynamically varying weights over time) would yield complex network evolution.

When dynamic reconfiguration (DR) is activated the network evolution begins in the regular manner but then undergoes a very complex transformation as the network weights are adjusted in response to connectivity feedback. The complex evolution can be seen more fully in the supplementary video recordings located at (http://uncomp.uwe.ac.uk/jeff/dynamic.htm). The two nodes (indicated in Figure 8(a), circled) are of degree three and so their projection weight is progressively weakened by the feedback mechanism. When the weight of the two inner nodes is sufficiently weak their 'hold' on the network is removed and the network detaches from the node and contracts, mov-

14



FIGURE 7 Results from separate experiments using fixed node weights. Left to right: Node weights 5, 1, 0.1, 0.01, 0.005.

ing upper and lower network paths towards each other (b). When the upper and lower paths contact their flows merge and the 4 way junction is minimised (c), resulting in both network cycles being pulled to the outer nodes of the shape (d). Because the two vacant nodes (seen in (e), circled) are no longer visited by particles, their inhibition is reduced as the cumulative 'memory' of the prior connectivity at each node is gradually erased. When this occurs, the node weights are progressively increased until the lower node attracts the network path by seeding new network trails at random positions close to the lower node (f). The upper node reattaches itself to the network in a similar way (g). Although it is not apparent in the static images, there is considerable interplay between the previous network history (due to the lag caused by the cumulative nature of the network analysis combined the refractory period of the inhibition of node weights) and this causes temporary detachment of nodes from the network (h). The network evolution continues (i-j) until all nodes are of degree two and the network then remains in this stable configuration. It is notable that the DR approach is successful in generating a minimal tour around the nodes, even though the 'search' performed by the feedback mechanism is effectively blind.

When a pattern of 12 nodes arranged in a cross-like fashion is presented to the network we saw that the outermost nodes were minimised to degree two before the effects of the DR procedure had any influence on inhibiting the node weights (Figure 9, a-c). When the weights of the inner nodes (degree 4) were reduced the network was pulled to the outer nodes (d-e). On removal of inhibition, and as the inner node weights increased, the nodes attracted the network paths from the outer nodes and a minimal tour was again generated to form a stable state. The anchoring of the network to outer nodes is a common motif and appears to stabilise the evolution of the network, ensuring that changes in configuration in one area do not impinge on the structure of the network too greatly at other locations.

Particular spatial patterns in the transitions between network configurations are regularly seen. Some of these motifs are relatively simple, relating directly to the feedback mechanism, including the passive attachment (of 'snagging') of nearby paths by attraction to the nodes, the minimisation of path length between two nodes ('tightening'), the releasing ('detaching') of network paths from nodes at very low node weights, and the active

a)	b)		d)	e)
f)	g)	h)	i)	j)

FIGURE 8

Dynamical reconfiguration of six node network, see text for details.



FIGURE 9 External node fixation as a factor in network minimisation.

re-connection ('grabbing') of isolated nodes to the network by seeding new particles around the isolated nodes. Other motifs are more complex and cannot simply be reduced to the feedback options at a single node. For example, interactions with nearby nodes selectively weaken certain nodes and can guide network evolution away from deadlock conditions. This allows a network path edge to shift position ('sliding') between nearby nodes, leaving nearby connectivity relatively unaffected and thus subtly adjusting network structure. The network evolution motifs newly observed in the DR method complement the constraining of paths by nodes and the 'zipping' and 'unzipping' of network paths previously observed in the virtual material [18].

When examining the evolution of the DR network there are transition periods as the network responds to changes to the node weights. These are followed by relatively stable periods as the new network configuration has an effect on the collective record in the feedback system (as new configurations are reinforced and the 'memory' of older configurations fades). Each stable period of connectivity may be considered as the spatial equivalent of a high level computational output arising from low level functionality: The function 'output' is a new (relatively) stable configuration and the low level function, or procedure, is the sequence of atomic interactions where network paths are altered by changes in local attraction by the amplification and inhibition of node weights. As an example of the relationship between high level and low level spatial activity, consider the sequence of temporarily stable configurations in Figure 10 from an experiment with a 4×3 regular network of nodes.

The sequence shows 23 transitions until the final pattern, corresponding to a minimal tour, stabilises the network. Although the minimal tour is found the blind nature of the search ensures that the search is not optimal. In a number of cases a pattern very close to that of the final tour is found, only to be lost in the next reconfiguration. A number of patterns are unique but occasionally identical individual patterns and identical transitions between patterns are seen. For example, the two sequences of patterns surrounded by dashed rectangles in Figure 10 are identical (albeit mirror images of each other) and undergo two identical transitions before differing only in the output pattern of the third transition. To find the cause of the different transition outputs we must examine the low-level interactions which generate the relatively stable high-level patterns. The transitions examined are those indicated by dashed ellipses in Figure 10 because the first images in each transition pattern identical and not simply mirror images, making the low-level transitions simpler to follow. The low-level transformations of the identical pattern that result in differing outputs are shown in Figure 11.

Although the connectivity of the two initial patterns (i) and (ii) is identical, the connection weights at the nodes are not exactly the same. In (i)a

1.1		H	; R1		E2H	ΕX
	ΞĂ	<u>III</u>			EX.) EXH
	ΕIJ	Ш		62	HI	БJ
		IH	R		БЛ	





FIGURE 11 Differences in low-level interactions result in differing transitions between patterns.

there is only one 'weakened' node (circled) whereas in (ii) a there are two nodes (circled) which are influenced by the feedback mechanism. This difference is responsible for the different pattern in the stable output result because transition (i)b-f only involves the lower nodes in the network whereas in transition (ii)b-g both lower and upper nodes are incorporated into the network evolution (direction of network path movement is indicated by arrows and newly reactivated nodes are circled). This result illustrates both the sensitivity of the mechanism to small changes in connectivity and also the effect of previous network connectivity history on future evolution of the network.

The examples of the DR mechanism shown previously have used regular spaced nodes. This results in relatively simple transformations with only a narrow range of angles in the networks (typically 90° or 45°). How does the DR mechanism respond to more complex arrangements of nodes, with irregular placement in terms of distance between nodes and node angle?

Figure 12 shows the evolution of one network which follows a similar evolution to those with regular node arrays: A 'backbone' of outer nodes is the first to stabilise and then the configuration of inner nodes is incorporated into the network structure. Unlike the regular arrays, however, the network configuration never fully stabilises. Certain configurations are generated (for example those shown on the bottom row of Figure 12), only to be remodelled (because one or more nodes has mean degree of 3). This pattern of evolution was found to be very repetitive with the network oscillating between semi stable states.

Figure 13 illustrates the effect of the DR method on the mean degree of node connectivity throughout an experimental run. The mean degree fluctuates as the network configuration changes in response to the dynamical adjustment of node weights. The images above the plot (dashed boxes) show configurations at the time of increasing mean degree (i.e. sub-optimal networks at the crests of the plot). These configurations are typically transient and occur during major shifts of network patterns. The lower images below the plot (solid boxes) coincide with local minima of mean degree. Note that



FIGURE 12 Dynamical reconfiguration showing the evolution of more complex spatial node arrangements.





Oscillations in mean degree during dynamic reconfiguration caused by changing node weights. Left: Plot of mean degree with indicative network structures

Right: 3d plot of node weight strength throughout the experiment

although these local minima have mean degree close to the desired specification (2), the patterns are not stable (due to isolated nodes or occasional degree of 3) and the network soon transforms into another configuration. The evolution of the network only halts when stability of configuration has been achieved. Also shown in Figure 13 (right) is an illustration of the changing node weights for each of the 11 nodes throughout the same experiment (time direction is arrowed). The highest point on each node 'lane' corresponds to maximal node weight (10, no inhibition of node weight at all) and the lowest point corresponds to the minimum weight (zero, maximum inhibition of node weight, i.e. no projection of attractants). The pattern of node weights shows the independent nature of the node weight adjustment by the feedback mechanism, and it is only when the network stabilises into the final pattern that the node weights are synchronised.

4.1 Characteristics, Limitations and Interpretations of Results

20

We characterise the behaviour of the DR method using examples in which the method is able to evolve stable networks of degree 2 connectivity which correspond to valid TSP solutions alongside examples in which TSP tours are either transient or incomplete. In examples with rectilinear and regularly spaced nodes complete TSP tour solutions were found for examples of up to 12 nodes (cities). This compares favourably with the original slime mould research of Aono et al. [8] which contained a 4 city problem and has since been expanded to 8 city instances. Of course both this approach and the original slime mould methods are mainly concerned understanding the complex dynamics within their respective materials and have some way to progress before matching current state-of-the-art classical approaches capable of solving many thousands of city instances.

In problems with irregularly placed cities the performance of the DR method was more variable (Figure 12 to Figure 15) and the causes of the variation are worthy of further consideration. On some runs a complete circuit was not found at all and the connectivity pattern instead approximated a Hamiltonian path (a network where each node is visited once only). The results in Figure 14 show the similarities between the DR networks and the optimum Euclidean TSP tours (column 2, calculated by Simulated Annealing method) for three different sets of points. The general network shape (column 3) appears to approximate the TSP tour but some of the nodes are not visible as they are inhibited by the feedback system. The connectivity diagram (column 4) shows the 'problem areas' in the dynamical network (lighter coloured

Nodes	Optimal TSP	Transport Network	Logical Connectivity	Hamiltonian Path
		Þ.	ES .	C.
	\square		$\widehat{\boldsymbol{\varphi}}$	
		G	B	Û,





FIGURE 15

Incomplete or invalid tours generated by the DR method show limitations of the analysis and feedback mechanisms.

circles) which prevent complete optimisation of the network. For the first two sets of points (top and middle row) the correct optimisation is prevented because the network cannot remain attached around a node which is relatively far away from other nodes (column 3, dashed circles). Instead the network detaches from this node and forms a Steiner point between the three surrounding nodes. The detachment from this node was observed even when stronger maximum node weights were used. In the case of the bottom row set of points it appears that the network cannot stabilise due to the very acute angle between two nodes (column 2, dashed circle) which causes paths to merge and detach from the node. In this case the network cannot stabilise and oscillates between nearby configurations (as in Figure 12, bottom row). This appears to be very similar behaviour to that discussed in the original slime mould research in which oscillation between nearby tours was also observed [8]. The final column shows Hamiltonian paths for two of the data sets (the middle row dataset never converged to form the path). In these cases the paths did not connect to form a TSP tour because either the distance between the nodes was too great (top example) or the angles between the nodes was too acute (seeded particles were quickly absorbed into nearby network paths instead of spanning the two nodes).

It is possible that these partial or incomplete solutions may be amenable to further modification or refinement to provide complete TSP solutions. Such an approach was used by Tero et al. to refine partial solutions to the Steiner problem using a computational model of *Physarum polycephalum* [33]. Alternately, the strong visual similarity of the solutions to optimal TSP tours suggests that an automated interpretation mechanism may be possible to 'read off' the result from the transport network. For example, an approach based on a large particle population was found to approximate simple TSP tours when traversing the peripheral outline of the collective [19].

Other limitations of the prototype mechanism are concerned with the feedback rules. For example, Figure 15 (first two images) shows some instances where the network is not fully connected and a complete cycle is not found. These examples were caused by setting a very low value to the scatter_distance parameter. Although this can be solved simply by increasing the parameter it represents a limitation when we consider networks where we do not know in advance the distances between the nodes.

The final image shows a network which has mean degree of 2 yet does not form a single cycle. Both sources of error (incomplete tours and multiple cycle tours) are due to limitations in the control mechanism arising from the motivation to keep the prototype mechanism as simple as possible, and to employ only local node information about the network structure. We envisage that additional global tests of network connectivity (for example to check that a tour is valid by checking if a connected path between all nodes exists), and an enforcement of correct tour "syntax" (as referred to in the original Hopfield-Tank control method [17]) could be added to future versions of the analysis/feedback control mechanism.

5 DISCUSSION

We have presented a prototype mechanism to explore the problem of dynamic reconfiguration of transport networks. The mechanism is used to modify the complex collective behaviour of a virtual material, inspired by the true slime mould *Physarum polycephalum*, whose 'default' behaviour demonstrates network minimisation. The dynamical reconfiguration mechanism utilises real-time pattern analysis to attain a collective record of network connectivity of the virtual material as it adapts around the spatial configuration of 'food' node stimuli projecting attractants into the environment. This record is used to measure the degree of connectivity at each node and this information is then fed back to the nodes – enhancing the node strength for low connectivity ($\leq=2$) and suppressing the node strength at high connectivity ($\geq=3$). A seeding method (creation of new particles near the local area of disconnected nodes) is utilised to re-establish network connections at fully or partially disconnected nodes ($\leq=1$).

The feedback mechanism alters the attraction of the nodes to the network and the network automatically reconfigures its shape in response to the changing stimuli. An emergent refractory period is effected by controlling the length of time in which the 'memory' of the current configuration is retained. This refractory period is necessary to avoid spatial deadlock situations in which the network evolution is trapped between two similar configurations by nearby competing nodes. The DR mechanism generated very complex network transitions which, when analysed, demonstrated apparently high-level stable spatial structures from the low level 'atomic' and indirect actions of changing node weights. The transitions were composed from characteristic spatial motifs including so-called *snagging*, *tightening*, *detaching*, *grabbing* and *sliding* actions.

The method demonstrates a potential approach for the *spatially represented* unconventional computation of computational geometry and combinatorial optimisation problems. In many cases of bio-inspired computation inspired by living systems (for example ant algorithms or genetic algorithms) the complex spatial behaviour and underlying mechanisms of the living systems are abstracted into conventional symbolic computer instructions and the computation is performed only by analogy with the living system. For example in the Ant Colony Algorithm approach, no actual deposition of pheromone trail, or its subsequent propagation by evaporation and diffusion, is used. Conversely, unconventional computation schemes seek to utilise the spatial properties of physical and living systems directly in the computational process, for example the propagation of information by diffusion in chemical computation, or growth and adaptation of the Physarum plasmodium itself. Utilising the direct physical processes for computation provides a very rich set of behaviours, some of which have been investigated in this report. However, as this report also demonstrates, the complex behaviour also comes at some cost - namely that of trying to control or govern the very complex evolution of the network and 'tune' the analysis and feedback mechanism. The computational 'richness' afforded by the spatial computation may be instead interpreted more simply as 'difficulty' and akin to the idiom of "nailing jelly to a tree". However, even the prototype analysis and blind feedback mechanisms developed in this report were shown to approximate TSP tours and Hamiltonian paths in relatively complex spatially represented datasets, suggesting that approach may be worth further investigation.

But why investigate and wrestle with the control of complex spatial behaviour at all? Although we can indeed avoid the computational complexity of spatial systems by resorting to symbolic abstraction in conventional computing devices, it will not be possible to do this in future physical embodiments of smart materials which exist outside the simple and controllable confines of classical computation, in spatially complex and noisy environments. It is hoped that by attempting to 'tame' the complex network evolution in the restricted form of a virtually represented spatial system, we may be able to generate interesting and useful approaches and solutions to computational problems which may also be applicable to physical instances of smart materials.

Even with the limited complexity of the analysis and feedback systems introduced in this report, we have revealed a rich, and surprising, set of transition behaviours emerging from very simple low-level interactions. We speculate that this is only a small possible subset of behaviours which may exist in alternative analysis and feedback schemes, and that 'mining' these behaviours may prove useful for the control of smart materials. To this end we plan to improve the method in the near future: Firstly, by increasing the sophistication of the pattern analysis method whilst maintaining computational tractability. Secondly we propose to use a more complex feedback system which will tailor the evolution of the reinforcement and inhibition to more successfully reward more optimal patterns, weaken the memory of less optimal patterns, and remove invalid patterns. We hope this approach will prove a useful method in the dynamical control of the behaviour of smart materials.

ACKNOWLEDGEMENTS

The work was supported by the Leverhulme Trust research grant F/00577/1 "Mould intelligence: biological amorphous robots" led by Professor Andrew Adamatzky.

REFERENCES

- Adamatzky, A., Physarum Machine: Implementation of a Kolmogorov-Uspensky machine on a biological substrate. Parallel Processing Letters, 2007. 17(4): p. 455–467.
- Adamatzky, A., Physarum machines: encapsulating reaction-diffusion to compute spanning tree. Naturwissenschaften, 2007. 94(12): p. 975–980.
- [3] Adamatzky, A., Developing proximity graphs by Physarum Polycephalum: Does the plasmodium follow Toussaint hierarchy? Parallel Processing Letters, 2008. 19(1): p. 105–127.
- [4] Adamatzky, A., Manipulating substances with Physarum polycephalum. Materials Science and Engineering: C, 2010. 30(8): p. 1211–1220.
- [5] Adamatzky, A. and J. Jones, *Towards Physarum robots: computing and manipulating on water surface*. Journal of Bionic Engineering, 2008. 5(4): p. 348–357.
- [6] Adamatzky, A. and J. Jones, *Programmable reconfiguration of Physarum machines*. Natural Computing, 2009: p. (DOI: 10.1007/s11047-009-9146-8).
- [7] Adamatzky, A. and J. Jones, *Road planning with slime mould: If Physarum built motor-ways it would route M6/M74 through Newcastle.* International Journal of Bifurcation and Chaos, 2010. 20(10): p. 3065–3084.
- [8] Aono, M. and M. Hara, Amoeba-Based Nonequilibrium Neurocomputer Utilizing Fluctuations and Instability. Lecture notes in computer science, 2007. 4618: p. 41–54.
- [9] Aono, M., et al., Amoeba-based chaotic neurocomputing: Combinatorial optimization by coupled biological oscillators. New Generation Computing, 2009. 27(2): p. 129–157.
- [10] Ariga, K., et al., *Challenges and breakthroughs in recent research on self-assembly*. Science and Technology of Advanced Materials, 2008. 9: p. 014109.
- [11] Chandler, D., *Interfaces and the driving force of hydrophobic assembly*. Nature, 2005. 437(7059): p. 640–647.
- [12] Chopra, I., Review of state of art of smart structures and integrated systems. AIAA journal, 2002. 40(11): p. 2145–2187.
- [13] Darling, S.B., Directing the self-assembly of block copolymers. Progress in Polymer Science, 2007. 32(10): p. 1152–1204.
- [14] Dorigo, M., E. Bonabeau, and G. Theraulaz, Ant algorithms and stigmergy. Future Generation Computer Systems, 2000. 16(8): p. 851–871.
- [15] Gandhi, M.V. and B.S. Thompson, Smart materials and structures. 1992: Springer.
- [16] Hayashi, T. and R.W. Carthew, Surface mechanics mediate pattern formation in the developing retina. Nature, 2004. 431(7009): p. 647–652.
- [17] Hopfield, J.J. and D.W. Tank, *Computing with neural circuits: A model*. Science, 1986. 233(4764): p. 625.
- [18] Jones, J., Characteristics of Pattern Formation and Evolution in Approximations of Physarum Transport Networks. Artificial Life, 2010. 16(2): p. 127–153.
- [19] Jones, J., The Emergence and Dynamical Evolution of Complex Transport Networks from Simple Low-Level Behaviours. International journal of Unconventional Computing, 2010. 6(2): p. 125–144.

- [20] Jones, J., Influences on the formation and evolution of Physarum polycephalum inspired emergent transport networks. Natural Computing, 2010. November 2010: p. 1–25.
- [21] Jones, J. and A. Adamatzky, *Towards Physarum Adders*. BioSystems, 2010. 101(1): p. 51–58.
- [22] Jones, J., S. Tsuda, and A. Adamatzky, *Towards Physarum Robots*, in *Bio-Inspired Self-Organizing Robotic Systems, Studies in Computational Intelligence series*, Y. Meng and Y. Jin, Editors. 2011, Springer-Verlag.
- [23] Lihoreau, M., L. Chittka, and N.E. Raine, *Travel optimization by foraging bumblebees through readjustments of traplines after discovery of new feeding locations*. The American Naturalist. **176**(6): p. 744–757.
- [24] Lobovkina, T., et al., Shape optimization in lipid nanotube networks. The European Physical Journal E-Soft Matter, 2008. 26(3): p. 295–300.
- [25] Nakagaki, T., ed. *The Birth of Physarum Computing*. Int. Jnl. Unconventional Computing., ed. A. Adamatzky. 2010, Old City Publishing: Philadelphia.
- [26] Nakagaki, T., H. Yamada, and A. Toth, *Maze-solving by an Amoeboid Organism*. Nature, 2000. 407(470).
- [27] Otsuka, K. and C.M. Wayman, Shape memory materials. 1999: Cambridge Univ Pr.
- [28] Ozasa, K., et al., Simulation of neurocomputing based on the photophobic reactions of Euglena with optical feedback stimulation. BioSystems, 2010. 100(2): p. 101–107.
- [29] Reyes, D.R., et al., *Glow discharge in microfluidic chips for visible analog computing*. Lab On A Chip, 2002. 2(2): p. 113–116.
- [30] Shirakawa, T. and Y. Gunji, Computation of Voronoi diagram and collision-free path using the plasmodium of Physarum polycephalum. International journal of Unconventional Computing, 2010. 6(2): p. 79–89.
- [31] Stupp, S.I., Self-Assembly and Biomaterials. Nano Letters: p. 249-255.
- [32] Tero, A., R. Kobayashi, and T. Nakagaki, *Physarum solver: A biologically inspired method of road-network navigation*. Physica A: Statistical Mechanics and its Applications, 2006. 363(1): p. 115–119.
- [33] Tero, A., et al., A method inspired by Physarum for solving the Steiner problem. International journal of Unconventional Computing, 2010. 6(2): p. 109–125.
- [34] Tero, A., et al., Rules for Biologically Inspired Adaptive Network Design. Science, 2010. 327(5964): p. 439–442.
- [35] Tsuda, S., M. Aono, and Y. Gunji, *Robust and emergent Physarum logical-computing*. BioSystems, 2004. 73(1): p. 45–55.
- [36] Tsuda, S. and J. Jones, *The emergence of synchronization behavior in Physarum polyceph-alum and its particle approximation*. BioSystems, 2011. **103**(3): p. 331–341.
- [37] Tsuda, S., K. Zauner, and Y. Gunji, *Robot control with biological cells*. BioSystems, 2007. 87(2–3): p. 215–223.