Chapter 9
Material Approximation of Combinatorial Optimisation

"There's a city in my mind,
Come along and take that ride "

(David Byrne, 1985)

9.1 Introduction

The Travelling Salesman Problem (TSP) is a combinatorial optimisation problem well studied in computer science, operations research and mathematics. In the most famous variant of the problem a hypothetical salesman has to visit a number of cities, visiting each city only once, before ending the journey at the original starting city. The shortest path, or tour, of cities, amongst all possible tours is the solution to the problem. The problem is of particular interest since the number of candidate solutions increases greatly as n, the number of cities, increases. The number of possible tours can be stated as $(n-1)!/2$ which, for large numbers of n, renders assessment of every possible candidate tour computationally intractable. Besides being of theoretical interest, efficient solutions to the TSP have practical applications such as in vehicle routing, tool path length minimisation, and efficient warehouse storage and retrieval.

The intractable nature of the TSP has led to the development of a number of heuristic approaches which can produce very short — but not guaranteed minimal — tours. A number of heuristic approaches are inspired by mechanisms seen in natural and biological systems. These methods attempt to efficiently traverse the candidate search space whilst avoiding only locally minimal solutions and include neural network approaches (most famously in [204]), evolutionary algorithms [207], simulated annealing methods [208], the elastic network approaches prompted in [209], ant colony optimisation [206], living [195] and virtual [210] slime mould based approaches, and bumblebee foraging [205].

J. Jones, *From Pattern Formation to Material Computation,*
Emergence, Complexity and Computation 15, DOI: 10.1007/978-3-319-16823-4_9

Human performance on the TSP has also been studied in both naive and tutored subjects (see, for example, [211]). This is of particular interest because, unlike many nature inspired approaches, the human computation of TSP is by an individual and not based on population methods which evaluate a number of candidate solutions. Human performance on the TSP is also, for a limited number of cities at least, comparable in performance with heuristic approaches [212], [213]. Although there are a number of competing theories as to how exactly humans approximate the TSP [214],[212], [215], discovery of the methods employed may be useful as an insight into the mechanisms underlying complex perceptual and cognitive processes and potentially as an aid for the development of computational algorithms.

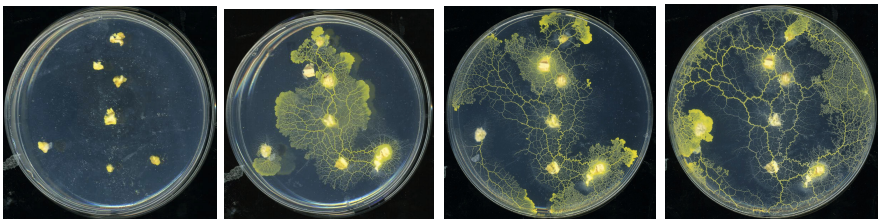
In this chapter we adopt a material-based, minimum complexity approach. We show how a spatially represented non-classical, or unconventional, computational mechanism can be used to approximate the TSP. Taking inspiration from the non-neural, material-based computational behaviour of slime mould, we employ a sheet, or 'blob' of virtual material which is placed over a spatial map of cities. By shrinking this blob over time, it conforms and adapts to the arrangement of cities and a tour of the TSP is formed. We give an overview of the inspiration for the method in Section 9.2. The shrinking blob method is described in Section 9.3. Examples of the performance of the method compared to exact solutions generated by a TSP solver are given in Section 9.4, along with an analysis of the underlying mechanism and factors affecting the performance of the approach. We conclude in Section 9.5 by summarising the approach and its contribution in terms of simplicity. We examine similarities between the underlying mechanism of the shrinking blob method and proposed models of TSP tour perception and construction in studies of human performance on the TSP. We suggest further research aimed at improving the method.

9.2 Can Slime Mould Directly Compute the TSP?

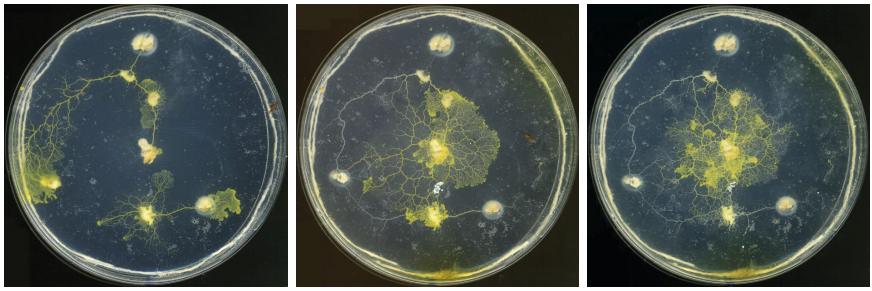
Although *Physarum* has been previously been used in the approximation of TSP [195], this was achieved by an indirect encoding of the problem representation to enable it to be presented to a confined plasmodium in a controlled environment. In the work by Aono et. al. it was shown that the morphology of the plasmodium confined in a stellate chamber could be dynamically controlled by light irradiation of its boundary. When coupled to an elegant feedback mechanism using an analysis method (to assess the presence of plasmodium at the extremities of the chamber), combined with Hopfield-Tank type neural network rules [204], the plasmodium was used to generate candidate solutions to simple instances of the TSP [85, 86]. In its natural propagative state, however, *Physarum* does not approximate area representations of a set of points, including the Convex Hull, Concave Hull [81] and the TSP. This is because the material comprising the plasmodium spontaneously

forms networks spanning the nutrient sources. Even when the plasmodium is arranged initially as a solid sheet of material, the sheet is soon transformed into a network structure by competitive flux of material within the sheet [9]. It is physically impractical to force a freely foraging plasmodium to conform to a TSP network structure during its nutrient foraging, as shown in Fig. 9.1.

Nevertheless, the material computation embodied within *Physarum* presents interesting possibilities towards generating novel spatially represented methods of unconventional computation. The feedback control of network evolution demonstrated in Chapter 8 resulted in extremely complex transitions of network dynamics and only partial success in constructing TSP tours. In the approach outlined in this chapter we attempt a simpler approach which utilises a larger aggregate mass of the same multi-agent collective which behaves as a morphologically adaptive cohesive 'blob' of virtual material.



(a) 0h (b) 12h (c) 24h (d) 36h



(e) 10h (f) 24h (g) 36h

Fig. 9.1 Foraging plasmodium of *Physarum* does not approximate the TSP in both unconstrained and constrained environments. (a) *Physarum* plasmodia are inoculated at oat flakes on non-nutrient agar, (b) individual plasmodia extend from oat flakes and fuse, (c-d) the plasmodium continues to forage and the shape of the TSP is not represented. (e) foraging of plasmodium is constrained by placing a ring of saline soaked thread (1g NaCl / 100g water) at the periphery of the arena, (f) as the salt diffuses into the agar the shape of the plasmodium is confined, (g) the pattern of the plasmodium at 36h is confined but is disconnected from outer nodes (note empty tube remnants) and does not approximate the TSP. Images from [216].

9.3 Material Approximation of the TSP by a Shrinking Blob

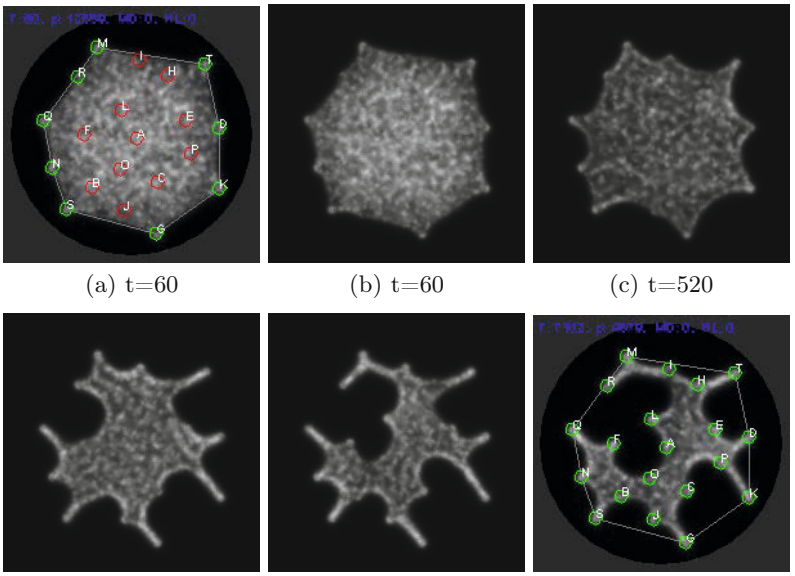
In the shrinking blob method we use a piece, or 'blob' of a virtual plasmodium material to approximate the TSP. In this application we use a relatively large population of particles which collectively behaves as a sheet of deformable virtual material. An overview of the method follows.

9.3.1 Shrinkage Process

We initialise a sheet of the virtual material around a set of data points corresponding to TSP city nodes (Fig. 9.2a). Chemoattractant is projected into the diffusive lattice at node locations, however, projection is reduced at regions which are covered by the blob sheet. The initial shape of the sheet corresponds to the Convex Hull of the data points. We then shrink the material by systematically removing some of its constituent particle components. The city nodes act as attractants to the material, effectively 'snagging' the material at the locations of uncovered nodes and affecting its subsequent morphological adaptation. As the material continues to shrink its innate minimising properties conform to the locations of the city nodes and the area occupied by the material is reduced, becoming a concave area covering the nodes (Fig. 9.2b-e). The shrinkage is stopped when all of the nodes are partially uncovered by the sheet (Fig. 9.2f). The reader is encouraged to view the supplementary video recordings of the shrinkage process as described in the Appendix. The adaptation of the blob to the data stimuli is not entirely smooth, the video recordings show that the blob sheet adapts to the changing stimuli as data nodes are temporarily uncovered and re-covered by the blob. When the shrinkage is halted the area of the sheet corresponds to the area enclosed by a tour of the Euclidean Travelling Salesman Problem. The exact tour formed by the blob can be elucidated by tracking along the perimeter of the blob, adding a city to the tour list when it is first encountered. The tour is complete when the start city is re-encountered. The approach is simple, making use of the innate adaptive emergent properties of the material. Despite being completely unguided and containing no population based heuristic optimisation strategies the approach yields efficient tours. The separate stages of the approach will now be described in detail.

9.3.2 Halting the Computation

It is important to halt the shrinkage of the blob at the right time. If the shrinking is stopped too early an incomplete tour will be formed (i.e. only a partial subset of the nodes will be included in the tour if not all of the nodes are uncovered). Unlike guided heuristic methods a set of candidate tours is not initially formed and subsequently modified. Only a single tour



(a) t=60 (b) t=60 (c) t=520

(d) t=3360 (e) t=6360 (f) t=7102

Fig. 9.2 Visualisation of the shrinking blob method. (a) sheet of virtual material initialised within the confines of the convex hull (grey polygon) of a set of points. Node positions are indicated by circles. Outer partially uncovered nodes are light grey, inner nodes covered by the sheet are in dark grey, (b-e) sheet morphology during shrinkage at time 60, 520, 3360, 6360 respectively, (f) shrinkage is stopped automatically when all nodes are partially uncovered at time 7102.

is formed and the shrinking blob approach is akin to the 'instance machines' (as opposed to universal machines) proposed by Zauner and Conrad [61]. To automatically halt the computation we use a so-called 'traffic light' system. At the start of the method the sheet covers the entire set of nodes. Only the outer nodes are partially covered by the blob. To measure whether a node is covered by the sheet we assess the number of particles in a 5×5 window around each node. If the number of particles is < 15 then the node is classified as uncovered and the node indicator is set to green. Otherwise the node is classified as covered and the node indicator is set to red. At each scheduler step the indicators of all nodes are checked. When all nodes are set to green, all nodes underneath the blob are partially uncovered and the shrinkage is stopped.

9.3.3 Reading the Result of the Computation

To trace the path of cities in the tour discovered by the blob a manual process is used. The collection of partially uncovered nodes and blob shape may

be interpreted as an island shape with the nodes representing cities on the coastline of the island (Fig. 9.3). We begin by selecting the city at the top of the arena. If more than one city is at this y location the left-most city at this y location is selected. This city is the start city of the tour and is added to the tour list **T**. Moving in a clockwise direction we trace the perimeter of the blob (walking around the shore of the island . . .). Each time we encounter a city, it is added to **T**. If a city is subsequently re-encountered (as in the case of narrow peninsula structures as described below) it is ignored. When the path reaches the starting city the tour is complete and the list in **T** represents the tour of the TSP found by the shrinking blob.

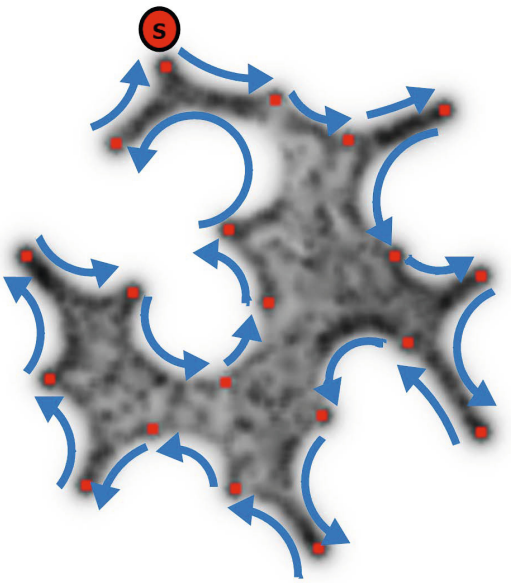


Fig. 9.3 Reading the TSP tour formed by the shrinking blob by perimeter tracking. (a) Tracking is initialised at the top most node. Perimeter of blob is traced in a clockwise direction. Each time a node is encountered for the first time it is added to the tour. The tour is completed when the start node is re-encountered.

Some special cases in the tracking process must be noted in the case where a city lies on a narrow 'peninsula' of the blob as indicated in Fig. 9.4. In Fig. 9.4a the city nearest position x in the path lies close to one side of a narrow peninsula. However the side at which the city is located can be deduced by a small convex bulge on the left side of the blob. In this case the city is not added until it is encountered on the left side of the peninsula. In the case of Fig. 9.4b, however, the city at x is located exactly in the middle of a peninsula and its closest side cannot be discerned. In this instance two interpretations are possible and the subsequent differences in possible tour paths are indicated

by the dotted lines in Fig. 9.4b, i) and ii). In interpretation i) the city is added to **T** immediately and in ii) it is not added until it is encountered on its opposite side. If this situation occurs during the tracking process we add the city to **T** when it is first encountered.

9.4 Results

We assessed the shrinking blob method by generating 20 datasets, each consisting of 20 randomly generated nodes within a circular arena in a 200×200 lattice. To aid the manual tracking process we added the condition that points must have a separation distance of at least 25 pixels. For each run a population of particles was generated and initialised within the confines of the convex hull (algorithmically generated) of the point set. Any particles migrating out of the convex hull area were removed. As the shrinkage process started the cohesion of the blob emerged and, as shrinkage progressed, the blob adapted to the shape of the city nodes. Ten experimental runs were performed on each dataset and the resulting blob shape was recorded and tracked by the manual tracking process to reveal the tour. The best, worst and mean performance over 10 runs for each 20 datasets was recorded and these results are shown in Fig. 9.5. Results of the shrinking blob method (Fig. 9.5, circles with standard deviation bars) are compared to the shortest exact tour (Fig. 9.5, diamonds) computed by the Concorde TSP solver [217].

Over the 20 datasets tested, the mean tour lengths found by the shrinking blob method was 6.41% longer than the exact minimum TSP tours. The mean best performance over all datasets was 4.27% longer than the exact tours and the mean worst performance was 9.22% longer than the minimum tours. There is significant variation in the performance of the blob method on different datasets. In some instances the minimum blob tour length is very close (0.45% longer) to the minimum tour whereas in other cases it is significantly more (20.13% longer). As indicated in Fig. 9.5 there are also significant differences between the *variations* in performance on the same dataset. Datasets 3 and 16 gave identical tours over their ten runs (1.84% and 0.45% longer than the minimum tour respectively), whereas the performance on dataset 7 ranged from between 7.72% and 20% longer than the minimal tour.

9.4.1 Tour Construction by Concavity Insertion

Although the final tour list is read off by tracking the perimeter of the shrunken blob, the construction of the tour actually occurs by an insertion process as the blob shrinks. The blob is initially patterned with the shape of the convex hull. This is only a partial tour, since only the peripheral nodes which are part of the Convex Hull are included. By recording the stages by which nodes are uncovered and added during the shrinkage process, the

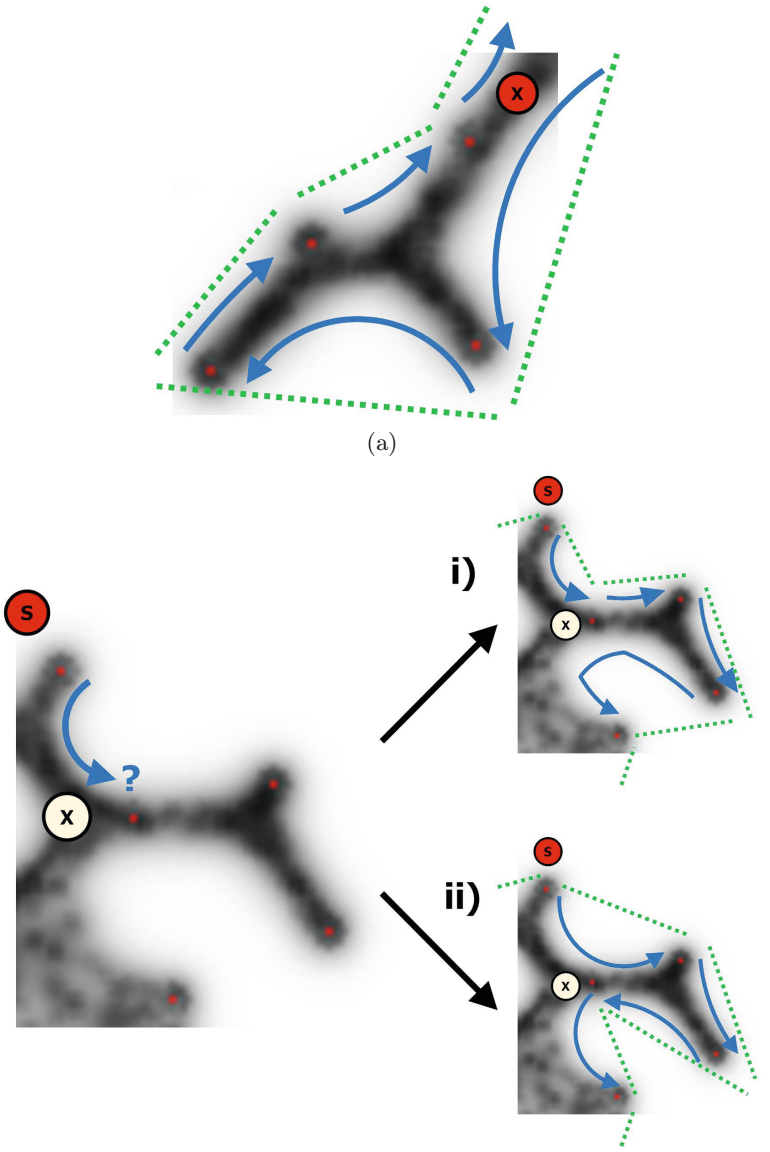


Fig. 9.4 Special cases of when nodes are located on a narrow peninsula, close, or equidistant from either side of the 'land'. (a) The node at 'x' is close to the middle of a narrow portion of the blob. The slight convex bulge in the blob indicates that it is closest to the left side and the node is not added to the tour until it is encountered on the left side. (b) The node at 'x' is directly in the middle of a narrow portion of the blob. Two potential tours are possible, shown in i) and ii) with their respective tours as dotted lines. If this case occurs, the node is added to the tour the first time it is encountered, as in i).

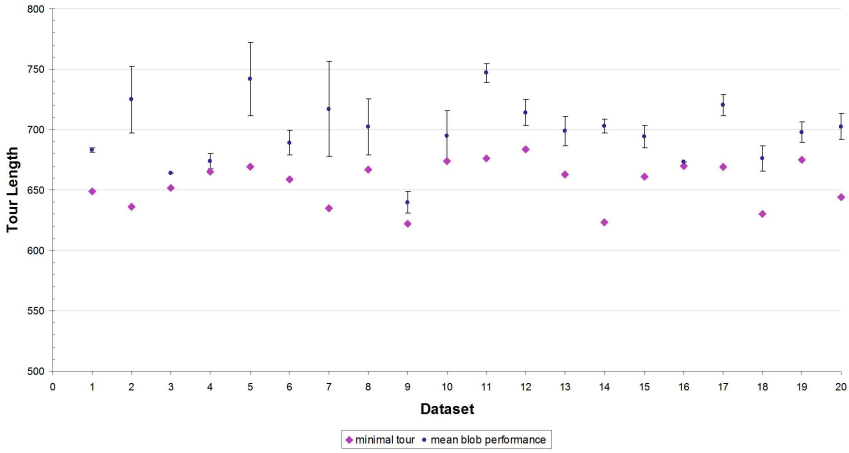
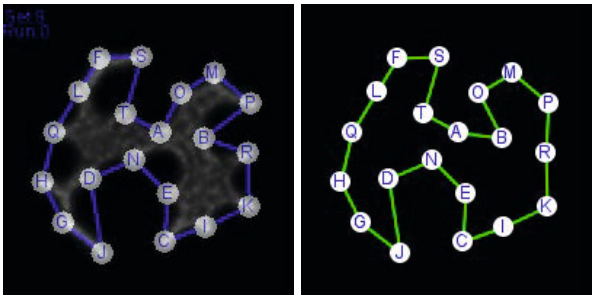


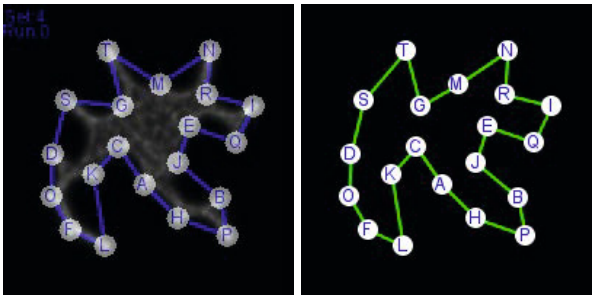
Fig. 9.5 Results of shrinking blob method over 10 runs on each of 20 randomly generated datasets of 20 points compared to exact results from the Concorde TSP solver. Mean tour length indicated by dark circles, standard deviation indicated by error bars, minimal TSP tour by TSP solver indicated by lighter diamonds.

method of construction can be elucidated. Fig. 9.8 shows the visual deformation of the Convex Hull structure as the blob shrinks and new city nodes are added to the list. Note that the blob shrinks simultaneously from all directions and the order of insertion is related to both the proximity of the point from the periphery of the blob and the distance between two outer stimuli at the current periphery of the blob where a concavity forms (discussed further in Section 9.4.3). The actual order of insertion of cities in this example is given in Fig. 9.9.

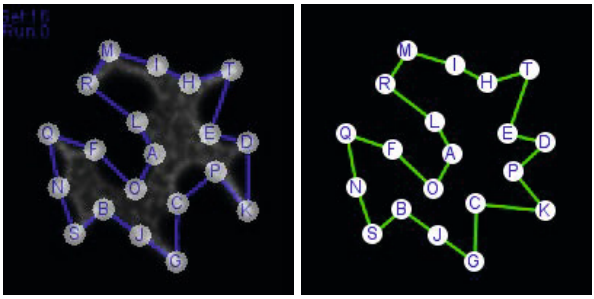
As the blob shrinks, concavities form in the periphery of the blob which move inwards to the centre of the blob shape. The concave deformation is a transformation of the Convex Hull (**CH**) into a Concave Hull (**OH**). The Concave Hull, the area occupied by — or the 'shape' of — a set of points is not as simple to define as its convex hull. It is commonly used in Geographical Information Systems (GIS) as the minimum region (or footprint [218]) occupied by a set of points, which cannot, in some cases, be represented correctly by the convex hull [219]. The Concave Hull is related to the structures known as α-shapes [220]. The α-shape of a set of points, P, is an intersection of the complement of all closed discs of radius $1/\alpha$ that includes no points of P. An α-shape is a convex hull when $\alpha \to \infty$. When decreasing α, the shapes may shrink, develop holes and become disconnected, collapsing to P when $\alpha \to 0$. A concave hull is non-convex polygon representing area occupied by P and the concave hull is a connected α-shape without holes. In contrast to α-shapes, the blob (more specifically, the set of points which it covers) does not become disconnected as it shrinks. As the blob adapts its



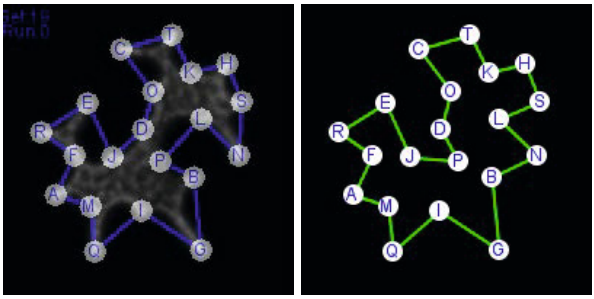
(a) Blob 670 (0.45%) (b) Concorde 667



(c) Blob 672 (1.05%) (d) Concorde 665

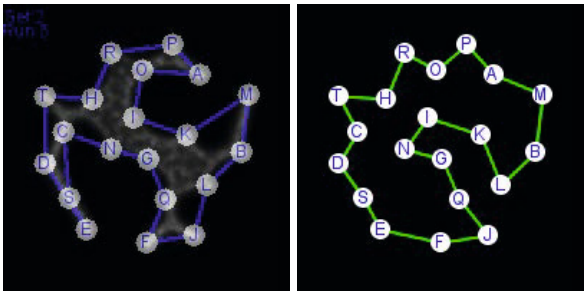


(e) Blob 673 (0.45%) (f) Concorde 670

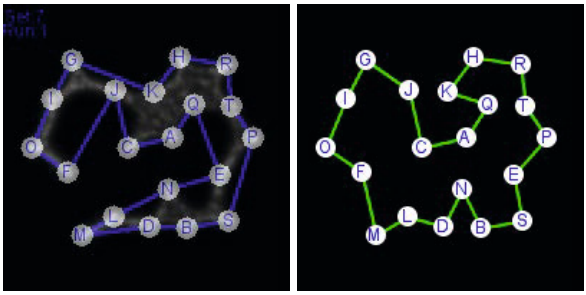


(g) Blob 683 (1.19%) (h) Concorde 675

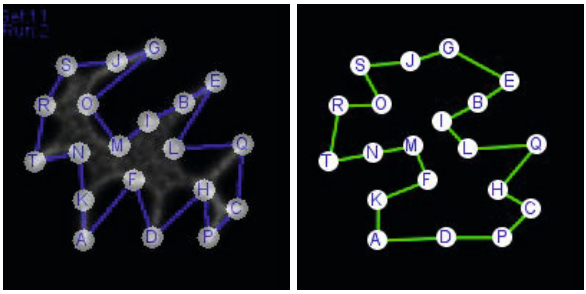
Fig. 9.6 Examples of good performance by the shrinking blob method. (a,c,e,g) Final blob shape with TSP tour overlaid, tour length and percentage greater than exact tour in parentheses, (b,d,f,h) Minimum exact tour found by the Concorde TSP solver.



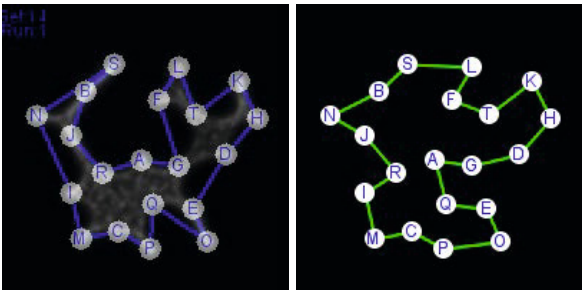
(a) Blob 742 (16.67%) (b) Concorde 636



(c) Blob 762 (20.00%) (d) Concorde 635



(e) Blob 761 (12.57%) (f) Concorde 676



(g) Blob 713 (14.45%) (h) Concorde 623

Fig. 9.7 Examples of relatively poor performance by the shrinking blob method. (a,c,e,g) Final blob shape with TSP tour overlaid, tour length and percentage greater than exact tour in parentheses, (b,d,f,h) Minimum exact tour found by the Concorde TSP solver.

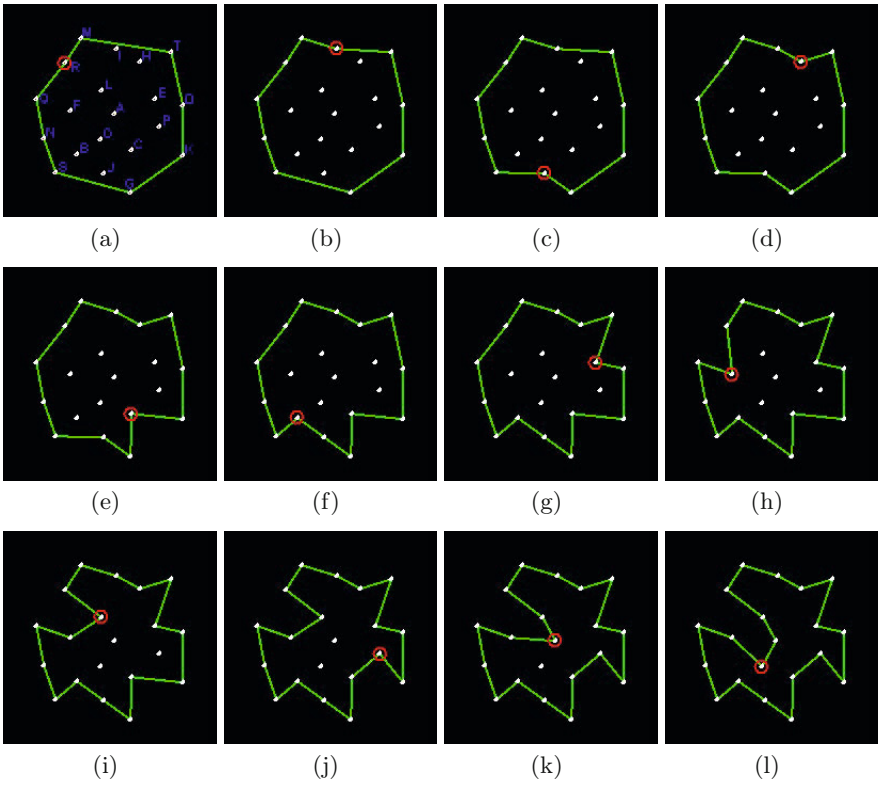


Fig. 9.8 Construction of TSP tour by shrinking blob includes the transformation between Convex Hull and Concave Hull. (a) Initial Convex Hull of dataset 16 (shown by points connected by path) is deformed to a concave shape by the shrinking blob. The stepwise construction of the tour is indicated by adding a circled point as each new city is discovered, (b-k) As the blob continues to shrink new points are included (circled) further reducing the area of the Concave Hull, (l) when shrinkage stops the set of encompassed points is a tour of the TSP.

morphology from Convex Hull to TSP is demonstrates increased concavity with decreased area. Although the shrinkage process is automatically stopped when a TSP tour is formed, the process could indeed continue past the TSP. If shrinkage continues then the blob (now adopting a network shape) will approximate the Steiner minimum tree (SMT), the minimum path between all nodes. As demonstrated in [221] the additional Steiner nodes in the SMT may be removed by increasing the attractant projection from the data nodes. The material adapts to the increased attractant concentration by removing the Steiner nodes to approximate the Minimum Spanning Tree (MST).

M T D K G S N Q
M T D K G S N Q **R**
M **I** *T D K G S N Q R*
M I T D K G **J** *S N Q R*
M I **H** *T D K G J S N Q R*
M I H T D K **C** *G J S N Q R*
M I H T D K C G J **B** *S N Q R*
M I H T **E** *D K C G J B S N Q R*
M I H T E D K C G J B S N Q **F** *R*
M I H T E D K C G J B S N Q F **L** *R*
M I H T E D K **P** *C G J B S N Q F L R*
M I H T E D K P C G J B S N Q F **A** *L R*
M I H T E D K P C G J B S N Q F **O** *A L R*

Fig. 9.9 Gradual construction of tour by city insertion during shrinkage process in dataset 16, as visualised in Fig. 9.8. Top row shows initial configuration of blob as convex hull. Each row inserts a new city (bold) into the tour as indicated in Fig. 9.8.

9.4.2 Blob TSP Tour is a Waypoint from Convex Hull to Spanning Tree

The insertion process of adding nodes to the Convex Hull reveals an orderly transition to the TSP which continues after further shrinkage, leading to the following finding.

The evolution of the blob shape by morphological adaptation is a transition from **CH** to **OH** to **TSP** to **MST** to **SMT**.

We do not explicitly include α-shapes in this transition since α-shapes can include holes and disconnected structures, which do not form in a defect-free shrinking blob. This transition is based on increasing concavity and decreasing area, and encompasses the a blob TSP tour **bTSP** as part of the hierarchy. Note that the blob tour **bTSP** is only one instance of the set of possible TSP tours **TSP** and is not guaranteed to be the minimal tour. The blob TSP tour is only a transient structure — a waypoint — in the natural shrinkage process (we halt the computation at this point merely because we are interested for the purposes of this report).

It is known from Toussaint that there is a hierarchy of proximity graphs (graphs where edges between points are linked depending on measures of neighbourhood and closeness) [188]. Each member of the hierarchy adds edges and subsumes the edges of lower stages in the hierarchy, and some common graphs (see Fig. 9.10a-e) include the Delaunay triangulation **DTN** to Gabriel Graph **GG** to Relative Neighbourhood Graph **RNG** to Minimum Spanning Tree **MST**. Also shown is the shortest possible tree between all nodes formed by adding extra Steiner nodes (Fig. 9.10f). It was found in [80] that *Physarum* approximates the Toussaint hierarchy of proximity graphs as it constructs

transport networks during its foraging and it was demonstrated in [221] that multi-agent transport networks mimicking the behaviour of *Physarum* also minimise these proximity graphs by following this hierarchy in its downwards direction. From a biological perspective traversing the Toussaint hierarchy suggests a mechanism by which *Physarum* can exploit the trade-off between foraging efficiency (many network links) and transport efficiency (fewer but fault tolerant transport links). This mechanism, may also be present in terms of maximising foraging area searched (exploration) and minimising area for efficient transport (exploitation), as suggested in [123]. We suggest that the hierarchy we observed in the shrinking blob from **CH** to **OH** to **TSP** to **MST** to **SMT** may encompass such an area-based exploration-exploitation mechanism (Fig. 9.10g-l). It is notable that there is some overlap between the Toussaint hierarchy and the shrinking blob hierarchy where deepening concavities in the blob hierarchy appear to correspond to the deletion of outer edges in the Toussaint hierarchy, suggesting that there may be some formal relationship between the two. This possible relationship may suggest further studies.

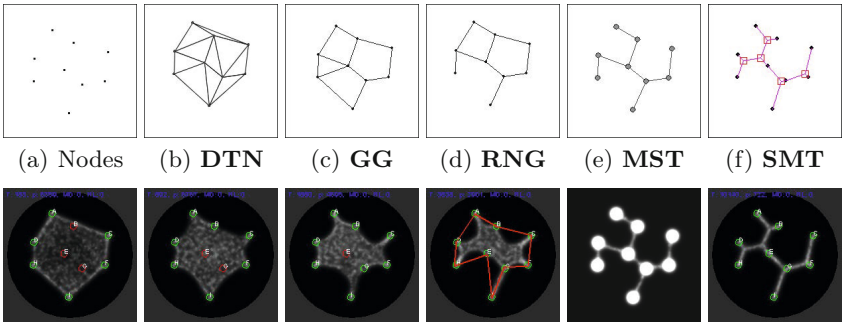


Fig. 9.10 Comparison of the Toussaint Hierarchy (top row) and Shrinking Blob Hierarchy (bottom row). (a) Initial source nodes, (b) Delaunay Triangulation (DTN), (c) Gabriel Graph (GG), (d) Relative Neighbourhood Graph (RNG), (e) Minimum Spanning Tree (MST), (f) Steiner Minimum Tree (SMT), (g) Initial blob is patterned as a Convex Hull (CH), (h-i) as the blob shrinks it adopts the Concave Hull, (j) after uncovering the last node a TSP tour is formed, (k) blob can be forced to adopt MST by increasing node concentration, (l) the 'natural' end point of blob shrinkage is the SMT.

9.4.3 Variations in Performance

The results of the shrinking blob method show variations in performance from very good approximations of close-to-minimum tours (Fig. 9.6) to less successful tours (Fig. 9.7). What is the reason for the disparity in performance

on these datasets? If we examine the tour paths we can glean some clues as to the difference in performance. In the 'good' results examples the major concave regions of the tour formed by the blob closely match the concavities in the exact computed TSP tour (e.g. Fig. 9.6a and b). However in the 'poor' approximation results we can see that the major concave regions of the blob tour do not match the major concavities in the respective exact computed tours (e.g. Fig. 9.7a and b). Given that these concave regions are formed from the deformation of the initial Convex Hull we can see that the concavities in the blob tour appear to be formed, and deepened where there are larger distances between the cities on the initial Convex Hull.

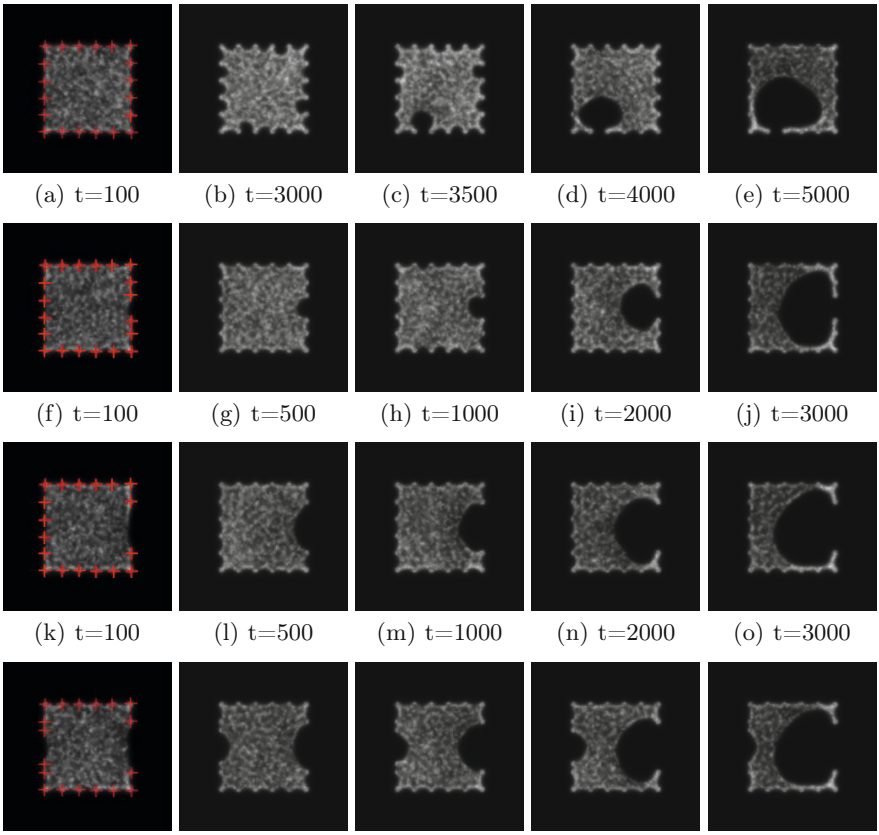
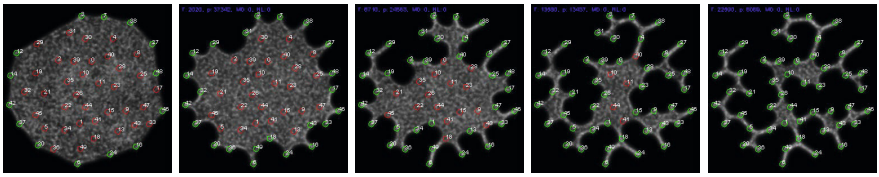


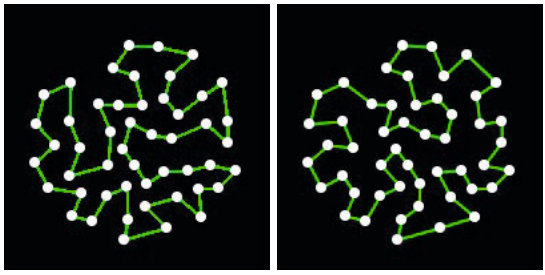
Fig. 9.11 Distance between nodes affects position and speed of concavity formation in shrinking blob. Nodes arranged in the border of square (indicated by crosses on leftmost images). (a-e) all nodes have identical distance of 20 pixels, (f-j) right side has node gap of 30 pixels, (k-0) right side node gap 60 pixels, (p-t) left side node gap 40 pixels, right side node gap 60 pixels.

To explore the role of distance on concavity formation we patterned a blob into a square shape by placing regularly placed stimuli around the border of a square (Fig. 9.11a, stimuli positions, 20 pixels apart, indicated by crosses). When shrinkage of the blob was initiated there is no difference between the stimuli distances. All regions between stimuli initially show small concavities (the 'perforations' in Fig. 9.11b) until one gradually predominates and extends inwards. Also of note is the fact that when one concave region predominates, the other concavities shrink (Fig. 9.11c-e). The position of the initial dominating concavity is different in each run (presumably due to stochastic influences on the collective material properties of the blob) and this may explain the small differences in performance on separate runs using the same dataset.

When there is a larger gap between stimulus points the predominating concavity forms more quickly and is larger. This is shown in Fig. 9.11f-j which has a gap of only 30 pixels between neighbouring stimulus points on the right side of the square and in Fig. 9.11k-o which has a gap of 60 pixels between neighbouring stimulus points. The shorter distance between points in (f-j) generates more tension in the sheet, prevent its deformation. The larger distance between stimuli in (k-o) results in less tension in the sheet at this region and the sheet deforms to generate the concavity.



(a) t=44 (b) t=2020 (c) t=6710 (d) t=13580 (e) t=22590



(f) Blob 1045 (6.42%) (g) Concorde 982

Fig. 9.12 The performance of the shrinking blob method on a larger dataset. (a-e) evolution of the shrinking blob on 50 node dataset, (f) tour formed by shrinking blob and percentage longer than optimal tour shown in parentheses, (g) exact minimum tour by Concorde solver.

When there are multiple instances of large distances between stimuli there is competition between the concave regions and the larger region predominates. This is demonstrated in Fig. 9.11p-t which has a distance gap of 40 pixels on the left side of the square and 60 pixels on the right side. Although two concave regions are formed, the larger deepens whilst the smaller concave region actually shrinks as the blob adapts its shape.

The synthetic examples illustrate the influence of city distance on concavity formation and evolution and these effects are more complex when irregular arrangements of city nodes are used. This is because arrangements of cities present stimuli to the blob sheet when partially uncovered, acting to anchor the blob at these regions, and the morphological adaptation of the blob is thus dynamically affected by the changing spatial configuration of uncovered city nodes. In the examples of relatively poor approximation of the minimum tour (Fig. 9.7) the initial incorrect selection of concavities are subsequently deepened by the shrinking process, resulting in tours which differ significantly in both their visual shape and in the city order from the optimum tour. In the examples of good comparative performance with the exact solver the blob tours differ only in a small number of nodes.

Although outright performance is not the focus of this report, we tested the blob method on a randomly generated dataset of 50 nodes in a preliminary assessment of scalability. We found that over 10 runs the blob method's best result was 6.42% longer than the exact minimum tour computed by the TSP solver. The worst result was 9.88% longer than optimal and the mean result was 7.57% longer than optimal (see Fig. 9.12 for an example). These preliminary results suggest that the method may scale well, however the scalability of the approach requires further investigation. It is worth noting that, in comparison to the shrinking blob approach, previous results using material computation with *Physarum* have been limited to 4 and 8 cities so, even at this early stage, the results obtained by the shrinking blob method — although not optimal — are promising, and may prove amenable to further improvement.

9.5 Summary

We have presented a simple material-based approach to the computation of the Travelling Salesman Problem using a shrinking blob. The method utilises the emergent morphological adaptation properties of a virtual material arising from local interactions within a multi-agent particle system. We shrink this material over time and its deformation and adaptation to the projected data points yields a tour of the TSP. We should again emphasise that the method is notable for its simplicity and novelty rather than its performance. Indeed the performance, when compared to exact TSP solvers or leading heuristic methods, compares relatively unfavourably in terms of absolute tour distance. The method does, however, contain a number of properties that are

intriguing. Firstly, unlike many other nature inspired approaches, the method is not population based and the blob only computes a single instance of a TSP tour. In addition, no attempt is made to modify or optimise the tour. The benefits of population based approaches are that very large search spaces in combinatorial optimisation problems can be traversed and candidate solutions can be compared in some way. This allows the efficient pooling of good solutions, generation of new candidate solutions and avoids local minima. The blob method does not contain any of these beneficial features.

How then, can a shrinking blob naturally generate a good quality (albeit non optimal) TSP tour? The intrinsic performance of the blob is based on its (virtual) material properties which exhibit innate minimisation behaviour. Previous research has demonstrated the network minimisation properties of the material approach, reproducing phenomena seen in soap film evolution [221] and lipid nanotube networks [111]. The maintenance of uniform shape during blob shrinkage also allows no crossing over of paths and it is known from [222] that crossing paths produce non-optimal tours. More specifically, the tour is constructed by insertion of cities into the list as concave regions are formed in the initial Convex Hull pattern and subsequently deepened. This is similar to algorithmic heuristics which, beginning with a Convex Hull, add cities to the list based on certain cost criteria [223], [224]. In the case of the blob, however, there is no explicit consideration of cost when adding cities to the tour. The mechanism of insertion selection in the blob (by deepening concavities) is intrinsic to its quasi-mechanical properties of the 'material' which are influenced by the depth of the city to the Convex Hull boundary and the span distance between boundary stimuli.

Although the blob approach differs from population based nature-inspired heuristics it is reminiscent, in character if not in direct operation, with other analogue based methods. In the Elastic Net algorithm, introduced by Durbin and Willshaw, a circular band is initialised at the approximate centre of a pattern of source TSP nodes. The band is expanded iteratively whilst two forces are applied to points on the band which attempt to minimise distances between cities on the band and the overall length of the band itself [209]. In the conceptually opposite approach of [225] the band is initialised on the Convex Hull and the two forces attempt to constrict the band whilst attracting the band towards a city. The tour formed by the band is then subject to a second 'non-deterministic improvement' algorithm to escape local minima. The main difference between the blob method and these 'band' approaches is that the material properties of the blob method are an emergent property of, and are distributed within, local interactions between components of the material. The computation is thus an embodied property of the material itself.

Can the mechanism underlying the simple material approximation of the TSP in the blob approach contribute to the question of human performance on the TSP? MacGregor and Ormerod noted that humans produced efficient results on the TSP [226] and this finding stimulated further research into

human performance on the problem and possible perceptual and cognitive mechanisms. In their analysis of experimental findings using human subjects Ormerod and Chronicle noted that global perceptual influences appear to play a role in human approximation of the TSP [227]. MacGregor et al. suggested a model based on insertion of cities into the Convex Hull [214]. This model is similar to the shrinking blob mechanism except that in the blob approach the addition of cities occurs in parallel whereas in the MacGregor et al. model it is a sequential process. The blob method also exhibits another property found in optimal tours, that boundary points in the original convex hull are connected in sequence (the sequence may, of course, be interrupted by interior points). Other competing models to explain human TSP performance exist, including variants of hierarchical pyramid models [212], [215] and the global-local model proposed by Best [228]. Merits, problems, biological plausibility and the role of local vs. global perceptual processes of the competing models have been the subject of lively debate and an assessment is beyond the scope of this chapter, but see the review in [211] for an overview. In this review MacGregor states that, despite the growing interest in research into the human performance on the TSP, and combinatorial optimisation problems in general: "As yet, no algorithms have been put forward to explain performance on the MSTP [Minimum Spanning Tree Problem] and the GSTP [Generalised Steiner Tree Problem]...". It is notable that the shrinking blob method incorporates approximations of all three problems and executes a natural transition from global to local 'perception' using material properties which emerge from very simple low-level and bottom-up interactions. Whether this natural computation employed by the blob is of interest, or utility, to human combinatorial optimisation problems is, however, an open question.

Limitations of the approach, as it stands, include its noted relatively modest performance and the reliance of a manual method to interpret the result of the blob computation. Manual interpretation of the blob tour is, of course, open to experimenter bias and for this reason a methodical process must be followed, as described in section 9.3.3. An automated method of tracking the perimeter and 'reading' the result of the blob tour would, nevertheless, be of benefit. The shrinkage process is an innate and emergent phenomenon generated by the particle interactions and is not itself affected by the number of data points in the lattice. However, the performance of the underlying system generating the material behaviour is slowed by increases in area and in particle population size. Although the shrinkage process does indeed innately approximate TSP tours the manual reading of the tour path incurs some (human) computational demands. It is difficult to quantify these demands in terms of classical computational complexity metrics. Providing that the result reading procedure is followed correctly any increase in problem scale should — in theory — show a linear increase in readout time. However this may be hampered by the increasing likelihood of mistakes caused by increasing path tortuosity, fatigue and even the repetitive and somewhat tedious nature of

the procedure. Again, the development of a suitably accurate automated of reading the result would aid the assessment of scalability and performance in future research.

Further work, including a comprehensive evaluation of model parameters affecting the material properties of the blob may suggest methods by which the basic features of the shrinking blob approach may be adapted, or improved, to improve the performance in comparison with leading heuristic methods. The material properties and computation of the blob emerge from a population of simple multi-agent particles and it would be satisfying if this virtual material could be implemented and embodied in a real physical substrate with the desired physical (for example visco-elastic, free energy minimisation) properties. Alternately it may be possible to translate the material operation of the unconventional computation blob method into a classical algorithmic method.