Brief Announcement: Shape Formation by Programmable Particles*

Giuseppe A. Di Luna[1], Paola Flocchini[2], Nicola Santoro[3], Giovanni Viglietta[4], and Yukiko Yamauchi[5]

1 **University of Ottawa, Ottawa, Canada**
gdiluna@uottawa.ca
2 **University of Ottawa, Ottawa, Canada**
paola.flocchini@uottawa.ca
3 **Carleton University, Ottawa, Canada**
santoro@scs.carleton.ca
4 **University of Ottawa, Ottawa, Canada**
gvigliet@uottawa.ca
5 **Kyushu University, Fukuoka, Japan**
yamauchi@inf.kyushu-u.ac.jp

Abstract

Shape formation is a basic distributed problem for systems of computational mobile entities. Intensively studied for systems of autonomous mobile robots, it has recently been investigated in the realm of programmable matter. Namely, it has been studied in the geometric Amoebot model, where the anonymous entities, called particles, operate on a hexagonal tessellation of the plane, have constant memory, can only communicate with neighboring particles, and can only move from a grid node to an empty neighboring node; their activation is controlled by an adversarial scheduler. Recent investigations have shown how, starting from a well-structured configuration in which the particles form a (not necessarily complete) triangle, the particles can form a large class of shapes. This result has been established under several assumptions: agreement on the clockwise direction (i.e., chirality), a sequential activation schedule, and randomization.

In this paper we provide a characterization of which shapes can be formed deterministically starting from any simply connected initial configuration of n particles. As a byproduct, if randomization is allowed, then any input shape can be formed from any initial (simply connected) shape by our algorithm, provided that n is large enough. Our algorithm works without chirality, proving that chirality is computationally irrelevant for shape formation. Furthermore, it works under a strong adversarial scheduler, not necessarily sequential. We also consider the complexity of shape formation both in terms of the number of rounds and the total number of moves performed by the particles executing a universal shape formation algorithm. We prove that our solution has a complexity of $O(n^2)$ rounds and moves: this number of moves is also asymptotically optimal.

1998 ACM Subject Classification F.1.1 Models of Computation, F.2.2 Nonnumerical Algorithms and Problems, I.2.11 Distributed Artificial Intelligence, I.2.9 Robotics

Keywords and phrases Shape formation, pattern formation, programmable matter, Amoebots, leader election, distributed algorithms, self-assembly

Digital Object Identifier 10.4230/LIPIcs.DISC.2017.48

* Full version available at https://arxiv.org/abs/1705.03538.

31st International Symposium on Distributed Computing (DISC 2017).
Editor: Andréa W. Richa; Article No. 48; pp. 48:1–48:3

Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

1 Background

The term *programmable matter*, introduced by Toffoli and Margolus [4], is used to denote matter that has the ability to change its physical properties in a programmable fashion, based upon user input or autonomous sensing. Often programmable matter is envisioned as a very large number of very small locally interacting computational particles, programmed to collectively perform a complex task. Such particles could have applications in a variety of important situations: smart materials, minimally invasive surgery, etc.

Of particular interest, from the distributed computing viewpoint, is the *geometric Amoebot* model. In this model, introduced in [3] and so called because inspired by the behavior of amoeba, programmable matter is viewed as a swarm of decentralized autonomous self-organizing entities, operating on a hexagonal tessellation of the plane. These entities, called *particles*, are constrained by having simple computational capabilities (they are finite-state machines), strictly local interaction and communication capabilities (only with particles located in neighboring nodes of the hexagonal grid), and limited motorial capabilities (from a grid node to an empty neighboring node); furthermore, their activation is controlled by an adversarial (but fair) synchronous scheduler. A feature of the Amoebot model is that particles can be in two modes: *contracted* and *expanded*. When contracted, a particle occupies only one node, while when expanded the particle occupies two neighboring nodes; it is indeed this ability of a particle to expand and contract that allows it to move on the grid.

The pioneering study of [1] on *shape formation* in the geometric Amoebot model showed how particles can build simple shapes, such as a hexagon or a triangle. Subsequent investigations [2] have recently shown how, starting from a well-structured configuration in which the particles form a (not necessarily complete) triangle, they can form a larger class of shapes under several assumptions, including randomization (which is used to elect a leader), chirality, and a sequential activation schedule (i.e., at each time unit the scheduler selects only one particle which will interact with its neighbors and possibly move). Notice that, without the availability of a unique leader (provided by randomization), dropping the chirality assumption becomes a problem with a non-sequential schedule.

2 Our Contributions

We continue the investigation, significantly extending the existing results. Among other things, we provide a constructive characterization of which shapes S_F can be formed *deterministically* starting from an unknown *simply connected* initial configuration S_0 of n particles (i.e., a connected configuration without "holes").

As in [2], we assume that the size of the description of S_F is constant with respect to the size of the system, so that it can be encoded by each particle in a part of its internal memory. Such a description is available to all the particles at the beginning of the execution as their "input". The particles will form a final configuration that is an appropriate scaling, translation, rotation, and perhaps reflection of the input shape S_F. Since all particles of S_0 must be used to construct S_F, they may have to scale up S_F in order to fit: we stress that an appropriate scale factor is unknown to particles, and they must determine it autonomously. (We assume that the input shape S_F that is actually given to the particles is the smallest possible among the scaled copies of itself that can be embedded in the hexagonal grid.)

Given two shapes S_0 and S_F, we say that the pair (S_0, S_F) is *feasible* if there exists a deterministic algorithm that, in every execution (thus, regardless of the activation schedule), allows the particles to form a scaled copy of S_F starting from S_0, and no longer move. Our characrerization of feasibility is based on symmetries that are unbreakable: a shape is said to be *unbreakably k-symmetric*, for some integer $k > 1$, if it has a center of k-fold rotational symmetry that does not coincide with any vertex of the hexagonal grid.

▶ **Theorem 1.** *If (S_0, S_F) is a feasible pair and S_0 is unbreakably k-symmetric, then S_F is also unbreakably k-symmetric.*

Interestingly, all the pairs not excluded by the above theorem turn out to be feasible (provided that the size of S_0 is large enough with respect to the size of S_F), and for them we give a *universal shape formation algorithm*: this algorithm does not need any information on S_0, except that it is simply connected. The algorithm first elects 1, 2, or 3 leaders among the particles. Electing a unique leader may be impossible due to the symmetry of S_0: if $k > 1$ leaders are elected, it means that S_0 is necessarily unbreakably k-symmetric. Each leader takes an equal portion of S_0 and rearranges it into a straight line. Then, all leaders reconfigure their respective lines to form a portion of S_F, scaled up by an appropriate factor. The optimal factor is computed by each leader by simulating a Turing machine on its line of particles: the leader acts as the head, and uses the particles as memory cells on a tape.

▶ **Theorem 2.** *Let P be a system of n particles forming a simply connected shape S_0. Let S_F be a shape of constant size m that is unbreakably k-symmetric if S_0 is unbreakably k-symmetric. If all particles of P execute the universal shape formation algorithm with input a representation of the final shape S_F, and if n is at least $\Theta(m^2)$, then eventually P forms a scaled copy of S_F, and the particles cease to move.*

The total number of movements performed by the system executing our algorithm is $O(n^2)$, which is asymptotically optimal: indeed, if S_0 is a full hexagon and S_F is a line segment, $\Omega(n^2)$ moves are needed. The number of *rounds* (i.e., periods of time in which each particle is activated at least once) that an execution of our algorithm takes is also $O(n^2)$.

Our algorithm works under a stronger adversarial scheduler than [2], as it activates an arbitrary number of particles at each execution step (i.e., not necessarily just one, like the sequential scheduler). We also need a slightly less demanding communication system. Moreover, in our algorithm, no chirality is assumed: indeed, unlike in [2], different particles may have a different notion of clockwise direction. Because of this difficutly, part of the algorithm is dedicated to a "handedness agreement" procedure. We stress that our results prove that *chirality is computationally irrelevant* for shape formation.

These results concern *deterministic* shape formation. If randomization were allowed, we could always elect a unique leader with arbitrarily high probability, and apply our algorithm to *any* pair of shapes (S_0, S_F) where S_0 is simply connected, regardless of their symmetry. This extends the result of [2], which assumes the initial configuration to be a (possibly incomplete) triangle. Additionally, our notion of shape generalizes the one used in [2], where a shape is only a collection of full triangles, while we include also 1-dimensional segments as its constituting elements. Our technique actually allows us to generalize the concept of shape much further, to include essentially anything that is Turing-computable.

References

1 Z. Derakhshandeh, R. Gmyr, A.W. Richa, C. Scheideler, and T. Strothmann. An algorithmic framework for shape formation problems in self-organizing particle systems. In *Proc. of NanoCom*, pages 21:1–21:2, 2015.

2 Z. Derakhshandeh, R. Gmyr, A.W. Richa, C. Scheideler, and T. Strothmann. Universal shape formation for programmable matter. In *Proc. of SPAA*, pages 289–299, 2016.

3 Z. Derakhshandeh, R. Gmyr, T. Strothmann, R.A. Bazzi, A.W. Richa, and C. Scheideler. Leader election and shape formation with self-organizing programmable matter. In *Proc. of DNA*, pages 117–132, 2015.

4 T. Toffoli and N. Margolus. Programmable matter: concepts and realization. *Physica D*, 47(1):263–272, 1991.