Brief Announcement: On the Exponential Growth of Geometric Shapes

Nada Almalki
Department of Computer Science, University of Liverpool, UK

Siddharth Gupta
Department of Computer Science & Information Systems, BITS Pilani Goa Campus, India

Othon Michail
Department of Computer Science, University of Liverpool, UK

Abstract

We explore how geometric structures (or *shapes*) can be grown exponentially fast from a single node, through a sequence of centralized growth operations, and if collisions during growth are to be avoided. We identify a parameter k, representing the number of turning points within specific parts of a shape. We prove that, if edges can only be formed when generating new nodes and cannot be deleted, trees having $O(k)$ turning points on every root-to-leaf path can be grown in $O(k \log n)$ time steps and spirals with $O(\log n)$ turning points can be grown in $O(\log n)$ time steps, n being the size of the final shape. For this case, we also show that the maximum number of turning points in a root-to-leaf path of a tree is a lower bound on the number of time steps to grow the tree and that there exists a class of paths such that any path in the class with $\Omega(k)$ turning points requires $\Omega(k \log k)$ time steps to be grown. In the stronger model, where edges can be deleted and neighbors can be handed over to newly generated nodes, we obtain a universal algorithm: for any shape S it gives a process that grows S from a single node exponentially fast.

2012 ACM Subject Classification Theory of computation → Computational geometry; Theory of computation → Design and analysis of algorithms

Keywords and phrases centralized algorithm, growth process, collision, programmable matter

Digital Object Identifier 10.4230/LIPIcs.SAND.2024.23

Related Version *Full Version*: https://arxiv.org/abs/2307.04385 [2]

1 Introduction

A *growth operation* (also called *doubling* [3] or *expansion* [8]) applied on a node u of a geometric shape S, generates a new node in one of the points adjacent to u and possibly translates some part of the shape. In this work, we explore the following two interrelated questions: *"What are the structural properties associated with exponential growth of geometric shapes?"* and *"How can some of these properties be exploited and others avoided in order to design algorithms that can grow desired shapes exponentially fast?"*

Though our model takes inspiration from natural growth processes, it also shares features with existing theoretical models of computation and robotics. Growth is a defining property of both our model and self-assembly models. In the majority of self-assembly models, growth is through passive attachment [7, 10] on the external layer of the formed structure, and, thus, is relatively slow. Our algorithms can actively control the structure's growth without any *a priori* limitation on where to apply the growth operations, resulting in sub-linear and often (poly)logarithmic growth in the size of the final structure. An example of a self-assembly model incorporating active molecular dynamics is the *Nubot* model [11]. A difference between our model and [11] is that our processes are only allowed to update instances through growth. As is also the case in [11], most of our algorithms use the fast process of growing a line as a sub-routine. Recently, there has been growing interest in studying the algorithmic

3rd Symposium on Algorithmic Foundations of Dynamic Networks (SAND 2024).
Editors: Arnaud Casteigts and Fabian Kuhn; Article No. 23; pp. 23:1–23:6
Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

foundations of *programmable matter* systems, focusing on their ability to alter their shape through local reconfiguration [1, 4, 6, 9]. The growth processes that we study could serve as a way to deploy programmable matter fast, either in its exact initial configuration or in a rough version of it that can be then refined through other types of operations. An assumption of our model is that individual operations have *linear strength*, meaning that they have enough power to move any part of the structure. This is a common simplifying assumption in the relevant literature [5, 11, 4] and can sometimes be dropped, e.g., when more than one operation can be applied in parallel. Our model also has some relevance to von Neumann's concept of self-replicating machines.

2 Models and Problem

We consider a two-dimensional square grid, each point of which is identified by its $x \geq 0$ and $y \geq 0$ integer coordinates, x indicating the column and y the row. A *shape* is defined as a graph $S = (V, E)$ drawn on the grid. V is a set of n nodes, where each node u occupies a distinct point (u_x, u_y) of the grid. $E \subseteq \{uv \mid u, v \in V \text{ and } u, v \text{ are adjacent}\}$ is a set of edges between pairs of adjacent nodes, where two nodes u and v are *adjacent* if their orthogonal distance on the grid is one. Our results hold for any geometry of individual nodes that does not trivially make nearby nodes intersect. A shape is *connected* if the graph that defines it is a connected graph. We restrict attention to connected shapes.

One or more growth operations applied in parallel to nodes of a shape S either cause a *collision* or yield a new shape S'. There are two types of collisions: *node collisions* and *cycle collisions*. We assume that the constructed shapes are equivalent up to translations. Let $T = (V, E)$ be a tree and $u_0 \in V$ its root. A single growth operation applied on a node $u \in V$ toward an adjacent point (x, y), results in either generating a new node u' at point (x, y) and connecting it to u, or, if (x, y) is already occupied by a node v connected to u, generating u' between u and v, connecting it to both u and v, and translating the subtree $T(v)$ by one unit away from u along the axis parallel to uv.

Let Q be a set of operations to be applied *in parallel* to a connected shape S, each operation on a distinct pair of nodes or a node and an unoccupied point. We assume that all operations in such a set are applied *concurrently*, have the same *constant execution speed*, and their *duration* is equal to one *time step*. A *node collision* occurs if the trajectories of any two nodes meet. If S is a connected shape with at least one cycle, then a set of parallel operations Q on S either causes a *cycle collision* or its effect is essentially equivalent to the application of Q on any spanning tree of S. In particular, a *cycle collision* occurs when two parts of a cycle grow unequally. A set of operations is said to be *collision free* if it does not cause any node or cycle collisions.

A growth process σ starts from an initial shape S_0 – often a single node – and, in each time step $t \geq 1$, applies a set of parallel growth operations – possibly a single operation – on the current shape S_{t-1} to give the next shape S_t, until a final shape S is reached at a time step t_f. In this case, we say that σ *grows S from S_0 in t_f time steps.*

The different models and processes we consider are defined as follows.

▶ **Definition 1.** *Let S_t^b and S_t^e denote the shapes formed by the beginning and by the end of time step t, respectively, and assume that $S_1^b = S_0$. A* cycle-preserving *growth process applies a collision free set of parallel growth operations Q_t to S_t^b, for all time steps $t \geq 1$. A* cycle-breaking *growth process additionally removes a – possibly empty – subset of the edges of S_t^b, whose removal does not disconnect the shape, before applying Q_t to it. If* neighbor handover *is allowed, growth of a node u generating a new node u' in direction d can hand* *any neighbor w of u perpendicular to d over to u'. In the* connectivity graph *model, for all $t \geq 1$, $S^b_{t+1} = S^e_t$ holds. In the* adjacency graph *model, for all $t \geq 1$, $S^b_{t+1} = AC(S^e_t)$ holds, where $AC(S)$ is the adjacency closure of a shape S.*

Intuitively, the additional assumption in the adjacency graph model is that, at the end of every time step, the graph model updates the shape by connecting all adjacent nodes that are not connected. Combining the adjacency graph model with cycle-breaking processes captures the less extreme case, in which the process can choose any spanning connected sub-shape of the adjacency closure.

We study a reachability problem between classes of shapes through growth. The definition of the problem is the same for all growth models of Definition 1.

▶ **Problem 1.** *Let $\mathcal{I}$ be a class of initial shapes and $\mathcal{F}$ a class of final shapes. We want to determine a bound τ such that for all $S_0 \in \mathcal{I}$ and all $S \in \mathcal{F}$ there is a growth process σ that grows S from S_0 in τ time steps.*

Given that our focus is on exponential growth, upper bounds must be of the form $\tau = O(\log n)$ or $\tau = (poly)\log n$. As there is a straightforward $\Omega(\log n)$ lower bound, non-trivial lower bounds should be at least $\omega(\log n)$. In all instances of the problem that we study, at least one of $\mathcal{I}$ and $\mathcal{F}$ is a singleton, the initial shape typically being a single node. Our upper bounds are constructive: for each instance of the problem an algorithm is presented, which for every (S_0, S) from the respective classes gives a process that grows S from S_0 in τ time steps.

3 Technical Overview

In this section, we present the main results of our work, starting with the results of the connectivity graph model and then moving on to the adjacency graph model.

Connectivity Graph Model. Starting growth from a single node, the class of shapes that can be grown in this model is limited to tree structures only. We start by focusing on efficiently growing trees. We identify a parameter k, representing the number of turning points within specific parts of a shape. A node u of a shape S is called a *turning point* if either u is a leaf or there are at least two neighbors v_1 and v_2 of u such that v_1u is perpendicular to uv_2. The nodes between any two consecutive turning points of a path form a line segment.

Upper Bounds Trees with $O(k)$ turning points on every root-to-leaf path can be grown in $O(k \log n)$ time steps through a breadth-first search (BFS) on their line segments. Starting from the root u_0, the algorithm proceeds in phases, growing all maximal line segments at distance i in parallel. Each line segment can be grown exponentially fast by doubling the number of nodes in every time step.

▶ **Theorem 2.** *Let T be any tree having $O(k)$ turns on every root-to-leaf path.* BFS on line segments *can grow T from a single node within $O(k \log n)$ time steps in the connectivity graph model.*

Similarly, spirals with $O(\log n)$ turning points can be grown within $O(\log n)$ time steps through a pipelined version of BFS. For further details, see [2].

Lower Bounds. We establish two lower bounds for trees and paths. For trees, we show that the maximum number of turning points in a root-to-leaf path of the tree is a lower bound on the number of time steps to grow the tree from a single node. Initially, we define a growth process σ for $S = (V, E)$ that induces a relation $\rightarrow_\sigma$ on V, where $u \overset{t}{\rightarrow}_\sigma v$ iff node u *generates* node v at time step t. We also write $u \rightarrow_\sigma v$ to mean $u \overset{t}{\rightarrow}_\sigma v$ for some $t \geq 1$ and $u \rightsquigarrow_\sigma v$ iff $u = u_1 \rightarrow_\sigma u_2 \rightarrow_\sigma \cdots \rightarrow_\sigma u_l = v$, for some $l \geq 2$. We omit σ, writing just $u \overset{t}{\rightarrow} v$, $u \rightarrow v$, or $u \rightsquigarrow v$ when the growth process is clear from context or when referring to any growth process. The relation $\rightarrow_\sigma$ defines a graph $G_{\rightarrow_\sigma} = (V, E_{\rightarrow_\sigma})$. Further, we define a relation $\mapsto_\sigma$ induced by $\rightarrow_\sigma$ on the turning points of tree shapes. Given a tree T and a growth process σ for T, for any two turning points u, v of T we write $u \overset{t}{\mapsto}_\sigma v$ iff (i) $u \overset{t}{\rightarrow}_\sigma v$ or (ii) $u \rightsquigarrow u' \overset{t}{\rightarrow}_\sigma v$ and u, u', v are on the same line segment at the end of time step t.

▶ **Lemma 3.** *In the connectivity model, let $T = (V, E)$ be a tree and σ a growth process for T starting from $u_0 \in V$. For any root-to-leaf path $(u_0, u_1, \ldots, u_l)$ of T, where the u_is are restricted to the turning points of the path, $u_0 \mapsto u_1 \mapsto \cdots \mapsto u_l$ holds.*

The theorem below formalizes the lower bound for growing trees.

▶ **Theorem 4.** *Let $T = (V, E)$ be a tree and k any positive integer satisfying that for every root $u_0 \in V$ there is a root-to-leaf path in T containing at least k turning points. Then any growth process σ for T in the connectivity model requires at least $k - 1$ time steps. This lower bound is maximized for the maximum such k.*

For paths, we show that there exists a class of paths such that any path in the class with $\Omega(k)$ turning points requires $\Omega(k \log k)$ time steps to be grown from a single node. Let P be a path with k turning points and $(tp_1, tp_2, \ldots, tp_k)$ be their order in P. Let σ be a process that grows P from a single node. Without loss of generality, we can assume that σ starts from a turning point tp_i of the path P. We prove that the sets $\{tp_{i+1}, tp_{i+2}, \ldots, tp_k\}$ and $\{tp_1, tp_2, \ldots, tp_{i-1}\}$ of turning points are generated in the order $(tp_{i+1}, tp_{i+2}, \ldots, tp_k)$ and $(tp_{i-1}, tp_{i-2}, \ldots, tp_1)$, respectively by σ. Moreover, σ respects the direction of P at every node while generating the next node from it.

Let P be an incompressible (meaning that it has no columns or rows without any turning points) spiral path between u and v with k turning points. Moreover, let u be the internal endpoint of P. The following lemma gives a lower bound on the number of time steps taken by any process that grows P from a single node starting from u.

▶ **Lemma 5.** *Let P be an incompressible spiral path between u and v with k turning points. Moreover, let u be the internal endpoint of P. Let σ be any process that grows P from a single node starting from u. Then, σ requires $\Omega(k \log k)$ time steps.*

Let $(tp_1 = u, tp_2, \ldots, tp_k = v)$ be the order of turning points of P from u to v. We know that σ generates the turning points in the order $(tp_1 = u, tp_2, \ldots, tp_k = v)$. Let GT_j be the time step when the turning point tp_j was generated by σ, for any $j \geq 2$. Let $\widehat{P}(t)$ be the path constructed by σ after time step t. Further, let a and b be two vertices of P. We denote by $P[a, b]$ the path between a and b (including both a and b) of P. Moreover, we denote by $|a - b|_P$ the number of edges in $P[a, b]$. Also, we denote by $X(a, P)$ the x-coordinate of the vertex a in P. To prove the above lemma, we first prove the following lemma about the path constructed by σ.

▶ **Lemma 6.** *For any $j \geq 5$, the path $\widehat{P}(GT_j - 1)$ grown by σ till time step $GT_j - 1$ should be the same as the subpath $P[tp_1, tp_{j-1}]$ of P between $tp_1 = u$ and tp_{j-1}.*

The lower bound follows by applying Lemma 5 on a path consisting of two interleaved spirals of equal size and observing that at least one of the two must be grown from its internal endpoint. See [2] for the full proof.

▶ **Theorem 7.** *Let σ be a process that grows a path from a single node. Then, there exists a path for which σ takes $\Omega(k \log k)$ time steps.*

Adjacency Graph Model. In this model, every pair of adjacent nodes is also connected in the shape. We study both cycle-preserving and cycle-breaking types of processes. For cycle-preserving processes, we cannot directly perform BFS on line segments through cycle-preserving growth due to the dependence between adjacent line segments. We give a modified BFS that overcomes this by growing adjacent line segments in different phases, and we prove that if a shape S has a spanning tree with $O(k)$ turning points on every root-to-leaf path, then the adjacency closure of S can be grown from a single node within $O(k \log n)$ time steps. For a complete description of this approach, see [2]. For cycle-breaking processes with the additional assumption that neighbors can be handed over to newly generated nodes (*neighbor handover*), our main result is an efficient universal algorithm that gives an $O(\log n)$ time steps *growth process* for any connected shape S. The algorithm achieves this by specifying an elimination order of the nodes within the shape and then inverting this order to produce the *growth process*.

▶ **Theorem 8.** *Given any connected shape, S with dimensions $l \times w$, the elimination algorithm grows S from a single node in $O(\log l + \log w)$ time steps.*

References

1 Hugo A. Akitaya, Esther M. Arkin, Mirela Damian, Erik D. Demaine, Vida Dujmović, Robin Flatland, Matias Korman, Belen Palop, Irene Parada, André van Renssen, et al. Universal reconfiguration of facet-connected modular robots by pivots: the O(1) musketeers. *Algorithmica*, 83(5):1316–1351, 2021.

2 Nada Almalki, Siddharth Gupta, and Othon Michail. On the exponential growth of geometric shapes, 2024. `arXiv:2307.04385`.

3 Nada Almalki and Othon Michail. On geometric shape construction via growth operations. *Theoretical Computer Science*, 984:114324, 2024.

4 Abdullah Almethen, Othon Michail, and Igor Potapov. Pushing lines helps: Efficient universal centralised transformations for programmable matter. *Theoretical Computer Science*, 830-831:43–59, 2020.

5 Greg Aloupis, Sébastien Collette, Erik D. Demaine, Stefan Langerman, Vera Sacristán, and Stefanie Wuhrer. Reconfiguration of cube-style modular robots using O(log n) parallel moves. In *International Symposium on Algorithms and Computation (ISAAC)*, pages 342–353, 2008.

6 Zahra Derakhshandeh, Robert Gmyr, Andréa W. Richa, Christian Scheideler, and Thim Strothmann. Universal shape formation for programmable matter. In *Proceedings of the 28th ACM Symposium on Parallelism in Algorithms and Architectures (SPAA)*, pages 289–299, 2016.

7 David Doty. Theory of algorithmic self-assembly. *Communications of the ACM*, 55(12):78–88, 2012.

8 Siddharth Gupta, Marc van Kreveld, Othon Michail, and Andreas Padalkin. Collision detection for modular robots – it is easy to cause collisions and hard to avoid them, 2023. `arXiv:2305.01015`.

9 Othon Michail, George Skretas, and Paul G. Spirakis. On the transformation capability of feasible mechanisms for programmable matter. *Journal of Computer and System Sciences*, 102:18–39, 2019.

10 Matthew J. Patitz. An introduction to tile-based self-assembly and a survey of recent results. *Natural Computing*, 13:195–224, 2014.

11 Damien Woods, Ho-Lin Chen, Scott Goodfriend, Nadine Dabby, Erik Winfree, and Peng Yin. Active self-assembly of algorithmic shapes and patterns in polylogarithmic time. In *Proceedings of the 4th conference on Innovations in Theoretical Computer Science (ITCS)*, pages 353–354, 2013.