Pushing Lines Helps: Efficient Universal Centralised Transformations for Programmable Matter

Abdullah Almethen, Othon Michail and Igor Potapov

Department of Computer Science
University Of Liverpool

ALGOSENSORS 2019
September 12, 2019
Munich, Germany

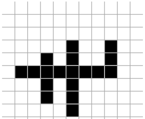
Outline

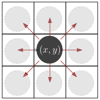
Introduction

- A problem of *shape formation*:
  - A group of n connected entities move in a space to from a goal shape.
- Many systems have been examined.
- *Programmable Matter* is the most recent theoretical area,
  - Refers to any type of matter that can **algorithmically** change its physical properties (e.g., shape, colour, etc).
  - The change is a result of executing an *underlying program*.
  - Several models have been introduced,

Settings

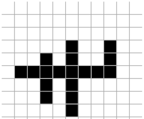
- A discrete system of n entities (nodes) residing on a 2D square grid.
- Each node occupying a distinct cell of the grid .
- The set of n nodes forms initially a connected shape A.


Iinitial connected shape A

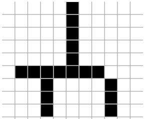


A node is *connected* to a neighbour at any directions.

- B is the given target shape.
- **The goal** is to *transform* A into B via a sequence of line movements.


Iinitial connected shape A

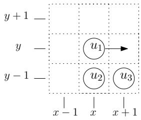
A transformers into *B*

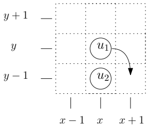

Target connected shape B

Existing similar models

[Dumitrescu et al., IJRR'04 and Michail et al., JCSS'19]:

- Both are special cases of the present model.
- One node u can move a single position in its local neighbourhood,
  - Slide u OR rotate u over neighbouring nodes.





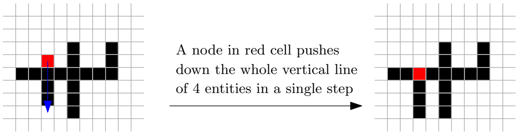
  - Inefficient, $\Theta(n^2)$ *universal transformations*.

Our Model

- Entities are now equipped with a linear-strength pushing mechanism.
- A node can push a whole line of nodes, from 1 to n, in a single time-step.

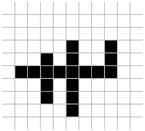
Definition

A line $L = (x_1, y), (x_2, y), \ldots, (x_k, y)$ of length k, where $1 \leq k \leq n$, can push all k nodes rightwards in a single step to positions $(x_2, y), (x_3, y), \ldots, (x_{k+1}, y)$ iff there exists an empty cell to the right of L at (x_{k+1}, y). The "down", "left", and "up" movements are defined symmetrically.

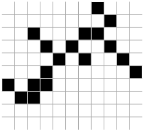


Worst-case shape

- We like to transform connected shapes of long lines ...

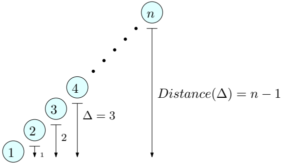


Shape of long lines



Shape of short lines

- The diagonal shape D is a potential worst-case to be transformed into L.
- Similar to the staircase worst-case shape of [Michail *et al.*, JCSS'19].



- $\sum_1^{n-1} \Delta = 1 + 2 + \ldots + (n-1) = \Theta(n^2)$

The main goal

- Investigate whether the new line pushing primitive can be exploited for efficient transformations and achieve a substantial gain in performance.

Problem Definitions

DiagonalToLine. Transform an initial connected diagonal line S_D into a spanning line S_L, without necessarily preserving the connectivity during the transformation.

DiagonalToLineConnected. Restricted version of DiagonalToLine in which connectivity must be preserved during the transformation.

UniversalTransformation. Give a general transformation for all pairs of shapes (S_I, S_F) of the same order, where S_I is the initial shape and S_F the target shape, without necessarily preserving connectivity.

Our contribution

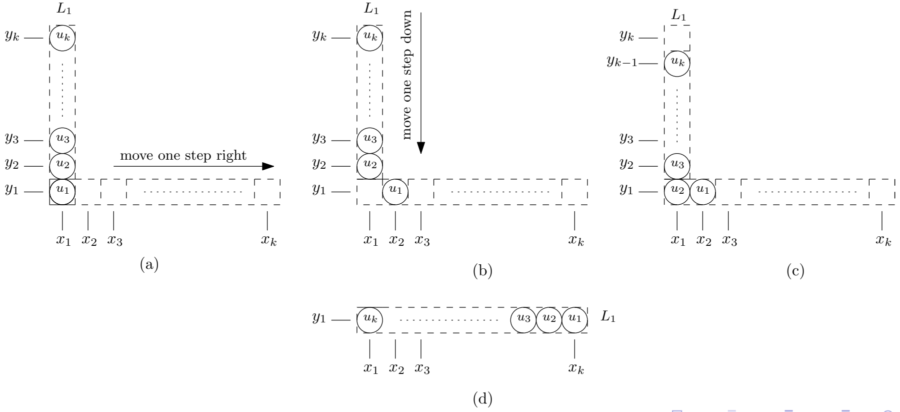
- This table summarises the running times of all transformations:

Transformation	Problem	Running Time	Lower Bound
DL-Partitioning	Diagonal	$O(n\sqrt{n})$	$\Omega(n)$
DL-Doubling	Diagonal	$O(n \log n)$	$\Omega(n)$
DL-Recursion	Diagonal	$O(n \log n)$	$\Omega(n)$
DLC-Folding	D-Connected	$O(n\sqrt{n})$	$\Omega(n)$
DLC-Extending	D-Connected	$O(n\sqrt{n})$	$\Omega(n)$
U-Box-Partitioning	Universal	$O(n\sqrt{n})$	$\Omega(n)$
U-Box-Doubling	Universal	$O(n \log n)$	$\Omega(n)$

Line Pushing Movement

Lemma

The minimum number of line moves by which a line of length k, $1 \leq k \leq n$, can completely change its orientation, is $2k - 2$.



Transformability & Reversibility of Line Movements

- **Universality:** Any pair of connected shapes (A, B) of order n are transformable to each other via a spanning line L.

- Our model simulates the combined rotation and sliding mechanisms,

- Restrict movements to lines of length 1 (i.e., individual nodes).
- Capable of universal transformations with at most twice the worst-case, $O(n^2)$.

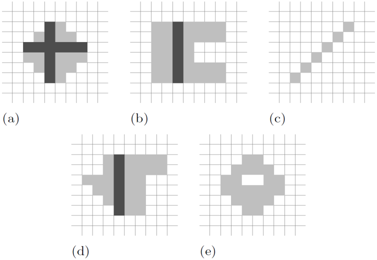
Lemma (Reversibility)

Let (S_I, S_F) be a pair of connected shapes of the same number of nodes n. If $S_I \rightarrow S_F$ ("$\rightarrow$" denoting "can be transformed to via a sequence of line movements") then $S_F \rightarrow S_I$.

Nice shapes

Definition (Nice Shape)

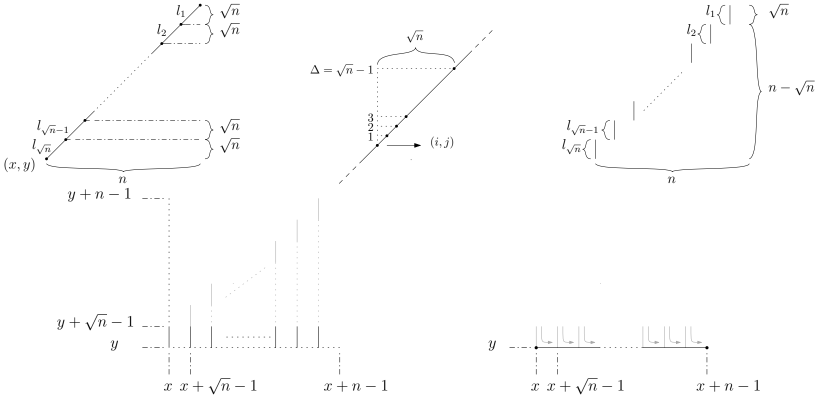
A connected shape $S \in NICE$ if there exists a central line $L_C \subseteq S$, such that every node $u \in S \setminus L_C$ is connected to L_C via a line perpendicular to L_C.



Proposition

- Let S_{Nice} be a nice shape and S_L a straight line, both of the same order n. Then $S_{Nice} \to S_L$ (and $S_L \to S_{Nice}$) in $O(n)$ steps.

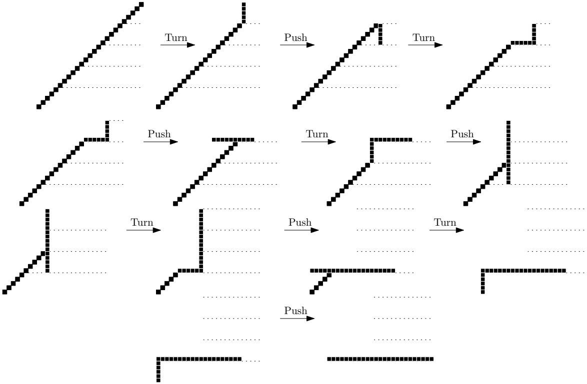
DL-Partitioning



Theorem

Given an initial diagonal of n nodes, DL-Partitioning solves the DIAGONALTOLINE *problem in* $O(n\sqrt{n})$ *steps.*

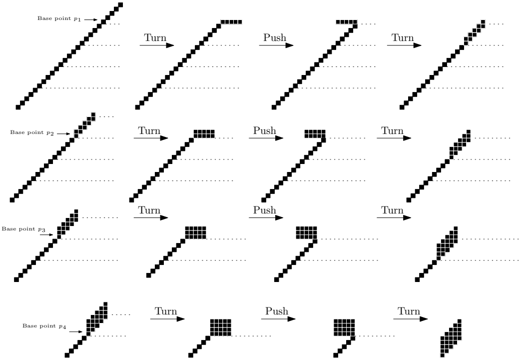
DLC-Extending



Theorem

Given an initial connected diagonal of n nodes, DLC-Extending solves the DiagonalToLineConnected *problem in* $O(n\sqrt{n})$ *steps.*

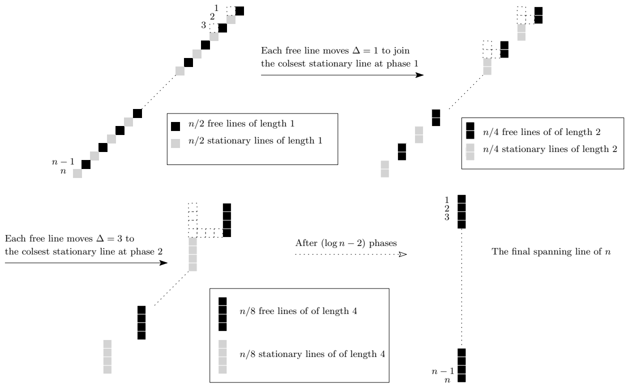
DLC-Folding



Theorem

Given an initial connected diagonal of n nodes, DLC-Folding solves the DIAGONALTOLINECONNECTED *problem in* $O(n\sqrt{n})$ *steps.*

DL-Doubling



Theorem

DL-Doubling converts any diagonal S_D of order n into a line S_L in $O(n \log n)$ steps.

DL-Recursion

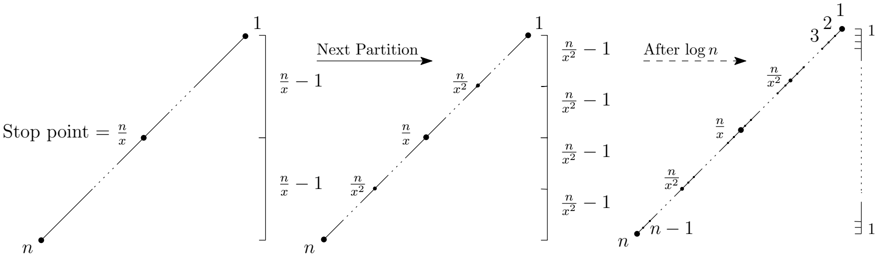


Figure: Shows all steps of subdividing the diagonal S_D recursively by a factor of $\frac{1}{x}$, where $x = 2$.

Theorem

DL-Recursion transforms any diagonal S_D of order n into a line S_L of the same order in $O(n \log n)$ steps.

DL-Recursion

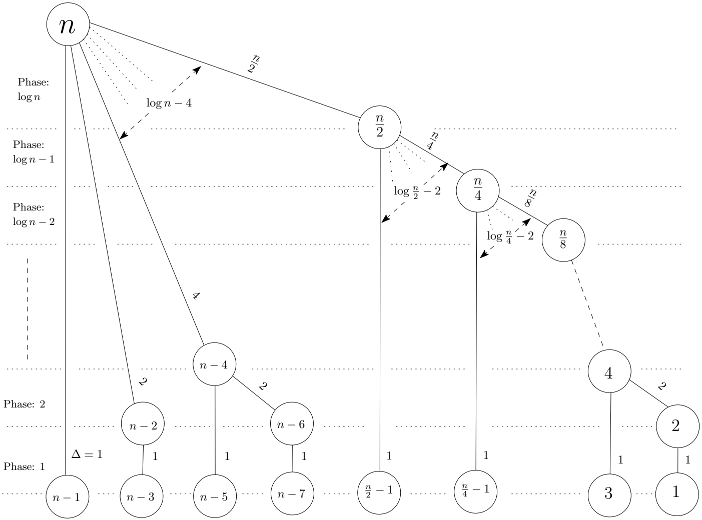


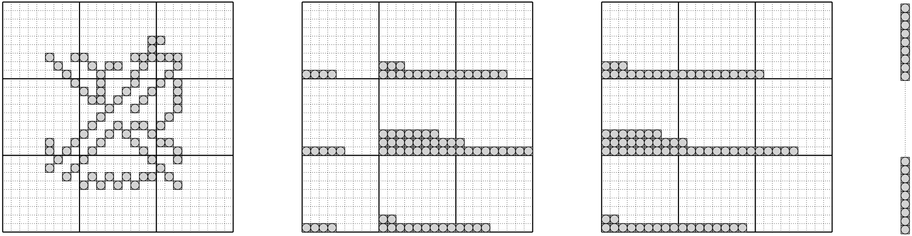
Figure: Tree representation of a recursive partitioning of S_D. Edges are weighted by the minimum distance (Δ) between nodes.

U-Box-Partitioning

Universal Transformations

Corollary

Given a uniform partitioning of $n \times n$ square box containing a connected shape S_I of order n into $d \times d$ sub-boxes, it holds that S_I can occupy at most $O(\frac{n}{d})$ sub-boxes.



Theorem

For any pair of connected shapes S_I and S_F of the same order n, U-Box-Partitioning can be used to transform S_I into S_F (and S_F into S_I) in $O(n\sqrt{n})$ steps.

U-Box-Doubling

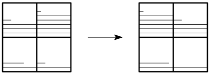
Universal Transformations

- Enclose the initial shape S_I into a square bounding box of size $n \times n$.

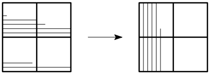
- Proceed in $\log n$ phases.

- In phase i, partition the $n \times n$ box into $2^i \times 2^i$ sub-boxes.

- For every $2^{i-1} \times 2^{i-1}$ sub-box, move each line from the previous phase into the left boundary of the sub-box.

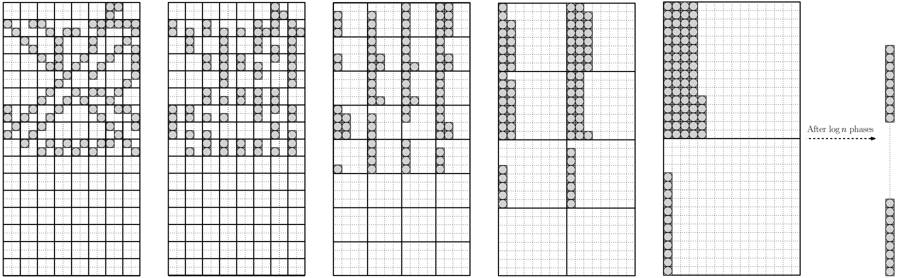


- By a linear procedure, filling in the columns of the $2^i \times 2^i$ sub-box from the leftmost column.



U-Box-Doubling

Universal Transformations



Theorem

For any pair of connected shapes S_I and S_F of the same order n, transformation U-Box-Doubling can be used to transform S_I into S_F (and S_F into S_I) in $O(n \log n)$ steps.

Future Research

- Is there an $\Omega(n \log n)$-time lower bound matching our best transformation?
- Investigate distributed versions of the transformations.
- The case in which connectivity has to be preserved during the transformations.
- Parallelism, if more than one line can move in a single time-step.
- Alternative types of grids, such as triangular and hexagonal.