Amoeba-inspired Self-organizing Particle Systems

BDA 2013

Shlomi Dolev, **Robert Gmyr**, Andréa W. Richa, and Christian Scheideler

October 14, 2013

Motivation

*"Over the next few decades, two emerging technologies—microfabrication and cellular engineering—will make it possible to assemble systems incorporating myriads of information-processing units at almost no cost [...] but we have few ideas for programming them effectively. The opportunity to exploit these new technologies poses a broad conceptual challenge—the challenge of **amorphous computing**."* [AAC$^+$00]

Our goal: rigorous algorithmic research on self-organizing particle systems

→ First step: appropriate model

Properties

- particles are **physical entities**
- particles have to **stay connected**
- all particles are **programmed identically**
- particles have **local knowledge**
  - no position
  - no orientation
  - only perceive immediate neighborhood
- particles have **modest computational power**
  - finite automata
- **unlimited number** of particles

A New Model

Previous models do not fit

- DNA computing, population protocols, ...
  → **no active movement**
- swarm robotics
  → **no collisions, no connectivity**
- modular robotics, metamorphic robotics
  → **global information, powerful particles (Turing machines)**

Further considerations

- **implementation** should be **feasible**
- allow particles to make **local decisions** and act in a **distributed** fashion
- for now: **2D** and **synchronous rounds**

A New Model

Previous models do not fit

- DNA computing, population protocols, ...
  $\rightarrow$ **no active movement**
- swarm robotics
  $\rightarrow$ **no collisions, no connectivity**
- modular robotics, metamorphic robotics
  $\rightarrow$ **global information, powerful particles (Turing machines)**

Further considerations

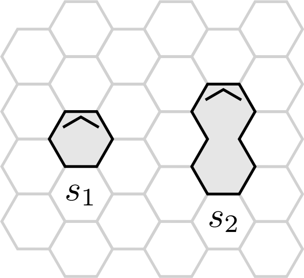
- **implementation** should be **feasible**
- allow particles to make **local decisions** and act in a **distributed** fashion
- for now: **2D** and **synchronous rounds**

Model

Particles

- are placed on a **hexagonal grid**
- assume one of **two shapes**
- assume one of **six orientations**
- are in one of **finitely many states**
- have to stay **connected**

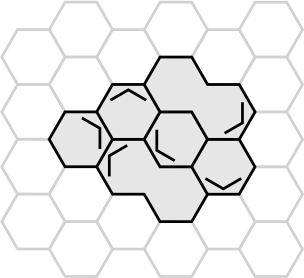
Model



Particles

- are placed on a **hexagonal grid**
- assume one of **two shapes**
- assume one of **six orientations**
- are in one of **finitely many states**
- have to stay **connected**

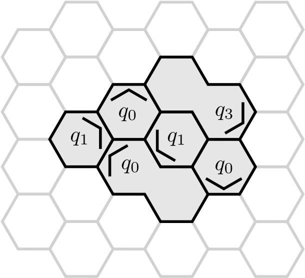
Model



Particles

- are placed on a **hexagonal grid**
- assume one of **two shapes**
- assume one of **six orientations**
- are in one of **finitely many states**
- have to stay **connected**

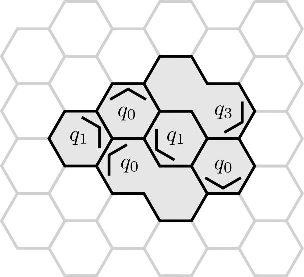
Model



Particles

- are placed on a **hexagonal grid**
- assume one of **two shapes**
- assume one of **six orientations**
- are in one of **finitely many states**
- have to stay **connected**

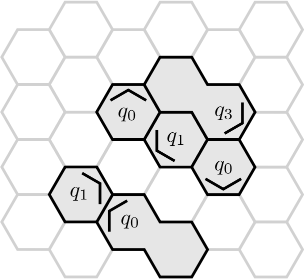
Model



Particles

- are placed on a **hexagonal grid**
- assume one of **two shapes**
- assume one of **six orientations**
- are in one of **finitely many states**
- have to stay **connected**

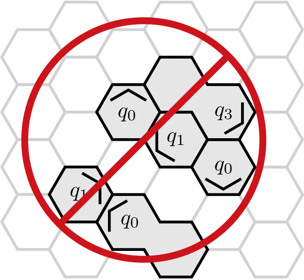
Model



Particles

- are placed on a **hexagonal grid**
- assume one of **two shapes**
- assume one of **six orientations**
- are in one of **finitely many states**
- have to stay **connected**

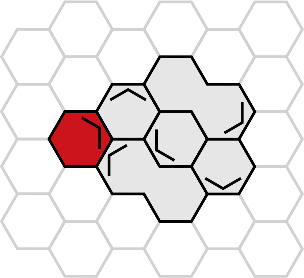
Model



Particles

- are placed on a **hexagonal grid**
- assume one of **two shapes**
- assume one of **six orientations**
- are in one of **finitely many states**
- have to stay **connected**

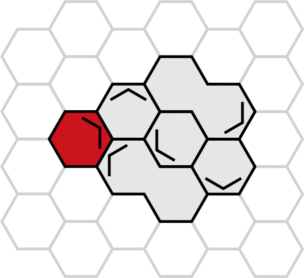
Model



In every round, a particle can **change its state** and execute one of **six actions**

1. **null**
2. turn
3. expand
4. contract
5. duplicate
6. kill

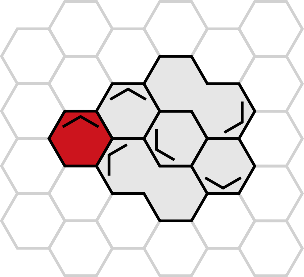
Model



In every round, a particle can **change its state** and execute one of **six actions**

1. null
2. **turn**
3. expand
4. contract
5. duplicate
6. kill

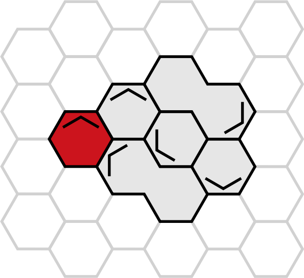
Model



In every round, a particle can **change its state** and execute one of **six actions**

1. null
2. **turn**
3. expand
4. contract
5. duplicate
6. kill

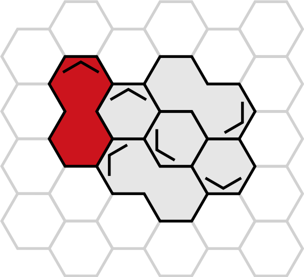
Model



In every round, a particle can **change its state** and execute one of **six actions**

1. null
2. turn
3. **expand**
4. contract
5. duplicate
6. kill

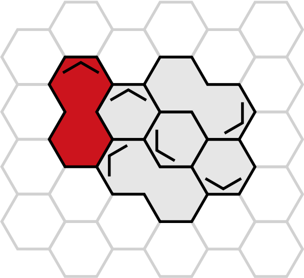
Model



In every round, a particle can **change its state** and execute one of **six actions**

1. null
2. turn
3. **expand**
4. contract
5. duplicate
6. kill

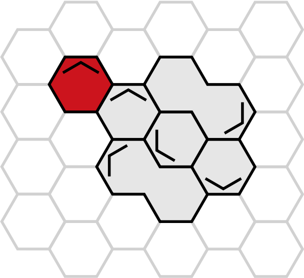
Model



In every round, a particle can **change its state** and execute one of **six actions**

1. null
2. turn
3. expand
4. **contract**
5. duplicate
6. kill

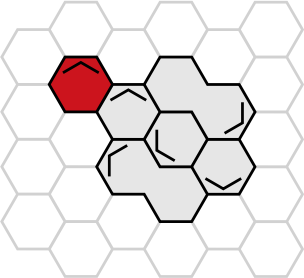
Model



In every round, a particle can **change its state** and execute one of **six actions**

1. null
2. turn
3. expand
4. **contract**
5. duplicate
6. kill

Model

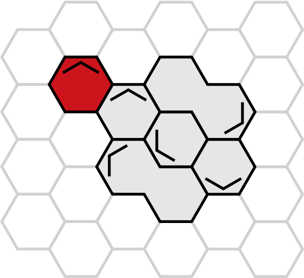


In every round, a particle can **change its state** and execute one of **six actions**

1. null
2. turn
3. expand
4. contract
5. duplicate
6. kill

→ **amoeboid movement or cell crawling**

Model

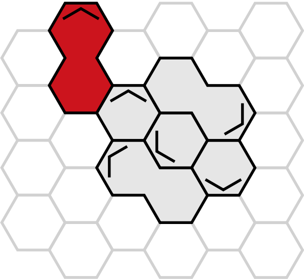


In every round, a particle can **change its state** and execute one of **six actions**

1. null
2. turn
3. expand
4. contract
5. **duplicate**
6. kill

→ **amoeboid movement or cell crawling**

Model

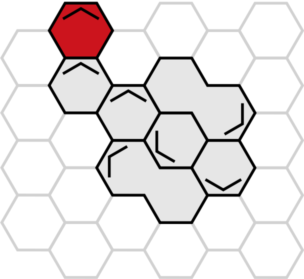


In every round, a particle can **change its state** and execute one of **six actions**

1. null
2. turn
3. expand
4. contract
5. **duplicate**
6. kill

→ **amoeboid movement or cell crawling**

Model

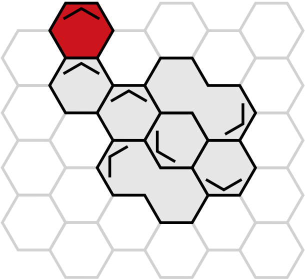


In every round, a particle can **change its state** and execute one of **six actions**

1. null
2. turn
3. expand
4. contract
5. **duplicate**
6. kill

→ **amoeboid movement or cell crawling**

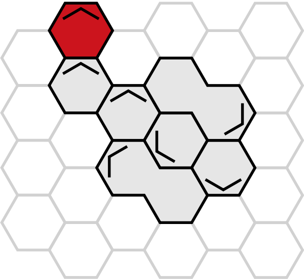
Model



In every round, a particle can **change its state** and execute one of **six actions**

1. null
2. turn
3. expand → **amoeboid movement or**
4. contract **cell crawling**
5. duplicate → **cell division**
6. kill

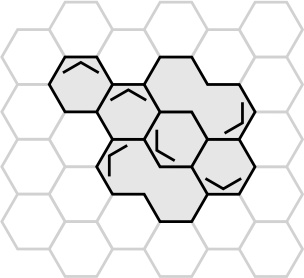
Model



In every round, a particle can **change its state** and execute one of **six actions**

1. null
2. turn
3. expand → **amoeboid movement or**
4. contract **cell crawling**
5. duplicate → **cell division**
6. **kill**

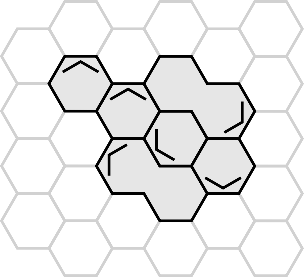
Model



In every round, a particle can **change its state** and execute one of **six actions**

1. null
2. turn
3. expand → **amoeboid movement or**
4. contract **cell crawling**
5. duplicate → **cell division**
6. **kill**

Model



In every round, a particle can **change its state** and execute one of **six actions**

1. null
2. turn
3. expand → **amoeboid movement or**
4. contract **cell crawling**
5. duplicate → **cell division**
6. kill → **cell death**

Amoeboid Movement [AE07]

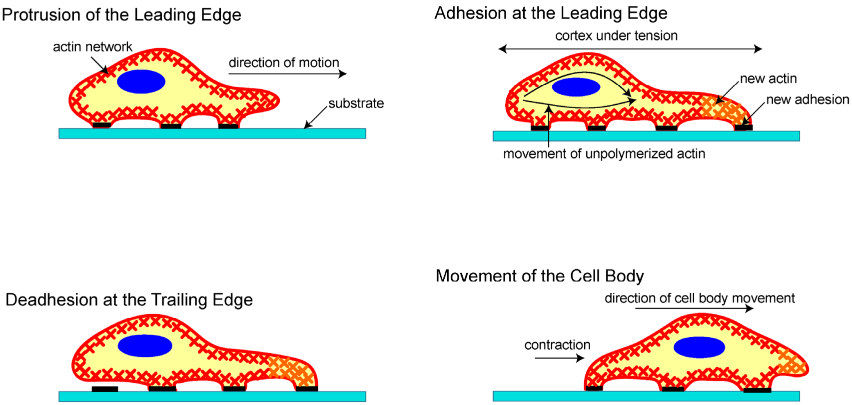
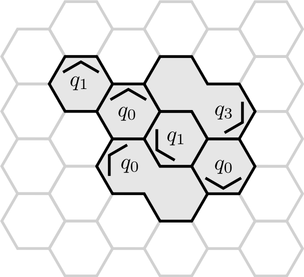


Figure adopted from [AE07].

Model

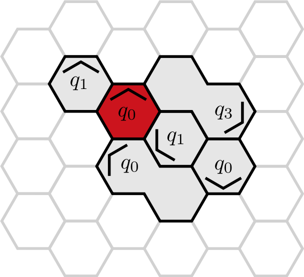


A particle uses

- its own **state** and **shape** and
- the **state**, **shape**, **relative position**, and **relative orientation** of immediate neighbors

to **probabilistically** determine its next state and action.

Model

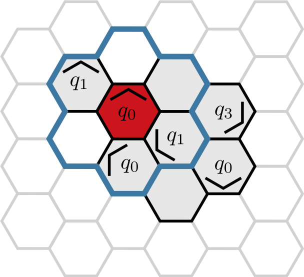


A particle uses

- its own **state** and **shape** and
- the **state**, **shape**, **relative position**, and **relative orientation** of immediate neighbors

to **probabilistically** determine its next state and action.

Model

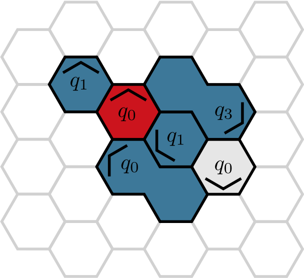


A particle uses

- its own **state** and **shape** and
- the **state**, **shape**, **relative position**, and **relative orientation** of immediate neighbors

to **probabilistically** determine its next state and action.

Model

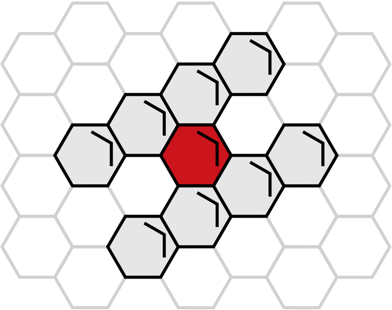


A particle uses

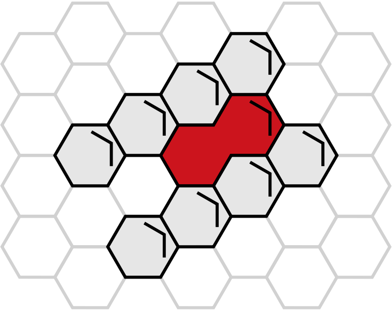
- its own **state** and **shape** and
- the **state**, **shape**, **relative position**, and **relative orientation** of immediate neighbors

to **probabilistically** determine its next state and action.

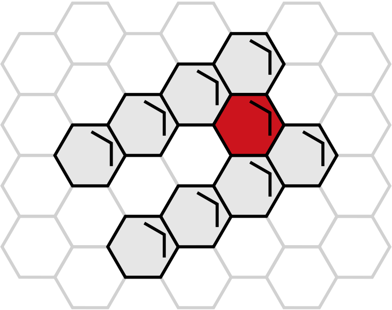
Why Amoeboid Movement?



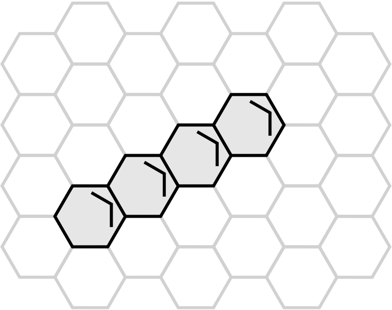
Why Amoeboid Movement?



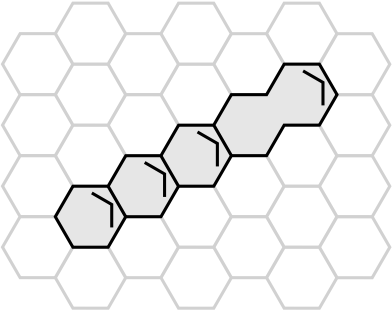
Why Amoeboid Movement?



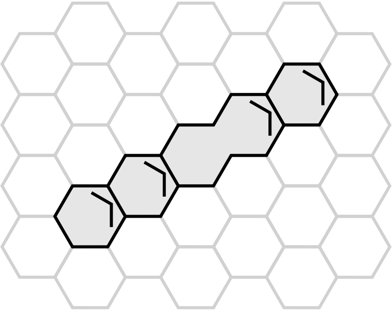
Why Amoeboid Movement?



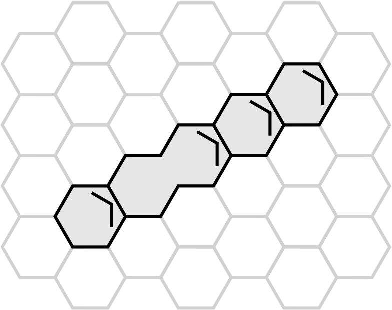
Why Amoeboid Movement?



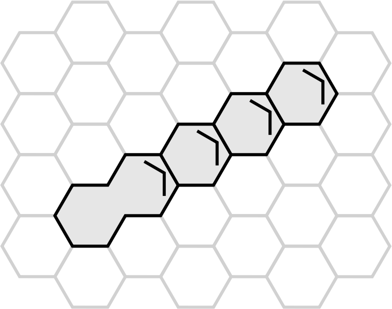
Why Amoeboid Movement?



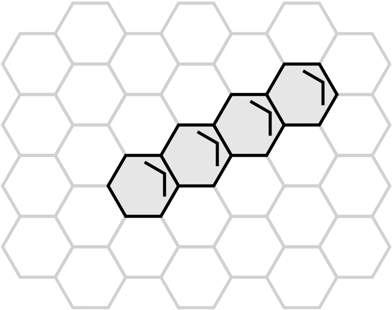
Why Amoeboid Movement?



Why Amoeboid Movement?

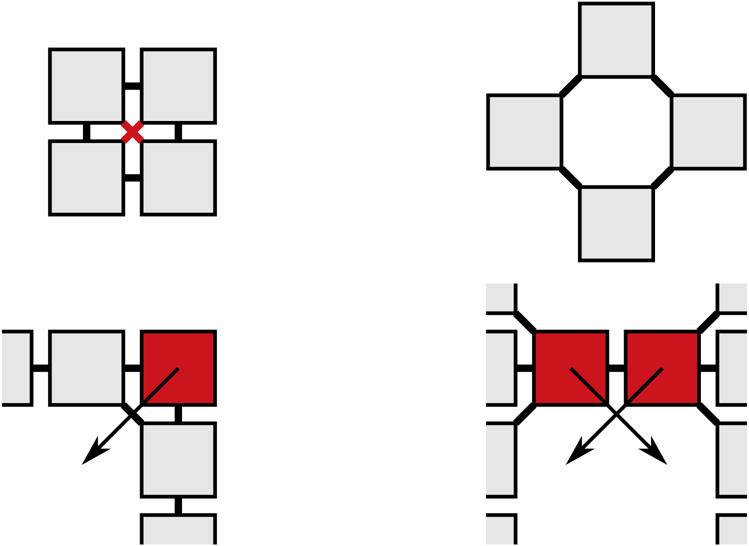


Why Amoeboid Movement?



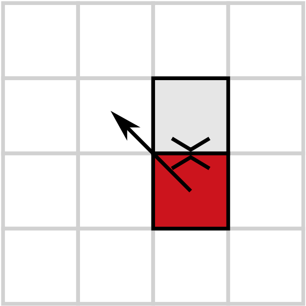
Why not a square model?

Moore-neighborhood for connectivity



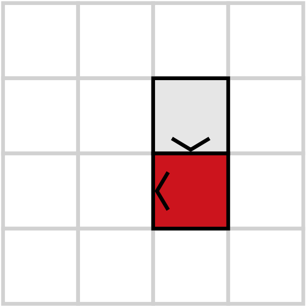
Why not a square model?

Von-Neumann-neighborhood for connectivity



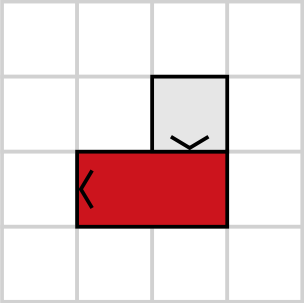
Why not a square model?

Von-Neumann-neighborhood for connectivity



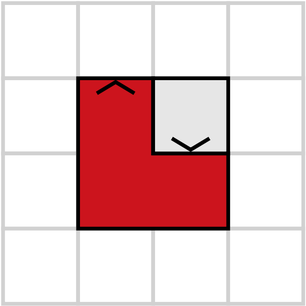
Why not a square model?

Von-Neumann-neighborhood for connectivity



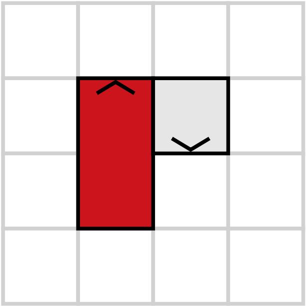
Why not a square model?

Von-Neumann-neighborhood for connectivity



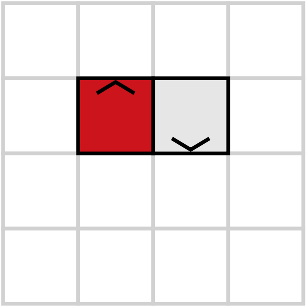
Why not a square model?

Von-Neumann-neighborhood for connectivity



Why not a square model?

Von-Neumann-neighborhood for connectivity



Variants of the Model

- asynchronous
- 3D
- no duplicate and kill action
- action to kill other particles
- failing or byzantine particles
- self-stabilization
- less information from neighbors
- morphogen gradients

Research Problems

”Simple” problems

- covering problems
- shape formation problems
- bridging problems

More involved problems

- macrophage problem

Bibliography

[AAC+00] H. Abelson, D. Allen, D. Coore, C. Hanson, G. Homsy, T. F. Knight, R. Nagpal, E. Rauch, G. J. Sussman, and R. Weiss.
Amorphous computing.
Communications of the ACM, 43(5):74–82, 2000.

[AE07] R. Ananthakrishnan and A. Ehrlicher.
The forces behind cell movement.
International Journal of Biological Sciences, 3(5):303–317, 2007.