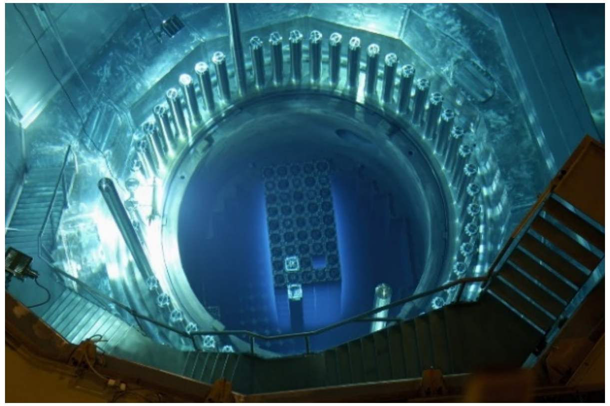
A Stochastic Approach to Shortcut Bridging in Programmable Matter

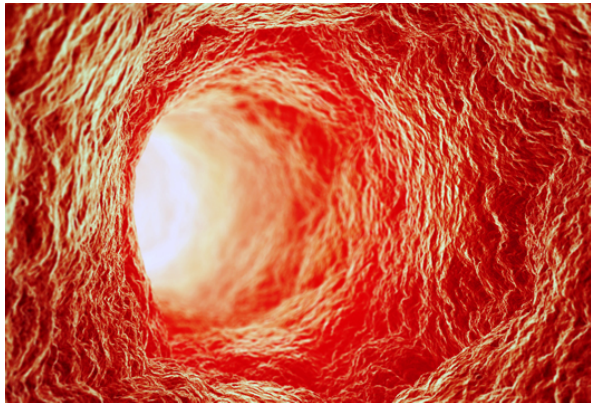
JOSHUA J. DAYMUDE AND ANDRÉA W. RICHA – ARIZONA STATE UNIVERSITY

SARAH CANNON AND DANA RANDALL – GEORGIA INSTITUTE OF TECHNOLOGY

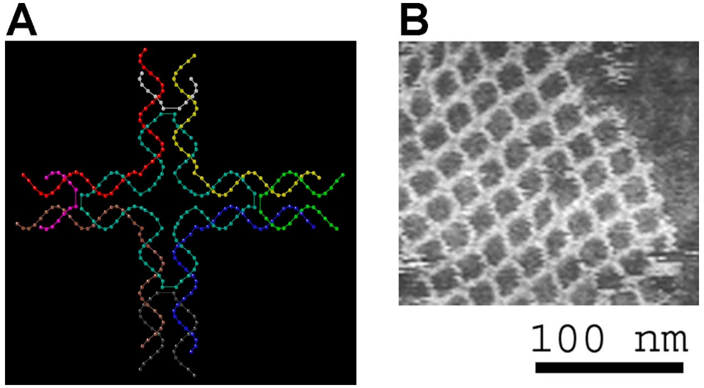
MARTA ANDRÉS ARROYO – UNIVERSITY OF GRANADA

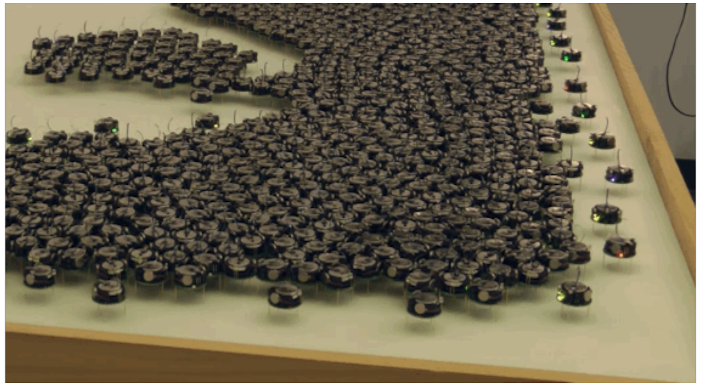
Inspirations & Applications

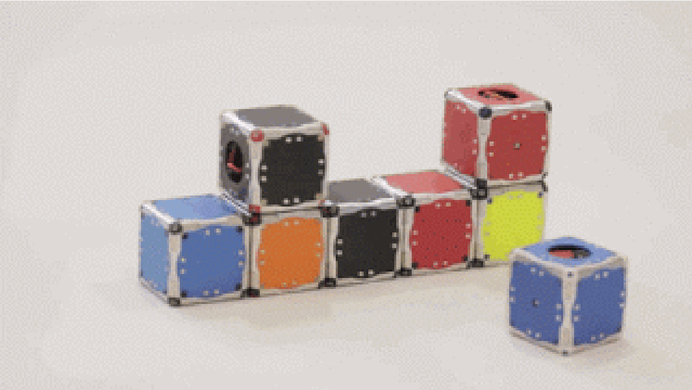


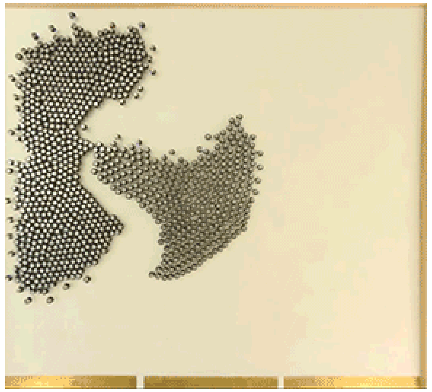


Current Programmable Matter







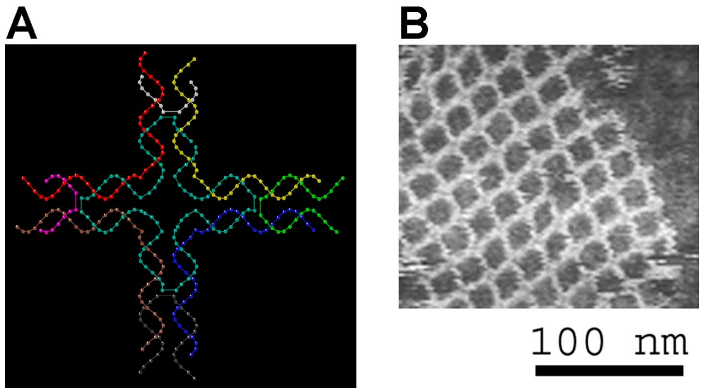


RGR 2013: "M-blocks: Momentum driven, magnetic modular robots" RCN 2014: "Programmable self-assembly in a thousand-robot swarm"

Current Programmable Matter

Programmable matter systems can be **passive** or **active**:

- **Passive**: no movement control, depends on environment.
- **Active**: can control actions and movements to solve problems.



Self-Organizing Particle System:

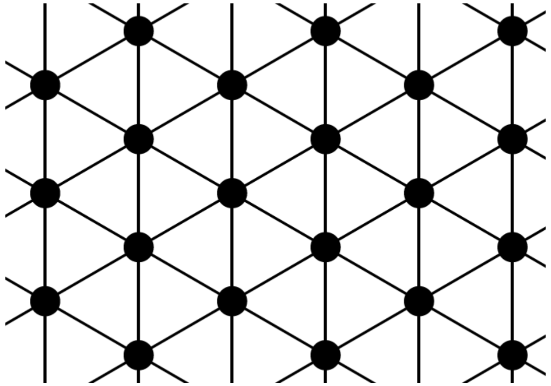
- Abstraction of **active** programmable matter systems.
- Simple computational units -> coordinated behavior.
- Constrain individual's abilities to ask what's possible.

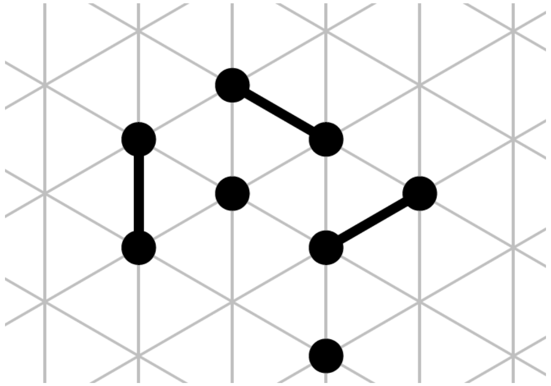
RCN 2014: "Programmable self-assembly in a thousand-robot swarm"

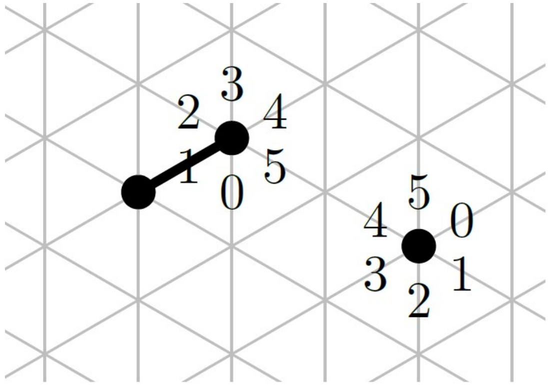
The Amoebot Model

Particles move by *expanding* and *contracting,* and are:

- Anonymous (no unique identifiers)
- Without global orientation or compass (no shared sense of "north")
- Limited in memory (constant size)
- Activated asynchronously







Previous Work in the Amoebot Model

Deterministic* algorithms exist for:

- Shape formation (triangles, hexagons, etc.).
- Evenly coating objects (infinite, bounded, and closed).
- Leader election (with high probability).

See
sops.engineering.asu.edu
for simulations!

Stochastic algorithms exist for:

- Compression, or gathering a particle system together as tightly as possible.
- Shortcut bridging (this talk).

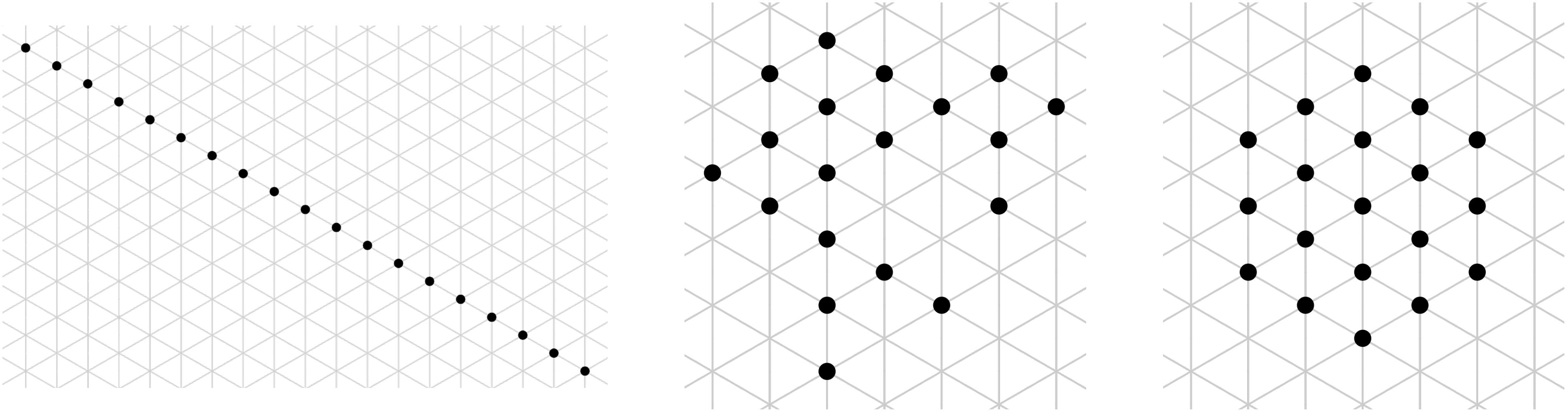
* some randomization is used

Why Stochastic?

An example from **compression**: form a configuration whose ***perimeter*** is as small as possible (same thing as gathering as tightly as possible).

O(sqrt(|*P*|)) 2 1

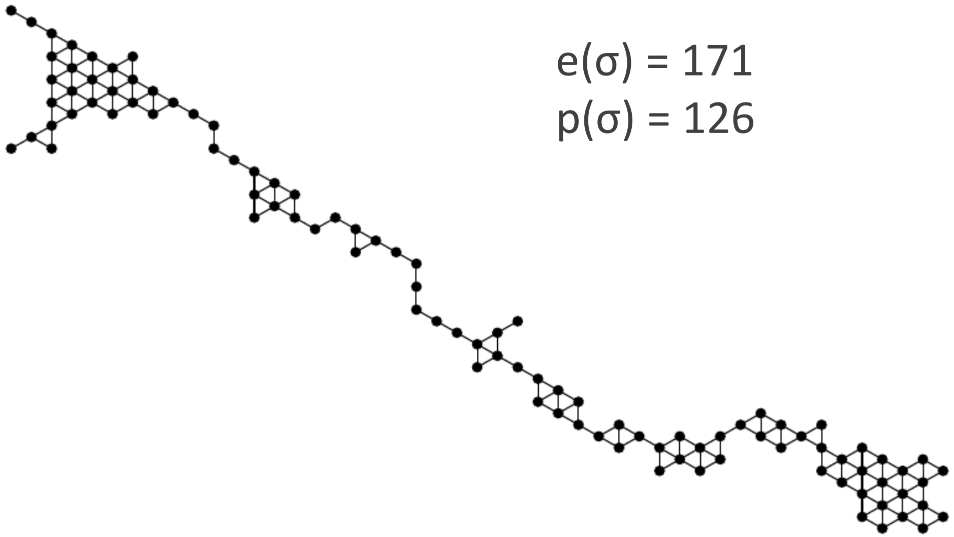
α



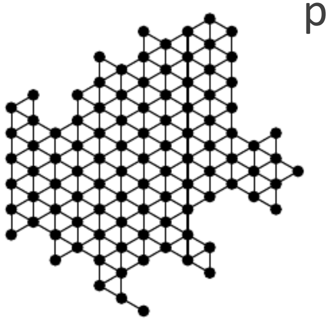
Why Stochastic?

Perimeter is a *global* property, but our particles are limited to *local* communication.

- **Lemma:** Maximizing the number of internal edges is equivalent to minimizing perimeter.
- First attempt: particles move to positions where they have more neighbors.



$e(\sigma) = 171$
$p(\sigma) = 126$

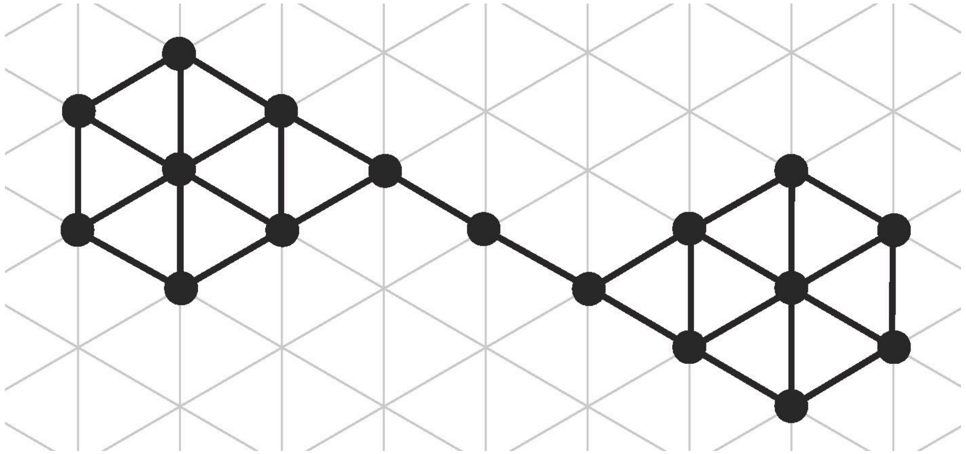


$e(\sigma) = 246$
$p(\sigma) = 51$

Why Stochastic?

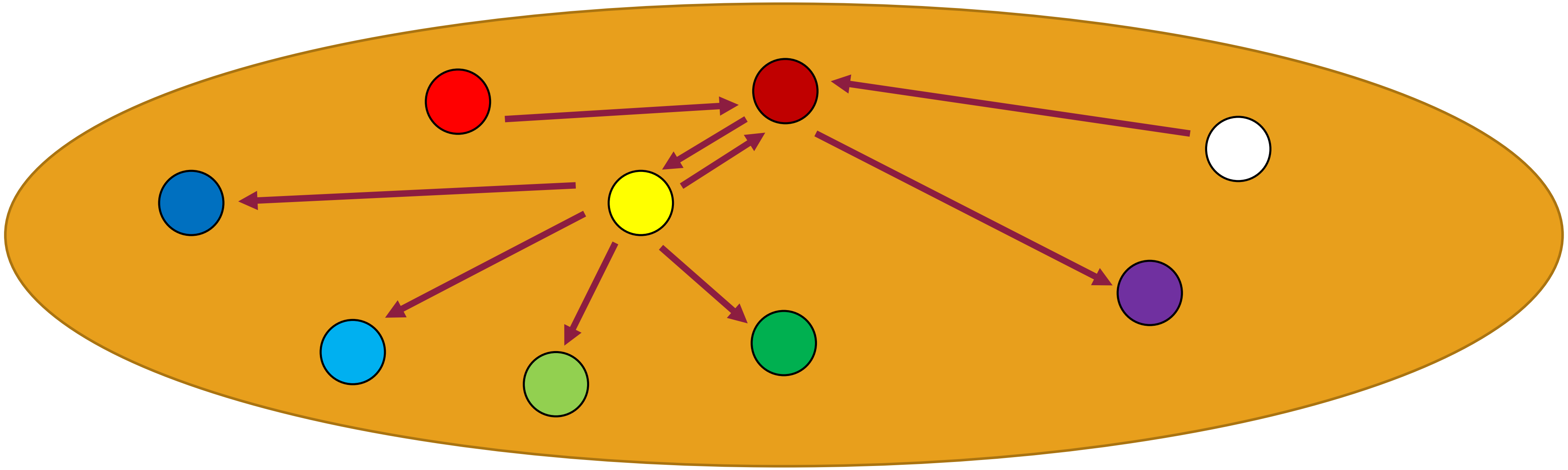
Perimeter is a *global* property, but our particles are limited to *local* communication.

- **Lemma:** Maximizing the number of internal edges is equivalent to minimizing perimeter.
- First attempt: particles move to positions where they have more neighbors.
- …However, need something more robust to local minima.



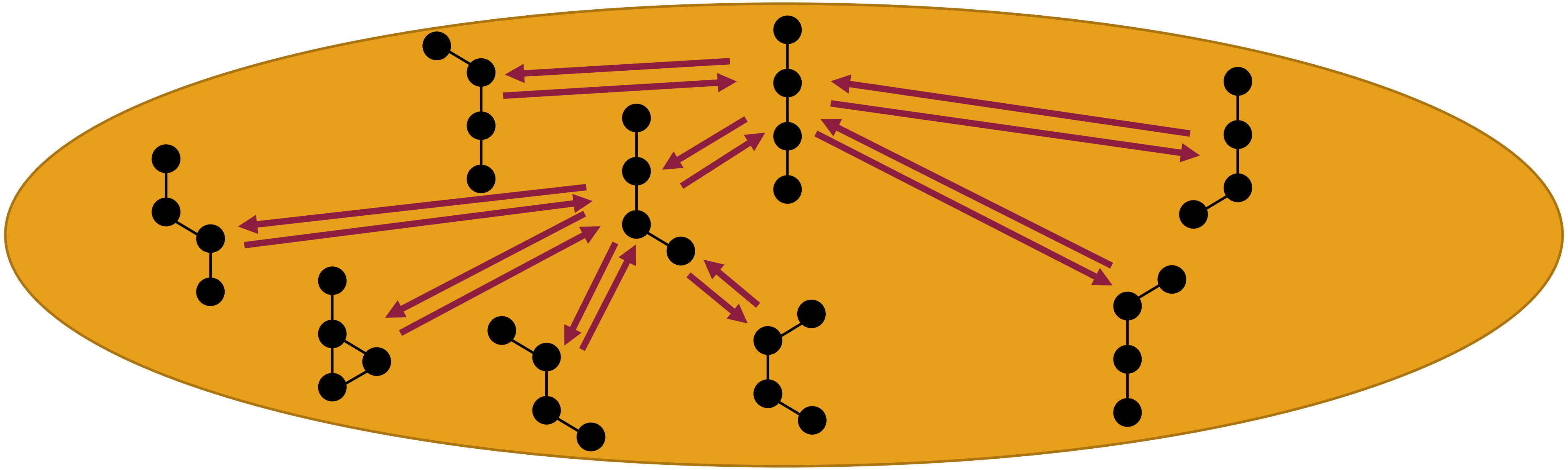
Markov Chains

- A *Markov chain* is a memoryless random process that undergoes transitions between states in some state space.



Markov Chains

- A *Markov chain* is a memoryless random process that undergoes transitions between states in some state space.
- In our context, states are *particle system configurations*, and transitions between them are individual particle movements.



Markov Chains for Particle Systems

Turn a Markov chain (global, step-by-step) into a **local**, distributed, asynchronous algorithm:

- Carefully define the Markov chain to only use **local** moves.

Markov chain algorithm:

Starting from any configuration, repeat:

1. **Choose a particle** at random.
2. Expand into a (random) unoccupied adjacent position.
3. Perform some arbitrary, bounded computation involving its neighborhood.
4. Contract to either the new position or the original position.

Distributed algorithm:

Each particle concurrently and continuously executes:

2. Expand into a (random) unoccupied adjacent position.
3. Perform some arbitrary, bounded computation involving its neighborhood.
4. Contract to either the new position or the original position.

Markov Chains for Particle Systems

Turn a Markov chain (global, step-by-step) into a **local**, distributed, asynchronous algorithm:

- Carefully define the Markov chain to only use **local** moves.

Markov chain algorithm:

Starting from any configuration, repeat:

1. **Choose a particle** at random.
2. Expand into a (random) unoccupied adjacent position.
3. If certain properties hold, contract to new position with probability Pr[move].
4. Else, contract back to the original position.

Distributed algorithm:

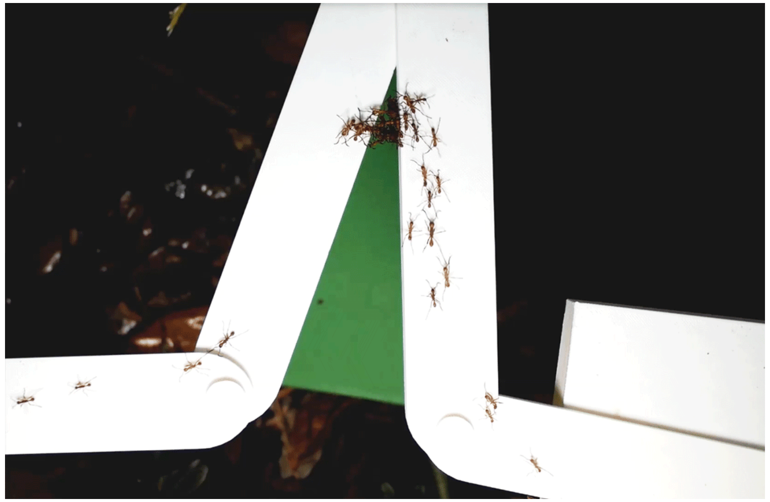
Each particle concurrently and continuously executes:

2. Expand into a (random) unoccupied adjacent position.
3. If certain properties hold, contract to new position with probability Pr[move].
4. Else, contract back to the original position.

Our Results: Shortcut Bridging

Reid et al. looked at army ants (*Eciton*) and how they self-assemble bridges. They found:

- Ants build the bridges to shorten the path distance other ants travel…
- …but to do so they take ants out of the workforce.
- Tradeoff: make the total path shorter, but without sacrificing too many workers.

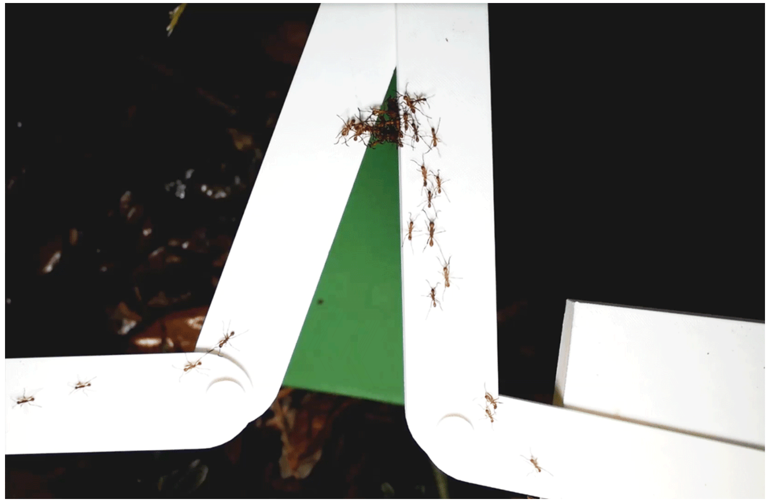


RLPKCG 2015: "Army ants dynamically adjust living bridges…"

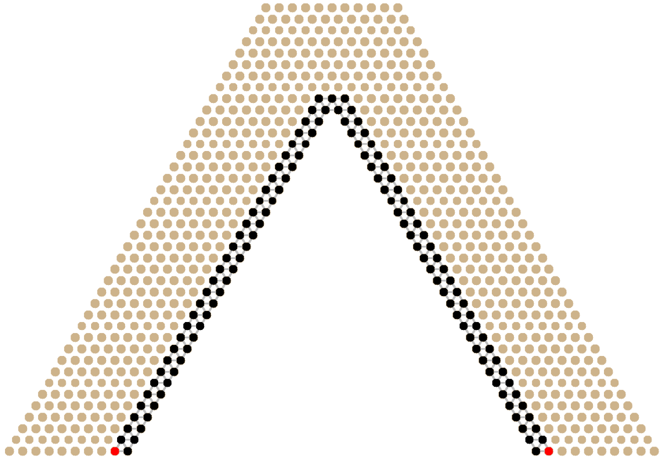
Our Results: Shortcut Bridging

Our Contribution: A stochastic, distributed, local, asynchronous algorithm for **shortcut bridging** in which particles maintain self-assembled bridges over gaps, balancing:

- The benefit of a shorter path.
- The cost in ant workers of a longer bridge.



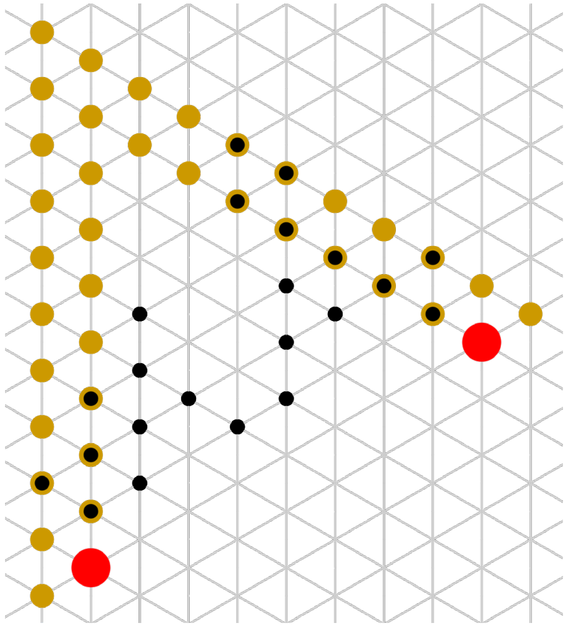
RLPKCG 2015: "Army ants dynamically adjust living bridges…"



The Shortcut Bridging Problem

We first need to add some problem-specific things to the model:

- Land and gap positions.
- Fixed objects (to anchor the particle system to land).

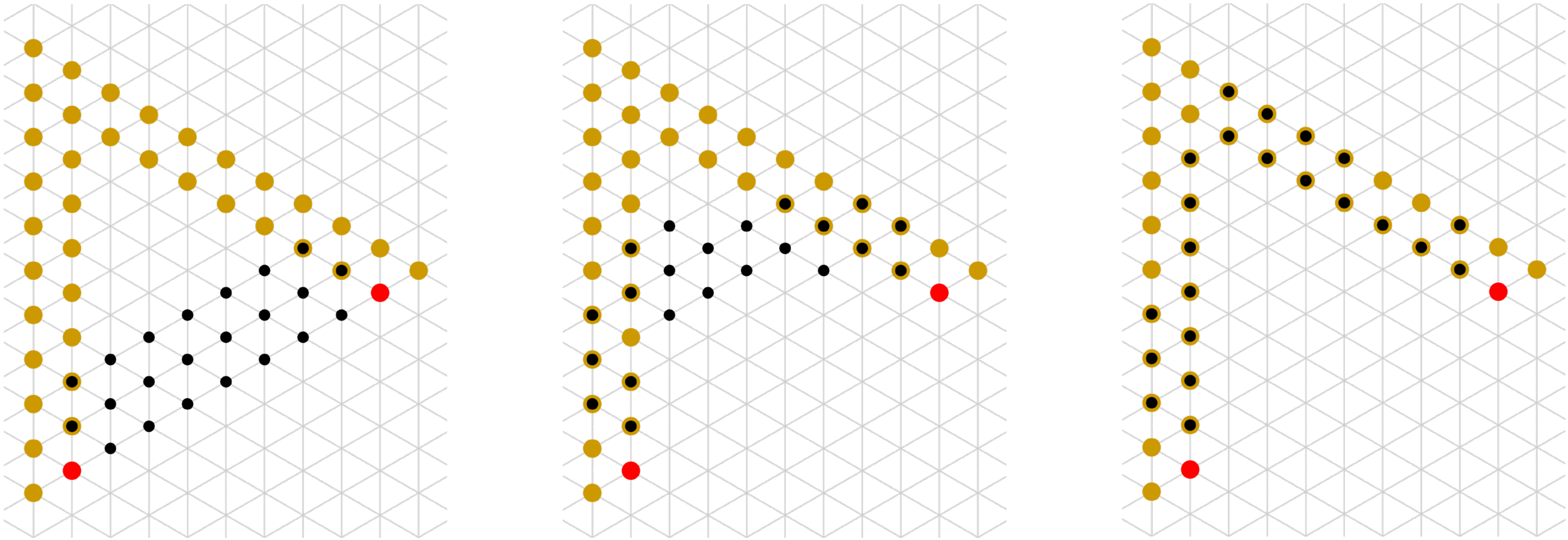


The Shortcut Bridging Problem

Goal: Dynamically adapt bridges to balance the benefit of a shorter path with the loss of ant workers.

- We'll minimize both the total perimeter p(σ) and the gap perimeter g(σ).

Min p(σ)
Max g(σ)



Max p(σ)
Min g(σ)

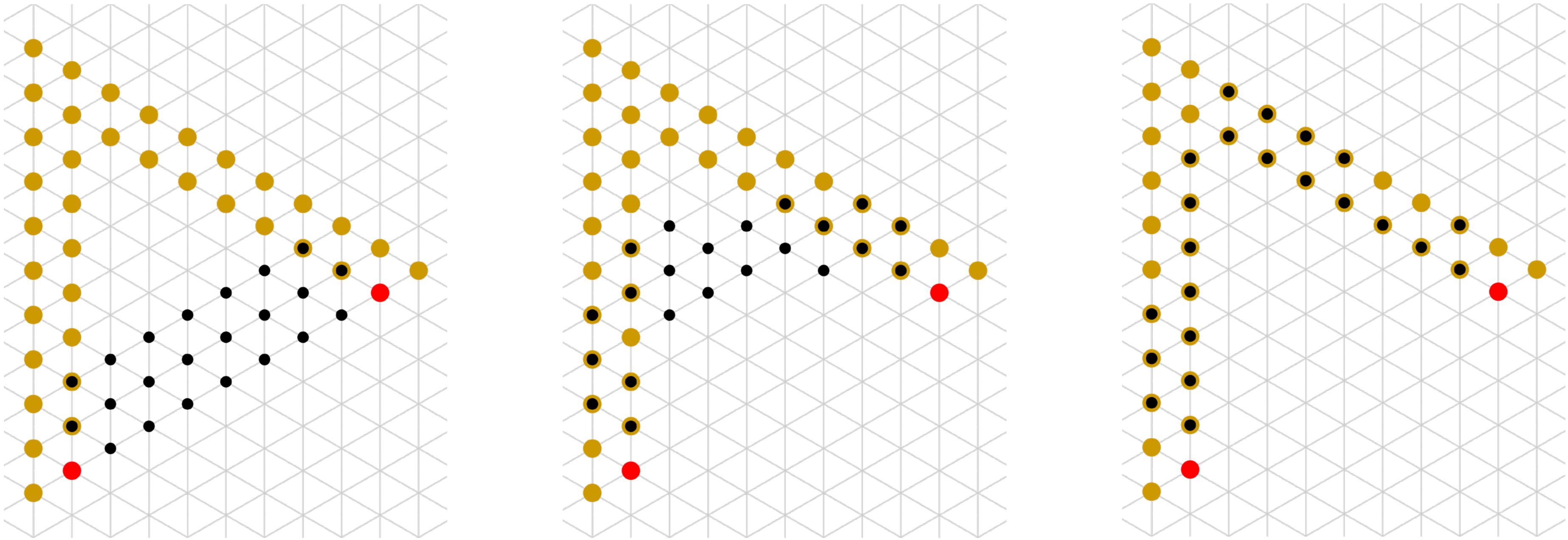
The Shortcut Bridging Problem

Goal: Dynamically adapt bridges to balance the benefit of a shorter path with the loss of ant workers.

- Formally, minimize **weighted perimeter** $p'(\sigma,c) = p(\sigma) + c \cdot g(\sigma)$, where $c > 0$.

Min p(σ)

Max g(σ)



Max p(σ)

Min g(σ)

The Shortcut Bridging Algorithm

Input: an initial (connected, hole-free) configuration σ_0 and bias parameters λ, γ > 1.

Repeat:

1. Choose a particle from the system uniformly at random.
2. Choose an adjacent position uniformly at random. If this position is occupied, go to Step 1.
3. If properties hold for maintaining connectivity and avoiding holes, move to the chosen position with probability $\min\{1, \lambda^{-\Delta p}\, \gamma^{-\Delta g}\}$.

Metropolis filter
(calculated w/ local info)

Proof: Detailed Balance

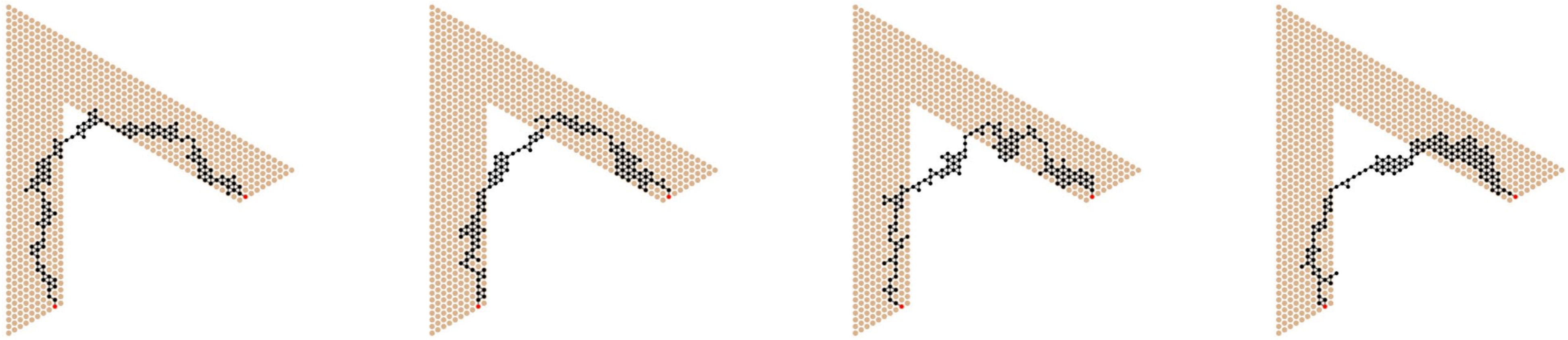
Theorem: We reach a stationary distribution π over configurations σ where $\pi(\sigma) \sim \lambda^{-p(\sigma)}\, \gamma^{-g(\sigma)}$.

Theorem: For any $\alpha > 1$, there are λ and γ (depending on α and c) so that in the stationary distribution $\pi(\sigma) \sim \lambda^{-p(\sigma)}\, \gamma^{-g(\sigma)}$ we have $p'(\sigma,c) \leq \alpha \cdot p'_{min}$ with high probability.

Proof: Peierls argument

Simulation: Shortcut Bridging, $\lambda = 4$, $\gamma = 2$

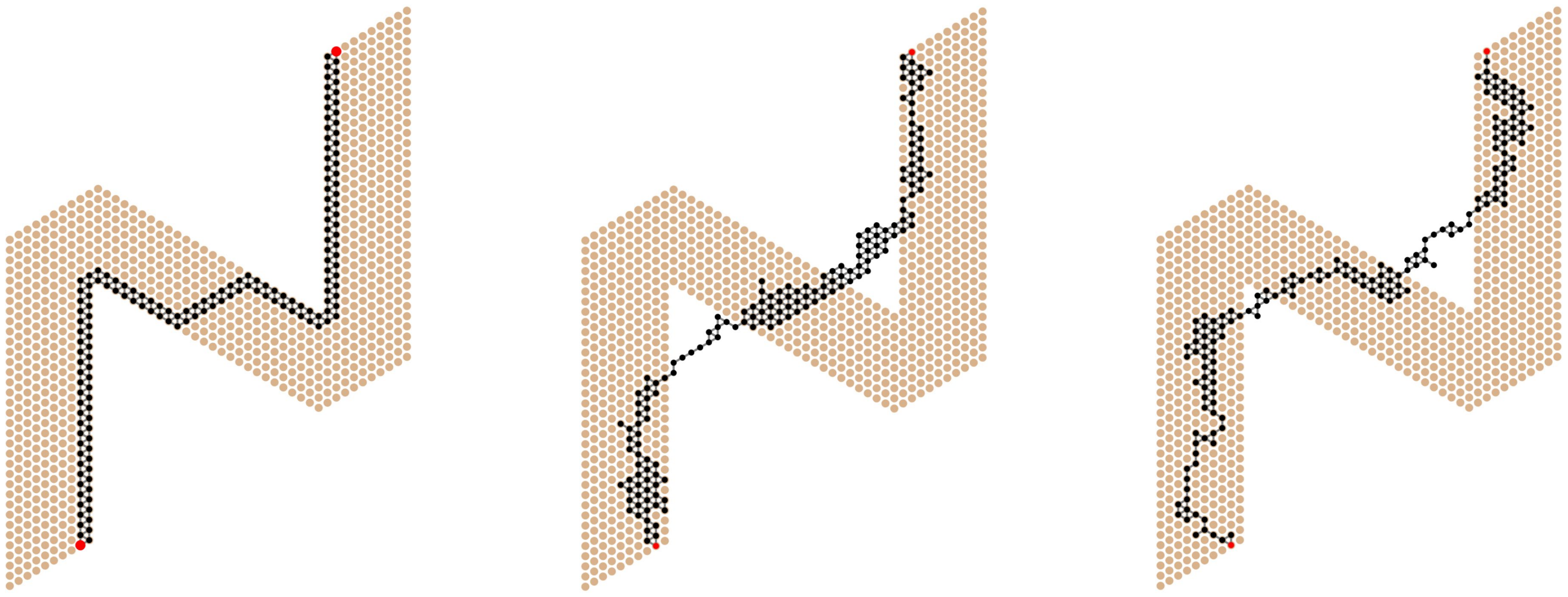
A particle system initially fully on a V-shaped land mass after (a) 2 million, (b) 4 million, (c) 6 million, and (d) 8 million iterations.



(a) (b) (c) (d)

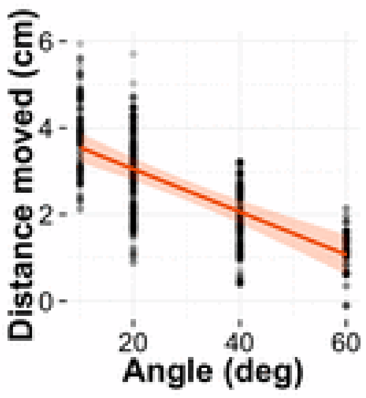
Simulation: Shortcut Bridging, $\lambda = 4$, $\gamma = 2$

A particle system initially fully on an N-shaped land mass after 10 million and 20 million iterations.



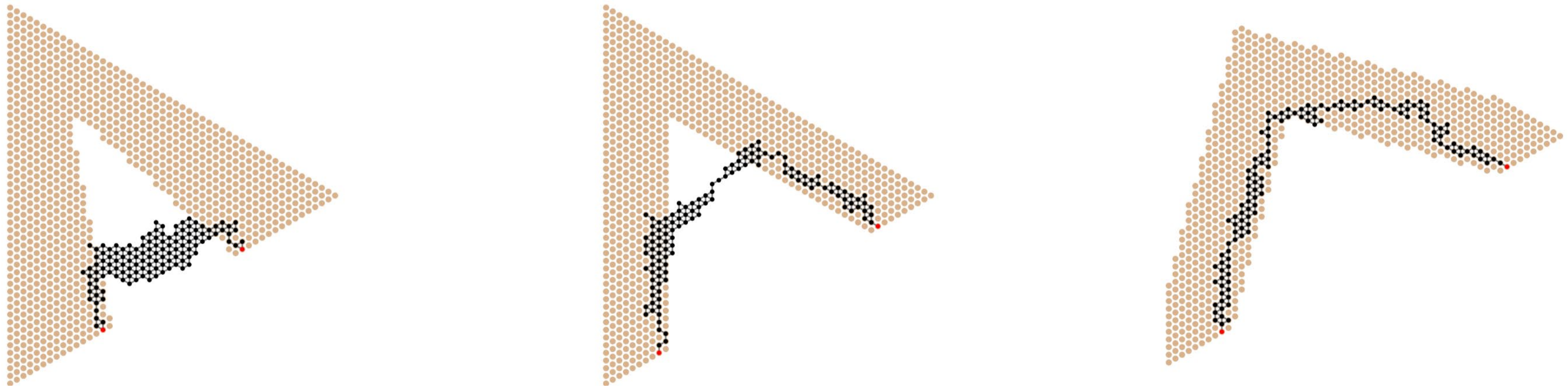
Dependence on Gap Angle

- For the *Eciton* army ants, the bridge which optimizes the tradeoff depends on the angle of the gap being shortcut.
- We proved similar behavior in our algorithm.



RLPKCG 2015: "Army ants dynamically…"

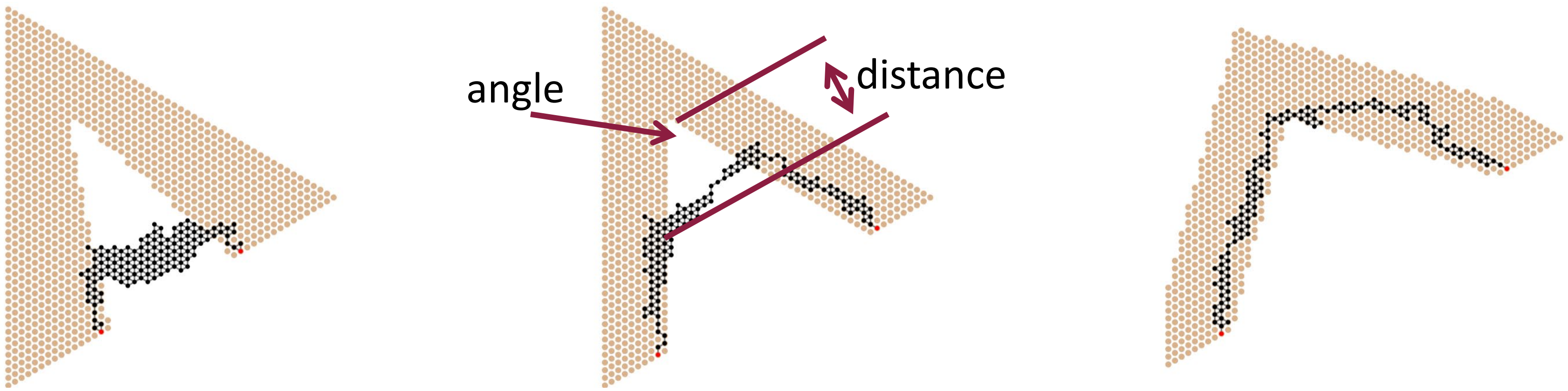
Simulations with λ = 4 and γ = 2 for angles of 30°, 60°, and 90°:



Dependence on Gap Angle

Theorem: For any $\lambda > 2 + \text{sqrt}(2)$ and $\gamma > 1$, there's an angle θ_1 (which depends on λ and γ) such that our algorithm has an exponentially small probability of forming a bridge "close to land" over any gap of smaller angle.

Theorem: For any $\lambda > 2 + \text{sqrt}(2)$ and $\gamma > (2 + \text{sqrt}(2))^4 \lambda^4$, there's a constant $\theta_2 > 60°$ such that our algorithm has an exponentially small probability of forming a bridge "far from land" over any gap with angle $60° < \theta < \theta_2$.



Local Properties for Movement

Input: an initial (connected, hole-free) configuration σ_0 and bias parameters λ, γ > 1.

Repeat:

1. Choose a particle from the system uniformly at random.
2. Choose an adjacent position uniformly at random. If this position is occupied, go to Step 1.
3. If properties hold for maintaining connectivity and avoiding holes move to the chosen position with probability $\min\{1, \lambda^{-\Delta p} \gamma^{-\Delta g}\}$.

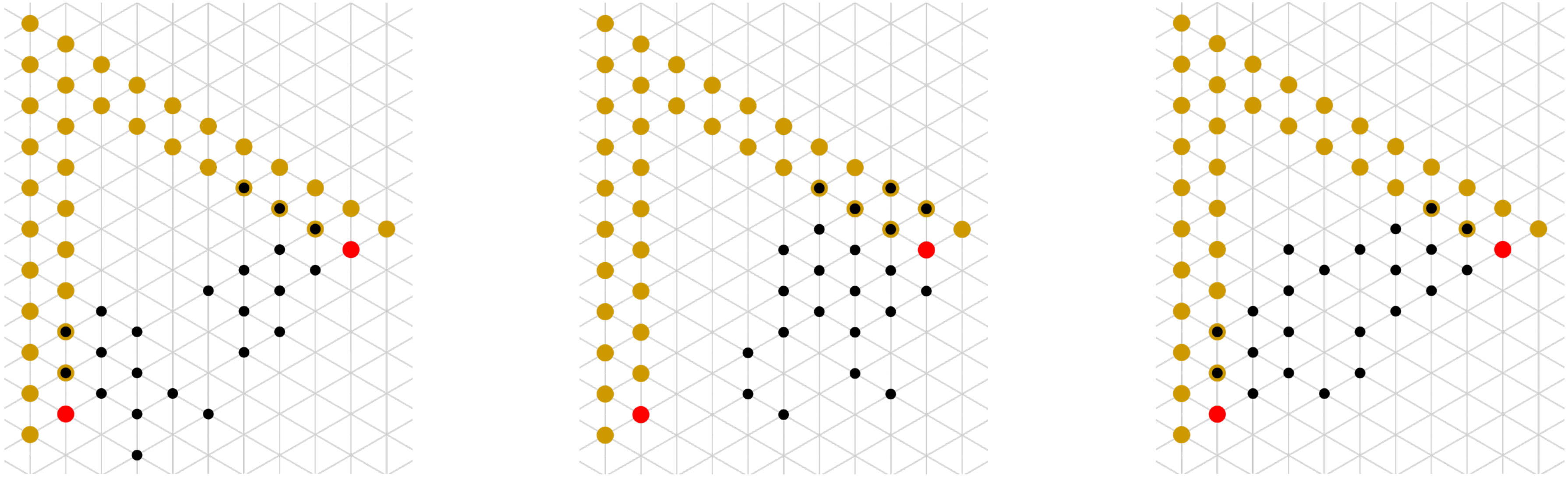
Theorem: We reach a stationary distribution π over configurations σ where $\pi(\sigma) \sim \lambda^{-p(\sigma)} \gamma^{-g(\sigma)}$.

Theorem: For any $\alpha > 1$, there are λ and γ (depending on α and c) so that in the stationary distribution $\pi(\sigma) \sim \lambda^{-p(\sigma)} \gamma^{-g(\sigma)}$ we have $p'(\sigma,c) \leq \alpha \cdot p'_{min}$ with high probability.

Local Properties for Movement

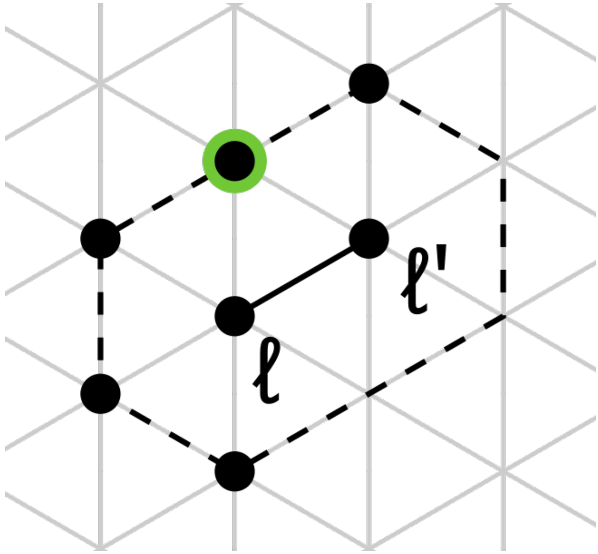
Qualitatively, what things do we **not** want to happen to our particle system?

- The particle system could become **disconnected** (within itself or from the land).
- A **hole** could be formed in the particle system.
- A move could be made which **couldn't be "undone"** (bad for reversibility).

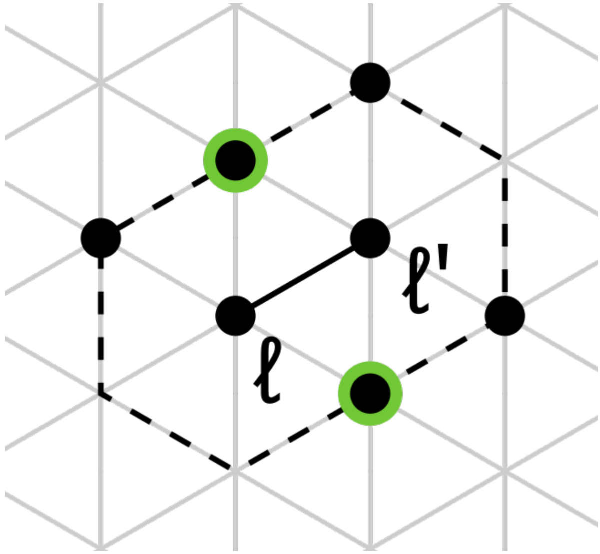


Local Properties for Movement

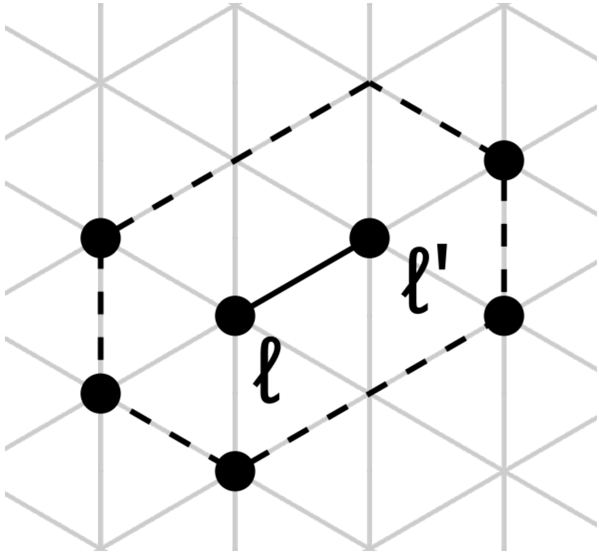
Allowed: Moves that avoid bad outcomes are either "slides" (1-2 pivots) or "jumps" (0 pivots).



1 pivot



2 pivots

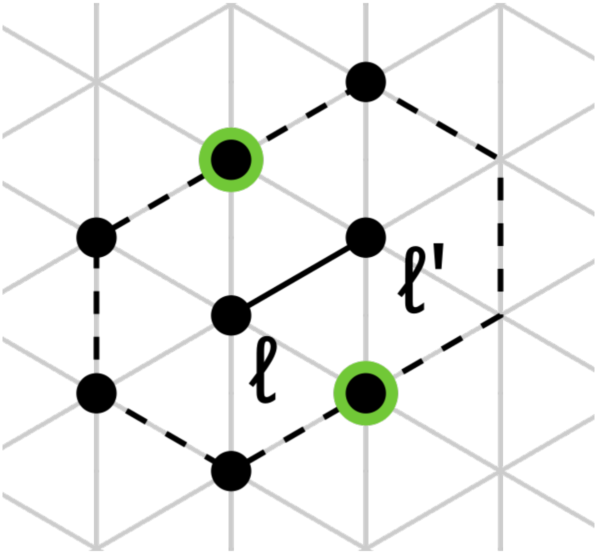


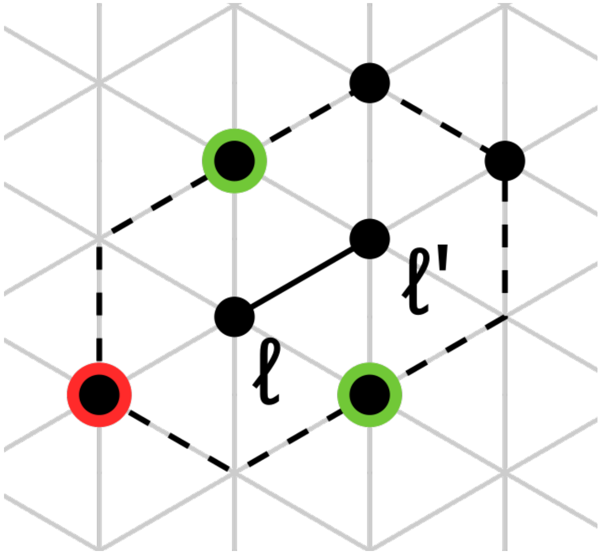
no pivot

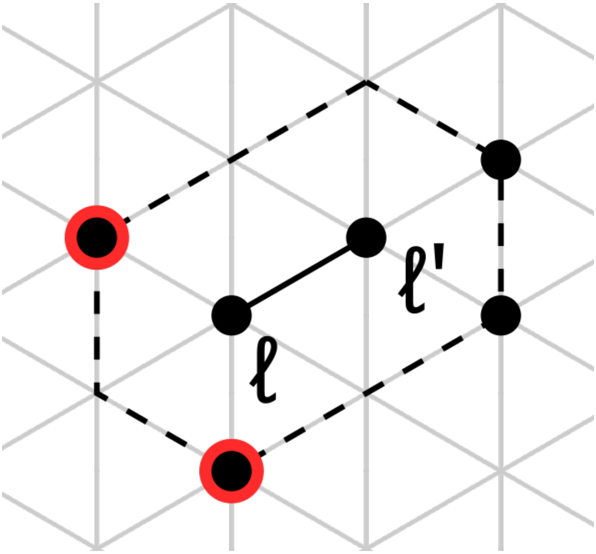
Local Properties for Movement

Not allowed: Moves that lead to disconnections or holes, or which are irreversible.

1. Current location should not have 5 neighbors (forms a hole).
2. 1-2 pivots: all neighbors should be locally connected to a pivot.
3. No pivot: both locations should have locally connected neighborhoods.







The Stationary Distribution

Input: an initial (connected, hole-free) configuration σ_0 and bias parameters λ, γ > 1.

Repeat:

1. Choose a particle from the system uniformly at random.
2. Choose an adjacent position uniformly at random. If this position is occupied, go to Step 1.
3. If properties hold for maintaining connectivity and avoiding holes, move to the chosen position with probability $\min\{1, \lambda^{-\Delta p}\gamma^{-\Delta g}\}$.

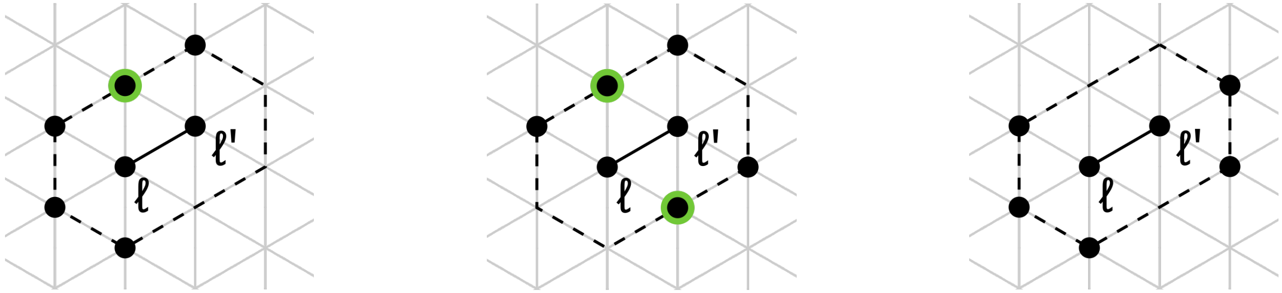
Theorem: We reach a stationary distribution π over configurations σ where $\pi(\sigma) \sim \lambda^{-p(\sigma)}\gamma^{-g(\sigma)}$.

Theorem: For any $\alpha > 1$, there are λ and γ (depending on α and c) so that in the stationary distribution $\pi(\sigma) \sim \lambda^{-p(\sigma)}\gamma^{-g(\sigma)}$ we have $p'(\sigma,c) \leq \alpha \cdot p'_{min}$ with high probability.

The Stationary Distribution

Our local rules for movement give us that:

- The particle system remains connected and anchored to the land objects.
- No holes form in the system.
- All moves are reversible.



Conjecture: It is possible to go from any particle system configuration which is anchored to the land objects to any other such configuration.

Theorem: The Markov chain is ergodic, and thus has a unique stationary distribution π.

The Stationary Distribution

A **Metropolis filter** can be used to design the right transition probabilities to obtain a desired π.

- Recall: we want to minimize **weighted perimeter** $p'(\sigma,c) = p(\sigma) + c \cdot g(\sigma)$, where $c > 0$.
- So set the desired weight of a configuration at stationarity to be $\pi(\sigma) \sim \eta^{-p'(\sigma,c)}$, with $\eta > 1$.

$$\pi(\sigma) \sim \eta^{-p'(\sigma,c)} = \eta^{-p(\sigma) - c \cdot g(\sigma)} = \eta^{-p(\sigma)} \cdot \eta^{c \cdot -g(\sigma)} = \lambda^{-p(\sigma)} \gamma^{-g(\sigma)} \text{, with } \lambda, \gamma > 1.$$

- Set the transition probability from σ to τ using a Metropolis filter:

$$P(\sigma,\tau) = \Pr[(\sigma \text{ to } \tau) \text{ being proposed}] \cdot \min\{ 1, \pi(\tau) / \pi(\sigma) \} = (1/6n) \cdot \min\{ 1, \pi(\tau) / \pi(\sigma) \}.$$

- Now, we can use what we want π to look like:

$$\pi(\tau) / \pi(\sigma) = (\lambda^{-p(\tau)} \gamma^{-g(\tau)} / Z) / (\lambda^{-p(\sigma)} \gamma^{-g(\sigma)} / Z) = \lambda^{-p(\tau) + p(\sigma)} \gamma^{-g(\tau) + g(\sigma)} = \lambda^{-\Delta p} \gamma^{-\Delta g}.$$

Detailed Balance

Input: an initial (connected, hole-free) configuration σ_0 and bias parameters λ, γ > 1.

Repeat:

1. Choose a particle from the system uniformly at random.
2. Choose an adjacent position uniformly at random. If this position is occupied, go to Step 1.
3. If properties hold for maintaining connectivity and avoiding holes, move to the chosen position $\boxed{\text{with probability } \min\{1, \lambda^{-\Delta p}\,\gamma^{-\Delta g}\}}$.

$\boxed{\textbf{Theorem:} \text{ We reach a stationary distribution } \pi \text{ over configurations } \sigma \text{ where } \pi(\sigma) \sim \lambda^{-p(\sigma)}\,\gamma^{-g(\sigma)}}$.

Theorem: For any $\alpha > 1$, there are λ and γ (depending on α and c) so that in the stationary distribution $\pi(\sigma) \sim \lambda^{-p(\sigma)}\,\gamma^{-g(\sigma)}$ we have $p'(\sigma,c) \leq \alpha \cdot p'_{min}$ with high probability.

Detailed Balance

Theorem: We reach a stationary distribution π over configurations σ where $\pi(\sigma) \sim \lambda^{-p(\sigma)} \gamma^{-g(\sigma)}$.

How do we know the Metropolis filter gets us where we want to go?

Proof:

- π is the stationary distribution if $\pi(\sigma)\,P(\sigma,\tau) = \pi(\tau)\,P(\tau,\sigma)$.
- Without loss of generality, suppose $\lambda^{p(\sigma) - p(\tau)} \gamma^{g(\sigma) - g(\tau)} \leq 1$. Then:

$$\pi(\sigma)\,P(\sigma,\tau) = (\lambda^{-p(\sigma)} \gamma^{-g(\sigma)} / Z) \cdot (1/6n) \cdot \min\{\,1, \lambda^{p(\sigma) - p(\tau)} \gamma^{g(\sigma) - g(\tau)}\,\}$$

$$= (\lambda^{-p(\sigma)} \gamma^{-g(\sigma)} \lambda^{p(\sigma) - p(\tau)} \gamma^{g(\sigma) - g(\tau)} / Z) \cdot (1/6n)$$

$$= (\lambda^{-p(\tau)} \gamma^{-g(\tau)} / Z) \cdot (1/6n) \cdot 1$$

$$= \pi(\tau)\,P(\tau,\sigma).$$

Correctness: A Peierls Argument

Input: an initial (connected, hole-free) configuration σ_0 and bias parameters λ, $\gamma > 1$.

Repeat:

1. Choose a particle from the system uniformly at random.
2. Choose an adjacent position uniformly at random. If this position is occupied, go to Step 1.
3. If properties hold for maintaining connectivity and avoiding holes, move to the chosen position with probability $\min\{1, \lambda^{-\Delta p} \gamma^{-\Delta g}\}$.

Theorem: We reach a stationary distribution π over configurations σ where $\pi(\sigma) \sim \lambda^{-p(\sigma)} \gamma^{-g(\sigma)}$.

Theorem: For any $\alpha > 1$, there are λ and γ (depending on α and c) so that in the stationary distribution $\pi(\sigma) \sim \lambda^{-p(\sigma)} \gamma^{-g(\sigma)}$ we have $p'(\sigma,c) \leq \alpha \cdot p'_{min}$ with high probability.

Correctness: A Peierls Argument

Theorem: For any $\alpha > 1$, there are λ and γ (depending on α and c) so that in the stationary distribution $\pi(\sigma) \sim \lambda^{-p(\sigma)} \gamma^{-g(\sigma)}$ we have $p'(\sigma,c) \leq \alpha \cdot p'_{min}$ with high probability.

How do we know our algorithm actually minimizes weighted perimeter?

Proof sketch:

- Let S_α be the set of configurations σ with $p'(\sigma,c) > \alpha \cdot p'_{min}$ (i.e., the bad ones).
- We'll show that it is exponentially unlikely to be in such a bad configuration, i.e.:

$$\pi(S_\alpha) \leq \delta^{sqrt(n)}, \text{ where } \delta < 1.$$

- Let $p'_1, p'_2, \ldots, p'_m$ be all the possible values of $p'(\sigma,c) = p(\sigma) + c \cdot g(\sigma)$.
- Let A_i be the set of "bad" configurations in S_α with $p'(\sigma,c) = p'_i$.
- How many configurations are in A_i?

Correctness: A Peierls Argument

Theorem: For any $\alpha > 1$, there are λ and γ (depending on α and c) so that in the stationary distribution $\pi(\sigma) \sim \lambda^{-p(\sigma)} \gamma^{-g(\sigma)}$ we have $p'(\sigma,c) \leq \alpha \cdot p'_{min}$ with high probability.

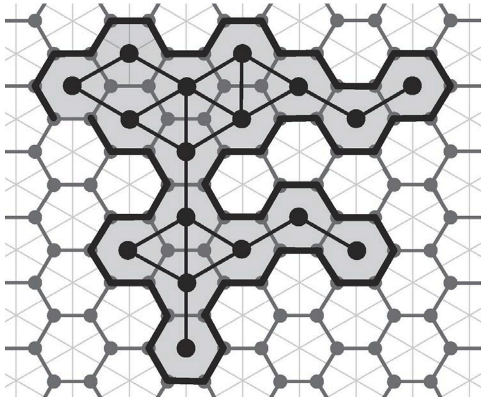
Proof sketch:

- **Theorem:** (from compression). There are at most $f(p)(2 + \mathrm{sqrt}(2))^p$ configurations with perimeter p, where f is subexponential.
- Any configuration σ in A_i has perimeter $p(\sigma) \leq p(\sigma) + c \cdot g(\sigma) = p'_i$, so:

$$|A_i| \leq f(p'_i)(2 + \mathrm{sqrt}(2))^{p'_i}$$

- Now we can calculate $\pi(A_i)$:

$$\pi(A_i) = \lambda^{-p(\sigma)} \gamma^{-g(\sigma)} \cdot |A_i| / Z \leq \lambda^{-p(\sigma)} \gamma^{-g(\sigma)} \cdot f(p'_i)(2 + \mathrm{sqrt}(2))^{p'_i} / Z.$$



Correctness: A Peierls Argument

Theorem: For any $\alpha > 1$, there are λ and γ (depending on α and c) so that in the stationary distribution $\pi(\sigma) \sim \lambda^{-p(\sigma)} \gamma^{-g(\sigma)}$ we have $p'(\sigma,c) \leq \alpha \cdot p'_{min}$ with high probability.

Proof sketch:

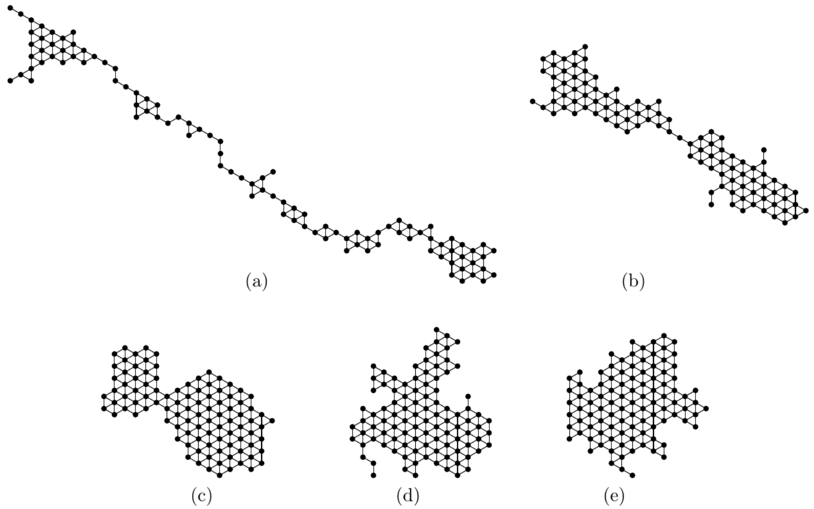
- The last step is to sum up all the $\pi(A_i)$ values to find $\pi(S_\alpha)$.
- There are $m \leq O(n^2)$ of them, since both $O(\mathrm{sqrt}(n)) \leq p(\sigma), g(\sigma) \leq 2n-2$.
- Carrying out the algebra from there, we get:

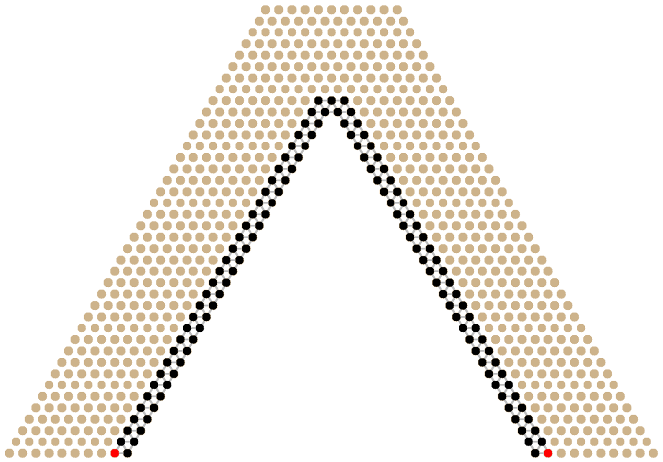
$$\pi(S_\alpha) = \sum_{i=1,\ldots,m} \pi(A_i) \leq \ldots \leq f(n)\, \delta^{\mathrm{sqrt}(n)}.$$

Stochasticity in Programmable Matter

(Recall) Stochastic algorithms exist for:

- **Compression**, or gathering a particle system together as tightly as possible. (Cannon, Daymude, Randall, and Richa @ PODC 2016).
- **Shortcut bridging**, what we saw in this talk. (Andrés Arroyo, Cannon, Daymude, Randall, and Richa @ DNA23).

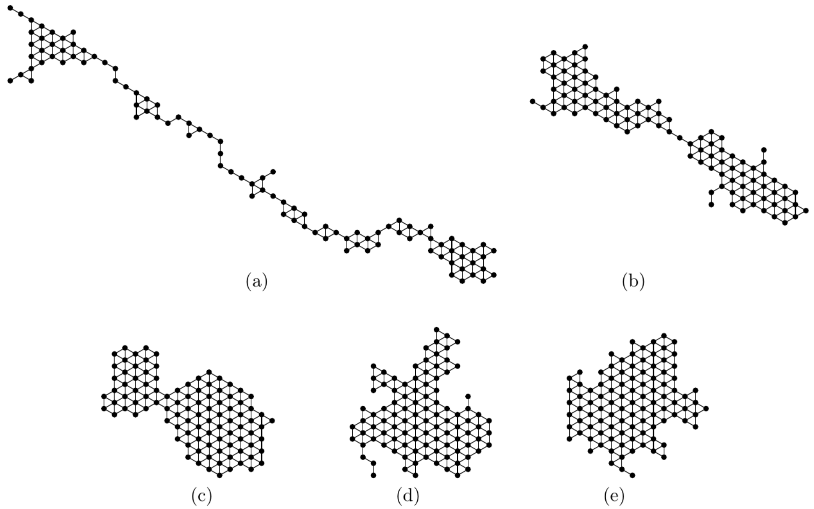


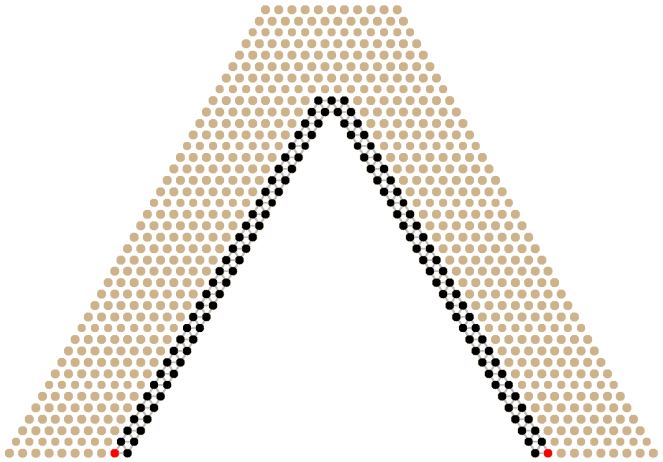


Stochasticity in Programmable Matter

Advantages of the stochastic, distributed, local approach:

- Completely decentralized (no leader necessary for coordination).
- Robust to crash/deletion failures and is self-stabilizing.
- Very little communication needed (1 bit is used for conflict resolution).





Stochasticity in Programmable Matter

Good candidate problems for the stochastic, distributed, local approach:

- Desired behavior optimizes some global energy function. For example, in shortcut bridging:

  minimize ***total perimeter*** *and minimize* ***gap perimeter*** -> $\pi(\sigma) \sim \lambda^{-p(\sigma)} \gamma^{-g(\sigma)}$.

- Changes in global energy resulting from one-step transitions can be calculated using only local information. For example, in shortcut bridging:

  $\pi(\sigma) \sim \lambda^{-p(\sigma)} \gamma^{-g(\sigma)}$ -> move with probability $\min\{1, \lambda^{-\Delta p} \gamma^{-\Delta g}\}$.

Future Work & Open Questions

Further extensions of our stochastic approach:

- Explore systems with heterogenous bias parameters.
- Investigate behaviors when particles can change their bias parameters over time.
- Mix this stochastic approach with non-stochastic elements.

What is the mixing time of our compression and shortcut bridging chains?

- Seems difficult to analyze, though in compression simulation it's $\approx O(n^{3.3})$.

Are there critical values for λ and γ (or the ratio between them) which cause a phase transition?

Collaborators

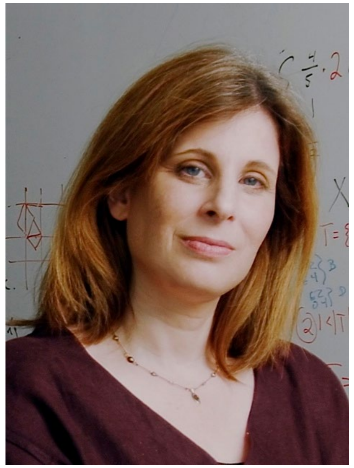
Arizona State University

Georgia Tech

UNIVERSIDAD DE GRANADA

Andréa W. Richa

Joshua J. Daymude



Dana Randall

Sarah Cannon

Marta Andrés Arroyo

Thank you!

sops.engineering.asu.edu