Discrete Coverage Control for Gossiping Robots

Joseph W. Durham

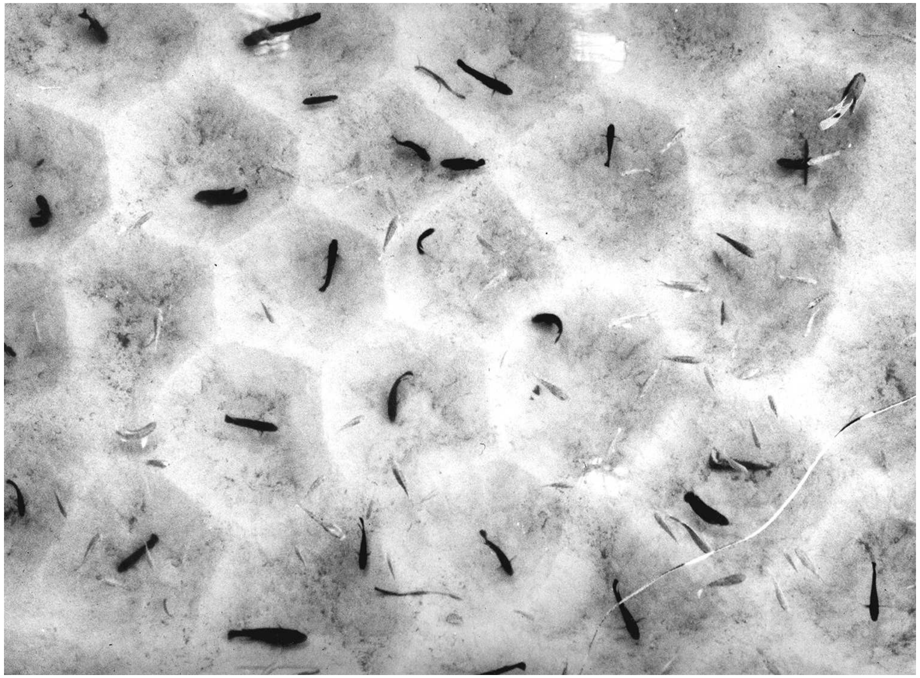
Center for Control, Dynamical Systems and Computation
University of California at Santa Barbara
http://motion.mee.ucsb.edu/~joey

DSC Conference, Hollywood, Oct 2009

Collaborators: Paolo Frasca, Ruggero Carli, Francesco Bullo

UCSB

Big picture



G. W. Barlow. Hexagonal Territories.
Animal Behaviour, 22(4):876-878, 1974

What is coverage control?

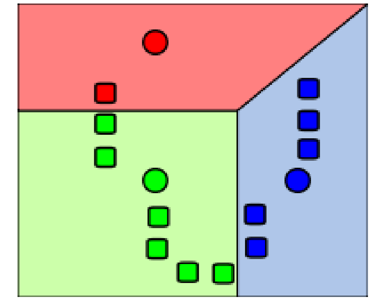
- Partition environment into territories for each agent
- Robotic load balancing

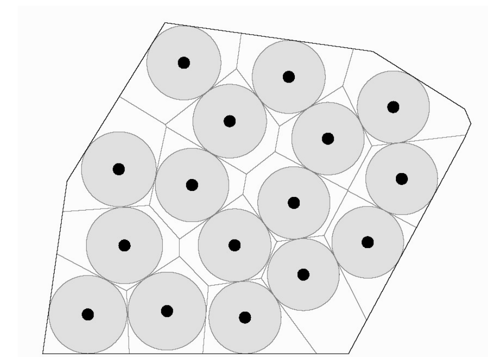
Why gossip communication?

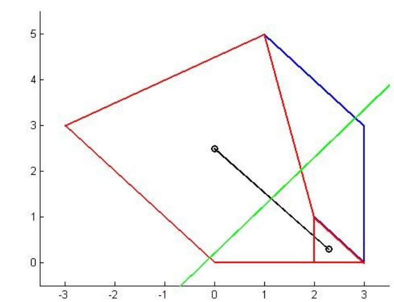
- Pairwise neighbor interactions are minimum necessary to achieve goal

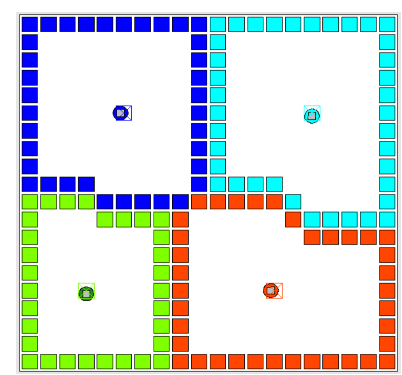
Outline

- Lineage of this work
  - Lloyd's algorithm
  - Distributed partitioning & centering
  - Gossip coverage
  - Discrete gossip coverage
- Gossip as minimal communication
- Partitioning and centering on a graph
- Example simulation
- Convergence & complexity results









UCSB

Lloyd's algorithm

- Place N robots at $c = \{c_1 \dots c_N\}$
- Partition environment into $p = \{p_1 \dots p_N\}$
- Define cost as expected distance:

$$H(c, p) = \int_{p_1} \|q - c_1\| dq + \dots + \int_{p_N} \|q - c_N\| dq$$

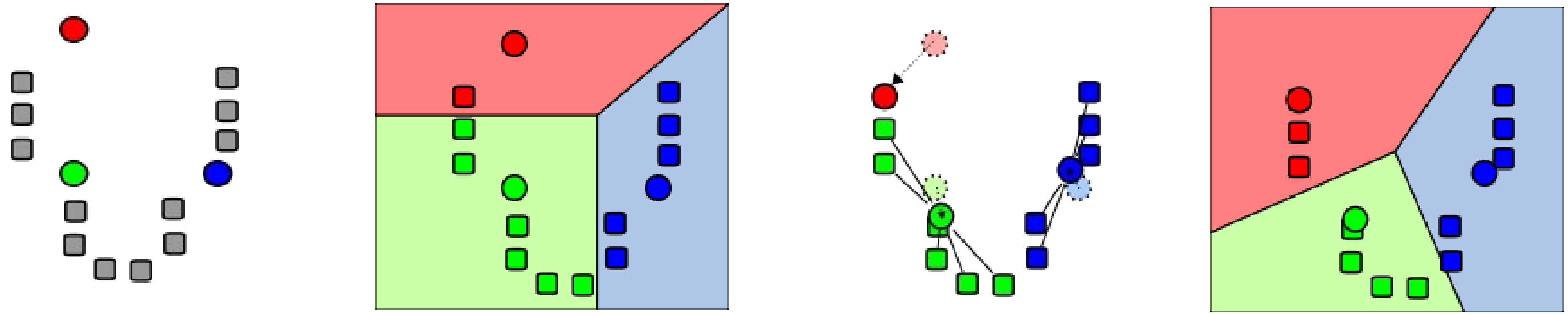
Lloyd's algorithm

Theorem (Lloyd '57 “least-square quantization”)

- For fixed partition, optimal positions are centroids
- For fixed positions, optimal partition is Voronoi

Lloyd's algorithm:

- Alternate position/partition optimization
- Result: convergence to a centroidal voronoi partition



Wikimedia Commons

Lloyd's algorithm

- Place N robots at $c = \{c_1 \dots c_N\}$
- Partition environment into $p = \{p_1 \dots p_N\}$
- Define cost as expected distance:

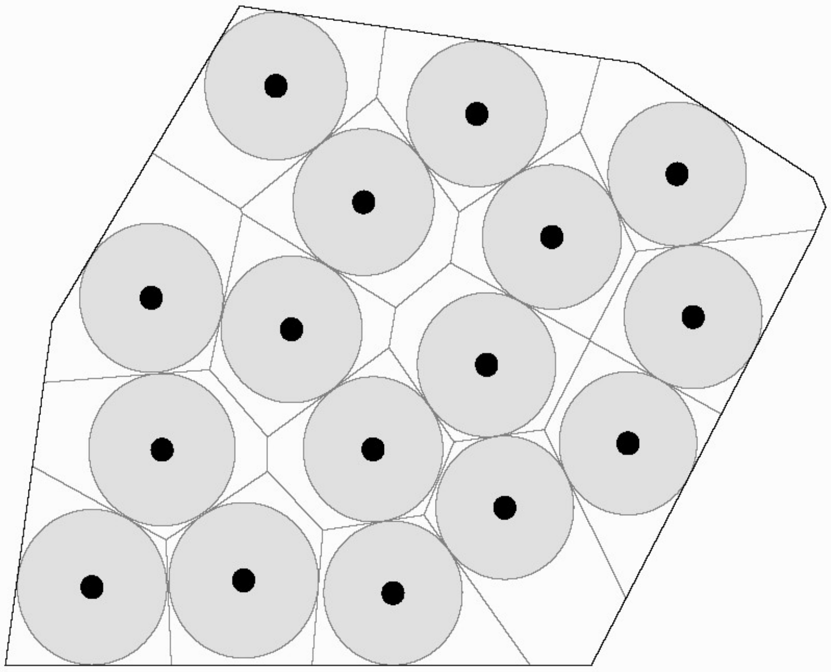
$$H(c, p) = \int_{p_1} \|q - c_1\| dq + \dots + \int_{p_N} \|q - c_N\| dq$$

Theorem (Lloyd '57 "least-square quantization")
- For fixed partition, optimal positions are centroids
- For fixed positions, optimal partition is Voronoi

Lloyd's algorithm:
- Alternate position/partition optimization
- Result: convergence to a centroidal voronoi partition

Distributed partitioning and centering law

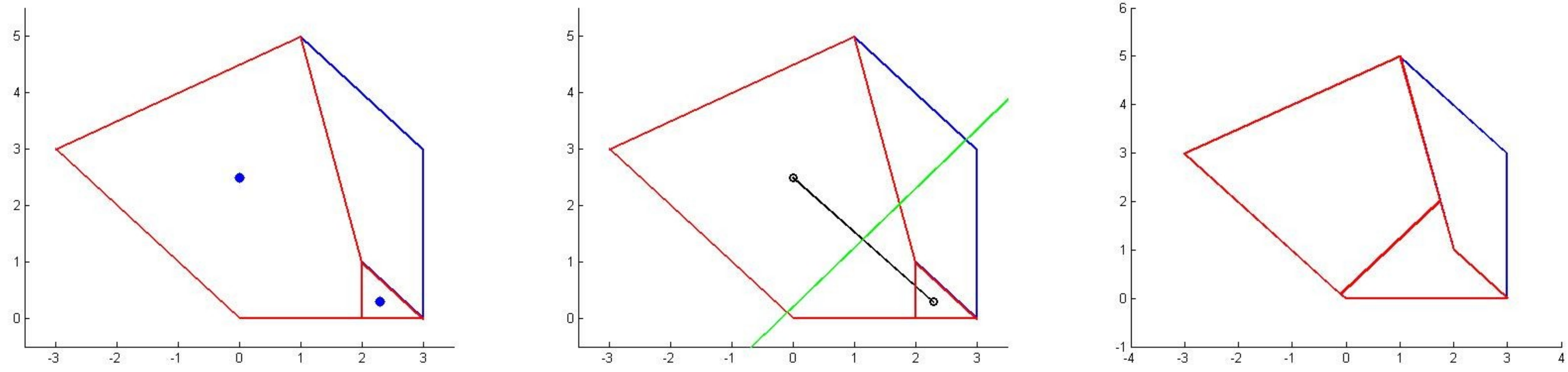
- At each comm round:
  - Acquire neighbor positions
  - Compute own Voronoi region
  - Move towards centroid of own Voronoi region
- Result: convergence to a centroidal Voronoi partition



J. Cortés, S. Martínez, T. Karatas, and F. Bullo. Coverage control for mobile sensing networks.
IEEE Trans Robotics & Automation, 20(2):243–255, 2004

Gossip communication

- Pairwise territory exchange between neighbors
- Partition regions based on bisector of centroids
- Result: Convergence to the set of centroidal Voronoi partitions



P. Frasca, R. Carli, and F. Bullo. Multiagent coverage algorithms with gossip communication: control systems on the space of partitions.
In American Control Conference, St. Louis, MO, pages 2228-2235, June 2009.

Discrete gossip coverage contributions

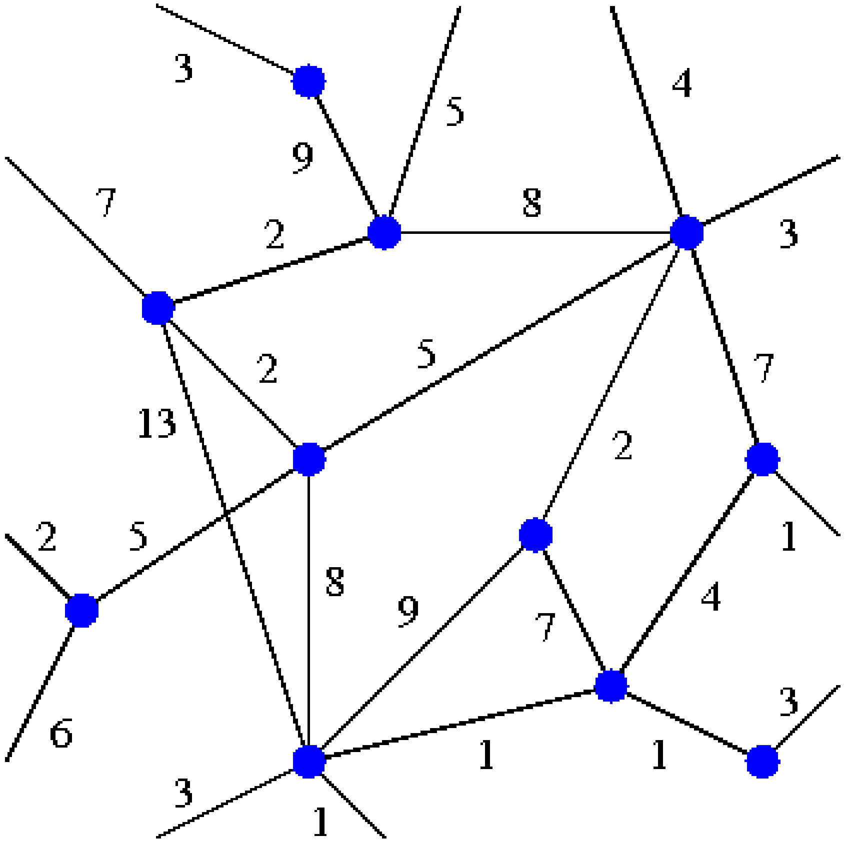
- **Main novelty:** graph representation of environment allows
  - Non-convex environments with holes
  - Hardware implementation
- Convergence to a single centroidal Voronoi partition of the graph in finite time
- Computational complexity results

Graph of environment

- Domain of new method is a weighted graph:

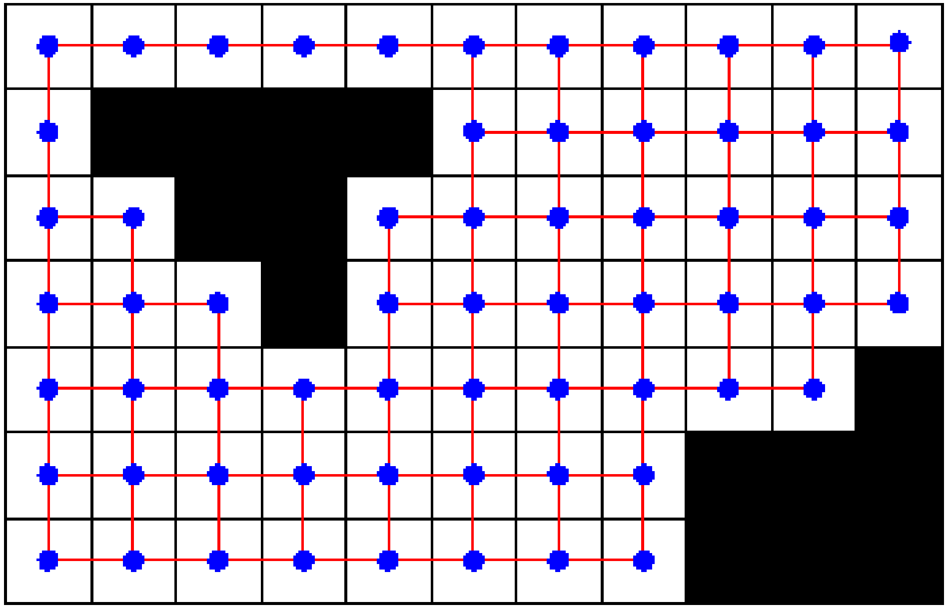
$$G = (Q, E, w)$$

- Required properties:
  - Connected
  - Positive edge weights

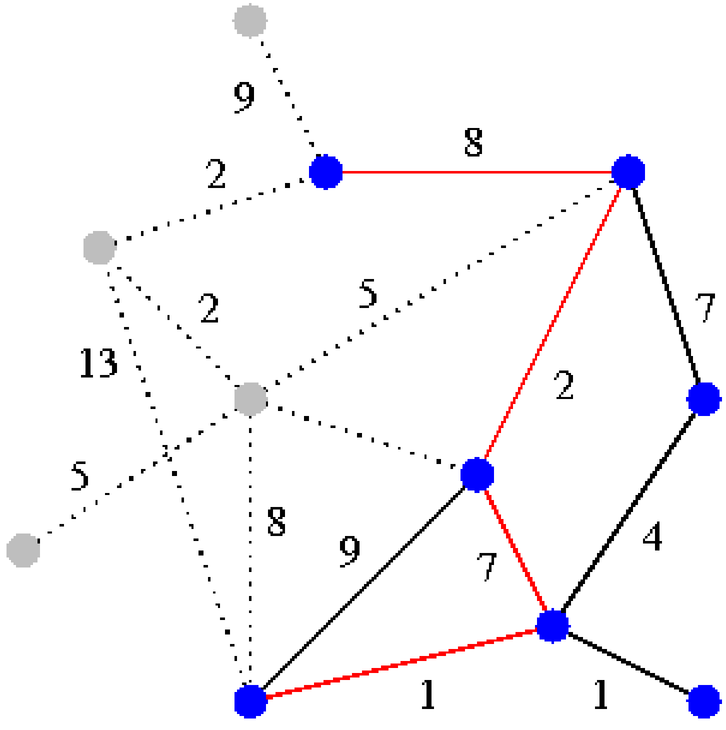


Occupancy grid graph

- Occupancy grid map
  - "Free" cells where no obstacle
- Graph representation
  - Free cells are vertices
  - Adjacent free cells are neighbors
  - All edge weights are grid resolution



Distances and sub-graphs

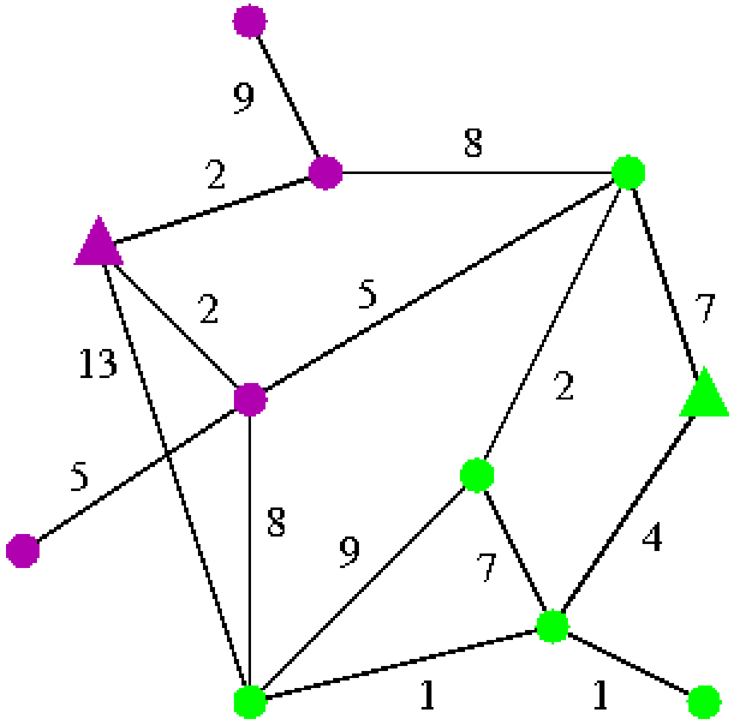


- $d_G(h, k)$ is the minimal weighted path length from vertex h to k in G

- p_i is a connected sub-graph of G induced by a subset of Q

$$d_{p_i}(h,k) \geq d_G(h,k) \quad \forall\, h, k \in p_i$$

Voronoi partition of a sub-graph



- Path distances in graph
- Vertex joins partition of centroid it is closest to
- Ties must be handled so partitions are connected

Centroids & cost function

- Centroid c_i of p_i is vertex which minimizes:

$$H_i(h, p_i) = \sum_{k \in p_i} d_{p_i}(h, k)$$

- Total cost function:

$$H_{\text{multi-center}}(c, p) = \sum_{i=1}^{N} H_i(c_i, p_i)$$

- Minimize expected distance robot i has to travel to reach a randomly selected vertex in p_i

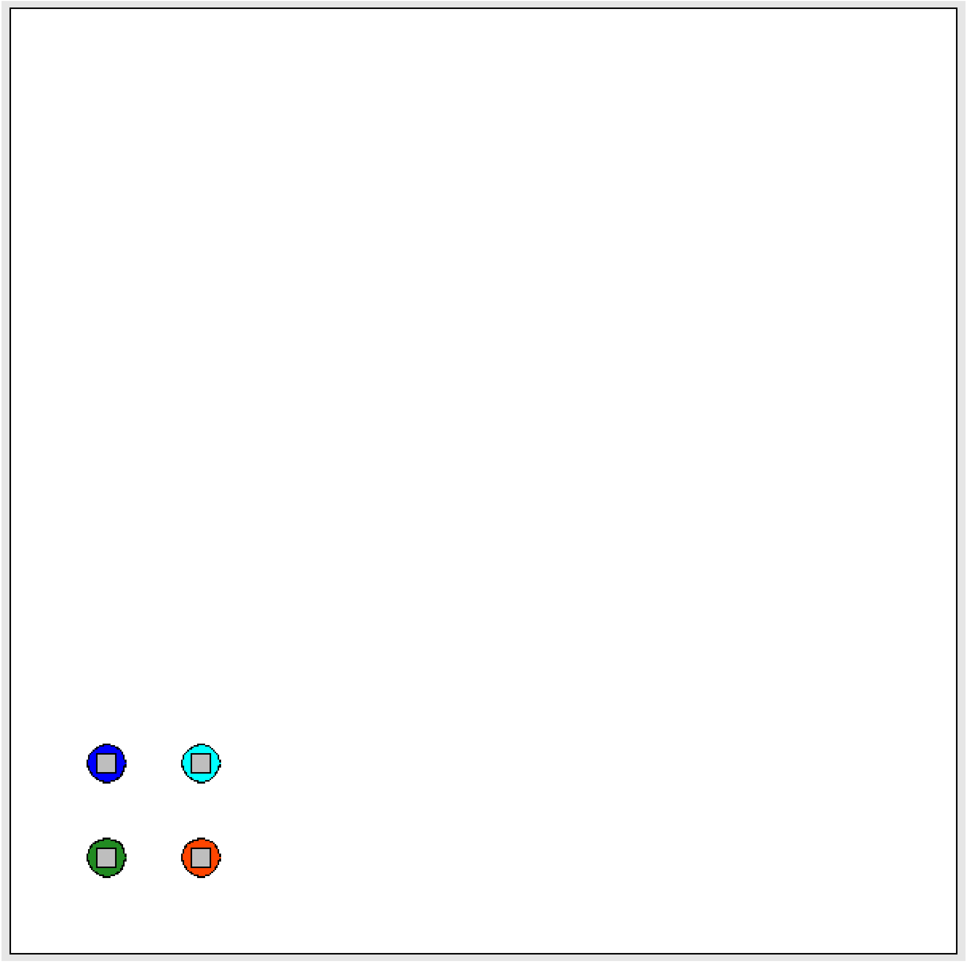
The algorithm

- Each agent i stores:
  - Sub-graph p_i
  - Centroid c_i
- When random neighboring pair communicate:
  - Find union of sub-graphs $p_i \cup p_j$
  - Compute Voronoi partition of union based on distances from c_i, c_j inside of $p_i \cup p_j$
  - Update c_i for new p_i, c_j for new p_j

Properties of the algorithm

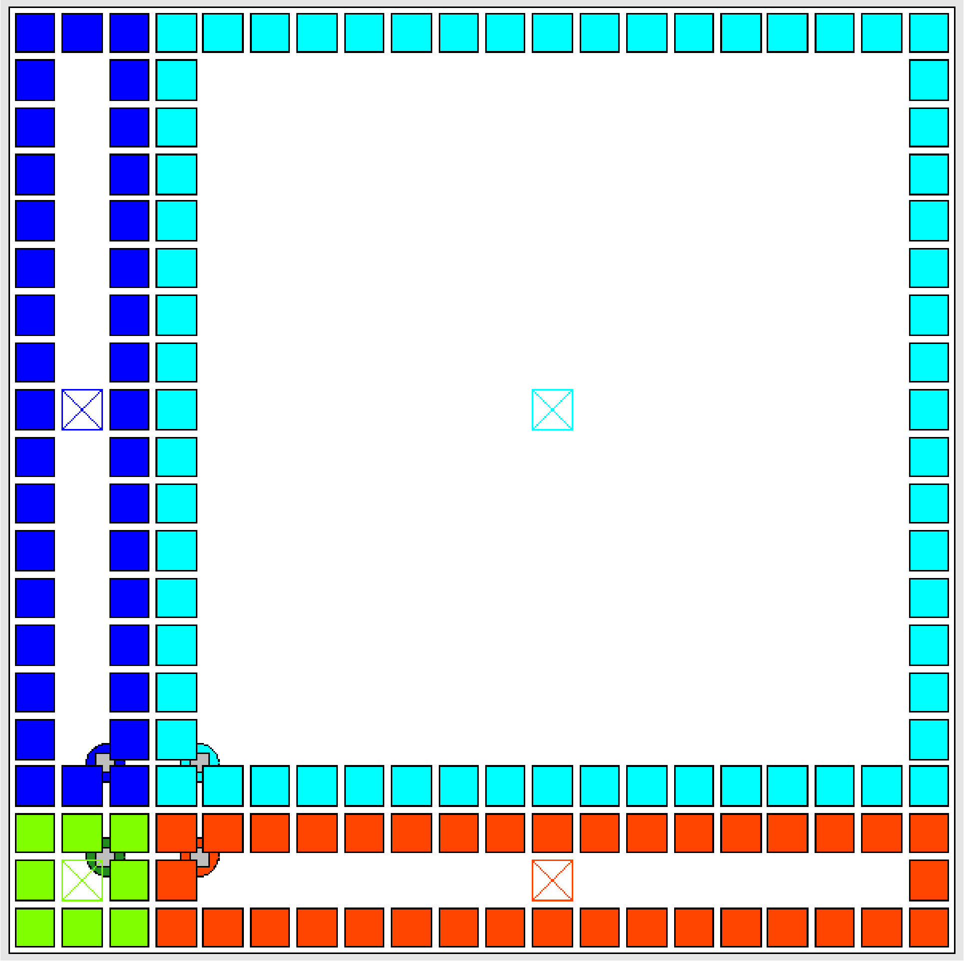
- Each p_i will remain connected during evolution
  - Result of partitioning based on distances in $p_i \cup p_j$
  - Centroid c_i is always a vertex of p_i
  - Therefore, cost function is well-defined
- Total cost decreases whenever partition or centroids change

A simple example



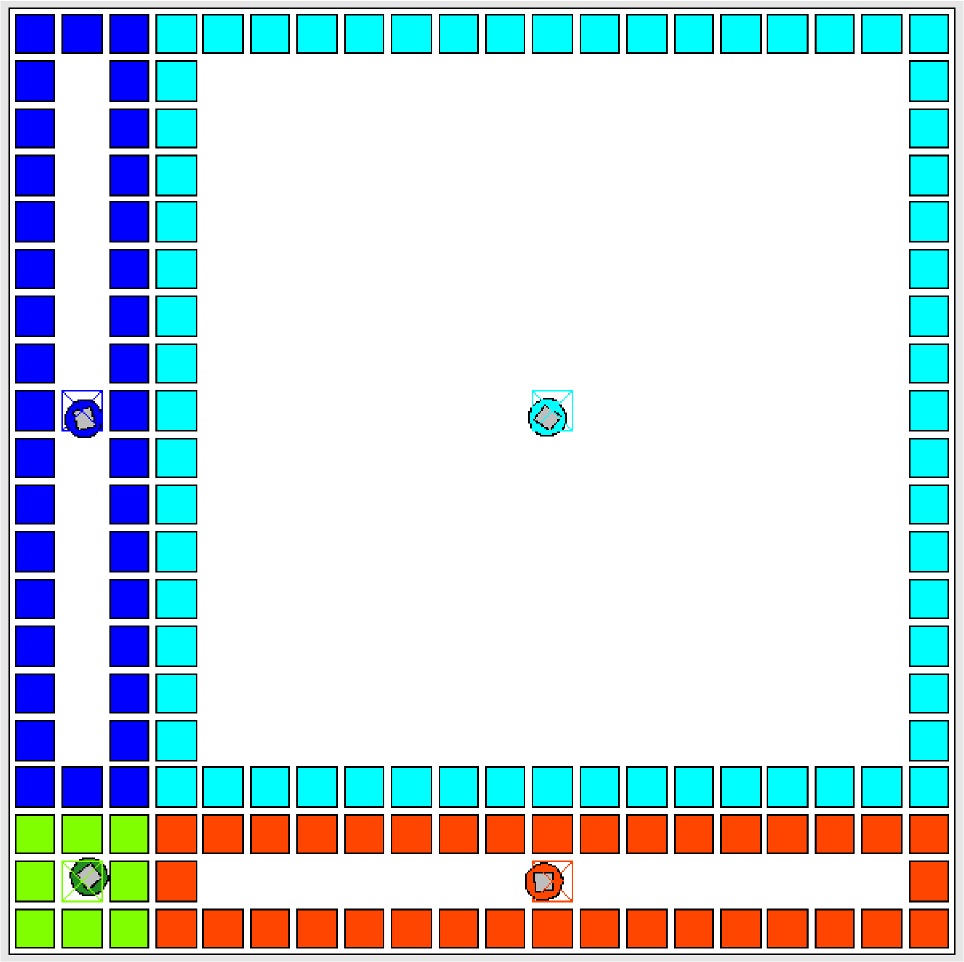
Four robots in an empty square room, simulated using Player/Stage

A simple example



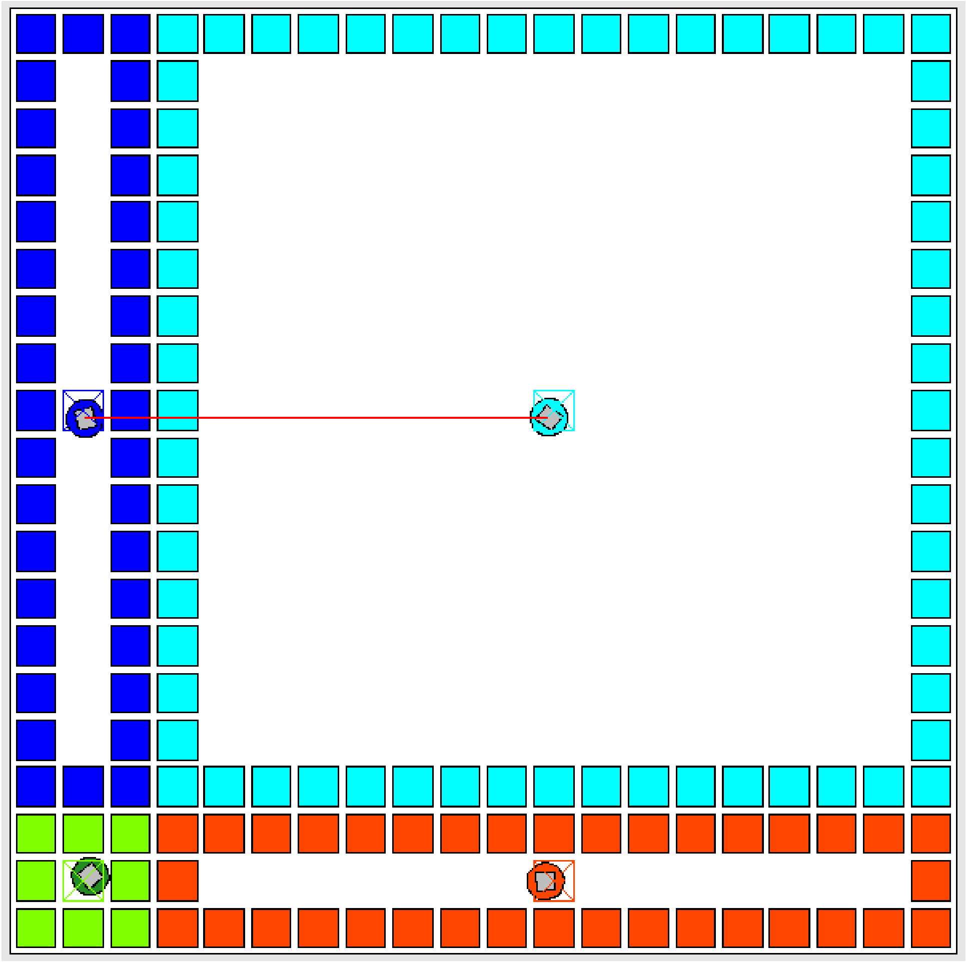
Initial partitioning of the environment

A simple example



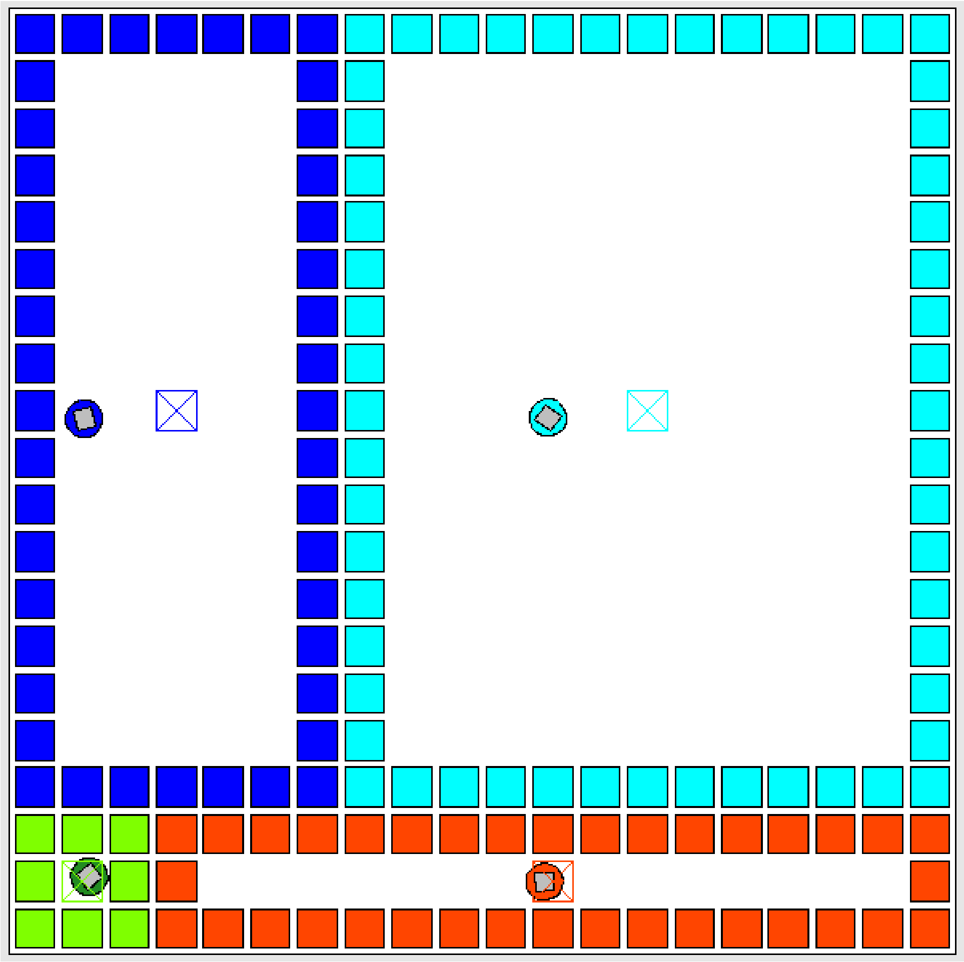
UCSB

A simple example



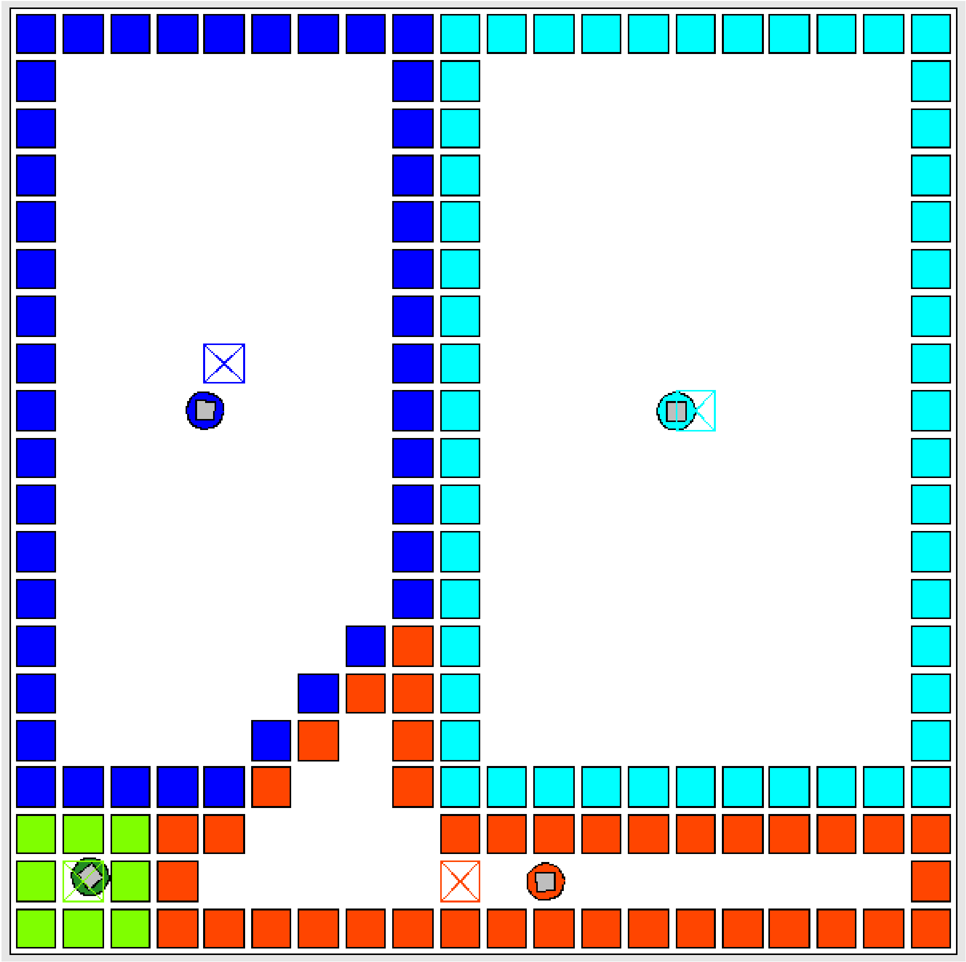
First pairwise communication

A simple example



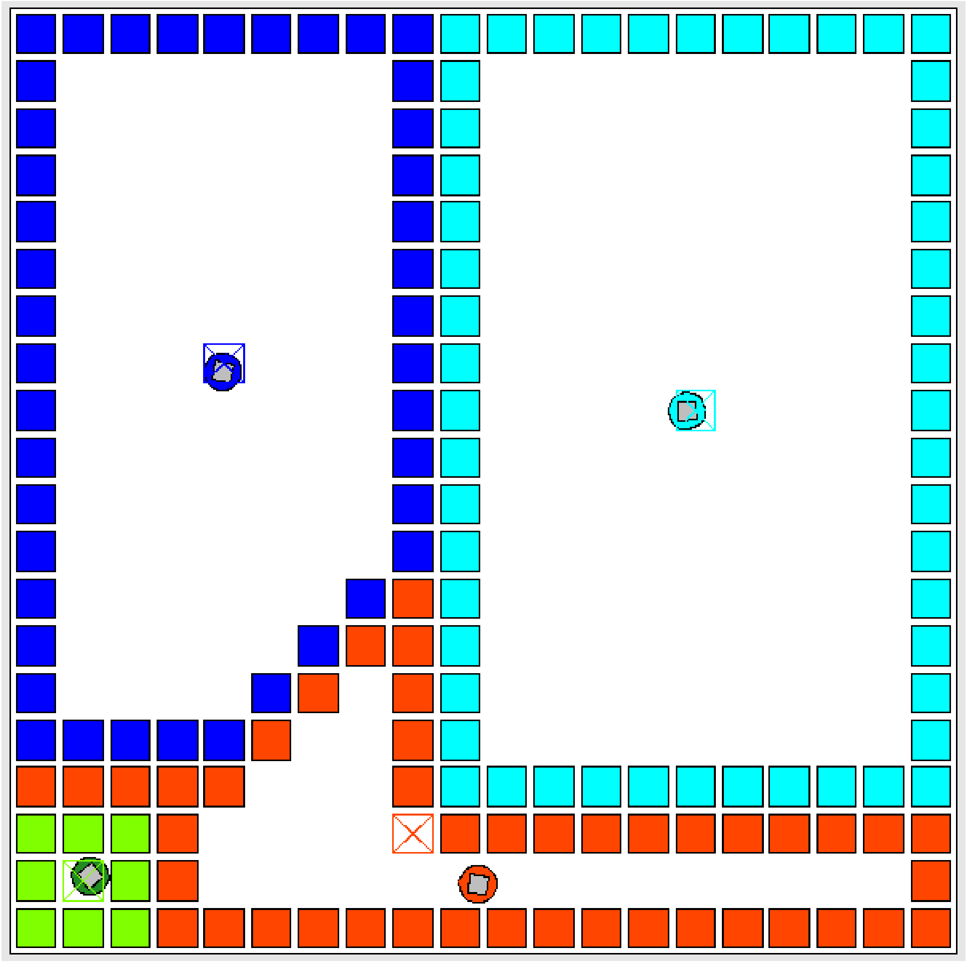
Result of first pairwise territory swap – Dark blue takes cells from Cyan

A simple example



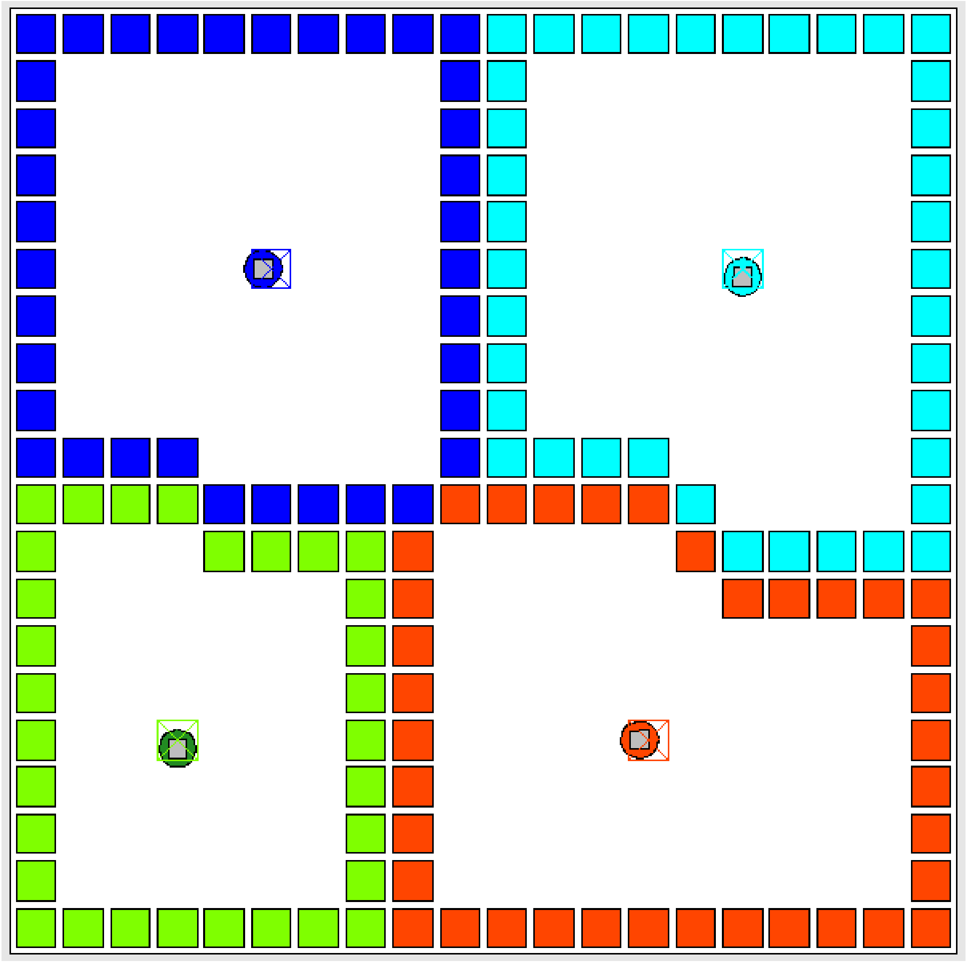
Second pairwise territory swap – Red takes cells from Dark blue

A simple example



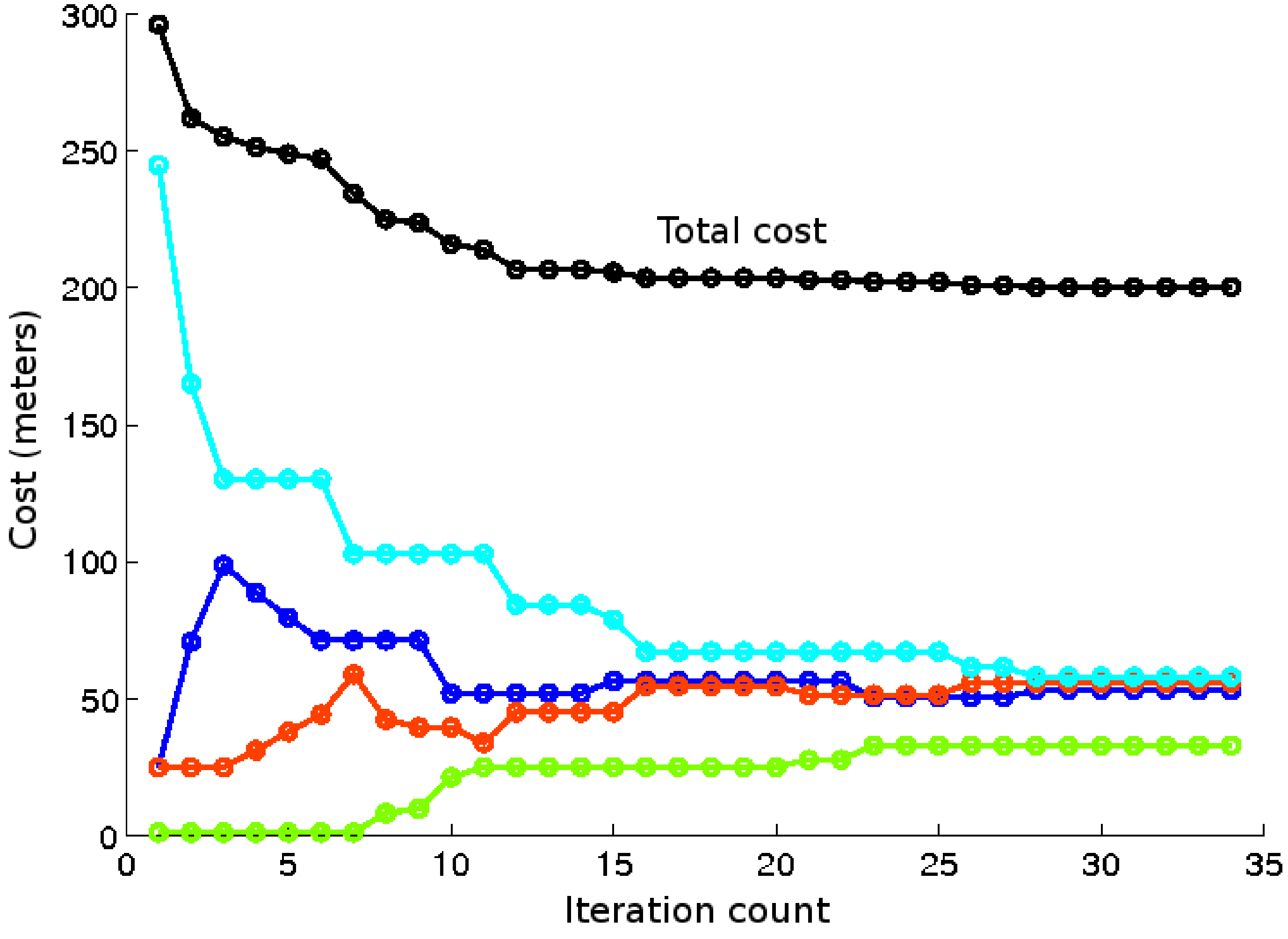
Third pairwise territory swap – Red again takes cells from Dark Blue

A simple example



Final equilibrium territories

A simple example



Cost functions over iterations

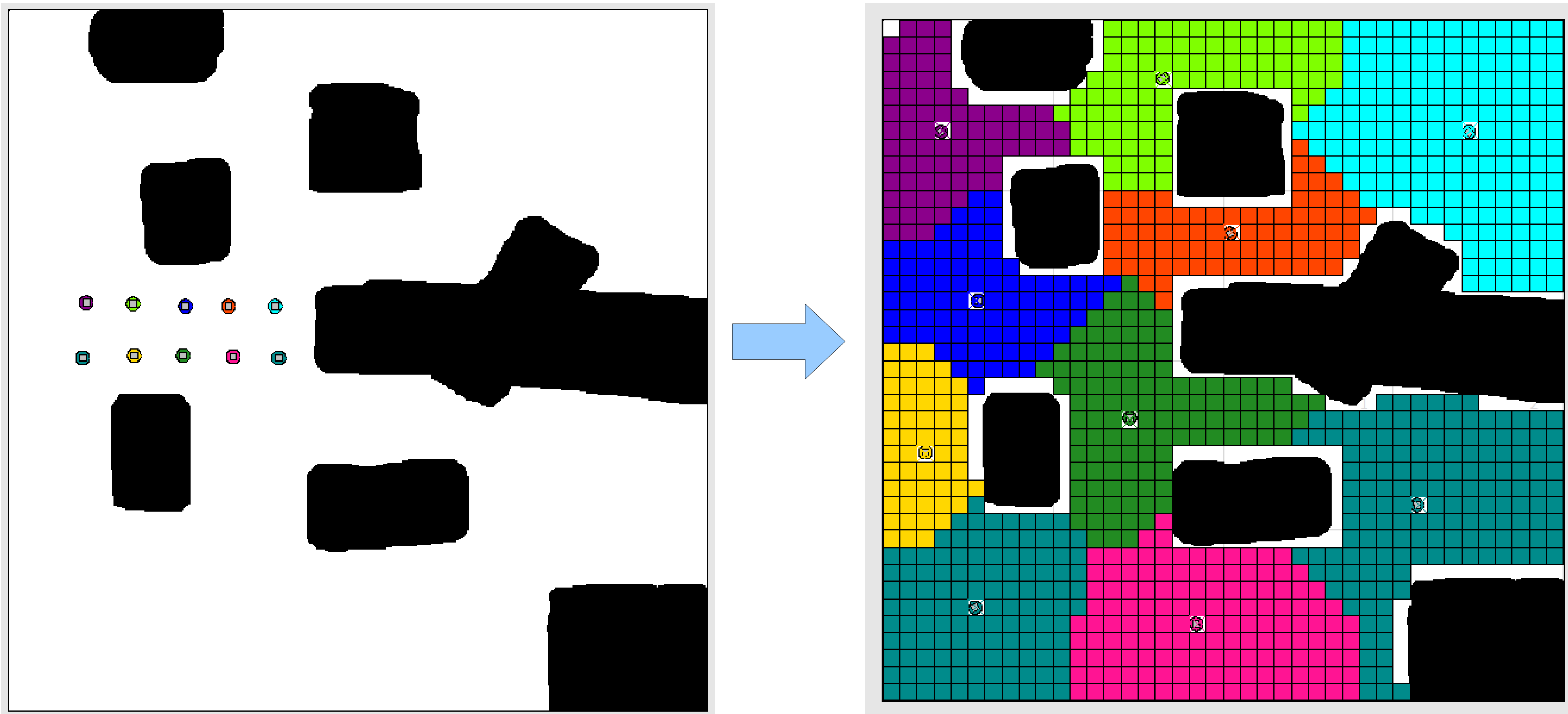
Convergence proof sketch

- Extension of LaSalle invariance principle
  - State space P: finite set of connected N-partitions of G
  - Algorithm defines a set-valued map $T : P \to P$
  - Cost-function decreases for each $T \setminus \{\text{identity}\}$
  - The equilibria of T are the set of centroidal Voronoi partitions of G
  - Therefore, the system converges to a centroidal Voronoi partition in finite time

Computational complexity

- **Key computation:** distance from one vertex to all others in sub-graph of G
  - If edge weights are uniform, can use Breadth-First-Search approach in linear time
  - Otherwise, Dijkstra's algorithm requires log linear time
- Computing centroid of sub-graph p_i is most complex aspect, three options:
  - Exhaustive search: $O\left(|p_i|^2\right)$
  - Gradient Descent: $O\left(|p_i|\log\left(|p_i|\right)\right)$
  - Linear-time approximation: $O\left(|p_i|\right)$

A more complex simulation



Ten agents in a non-convex environment with holes

Conclusions

- Distributed partitioning of a graph using gossip communication
  - Graph can represent complex non-convex environment
  - Each robot's sub-graph is always connected
- Convergence to a centroidal Voronoi partition in finite time
- Computational complexity can scale well

Future work

- Motion protocol so robots seek out their neighbors
- Agent arrival, departure, and failure
- Method to avoid local minima in cost function

Thank you

Questions?