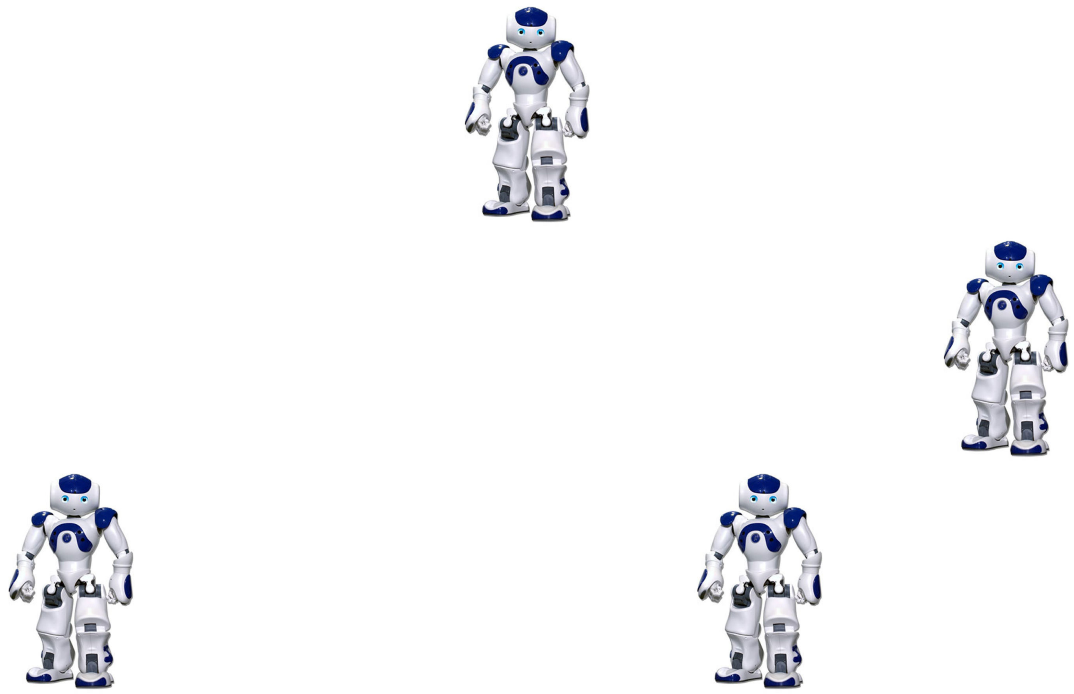
Optimal Exclusive Perpetual Grid Exploration by Luminous Myopic Opaque Robots with Common Chirality

Quentin Bramas, Stéphane Devismes, and Pascal Lafourcade

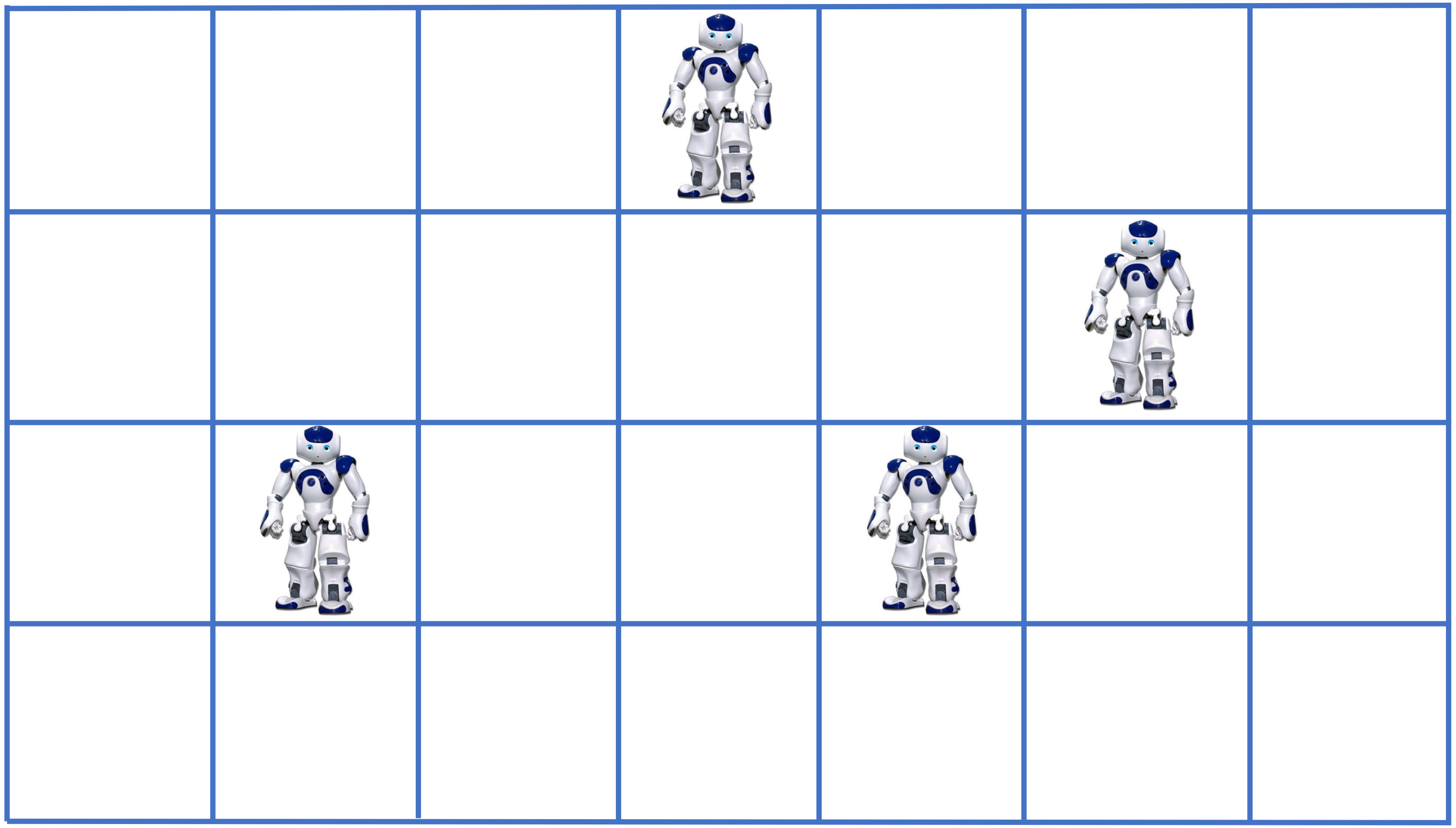
Question(s)

- How a very few very weak robots can collaborate to solve an infinite task?
- How robot skills influence the task resolution?
- **Case Study:** Exclusive Perpetual Exploration of a grid by a swarm of luminous myopic opaque robots with a common chirality

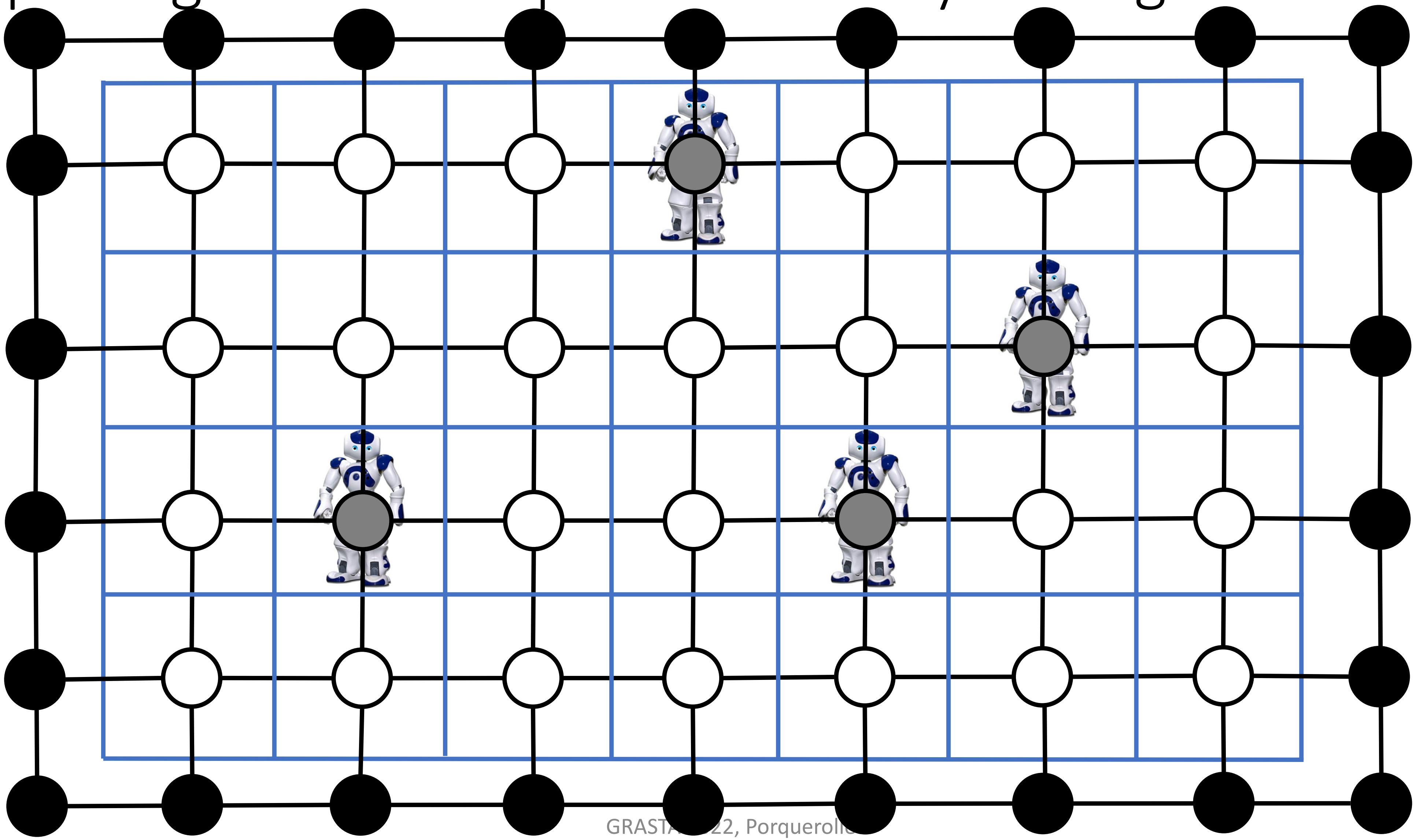
A finite number of robots



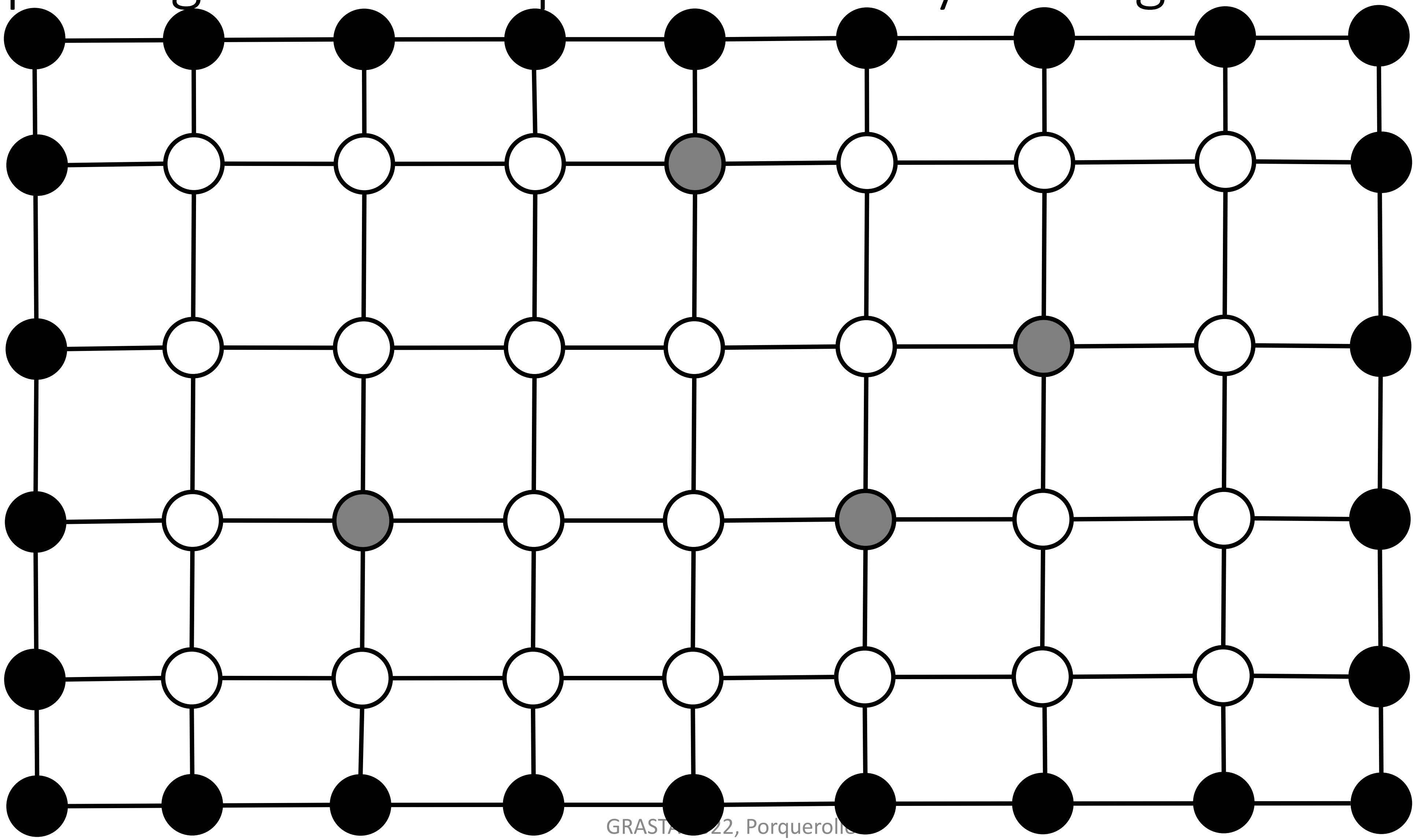
Exploring a rectangular area



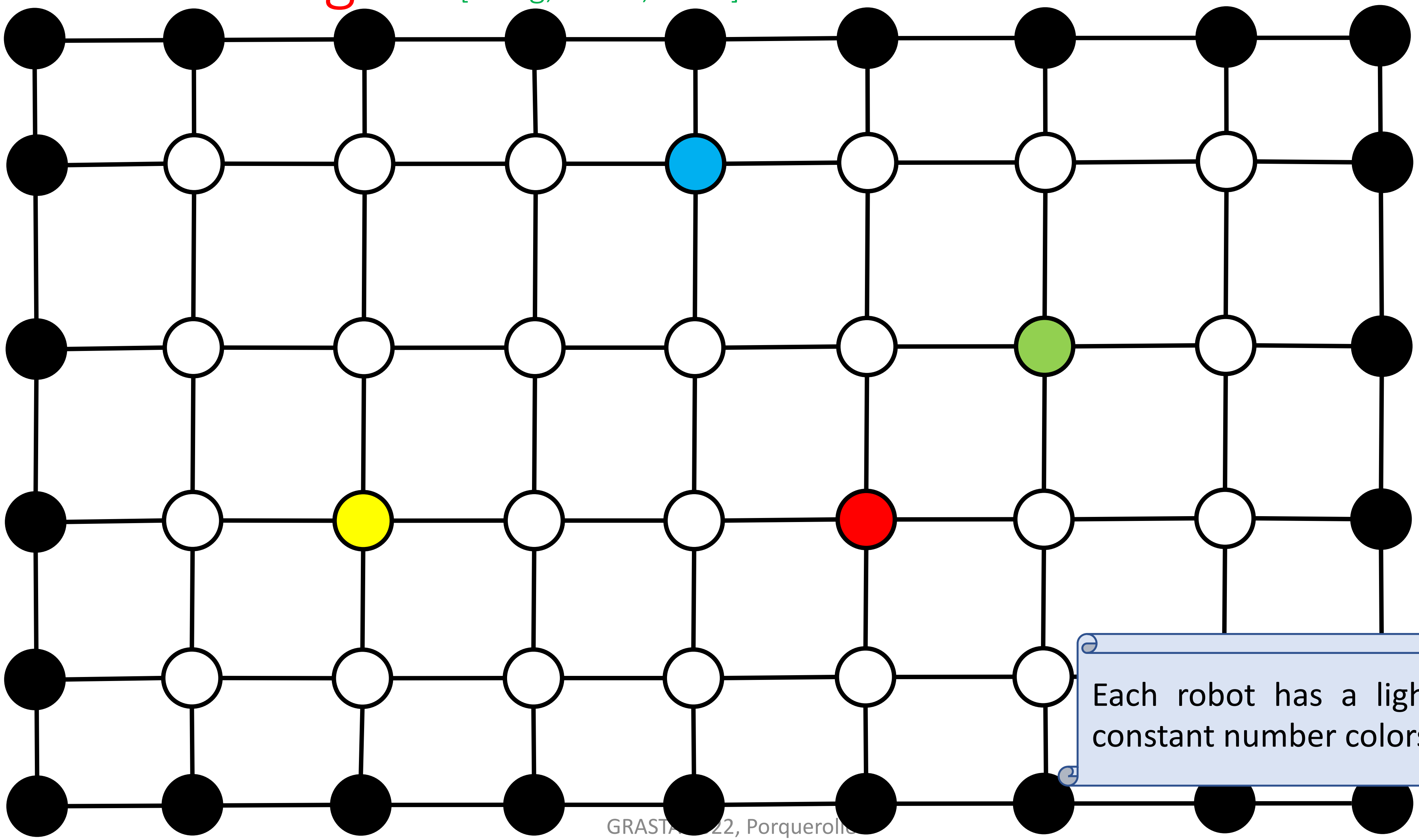
Exploring a discrete space: an anonymous grid



Exploring a discrete space: an anonymous grid

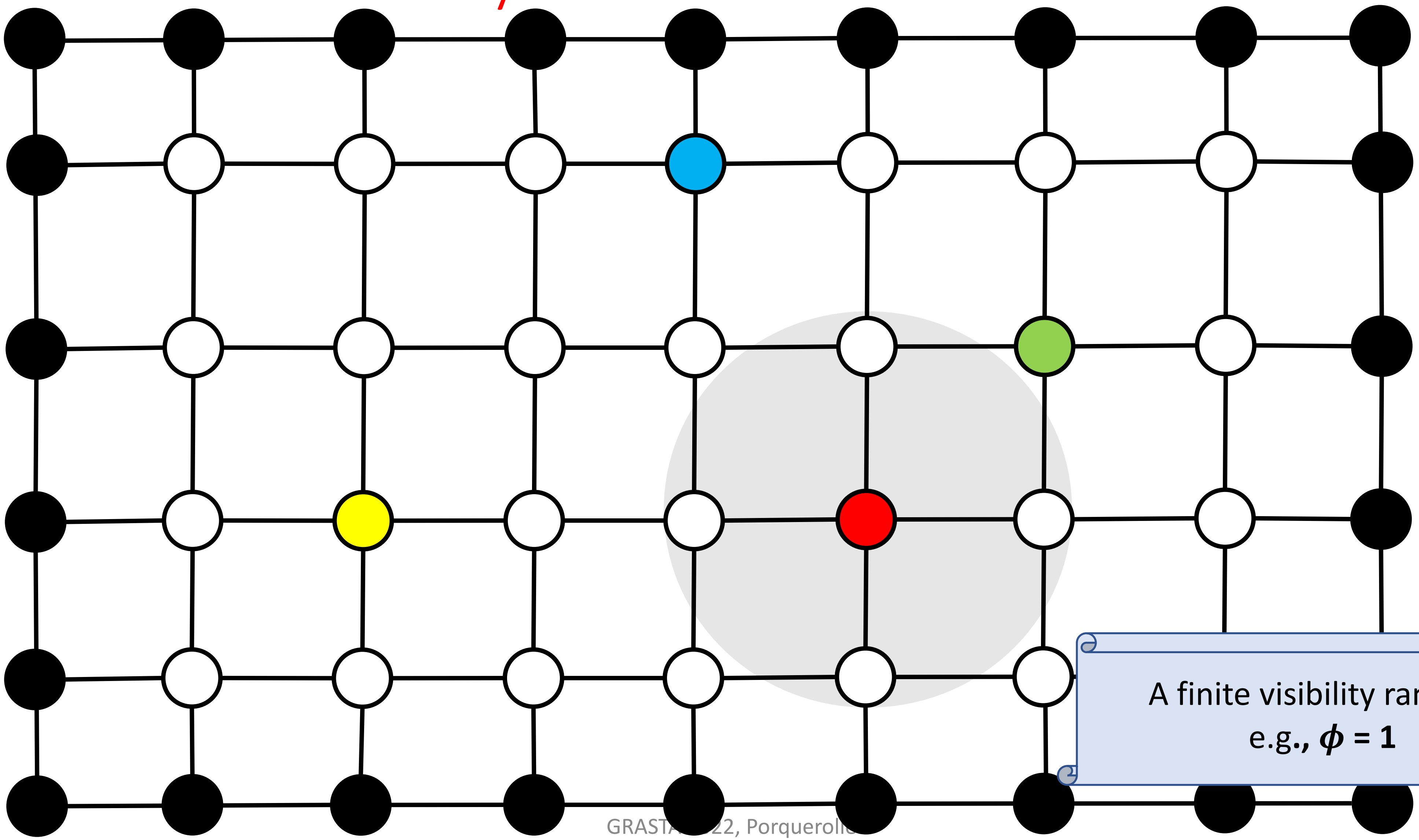


Robot skills: lights [Peleg, *IWDC*, 2005]



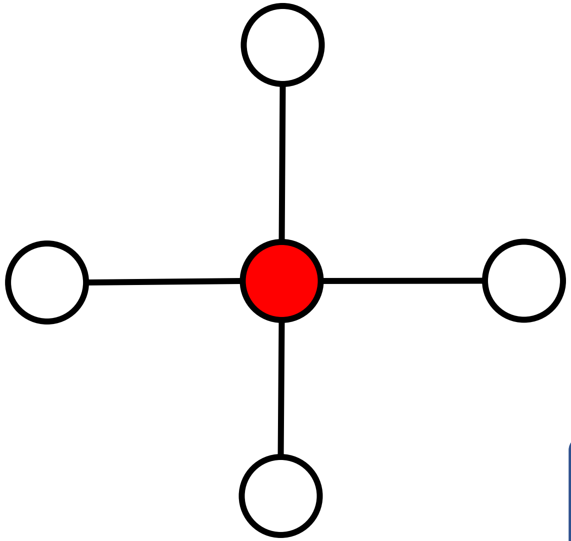
Each robot has a light with a constant number colors

Robot skills: visibility



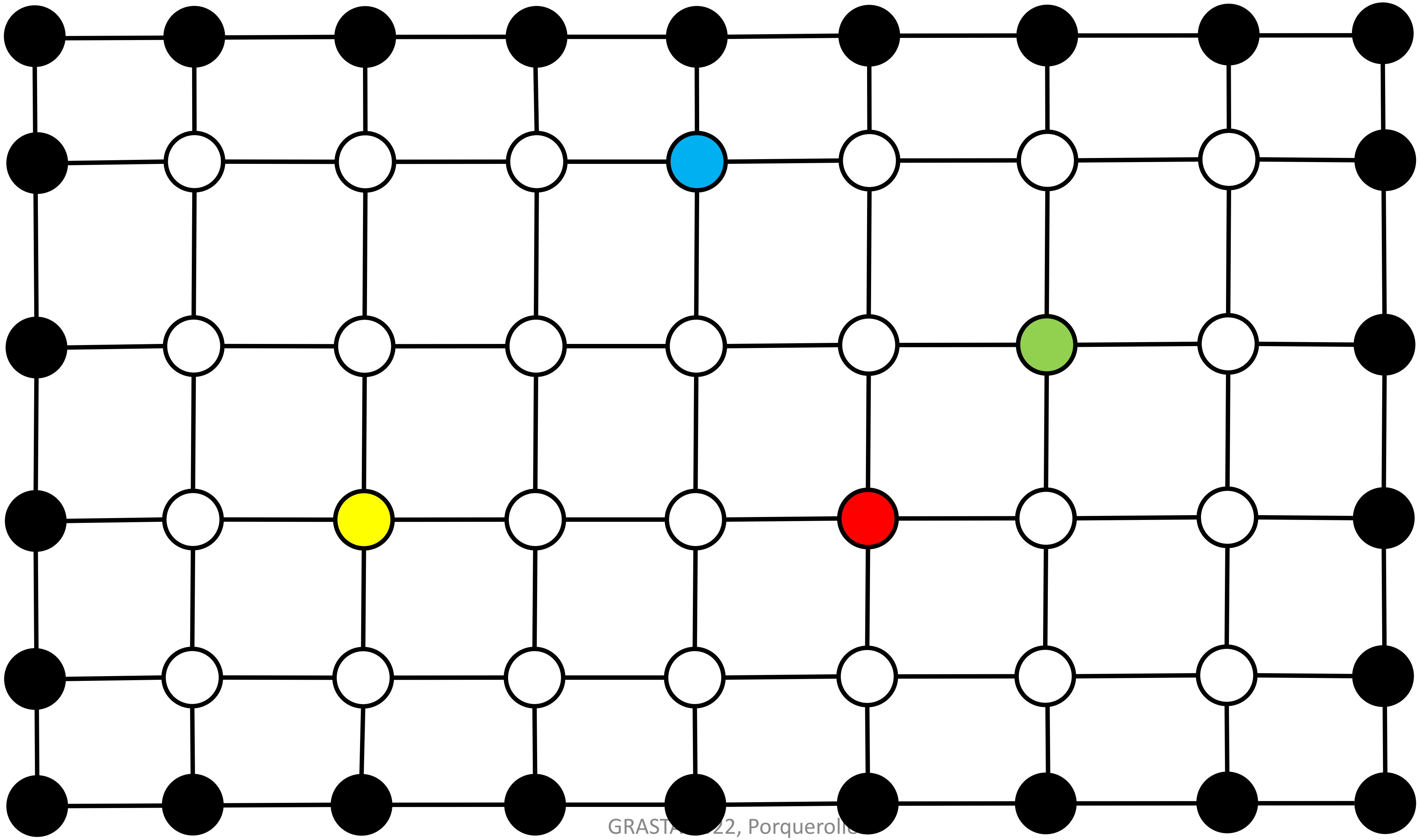
A finite visibility range ϕ
e.g., $\boldsymbol{\phi}$ **= 1**

Robot skills: visibility

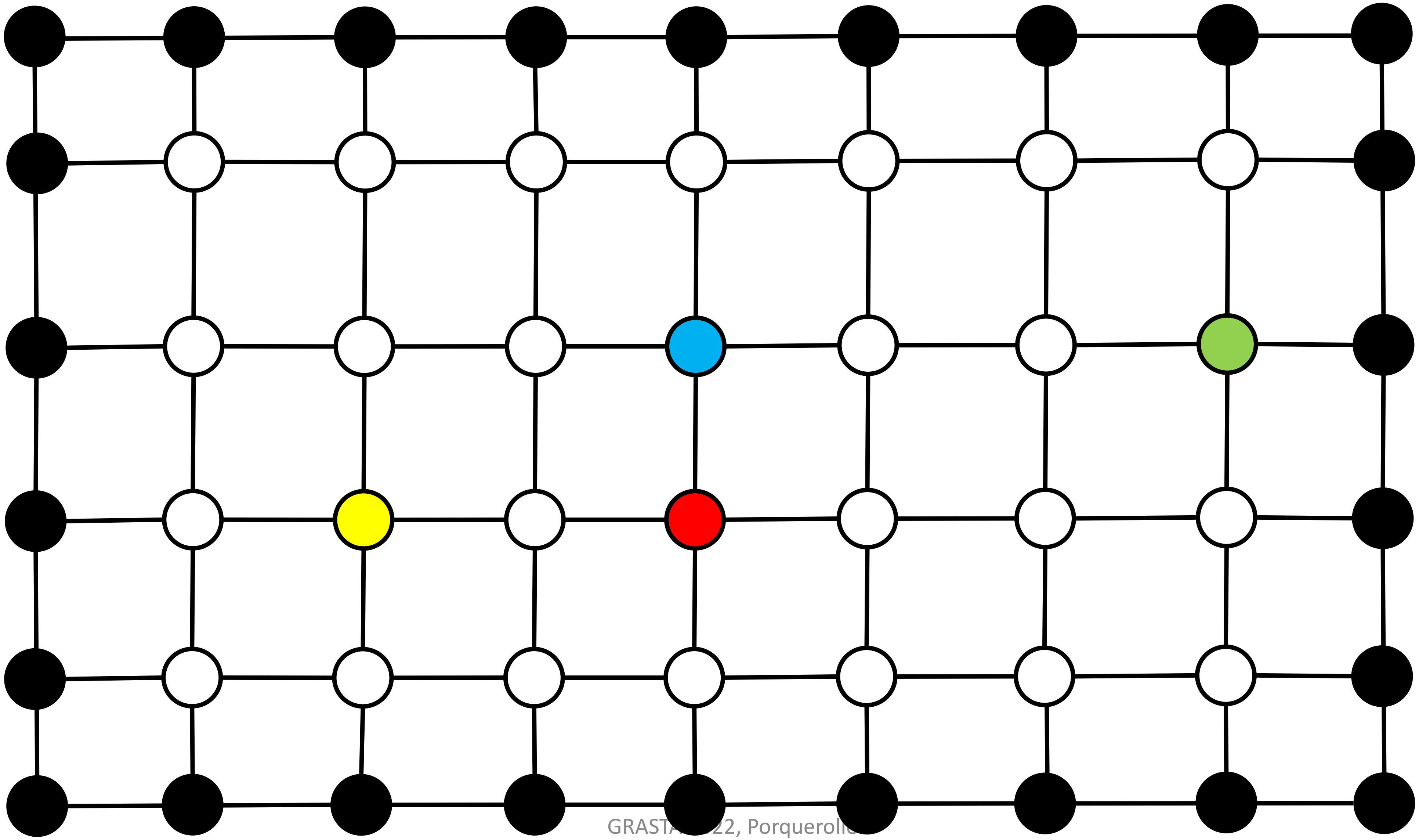


A finite visibility range ϕ
e.g., $\boldsymbol{\phi}$ **= 1**

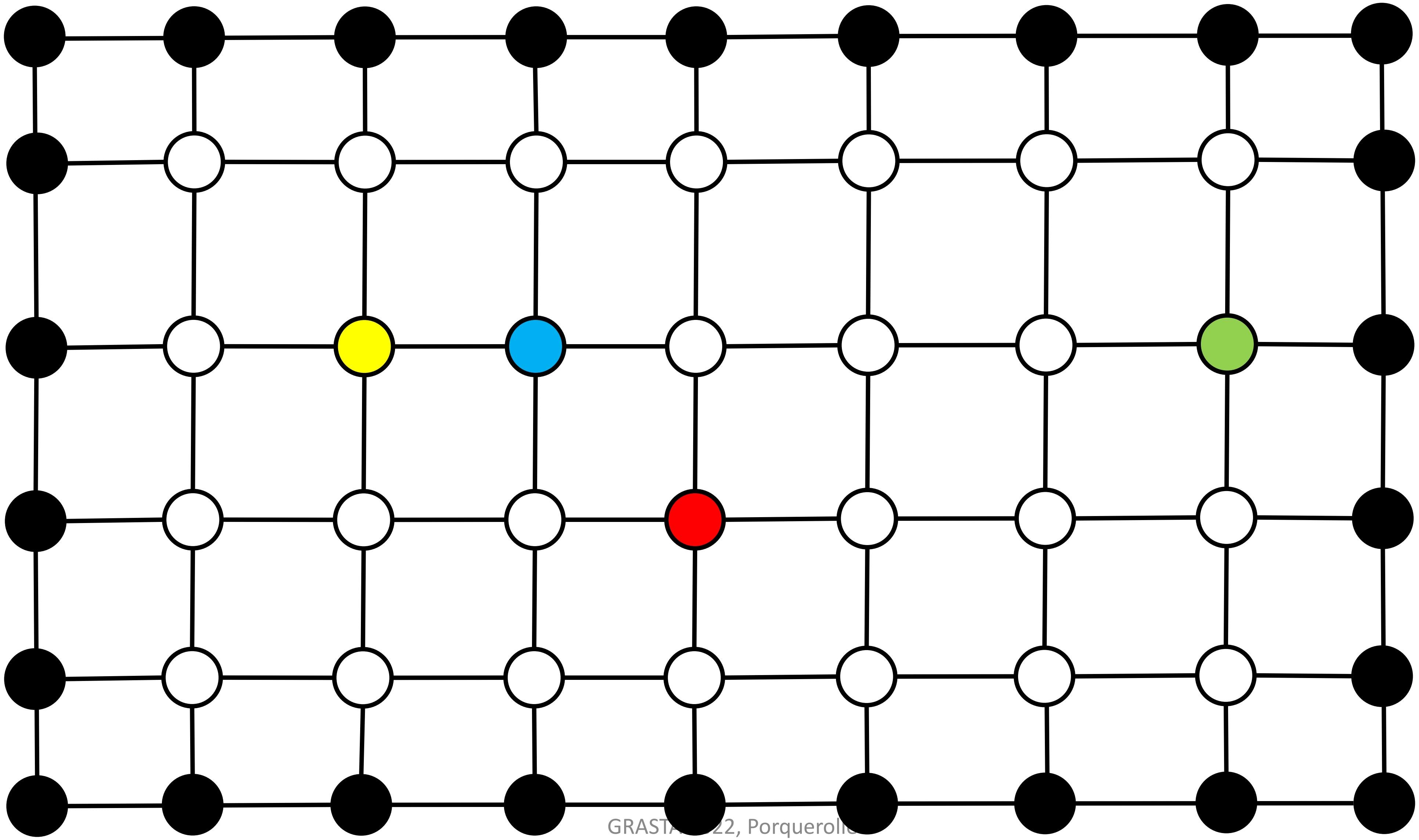
Robot skills: mobile



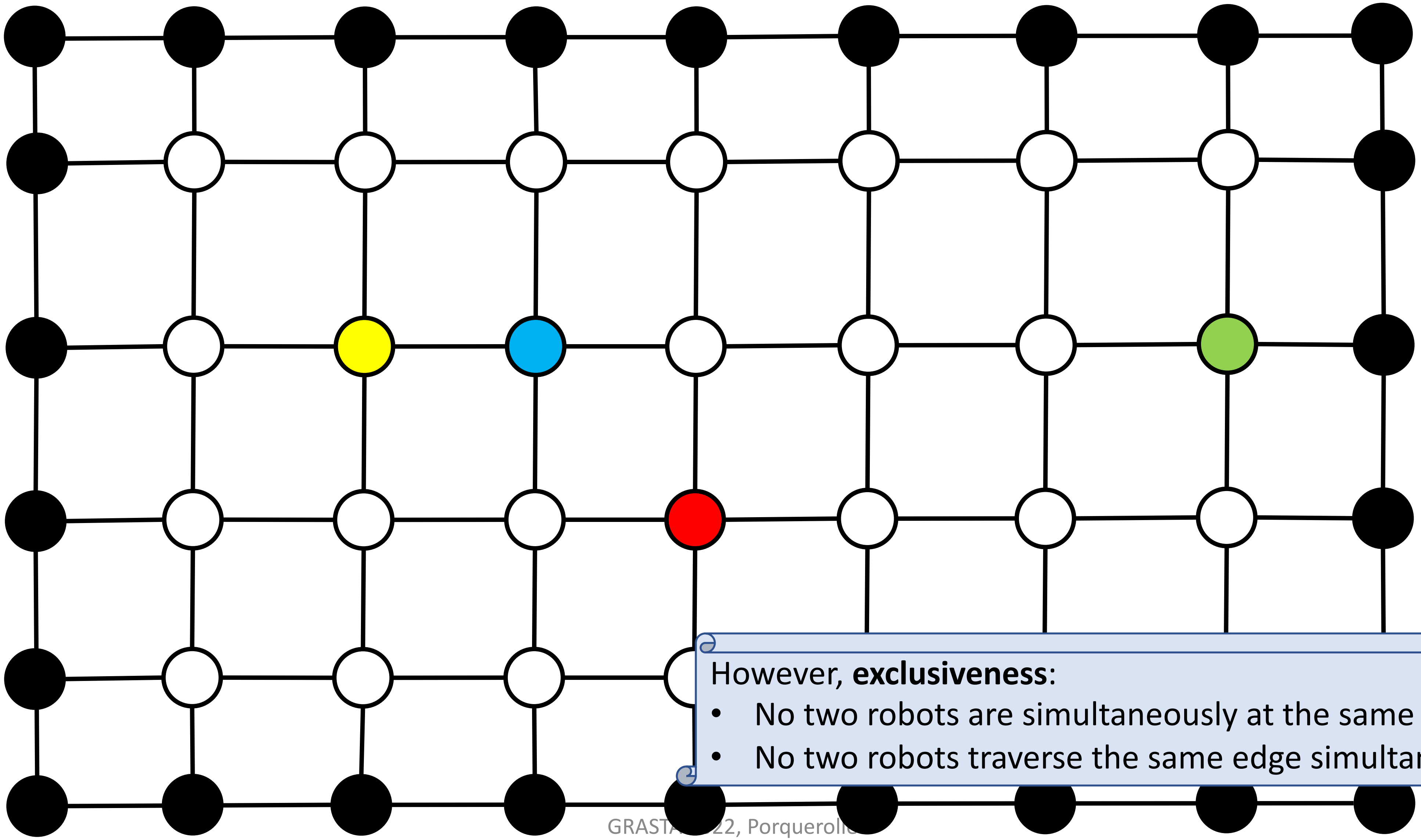
Robot skills: mobile



Robot skills: mobile

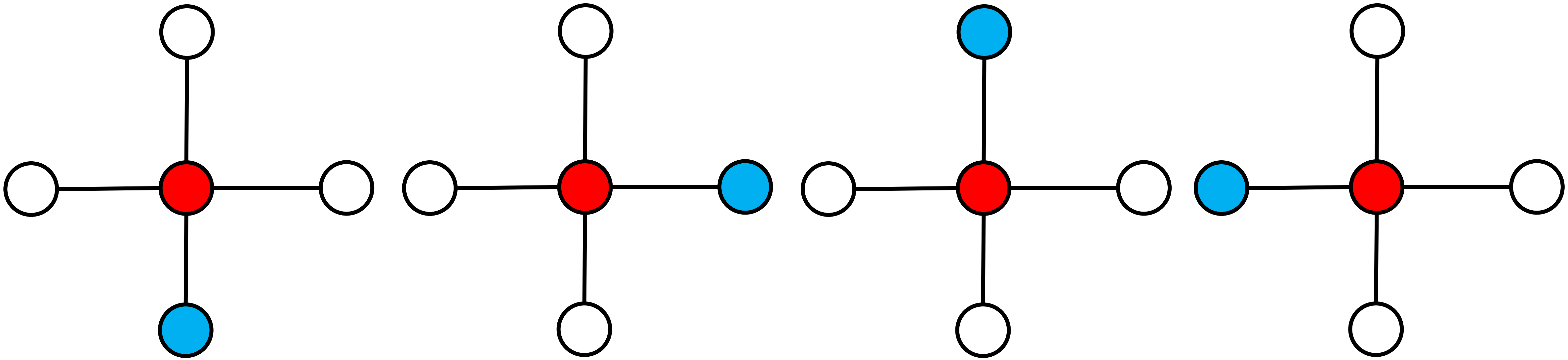


Robot skills: mobile

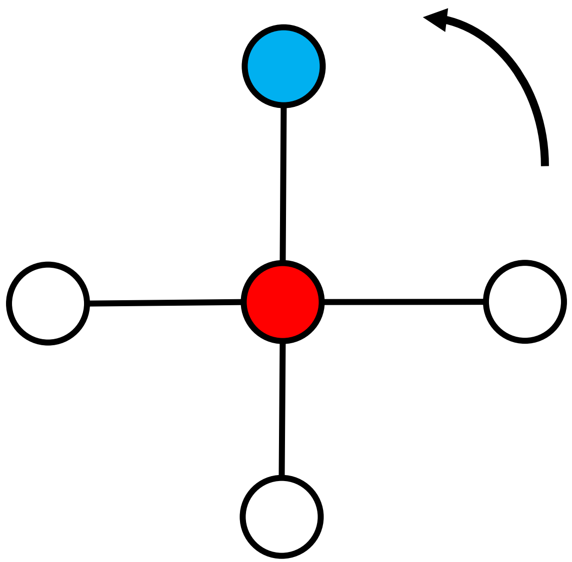


However, **exclusiveness**:
- No two robots are simultaneously at the same place
- No two robots traverse the same edge simultaneously

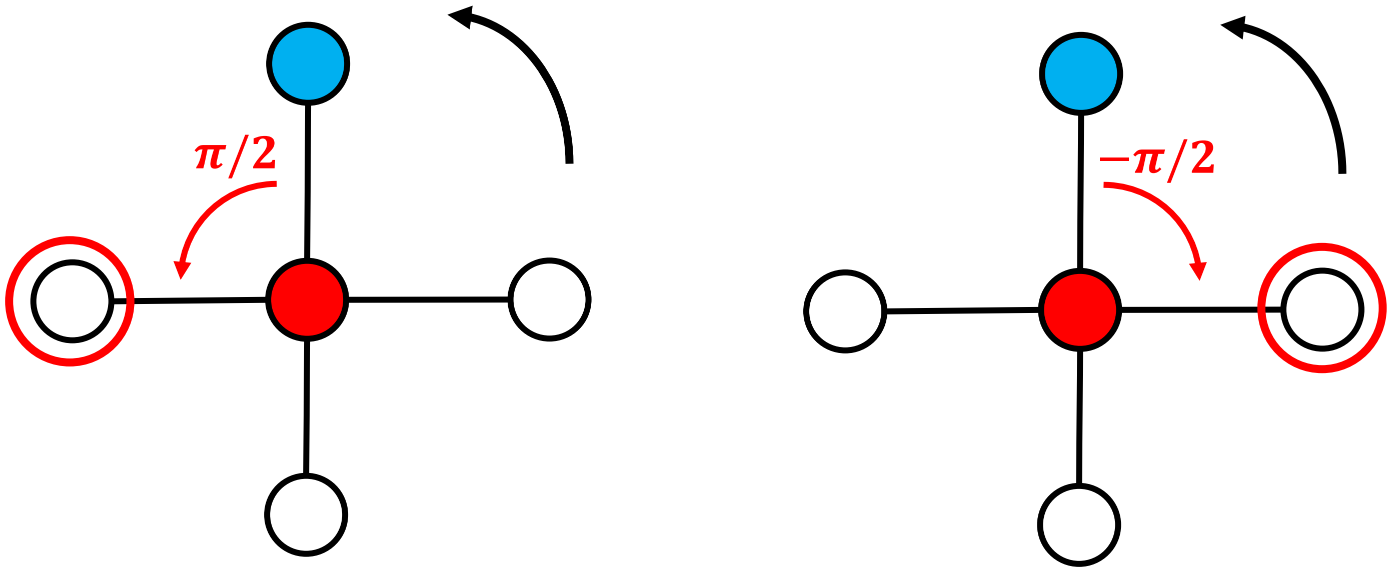
Robot skills: no compass



Robot skills: a common chirality

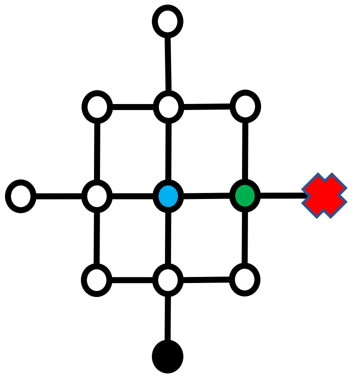


Robot skills: a common chirality



Other weaknesses

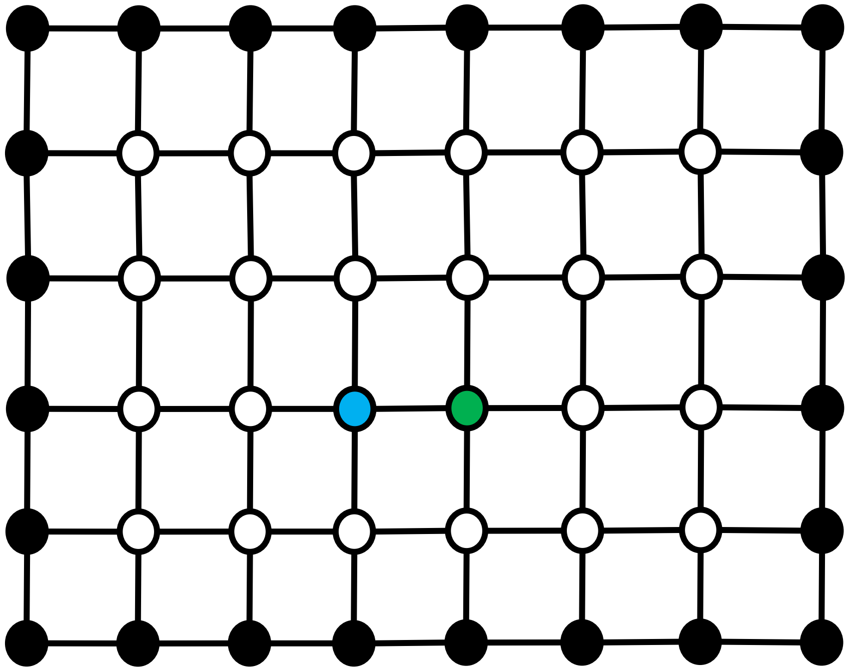
- Opaque



- No additional persistent memory (one color = oblivious)
- No direct communication mean (deaf mute)
- No whiteboard on nodes
- No pebble …

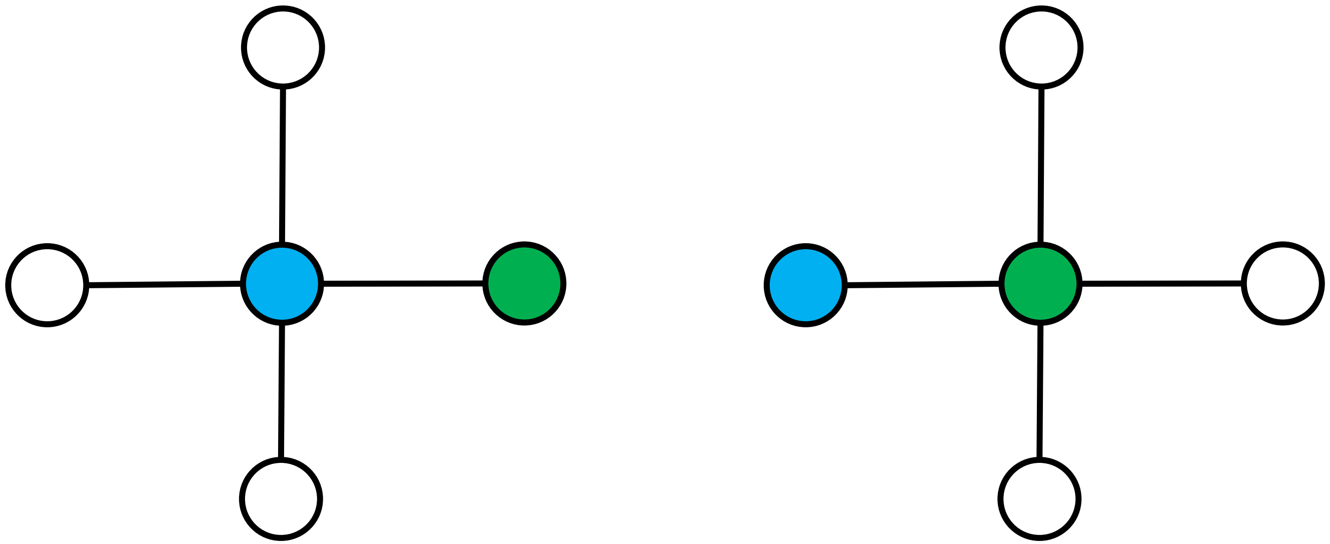
Execution Model [Suzuki & Yamashita, *SIAM Jrnl of Comp.*, 1999]

- Look-Compute-Move Cycles



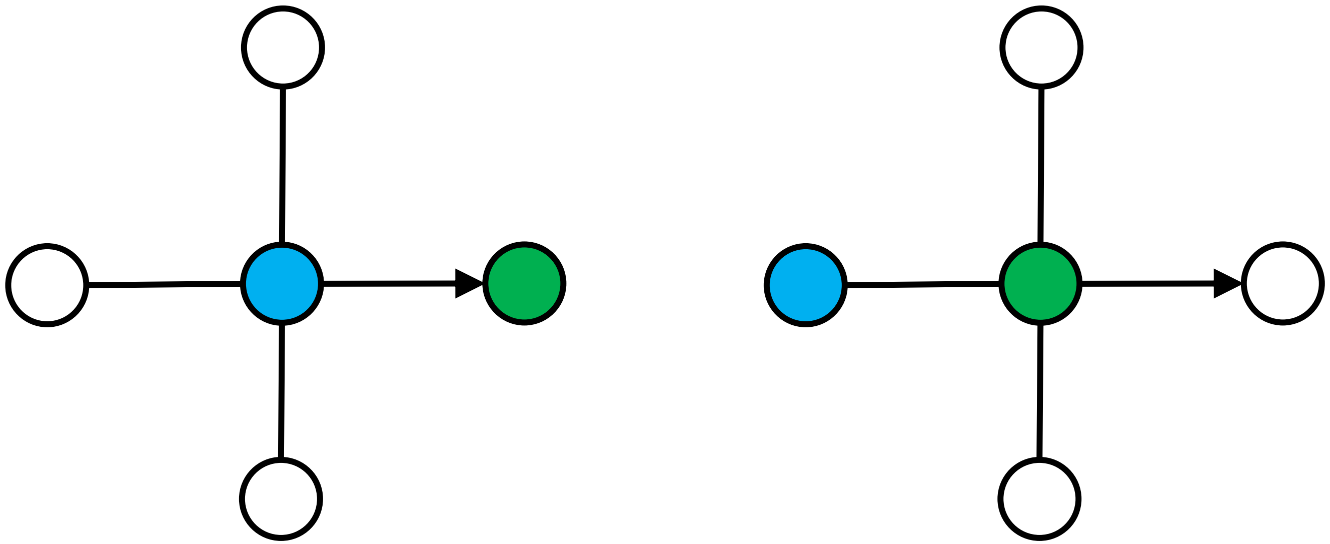
Execution Model [Suzuki & Yamashita, *SIAM Jrnl of Comp.*, 1999]

- Look-Compute-Move Cycles (here at distance 1)



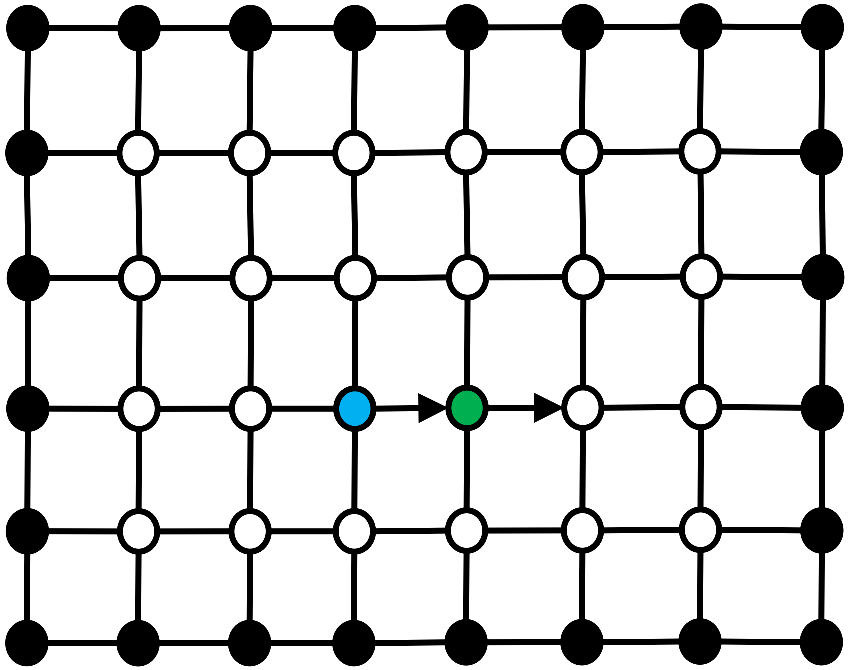
Execution Model [Suzuki & Yamashita, *SIAM Jrnl of Comp.*, 1999]

- Look-Compute-Move Cycles (destination + color)



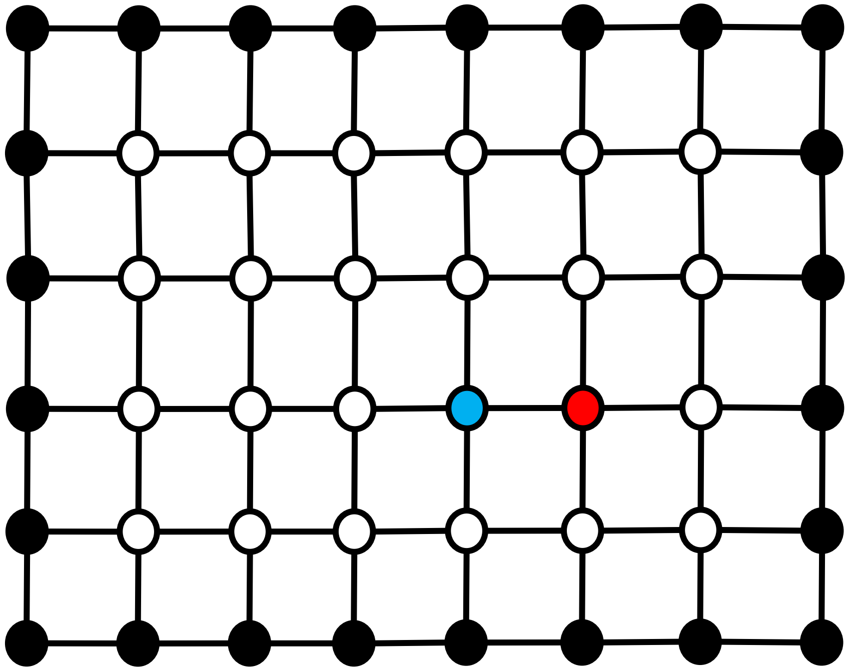
Execution Model [Suzuki & Yamashita, *SIAM Jrnl of Comp.*, 1999]

- Look-Compute-Move Cycles



Execution Model [Suzuki & Yamashita, *SIAM Jrnl of Comp.*, 1999]

- Look-Compute-Move Cycles



Synchrony Refinements

- Asynchronous

L	**C**		**M**		**L**		**C**	**M**	**L**		**C**
		L		C	M	L	C	M		L	

- Semi-Synchronous

L	**C**	**M**				**L**	**C**	**M**	**L**	**C**	**M**
L	C	M	L	C	M	L	C	M			

- Synchronous

L	**C**	**M**	**L**	**C**	**M**	**L**	**C**	**M**	**L**	**C**	**M**
L	C	M	L	C	M	L	C	M	L	C	M

Synchrony Refinements

- Asynchronous

L	**C**		**M**		**L**		**C**	**M**	**L**		**C**
		L		C	M	L	C	M		L	

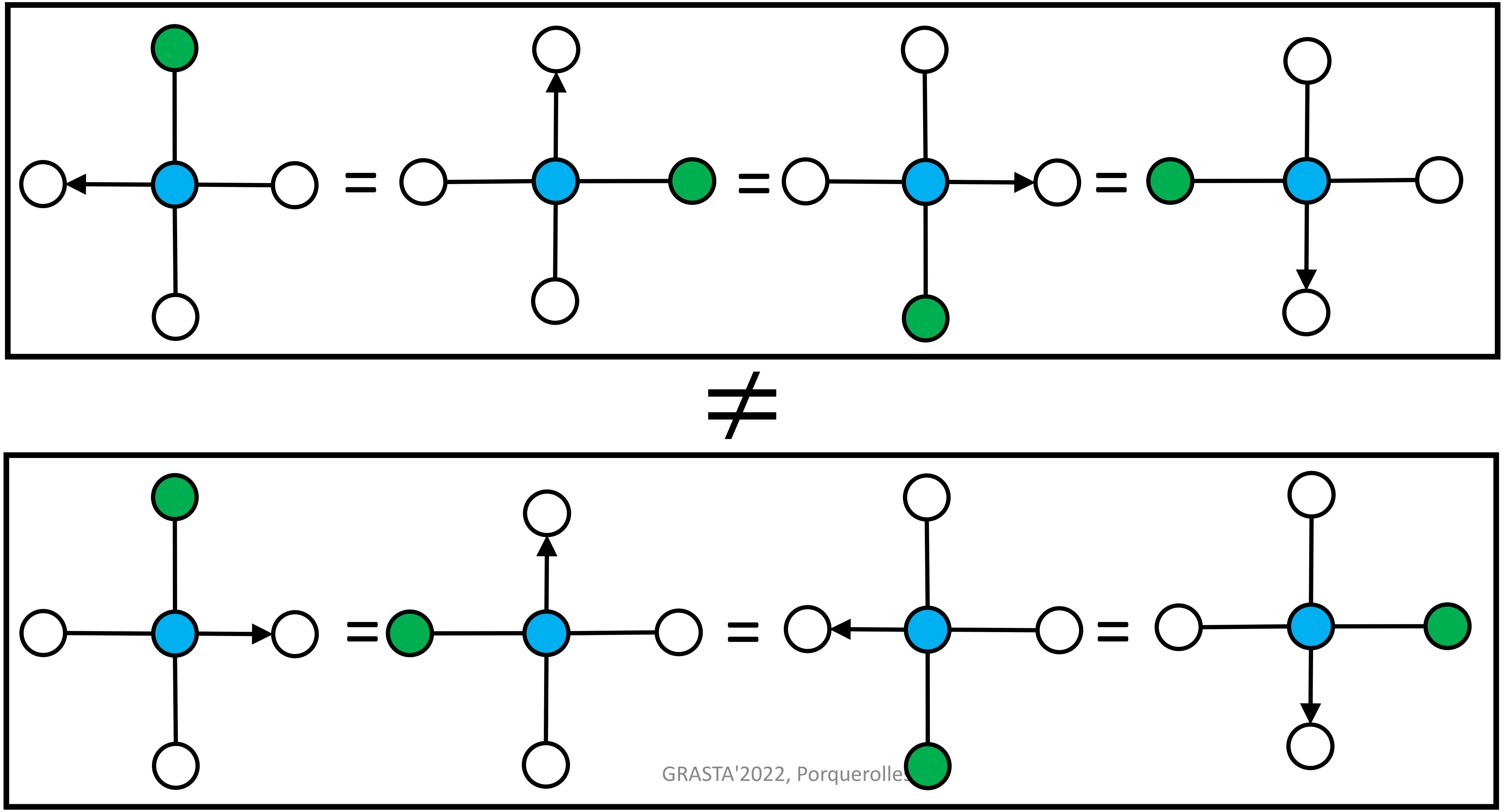
- Semi-Synchronous

L	**C**	**M**				**L**	**C**	**M**	**L**	**C**	**M**
L	C	M	L	C	M	L	C	M			

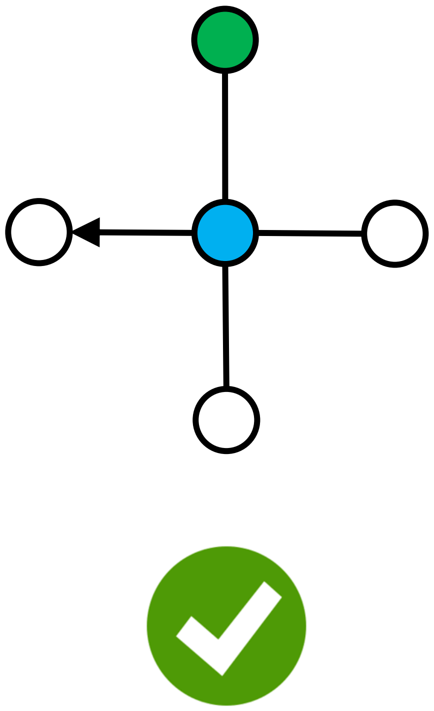
- Synchronous

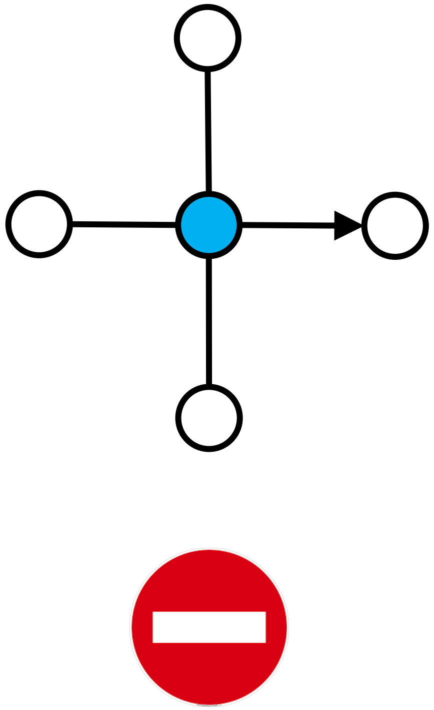
L	**C**	**M**	**L**	**C**	**M**	**L**	**C**	**M**	**L**	**C**	**M**
L	C	M	L	C	M	L	C	M	L	C	M

Rules and Chirality



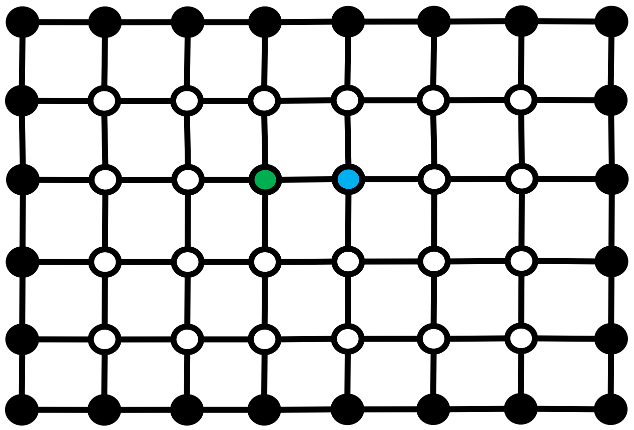
Deterministic Rules and Chirality

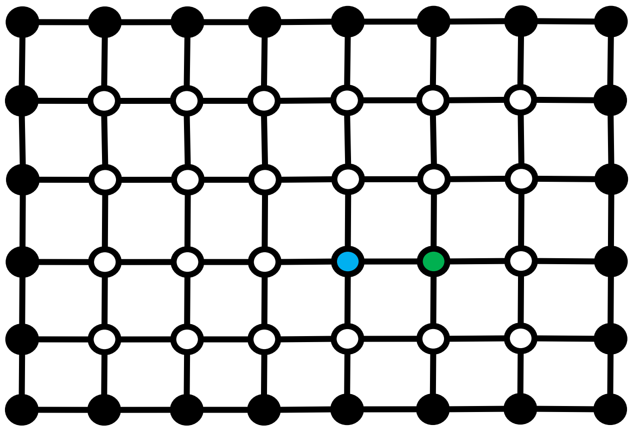


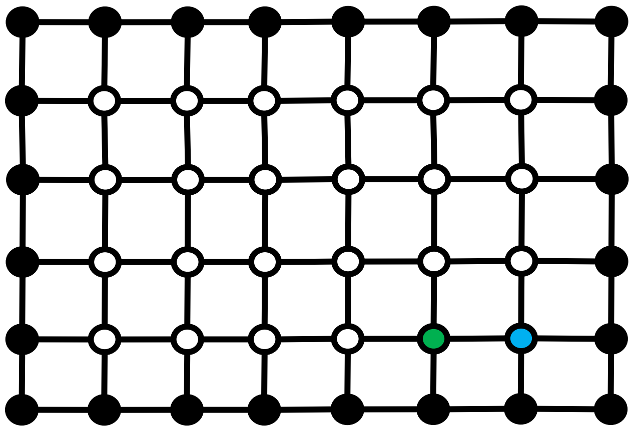


Algorithms

- Algorithm = initial configurations + rules
- In our algorithms:
  - rules are deterministic
  - locally-defined initial configurations:
    - A pattern, *i.e.*, colors + relative positions of the robots, *e.g.*, 
    - The pattern is **arbitrarily** placed in the grid



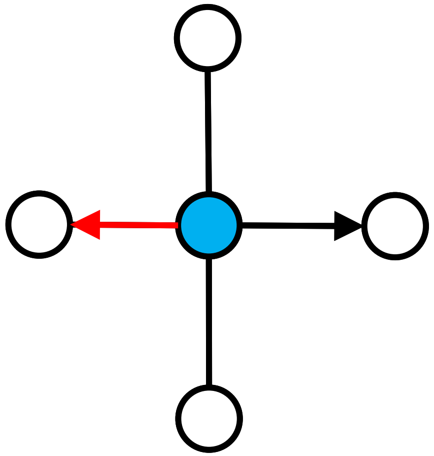




Lower Bounds

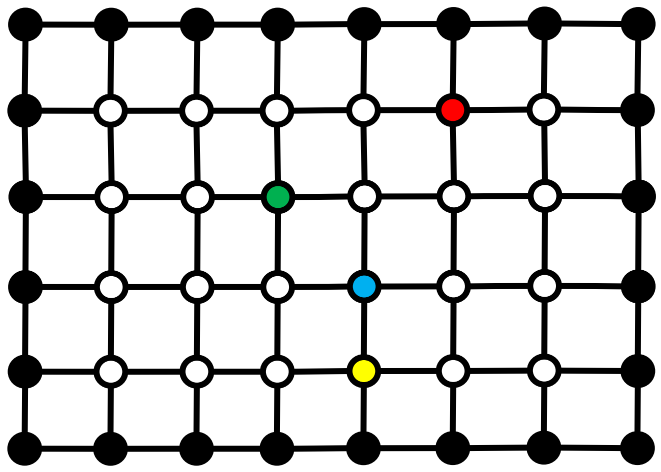
- Any restriction on initial configurations
- Nondeterministic rules are allowed
  - Nondeterminism solved by an adversary
  - But it may help !

    [Bramas *et al.*, *FUN'22*]



Perpetual Grid Exploration Problem

- With n robots
- For every grid G=(V,E) of size $X \times Y$, with $X \geq n$ and $Y \geq n$, each node of G should be visited <u>infinitely often</u> by a robot.



Overview of the results

Visibility range	# Robots	# Colors	Algorithm
Finite	1	Finite	Impossible
Finite	2	1	Impossible
1	2	2	Impossible
1	3	1	Impossible
1	2	3	Yes
2	2	2	Yes
2	3	1	Yes

Overview of the results

Visibility range	# Robots	# Colors	Algorithm
Finite	1	Finite	Impossible
Finite	2	1	Impossible
1	2	2	Impossible
1	3	1	Impossible
1	**2**	**3**	**Yes**
2	2	2	Yes
2	3	1	Yes

Overview of the results

Visibility range	# Robots	# Colors	Algorithm
Finite	**1**	**Finite**	**Impossible**
Finite	2	1	Impossible
1	2	2	Impossible
1	3	1	Impossible
1	**2**	**3**	**Yes**
2	2	2	Yes
2	3	1	Yes

Overview of the results

Visibility range	# Robots	# Colors	Algorithm
Finite	**1**	**Finite**	**Impossible**
Finite	**2**	**1**	**Impossible**
1	**2**	**2**	**Impossible**
1	3	1	Impossible
1	**2**	**3**	**Yes**
2	2	2	Yes
2	3	1	Yes

Overview of the results

Visibility range	# Robots	# Colors	Algorithm
Finite	**1**	**Finite**	**Impossible**
Finite	**2**	**1**	**Impossible**
1	**2**	**2**	**Impossible**
1	3	1	Impossible
1	**2**	**3**	**Yes**
2	**2**	**2**	**Yes**
2	3	1	Yes

Overview of the results

Visibility range	# Robots	# Colors	Algorithm
Finite	1	Finite	Impossible
Finite	2	1	Impossible
1	2	2	Impossible
1	3	1	Impossible
1	2	3	Yes
2	2	2	Yes
2	3	1	Yes

Overview of the results

Visibility range	# Robots	# Colors	Algorithm
Finite	1	Finite	Impossible
Finite	2	1	Impossible
1	2	2	Impossible
1	3	1	Impossible
1	2	3	Yes
2	2	2	Yes
2	3	1	Yes

First Solution: range 1, 2 robots, 3 colors

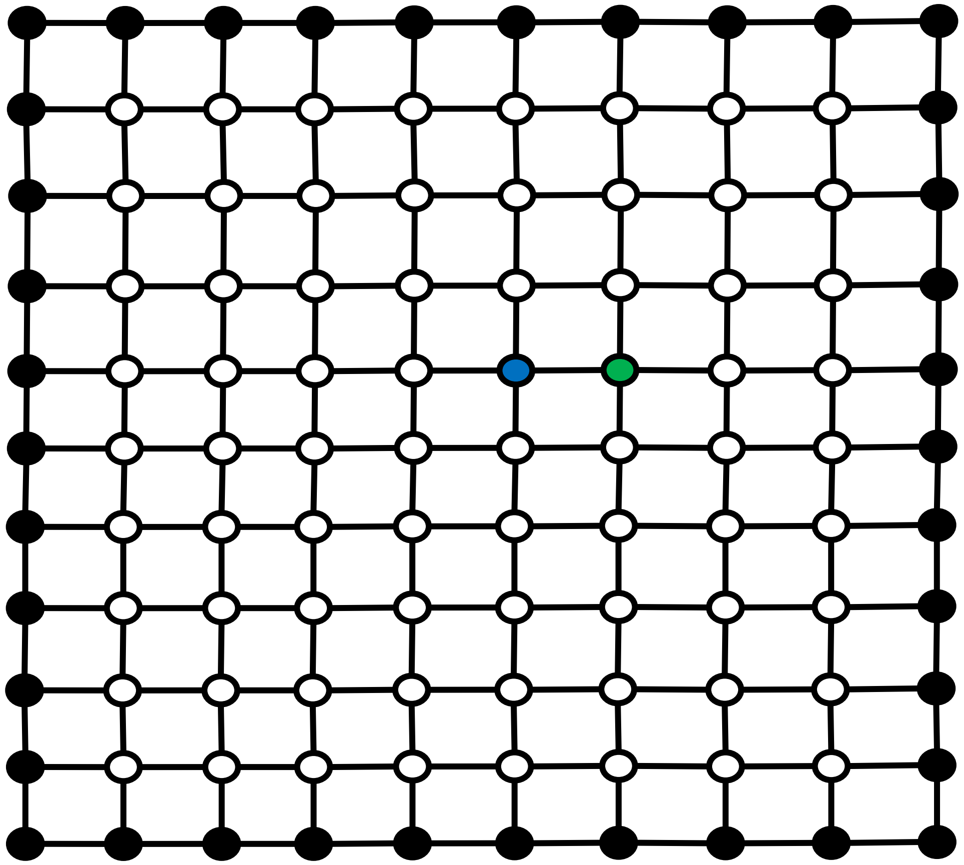
- **Principle**
  - Moving group: a leader + a follower

    - The follower follows the leader 😉
      - It is used to give an orientation to the group
      - Color: 🔵

    - The leader takes the decision on the behalf of the group
      - 2 colors: 🔴 (left) , 🟢 (right)

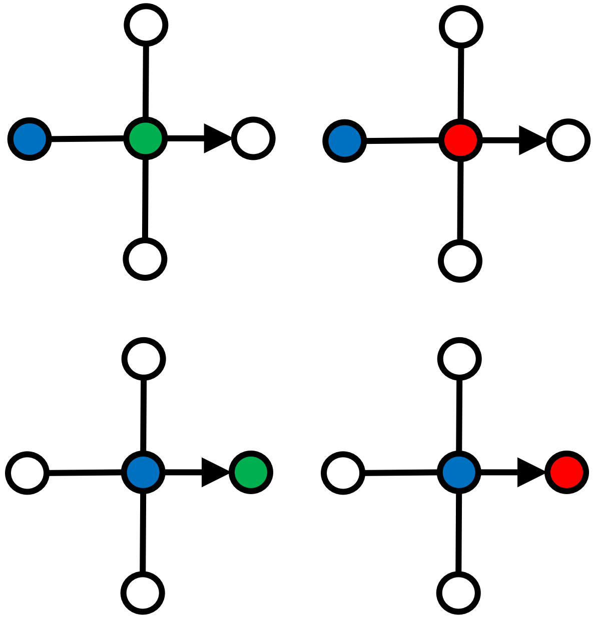
First Solution: range 1, 2 robots, 3 colors

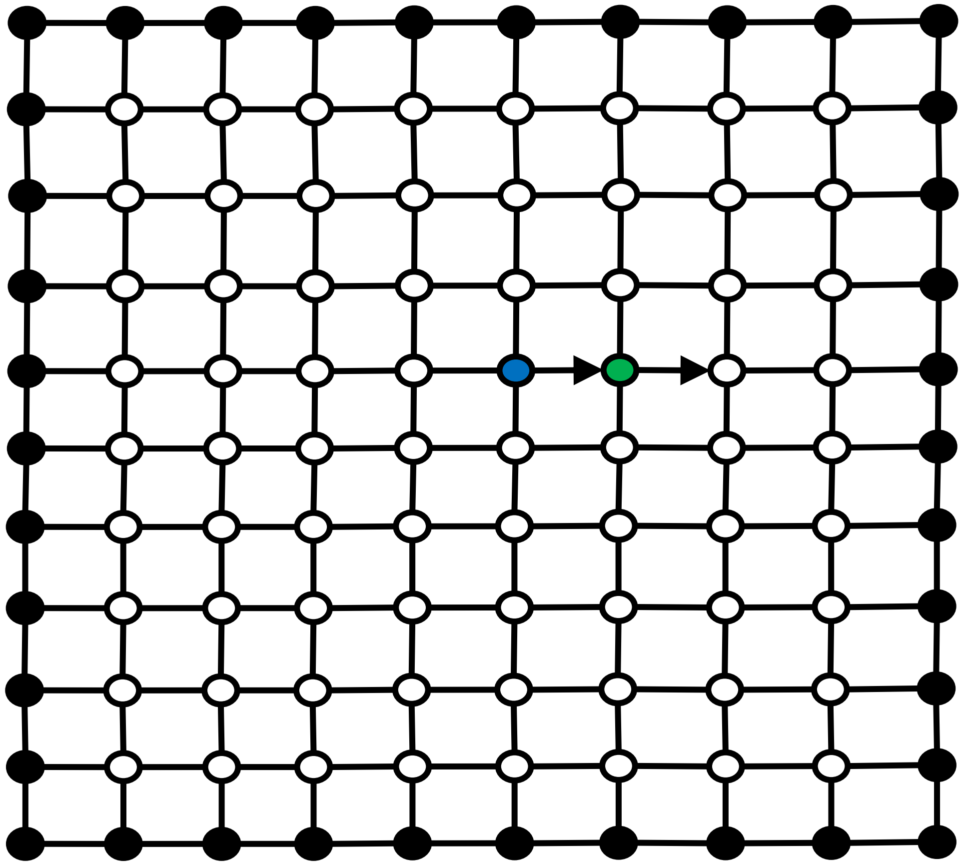
- Initial Configurations: 
  - Leader:
  - Follower



First Solution: range 1, 2 robots, 3 colors

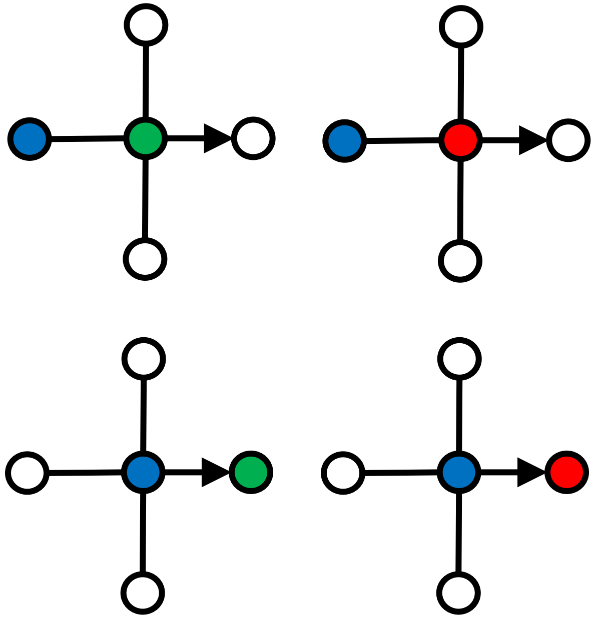
- Move toward a wall

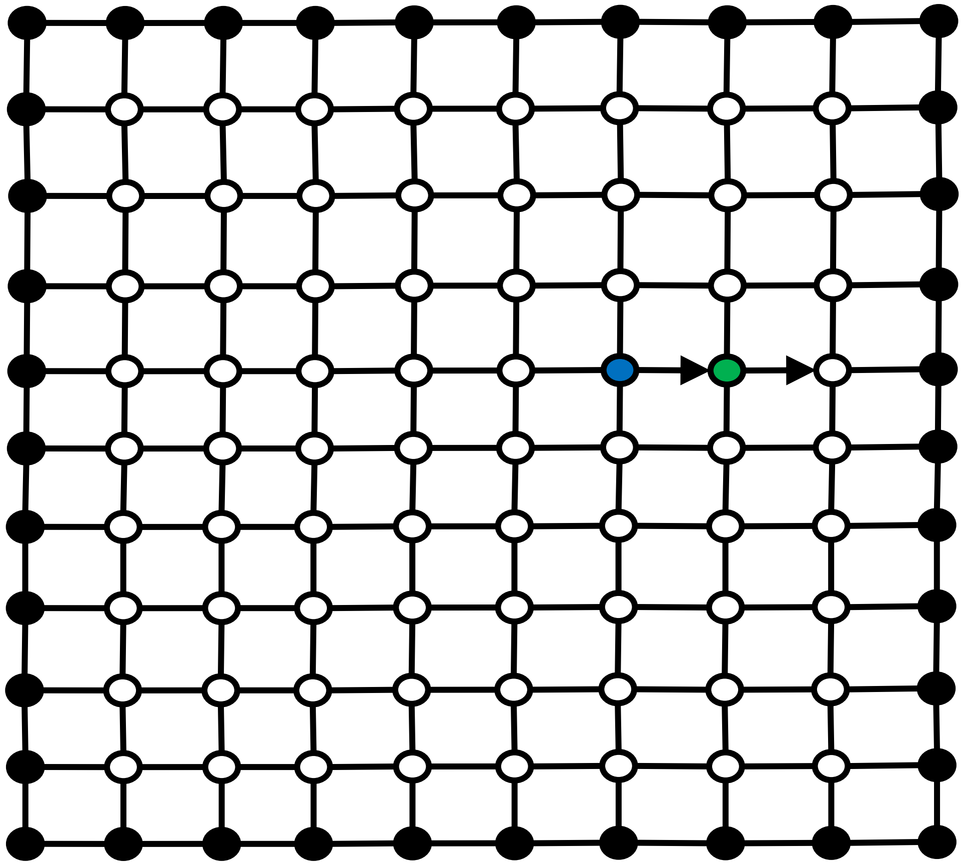




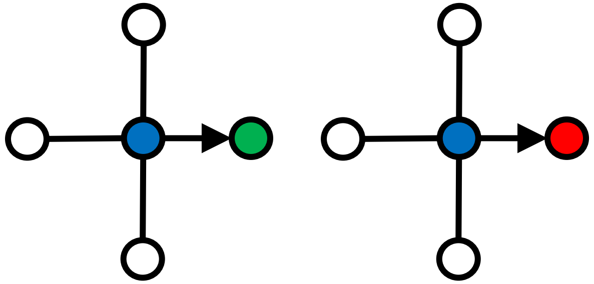
First Solution: range 1, 2 robots, 3 colors

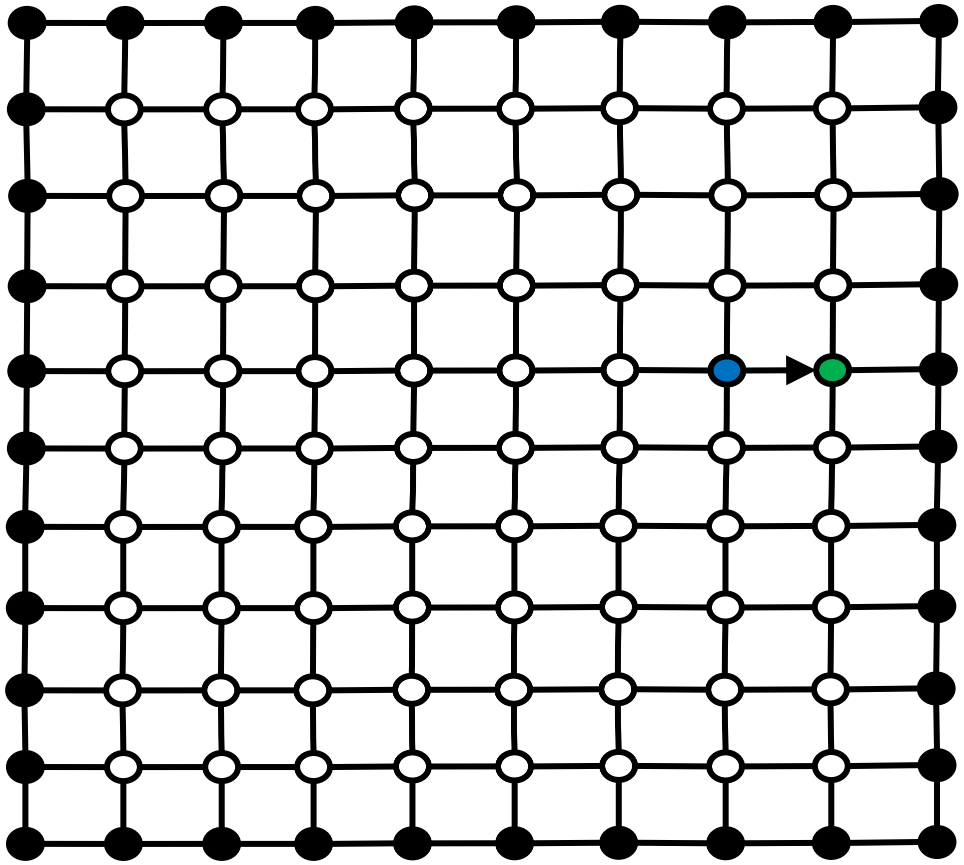
- Move toward a wall





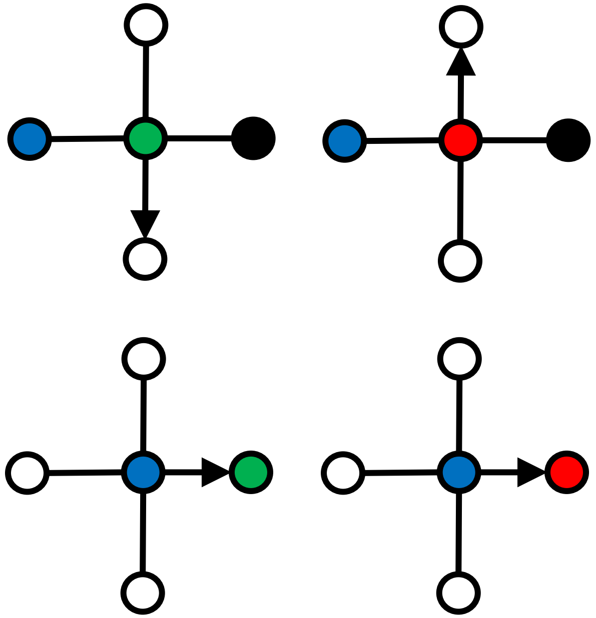
First Solution: range 1, 2 robots, 3 colors

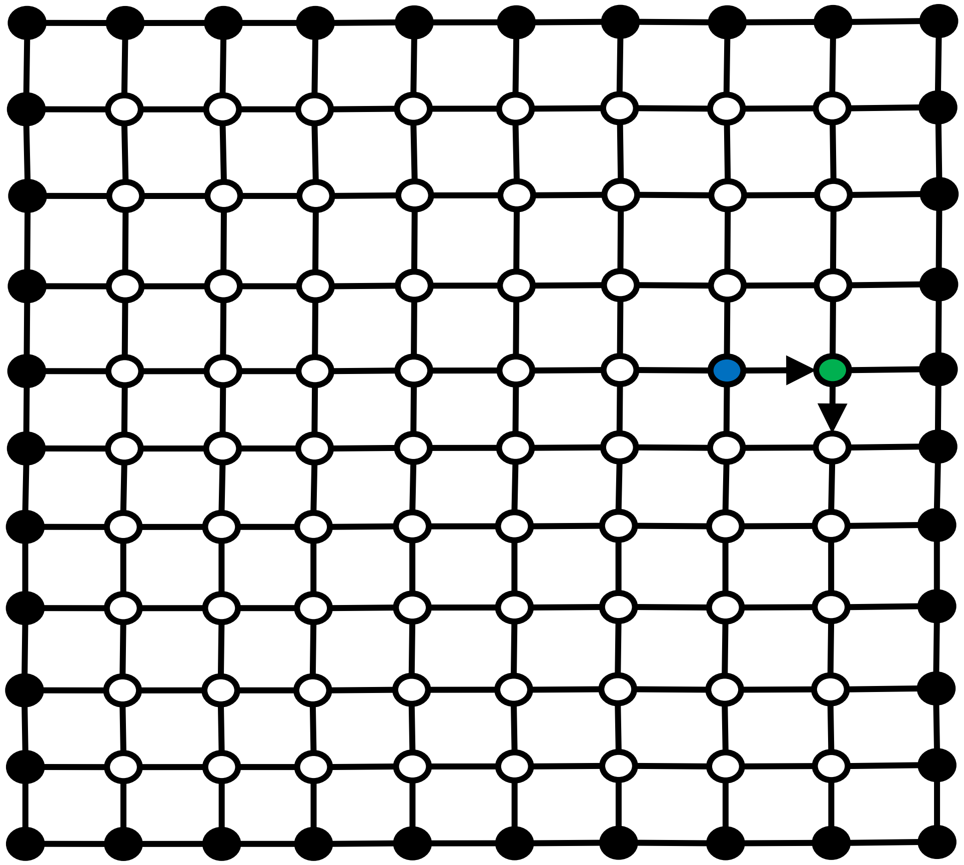




First Solution: range 1, 2 robots, 3 colors

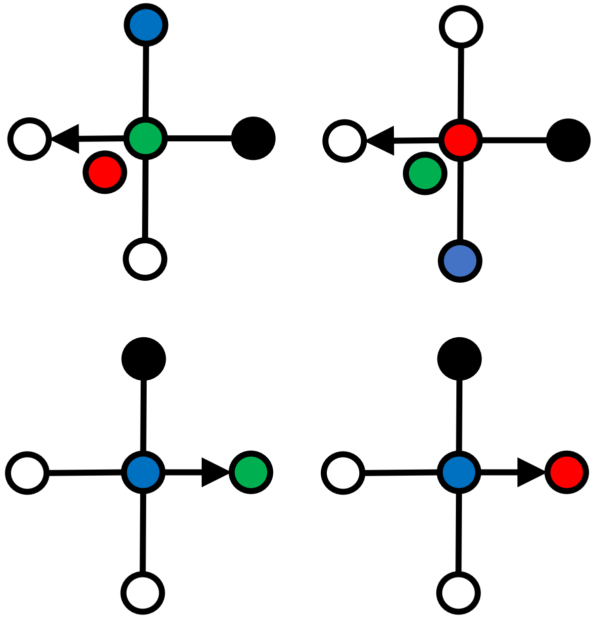
- Turn

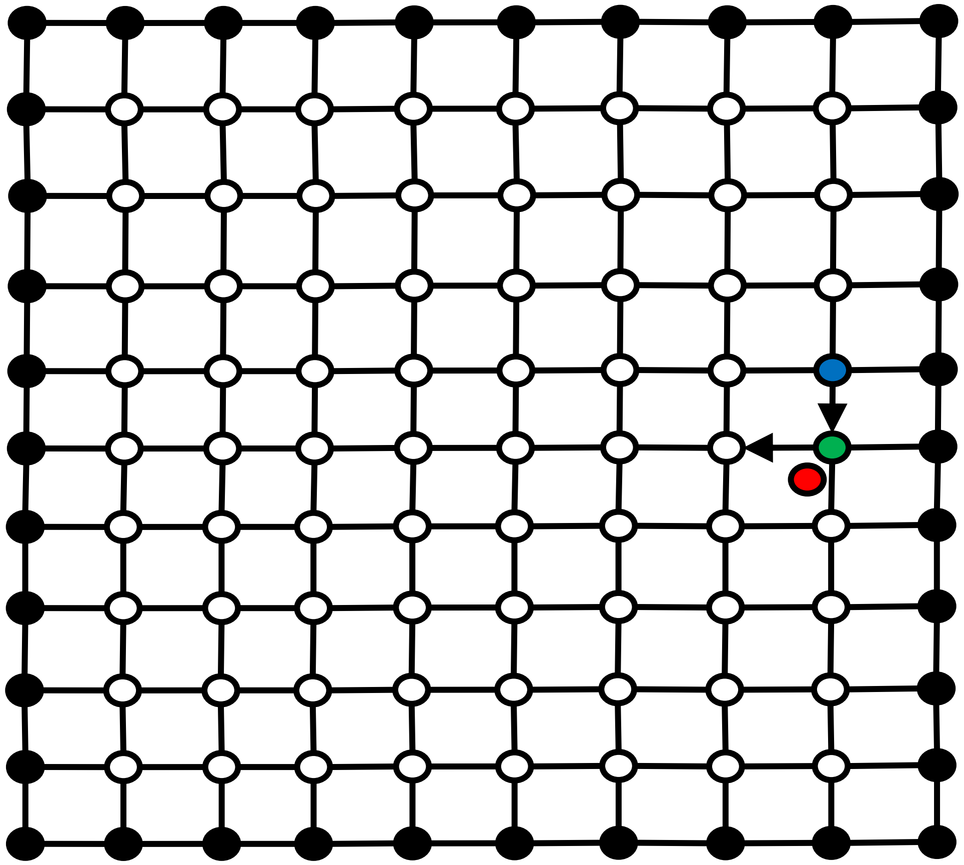




First Solution: range 1, 2 robots, 3 colors

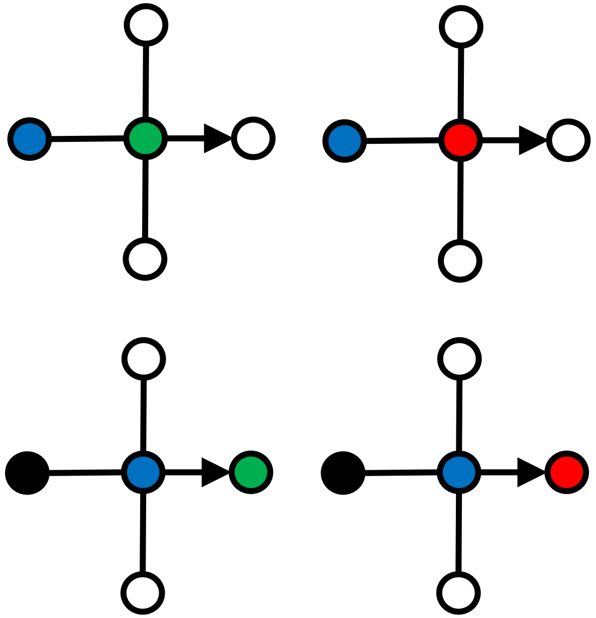
- Turn

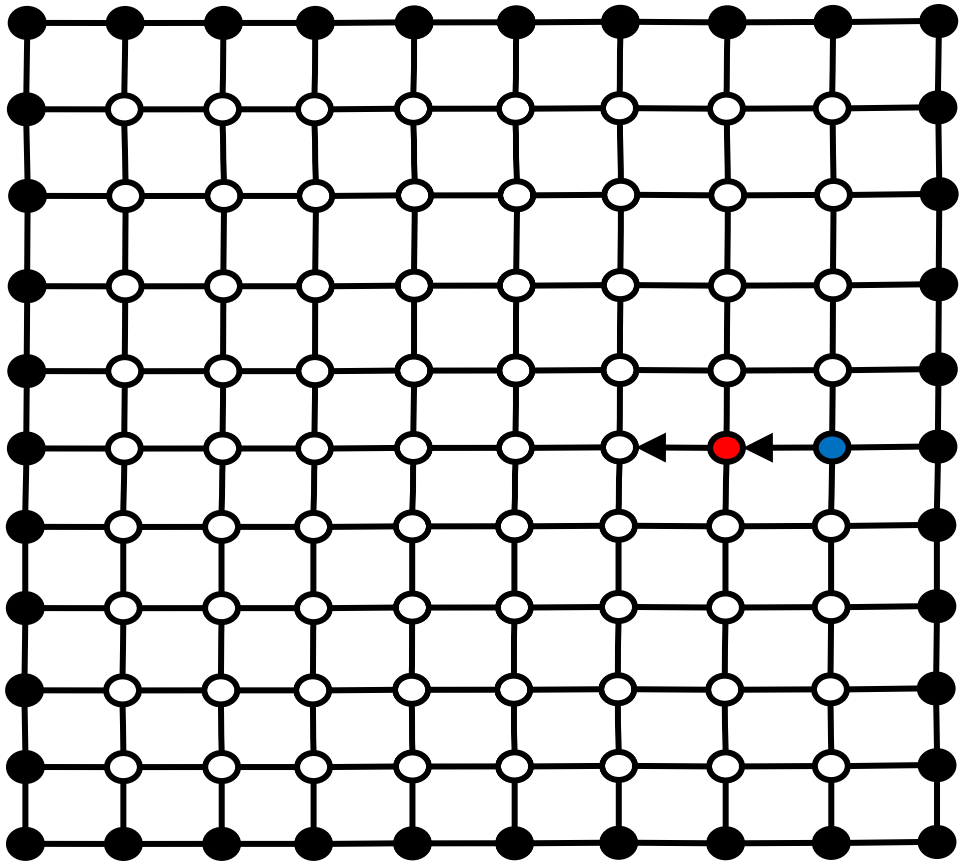




First Solution: range 1, 2 robots, 3 colors

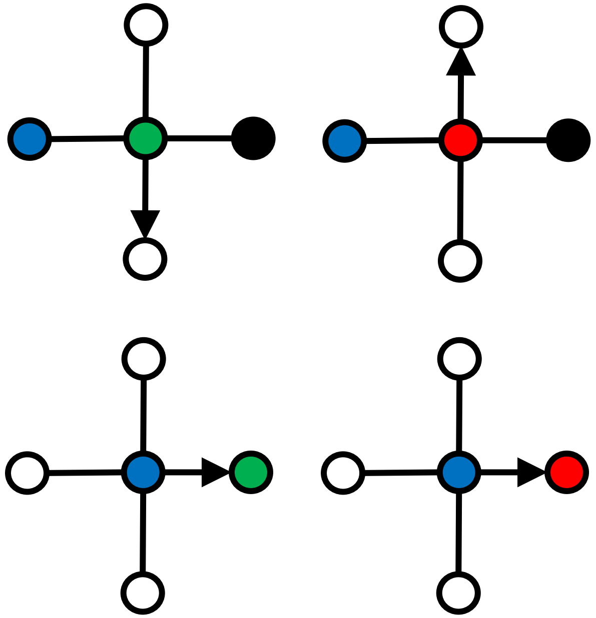
- Move toward a wall

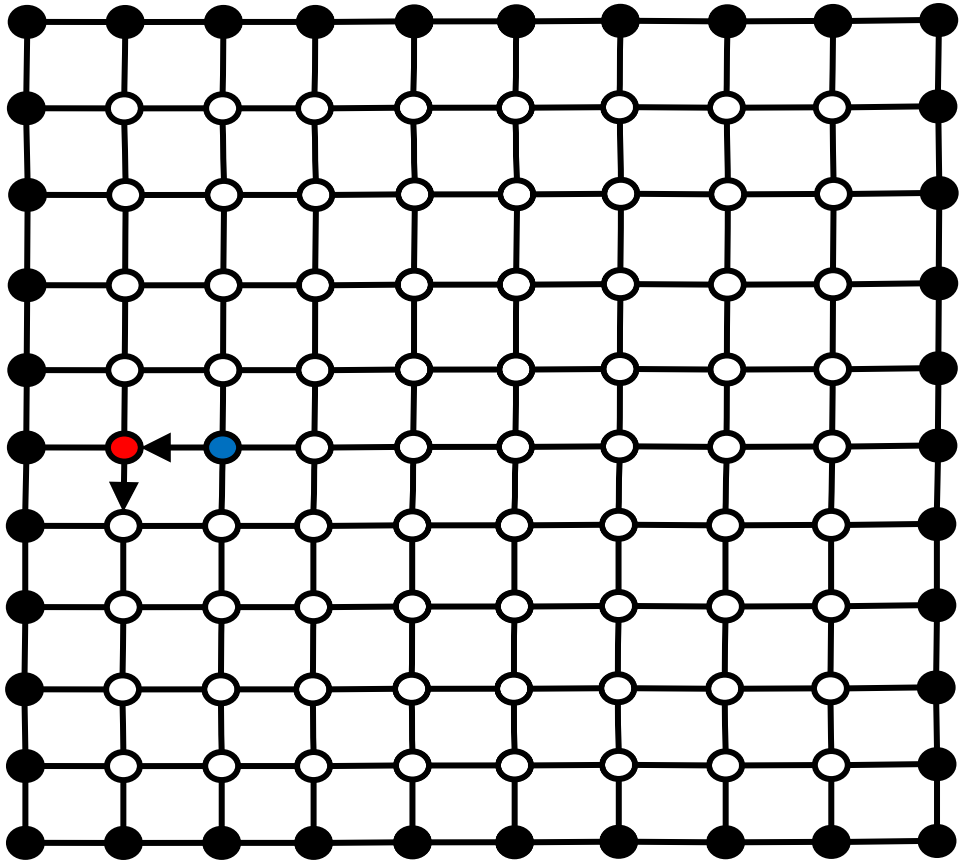




First Solution: range 1, 2 robots, 3 colors

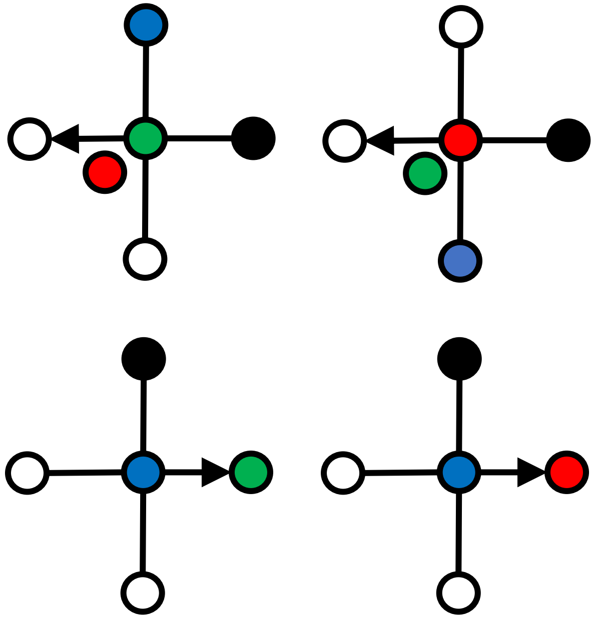
- Turn

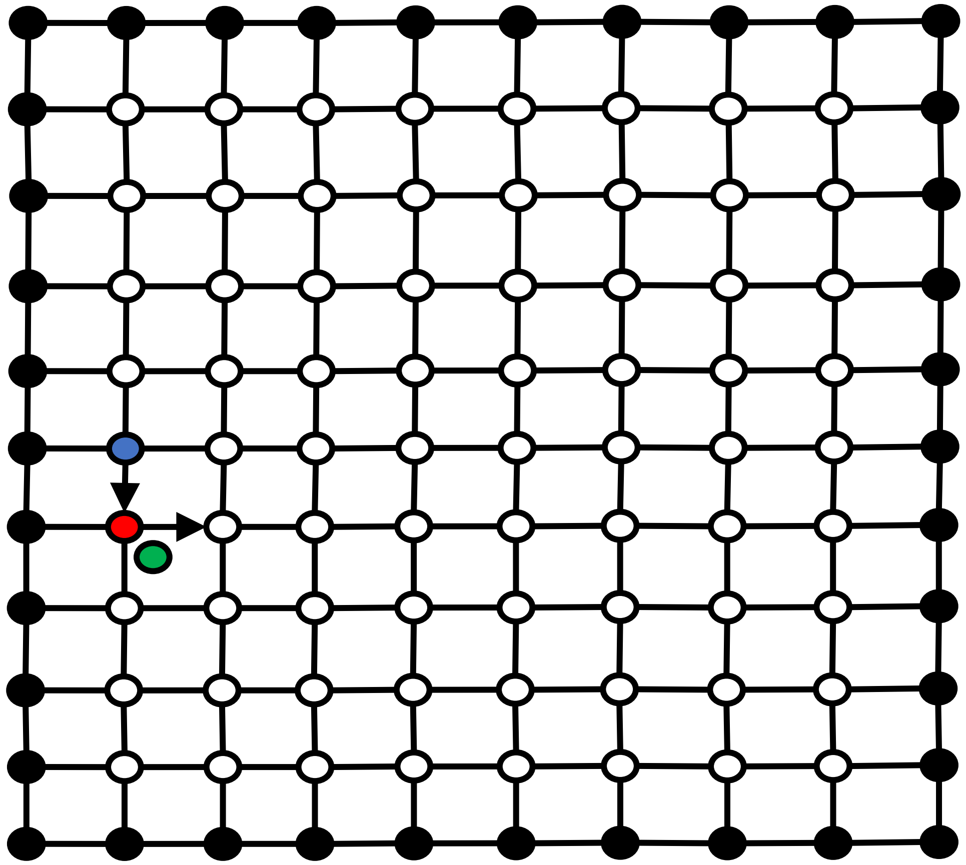




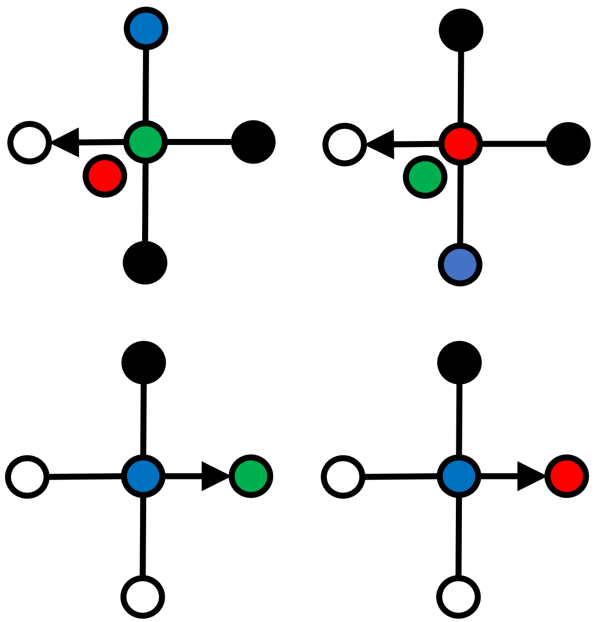
First Solution: range 1, 2 robots, 3 colors

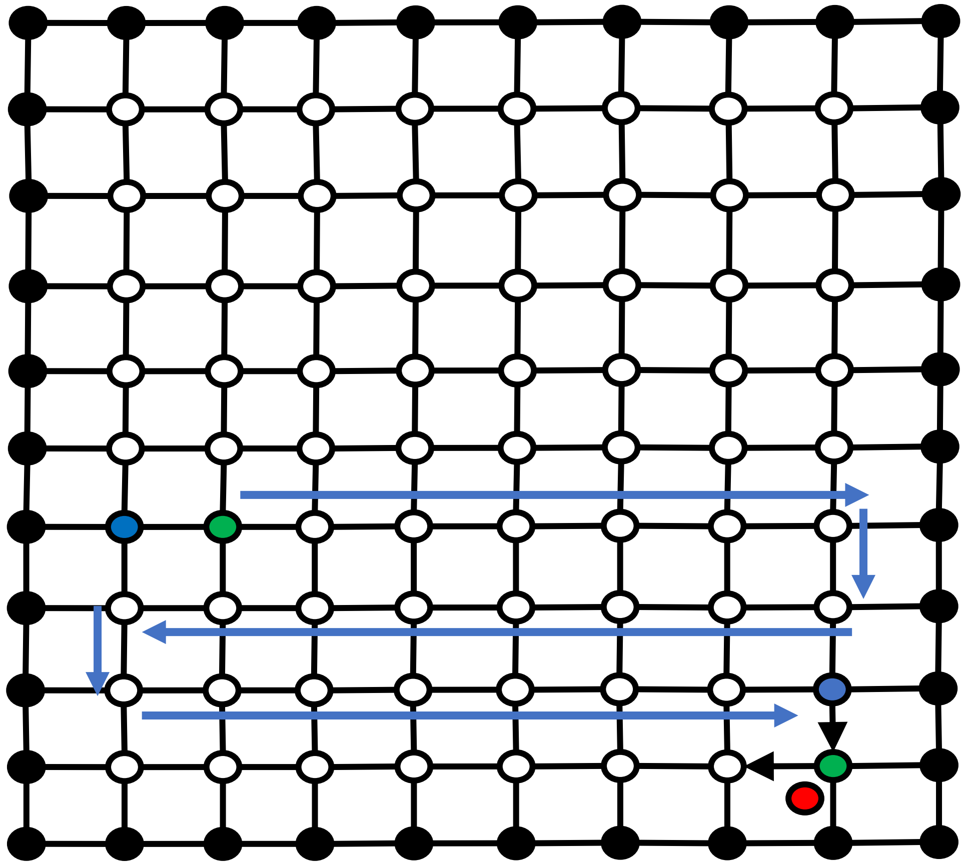
- Turn



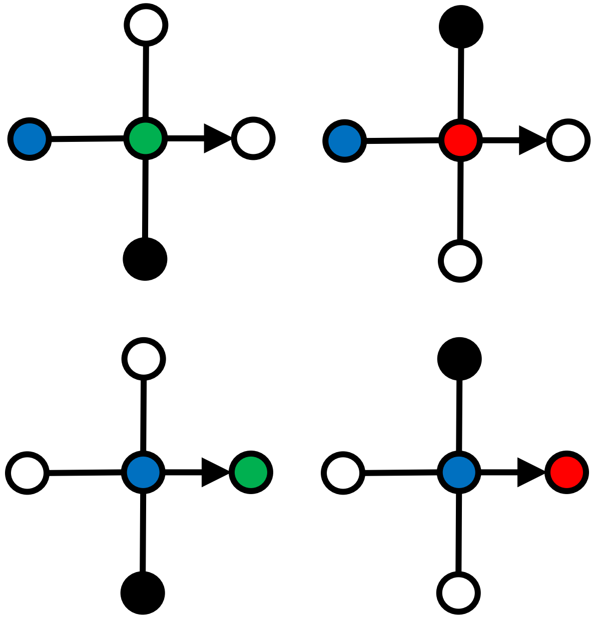


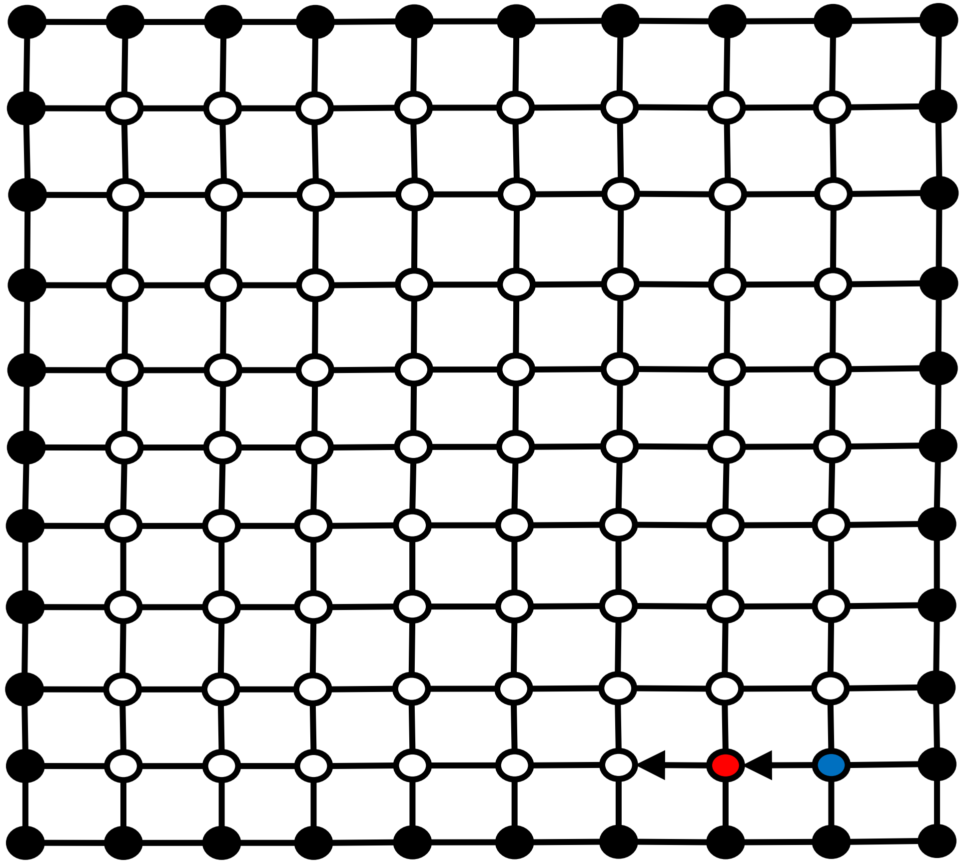
First Solution: range 1, 2 robots, 3 colors



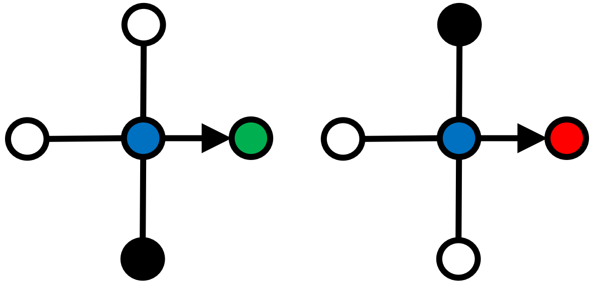


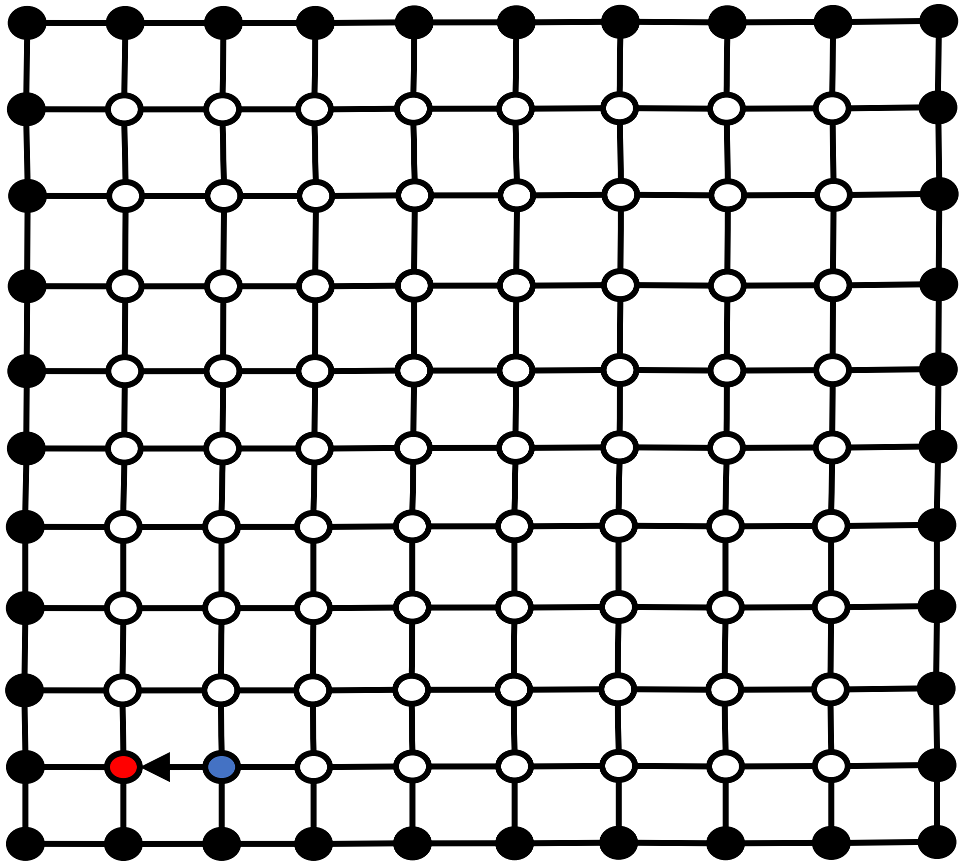
First Solution: range 1, 2 robots, 3 colors



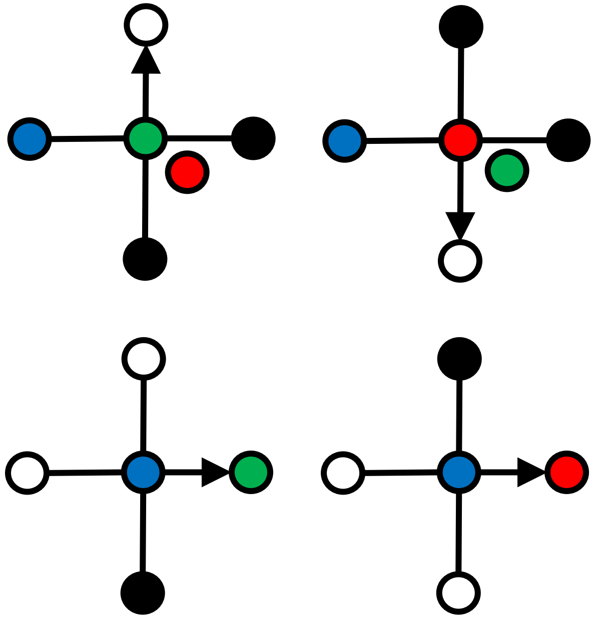


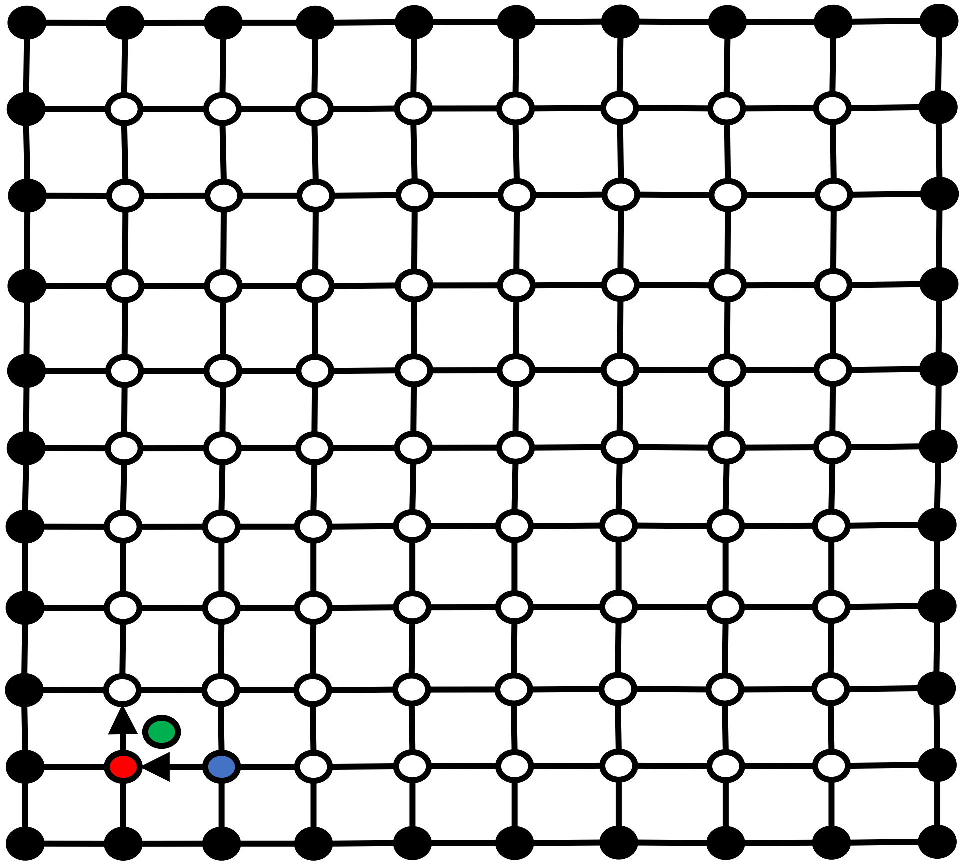
First Solution: range 1, 2 robots, 3 colors



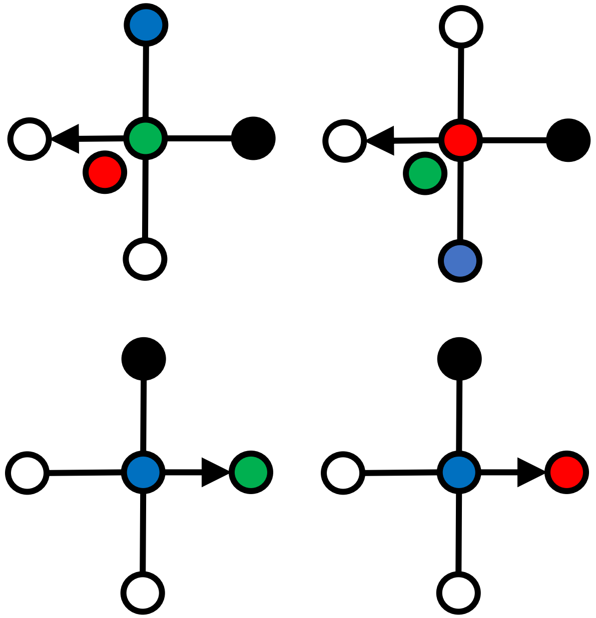


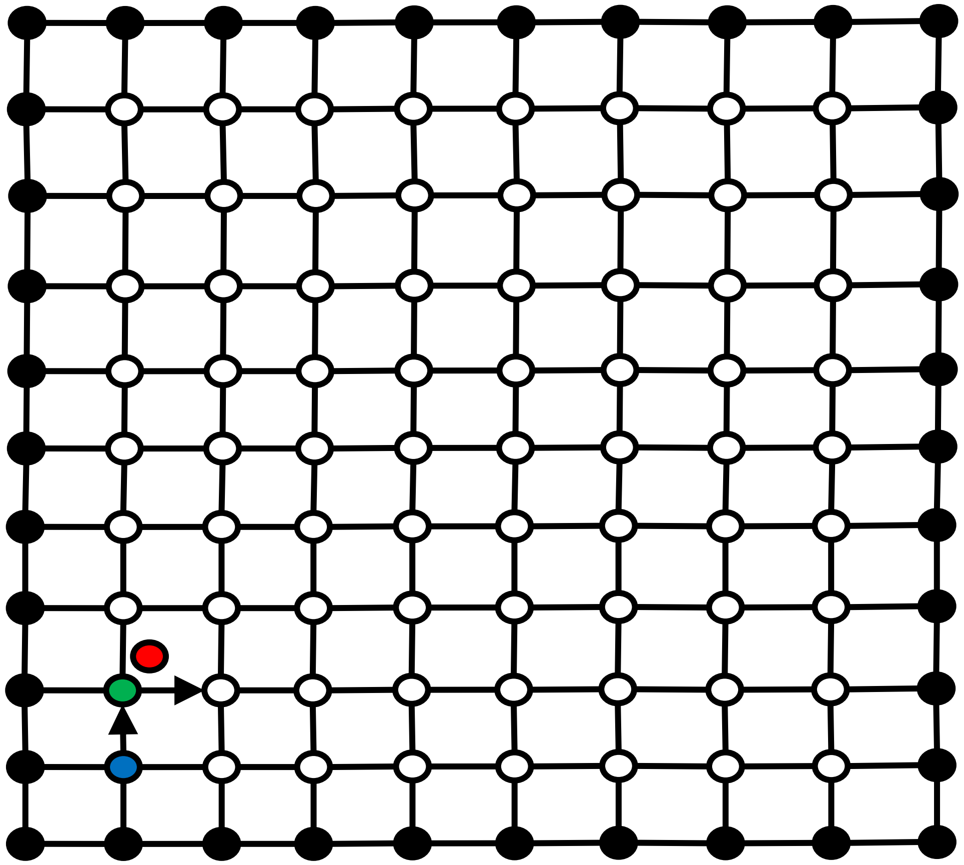
First Solution: range 1, 2 robots, 3 colors



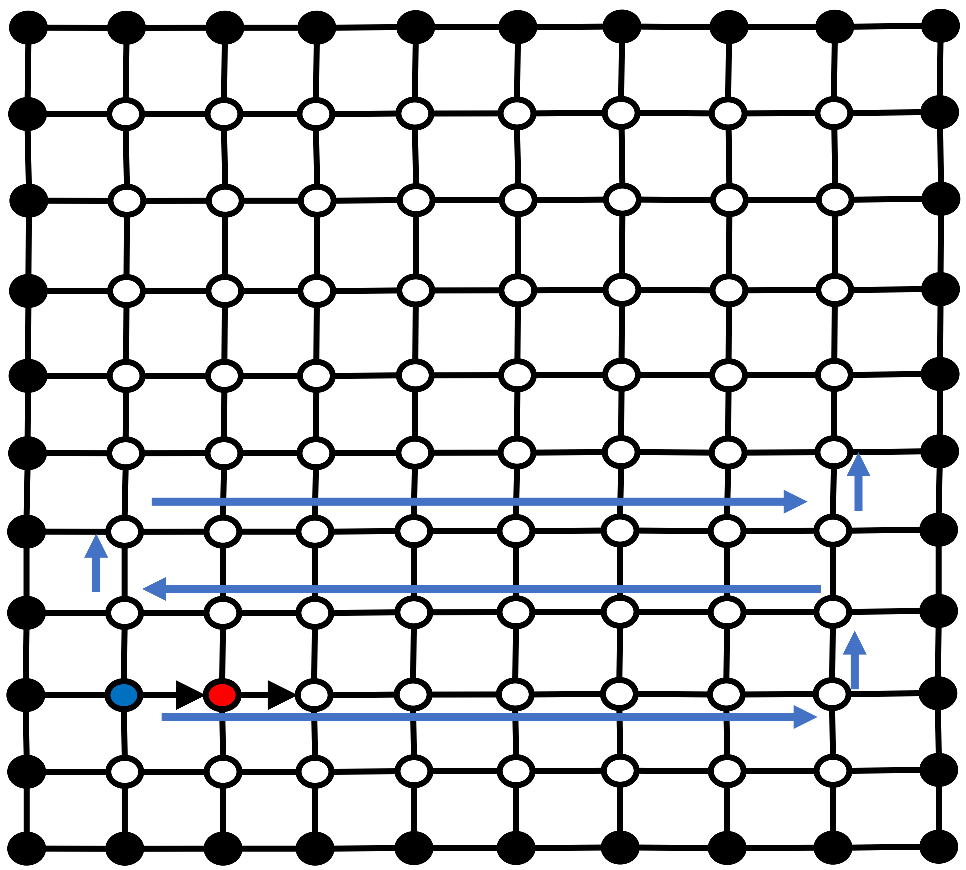


First Solution: range 1, 2 robots, 3 colors





First Solution: range 1, 2 robots, 3 colors



And so on …

Optimality

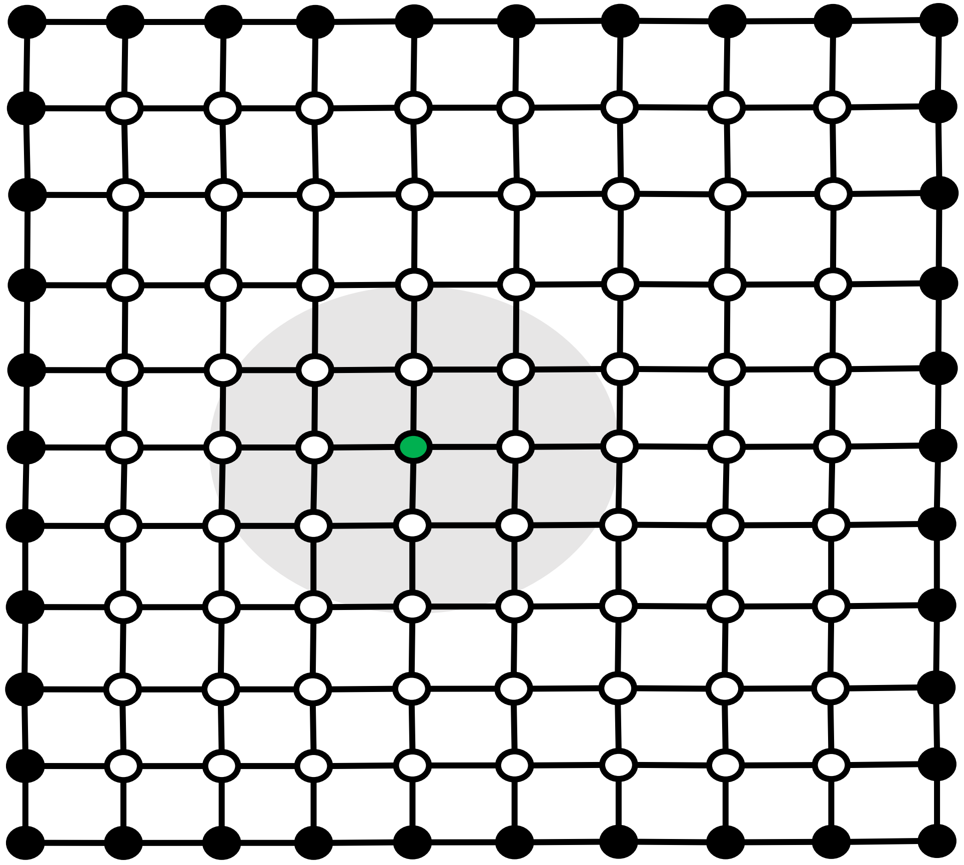
- 1 robot
- #color: finite
- Visibility range: finite

Impossible

Optimality

- 1 robot
- #color: finite
- Visibility range: finite

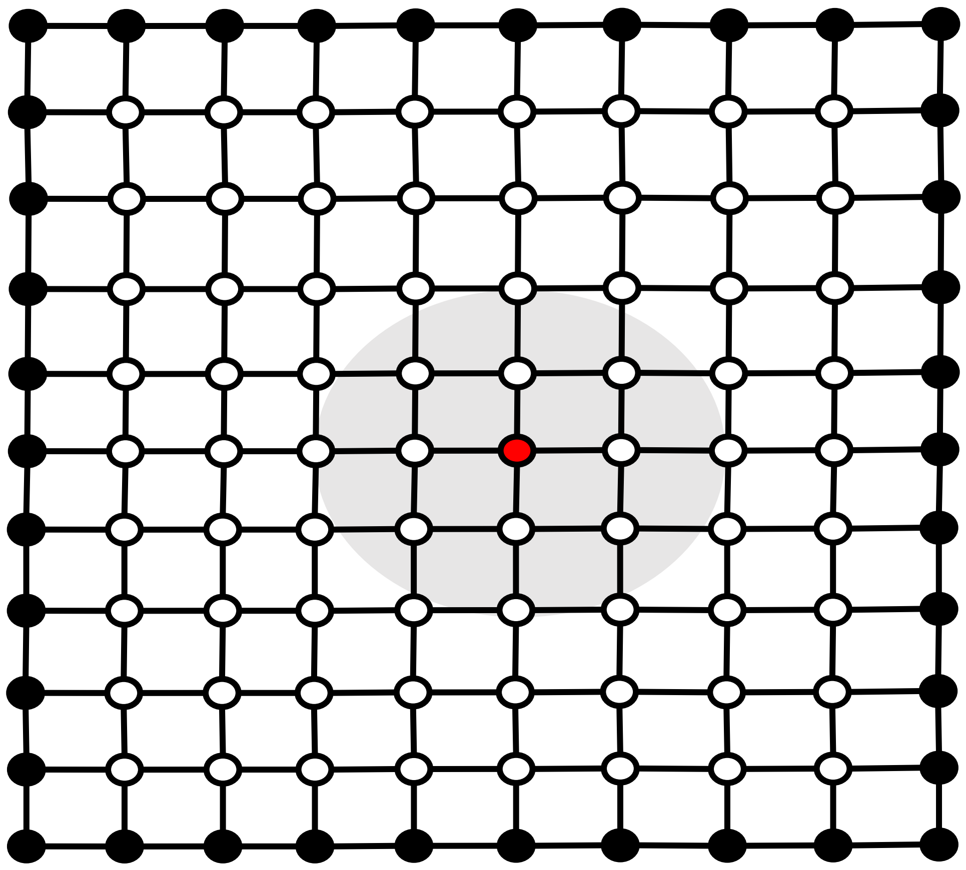
In a large enough grid,
when the robot reaches a center
it sees no wall



Optimality

- 1 robot
- #color: finite
- Visibility range: finite

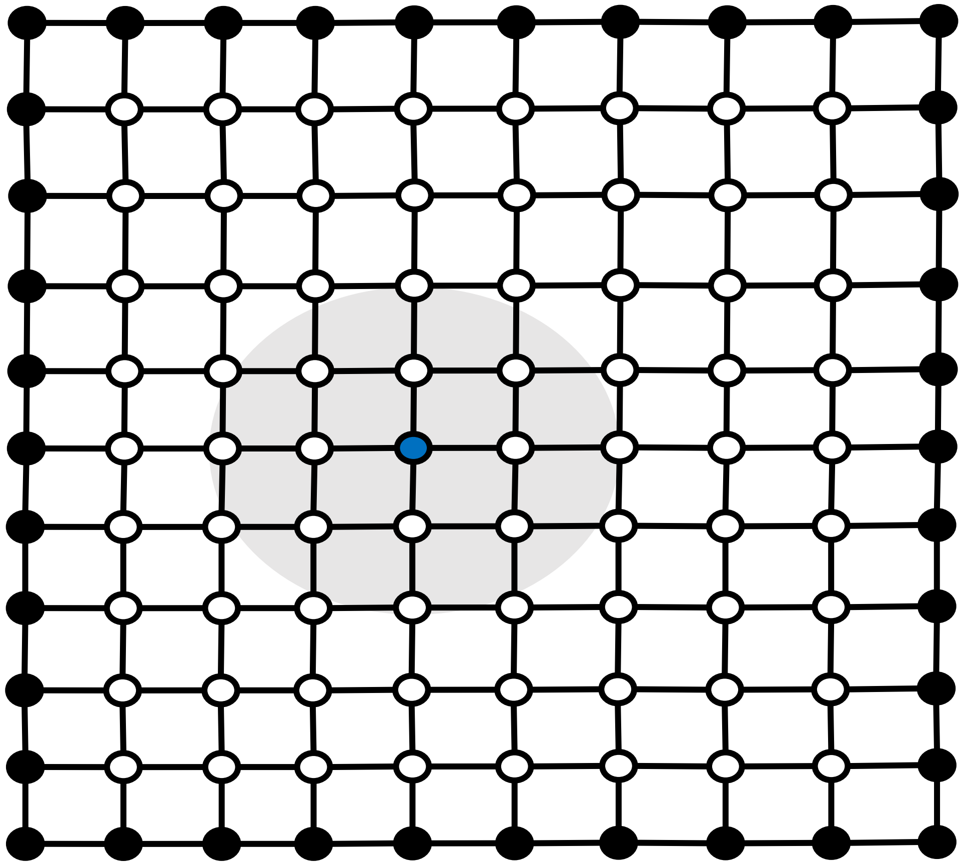
The robot may alternate between 2 positions



Optimality

- 1 robot
- #color: finite
- Visibility range: finite

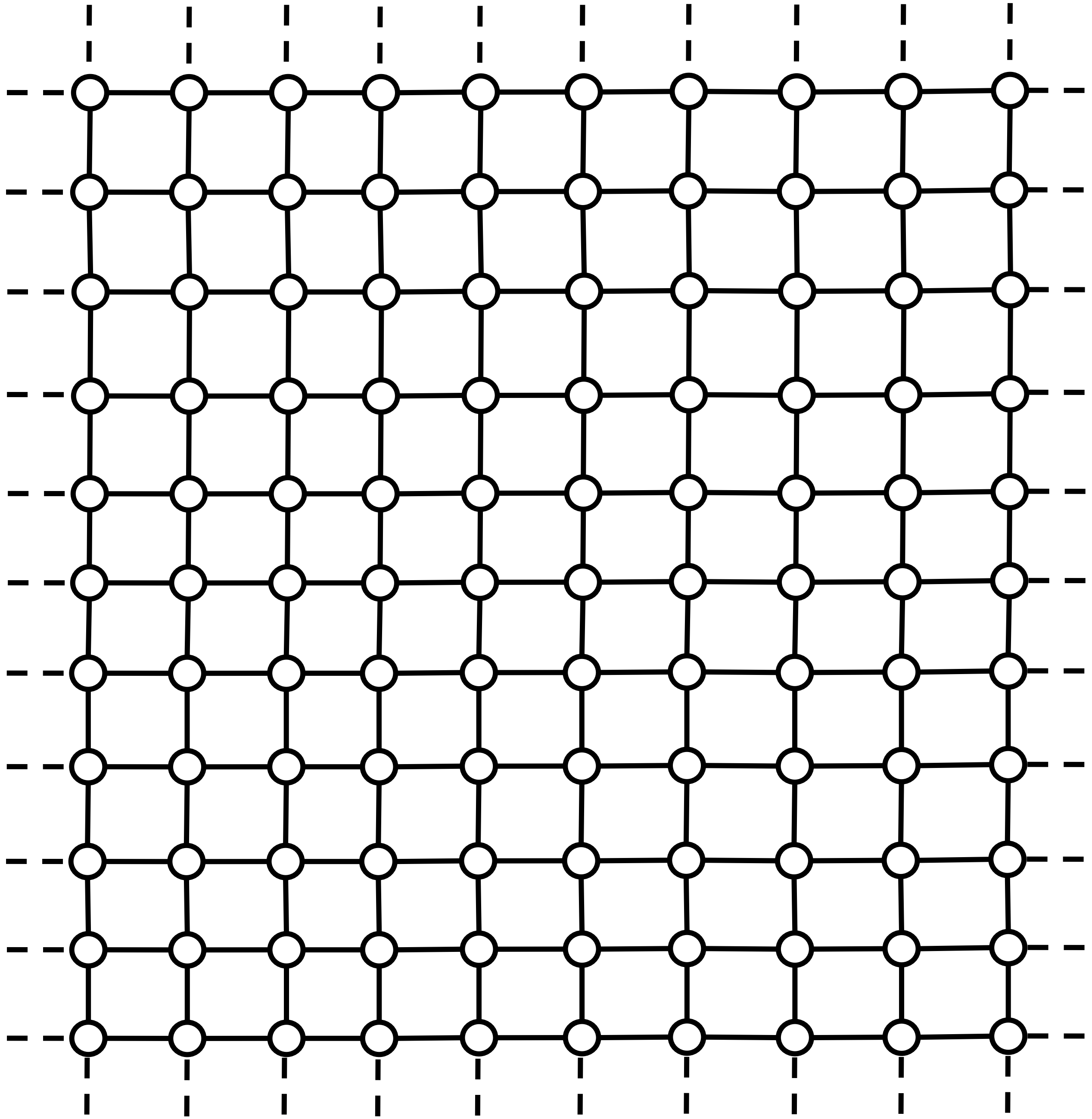
The robot may alternate between 2 positions



Optimality

- 2 robots
- 1 color
- Visibility range: finite

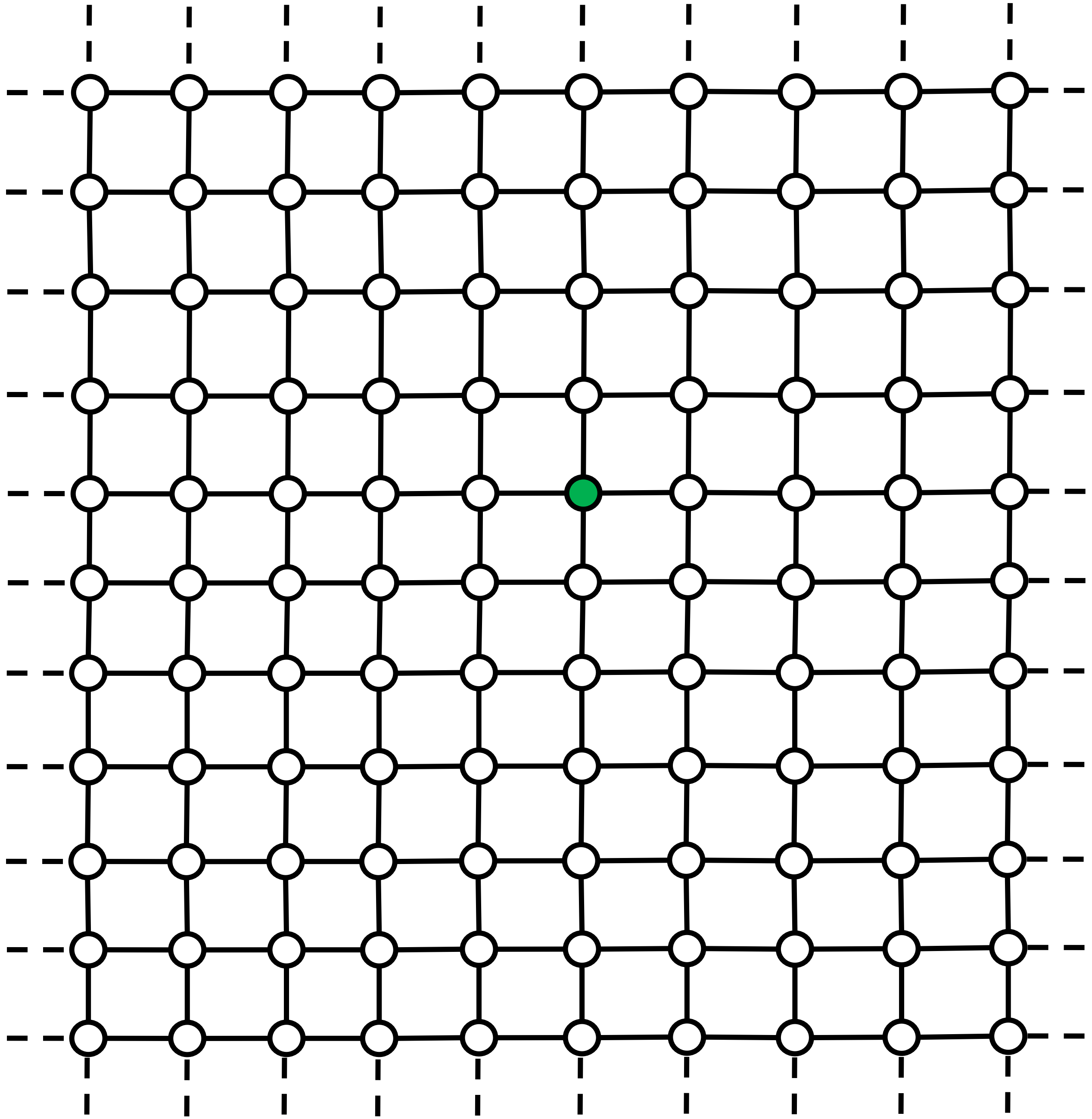
Impossible



Optimality

- 2 robots
- 1 color
- Visibility range: finite

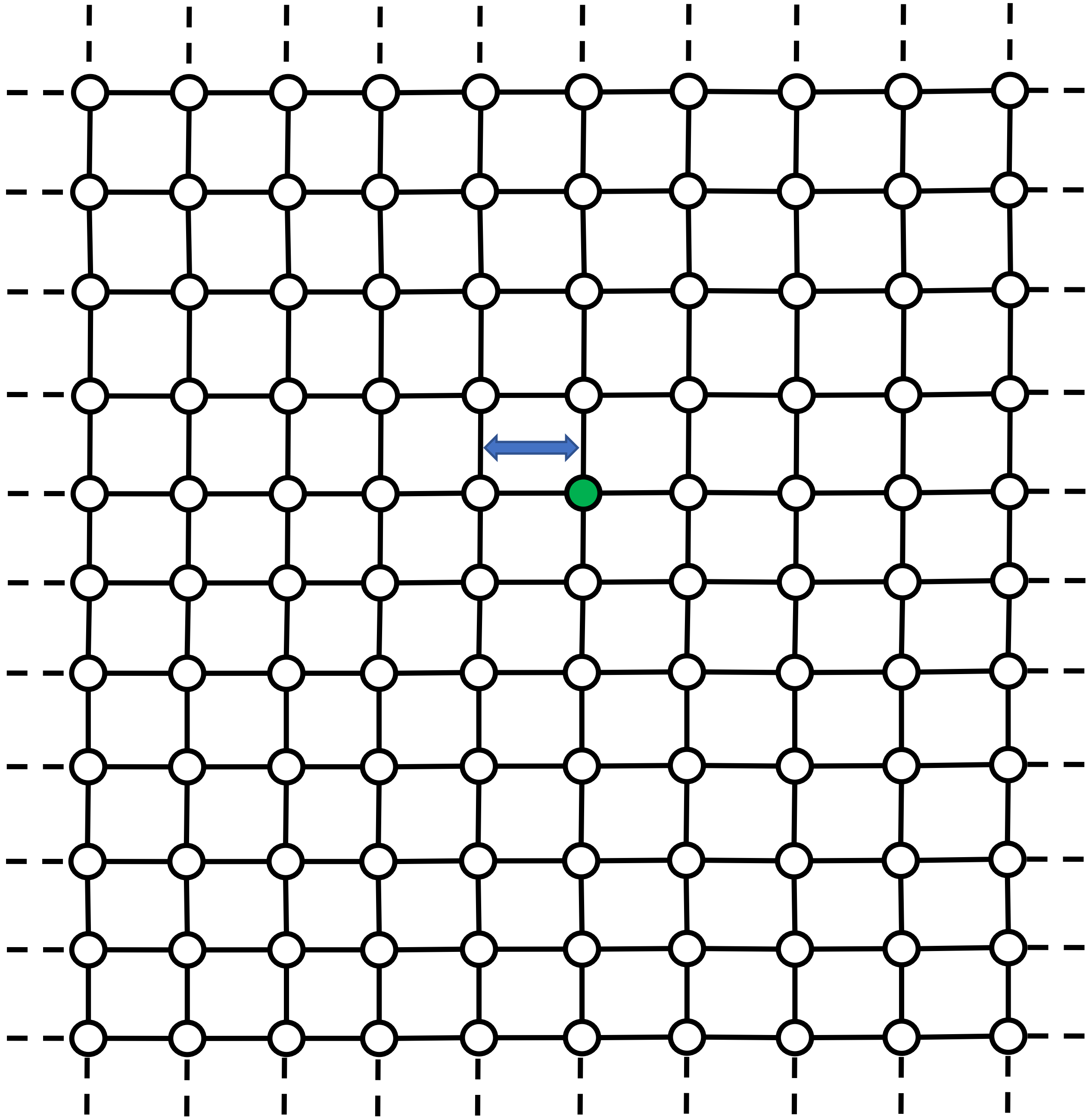
In a large enough grid,
a robot eventually reaches a
center where it sees no wall



Optimality

- 2 robots
- 1 color
- Visibility range: finite

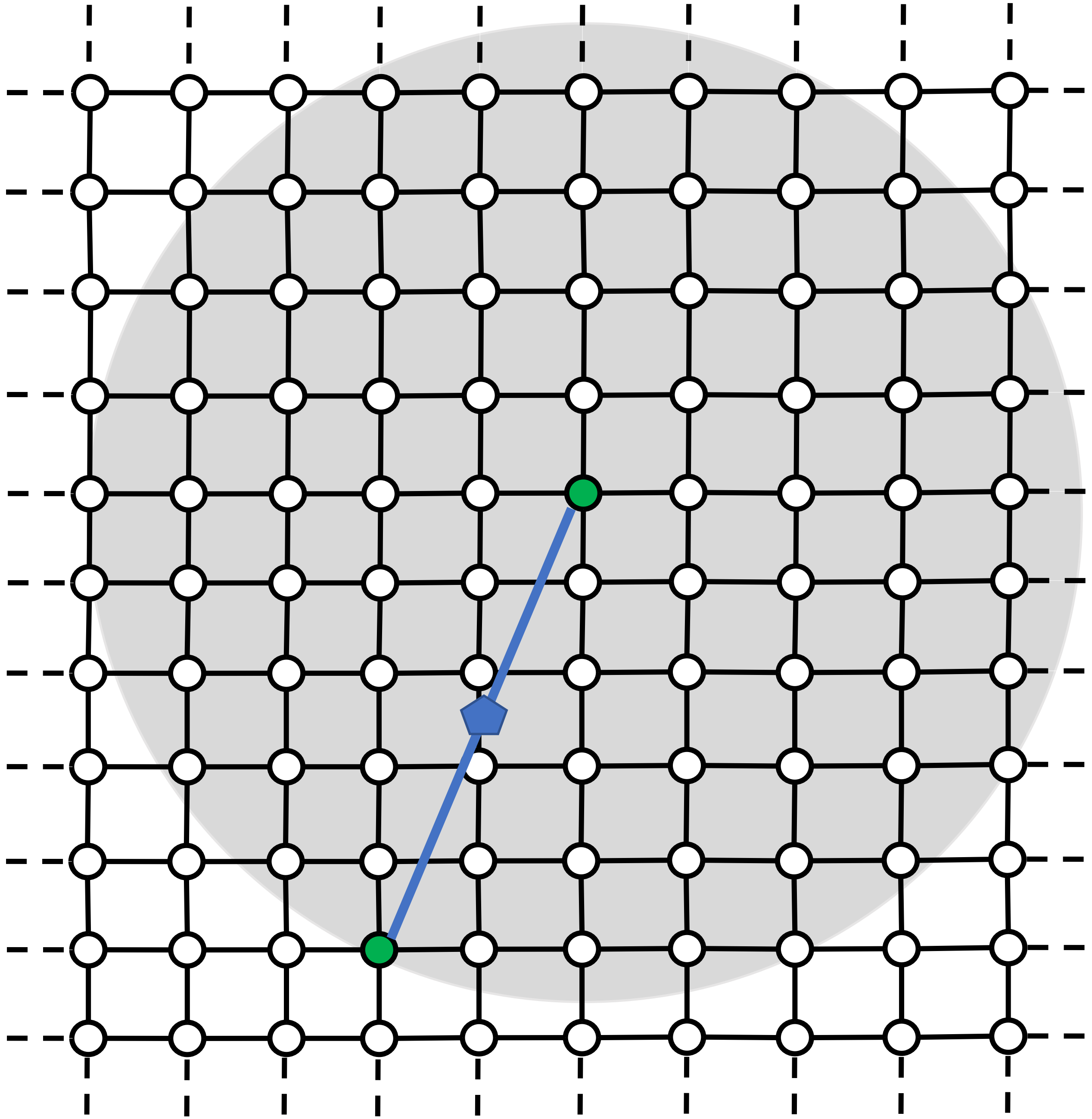
It may alternate between 2 positions until seing the second one



Optimality

- 2 robots
- 1 color
- Visibility range: finite

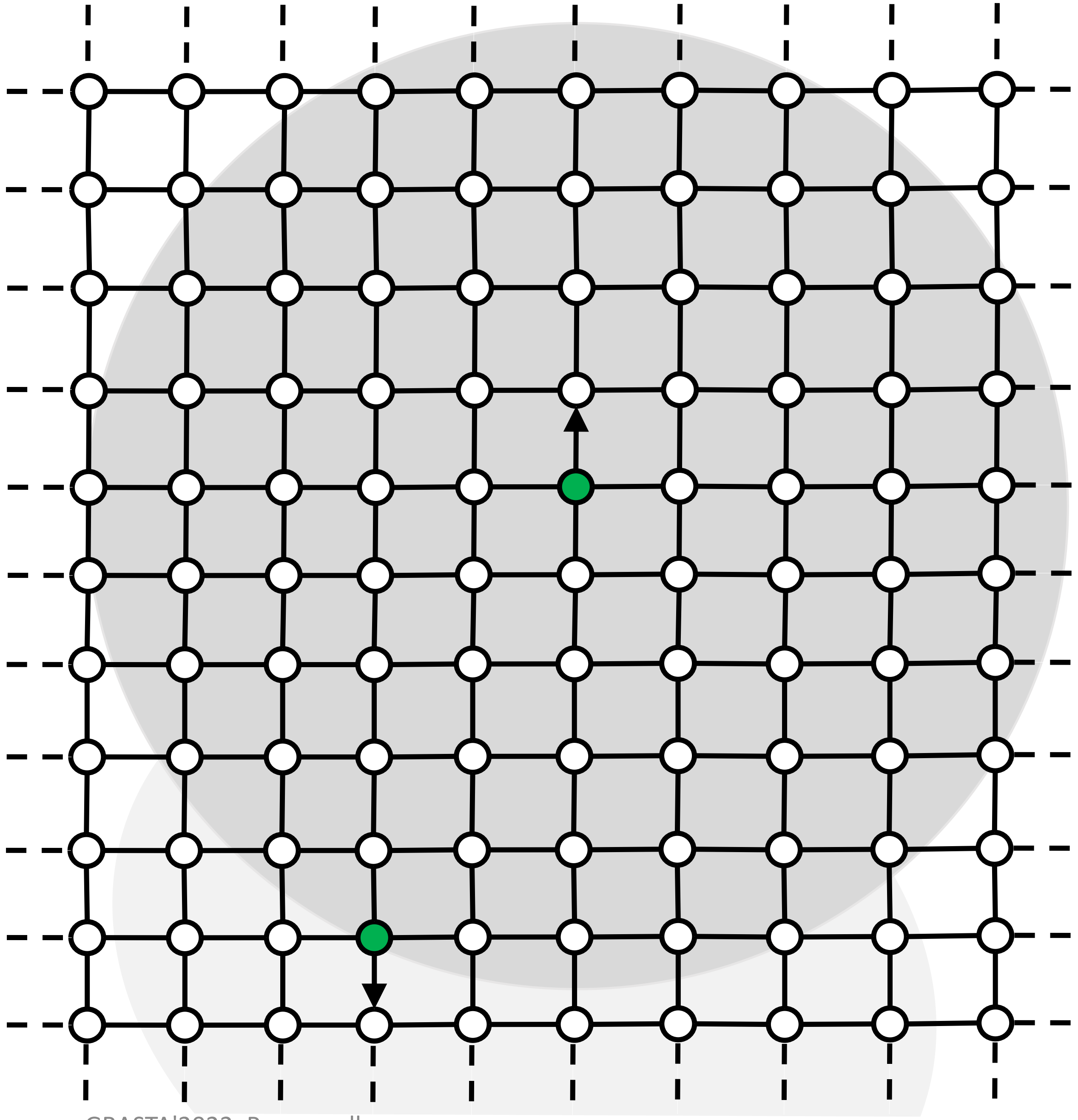
Once robots sees each others, the middle becomes invariant



Optimality

- 2 robots
- 1 color
- Visibility range: finite

They eventually move in opposite directions (to visit new nodes)

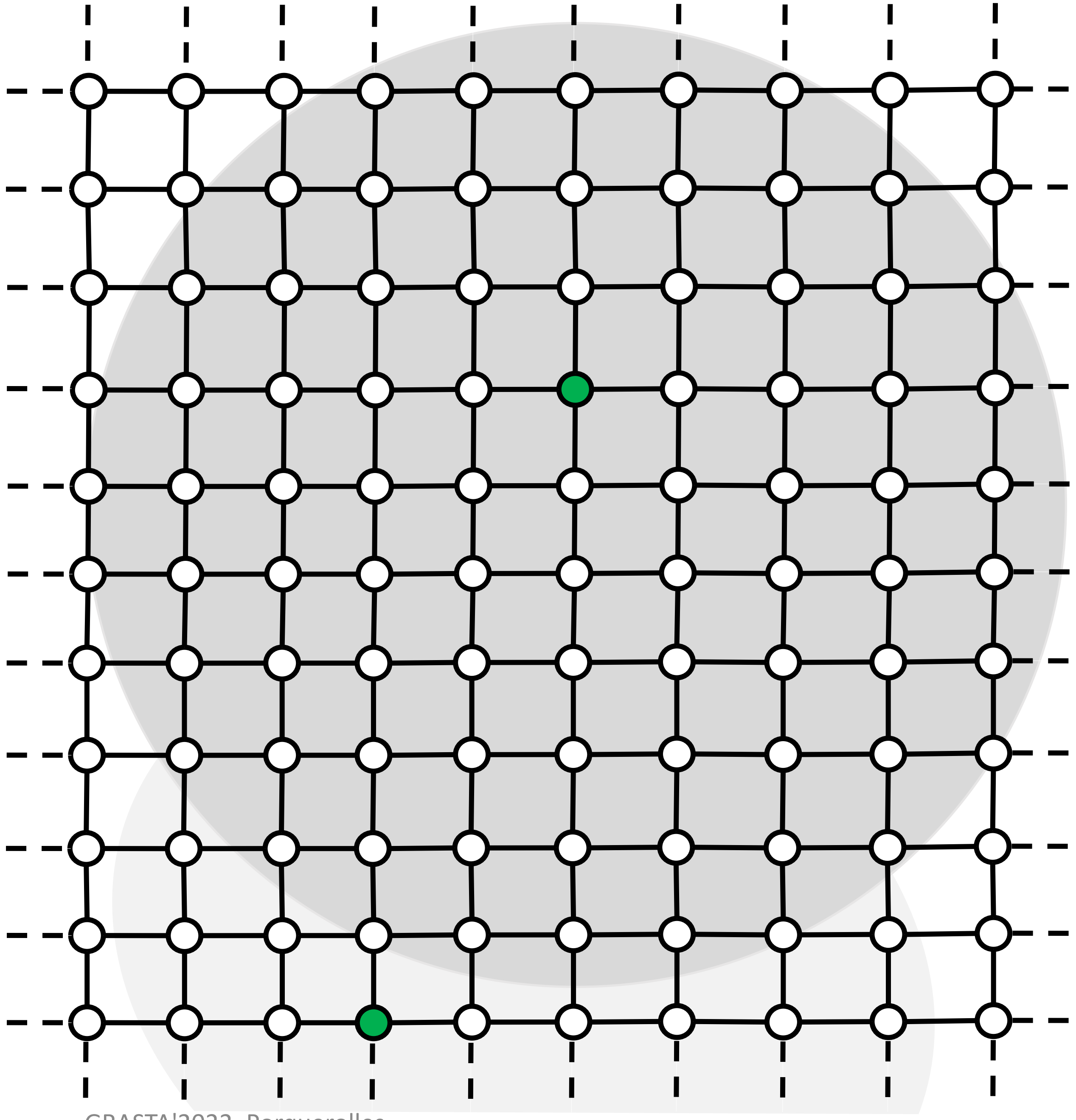


Optimality

- 2 robots
- 1 color
- Visibility range: finite

Eventually each robot sees neither a wall not the other robot

They can stay isolated forever while alternating between two positions



Optimality

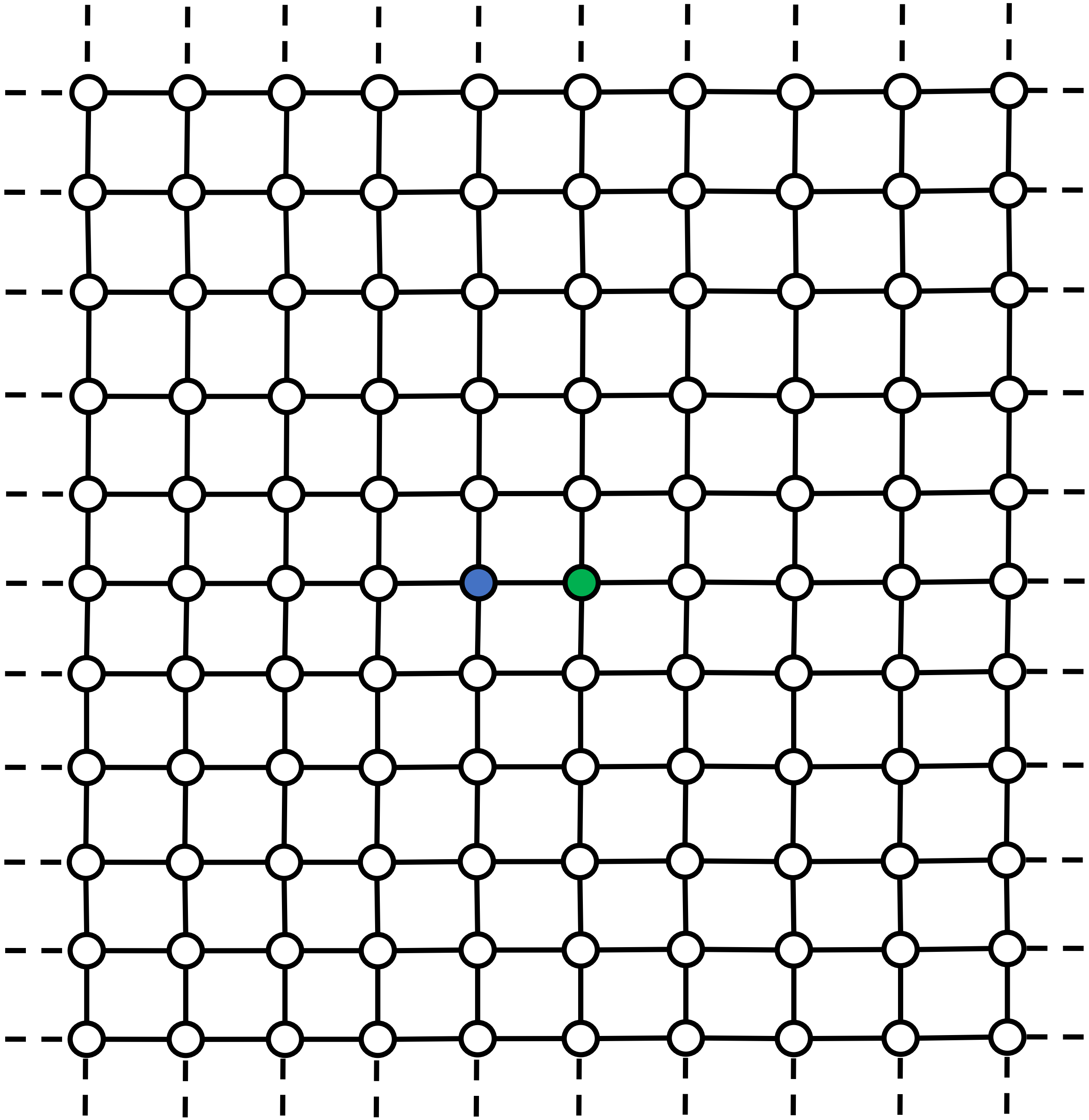
- 2 robots
- 2 colors
- Visibility range: 1

While a wall is not reached,

they should stay grouped

with different colors

(otherwise similar to previous cases)

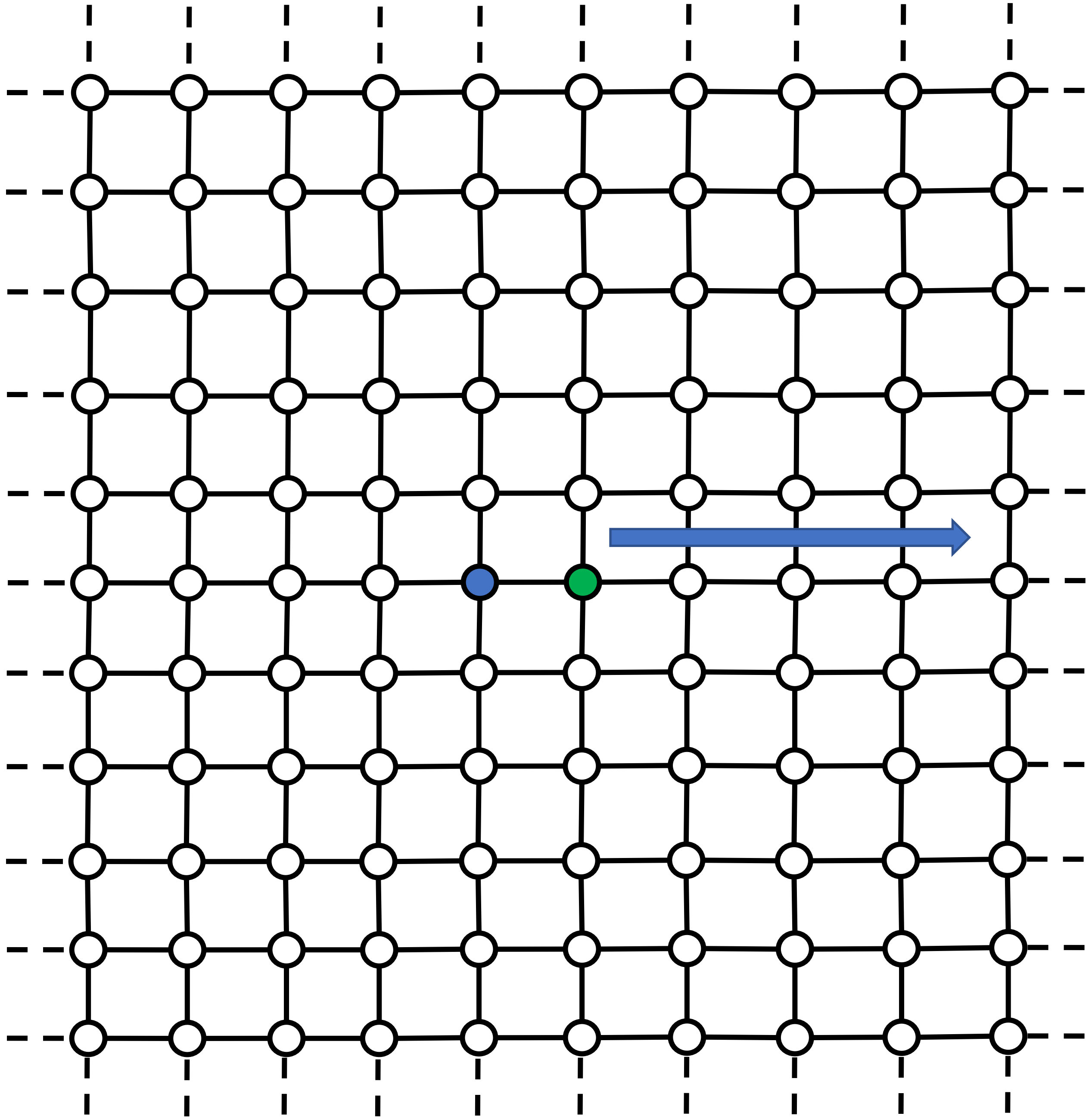


Optimality

- 2 robots
- 2 colors
- Visibility range: 1

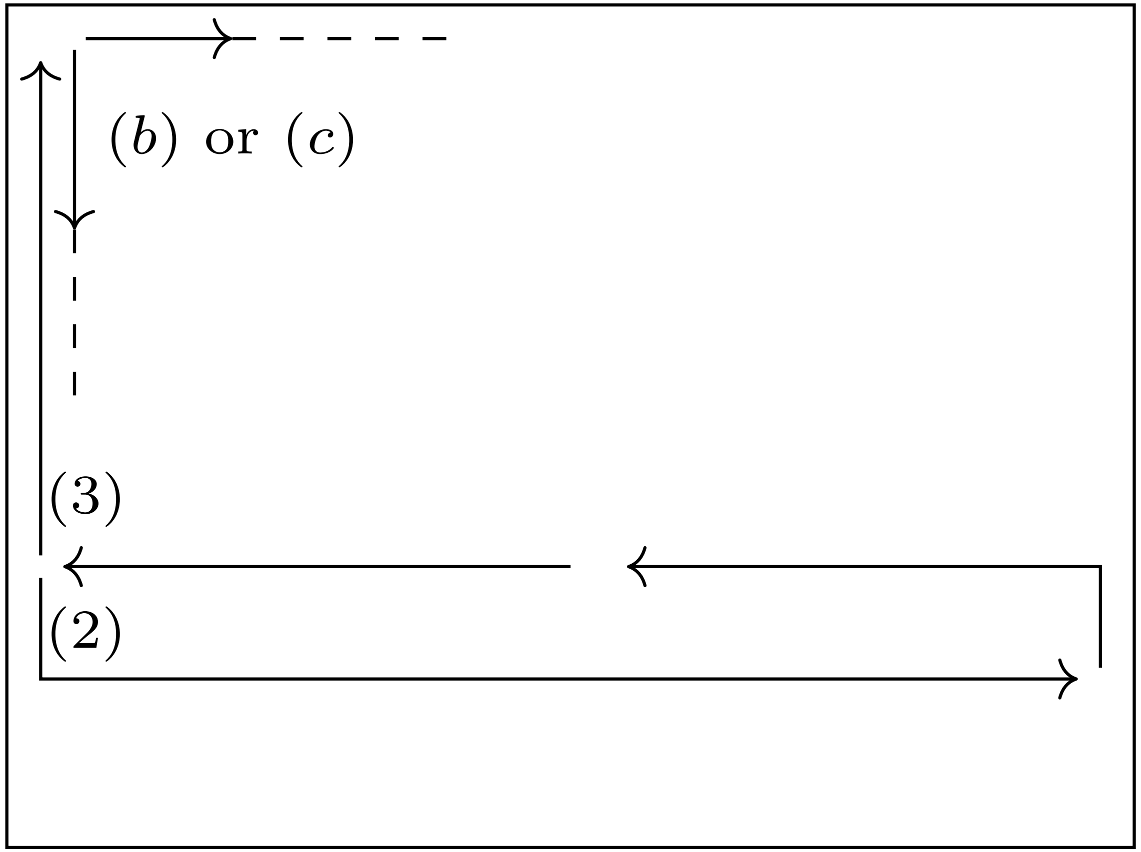
Until reaching a wall, they move in one direction

(*n.b.*, if they change their color, they are trapped into 3 nodes)



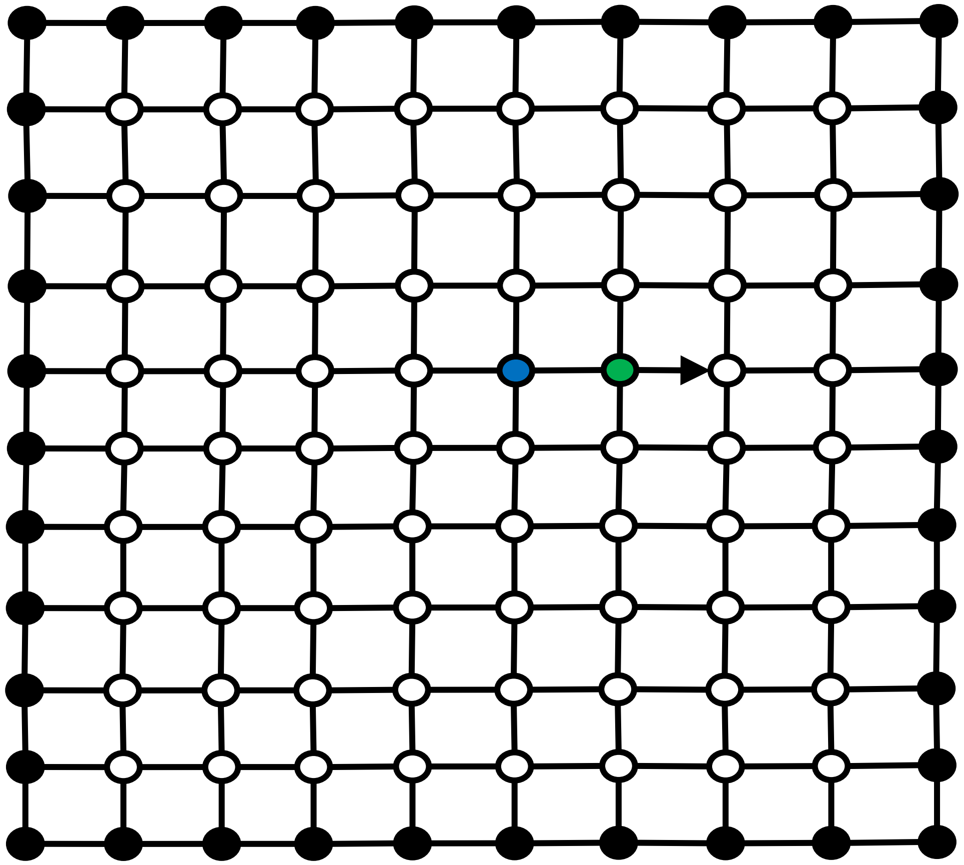
Optimality

- 2 robots
- 2 colors
- Visibility range: 1



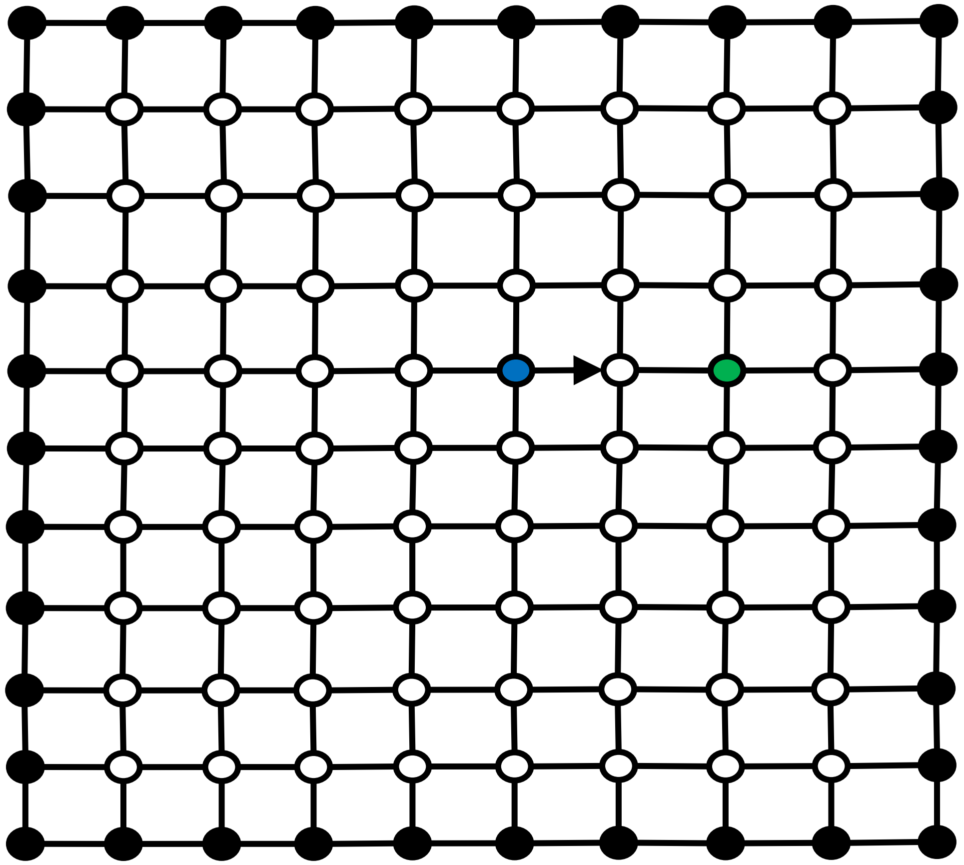
Asynchronous Version

- Visibility range **2**
- Similar principles, yet **sequentialization**
    - The leader executes as in the synchronous version
    - The follower waits until being at distance 2 before following the leader



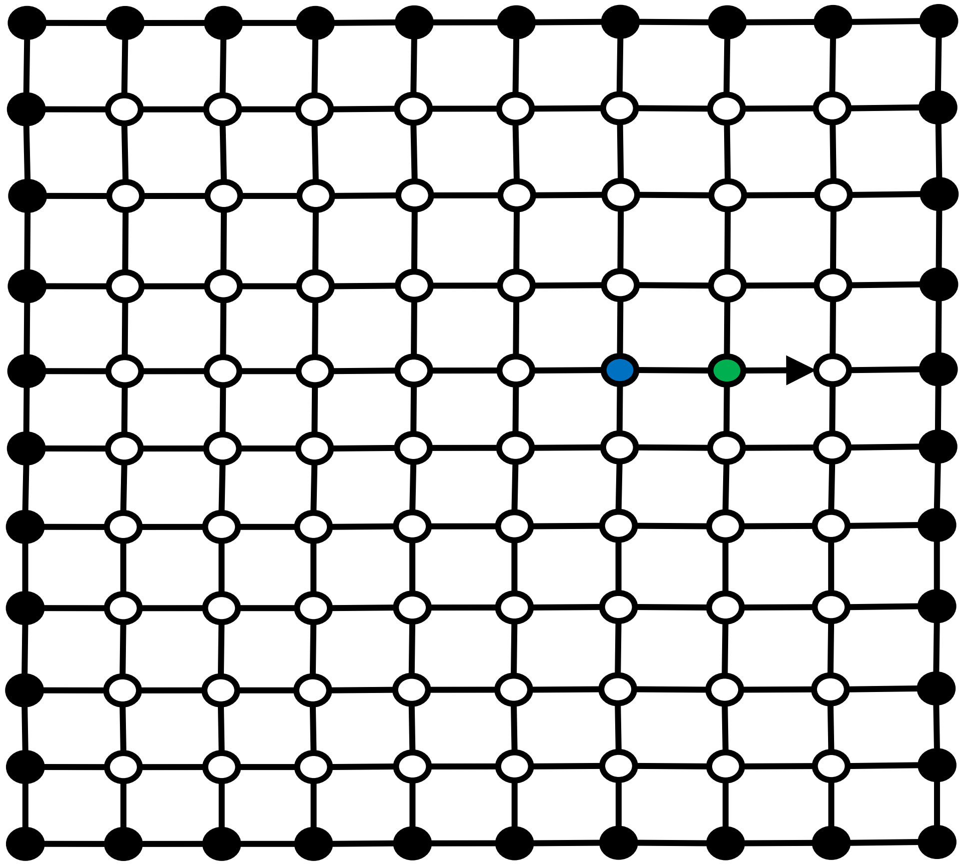
Asynchronous Version

- Visibility range **2**
- Similar principles, yet **sequentialization**
  - The leader executes as in the synchronous version
  - The follower waits until being at distance 2 before following the leader



Asynchronous Version

- Visibility range **2**
- Similar principles, yet **sequentialization**
  - The leader executes as in the synchronous version
  - The follower waits until being at distance 2 before following the leader



Other Solutions

- https://bramas.fr/static/ICDCN2021/2-robots-2-colors-range-2.html

- https://bramas.fr/static/ICDCN2021/3-robots-1-color-range-2.html

Terminating Variants

- $F \rightarrow F_0, F_1, F_2, F_3, F_4$
- Count the number of times the follower left a corner
- Similar to the non-terminating version
  - Just implement the counting mechanism
- Termination when its state is F_4
- https://bramas.fr/static/TCS2021/2-robot-7-colors-terminating.html

Extensions & Perspectives

- **Without chirality:** NETYS’2021
- **Termination and obliviousness**
  - with 4 robots, range 2, yet not always locally-defined (journal version). Optimal ?
  - https://bramas.fr/static/TCS2021/4-robots-1-color-terminating.html
- **Higher dimension (3D):** FUN’2022
- Widely explore the **asynchronous** case

Thank You!