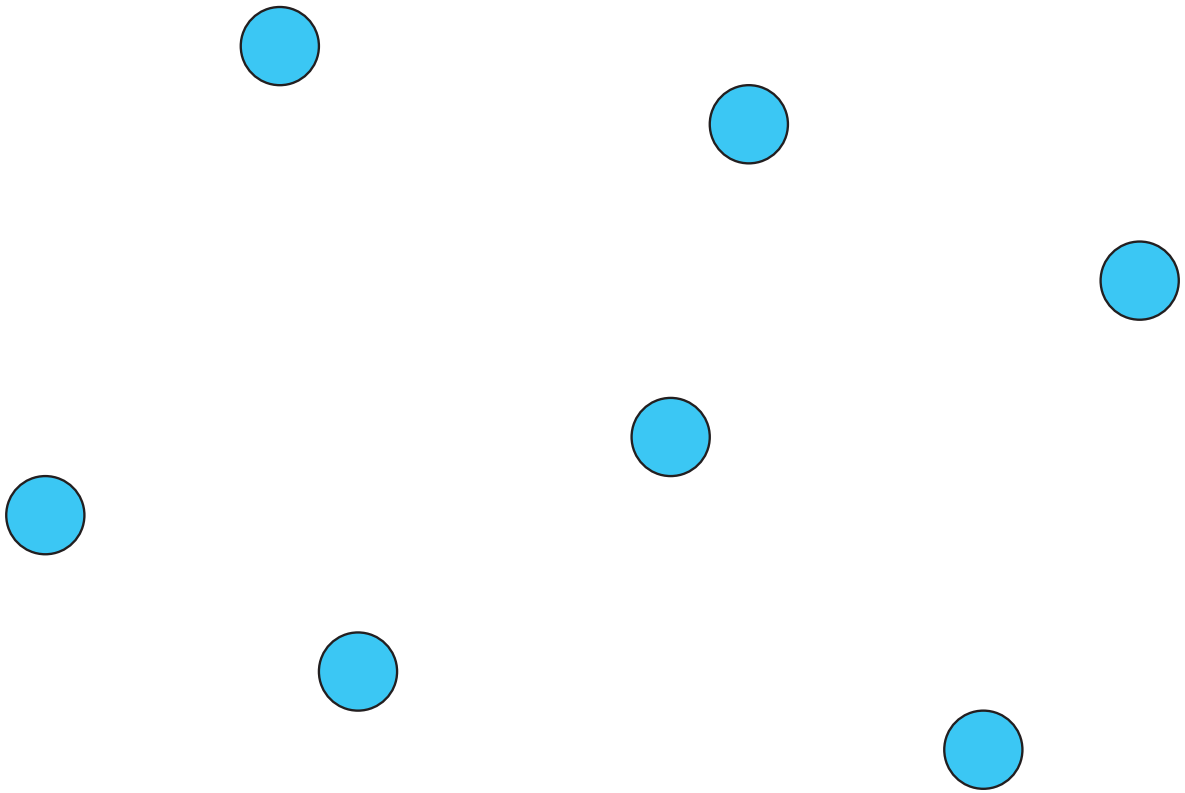
The Mutual Visibility Problem for Oblivious Robots

CCCG 2014

Giuseppe Antonio Di Luna, Paola Flocchini, Federico Poloni,
Nicola Santoro, and Giovanni Viglietta

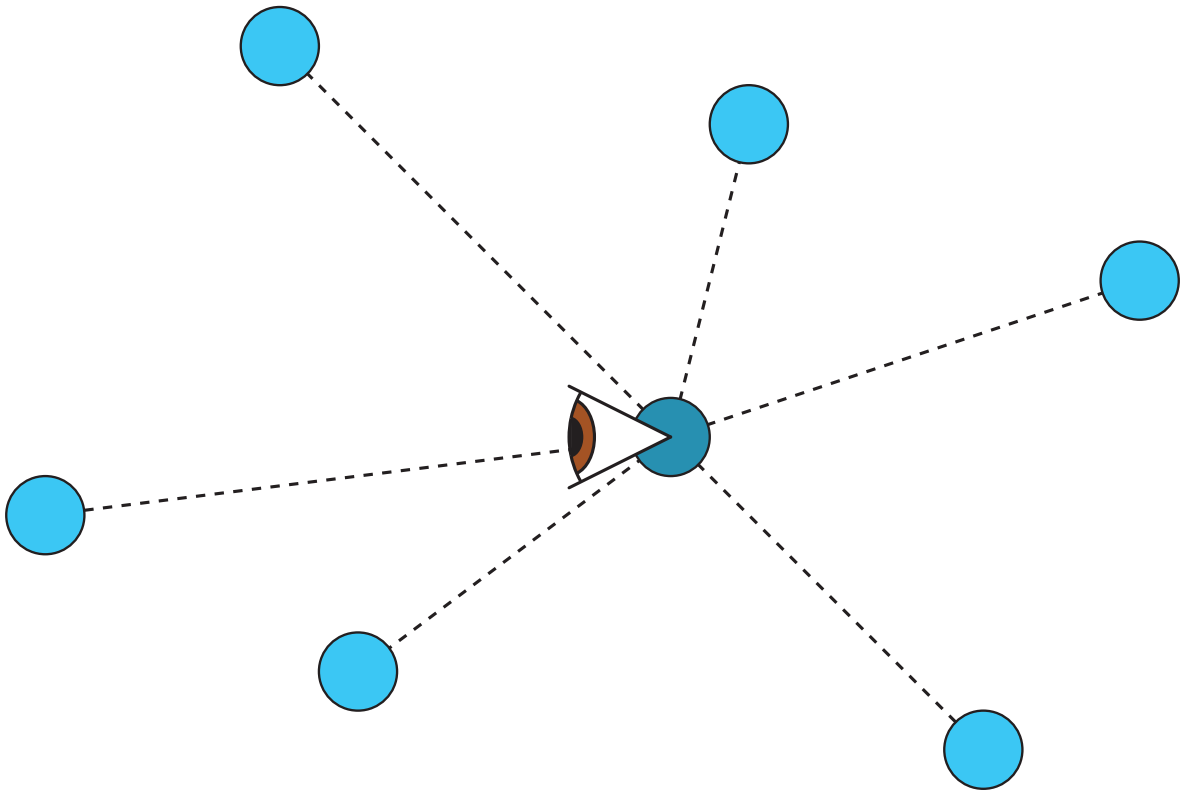
Halifax – August 13, 2014

Anonymous robots sensing and moving in the plane



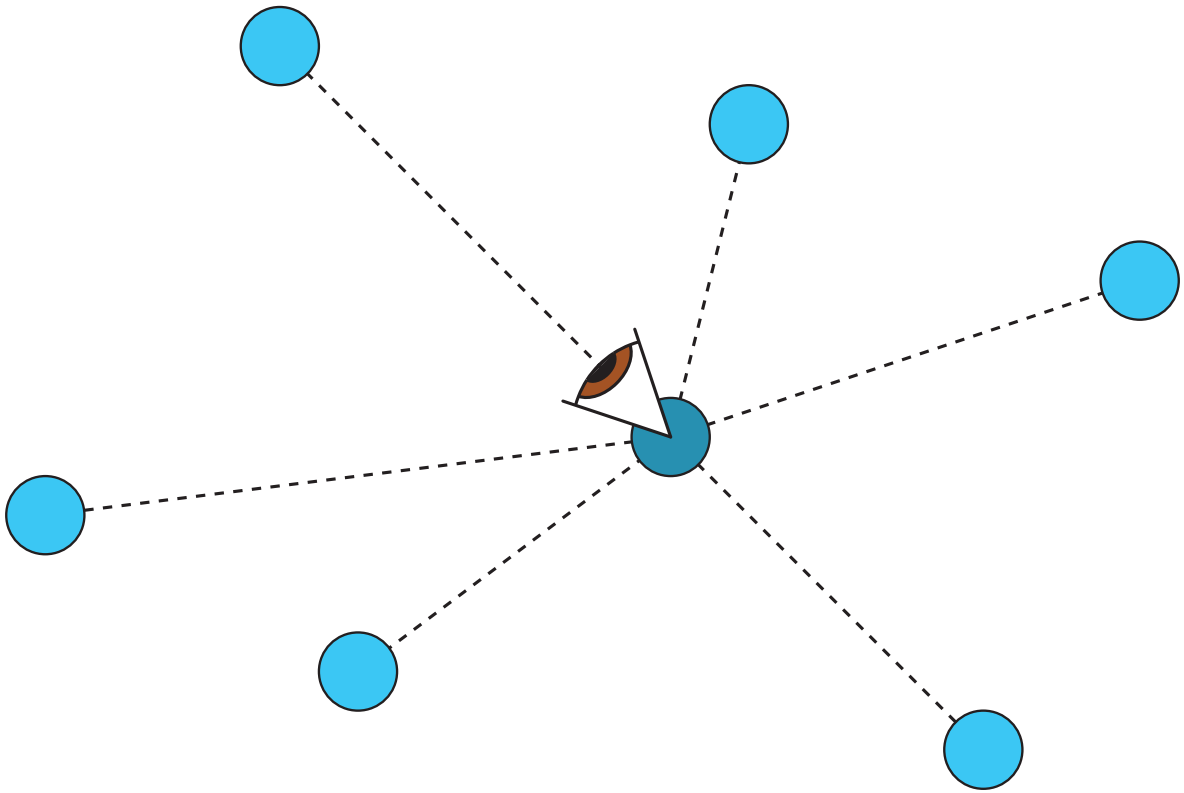
Swarm of anonymous robots

Anonymous robots sensing and moving in the plane



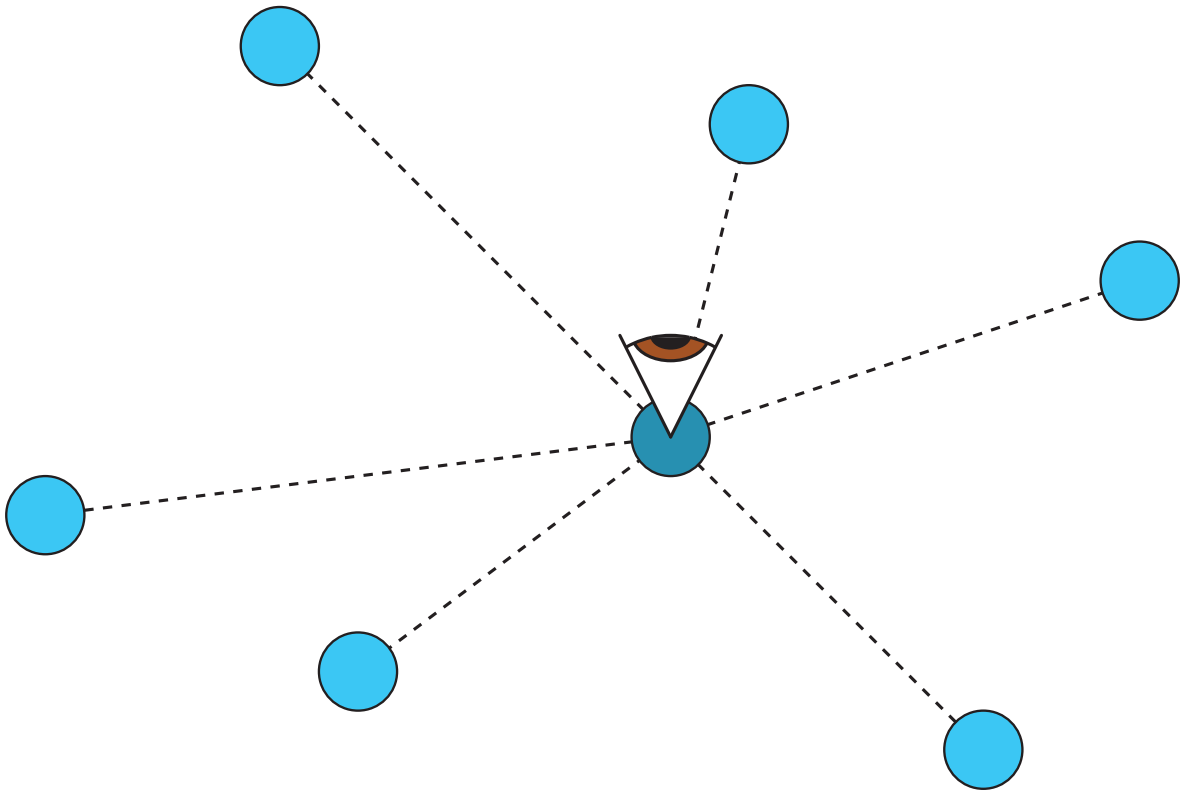
Sensing the positions of other robots

Anonymous robots sensing and moving in the plane



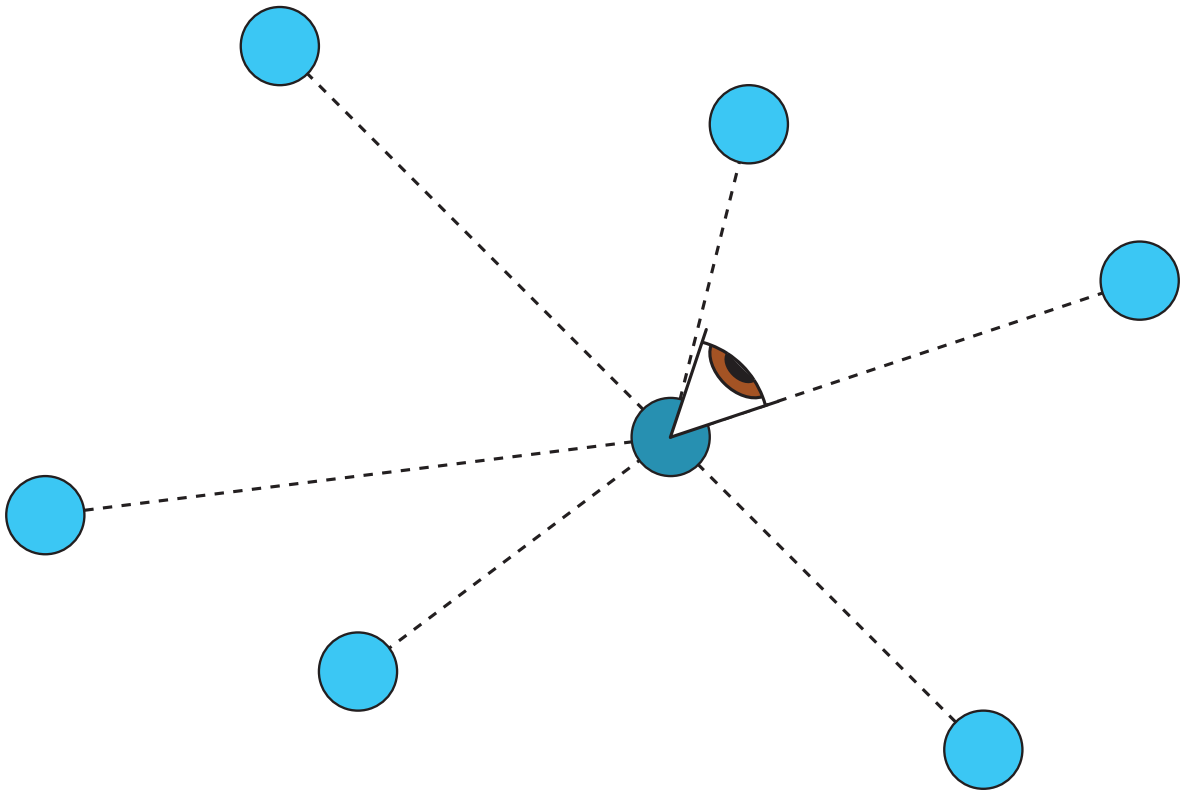
Sensing the positions of other robots

Anonymous robots sensing and moving in the plane



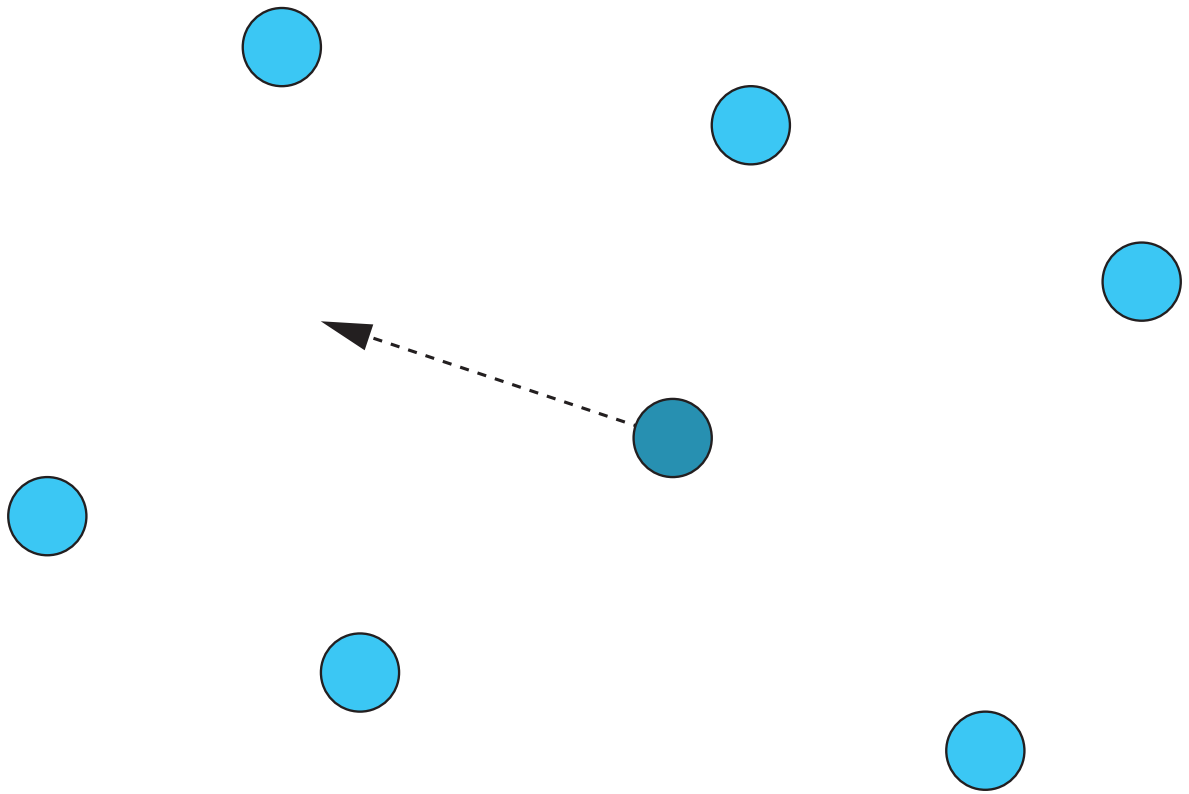
Sensing the positions of other robots

Anonymous robots sensing and moving in the plane



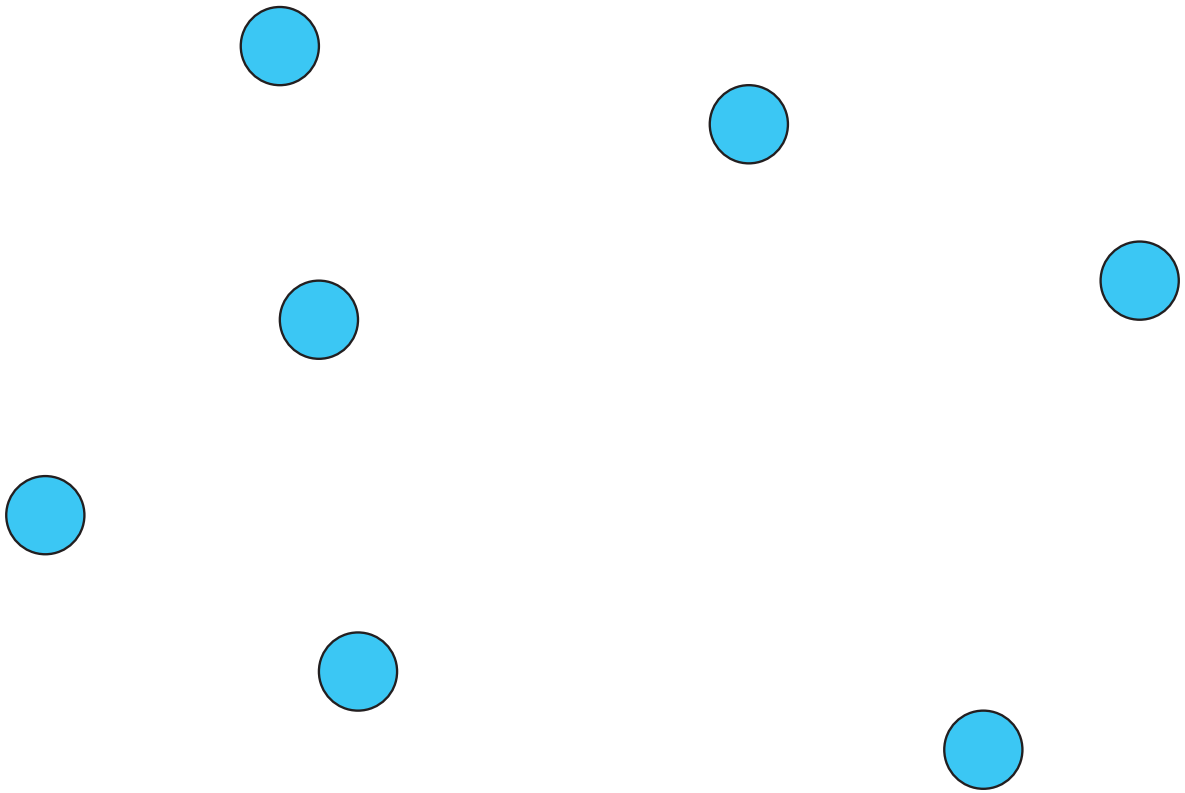
Sensing the positions of other robots

Anonymous robots sensing and moving in the plane



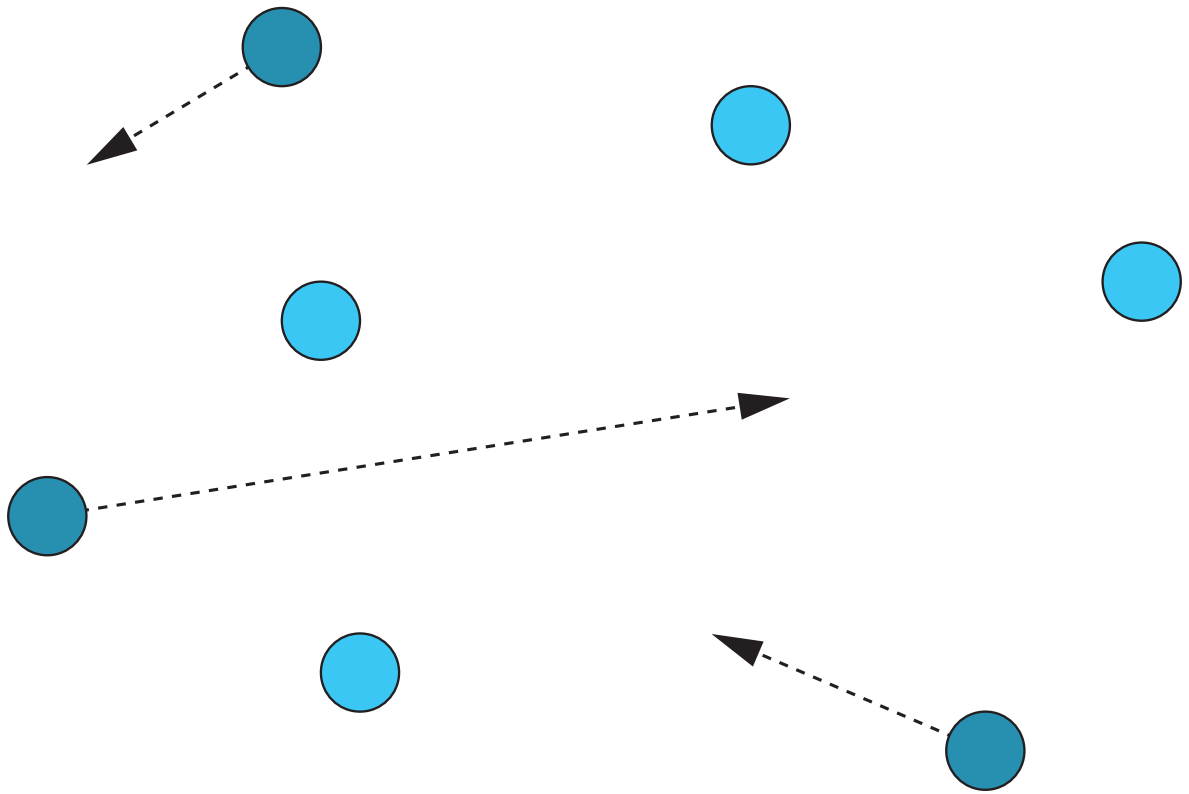
Moving accordingly

Anonymous robots sensing and moving in the plane



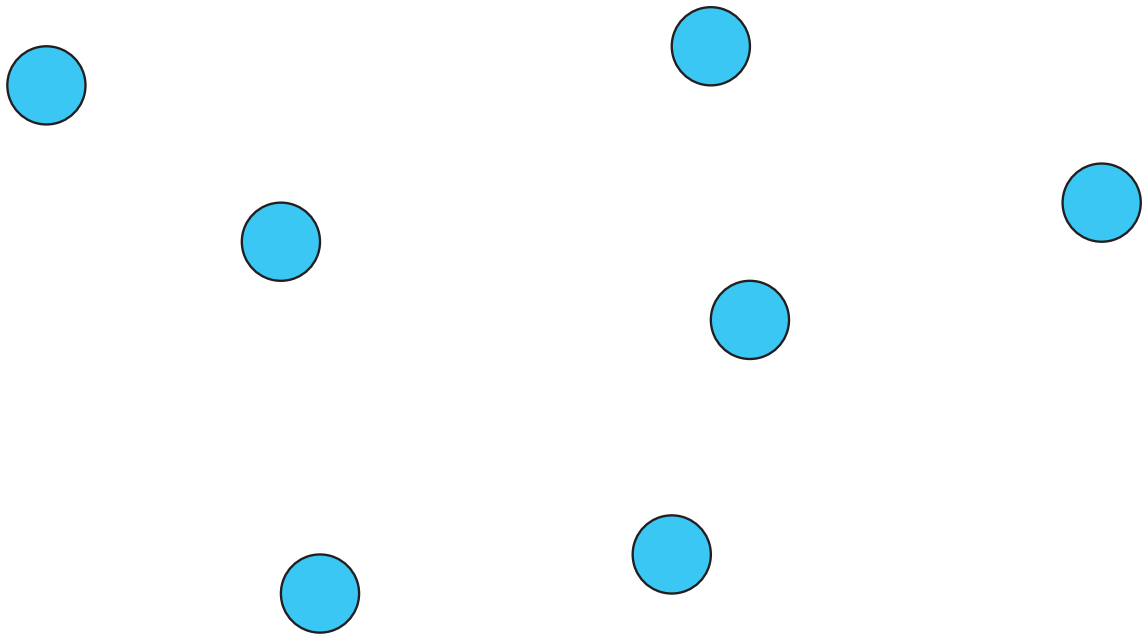
Moving accordingly

Anonymous robots sensing and moving in the plane



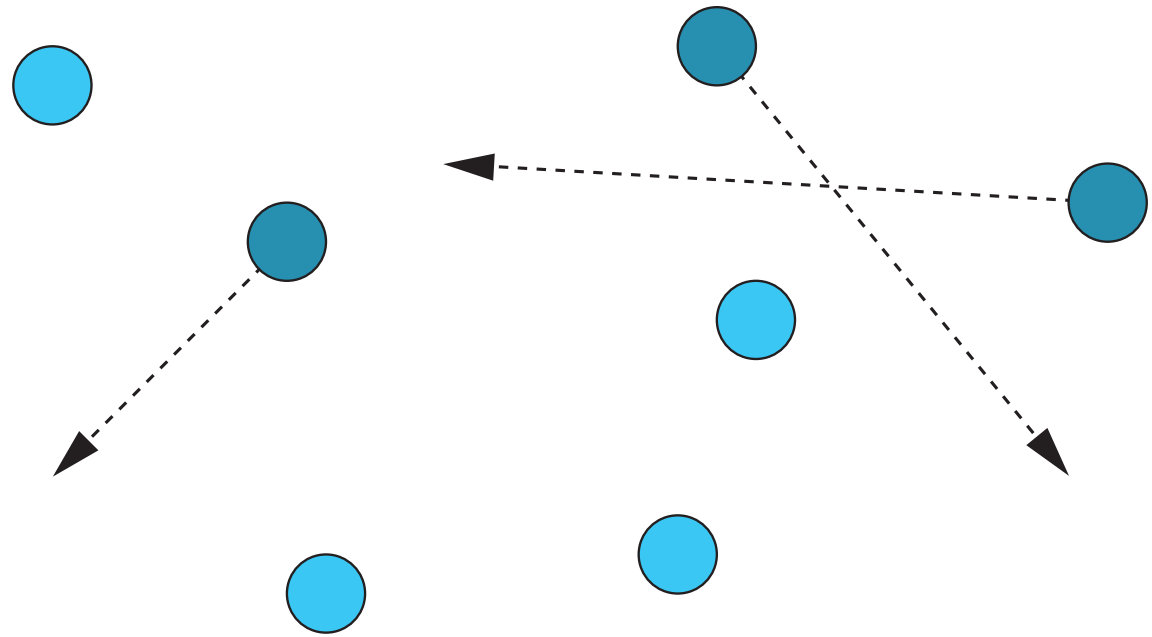
At different turns, different sets of robots may be activated

Anonymous robots sensing and moving in the plane



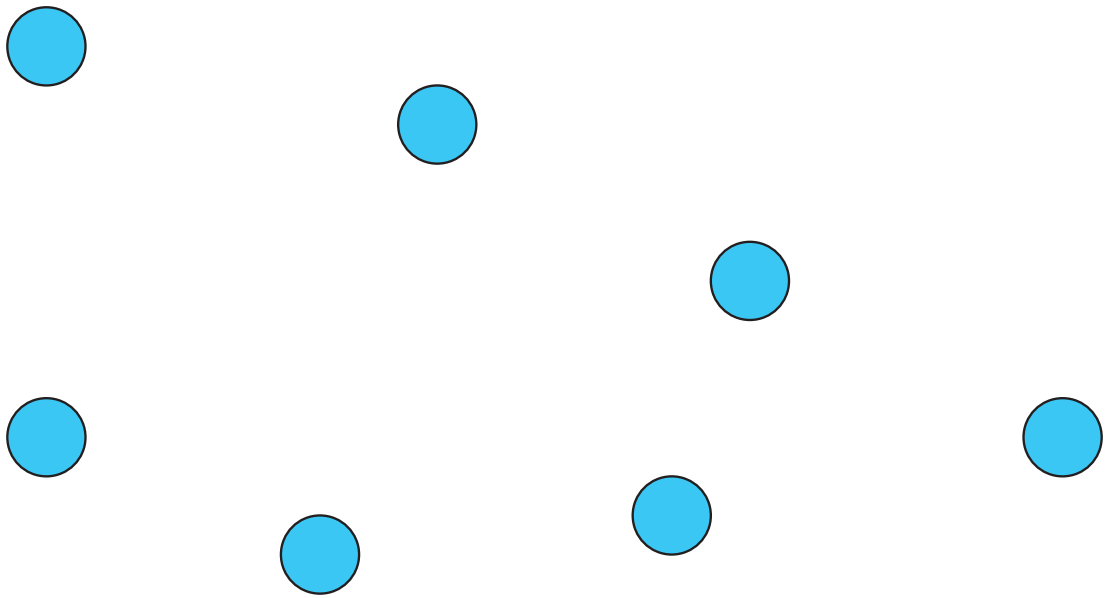
At different turns, different sets of robots may be activated

Anonymous robots sensing and moving in the plane



Fairness: each robot is activated infinitely many times

Anonymous robots sensing and moving in the plane



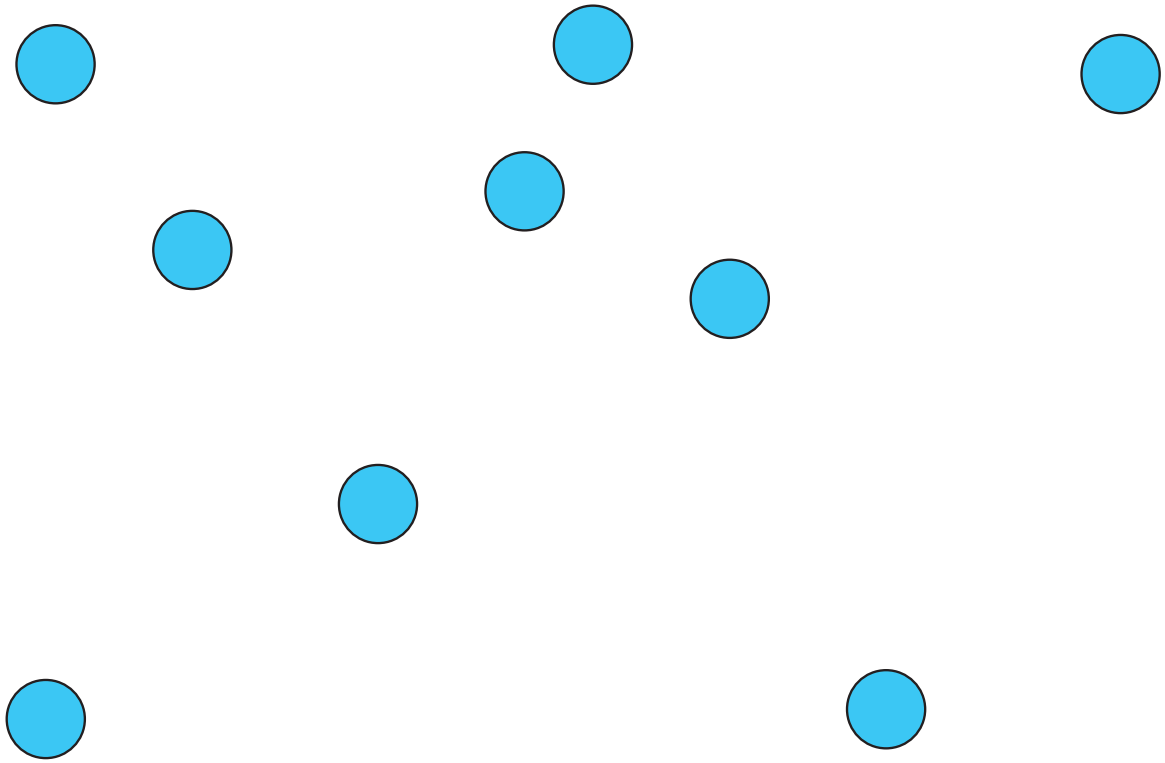
Fairness: each robot is activated infinitely many times

Anonymous robots sensing and moving in the plane

Robots are:

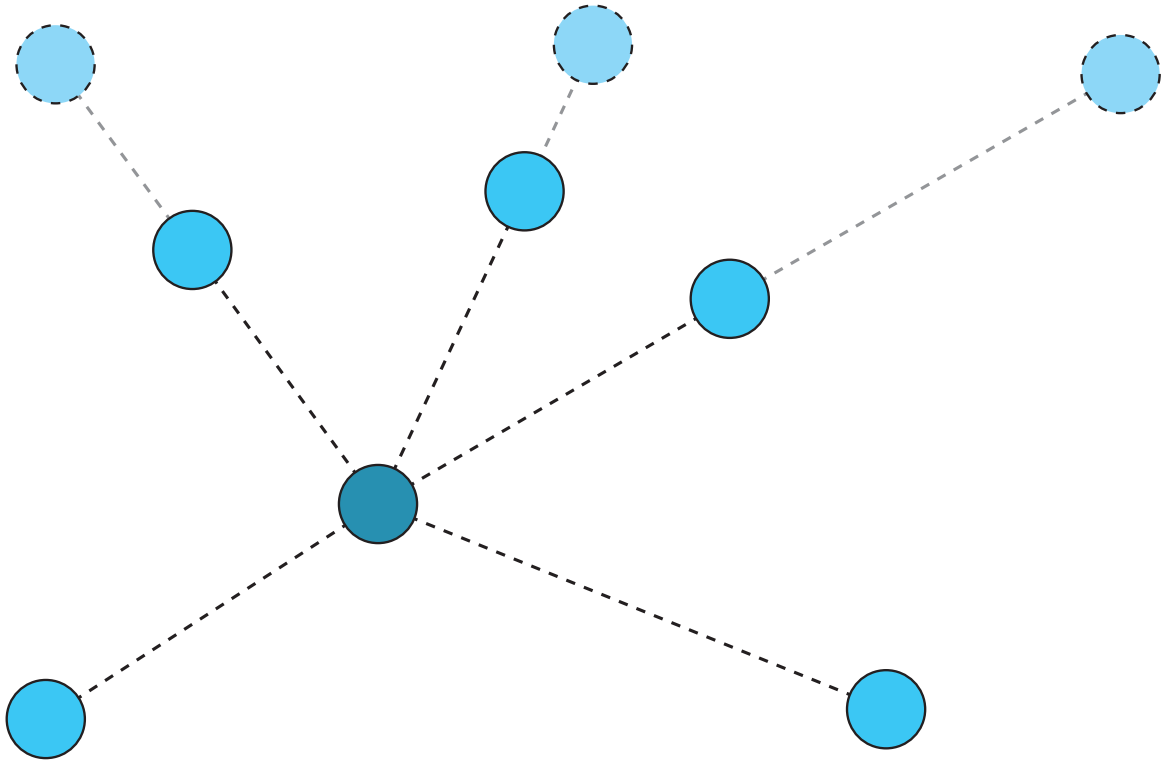
- **Dimensionless** (robots are modeled as geometric points)
- **Anonymous** (no unique identifiers)
- **Homogeneous** (the same algorithm is executed by all robots)
- **Autonomous** (no centralized control)
- **Oblivious** (no memory of past events)
- **Silent** (no explicit way of communicating)
- **Semi-synchronous** (turns are synchronized, but not all robots are active at each turn)
- **Disoriented** (robots do not share a common reference frame, and a robot's reference frame may change from turn to turn)
  - No common unit distance
  - No common compass
  - No common notion of clockwise direction

Robots obstructing other robots



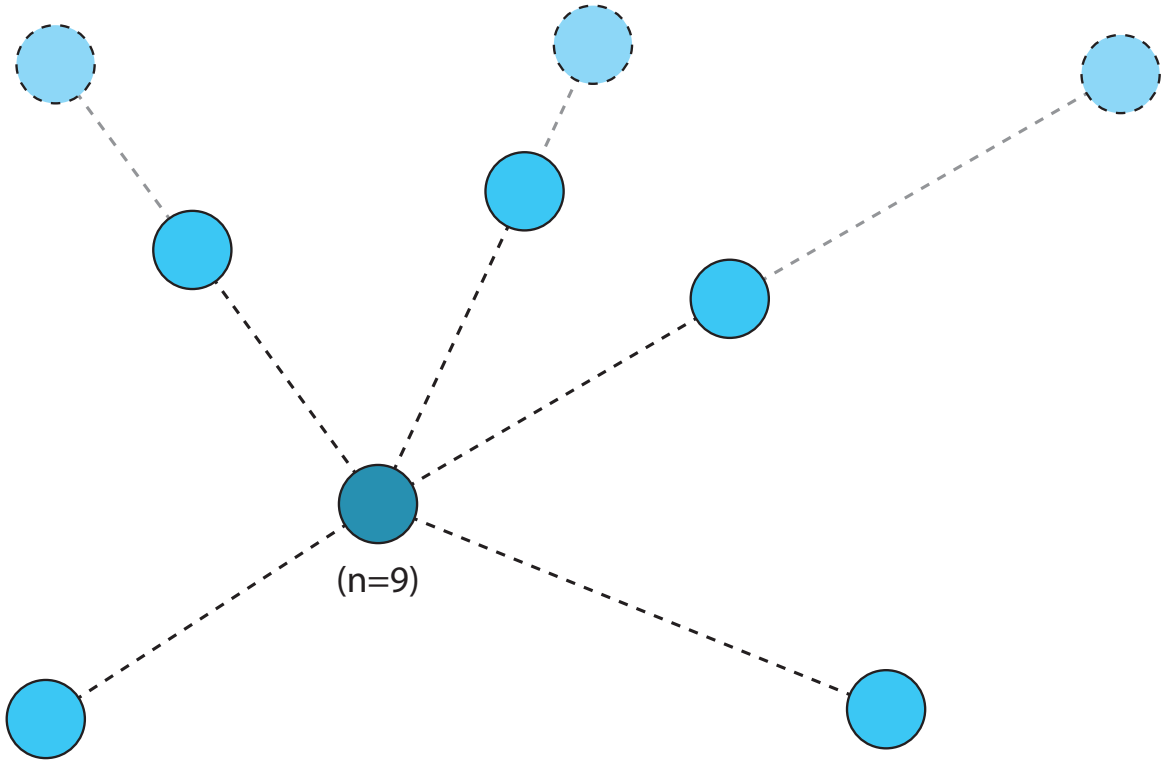
Robots do not see through other robots

Robots obstructing other robots



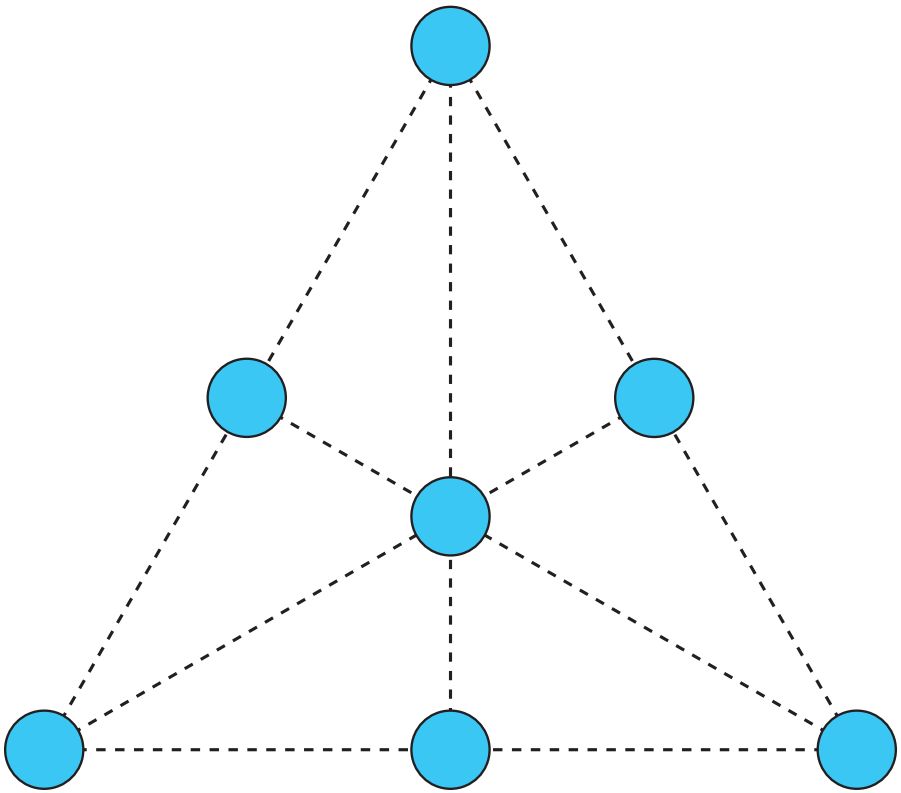
Robots do <u>not</u> see through other robots

Robots obstructing other robots



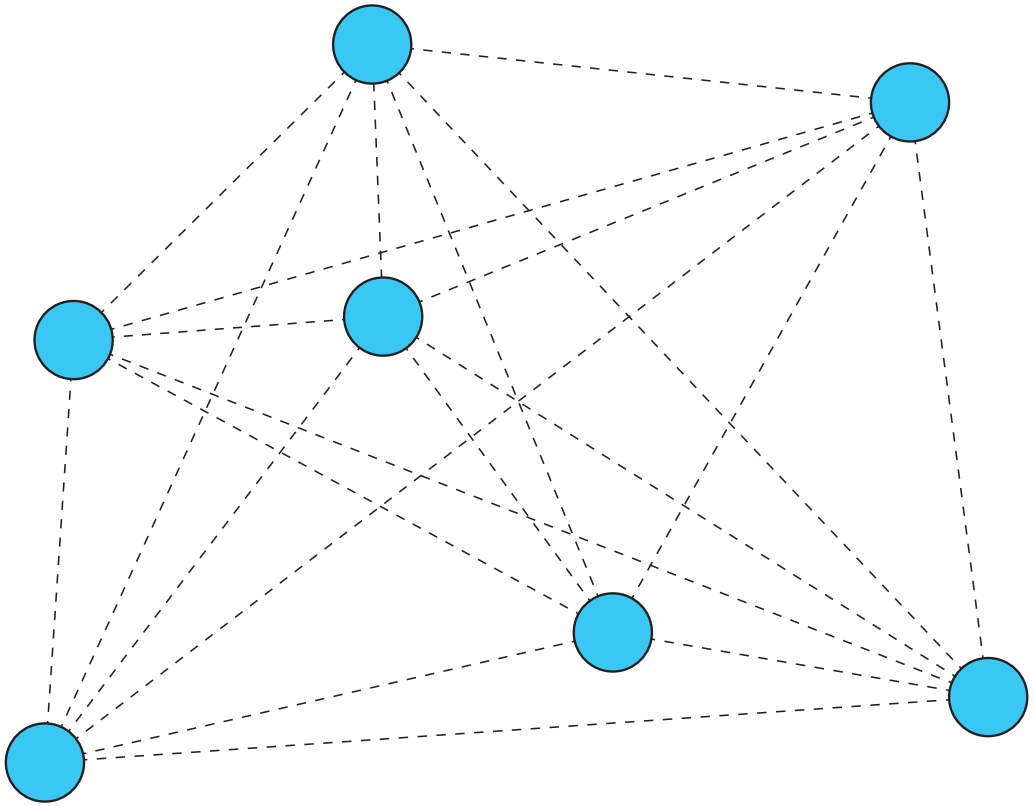
The total number of robots in the swarm, n, is known to the robots

The Mutual Visibility problem



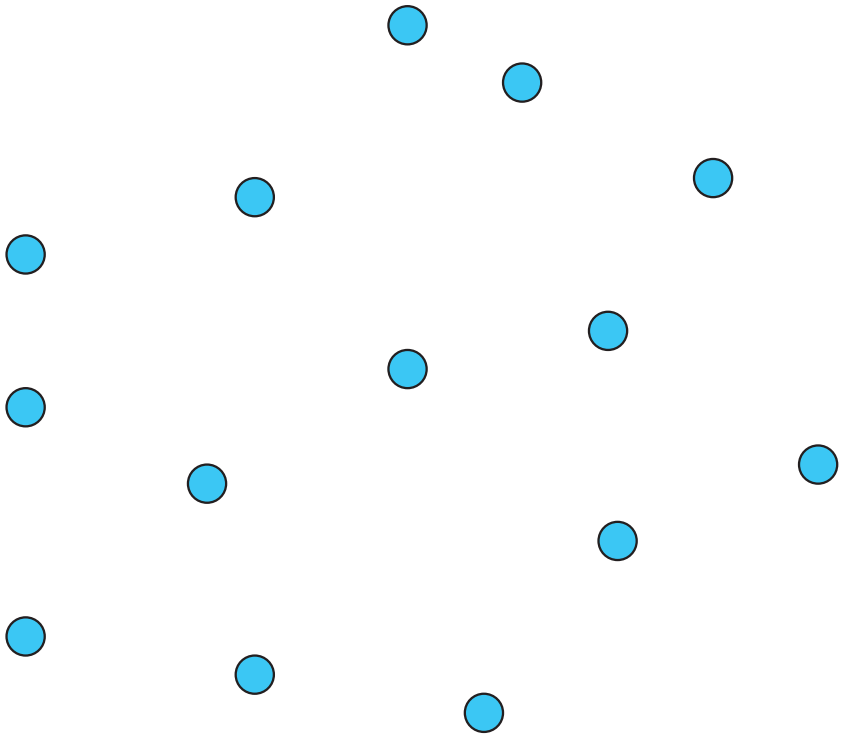
Goal: reach a configuration in which no three robots are collinear

The Mutual Visibility problem

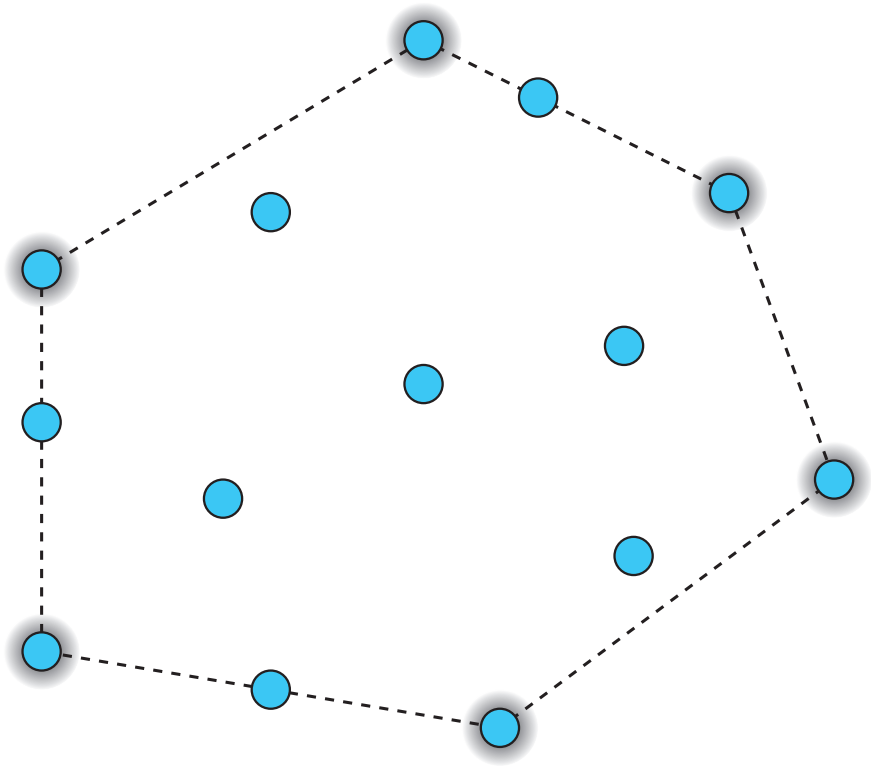


Goal: reach a configuration in which no three robots are collinear

Strategy for Mutual Visibility

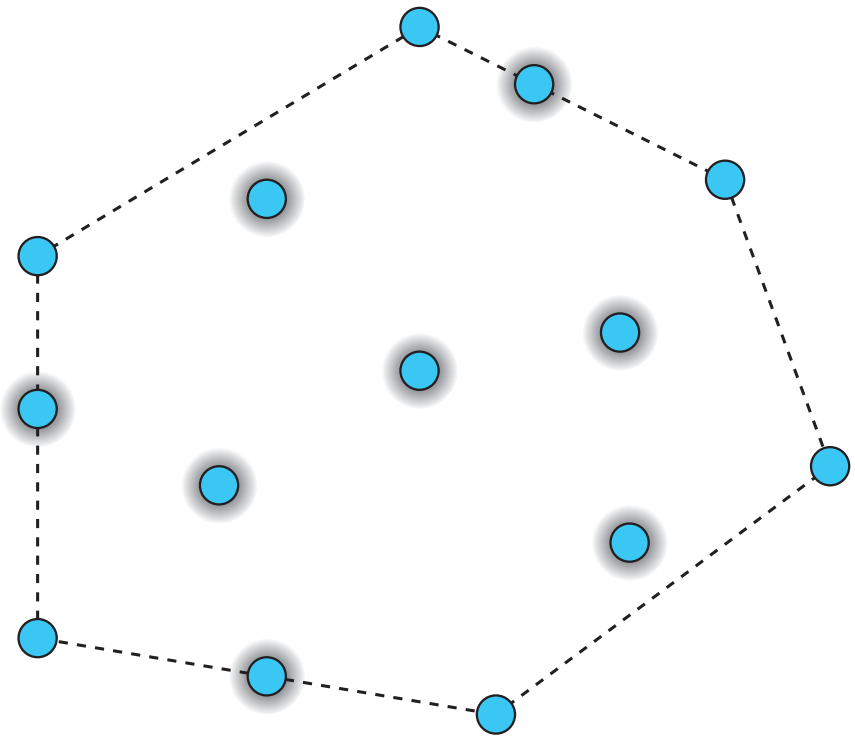


Strategy for Mutual Visibility



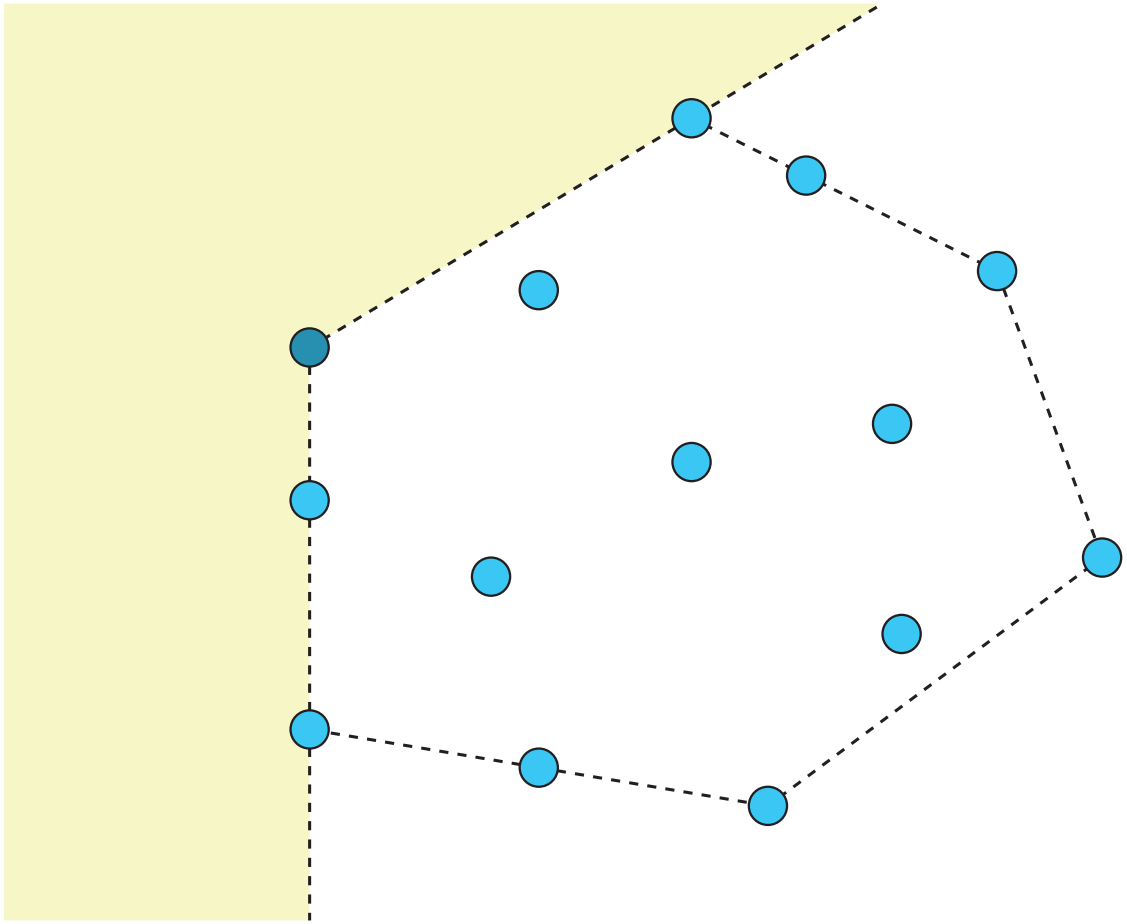
The vertices of the convex hull are the *external* robots

Strategy for Mutual Visibility



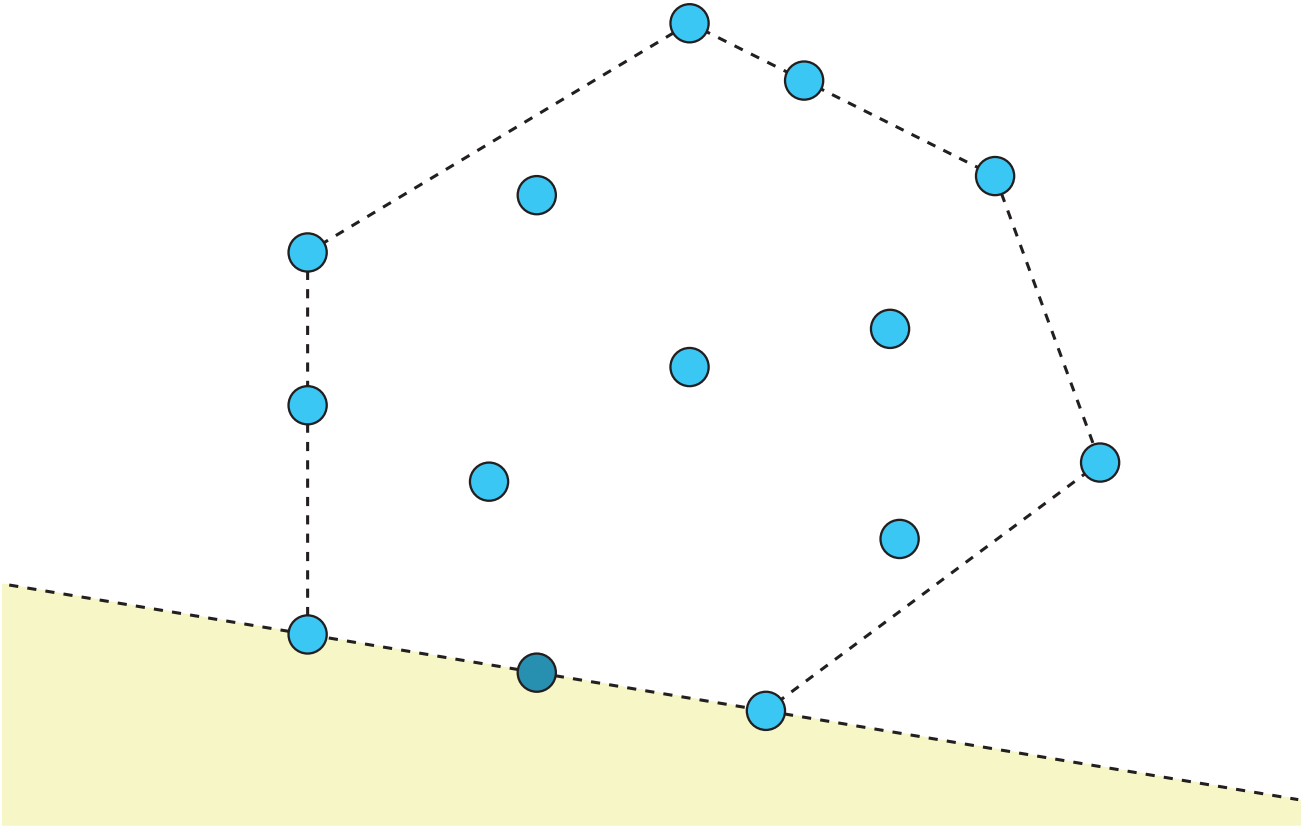
The other robots are *internal*

Strategy for Mutual Visibility



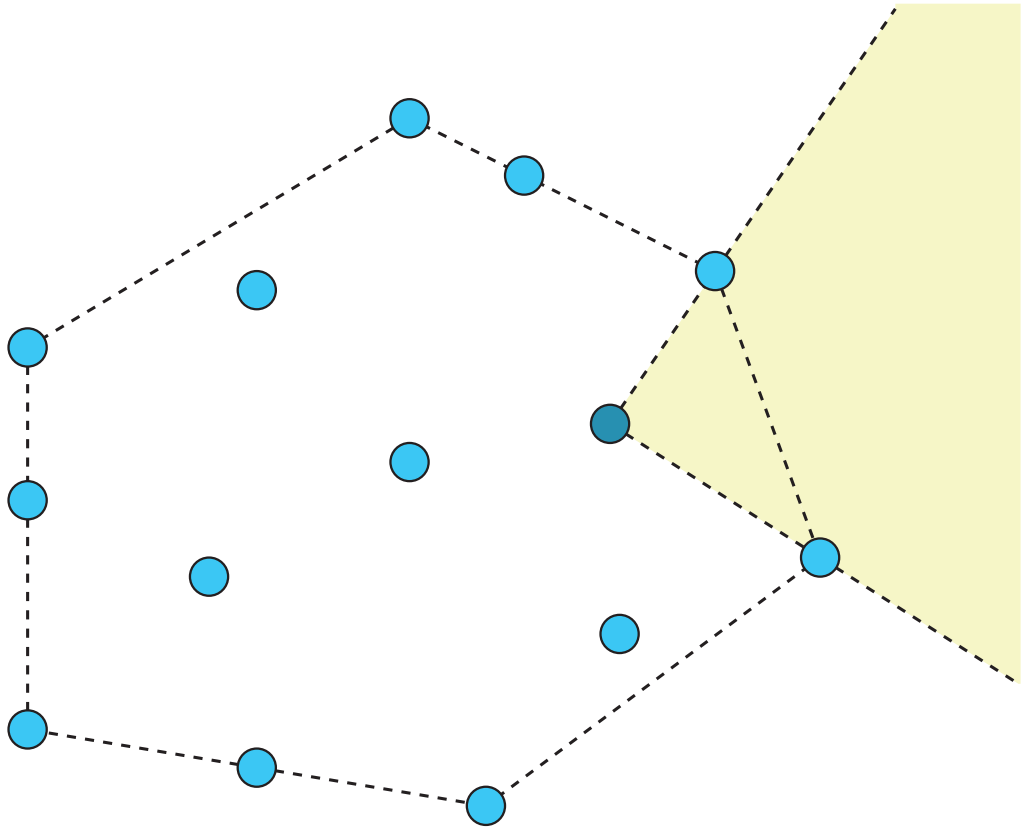
Observation: a robot is external iff it sees an empty reflex angle

Strategy for Mutual Visibility



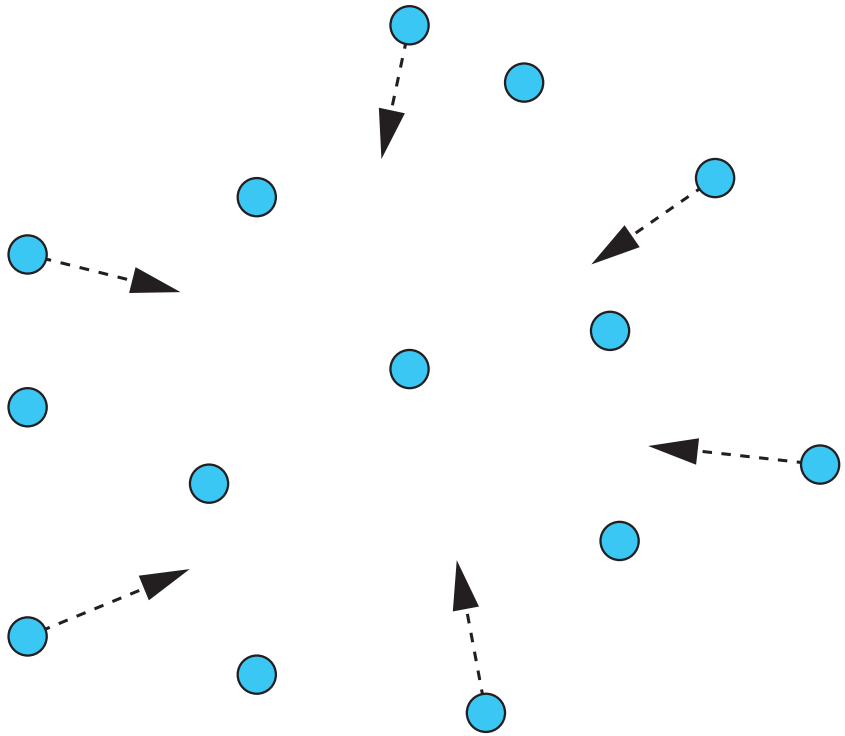
Observation: a robot is external iff it sees an empty reflex angle

Strategy for Mutual Visibility



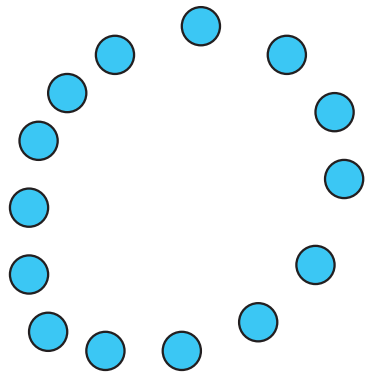
Observation: a robot is external iff it sees an empty reflex angle

Strategy for Mutual Visibility



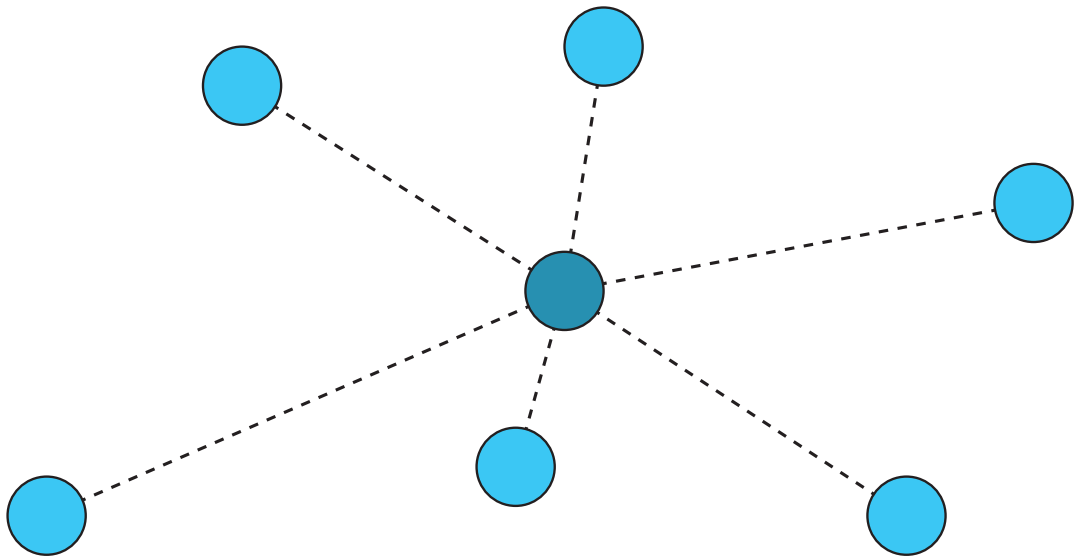
Strategy: external robots move inward, until all become external

Strategy for Mutual Visibility



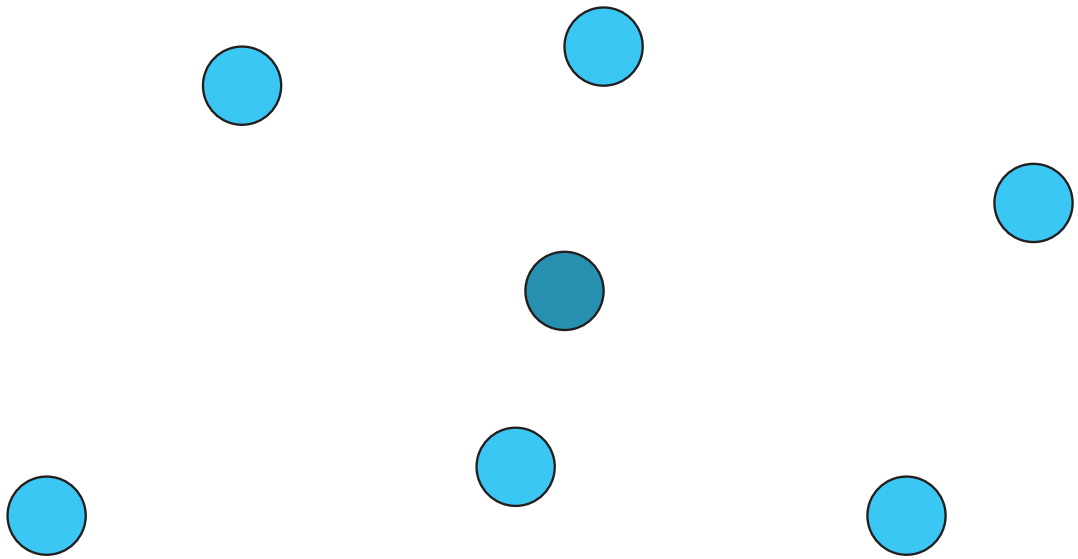
Strategy: external robots move inward, until all become external

Algorithm for Mutual Visibility



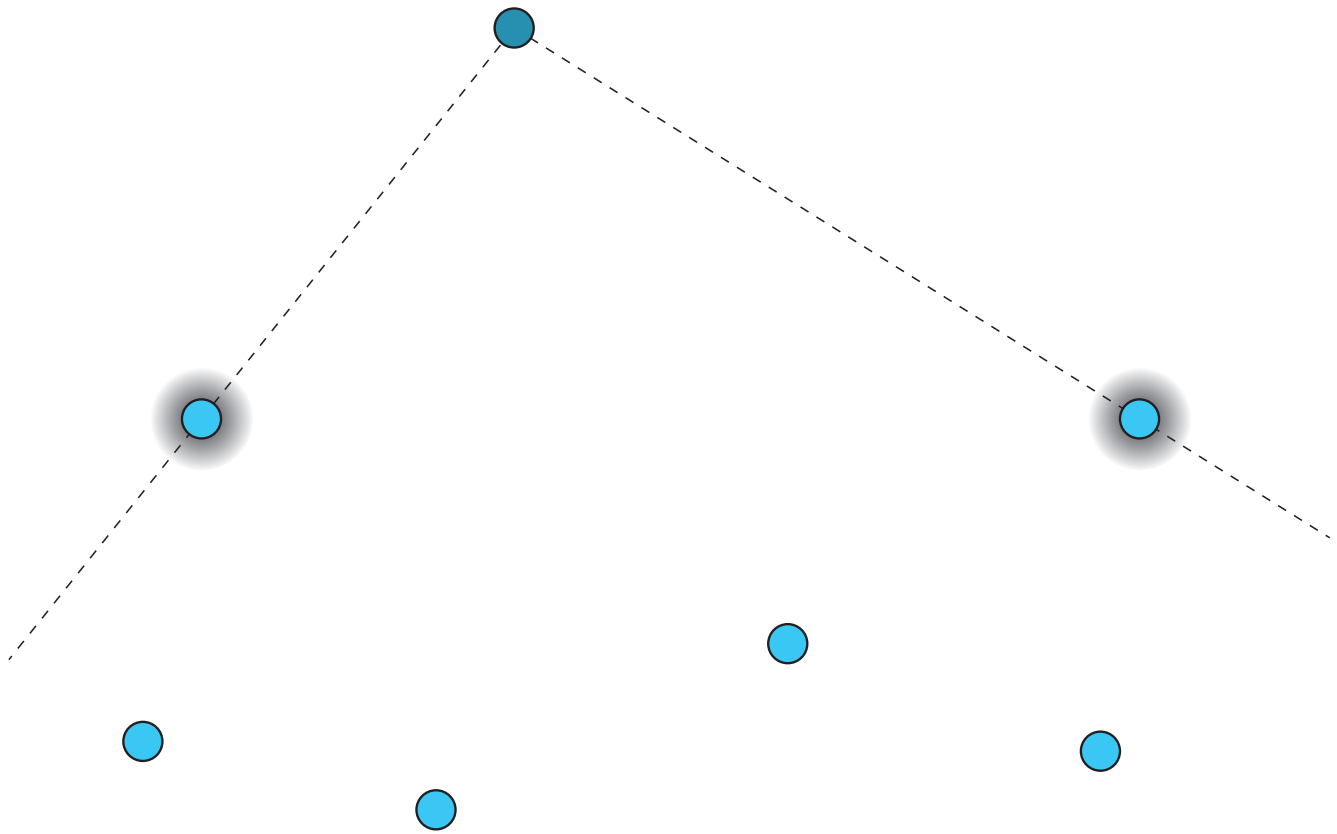
Get a snapshot of visible robots

Algorithm for Mutual Visibility



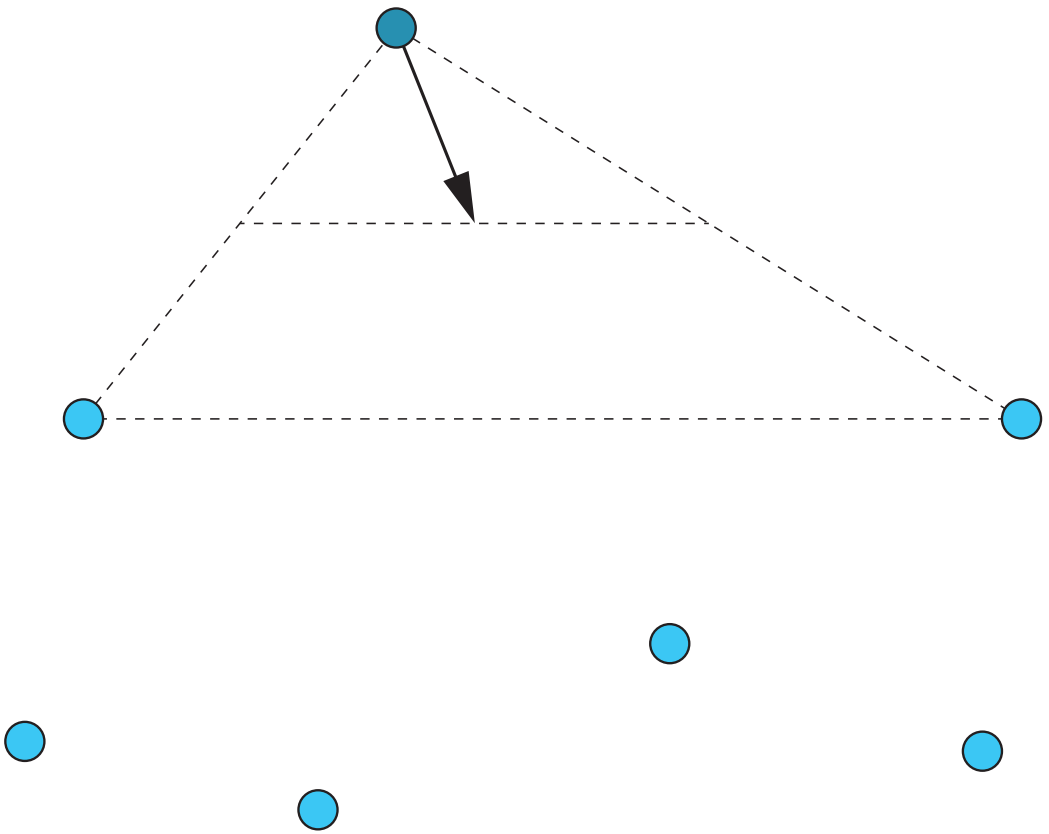
If you are internal, do not move

Algorithm for Mutual Visibility



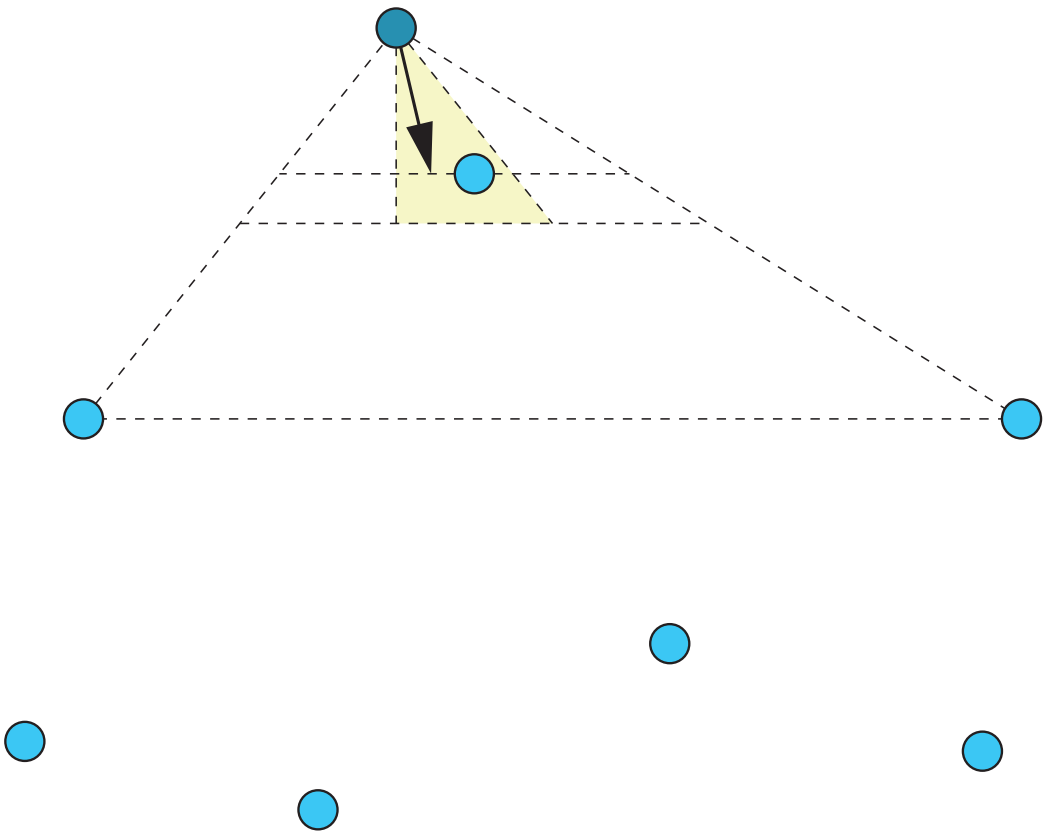
If you are external, locate your two neighbors on the convex hull

Algorithm for Mutual Visibility



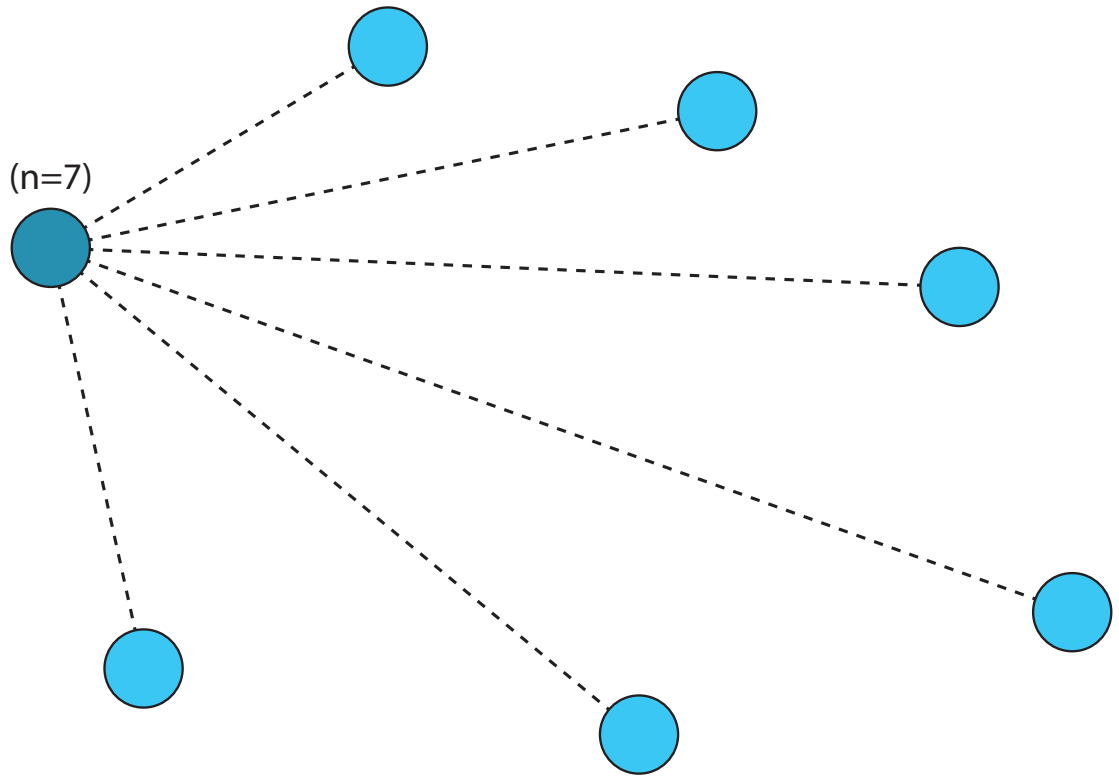
Default move

Algorithm for Mutual Visibility



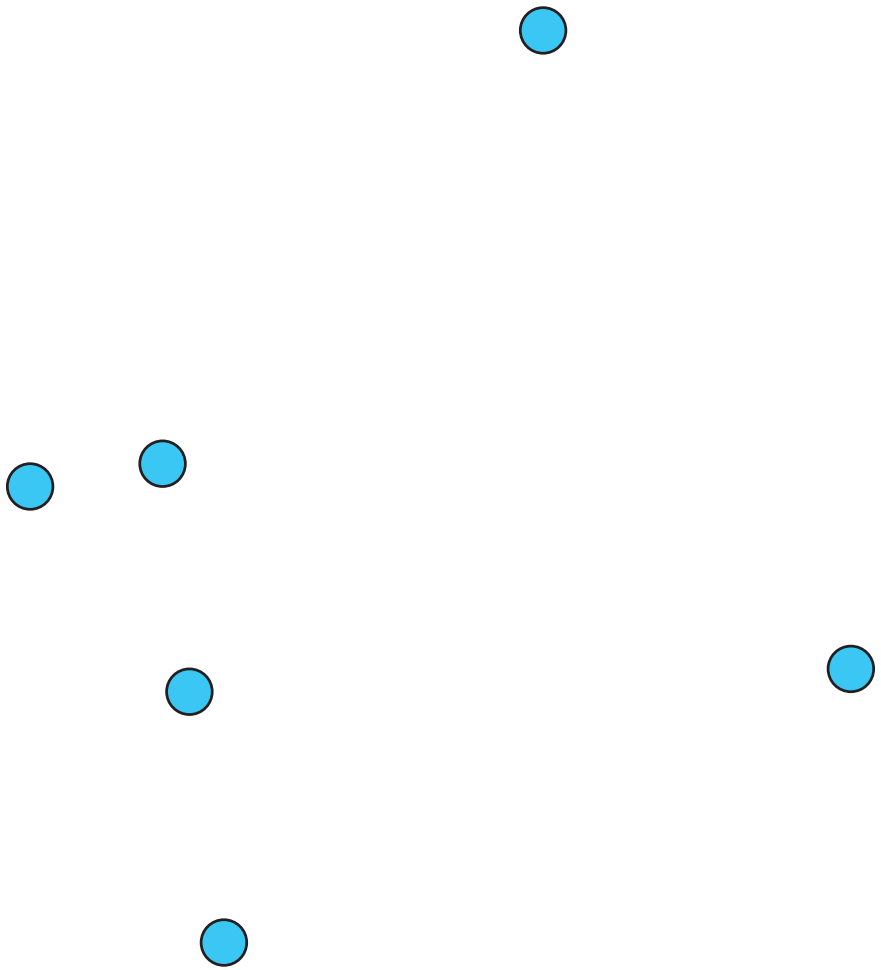
In general, do not pass internal robots, and stay in the yellow area

Algorithm for Mutual Visibility

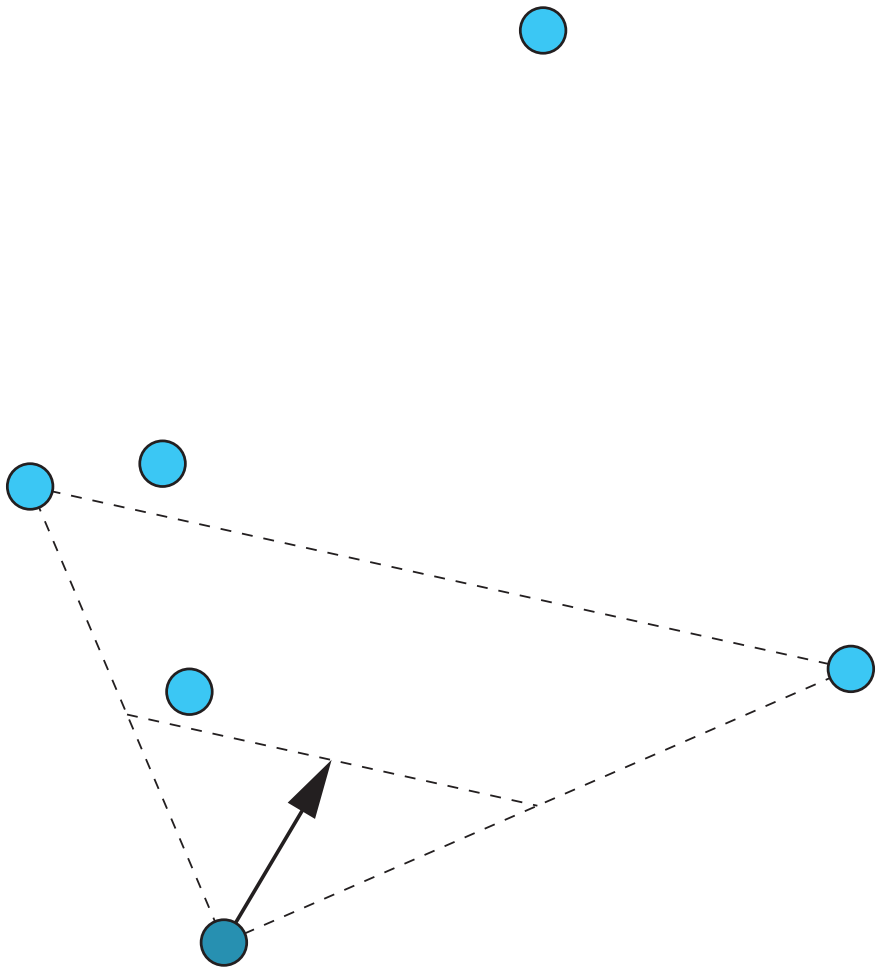


If you see n robots in a strictly convex position, terminate

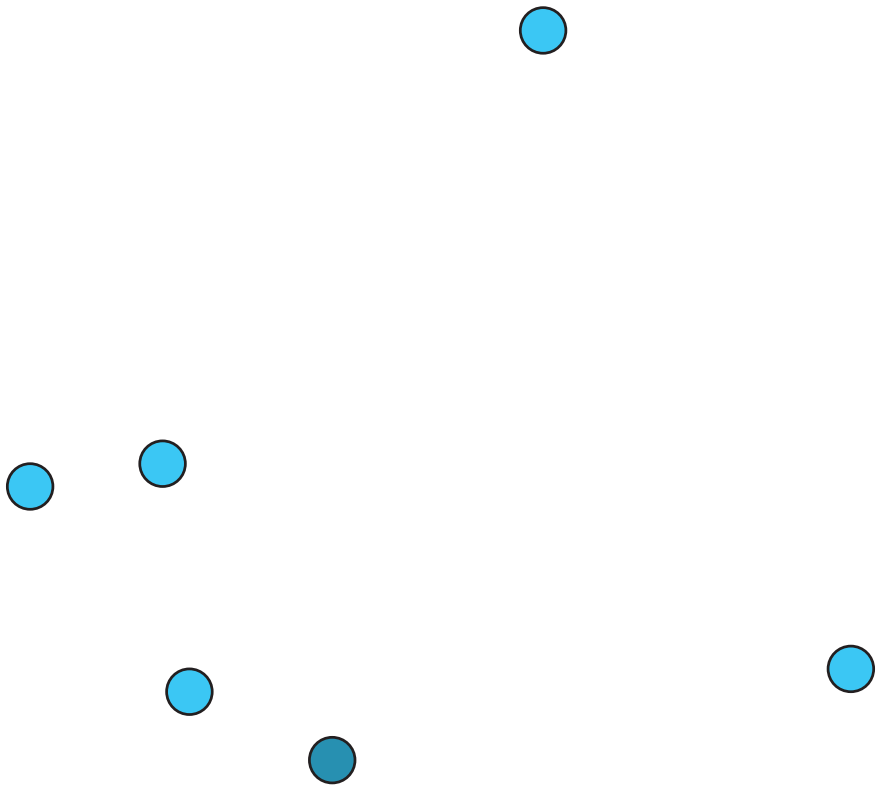
Executing the algorithm



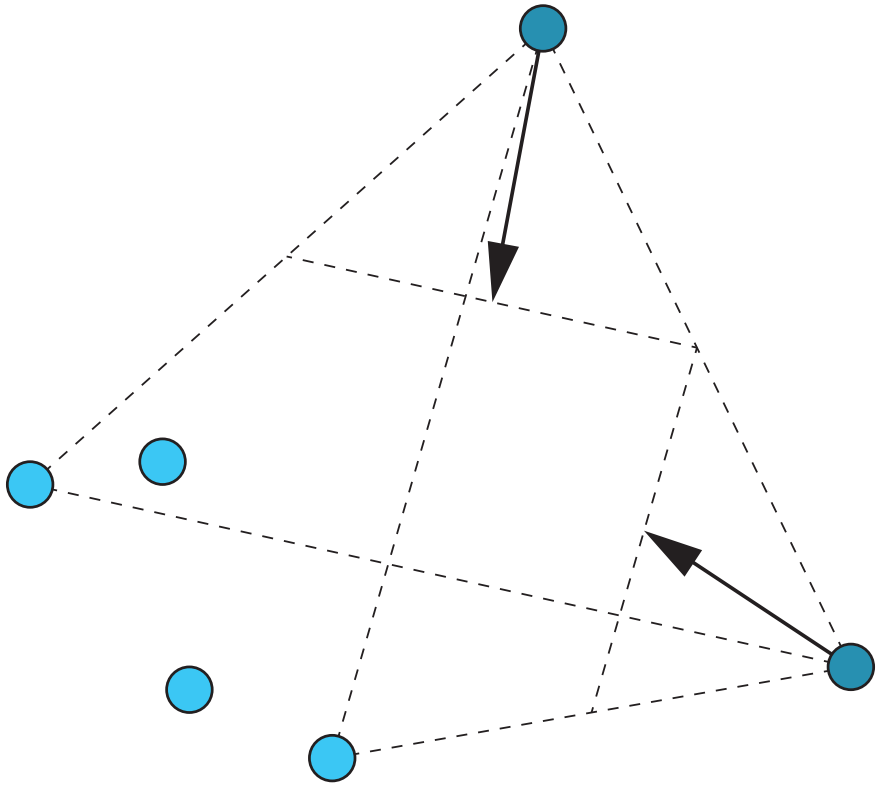
Executing the algorithm



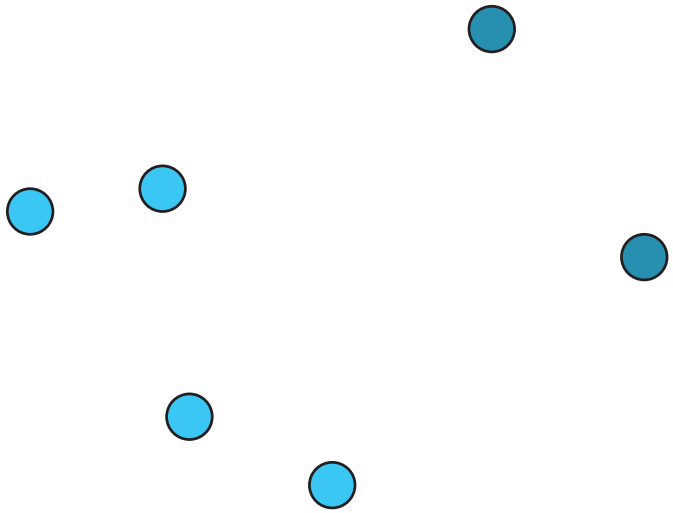
Executing the algorithm



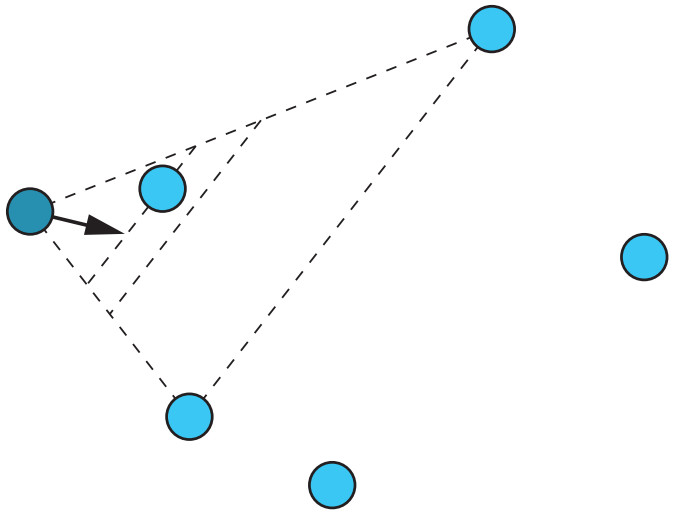
Executing the algorithm



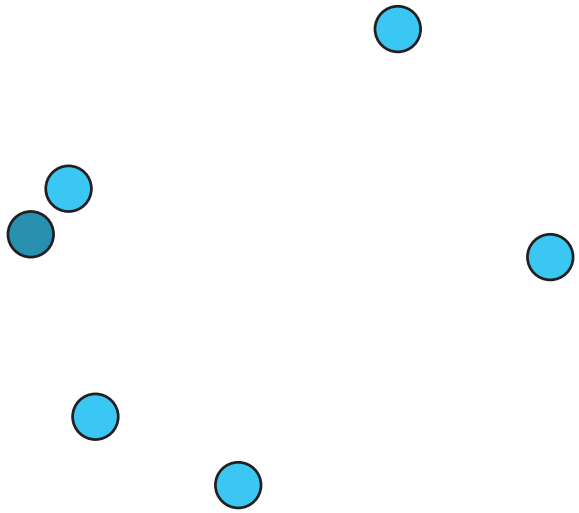
Executing the algorithm



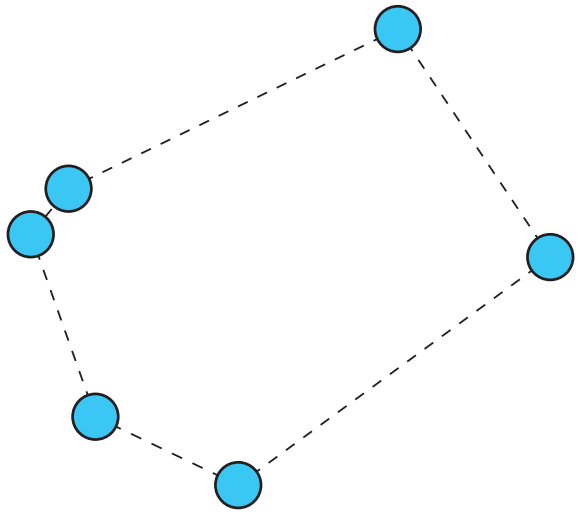
Executing the algorithm



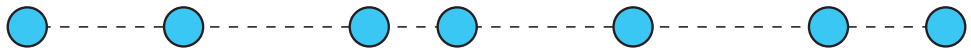
Executing the algorithm



Executing the algorithm

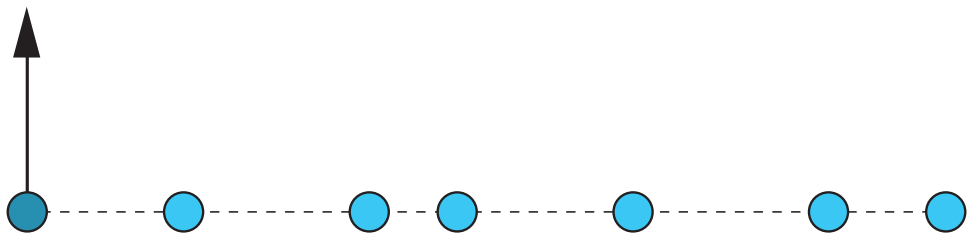


Exception: all robots collinear



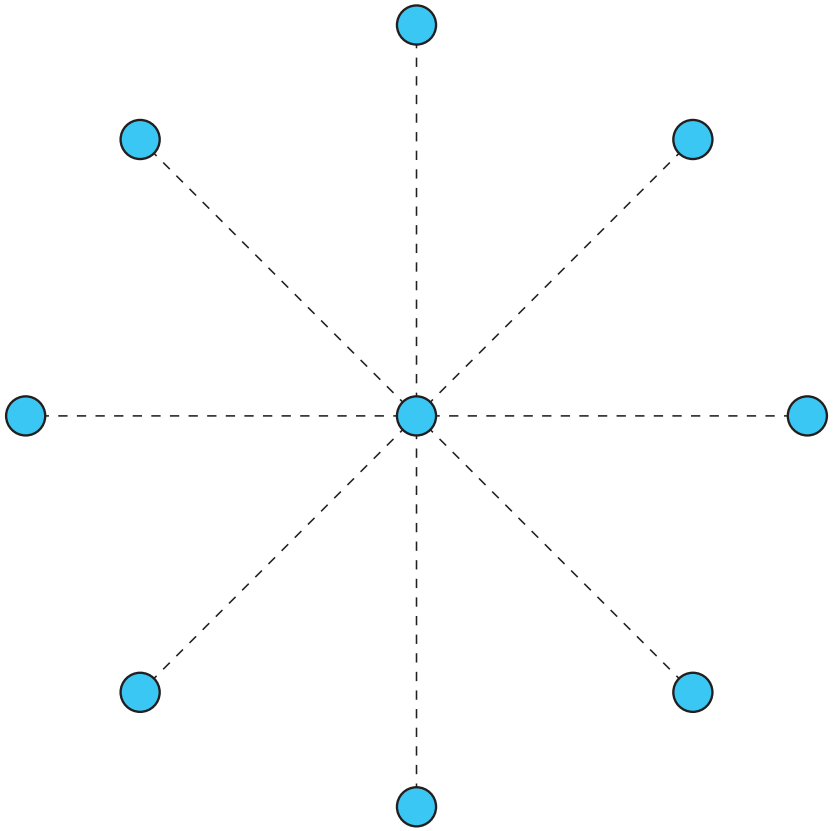
If all robots are collinear, the basic algorithm keeps them collinear

Algorithm for Mutual Visibility (cont.)



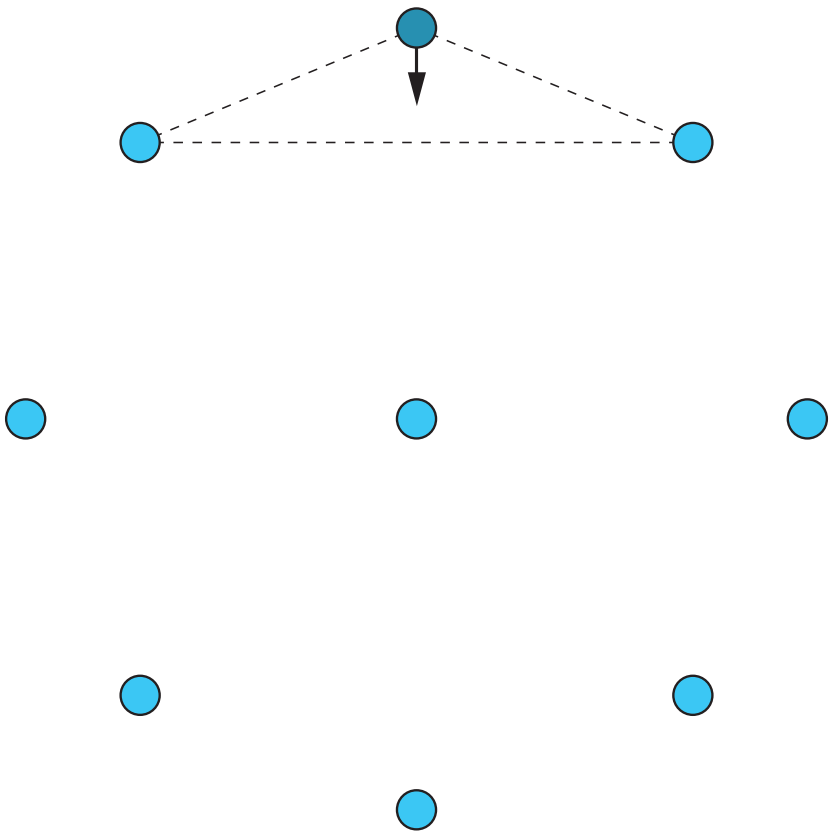
If all robots are collinear and you are an endpoint, abandon the line

Exception: unique internal robot



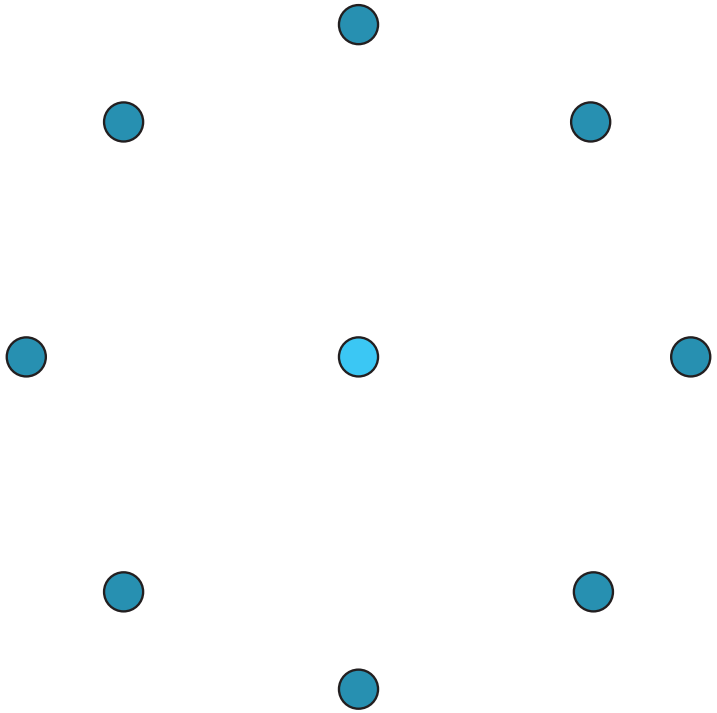
The algorithm may still fail if there is only one internal robot

Exception: unique internal robot



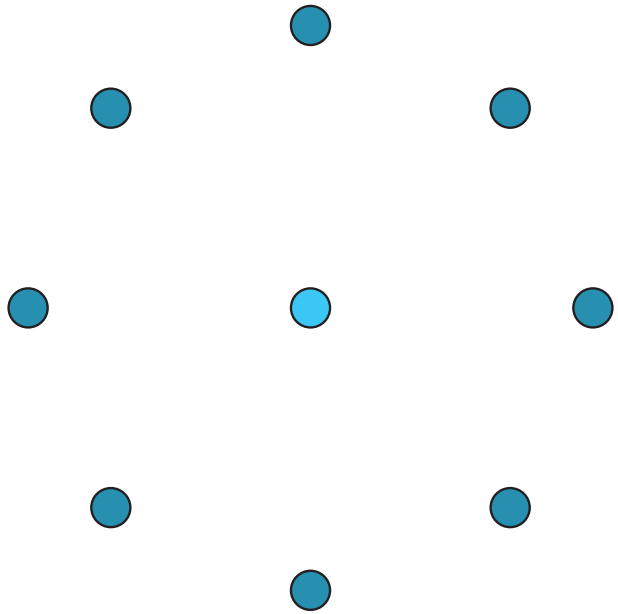
The algorithm may still fail if there is only one internal robot

Exception: unique internal robot



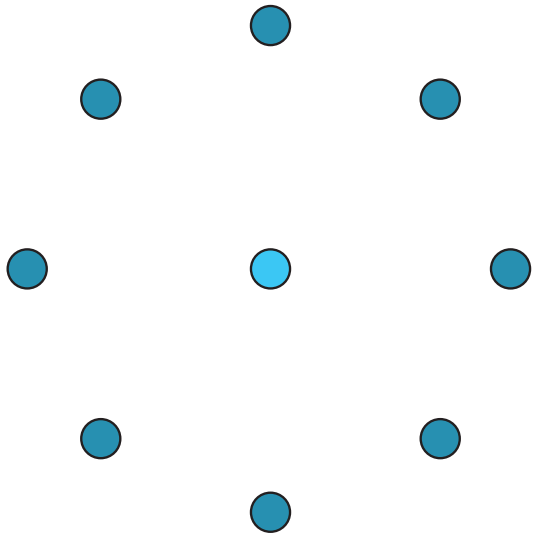
The algorithm may still fail if there is only one internal robot

Exception: unique internal robot



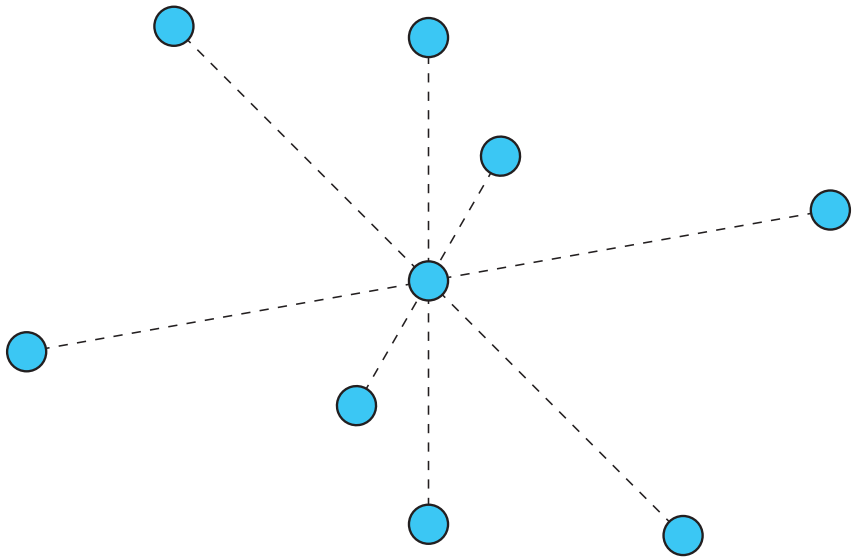
The algorithm may still fail if there is only one internal robot

Exception: unique internal robot



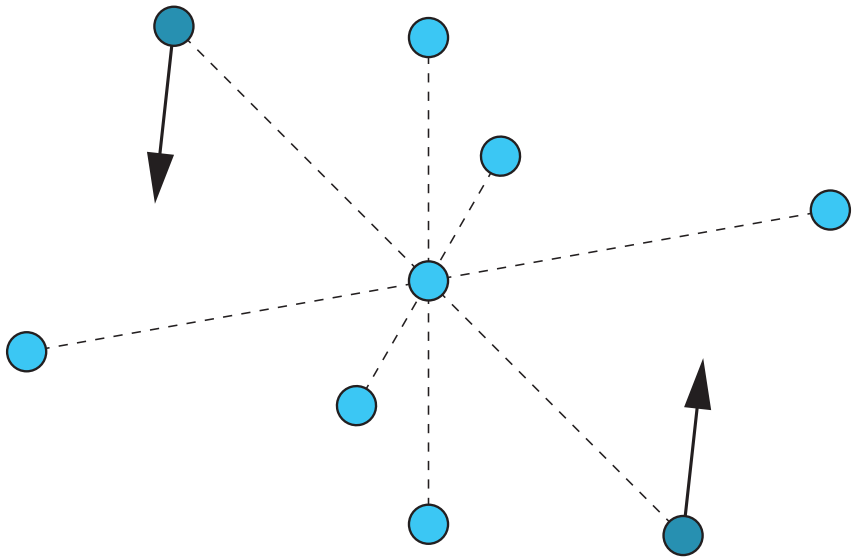
The algorithm may still fail if there is only one internal robot

Exception: unique internal robot



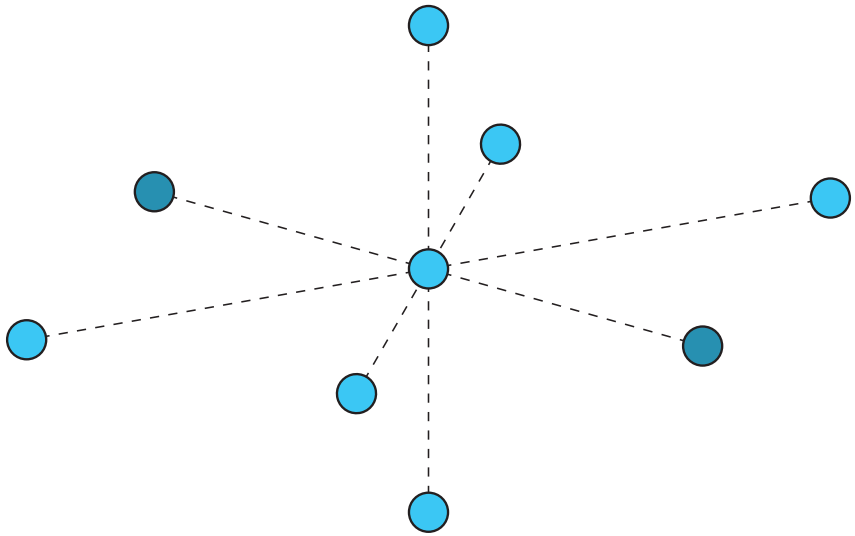
No algorithm for MUTUAL VISIBILITY moves external robots only

Exception: unique internal robot



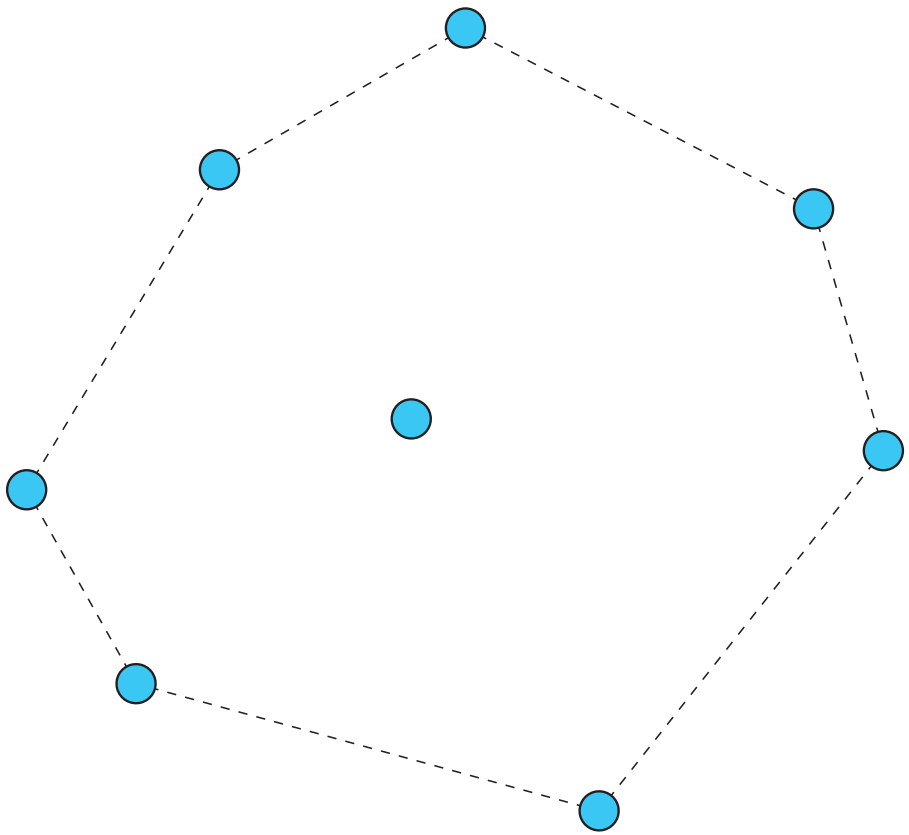
No algorithm for MUTUAL VISIBILITY moves external robots only

Exception: unique internal robot



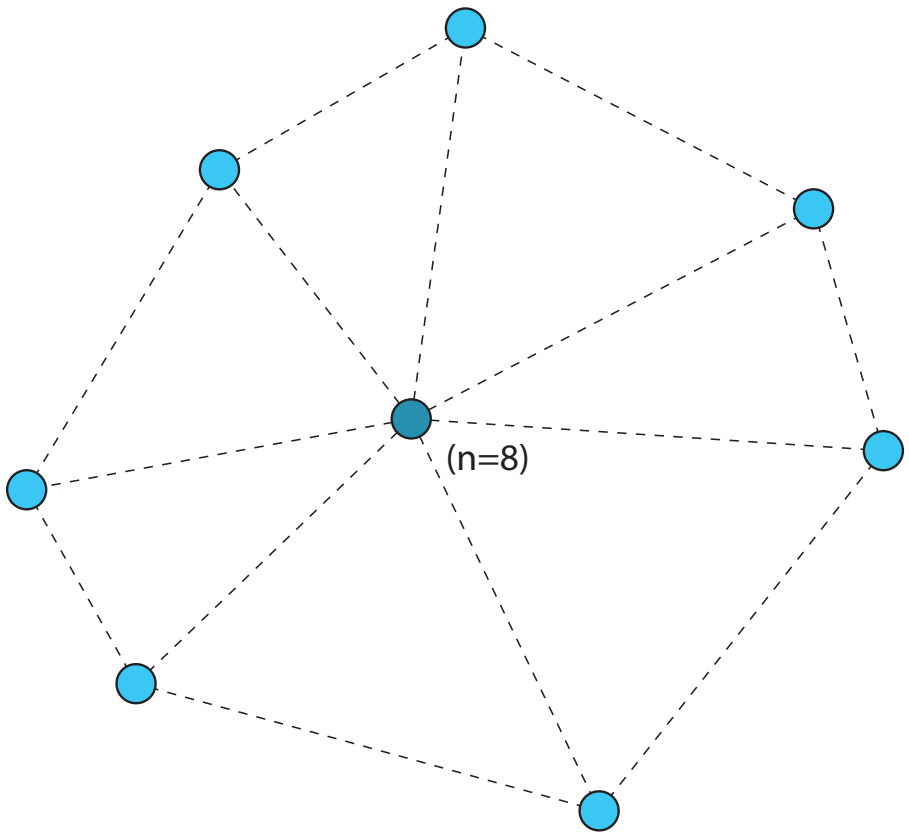
No algorithm for MUTUAL VISIBILITY moves external robots only

Algorithm for Mutual Visibility (cont.)



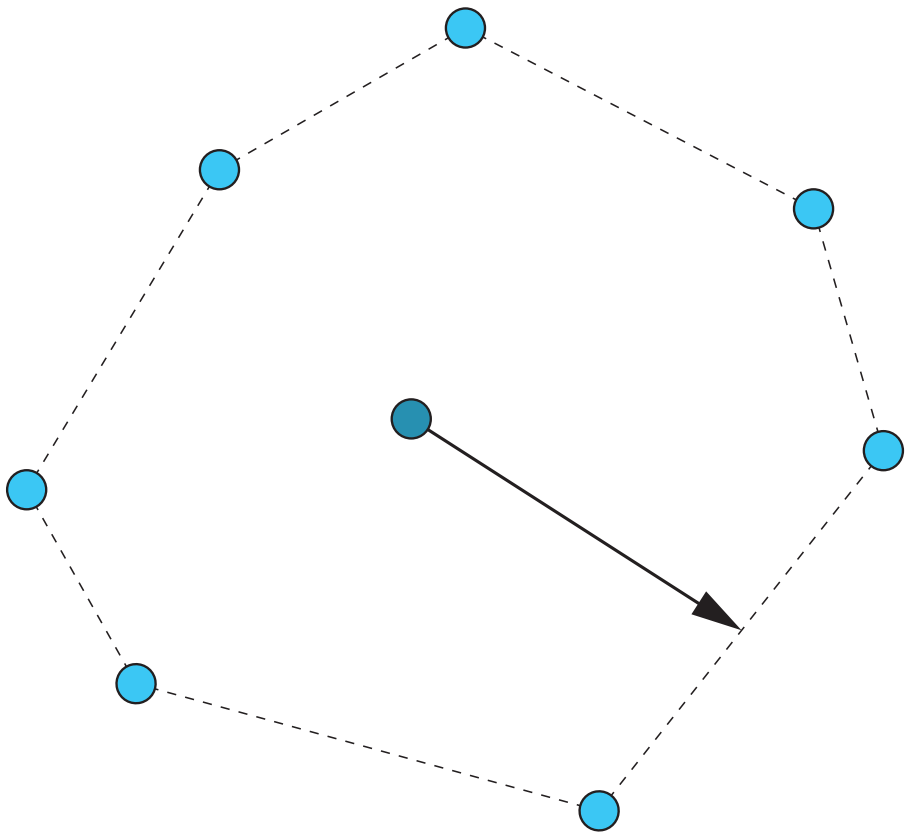
Since n is known, a robot can tell if it is the only internal robot

Algorithm for Mutual Visibility (cont.)



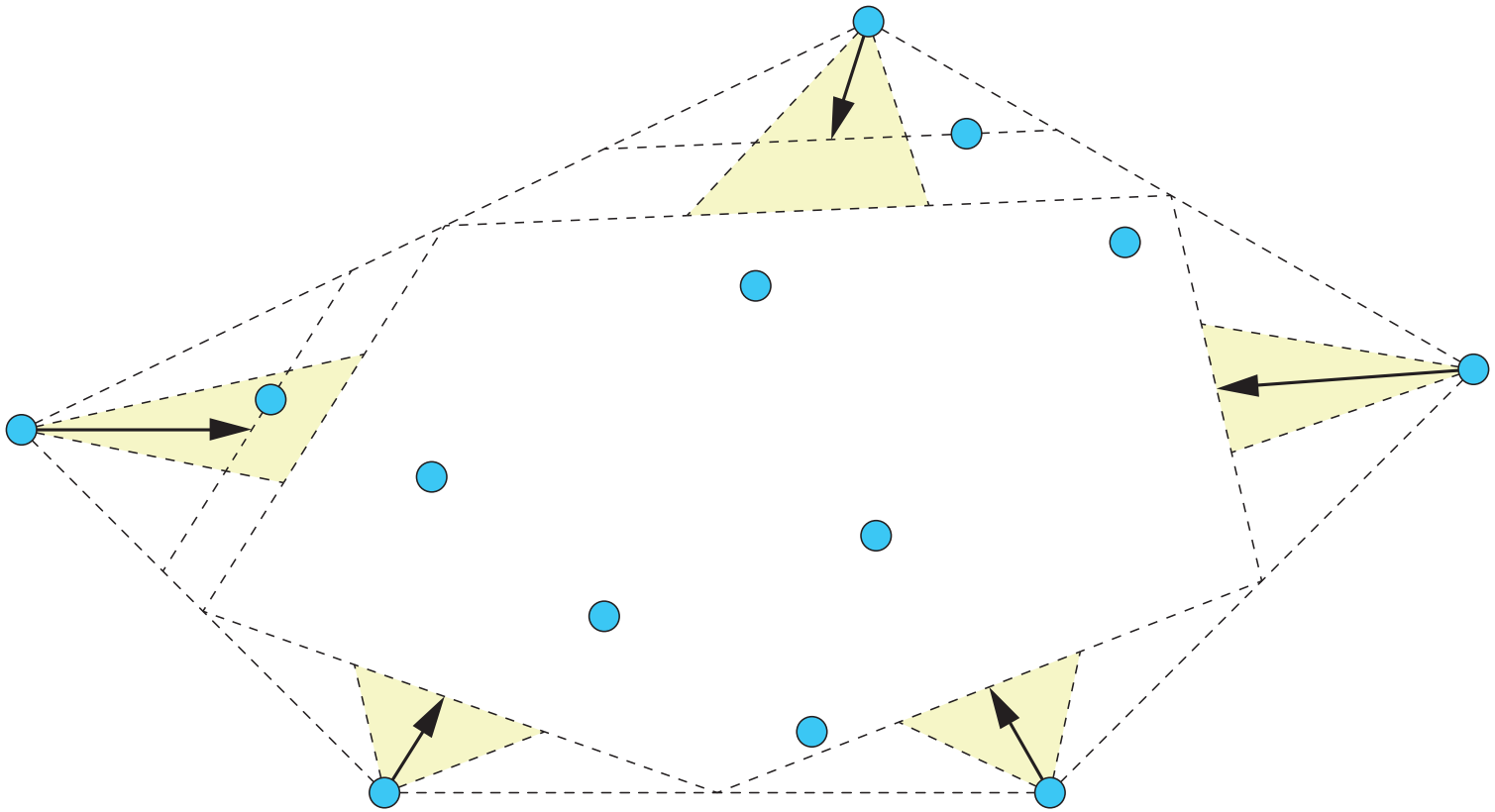
Since n is known, a robot can tell if it is the only internal robot

Algorithm for Mutual Visibility (cont.)



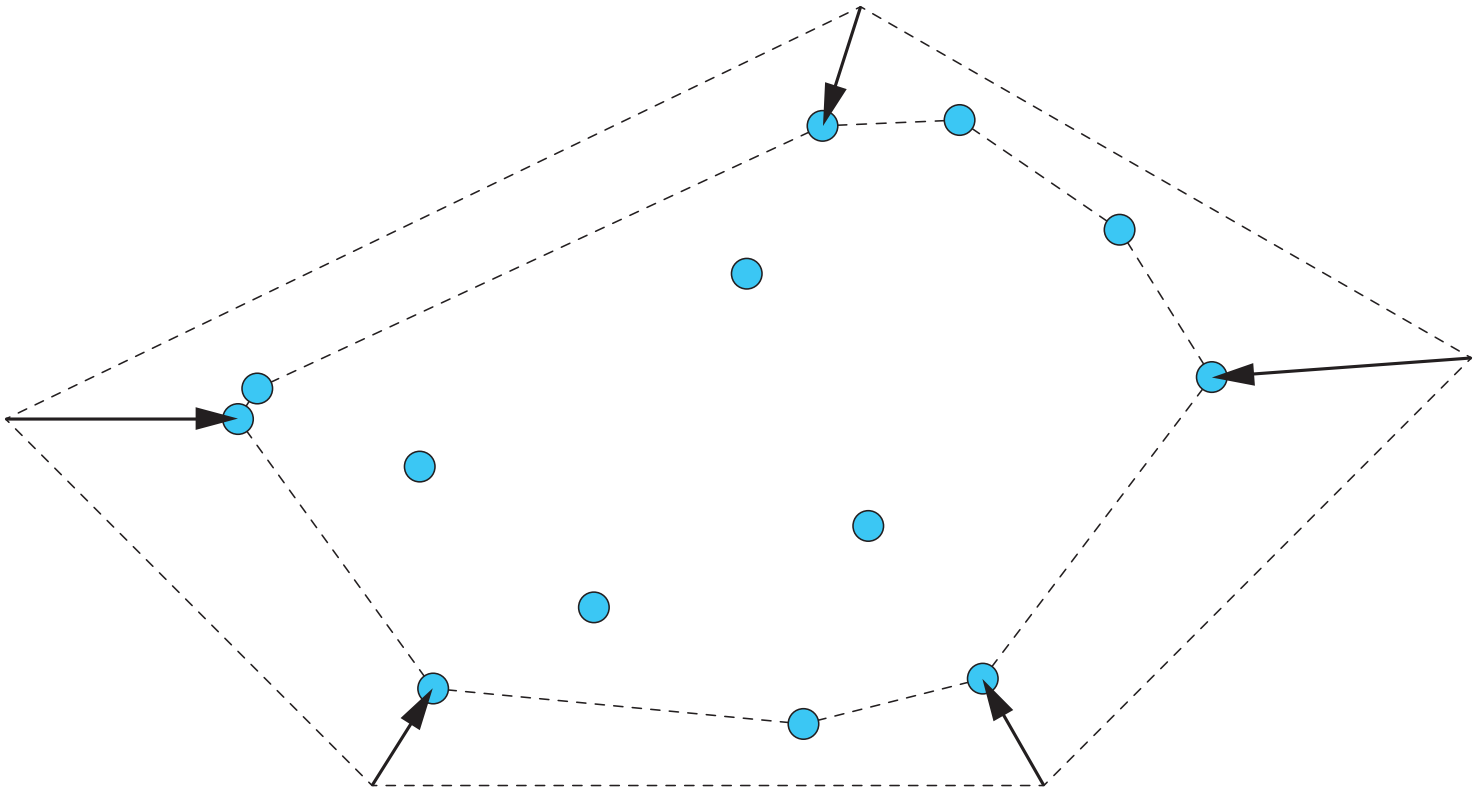
In this case, move to the midpoint of any edge of the convex hull

Correctness of the algorithm



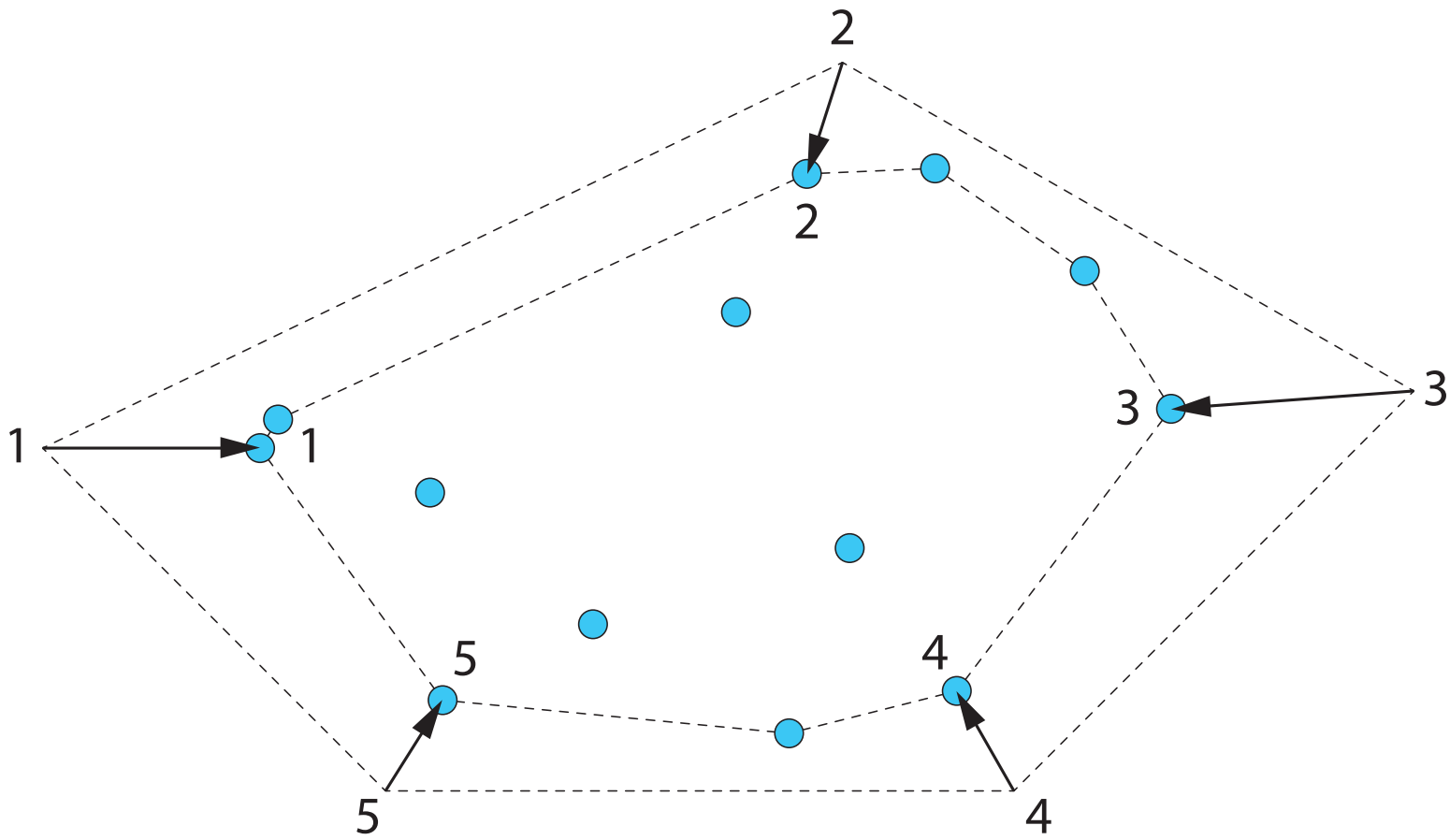
Moving robots remain in the yellow areas, hence they do not collide

Correctness of the algorithm



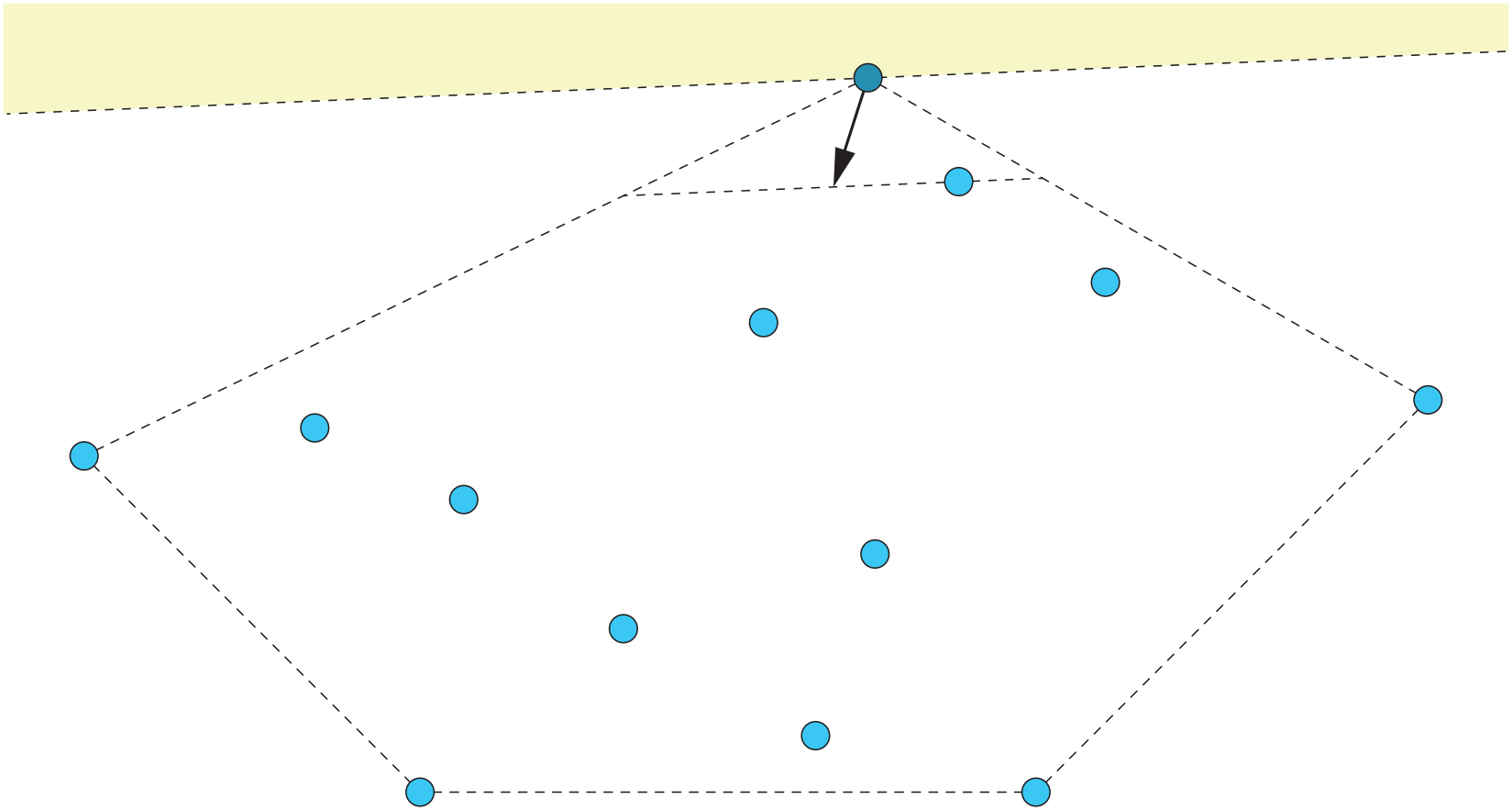
Robots move only within the convex hull, which keeps shrinking

Correctness of the algorithm



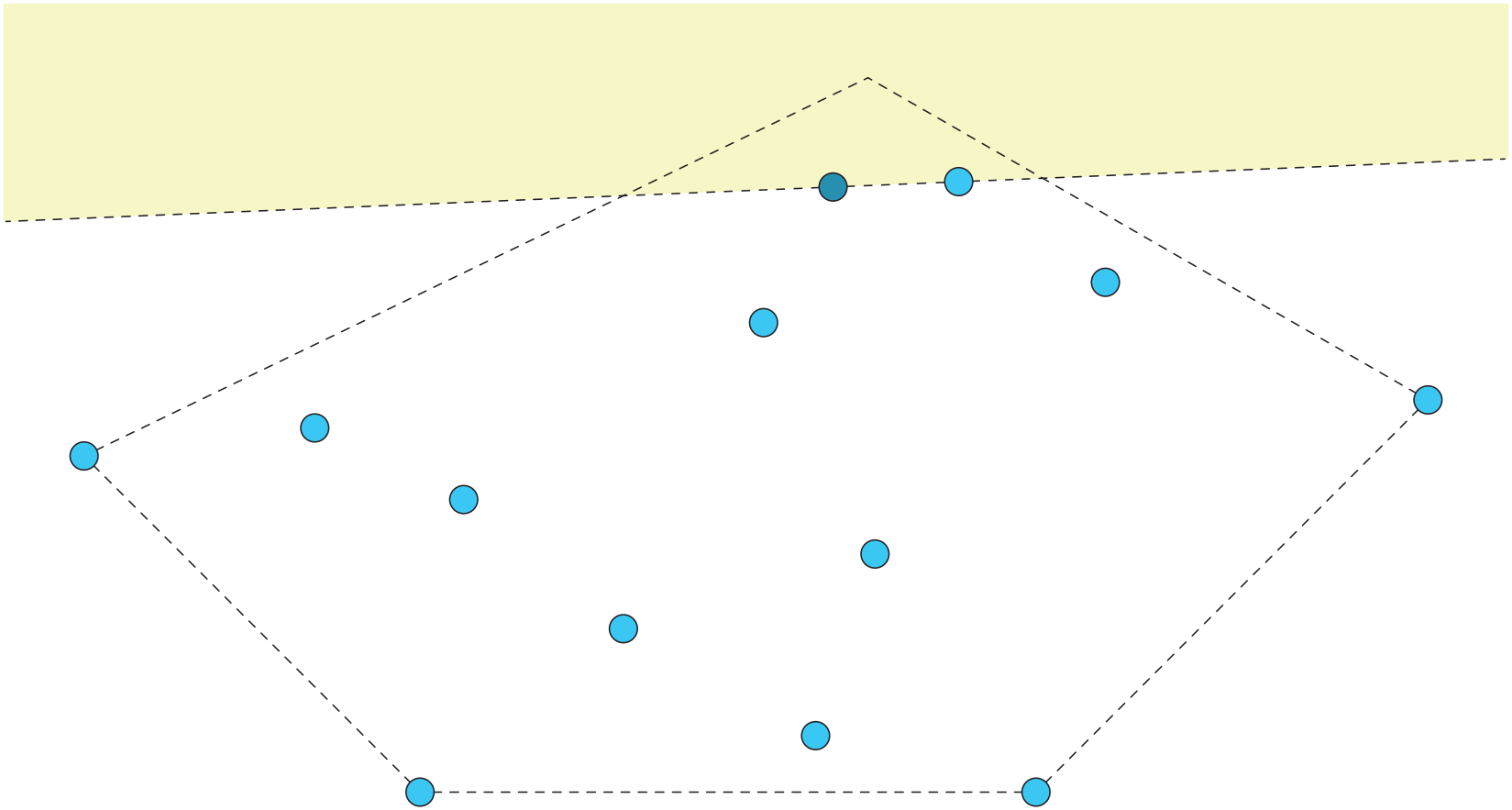
External robots preserve their relative order around the convex hull

Correctness of the algorithm



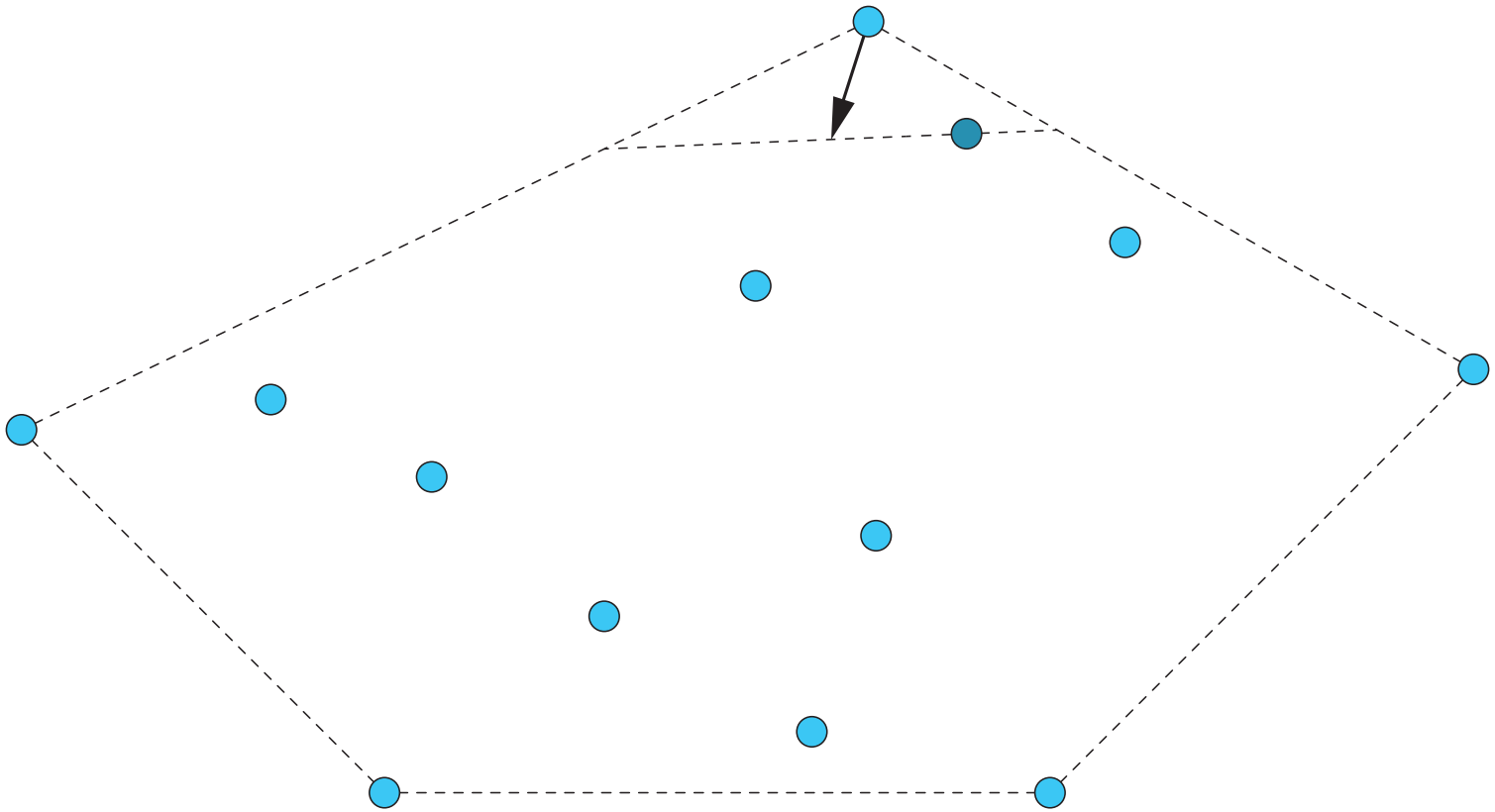
External robots remain external

Correctness of the algorithm



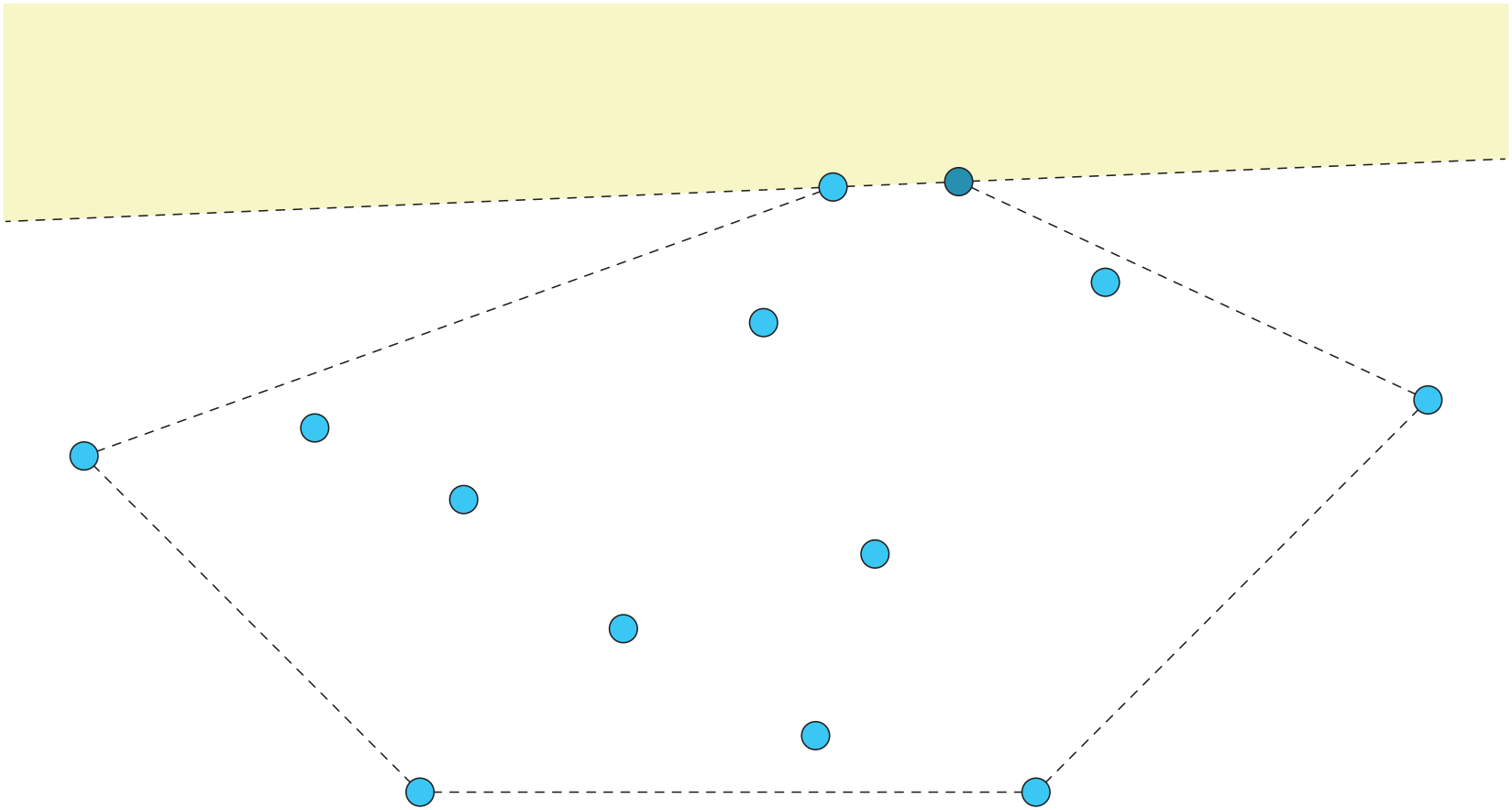
External robots remain external

Correctness of the algorithm



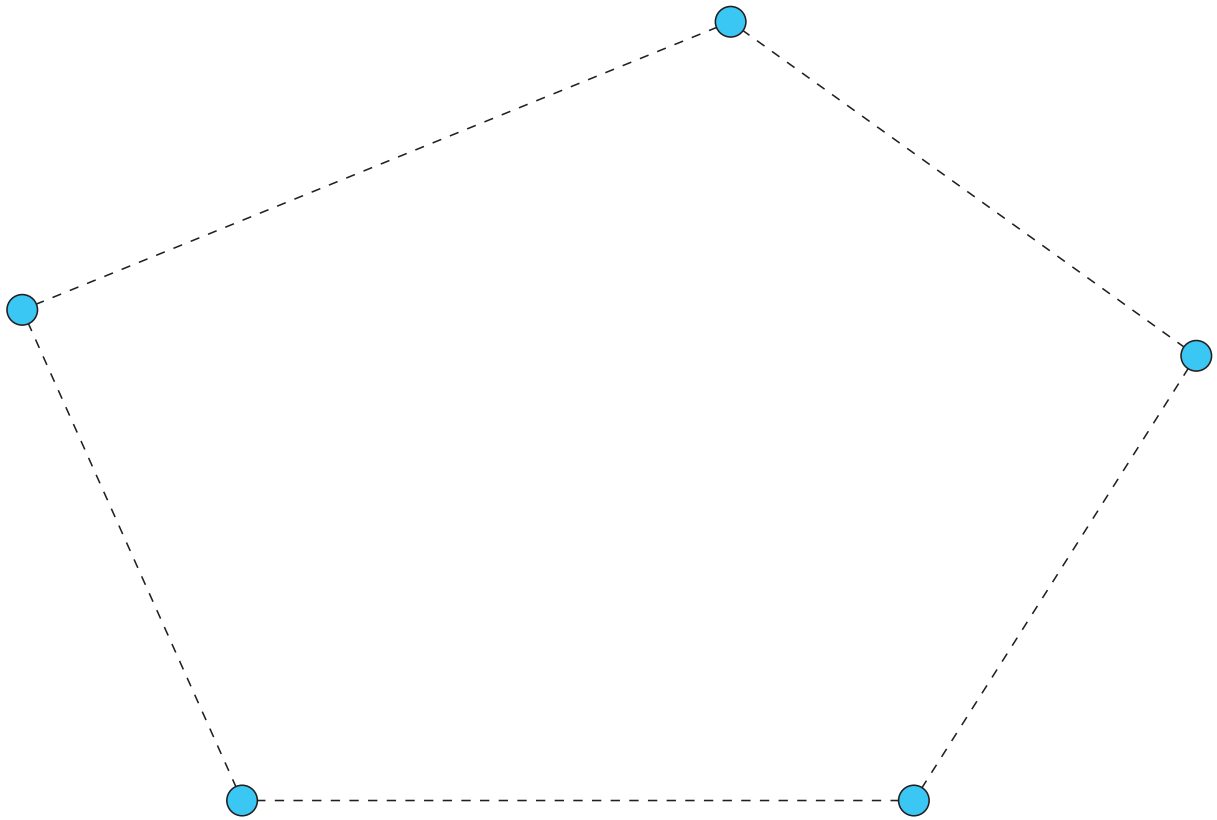
When a non-default move is made, a new robot becomes external

Correctness of the algorithm



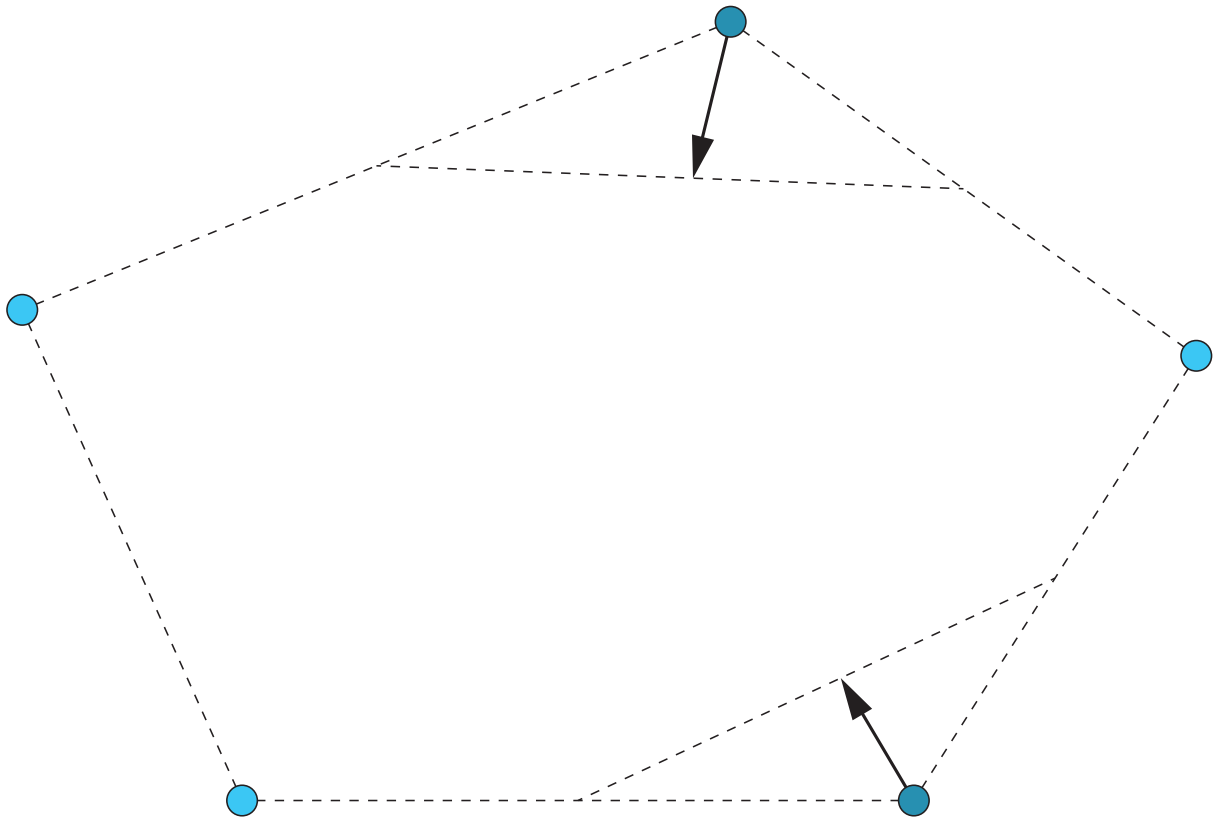
When a non-default move is made, a new robot becomes external

Correctness of the algorithm



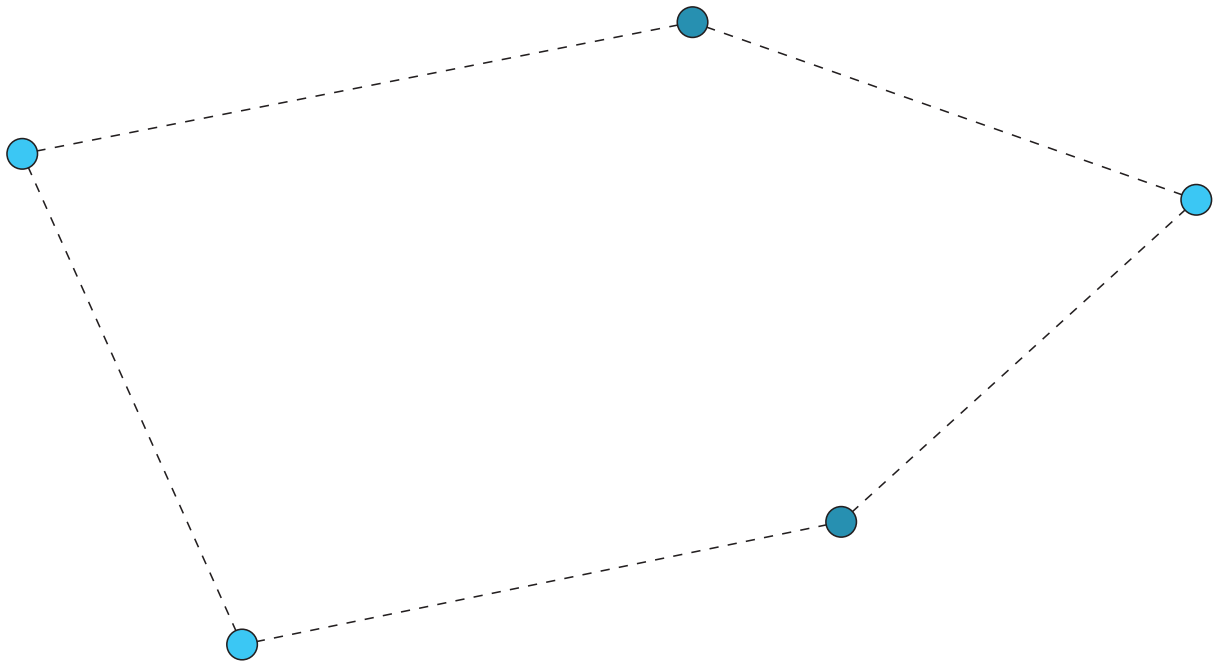
Claim: if only default moves are made, the robots converge

Correctness of the algorithm



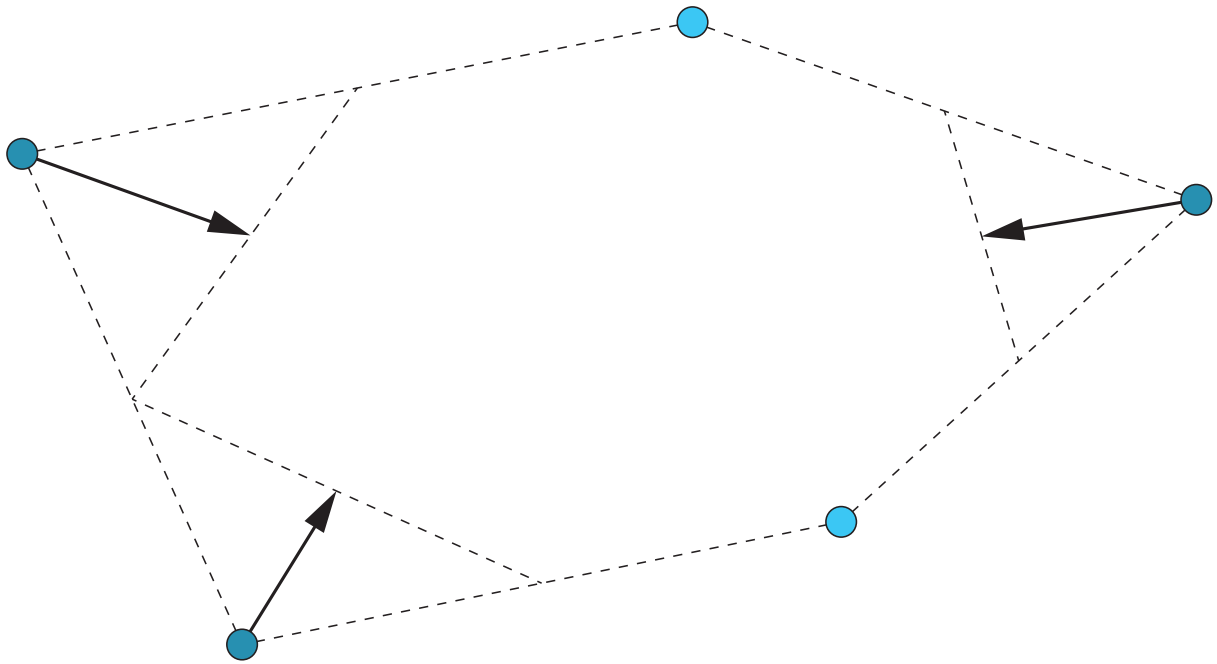
Claim: if only default moves are made, the robots converge

Correctness of the algorithm



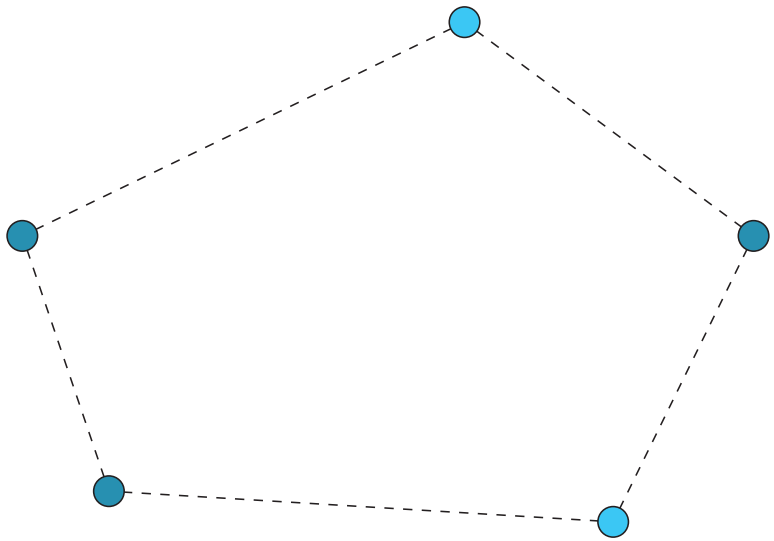
Claim: if only default moves are made, the robots converge

Correctness of the algorithm



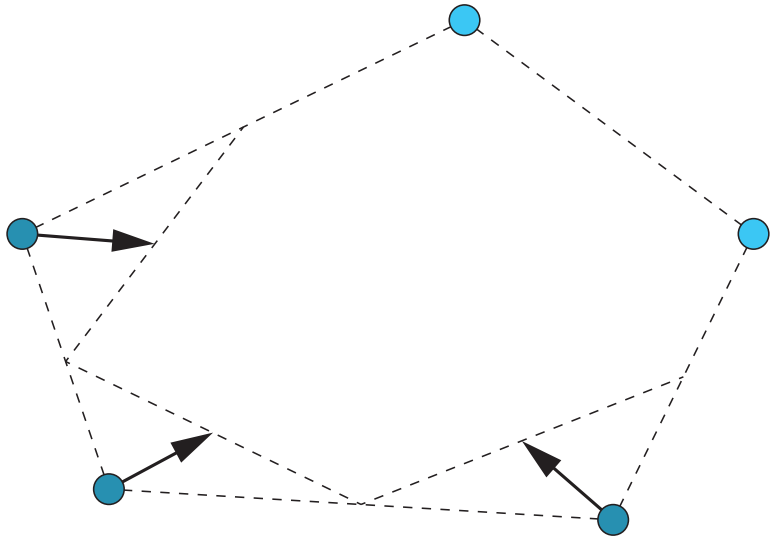
Claim: if only default moves are made, the robots converge

Correctness of the algorithm



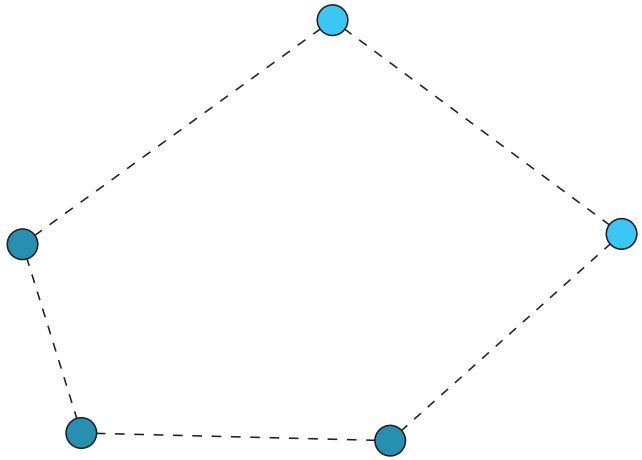
Claim: if only default moves are made, the robots converge

Correctness of the algorithm



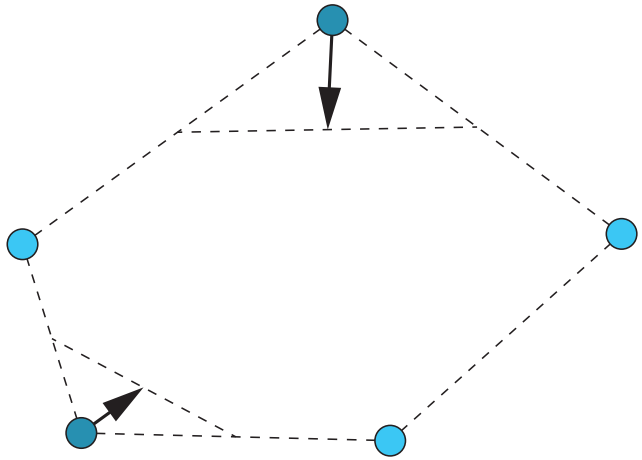
Claim: if only default moves are made, the robots converge

Correctness of the algorithm



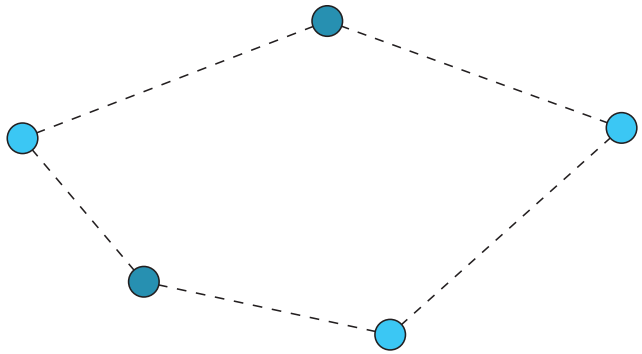
Claim: if only default moves are made, the robots converge

Correctness of the algorithm



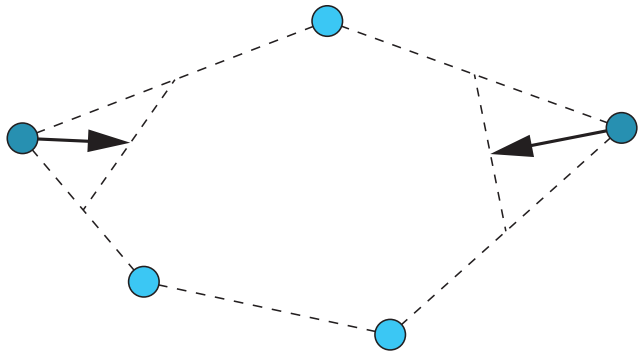
Claim: if only default moves are made, the robots converge

Correctness of the algorithm



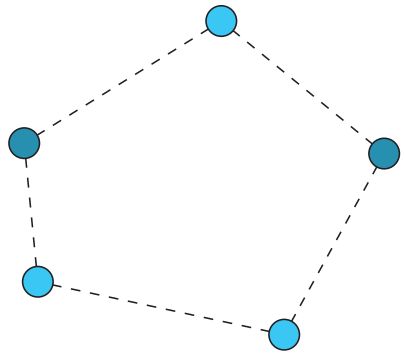
Claim: if only default moves are made, the robots converge

Correctness of the algorithm



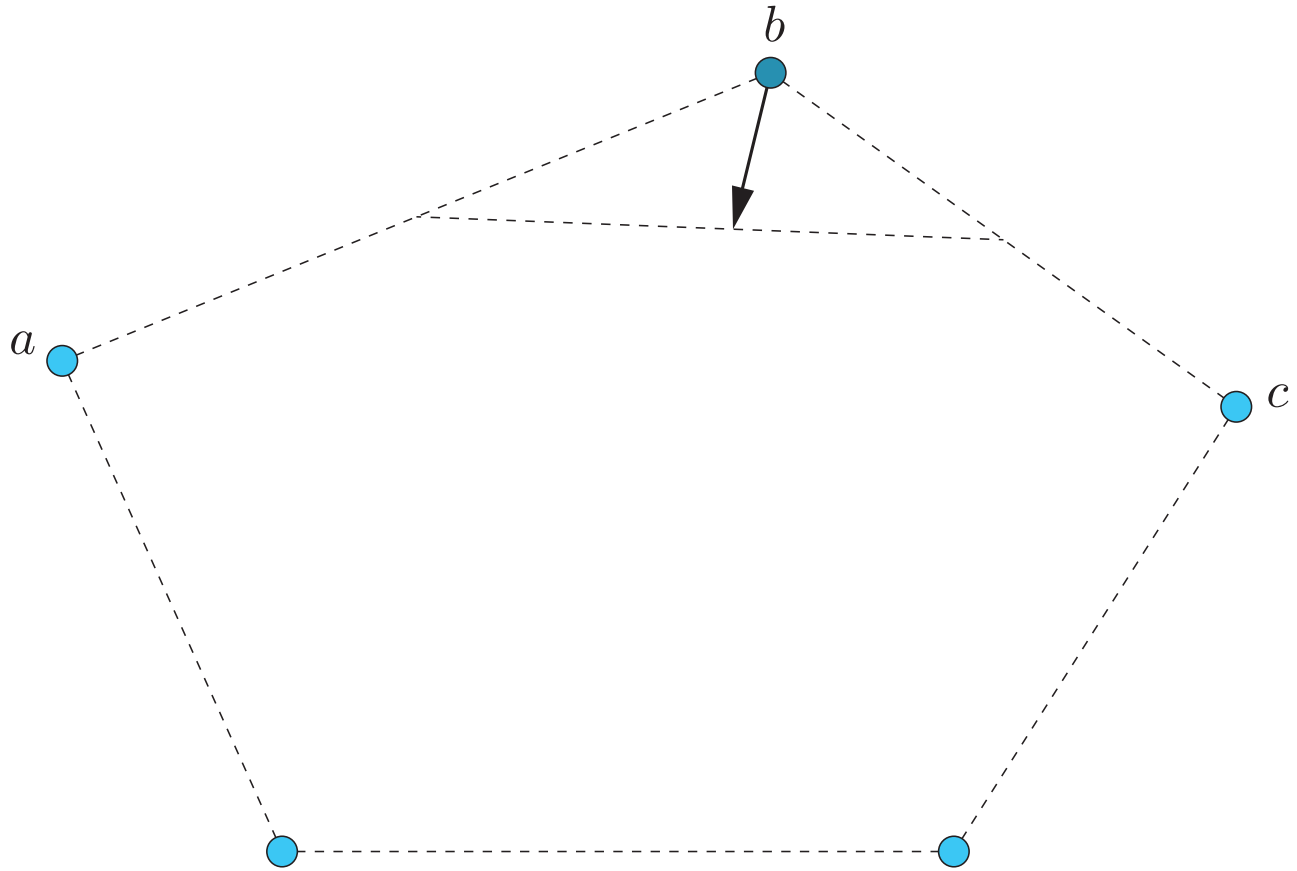
Claim: if only default moves are made, the robots converge

Correctness of the algorithm



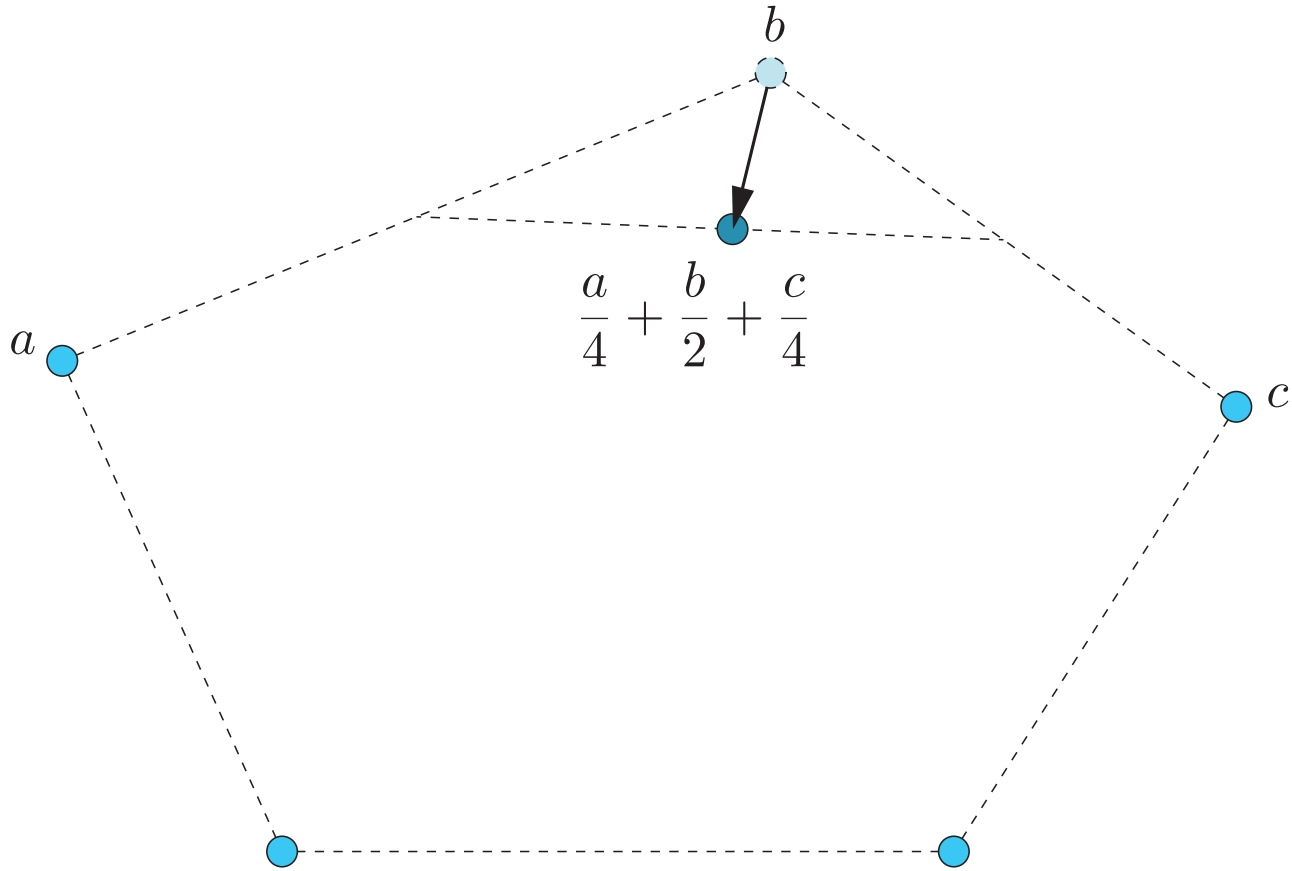
Claim: if only default moves are made, the robots converge

Correctness of the algorithm



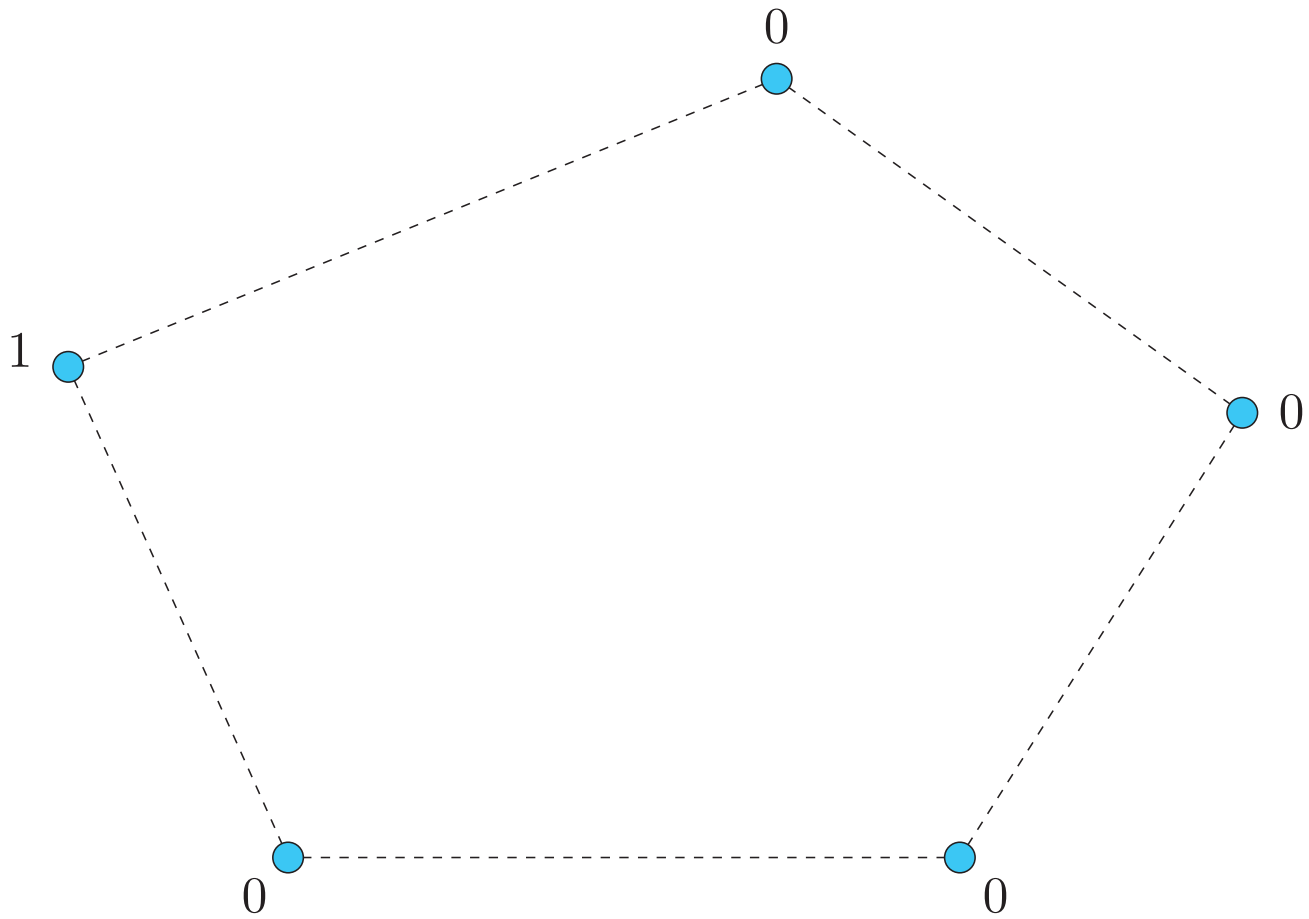
New position = convex combination of old positions

Correctness of the algorithm



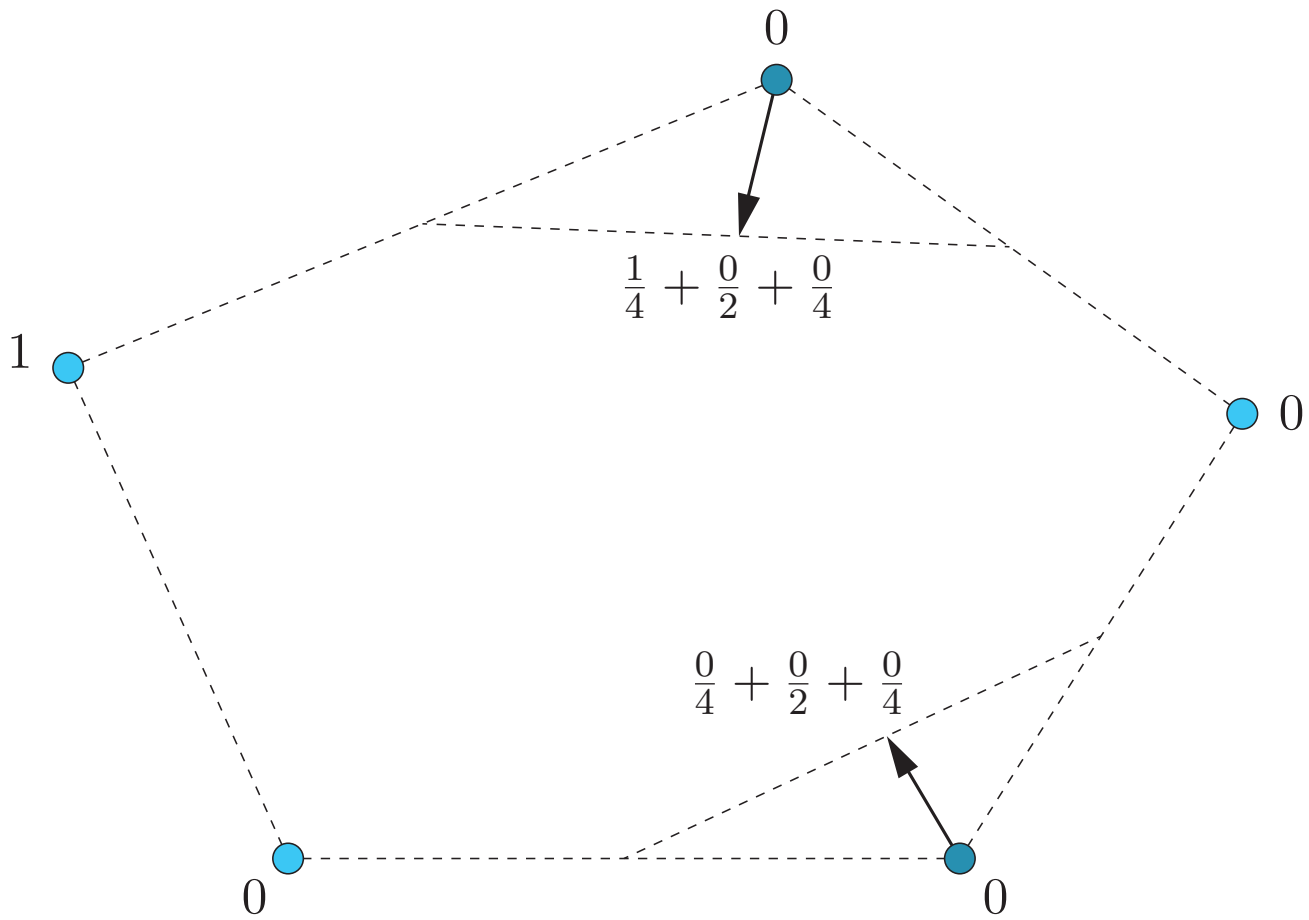
New position = convex combination of old positions

Correctness of the algorithm



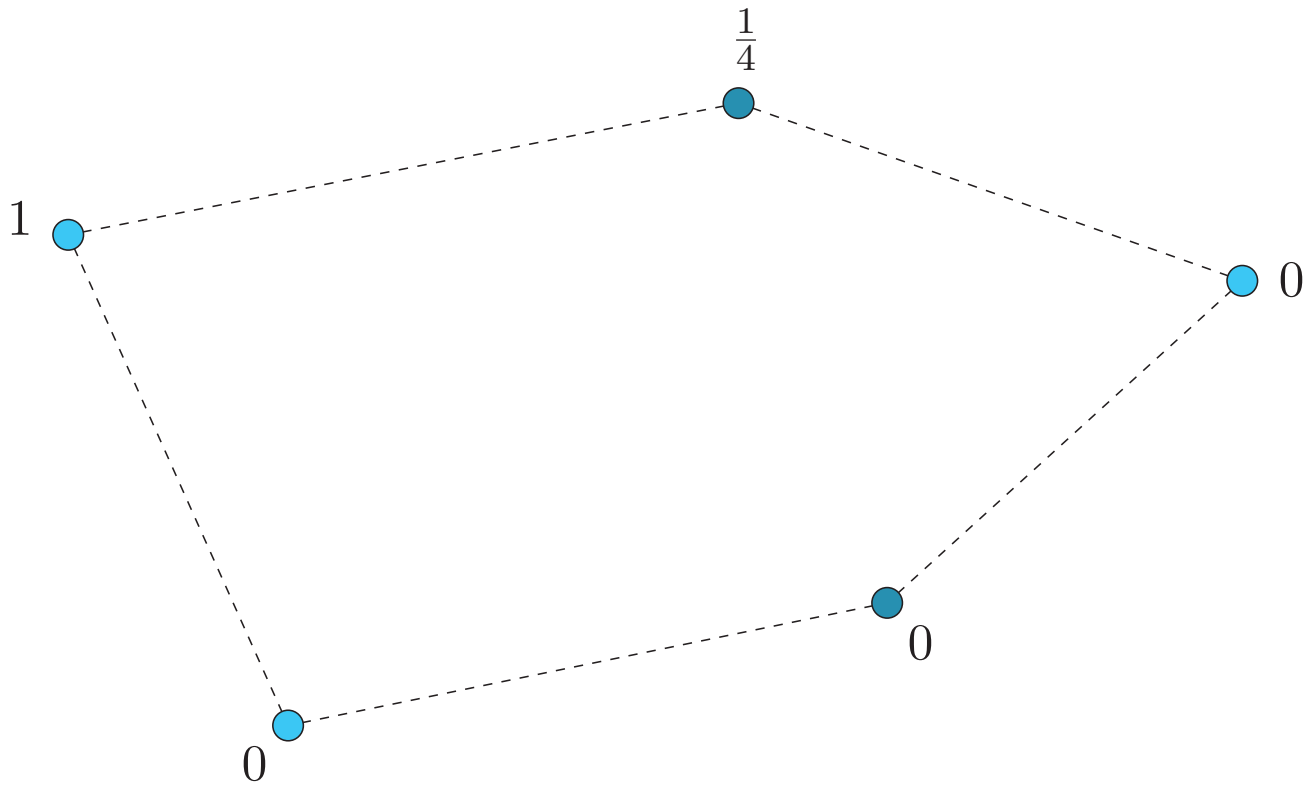
Consider only the coefficients corresponding to one robot

Correctness of the algorithm



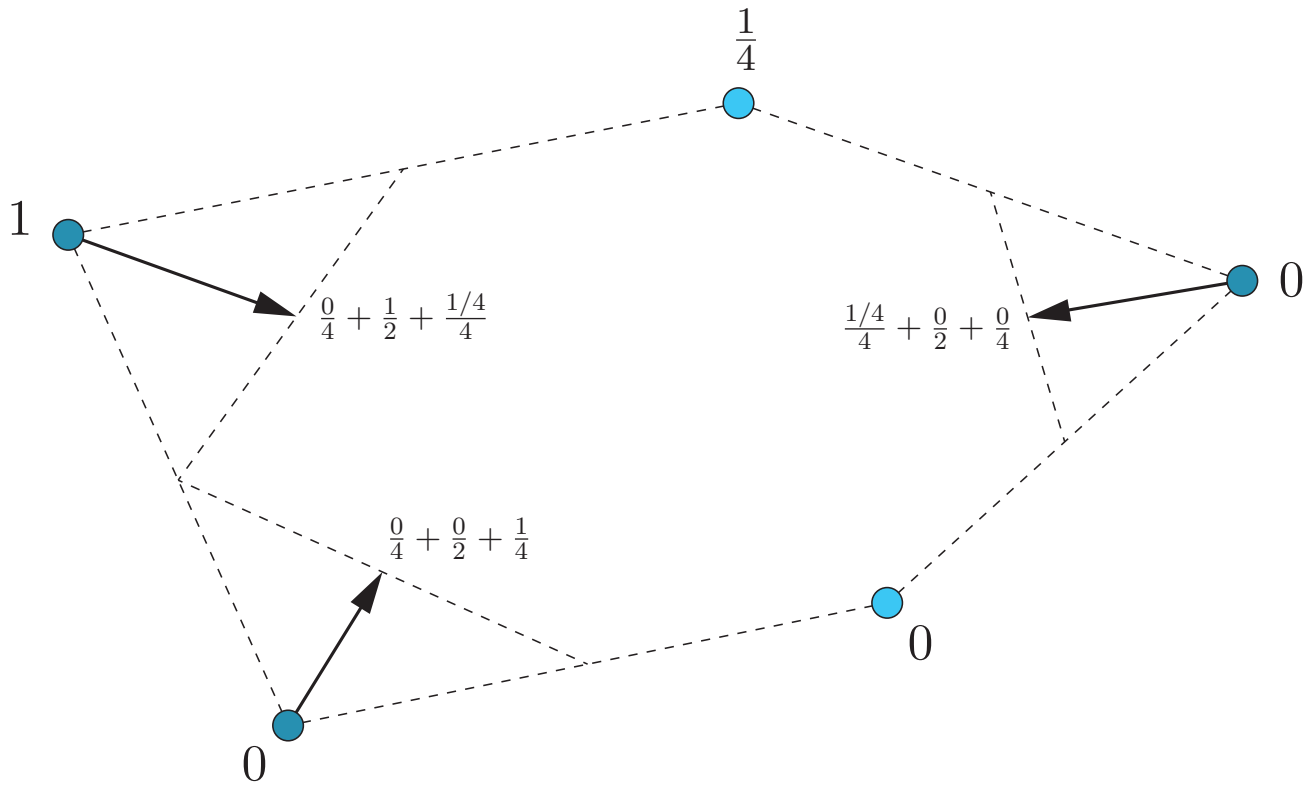
Consider only the coefficients corresponding to one robot

Correctness of the algorithm



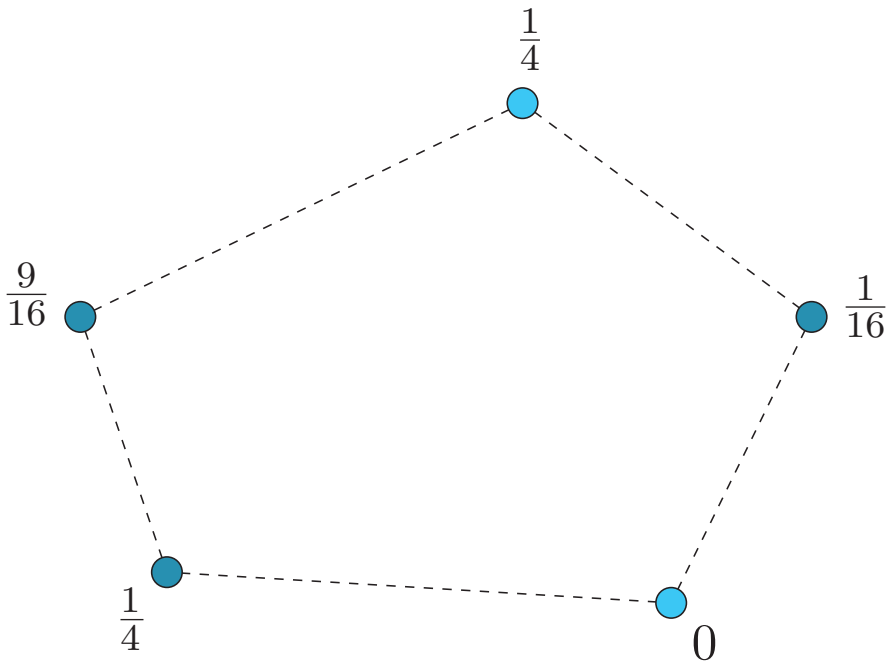
Consider only the coefficients corresponding to one robot

Correctness of the algorithm



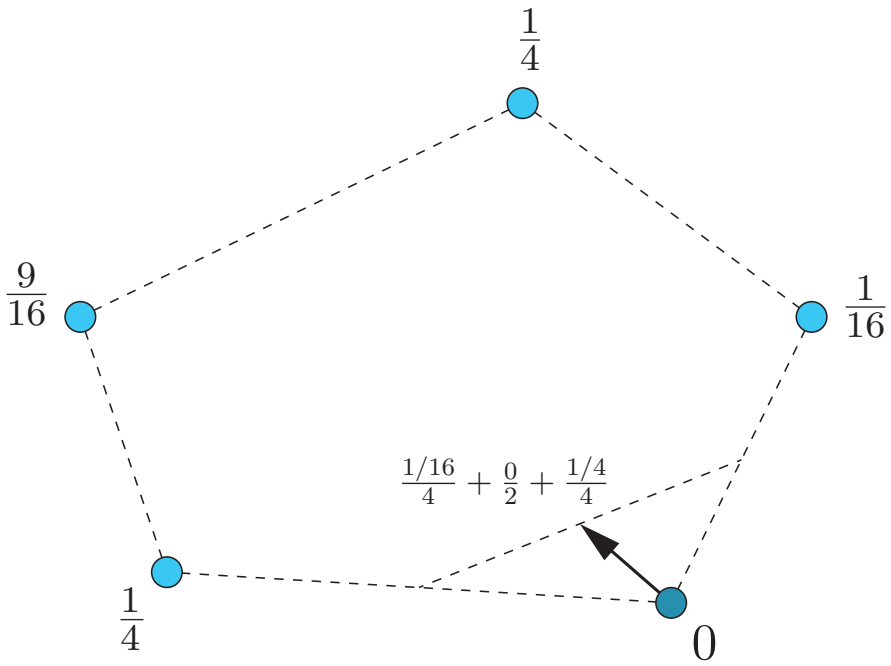
Consider only the coefficients corresponding to one robot

Correctness of the algorithm



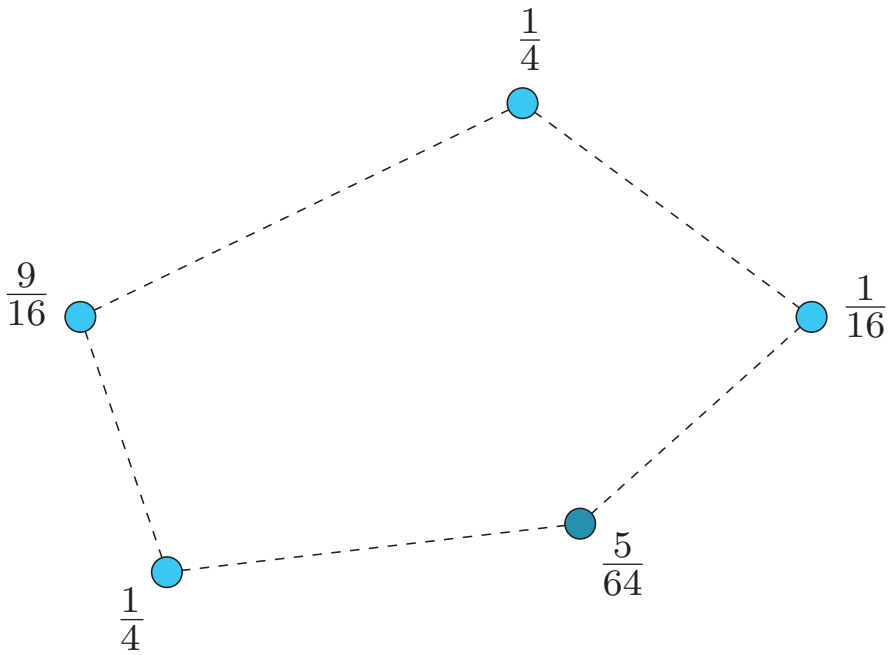
Consider only the coefficients corresponding to one robot

Correctness of the algorithm



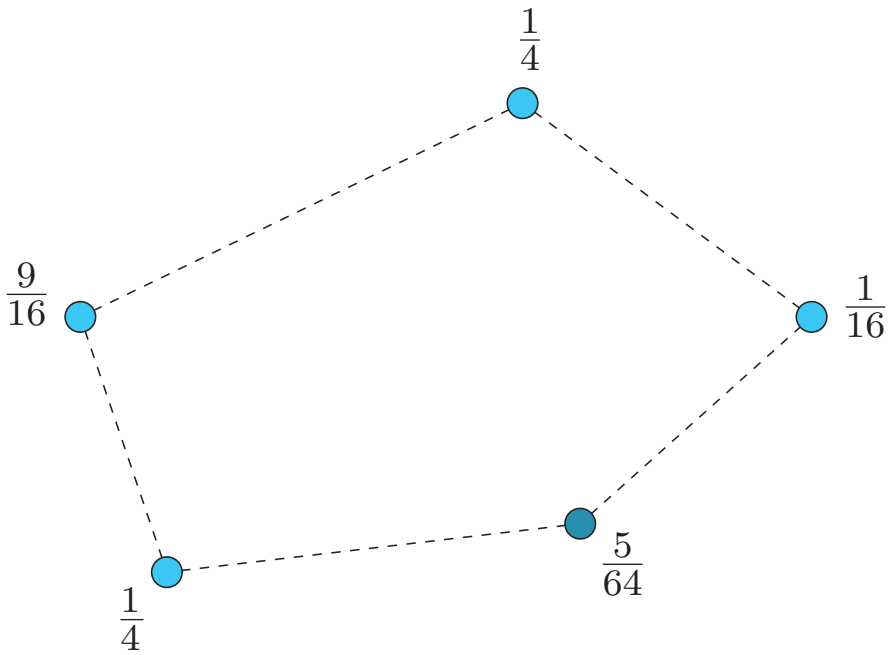
Consider only the coefficients corresponding to one robot

Correctness of the algorithm



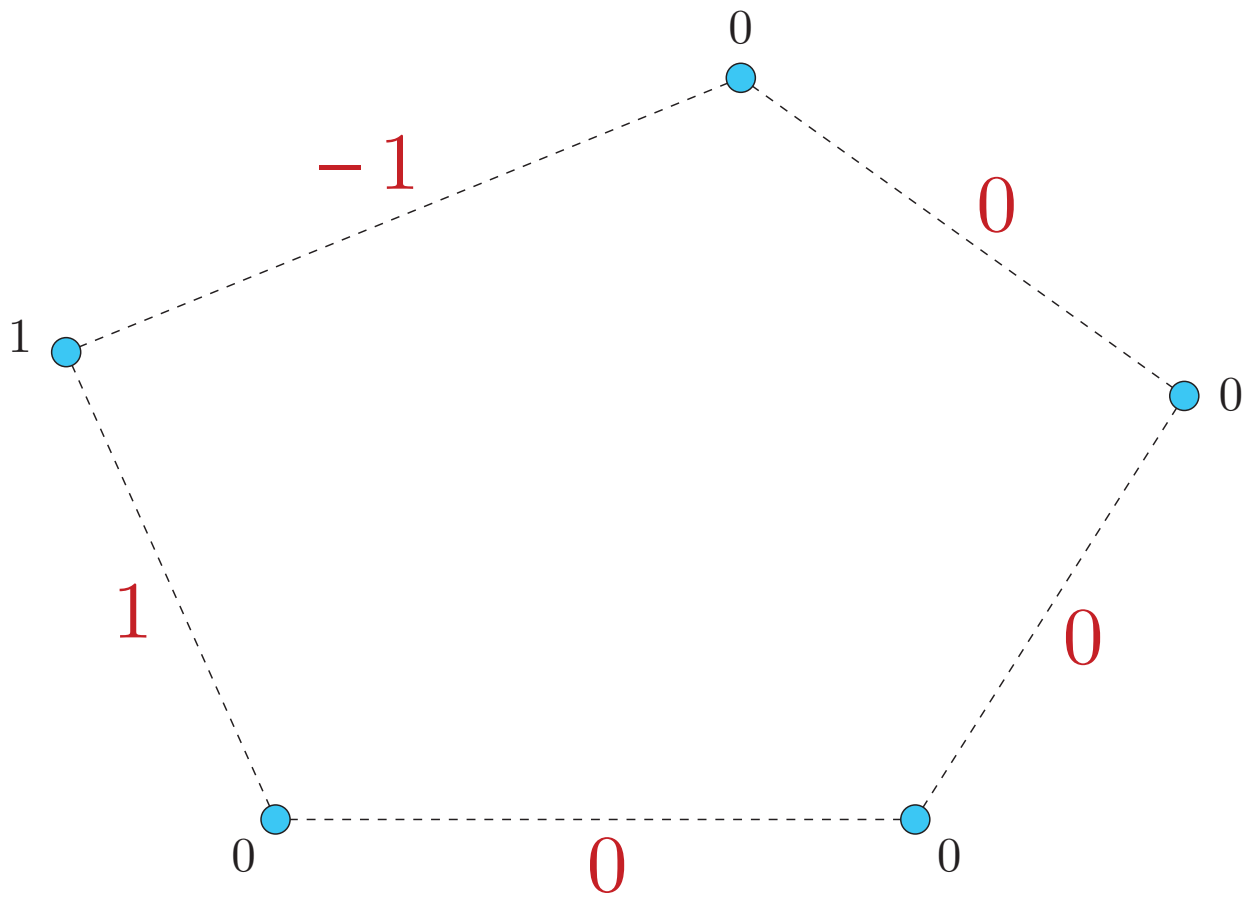
Consider only the coefficients corresponding to one robot

Correctness of the algorithm



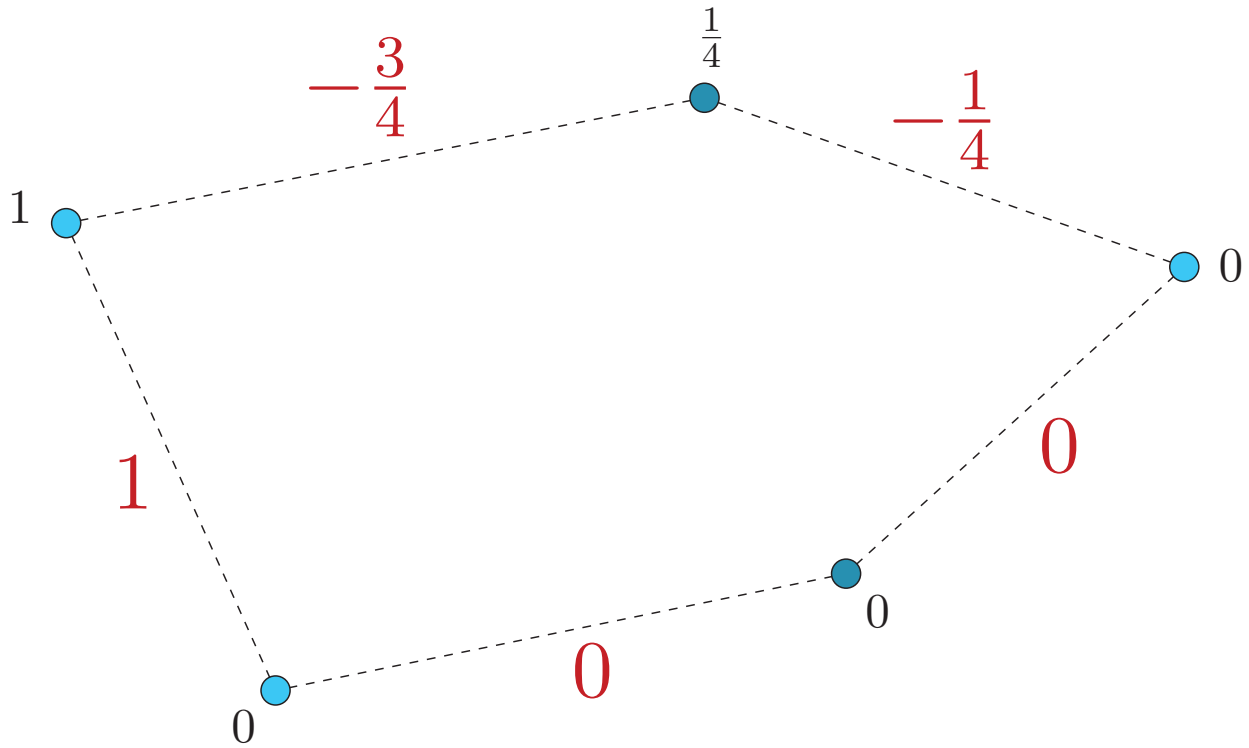
Claim: the coefficients converge to the same limit

Correctness of the algorithm



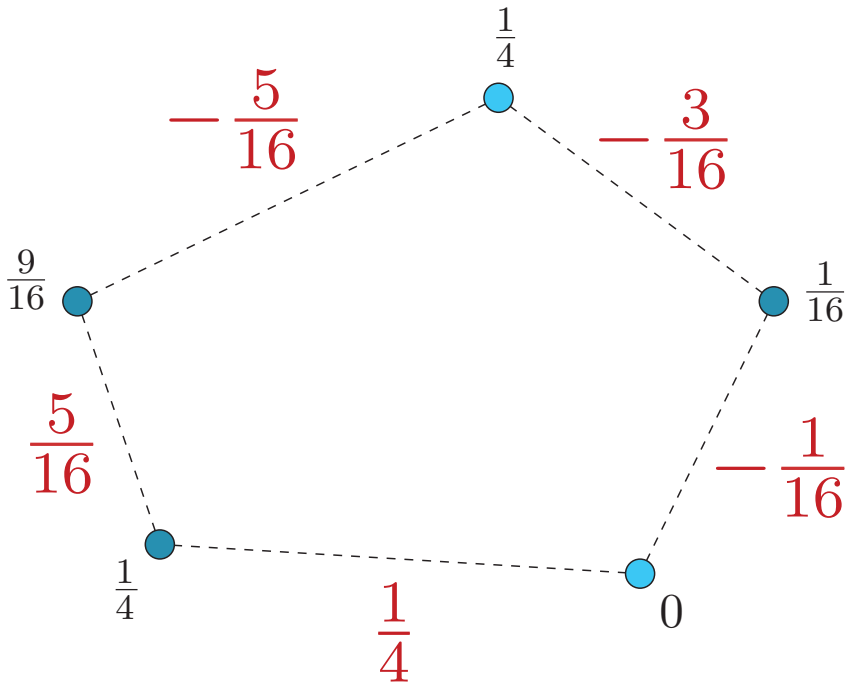
Consider the differences between adjacent coefficients

Correctness of the algorithm



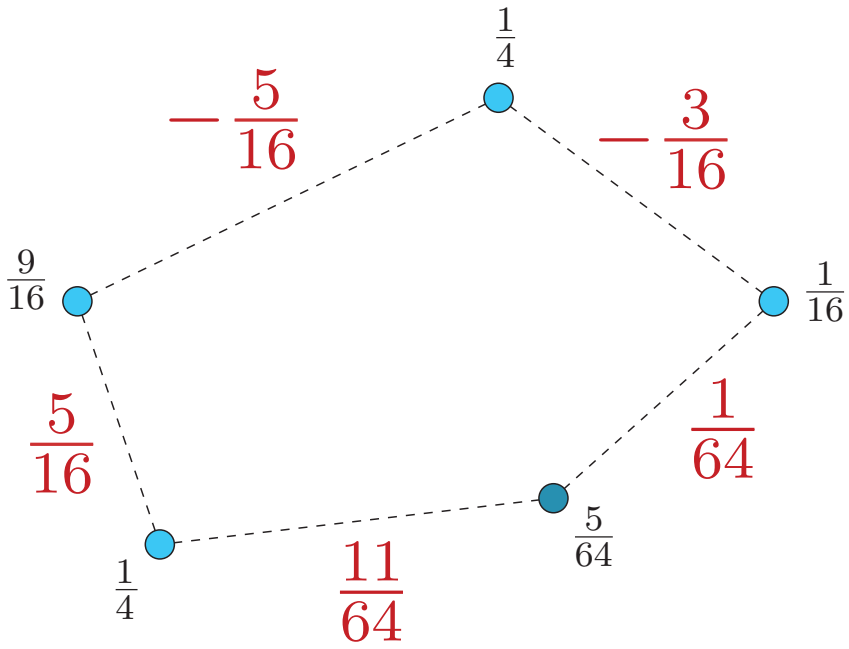
Consider the differences between adjacent coefficients

Correctness of the algorithm



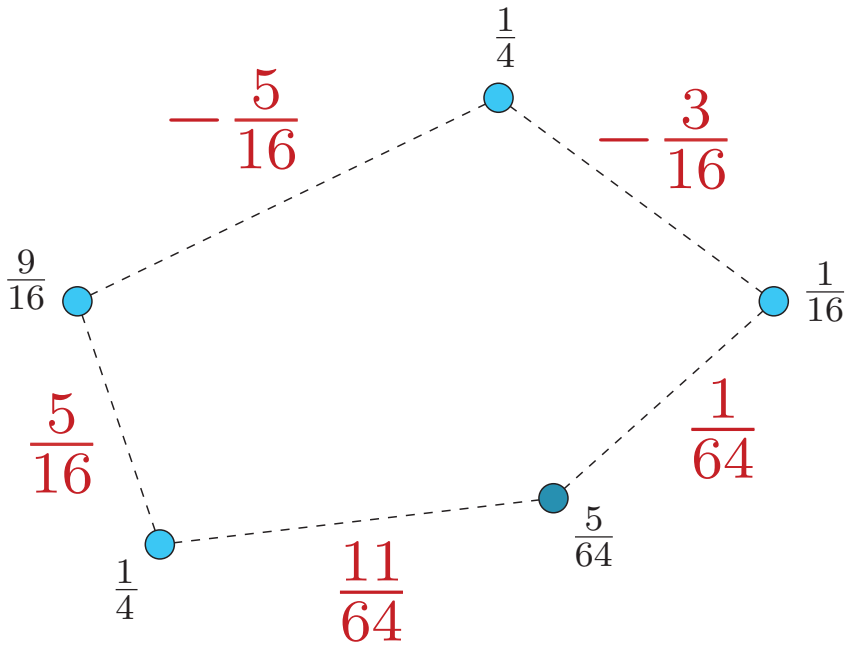
Consider the differences between adjacent coefficients

Correctness of the algorithm



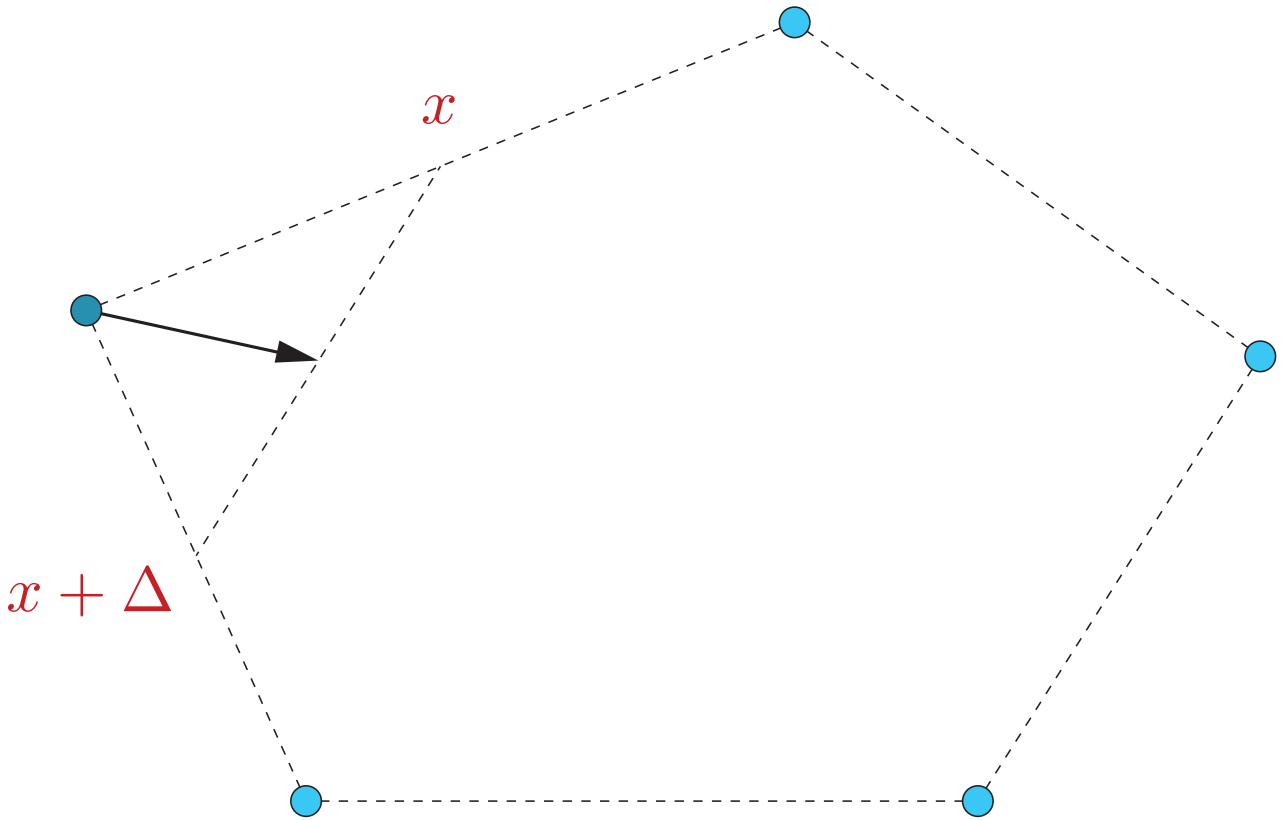
Consider the differences between adjacent coefficients

Correctness of the algorithm



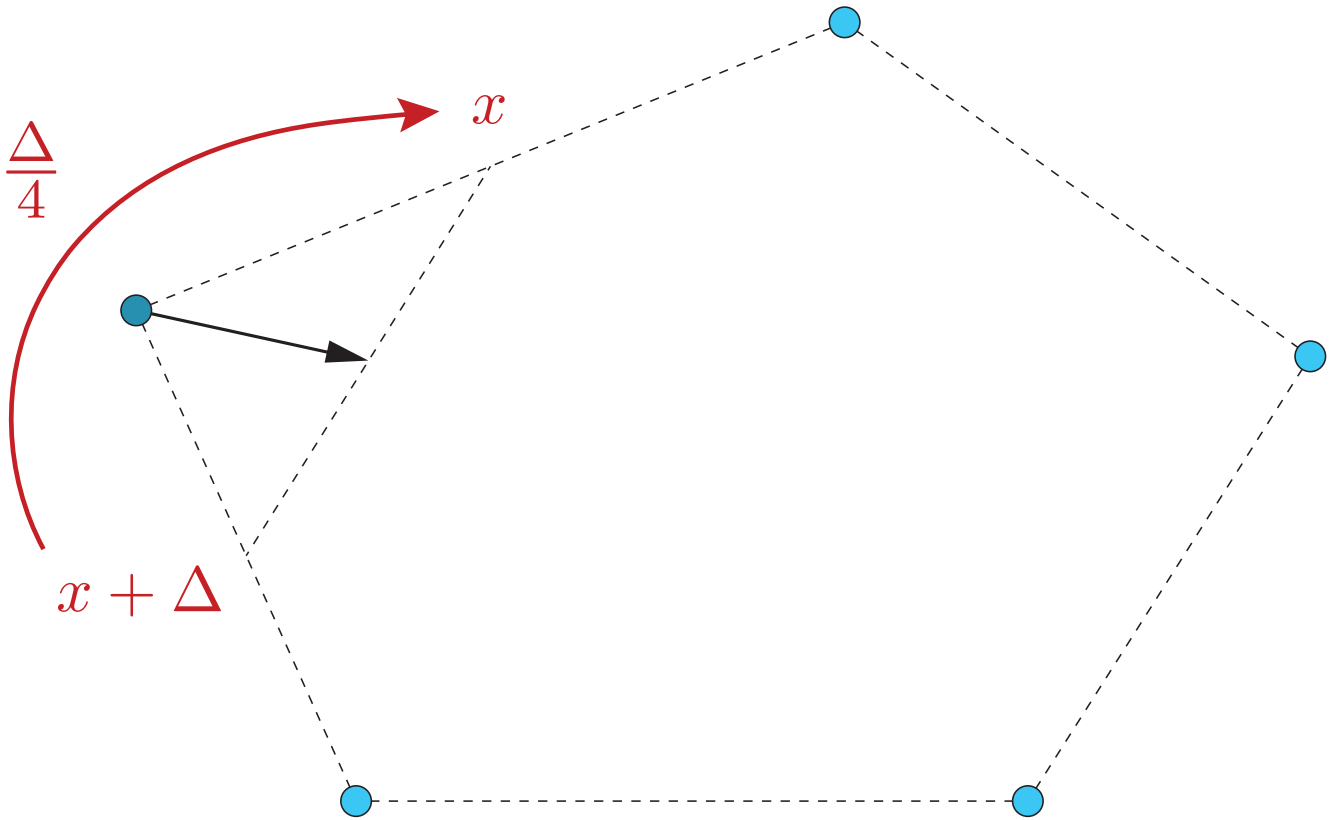
Claim: all the differences vanish

Correctness of the algorithm



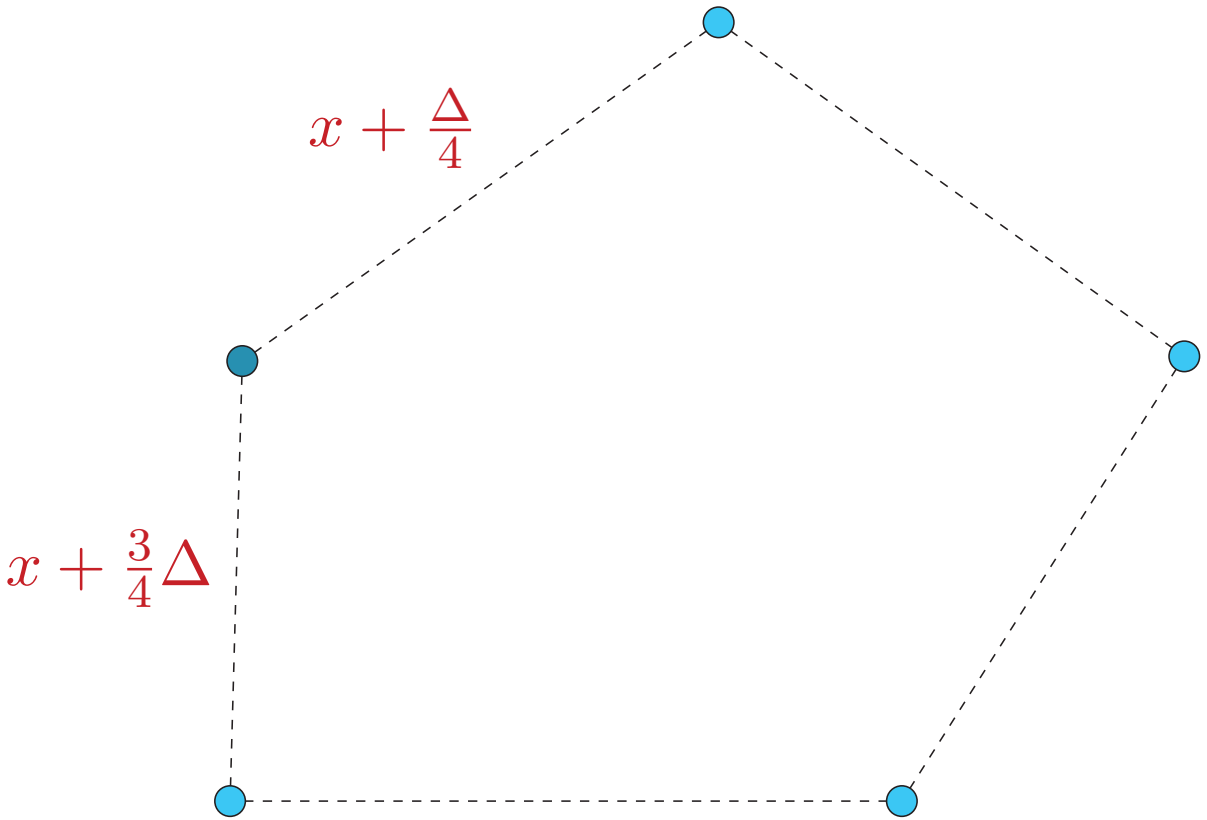
Upon activation of a robot, 1/4 of the surplus is transferred between the two adjacent coefficients

Correctness of the algorithm



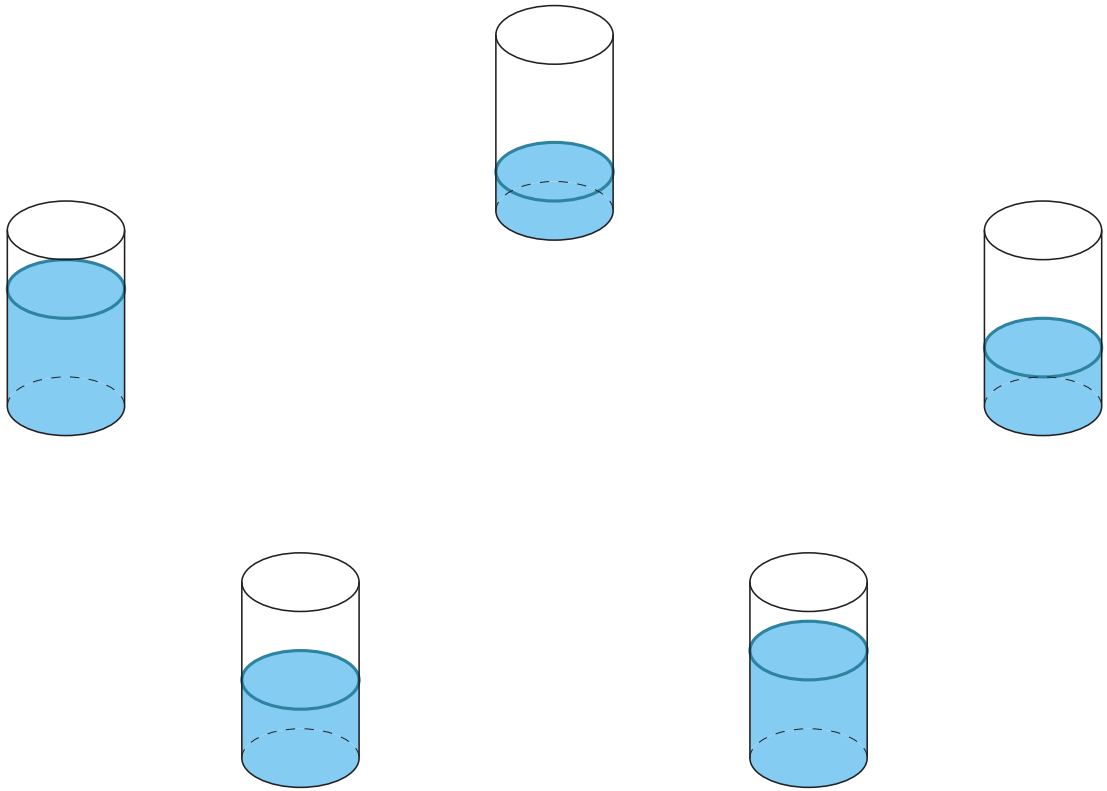
Upon activation of a robot, 1/4 of the surplus is transferred between the two adjacent coefficients

Correctness of the algorithm



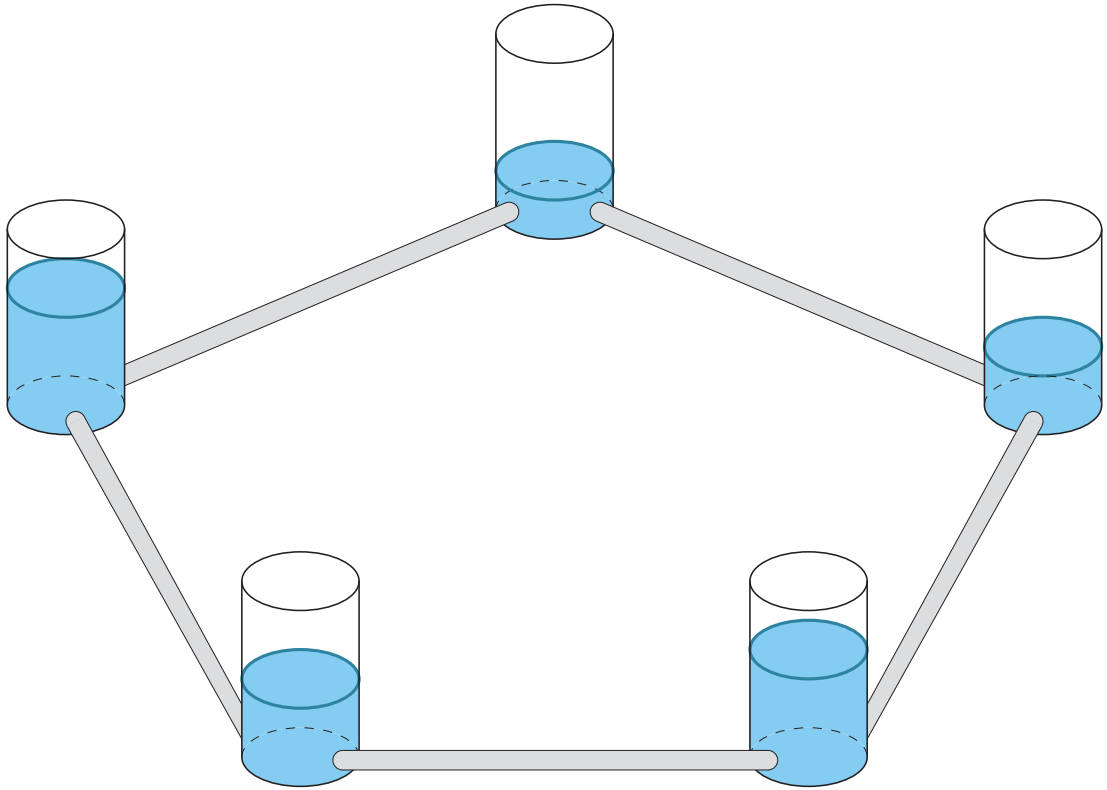
Upon activation of a robot, 1/4 of the surplus is transferred between the two adjacent coefficients

The Communicating Vessels theorem



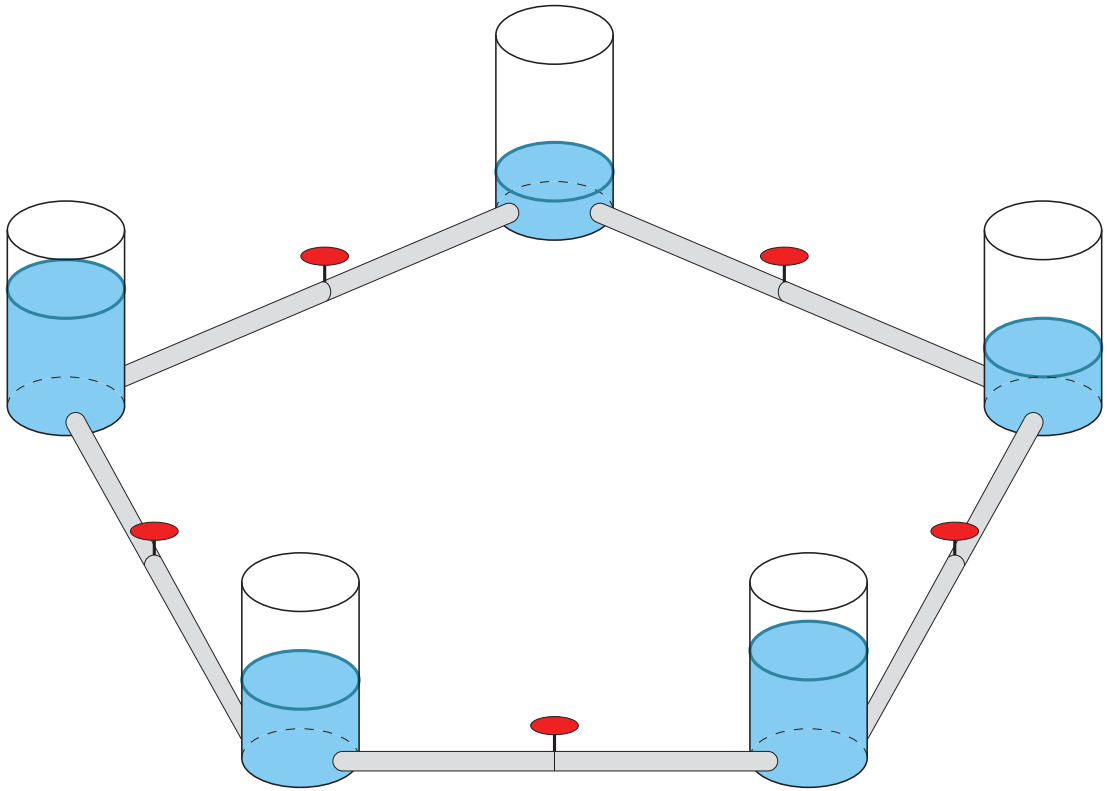
Alternative formulation: n vessels containing liquid, arranged in a circle and connected by pipes regulated by valves

The Communicating Vessels theorem



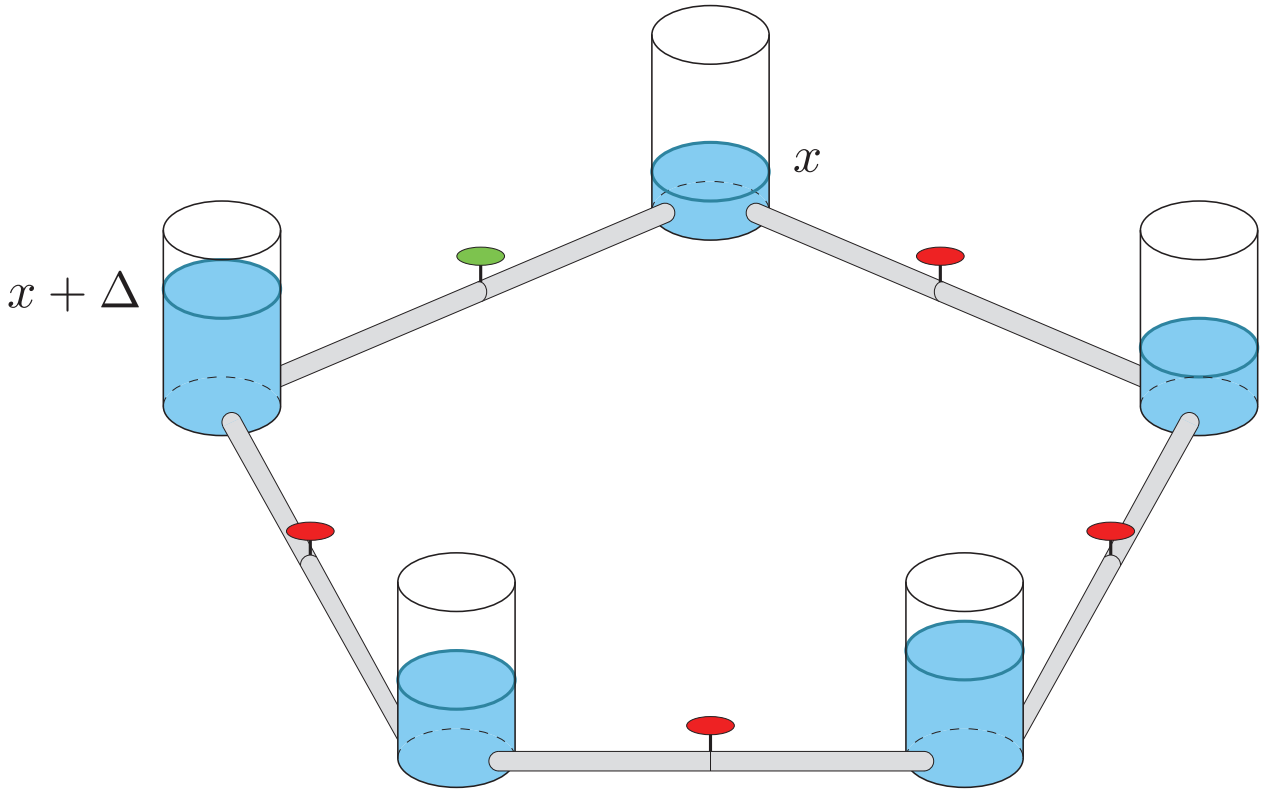
Alternative formulation: n vessels containing liquid, arranged in a circle and connected by pipes regulated by valves

The Communicating Vessels theorem



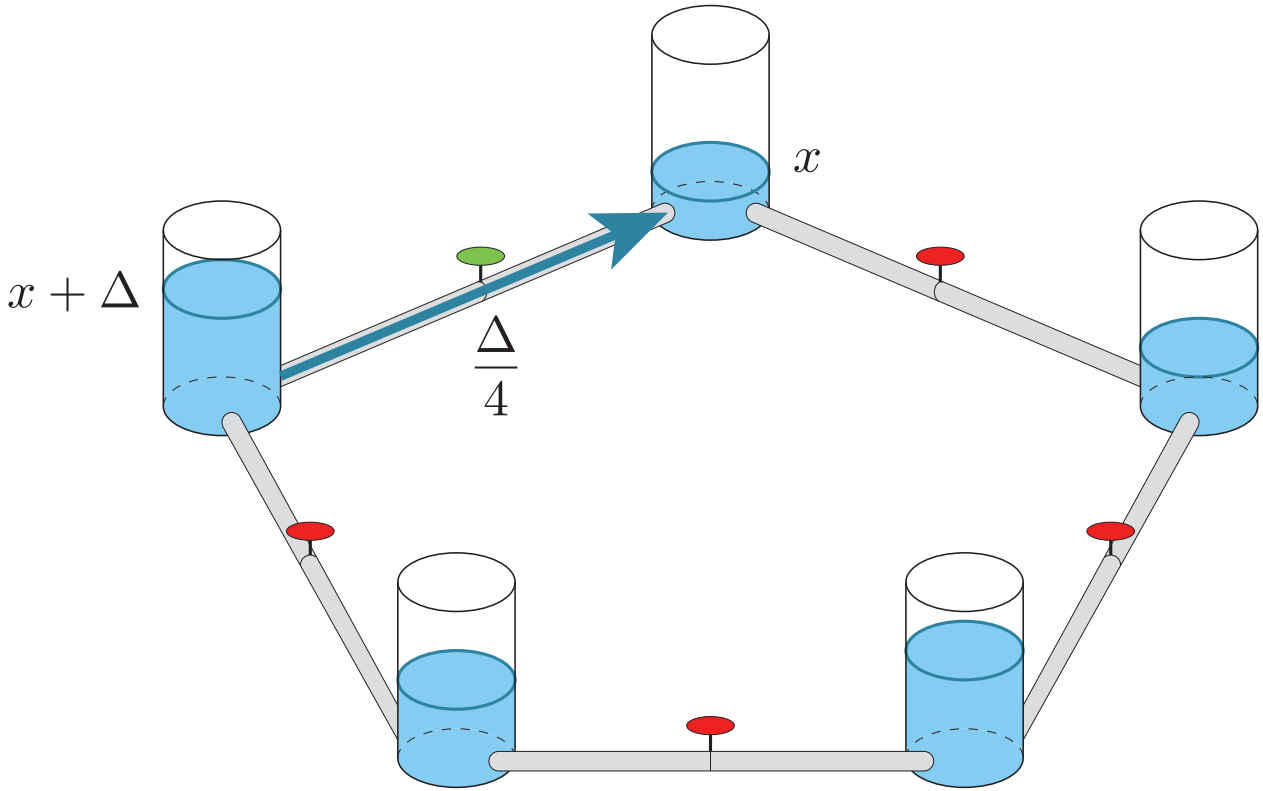
Alternative formulation: n vessels containing liquid, arranged in a circle and connected by pipes regulated by valves

The Communicating Vessels theorem



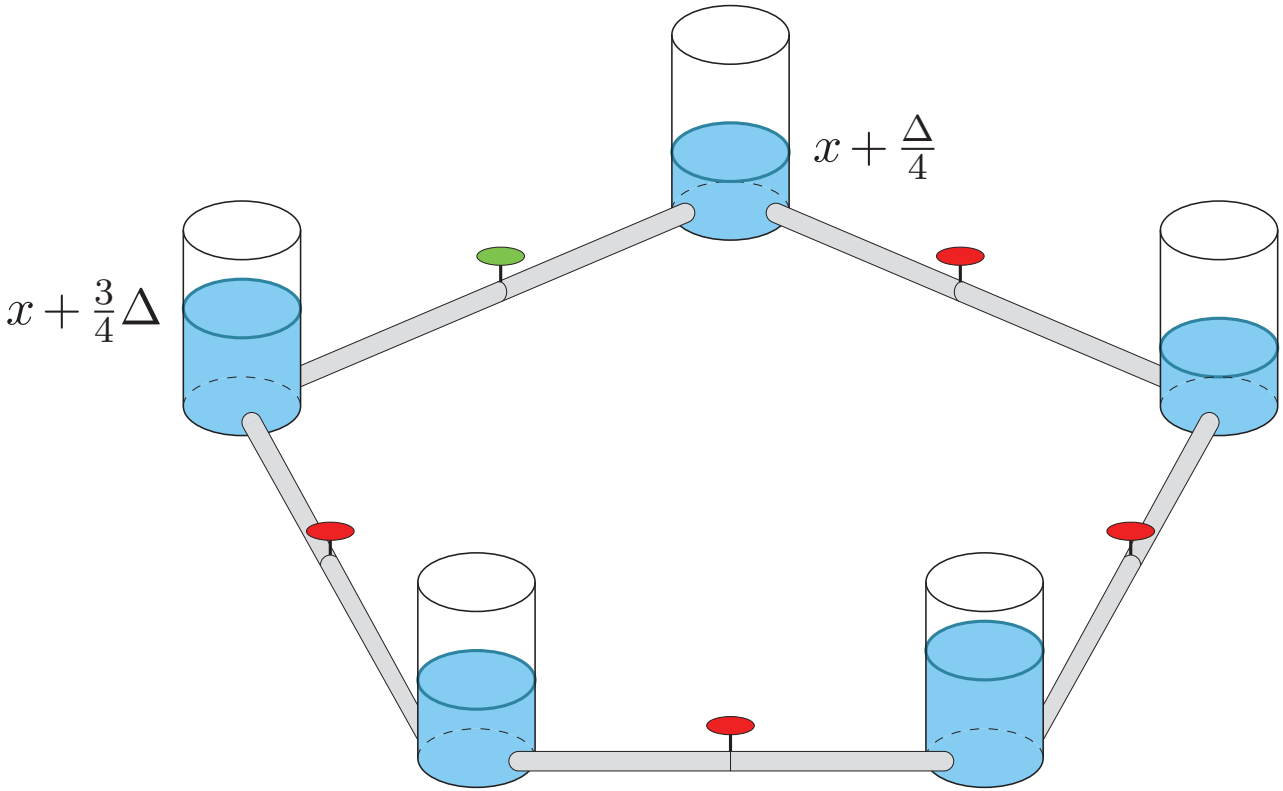
When a valve between two vessels is opened, 1/4 of the surplus of liquid flows from the fuller to the emptier vessel

The Communicating Vessels theorem



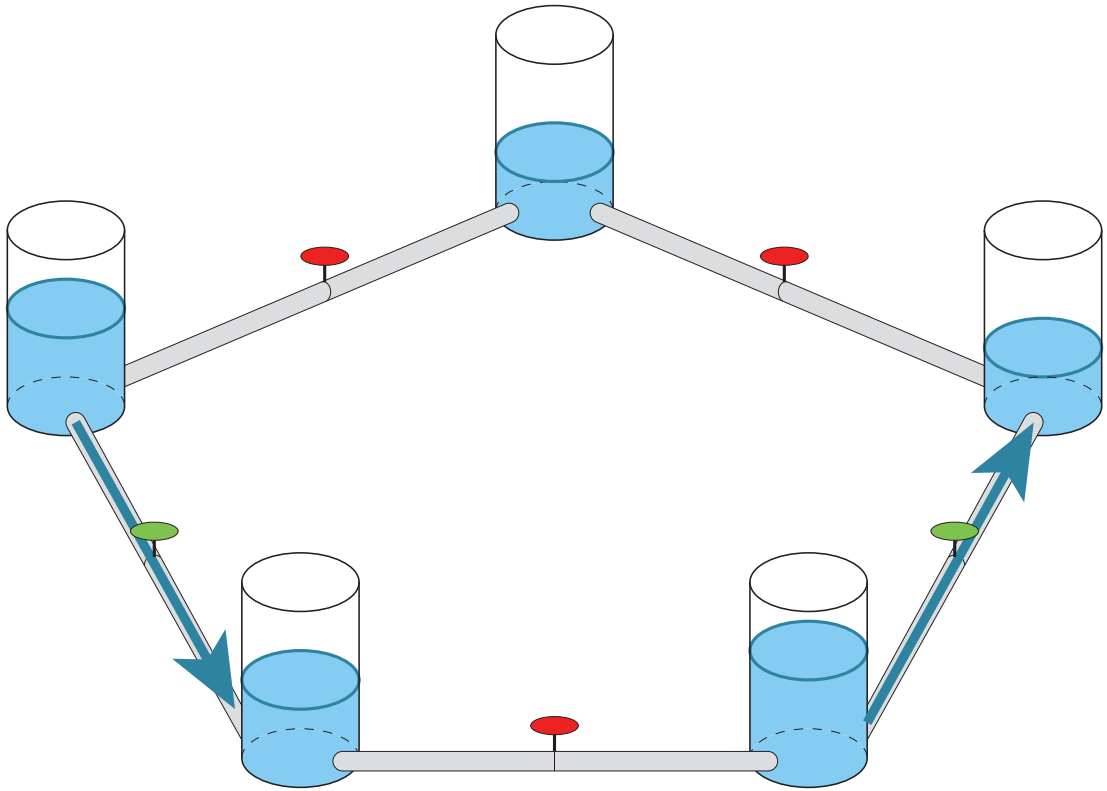
When a valve between two vessels is opened, 1/4 of the surplus of liquid flows from the fuller to the emptier vessel

The Communicating Vessels theorem



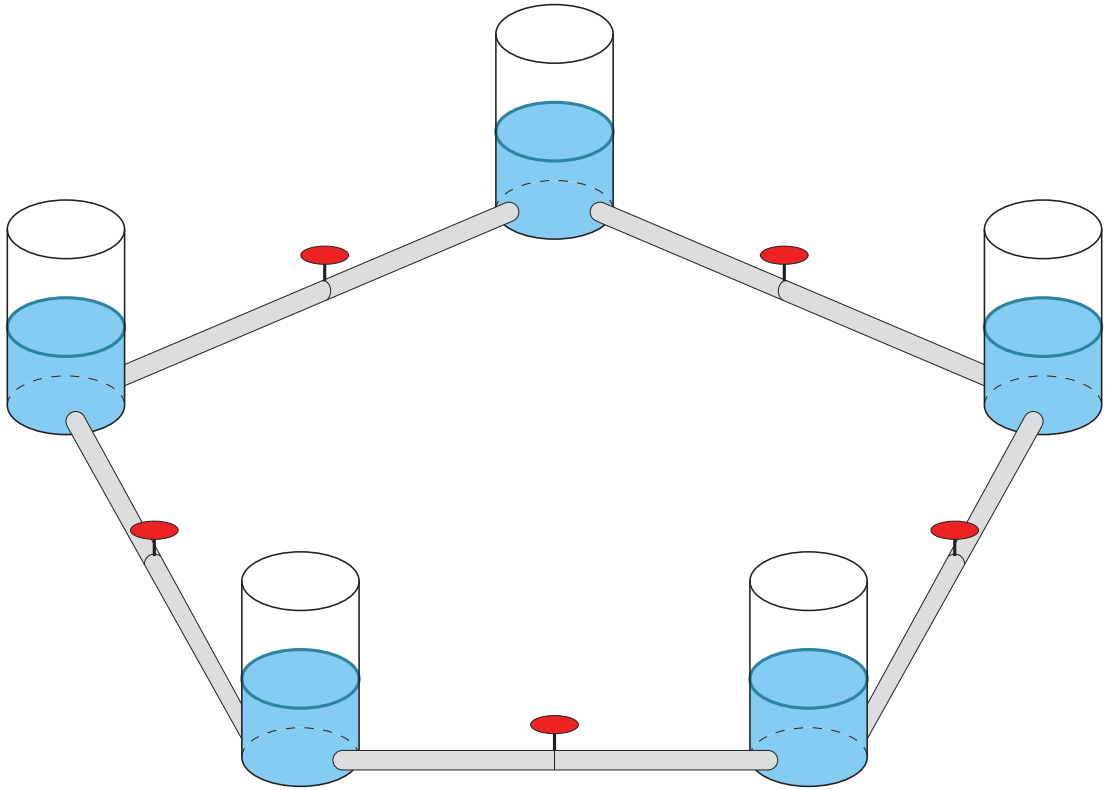
When a valve between two vessels is opened, 1/4 of the surplus of liquid flows from the fuller to the emptier vessel

The Communicating Vessels theorem



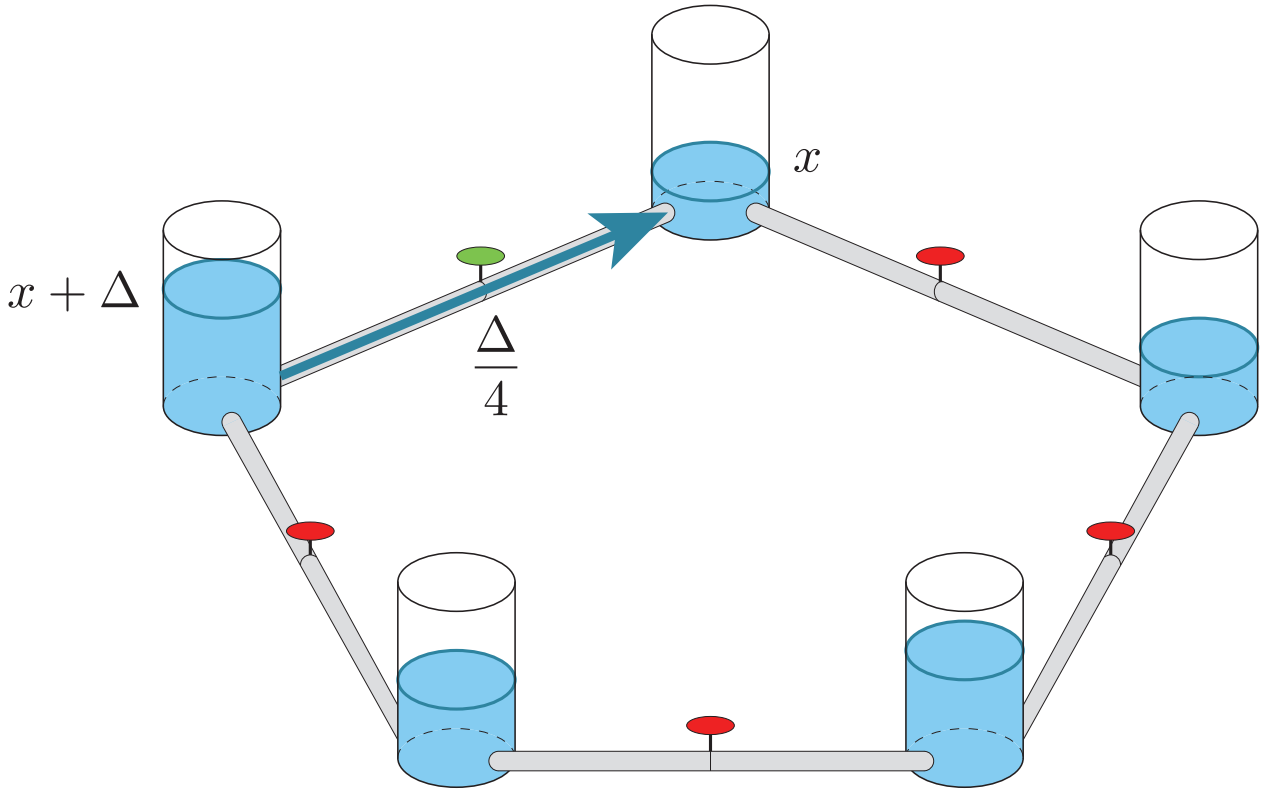
Fairness: at every second, some valves are opened and some are closed; each valve is opened infinitely many times

The Communicating Vessels theorem



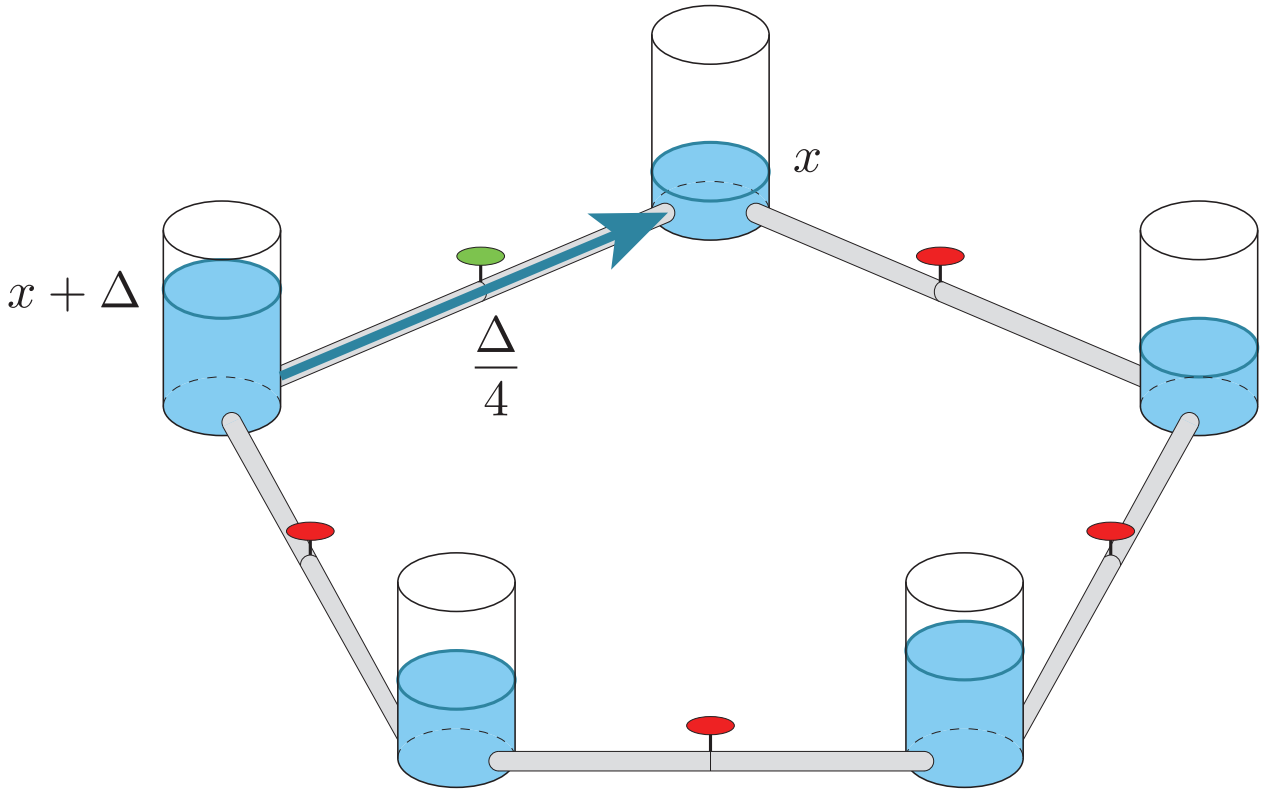
Claim: the levels of liquid in all vessels converge to the same limit

Proving the Communicating Vessels theorem



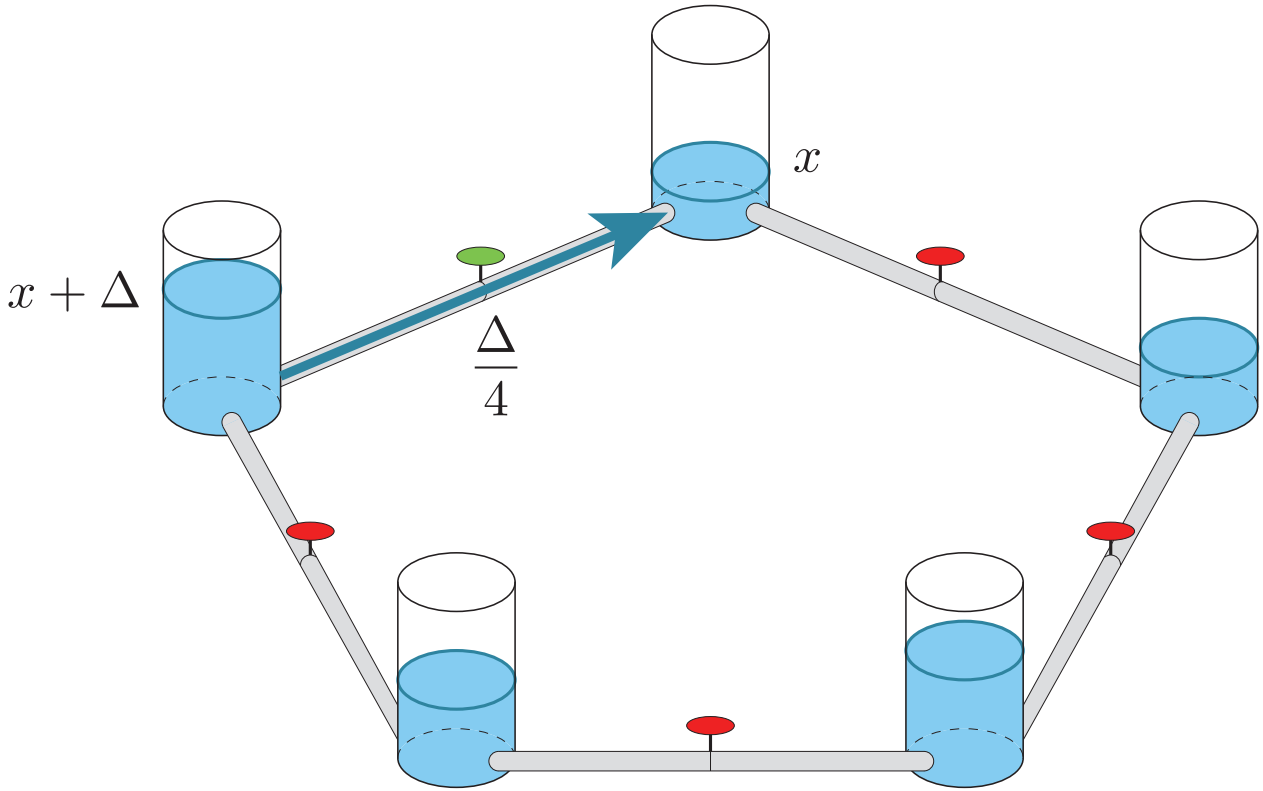
Observation: the sum of the levels remains constant, and so does the average level; what about the variance?

Proving the Communicating Vessels theorem



Let E_t be the sum of squared levels at time t; how does E change after a valve is opened?

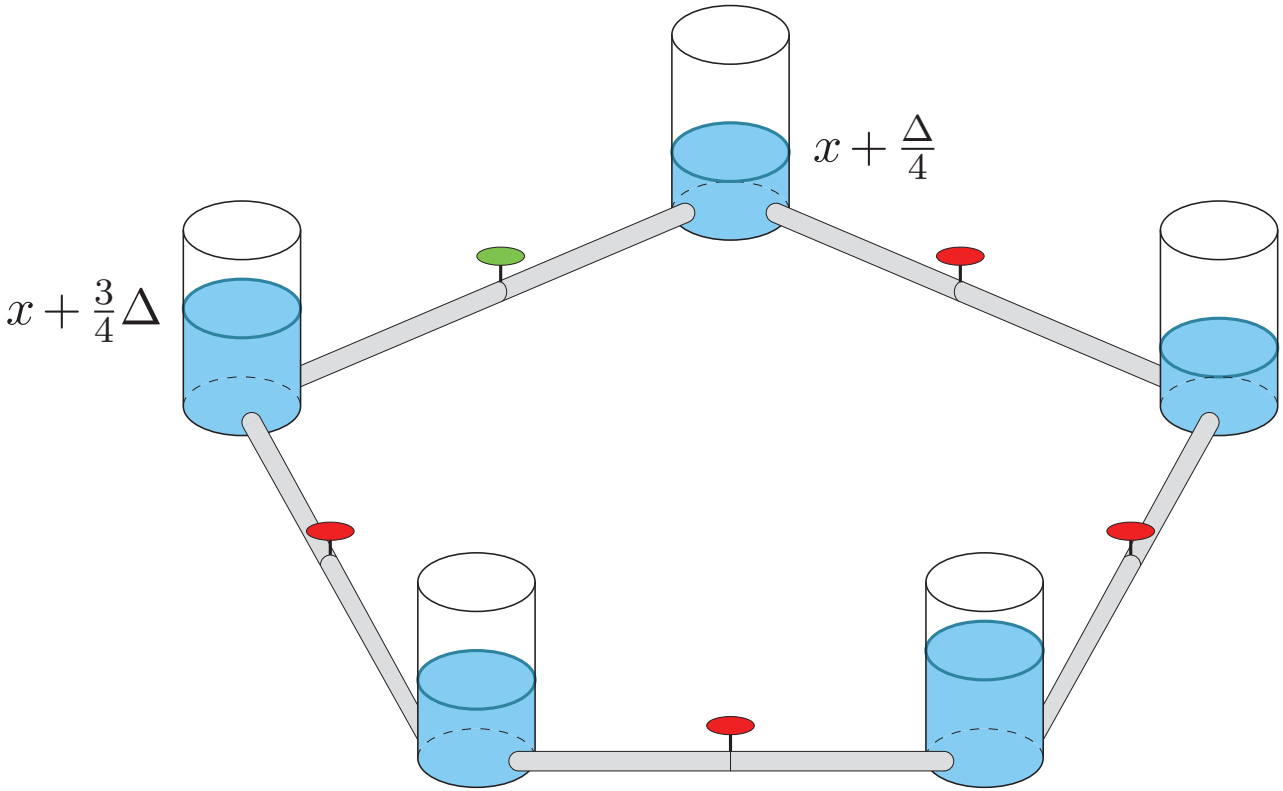
Proving the Communicating Vessels theorem



If only one valve is open:

$E_t - E_{t+1} = ((x + \Delta)^2 + x^2) - ((x + \frac{3}{4}\Delta)^2 + (x + \frac{\Delta}{4})^2) = \frac{3}{8}\Delta^2$

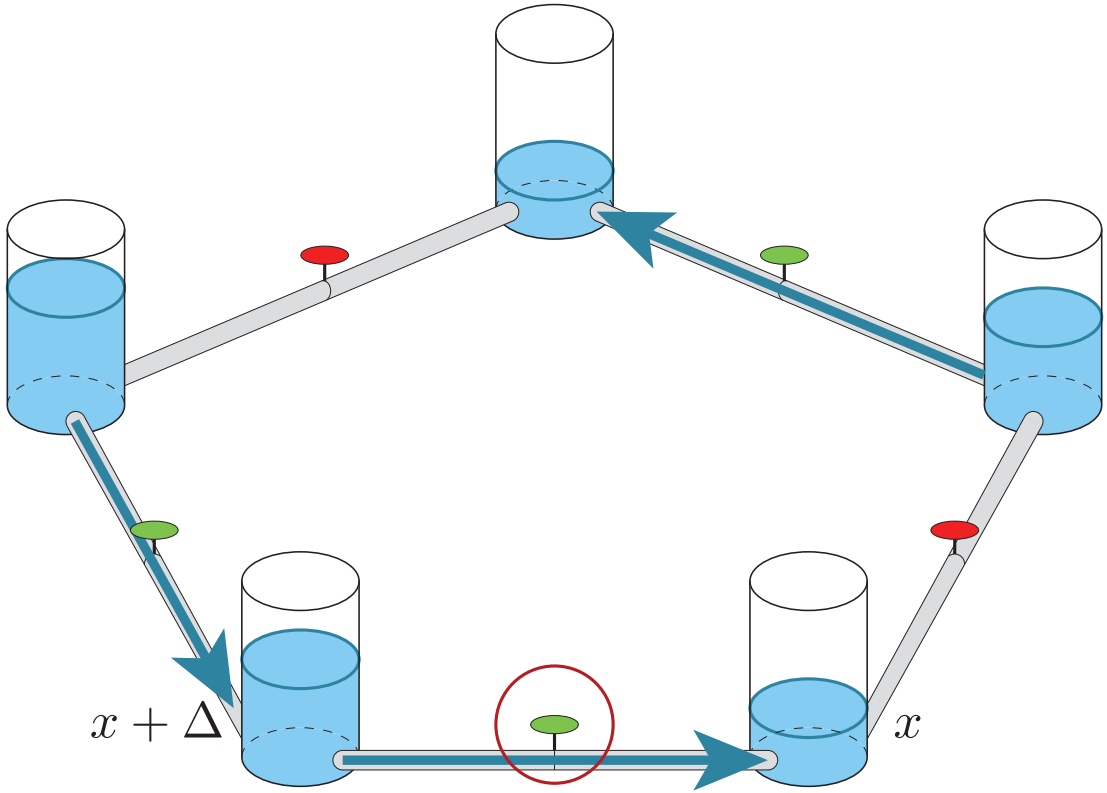
Proving the Communicating Vessels theorem



If only one valve is open:

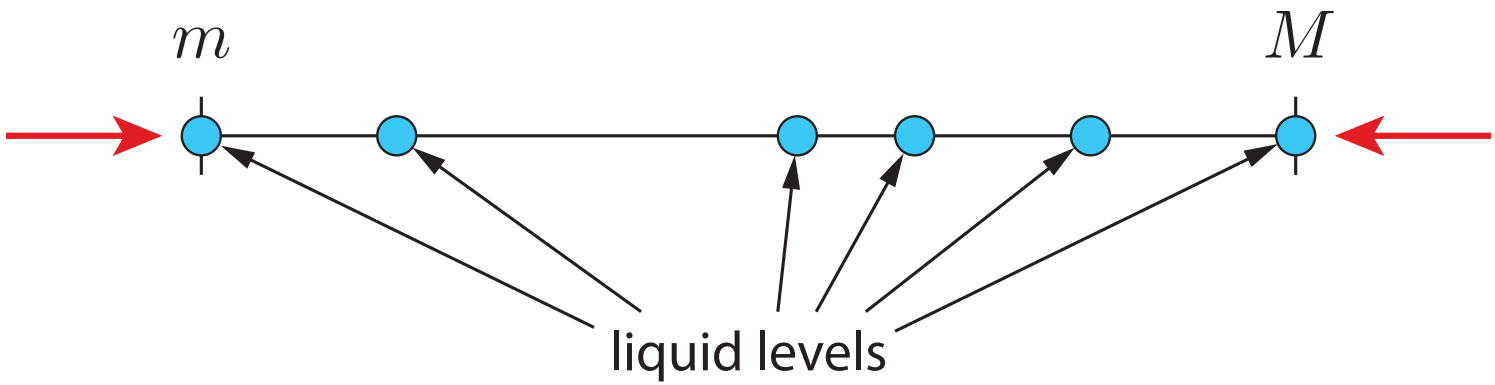
$E_t - E_{t+1} = ((x + \Delta)^2 + x^2) - ((x + \frac{3}{4}\Delta)^2 + (x + \frac{\Delta}{4})^2) = \frac{3}{8}\Delta^2$

Proving the Communicating Vessels theorem



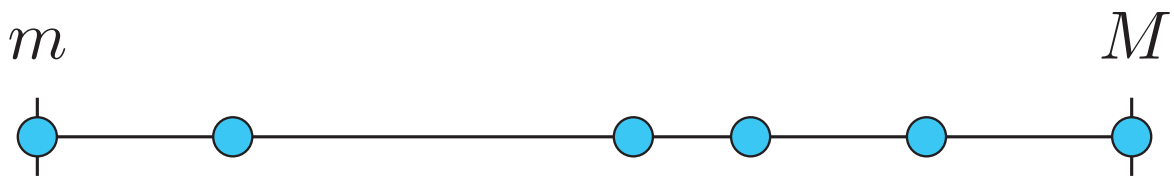
In general, if a valve is opened between two vessels with level difference Δ, then $E_t - E_{t+1} \geqslant \frac{\Delta^2}{4}$

Proving the Communicating Vessels theorem



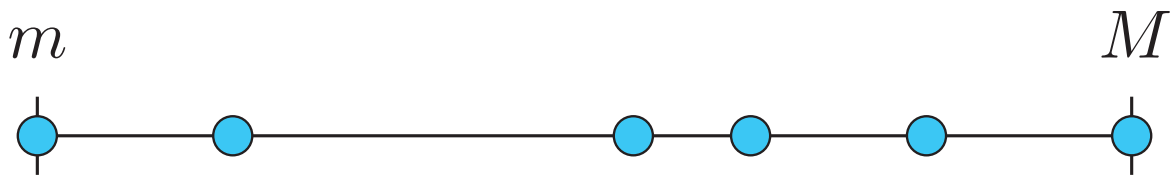
Observation: the minimum level can only increase and the maximum level can only decrease, hence they have limits m and M

Proving the Communicating Vessels theorem



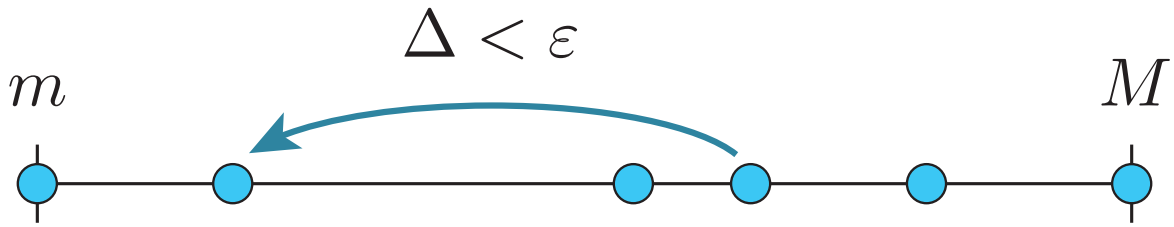
Observation: the variance is non-negative and can only decrease; hence, from some time on, it can change only by less than $\varepsilon^2/4$

Proving the Communicating Vessels theorem



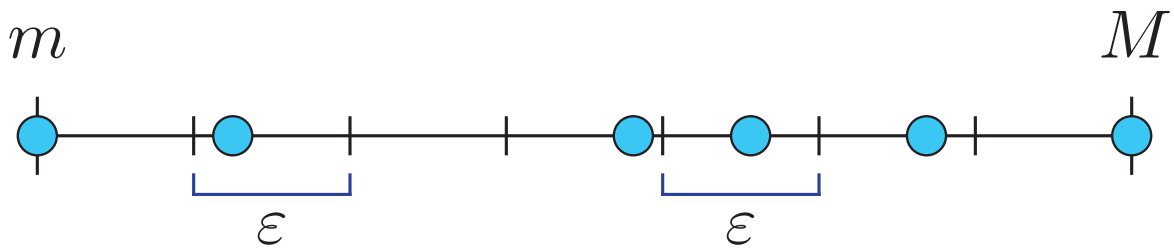
But $\Delta^2/4 \leqslant E_t - E_{t+1} < \varepsilon^2/4$, implying $\Delta < \varepsilon$

Proving the Communicating Vessels theorem



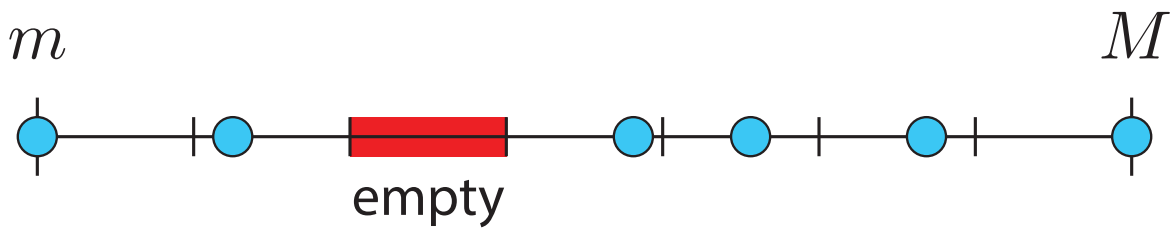
From a certain time on, if a valve is opened between two vessels, their level difference Δ must be less than ε

Proving the Communicating Vessels theorem



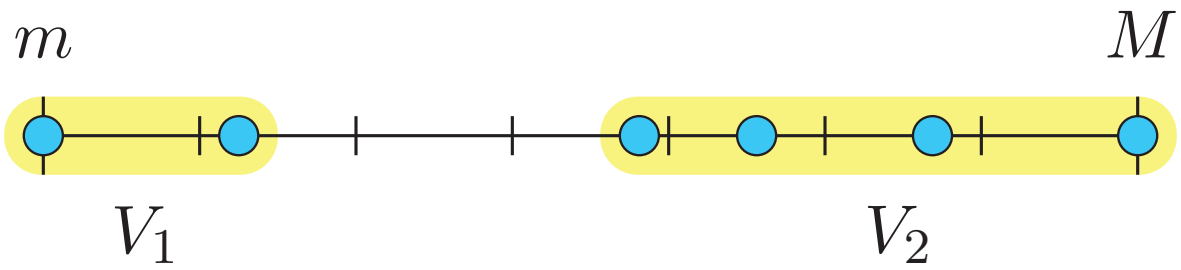
Assume by contradiction that $m < M$, and let $\varepsilon := \frac{M-m}{n+1}$

Proving the Communicating Vessels theorem



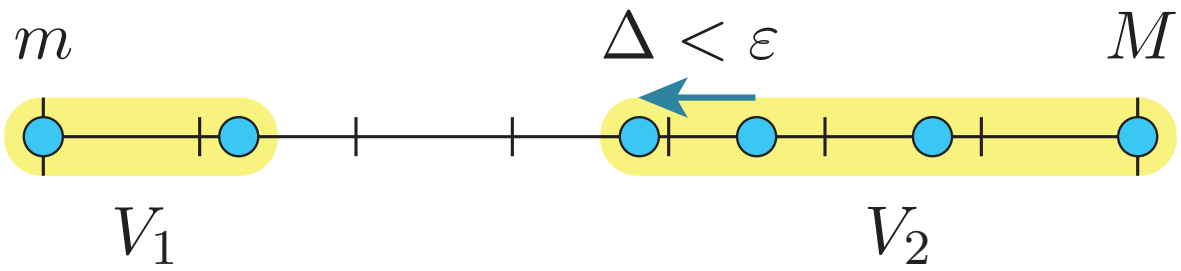
By the pigeonhole principle, none of the n levels of liquid falls in some ε-interval

Proving the Communicating Vessels theorem



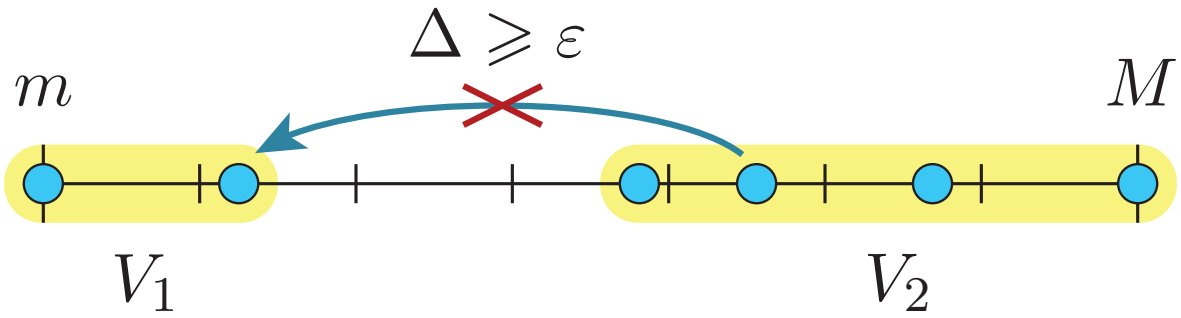
The liquid levels are split in two classes: lower than the empty interval and higher than the empty interval

Proving the Communicating Vessels theorem



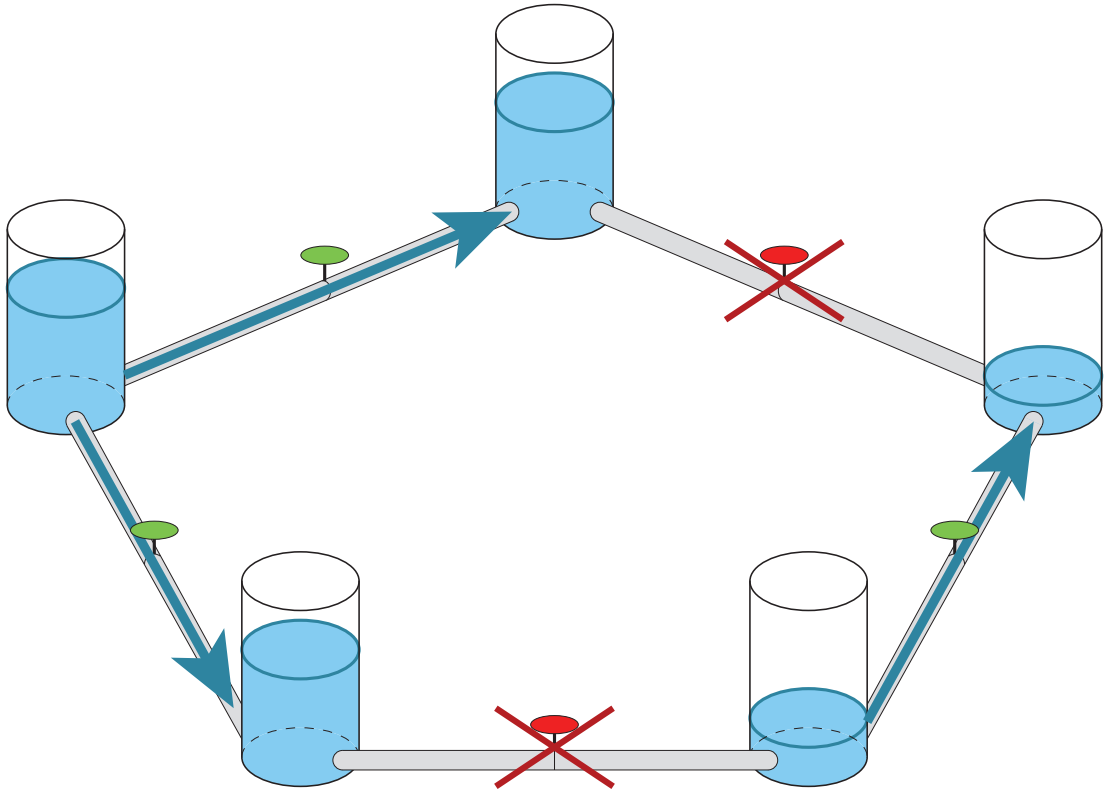
Only valves between vessels whose levels are in the same class can be opened

Proving the Communicating Vessels theorem



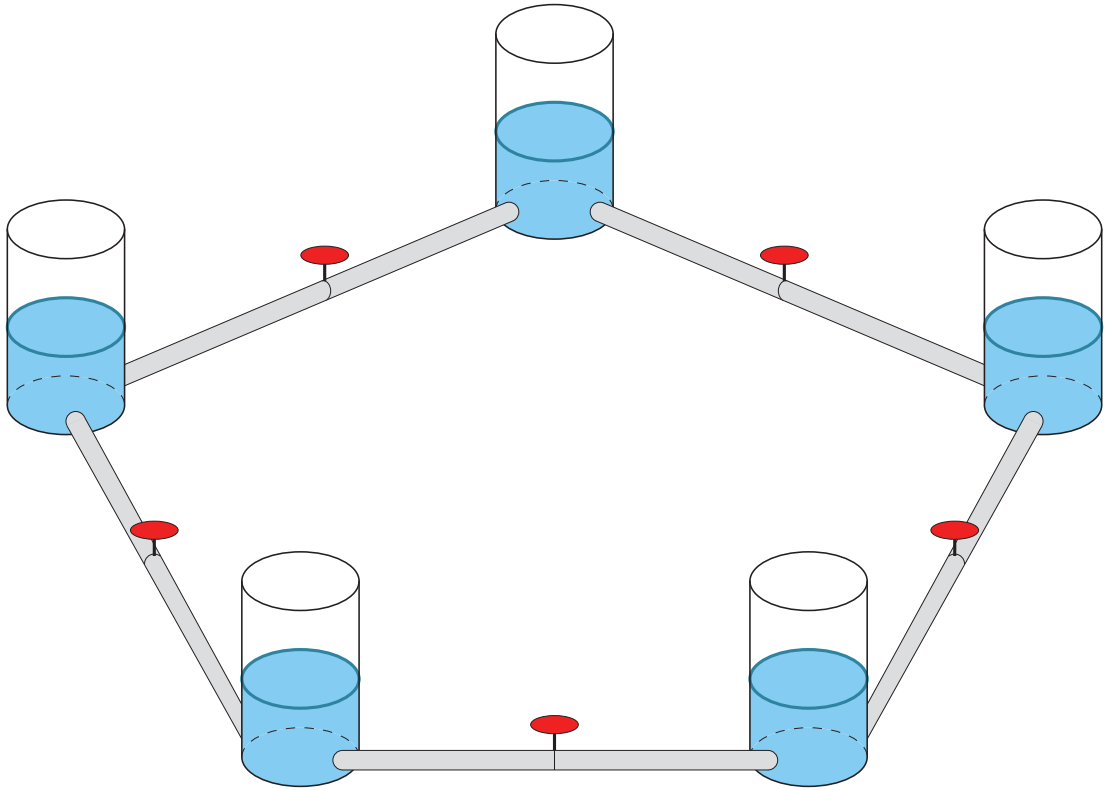
Only valves between vessels whose levels are in the same class can be opened

Proving the Communicating Vessels theorem



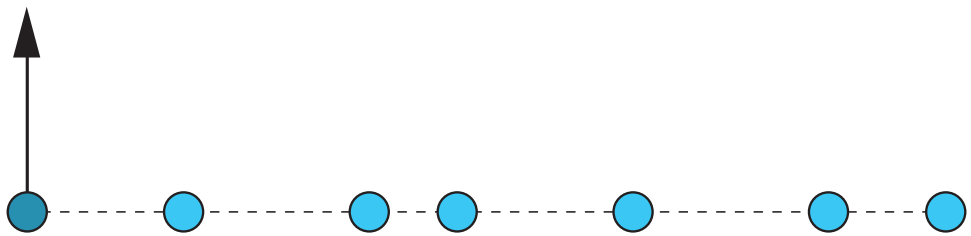
This implies that at least two valves are never open from some time on, contradicting the fairness assumption

Proving the Communicating Vessels theorem



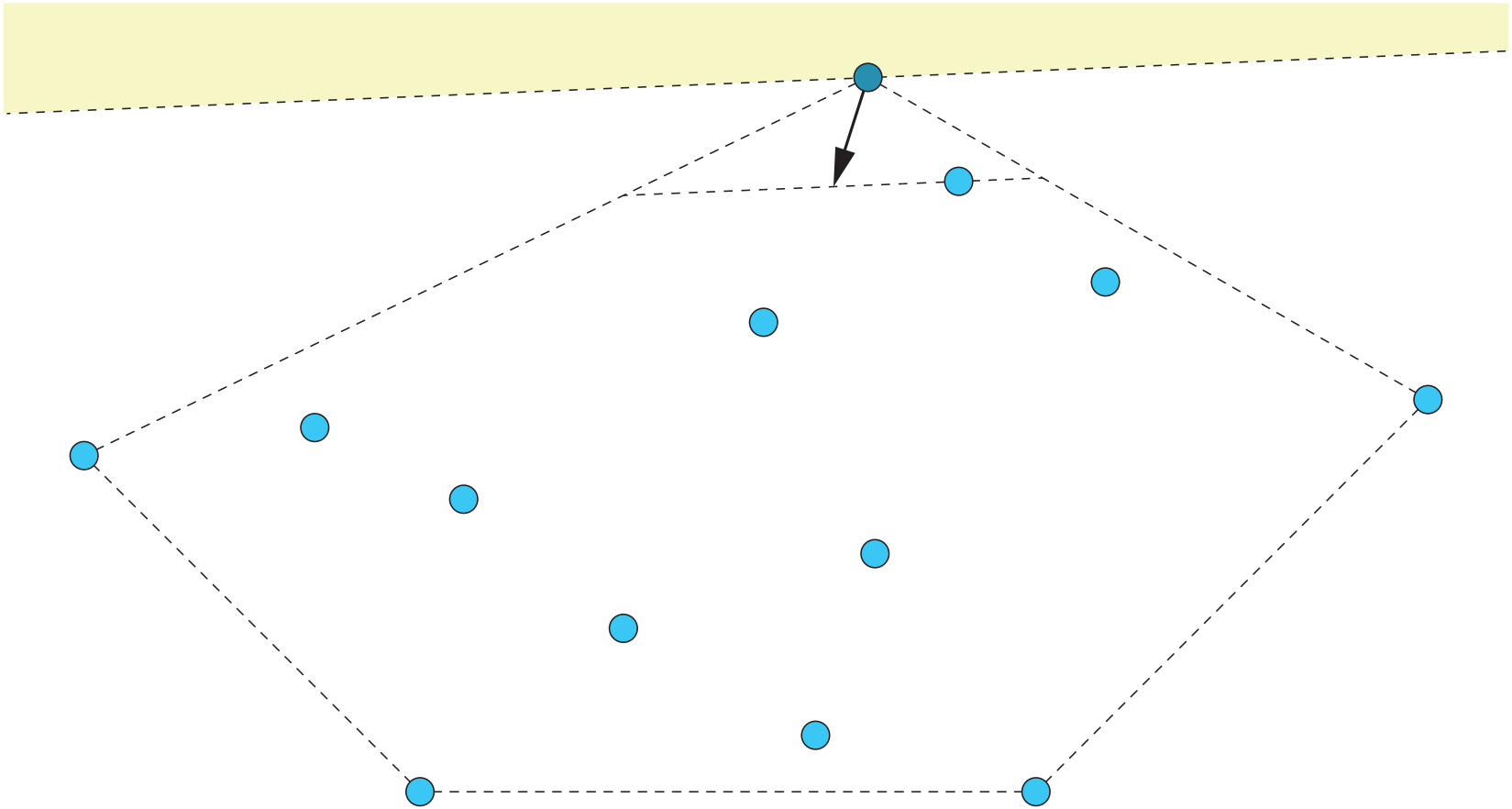
Conclusion: the liquid levels converge to the same limit

Correctness of the algorithm (cont.)



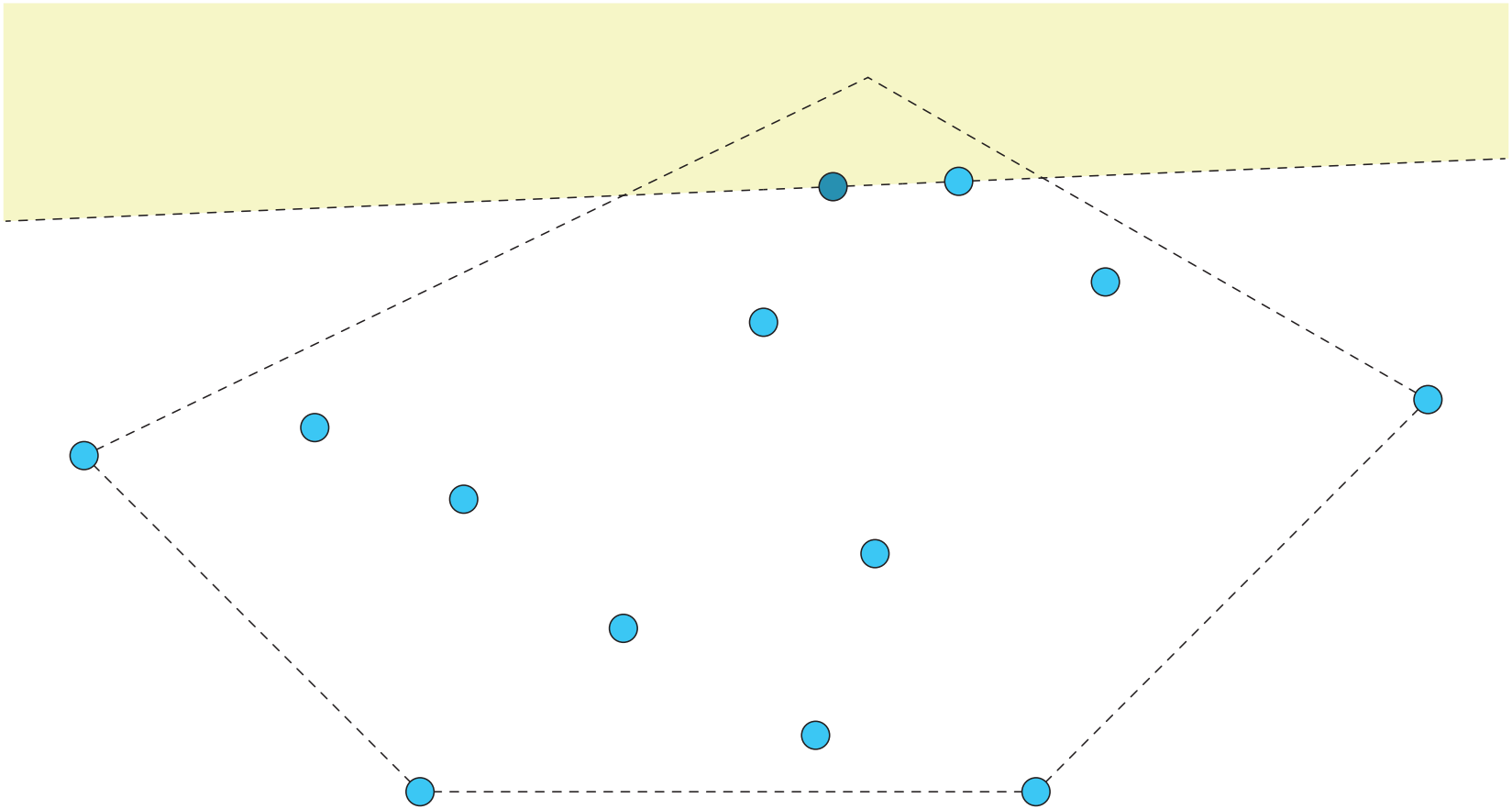
The collinear case is immediately resolved and never occurs again

Correctness of the algorithm (cont.)



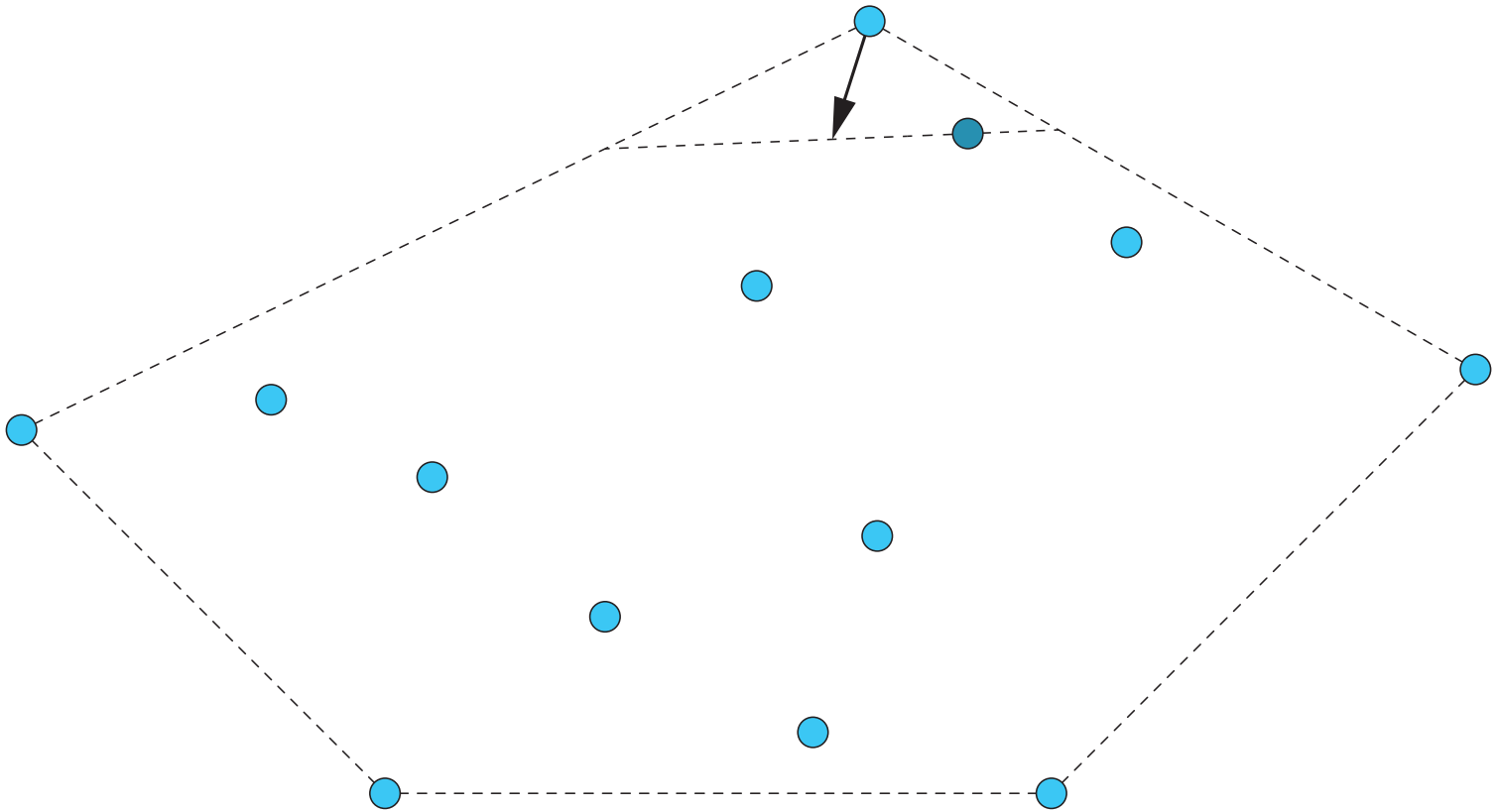
Each external robots remains external forever

Correctness of the algorithm (cont.)



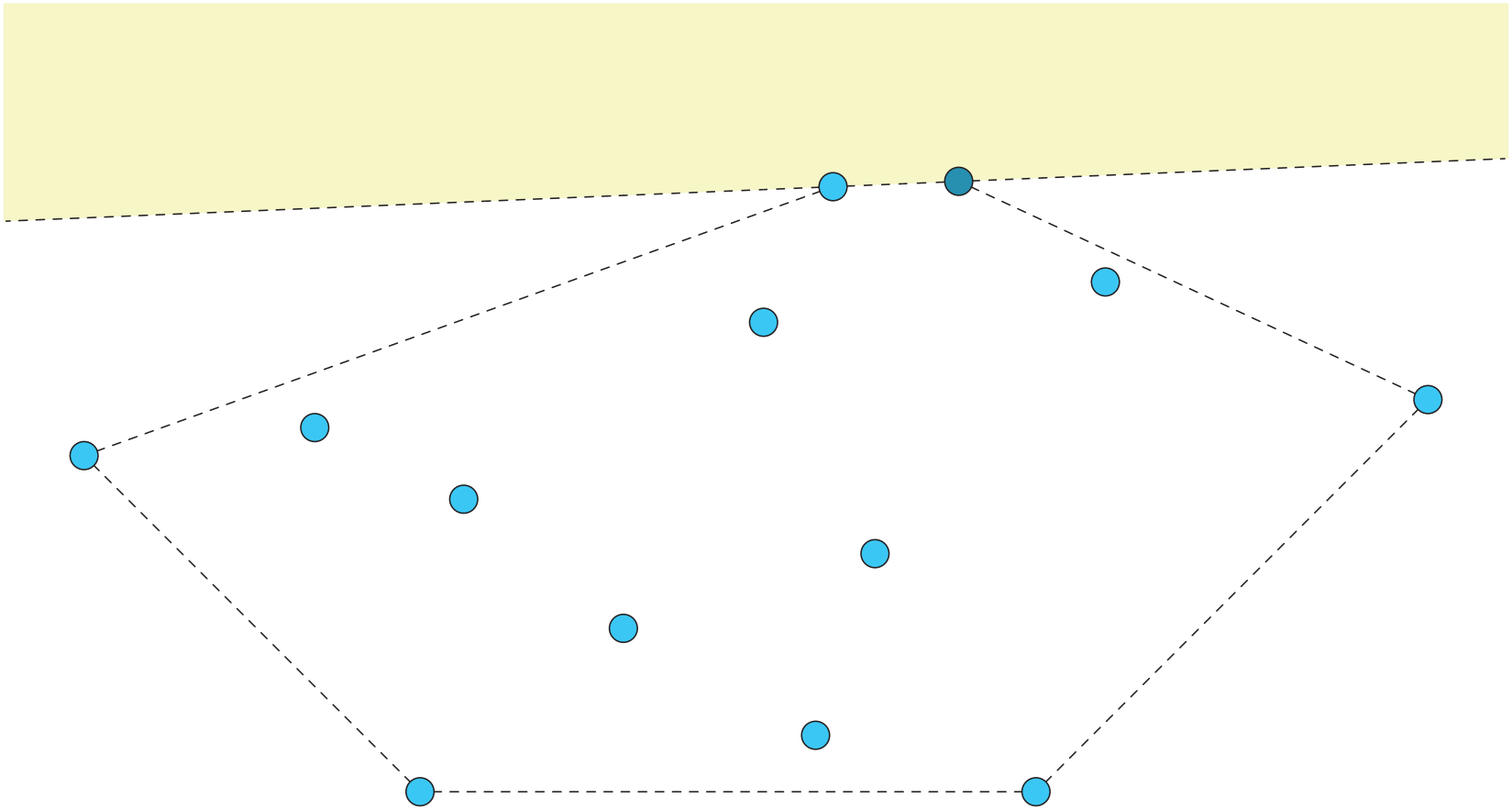
Each external robots remains external forever

Correctness of the algorithm (cont.)



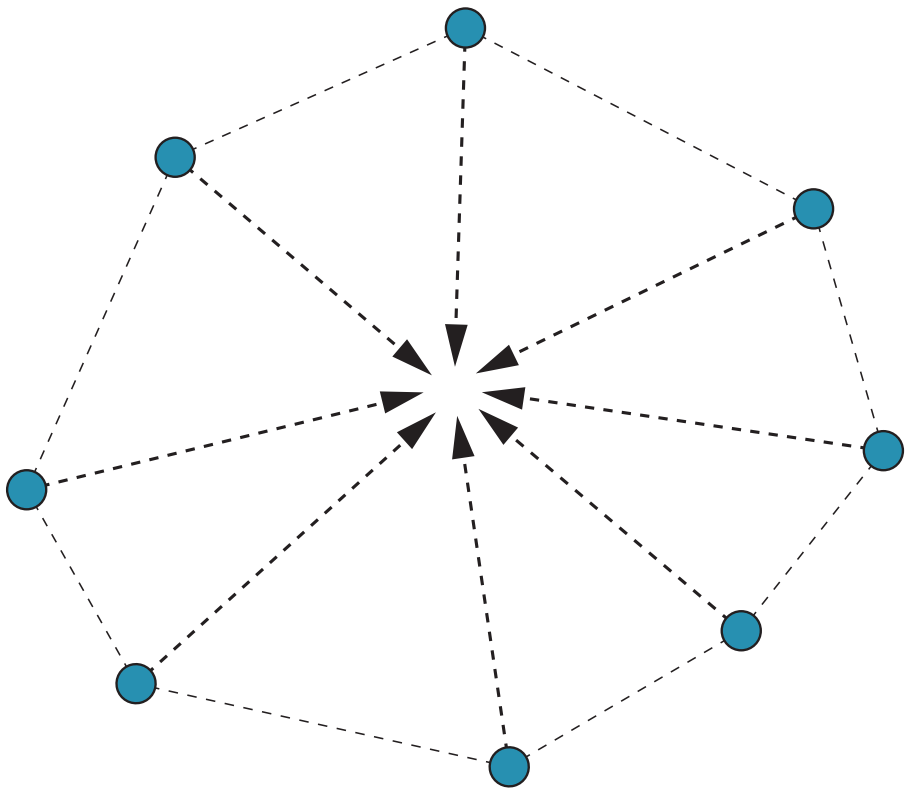
If a non-default move is made, a new robot becomes external

Correctness of the algorithm (cont.)



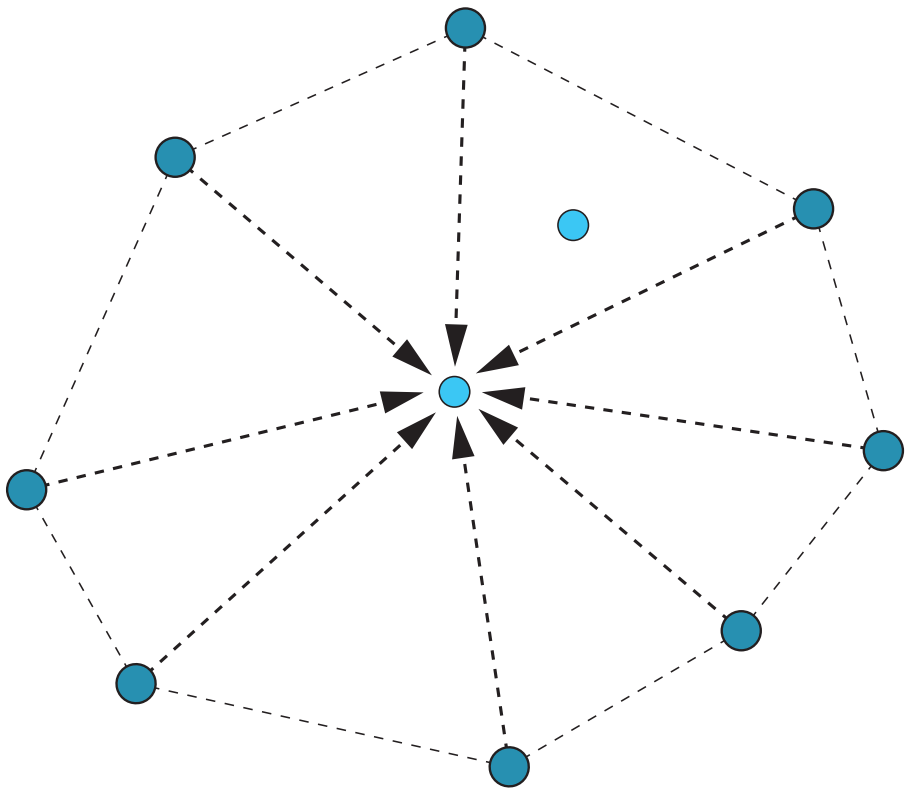
If a non-default move is made, a new robot becomes external

Correctness of the algorithm (cont.)



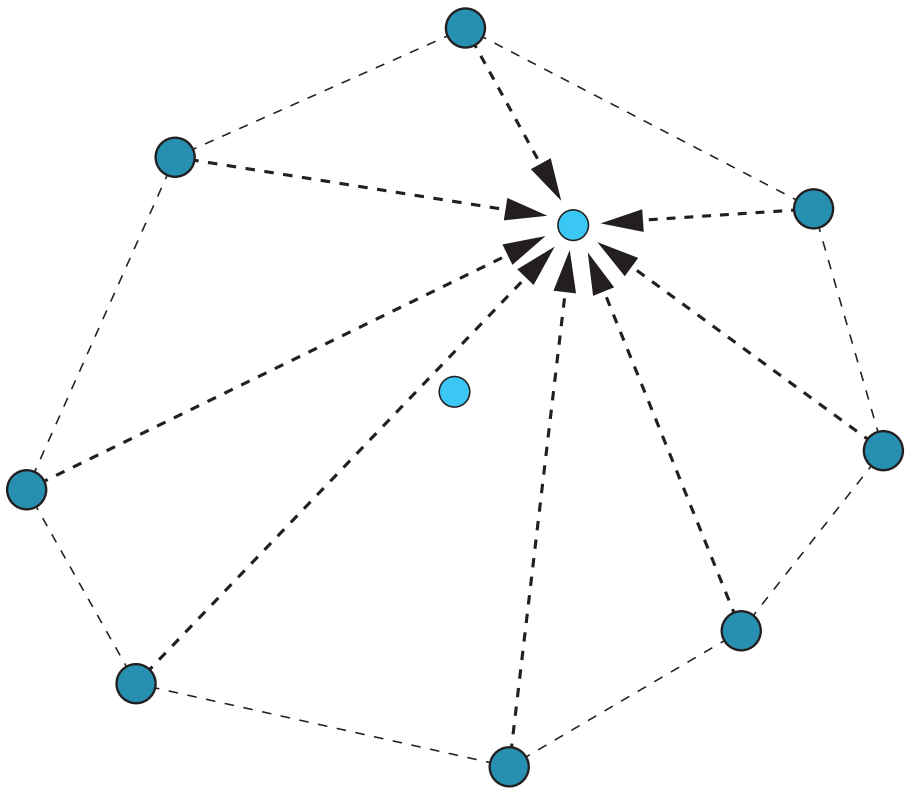
If default moves keep being made, the robots converge to a point

Correctness of the algorithm (cont.)



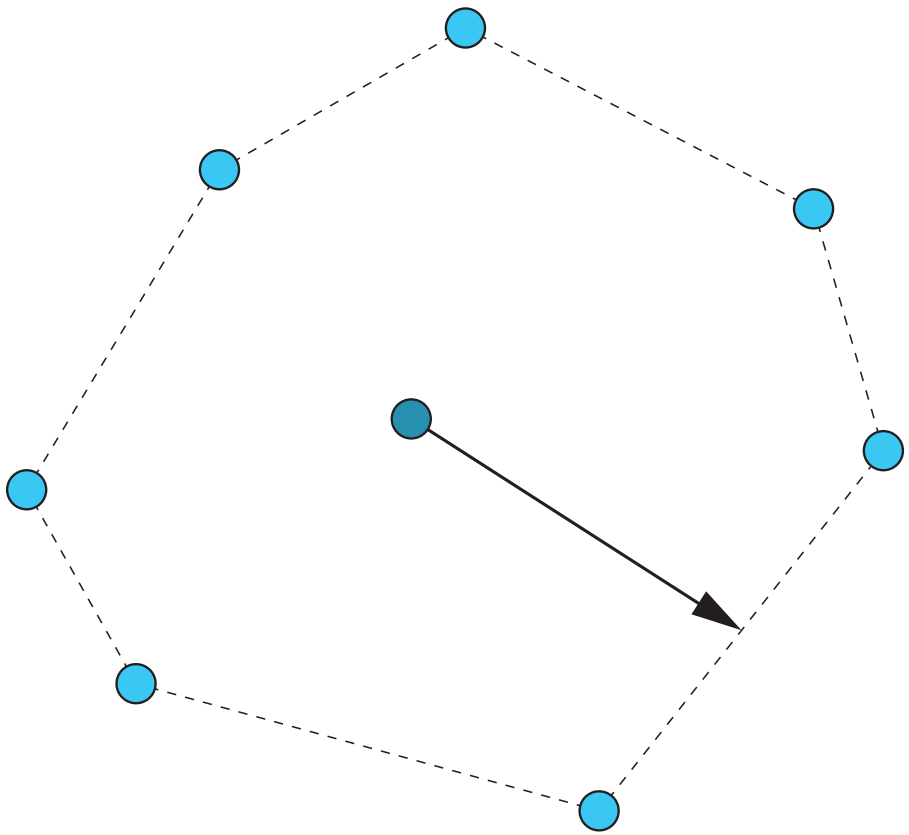
But if there are two internal robots, the external cannot converge

Correctness of the algorithm (cont.)



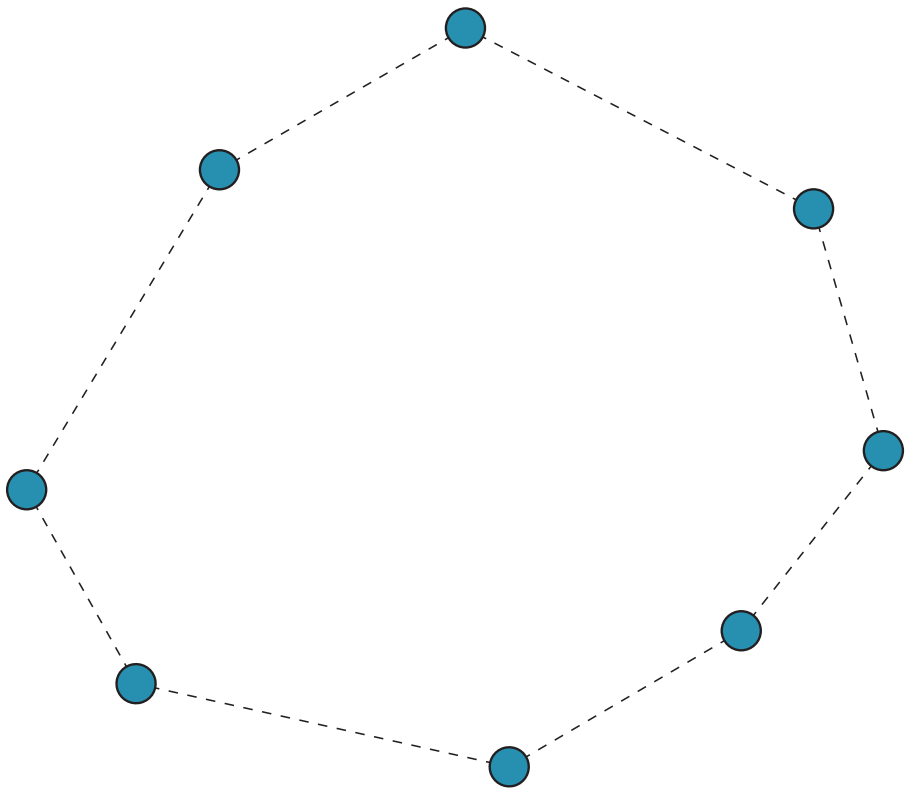
But if there are two internal robots, the external cannot converge

Correctness of the algorithm (cont.)



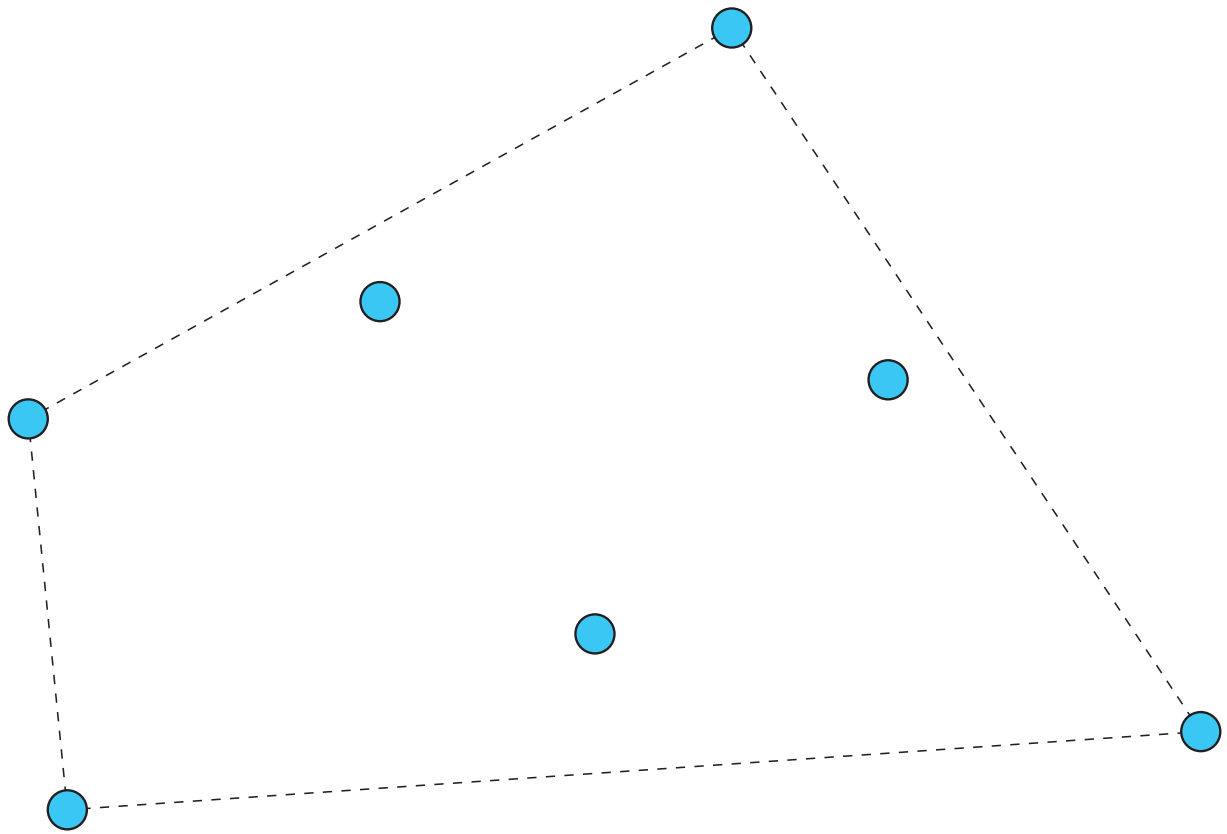
If there is only one internal robot, it moves to the boundary

Correctness of the algorithm (cont.)



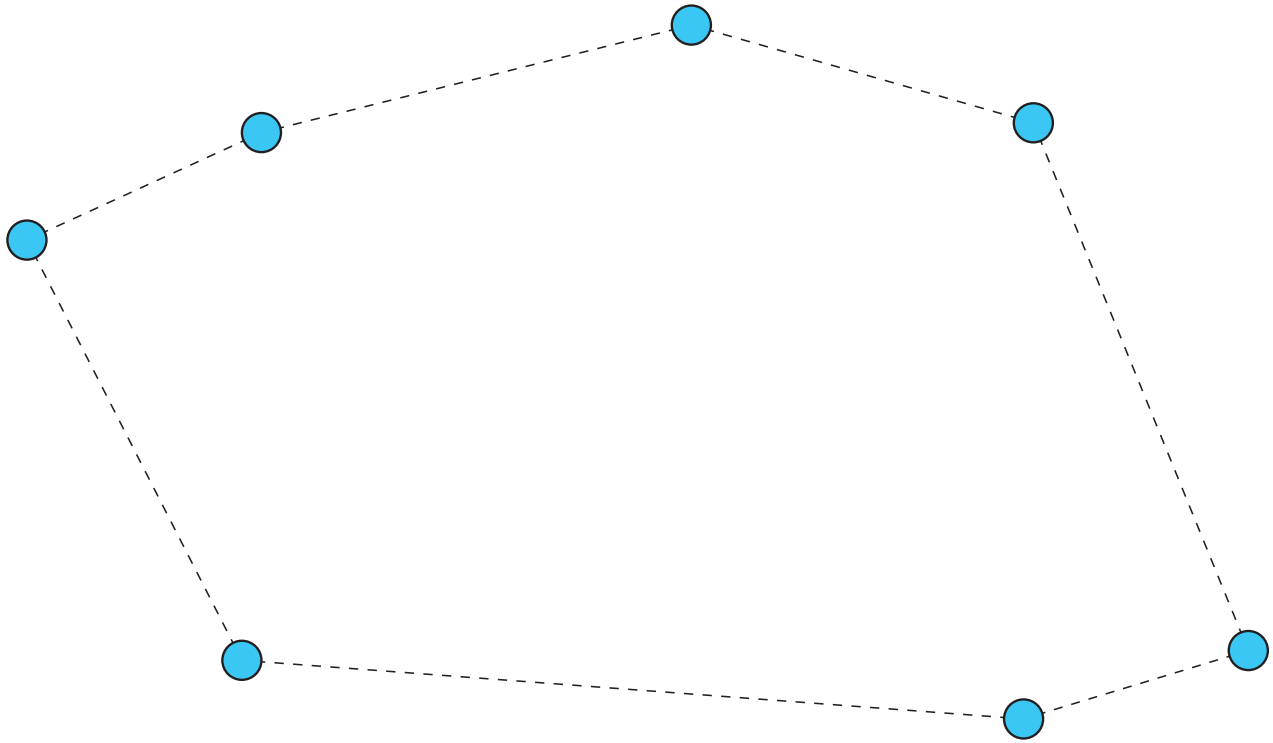
When the robots are all external, they terminate

Related problem: Convex Formation



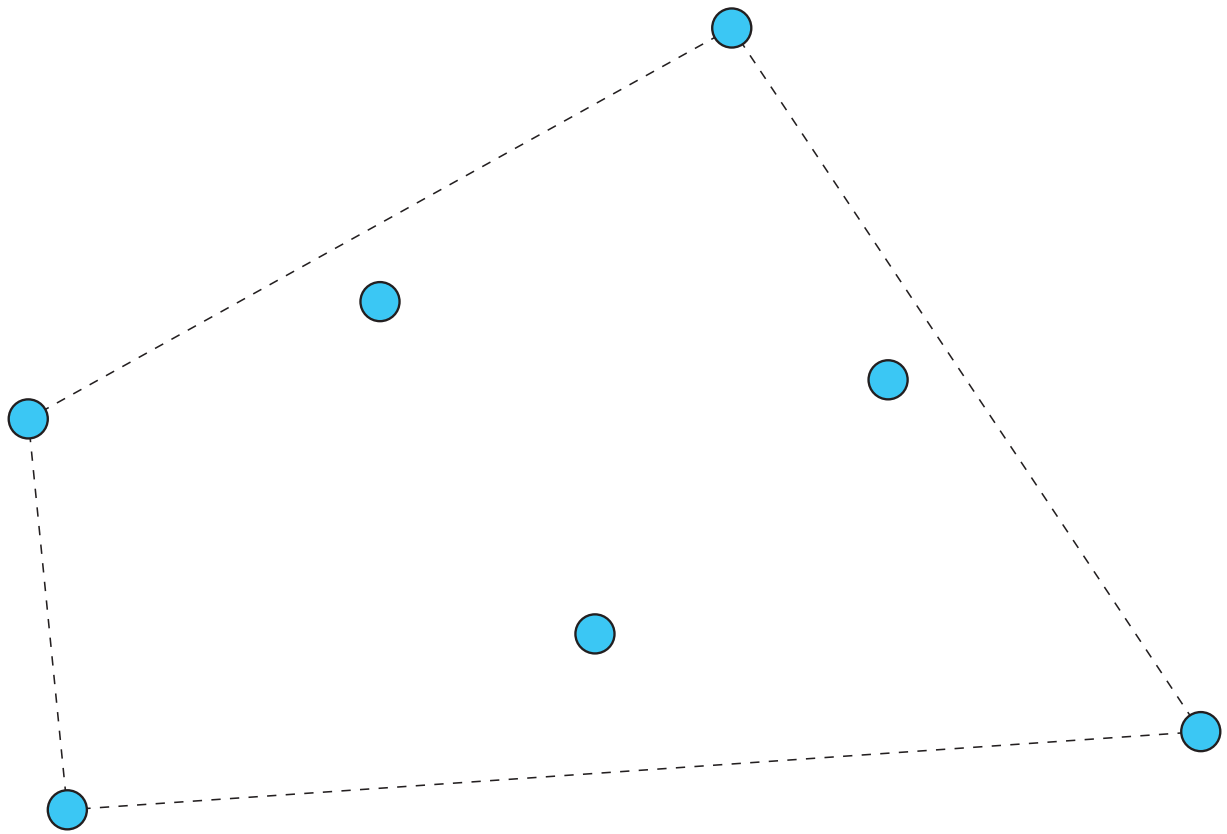
Goal: reach a strictly convex configuration

Related problem: Convex Formation



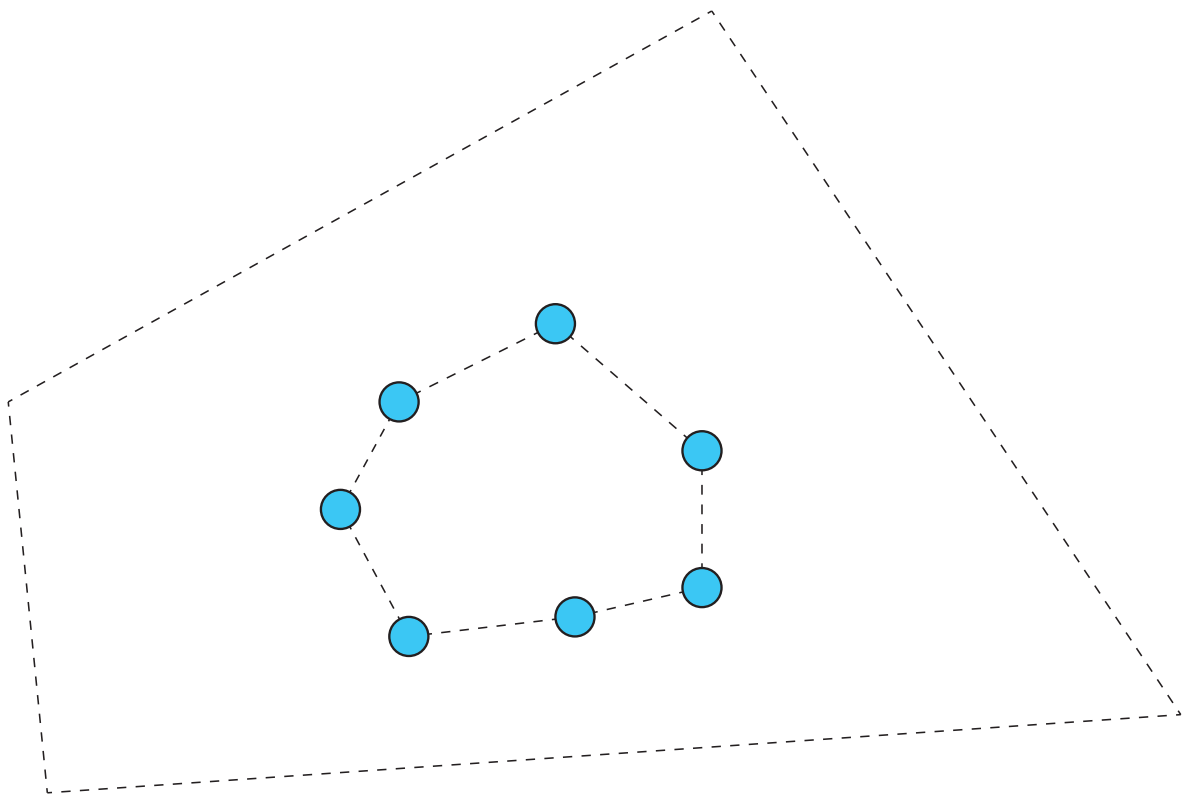
Goal: reach a strictly convex configuration

Related problem: Convex Formation



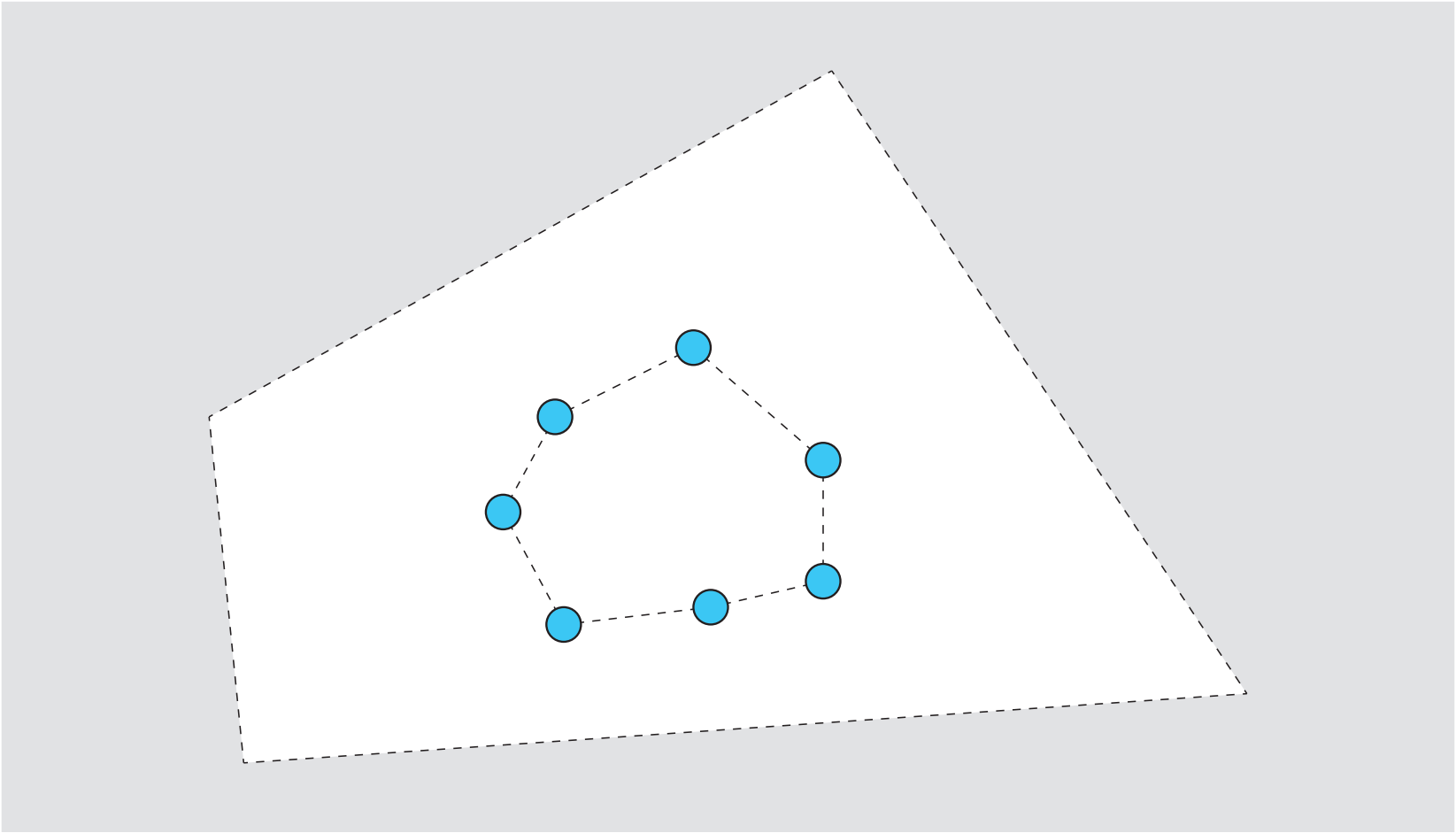
Solution: apply the algorithm for Mutual Visibility

Related problem: Convex Formation



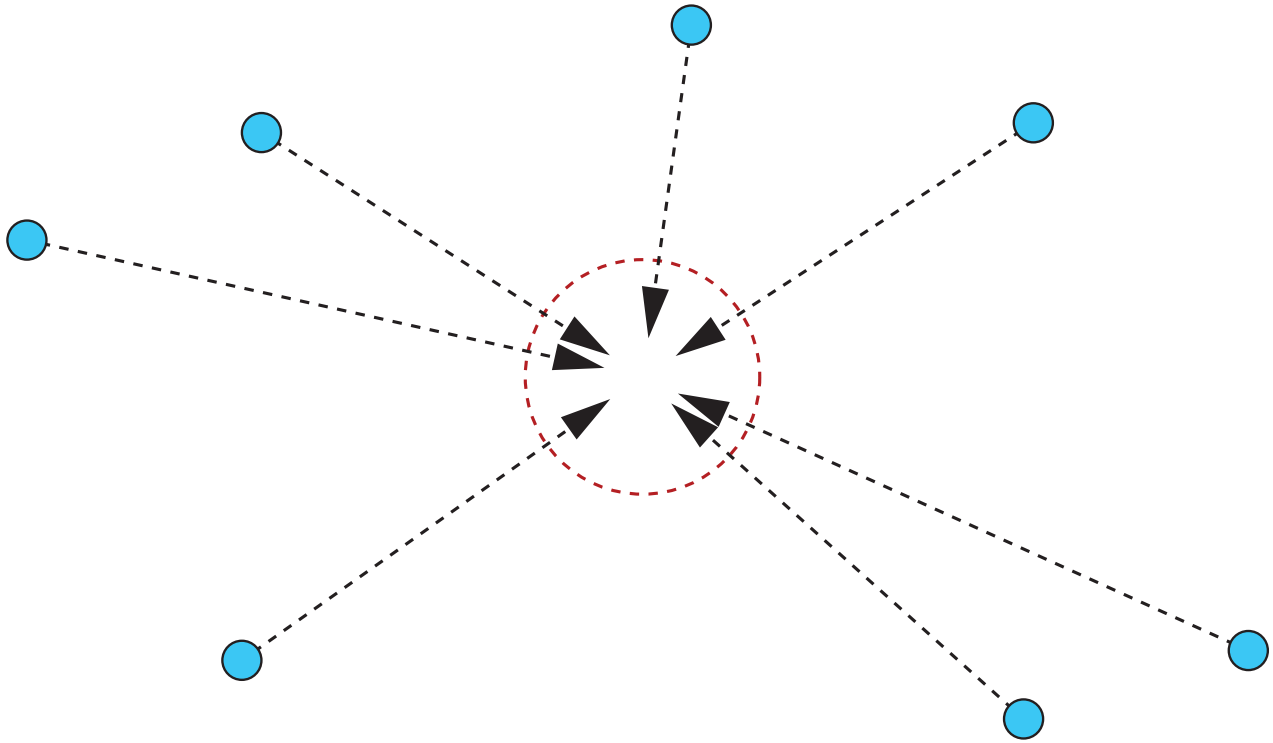
Solution: apply the algorithm for Mutual Visibility

Related problem: Convex Formation



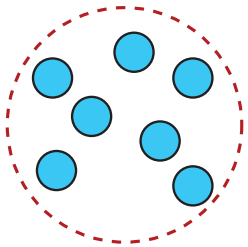
Bonus: robots never get out of the initial convex hull

Related problem: Near-Gathering



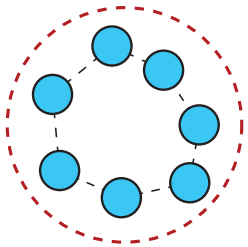
Goal: gather in a small-enough area without ever colliding

Related problem: Near-Gathering



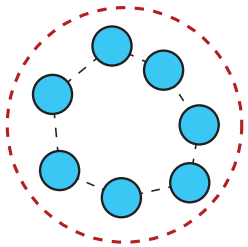
Goal: gather in a small-enough area without ever colliding

Related problem: Near-Gathering



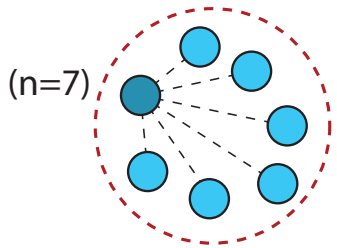
Solution: apply our algorithm but remove the termination check

Related problem: Near-Gathering



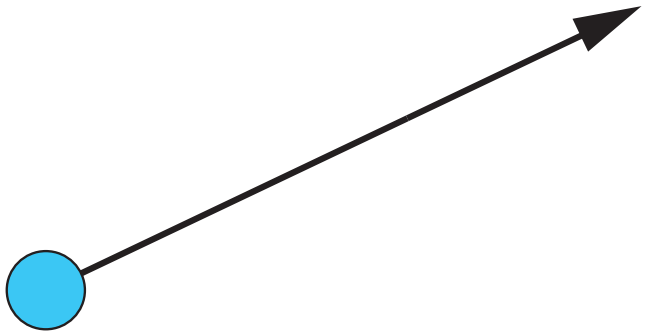
Bonus: the knowledge of n is not needed

Related problem: Near-Gathering



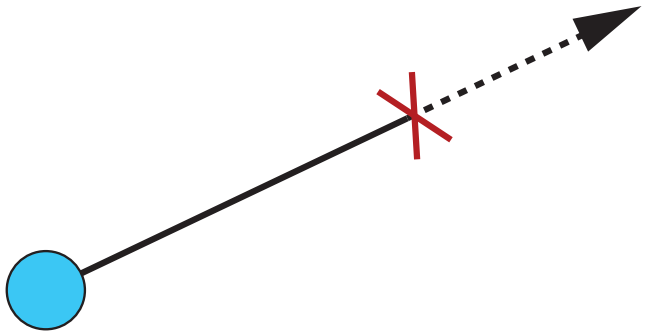
Bonus: if the robots know n, they can also terminate

Related model: unreliable moves



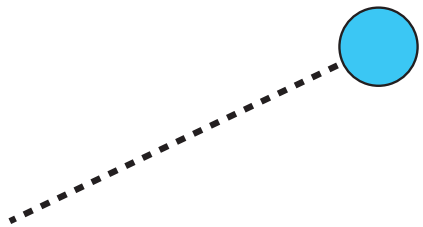
Model: robots may be stopped before reaching their destination

Related model: unreliable moves



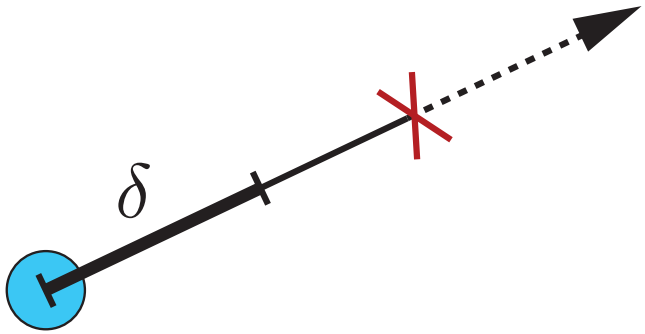
Model: robots may be stopped before reaching their destination

Related model: unreliable moves



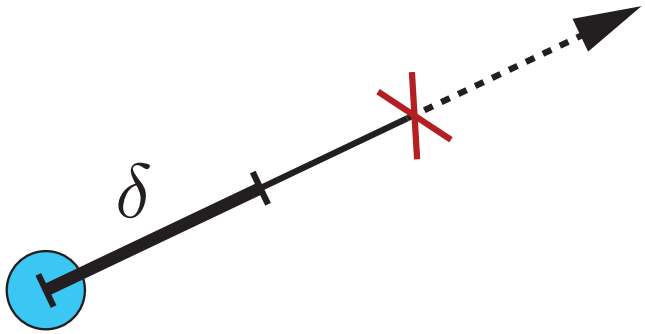
Model: robots may be stopped before reaching their destination

Related model: unreliable moves



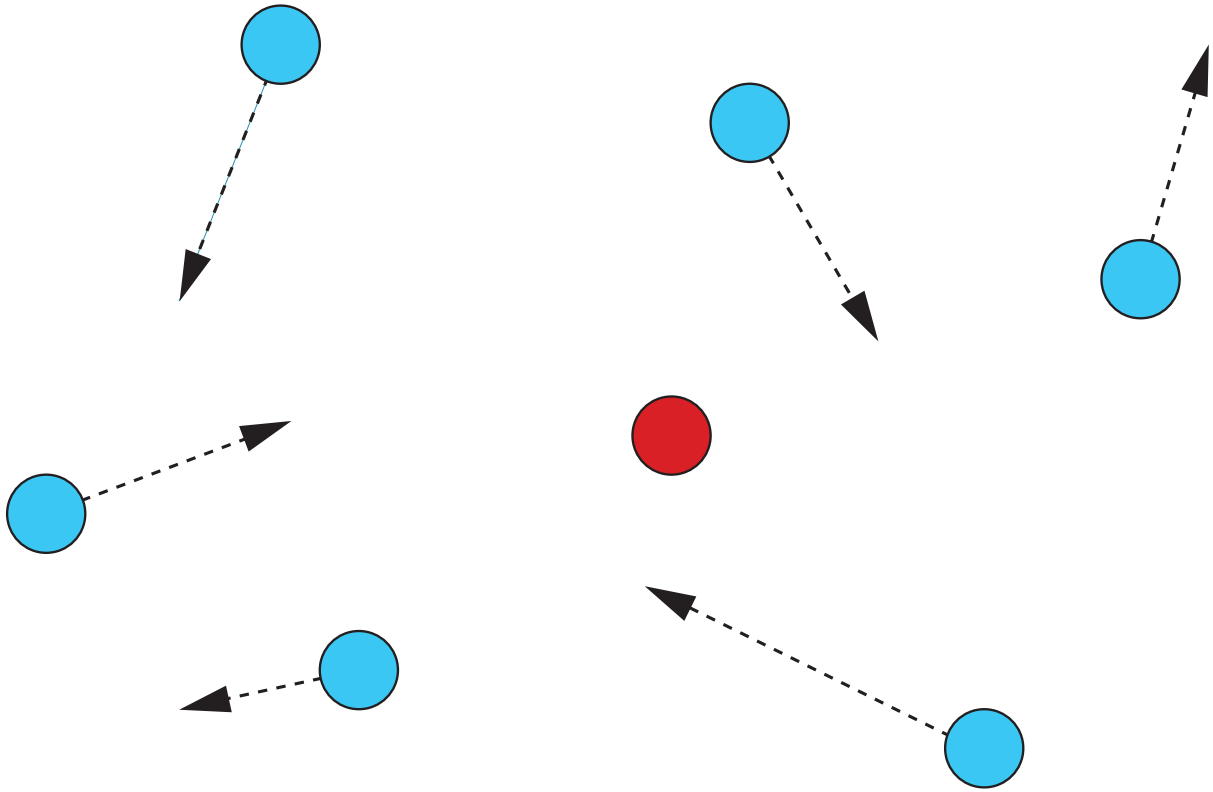
Model: robots may be stopped before reaching their destination
Fairness: robots are guaranteed to move by at least δ at each turn

Related model: unreliable moves



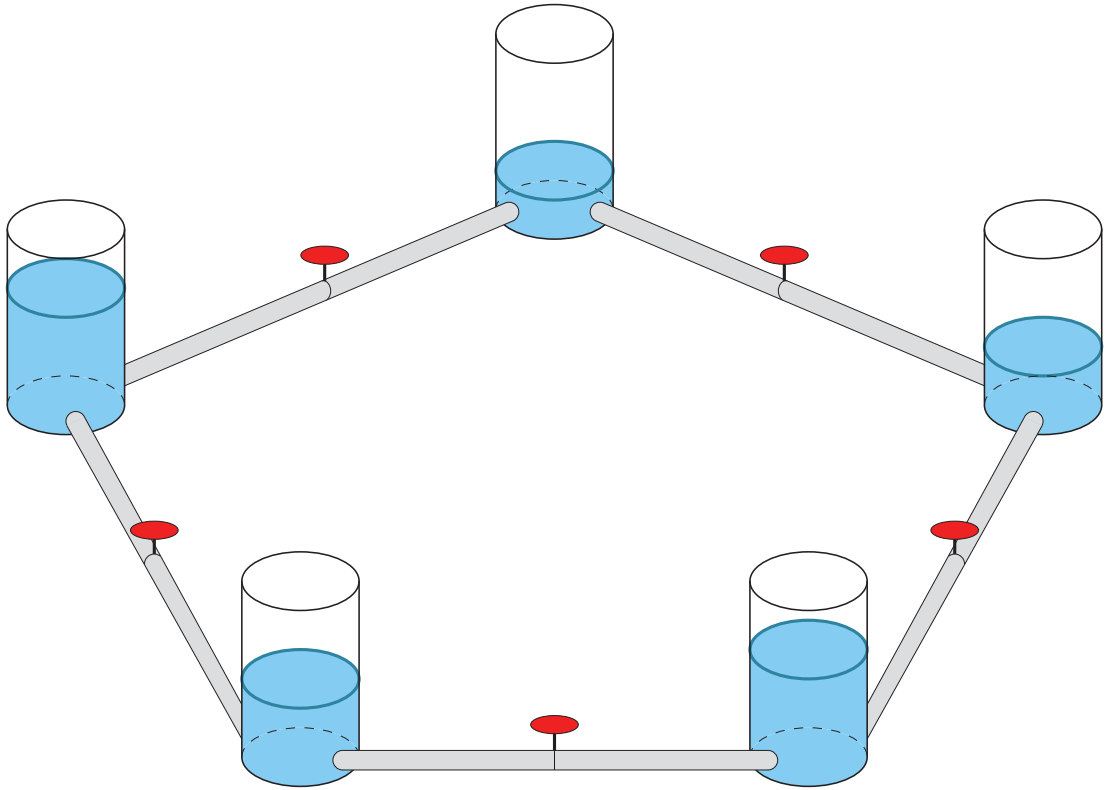
Solution: apply our algorithm to solve Mutual Visibility, Convex Formation, and Near-Gathering

Related model: one faulty robot



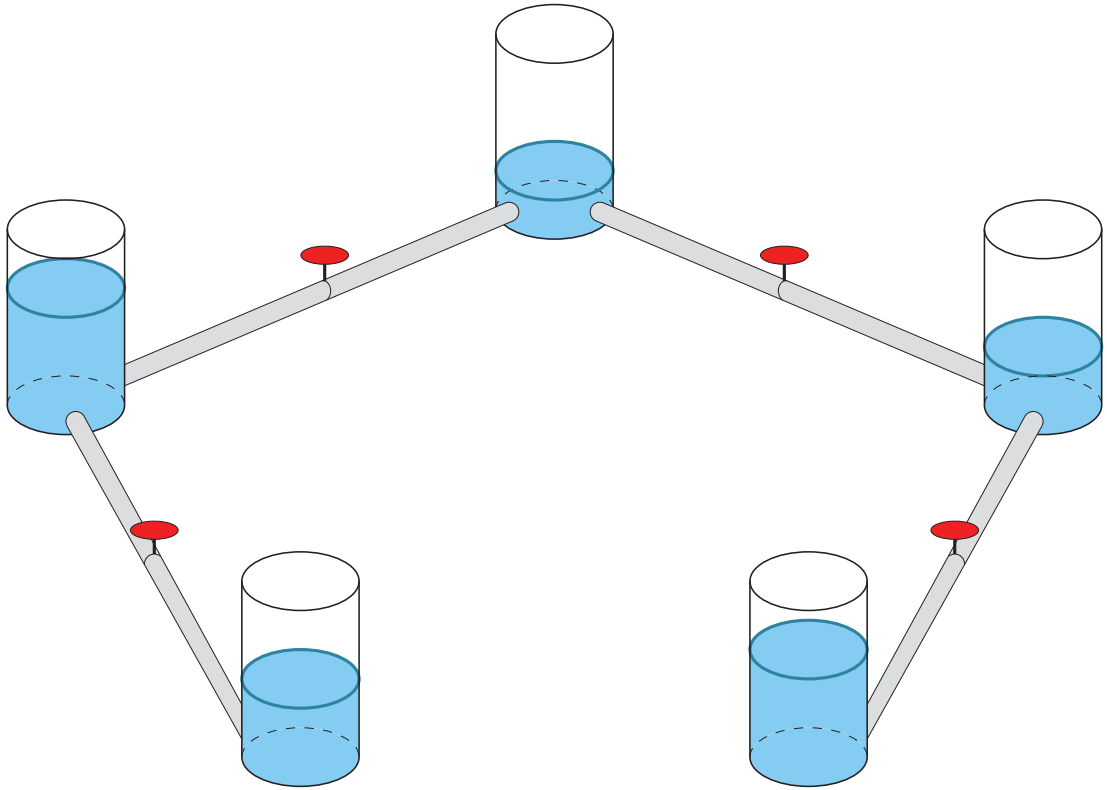
Model: one robot may malfunction and become unable to move

Related model: one faulty robot



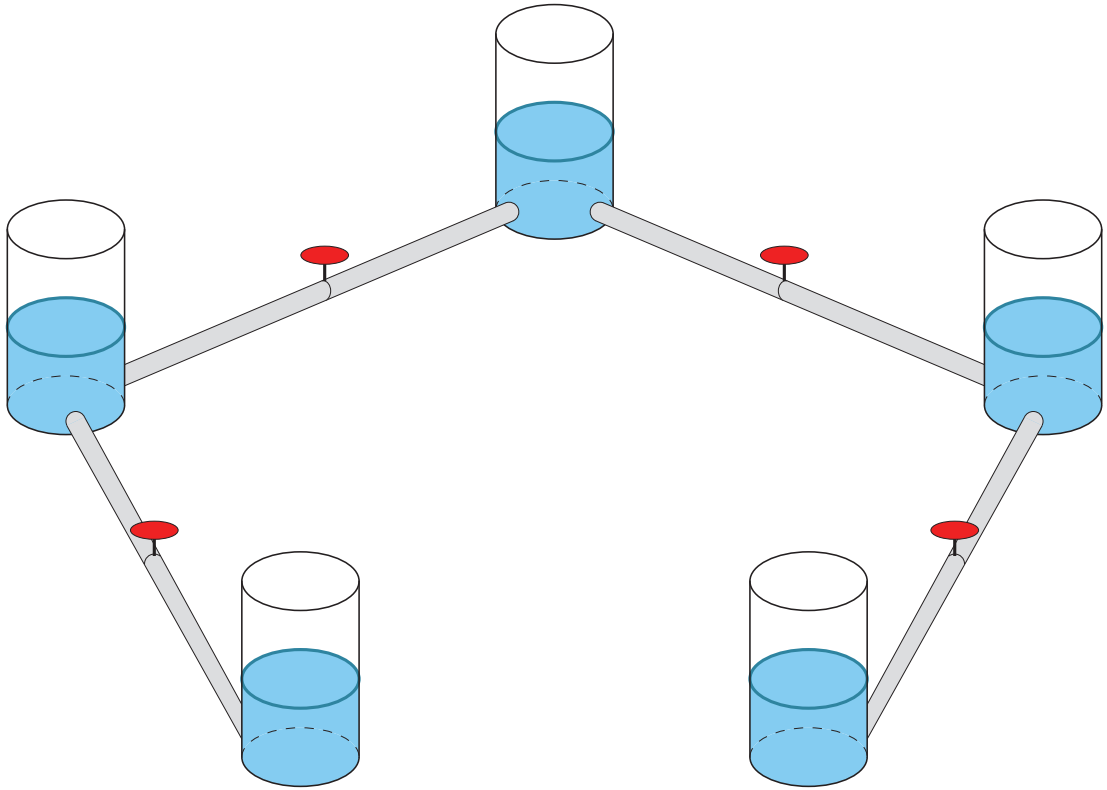
Observation: the proof of the COMMUNICATING VESSELS theorem goes through even if we remove one pipe

Related model: one faulty robot



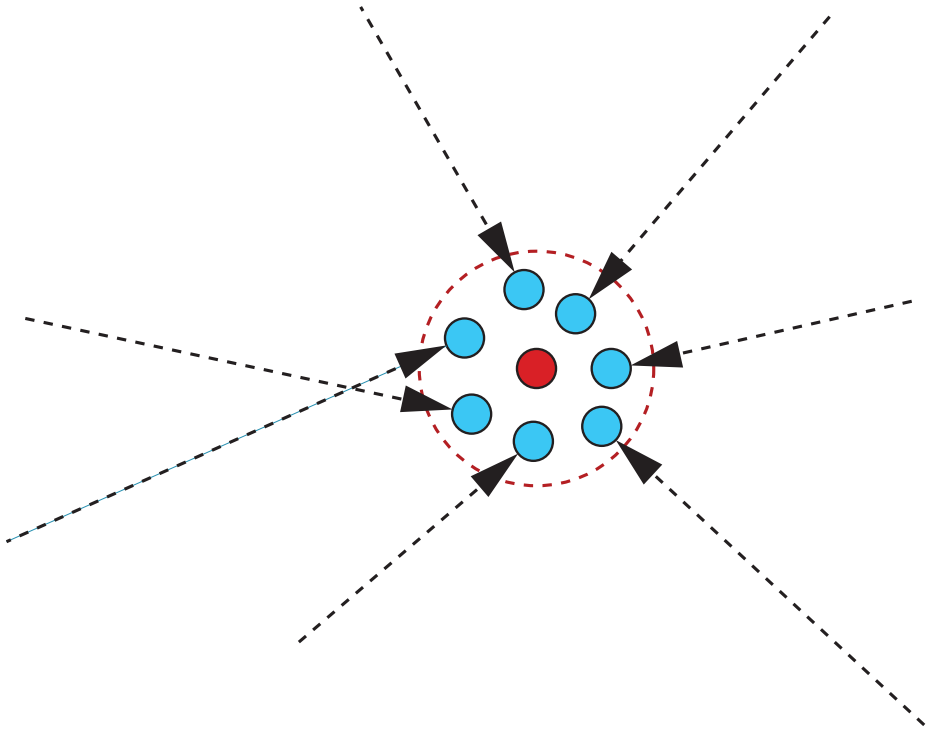
Observation: the proof of the COMMUNICATING VESSELS theorem goes through even if we remove one pipe

Related model: one faulty robot



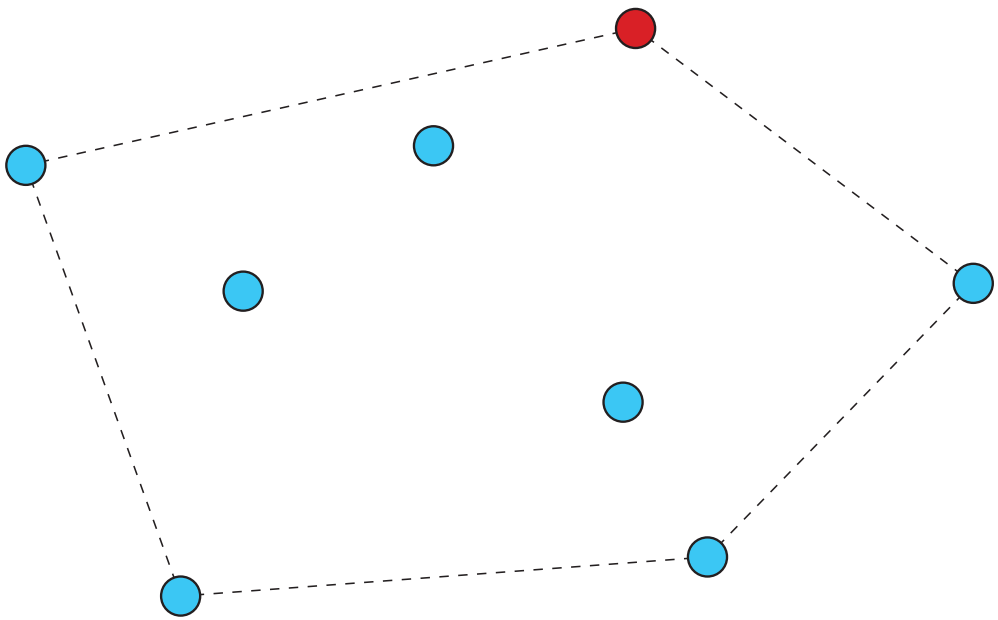
Observation: the proof of the COMMUNICATING VESSELS theorem goes through even if we remove one pipe

Related model: one faulty robot



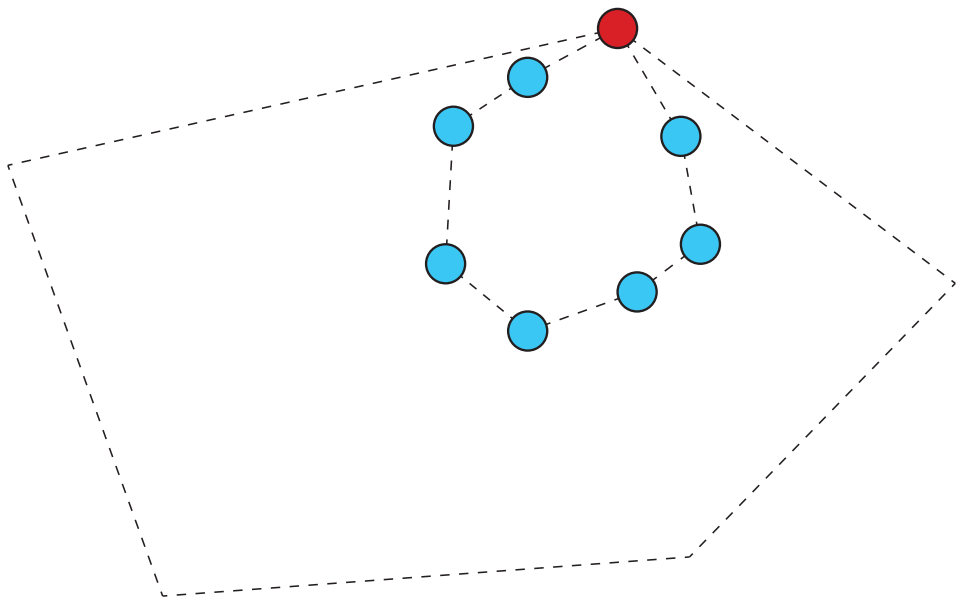
Solution: our algorithm still solves NEAR-GATHERING; the robots converge around the faulty robot

Related model: one faulty robot



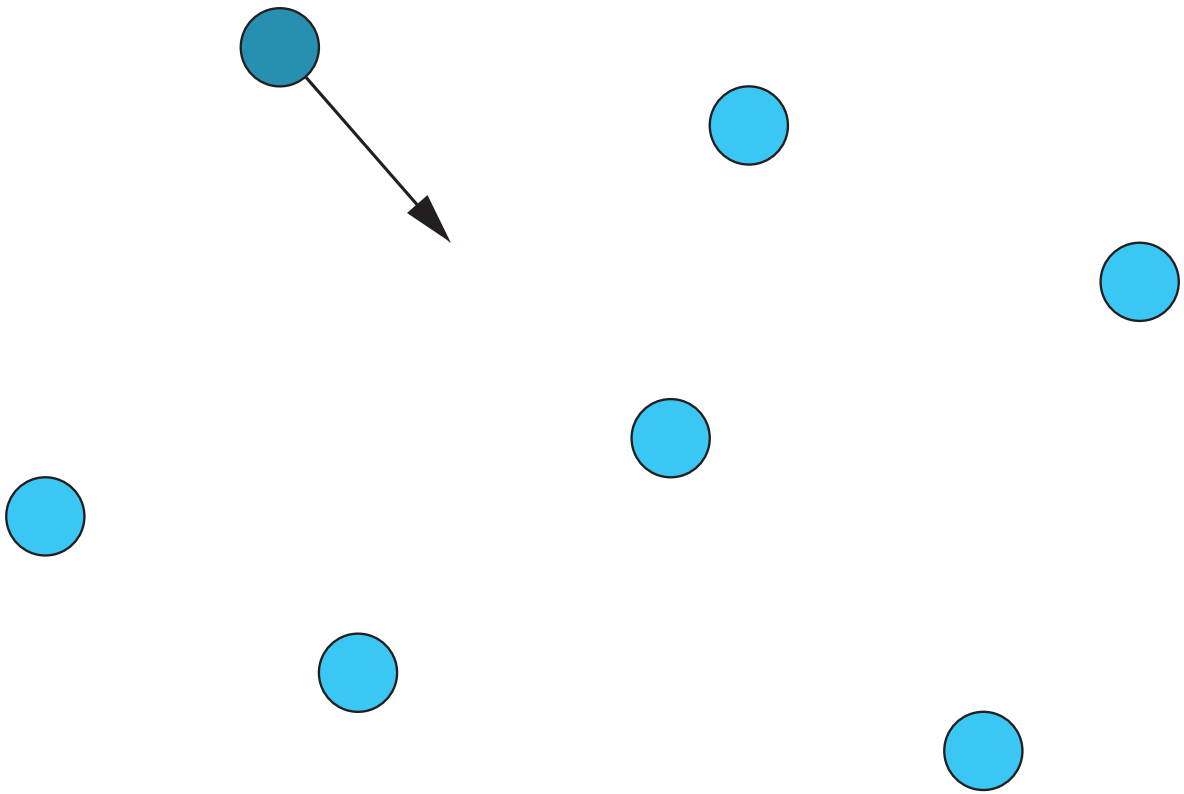
Bonus: it also solves Mutual Visibility and Convex Formation, provided that the faulty robot is external

Related model: one faulty robot



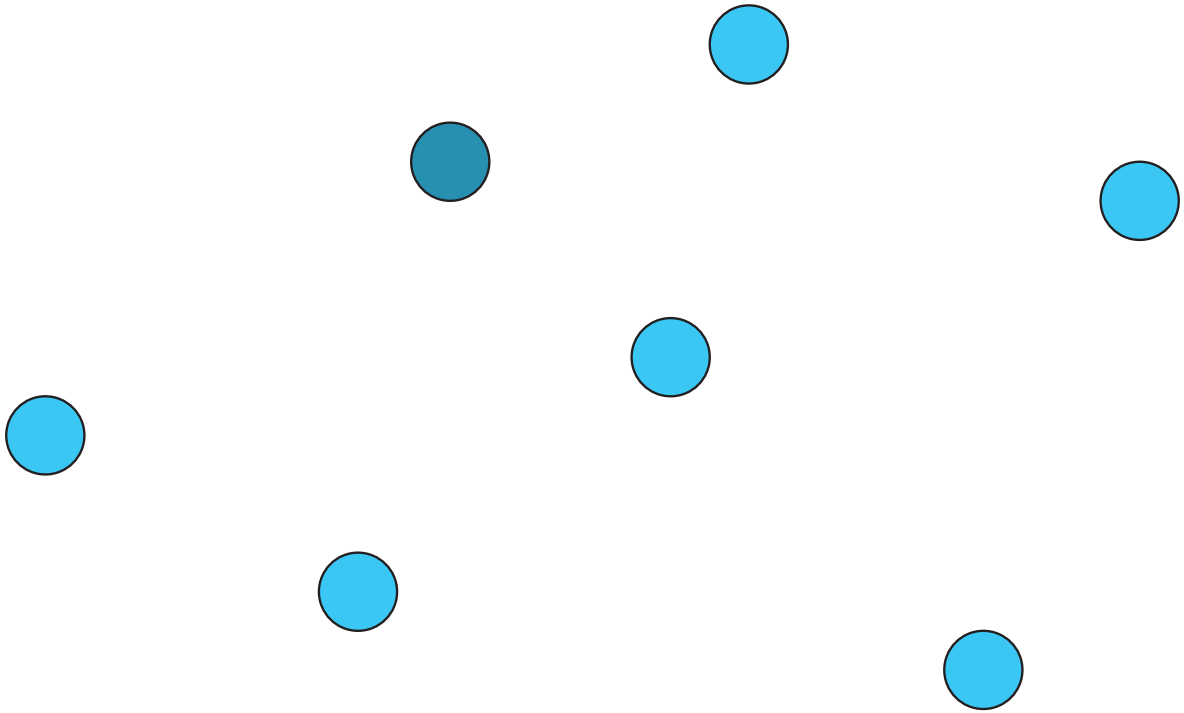
Bonus: it also solves Mutual Visibility and Convex Formation, provided that the faulty robot is external

Related model: sequential scheduler



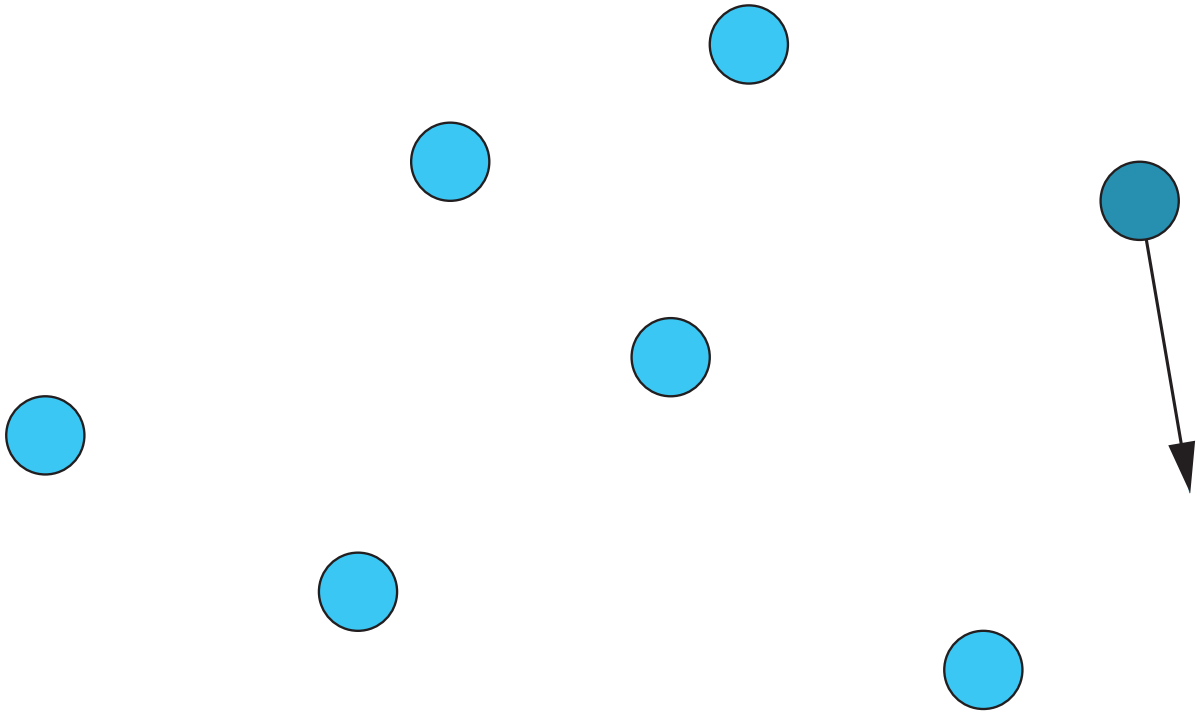
Model: at each turn, only one robot is activated

Related model: sequential scheduler



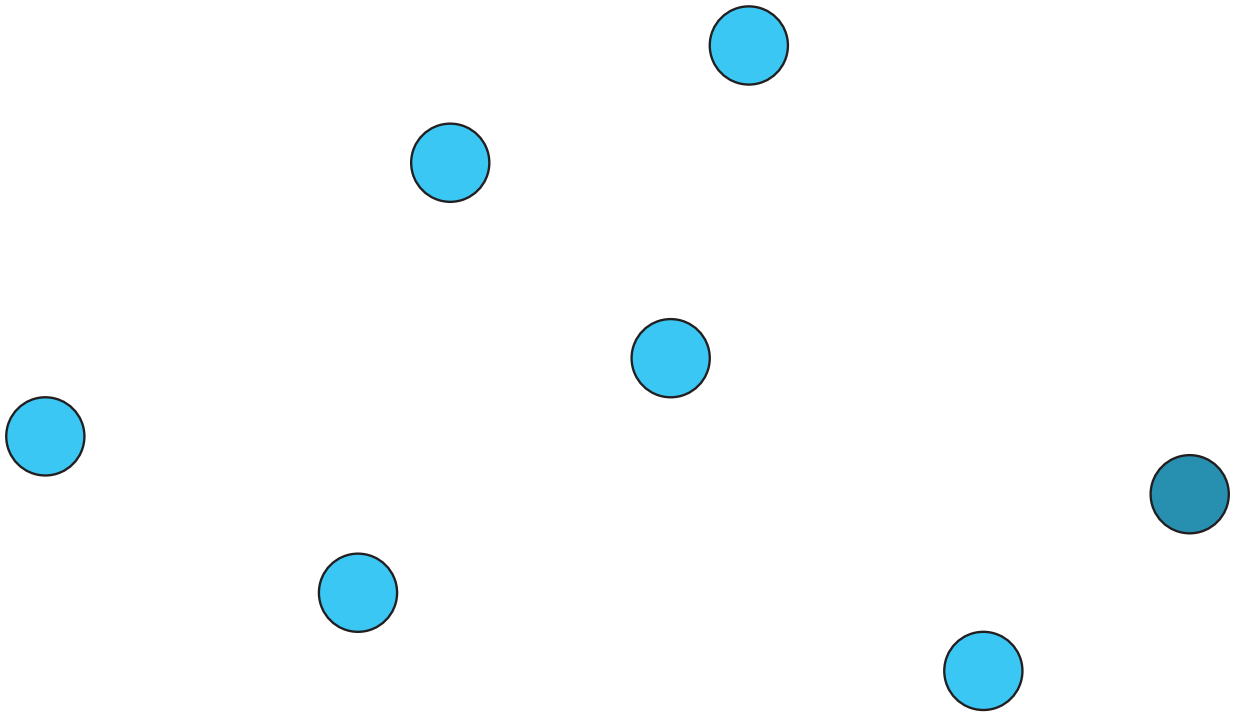
Model: at each turn, only one robot is activated

Related model: sequential scheduler



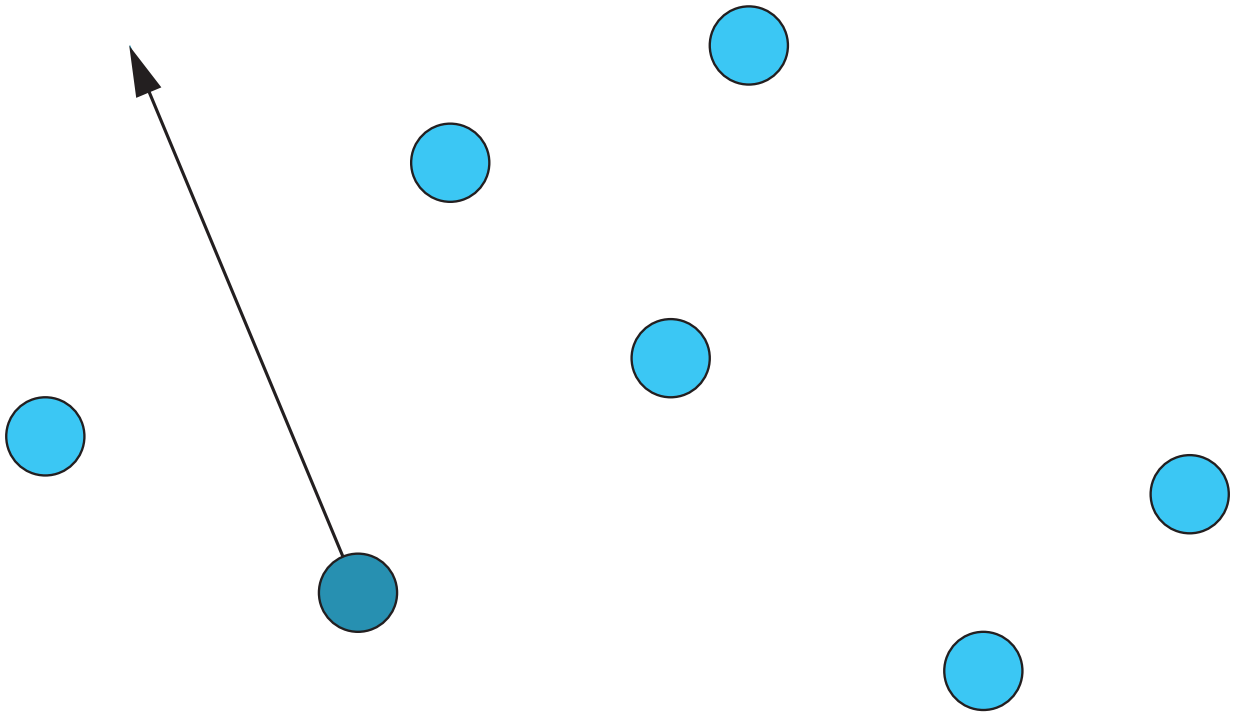
Model: at each turn, only one robot is activated

Related model: sequential scheduler



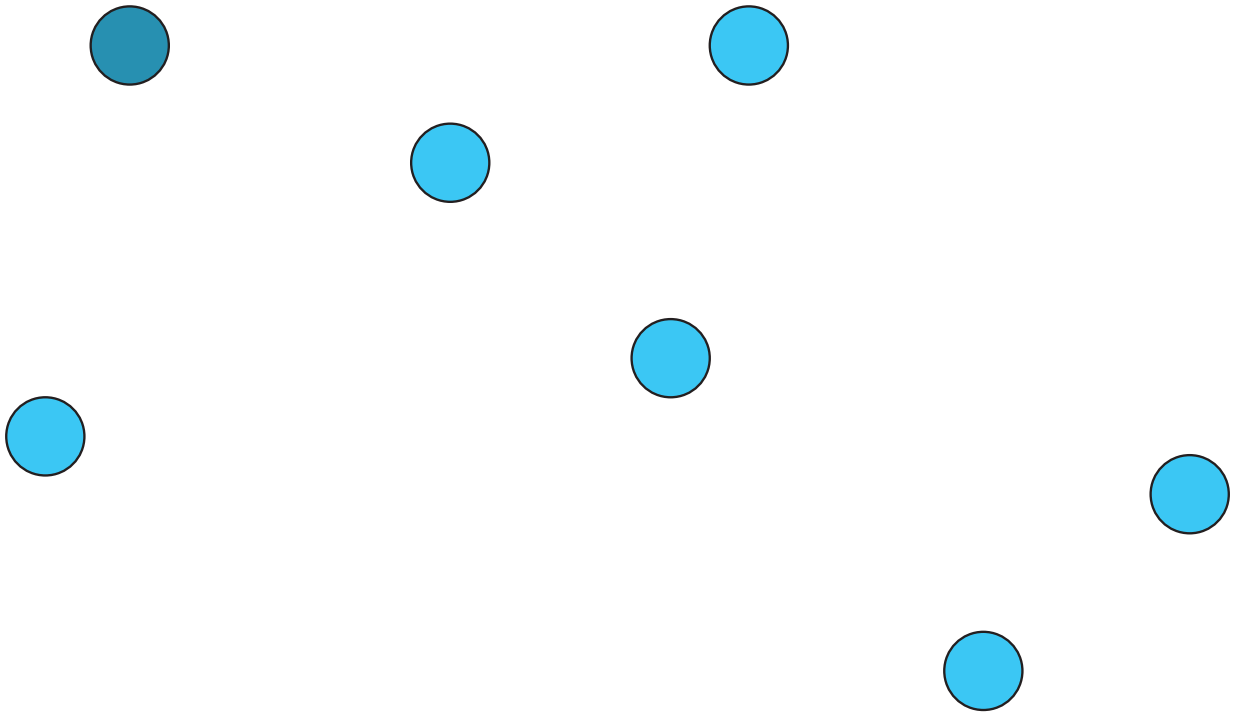
Model: at each turn, only one robot is activated

Related model: sequential scheduler



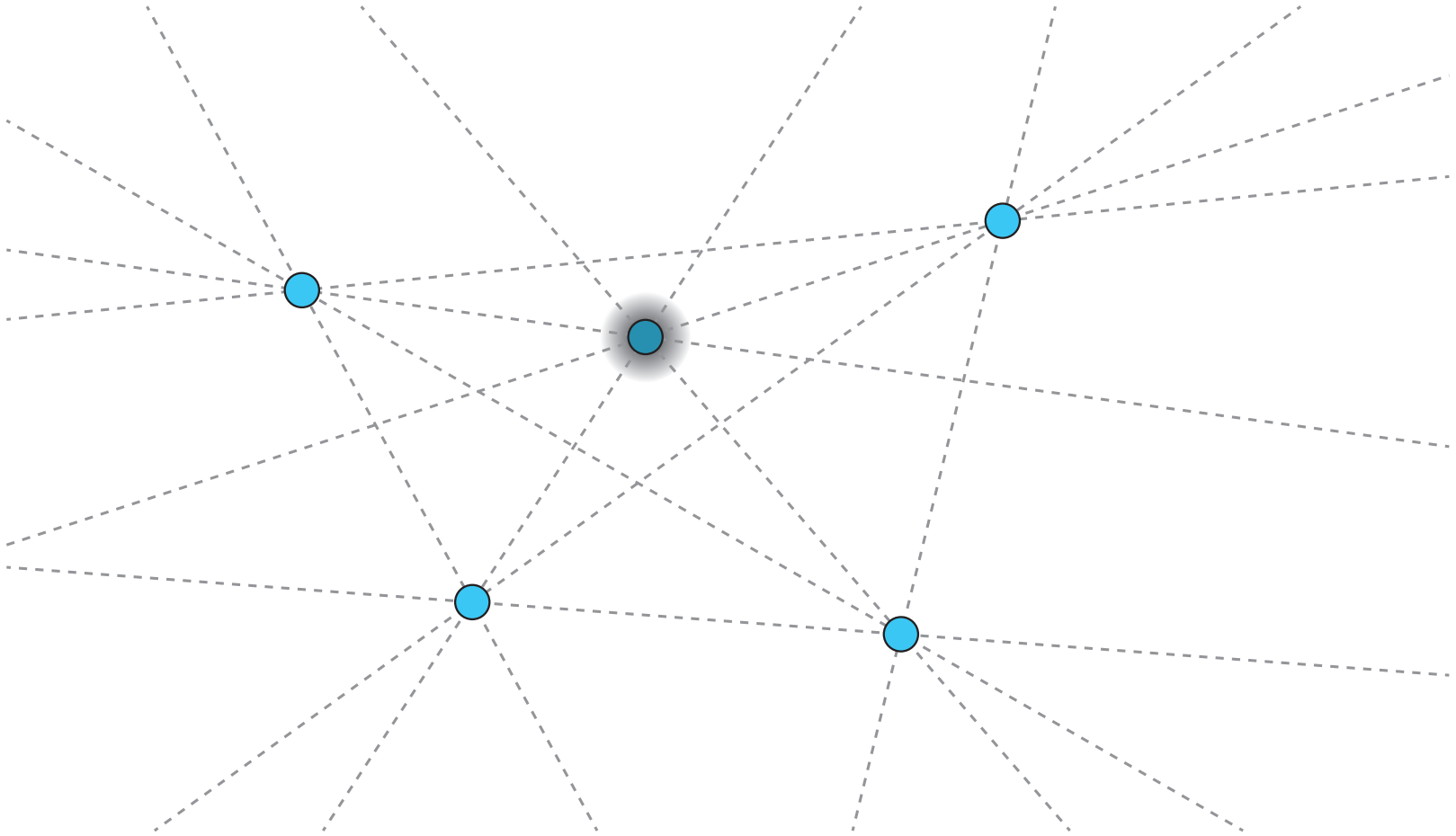
Model: at each turn, only one robot is activated

Related model: sequential scheduler



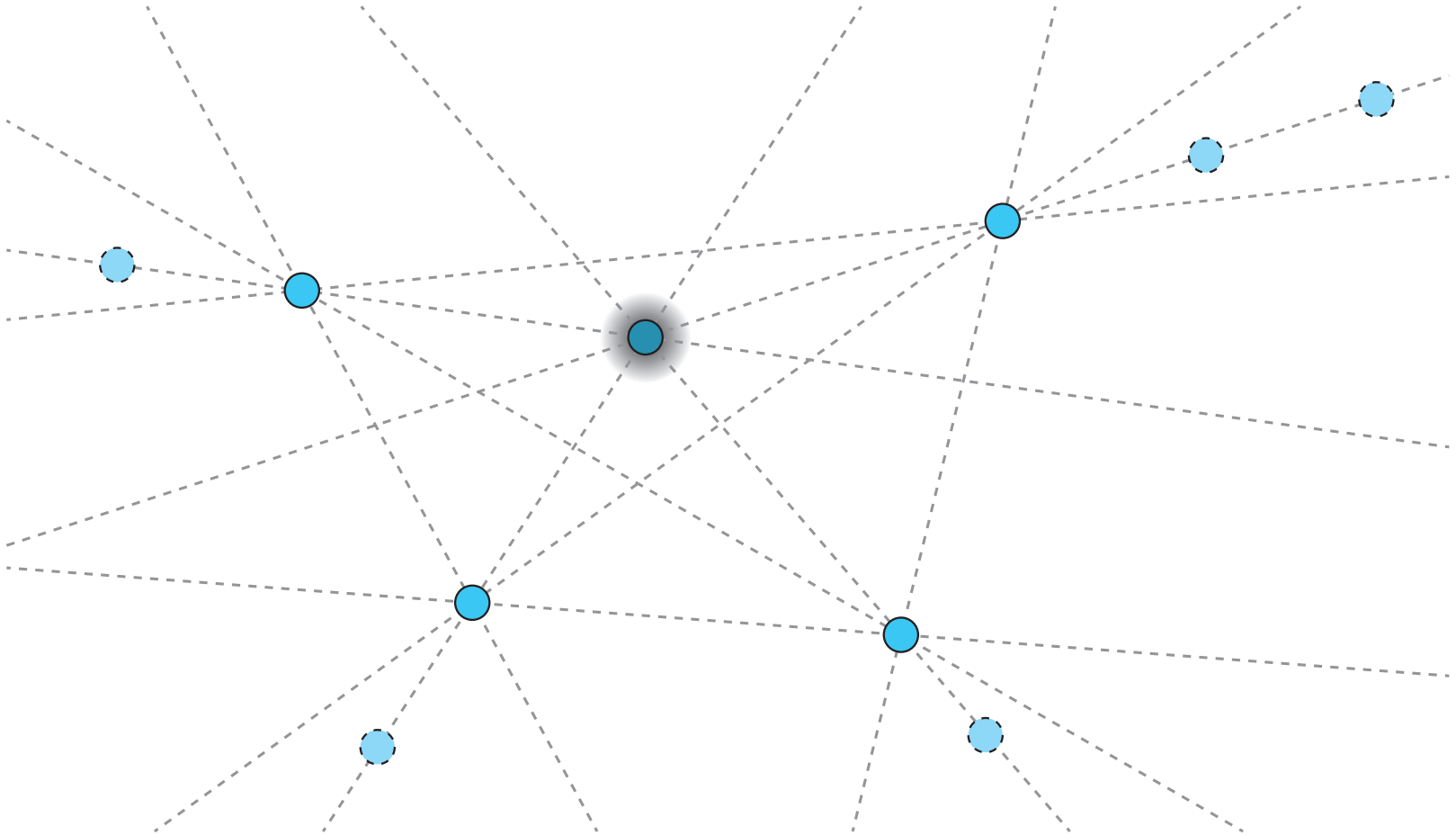
Model: at each turn, only one robot is activated

Related model: sequential scheduler



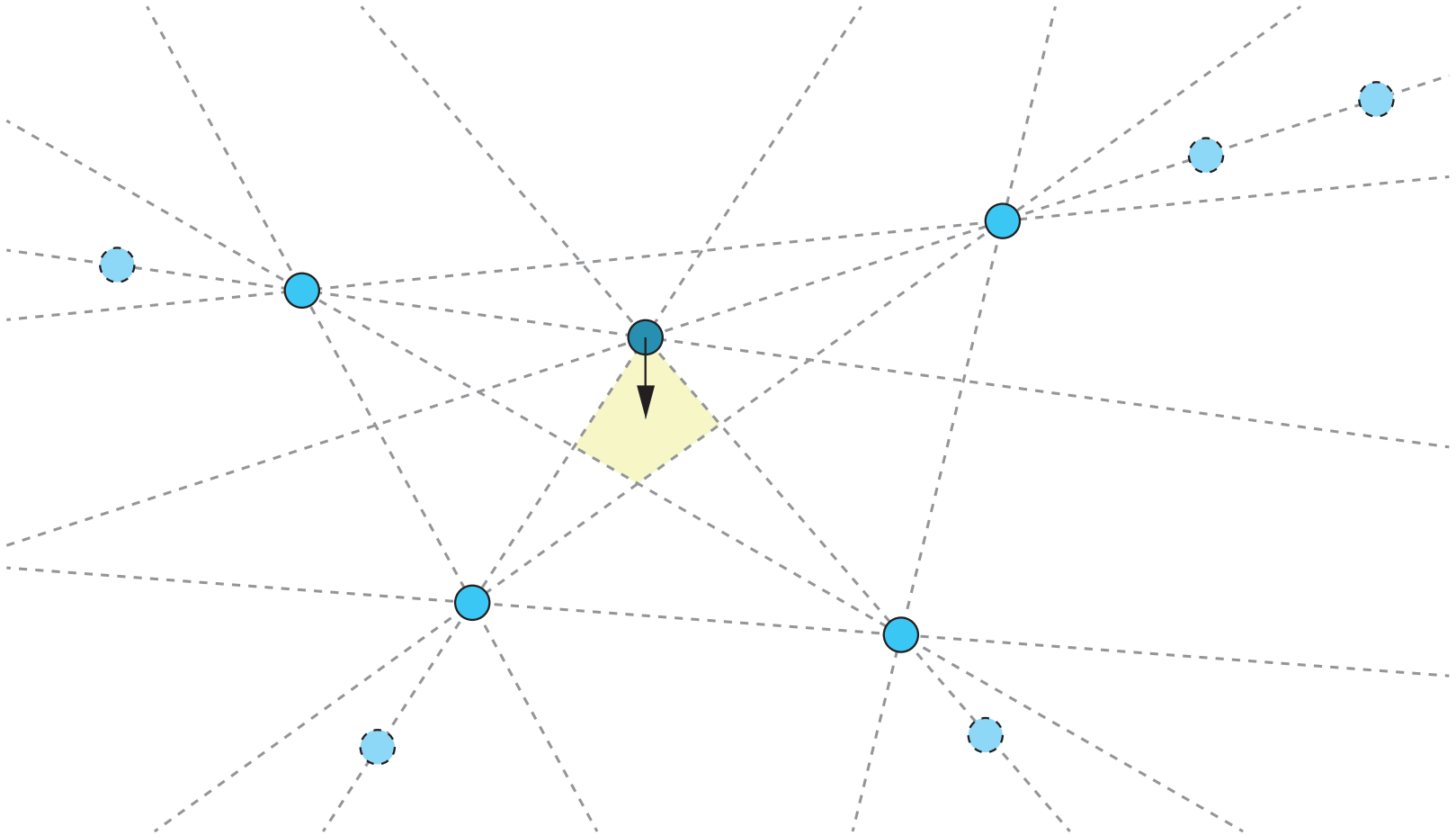
Solution: to solve MUTUAL VISIBILITY, move slightly to avoid lines through pairs of visible robots, then terminate

Related model: sequential scheduler



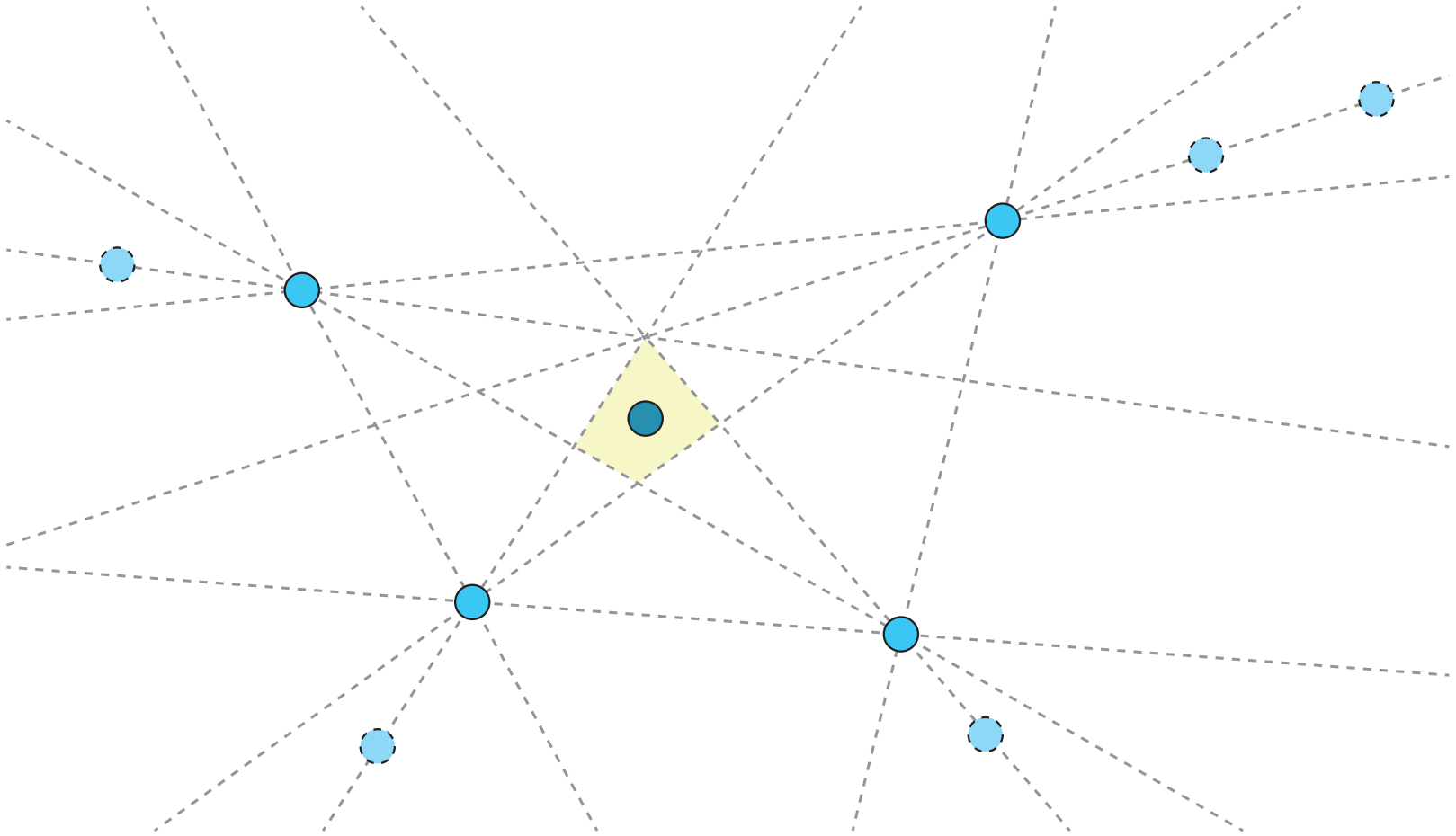
Solution: to solve MUTUAL VISIBILITY, move slightly to avoid lines through pairs of visible robots, then terminate

Related model: sequential scheduler



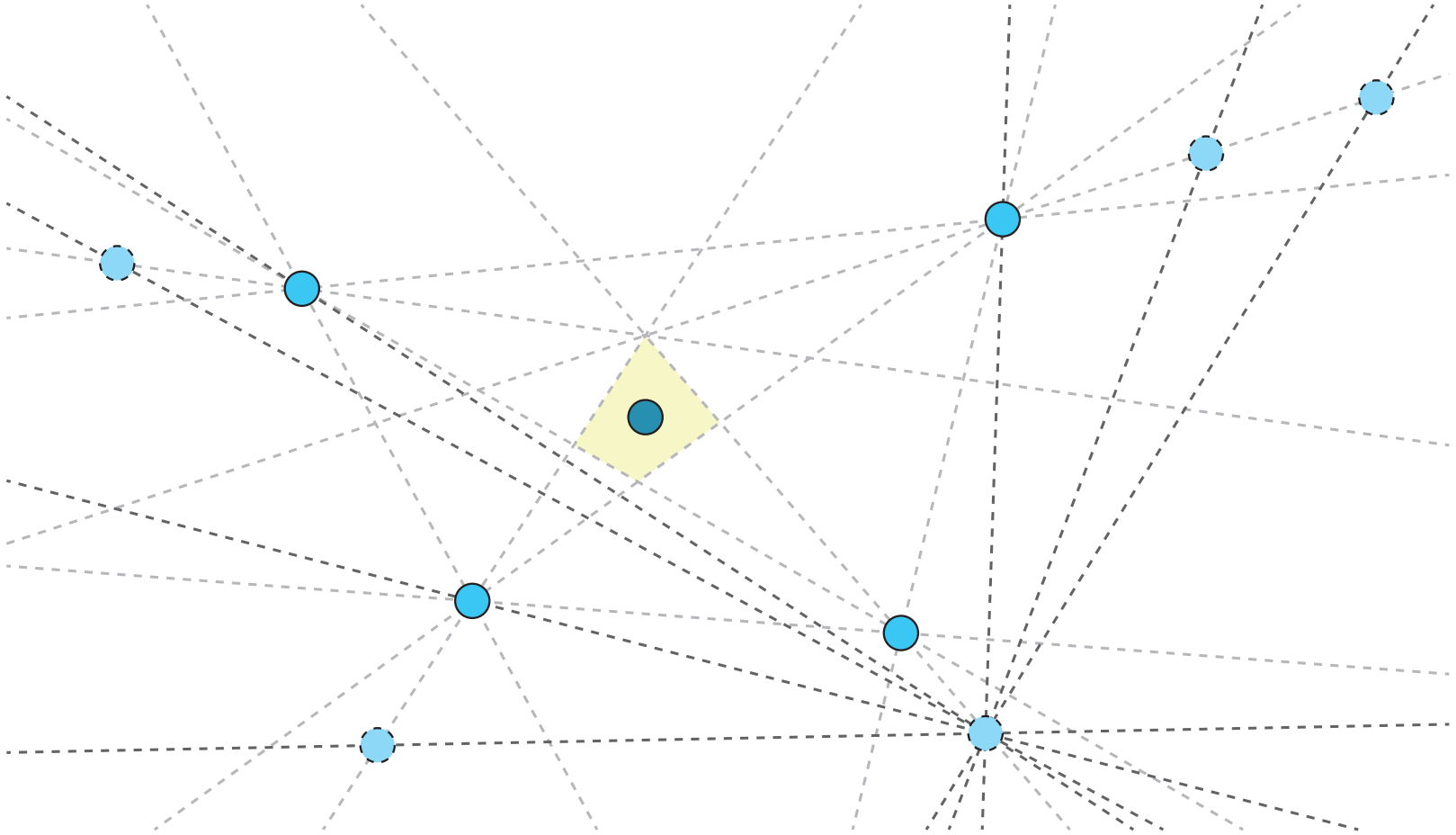
Solution: to solve MUTUAL VISIBILITY, move slightly to avoid lines through pairs of visible robots, then terminate

Related model: sequential scheduler



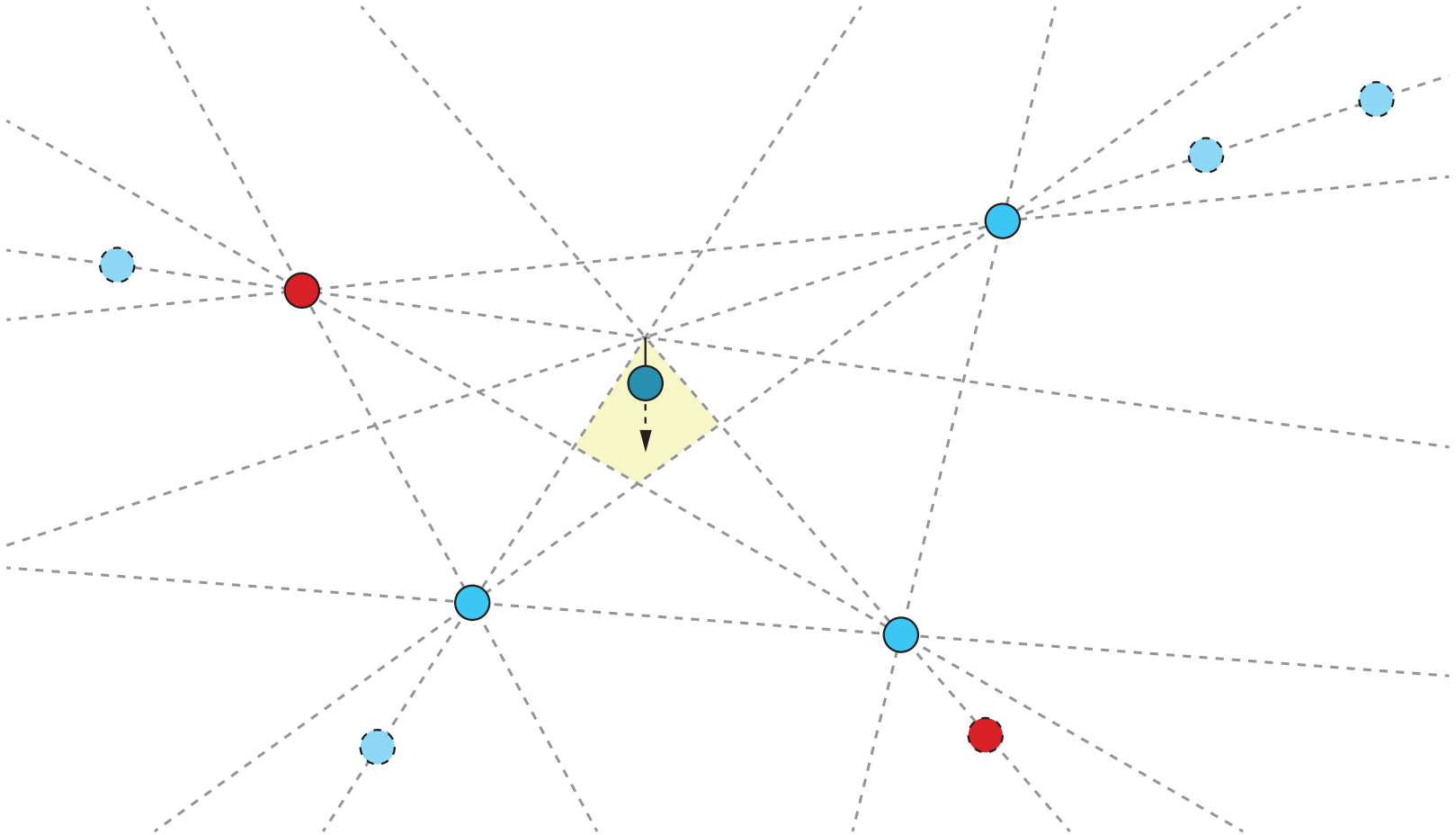
Solution: to solve MUTUAL VISIBILITY, move slightly to avoid lines through pairs of visible robots, then terminate

Related model: sequential scheduler



Solution: to solve MUTUAL VISIBILITY, move slightly to avoid lines through pairs of visible robots, then terminate

Related model: sequential scheduler



Bonus: works even with no knowledge of n, with unreliable moves, and with two faulty robots

WAKE UP!!!

You just missed an exciting talk! Let me summarize it:

- Oblivious mobile robots can solve Mutual Visibility
  - The algorithm uses the Communicating Vessels theorem
- As a by-product, we can solve some related problems:
  - Convex Formation
  - Near-Gathering, even with one faulty robot
  - 3-Sat, even in polynomial time (for details, read the paper...)
  - All of the above, also with unreliable moves