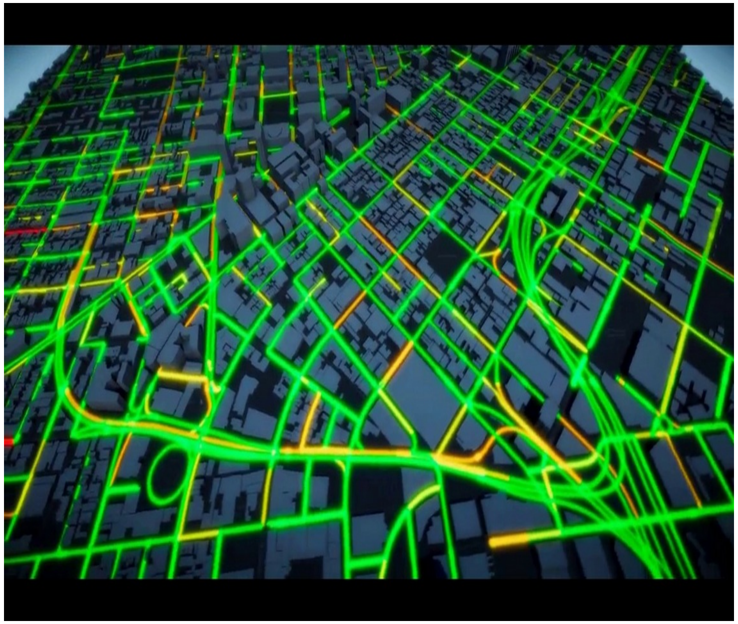
Asynchronous filling by myopic luminous robots

Attila Hideg & Tamás Lukovszki
ALGOSENSORS 2020

FILLING, UNIFORM DISPERSION

- Area
  - unknown
  - connected
  - represented by a graph of *n* vertices
  - for each vertex, adjacent vertices are arranged in a fixed cyclic order
- *n* Robots
  - enter at “Door” vertices
  - can move on edges to neighboring vertices
  - collision must be prevented
- Goal: Robots must occupy all vertices

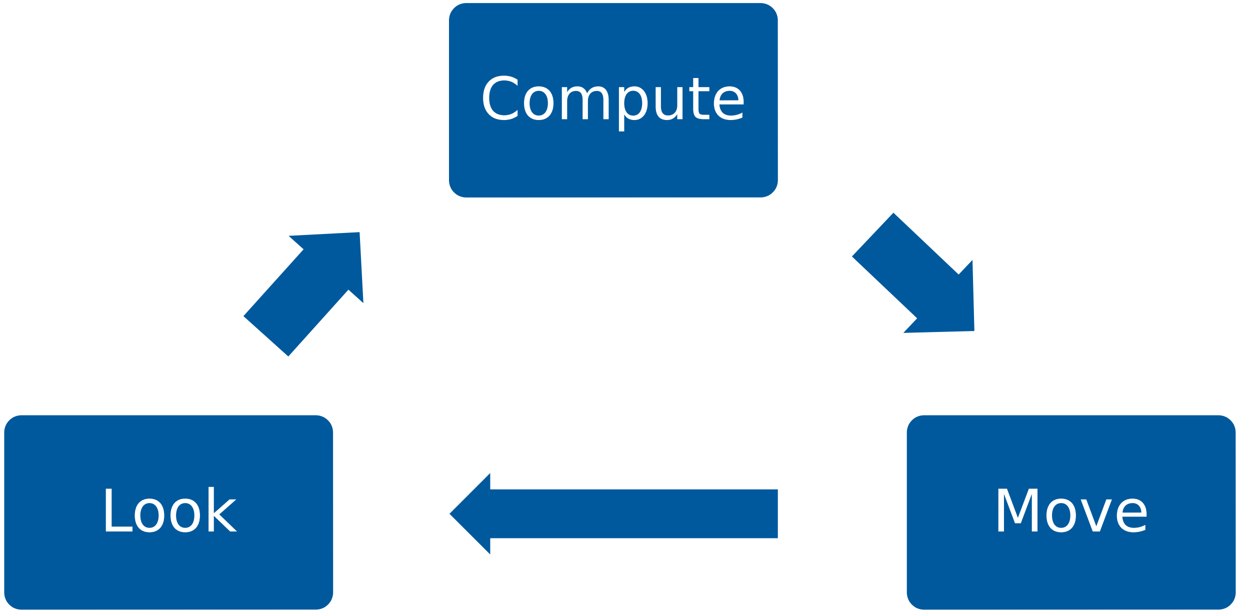


ROBOTS

- Robots have restricted capabilities
  - homogeneus
  - anonymous
  - limited viewig range
  - limited memory
  - no explicit communication
  - visible lights
  - asynchronous

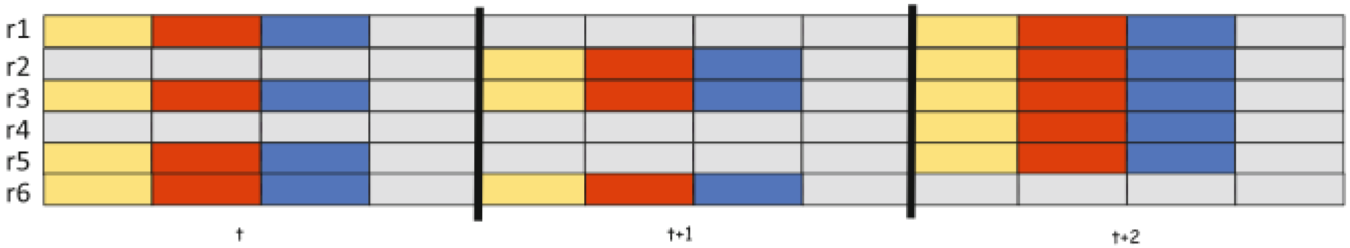
https://ssr.seas.harvard.edu/

LOOK-COMPUTE-MOVE (LCM) CYCLES

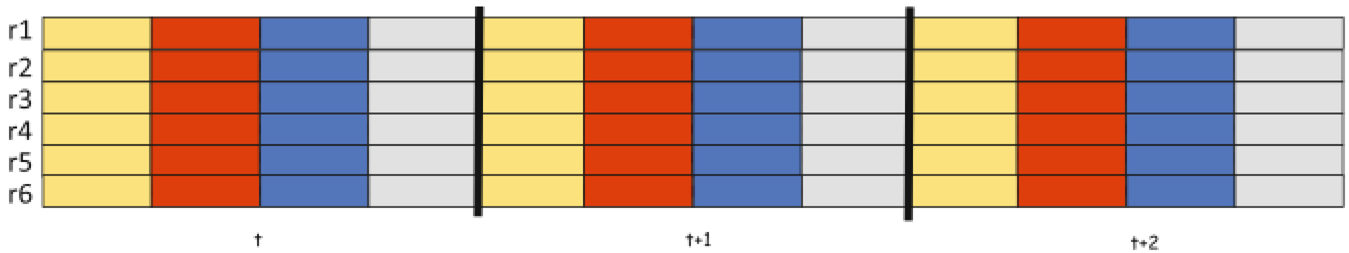


LCM CYCLES

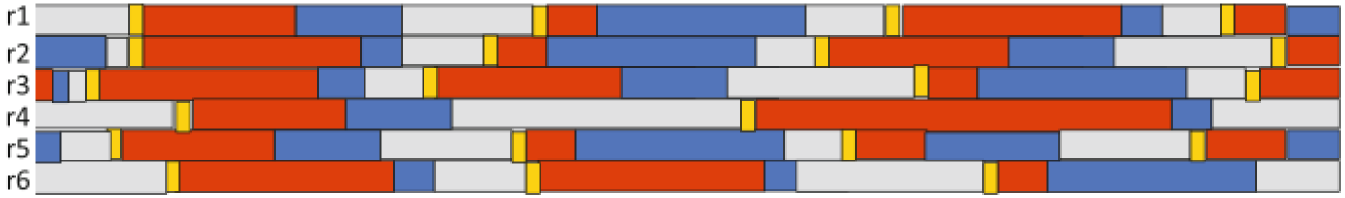
Semi synchronous (SSYNC)



Fully synchronous (FSYNC)

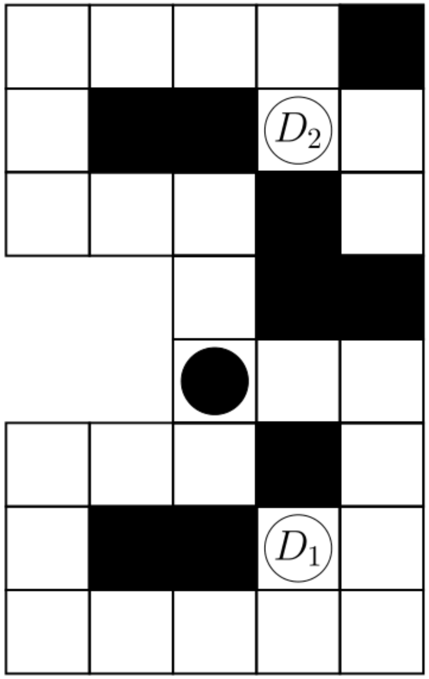


Aynchronous (ASYNC)



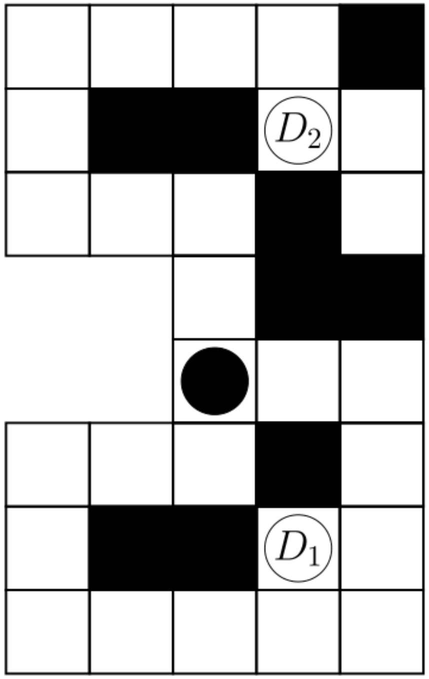
FILLING, DISPERSING

- Orthogonal

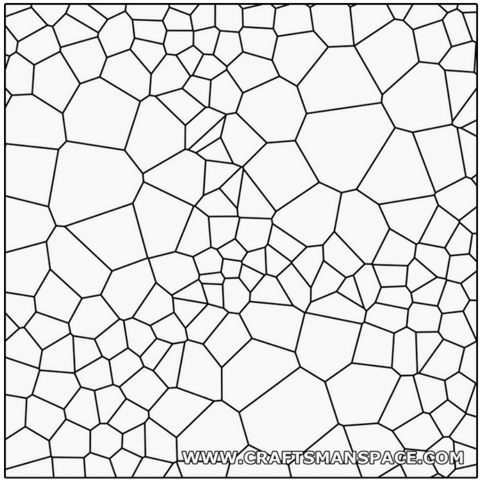


FILLING, DISPERSING

Orthogonal

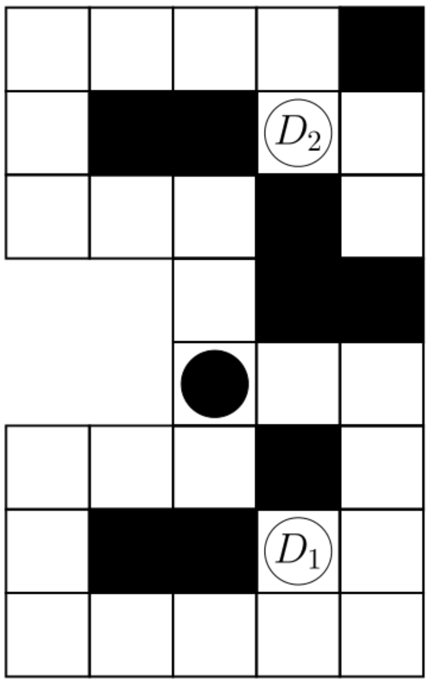


Arbitrary Graph

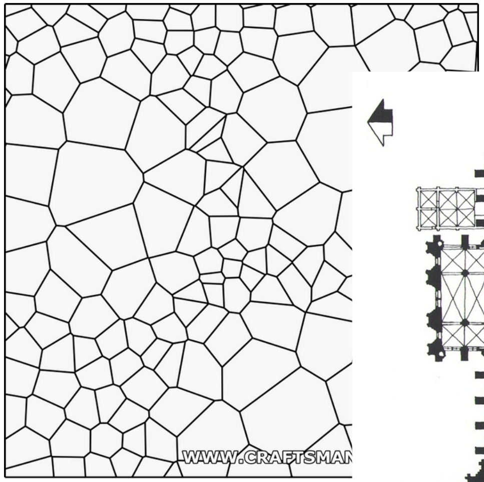


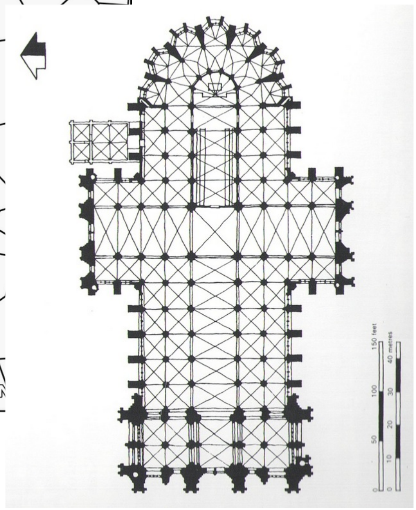
FILLING, DISPERSING

Orthogonal



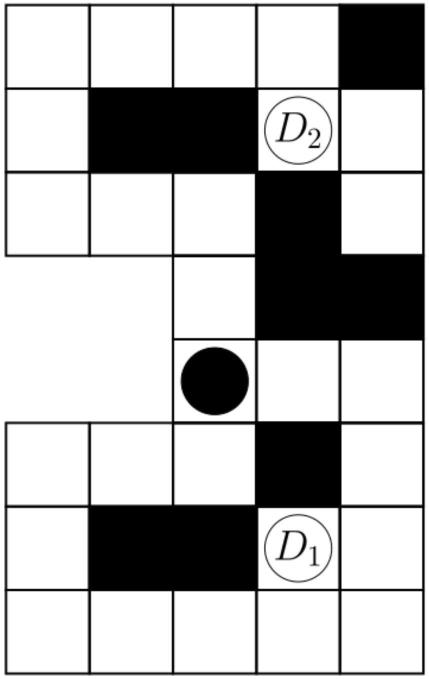
Arbitrary Graph



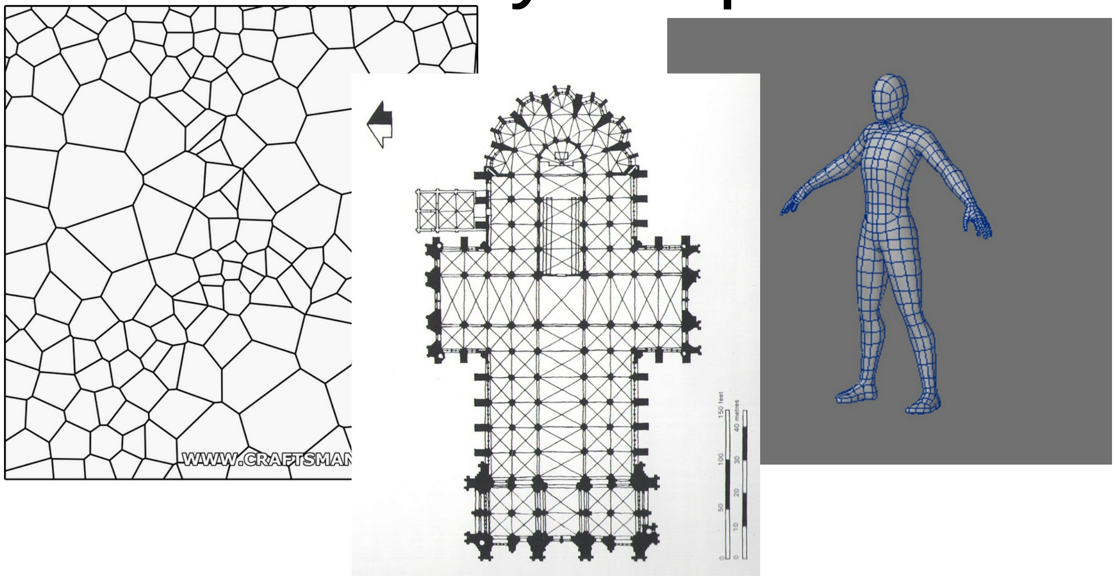


FILLING, DISPERSING

Orthogonal

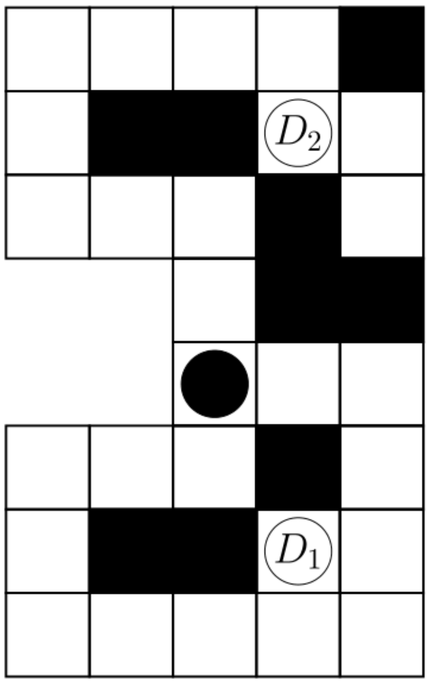


Arbitrary Graph

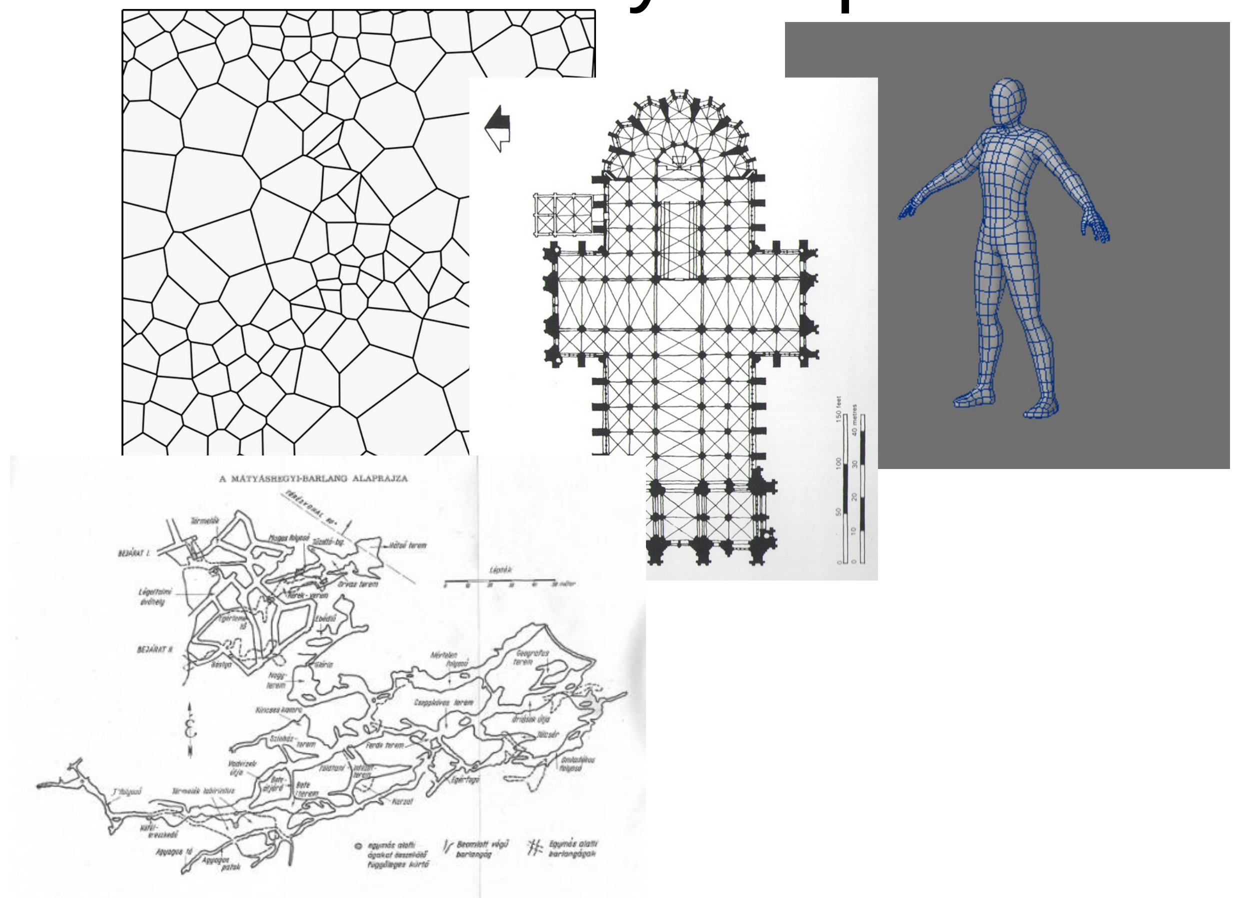


FILLING, DISPERSING

Orthogonal



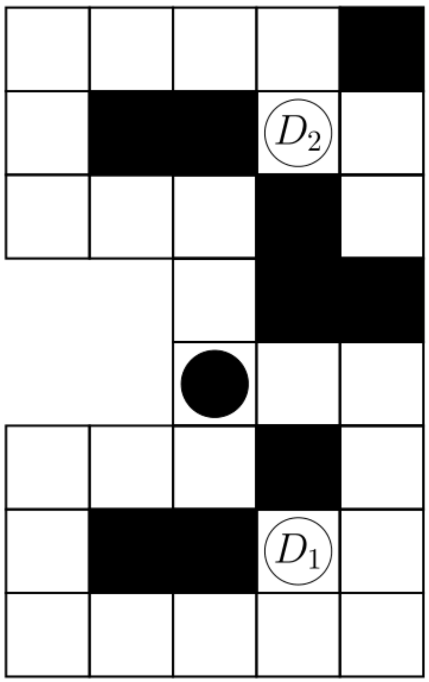
Arbitrary Graph



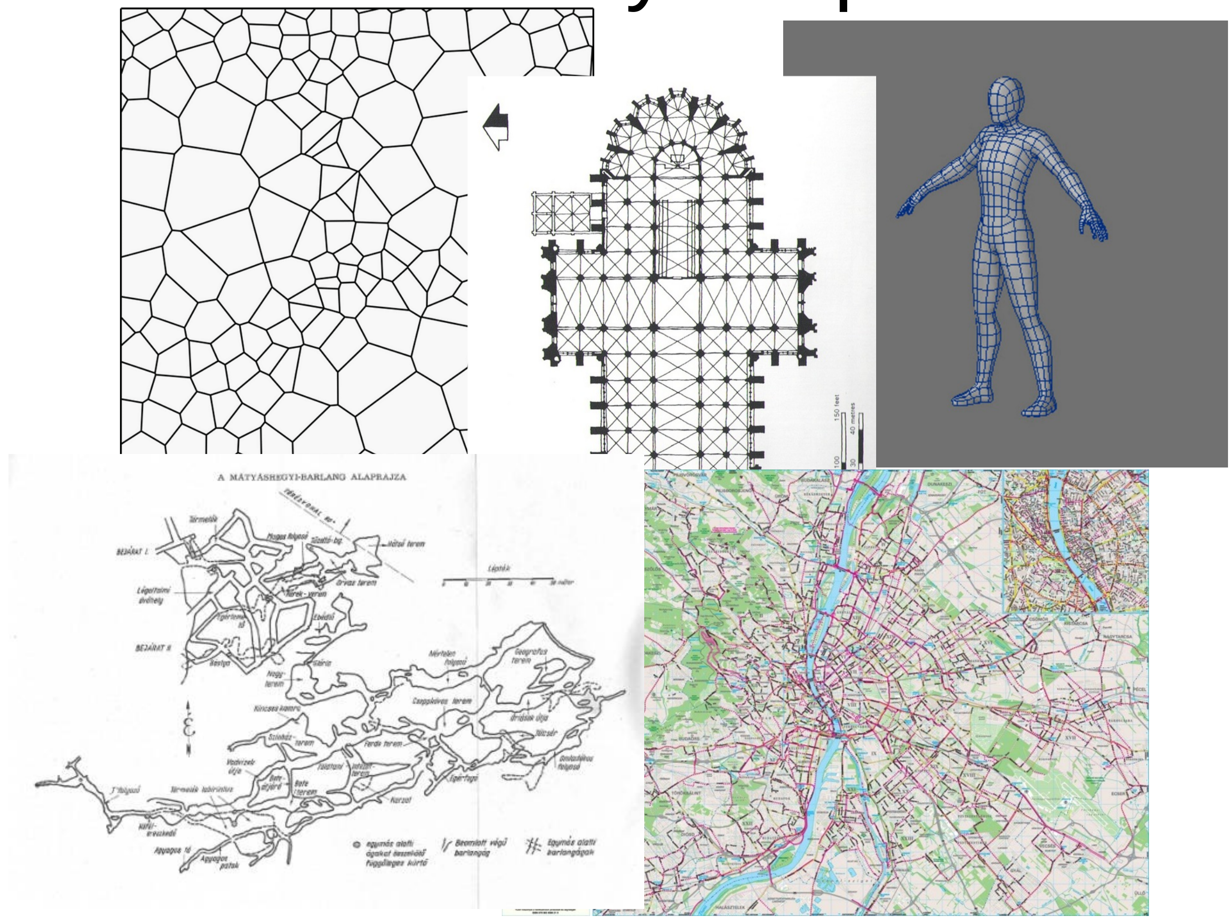
Attila Hideg & Tamás Lukovszki

FILLING, DISPERSING

Orthogonal



Arbitrary Graph

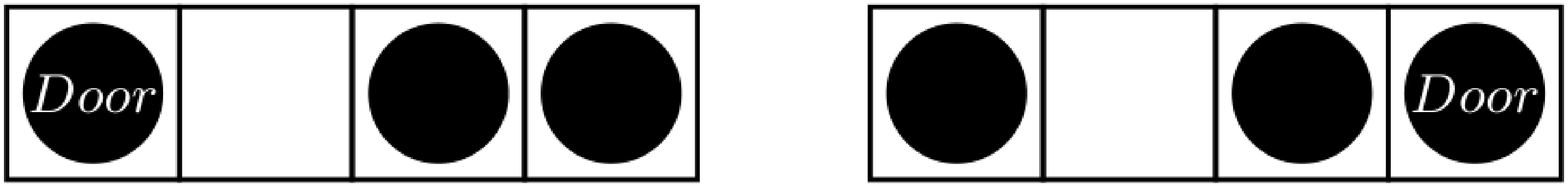


COLLISIONLESS DISPERSION

- Fundamental question:
  how “weak” the robots can be
  and still be able to solve the problem.

LOWER BOUNDS – COLLISIONLESS DISPERSION

- Visibility range: ≥1 hop
- Time: $\Omega(n)$ LCM-rounds
- Memory: $\Omega(1)$ bits [Barrameda et al. 2008]

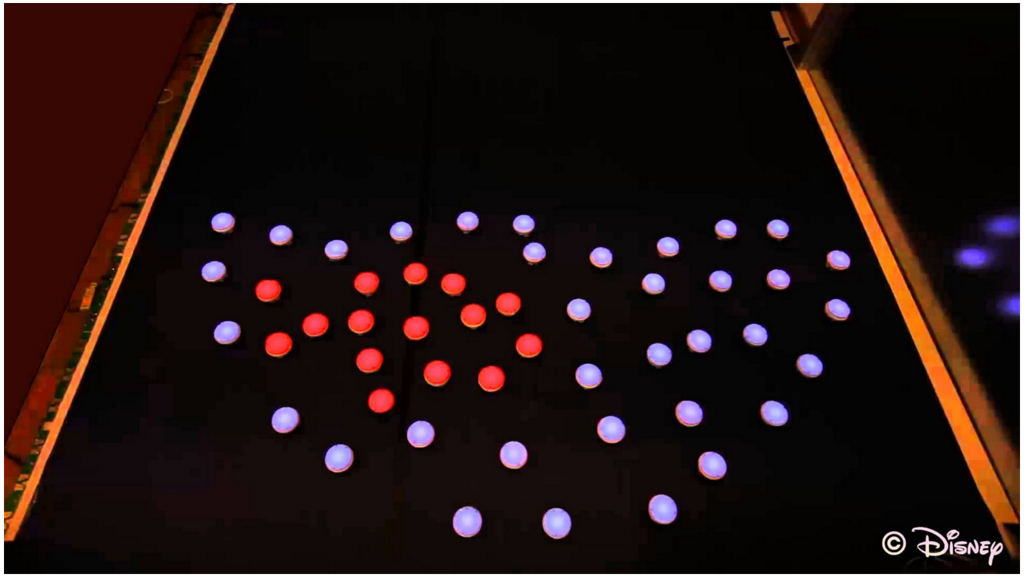


STATE OF THE ART – COLLISIONLESS DISPERSION

Method	FSYNC/ ASYNC	Doors	Viewing range	Comm. range	Memory bits	Area (Orthogonal/ Arbitrary)
BFLF, DFLF [Hsiang et al. 2004]	FSYNC	Single	2	2	2	O
TALK [Barrameda et al. 2013]	ASYNC	Single	2	2	4	O
MUTE [Barrameda et al. 2013]	ASYNC	Single	6	-	9	O
MULTIPLE [Barrameda et al. 2008]	ASYNC	Multiple	3	- *k* colors	4	O
Single Door [Hideg, Lukovszki 2017]	FSYNC	Single	1	-	13	O
Multiple Door [Hideg, Lukovszki 2017]	FSYNC	Multiple	1	-	13	O
VCM [Hideg, Lukovszki 2018]	FSYNC	Single	1	-	$O(\Delta)$	A
MD-VCM [Hideg, Lukovszki 2018]	FSYNC	Multiple	1	-	$O(\Delta \cdot \log k)$	A

LUMINOUS ROBOTS

- Robots are enhanced with VISIBLE LIGHTS
- Can change color
- Model [Peleg 2005]



Pixelbots, Disney & ETH Zürich

- FSYNC $\not\leq$ ASYNC$^{O(1)}$ and ASYNC$^{O(1)}$ $\not\leq$ FSYNC [D'Emidio et al. 2016]

OUR CONTRIBUTION

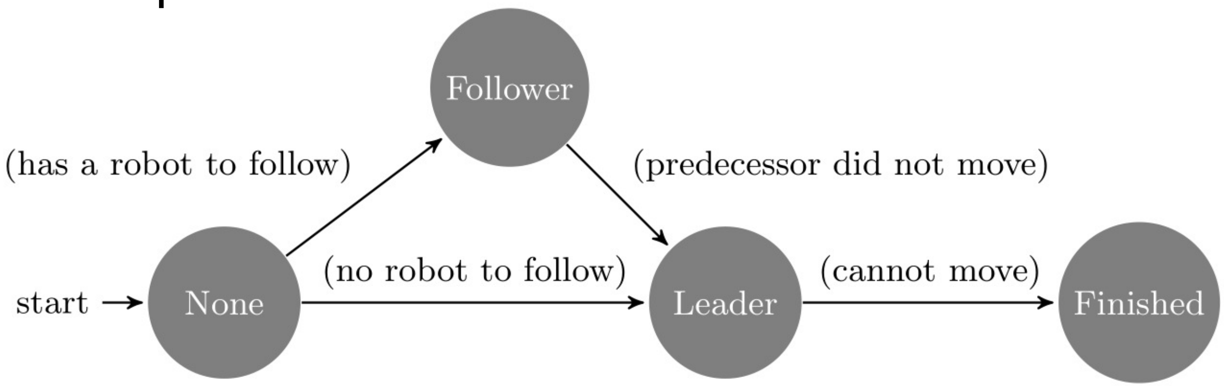
Method	FSYNC/ ASYNC	Doors	Viewing range	Runtime #async rounds	Persistent memory bits	Colors	Area (Orthogonal/ Arbitrary)
PACK	ASYNC	Single	1	$O(n^2)$	$O(\log \Delta)$	$\Delta+4$	A
Mod-PACK	ASYNC	Single	1	$O(n^2 \log \Delta)$	$O(\log \Delta)$	$O(1)$	A
BLOCK	ASYNC	Single	2	$O(n)$	$O(\log \Delta)$	$\Delta+4$	A
k-Door-BLOCK	ASYNC	Multiple	2	$O(n)$	$O(\log(\Delta+k))$	$\Delta+k+4$	A

First asymptotic bounds for filling in the ASYNC model.

Only termination in finite time has been proven previously.

VISIBILITY: 1-HOP – PACK ALGORITHM

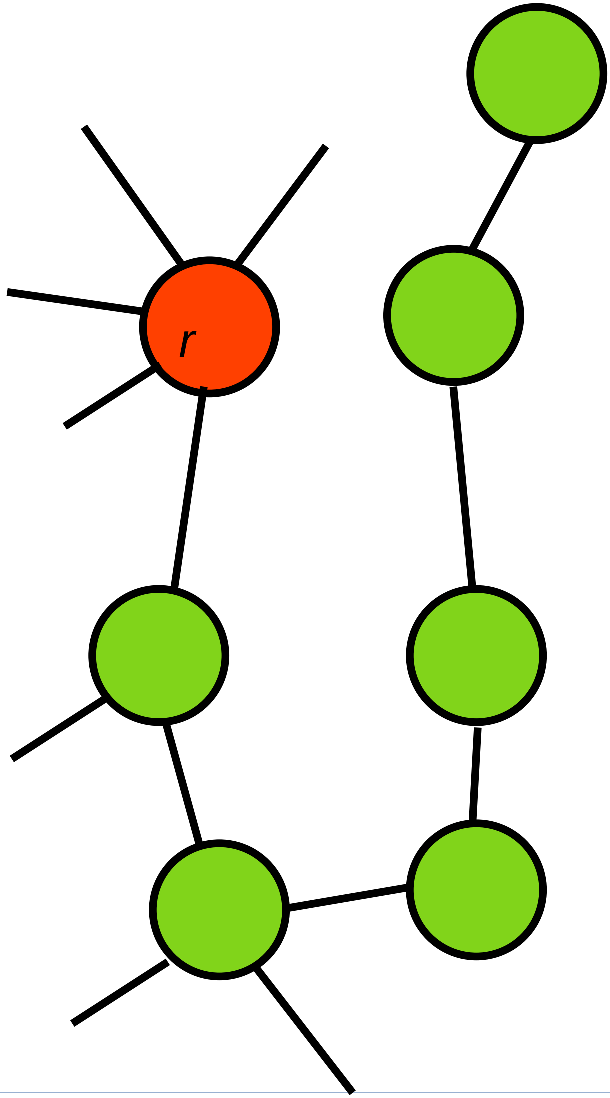
- Follow-the-Leader basic concept:



- Mimics a DFS traversal of the unknown graph
- Virtual Chain: Path of the current Leader from the Door
  - All not “Finished” robots are on the virtual chain
- Tasks to solve:
  - Prevent collision

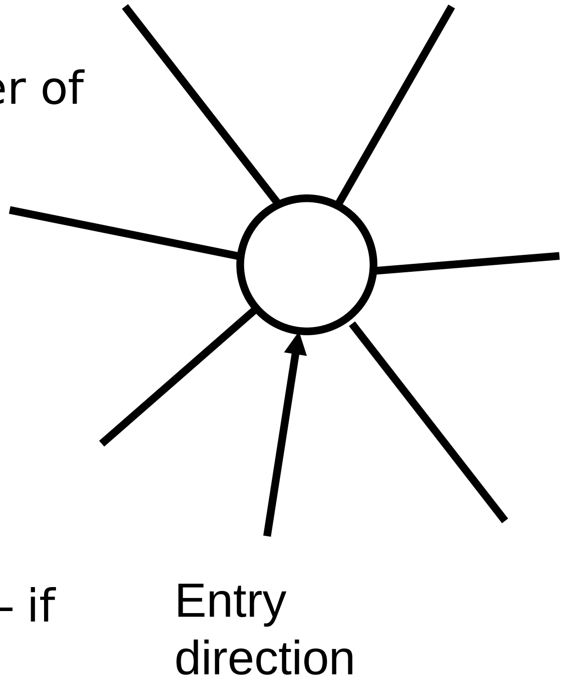
VISIBILITY: 1-HOP – PACK ALGORITHM

- Leader moves to unvisited vertices
- Packed state:
  Each Follower is immediately behind its predecessor
  - All vertices of the chain are occupied by a robot
- Leader only moves, when packed state is reached
  - No other robot can move in this state
  - Collision-freeness guaranteed



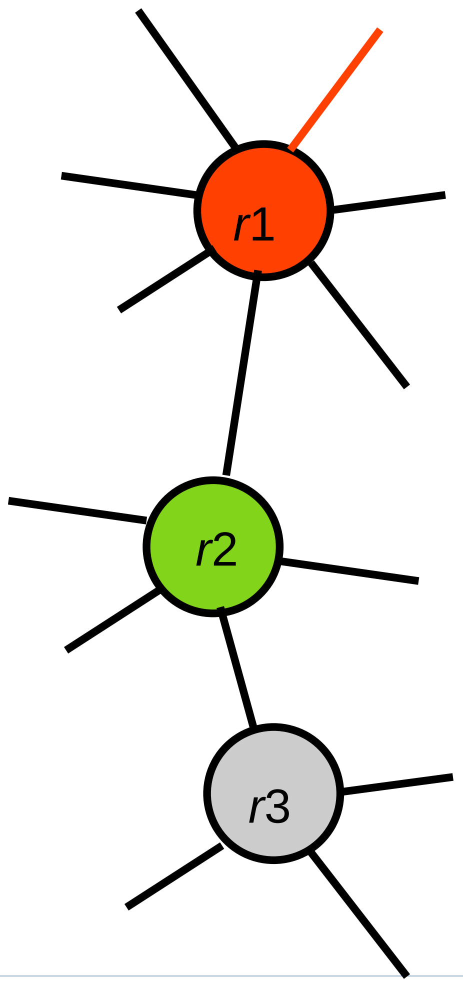
VISIBILITY: 1-HOP – PACK ALGORITHM

- Δ+4 colors:
  - Δ colors (DIR) indicating the direction of the target vertex (relative to the entry direction in cyclic order of neighbors)
  - 2 colors (CONF, CONF2) for confirmation
    - Robot can only move if
      the successor is behind it and
      the DIR color is confirmed
  - 1 color (MOV) during movement
  - Light is off (considered as color)
- Leader moves to an unoccupied neighboring vertex – if exists.
- If there is no unoccupied neighboring vertex, then
  - the Leader switches to Finished state and
  - the successor becomes the new Leader
    ”Taking the Leadership”



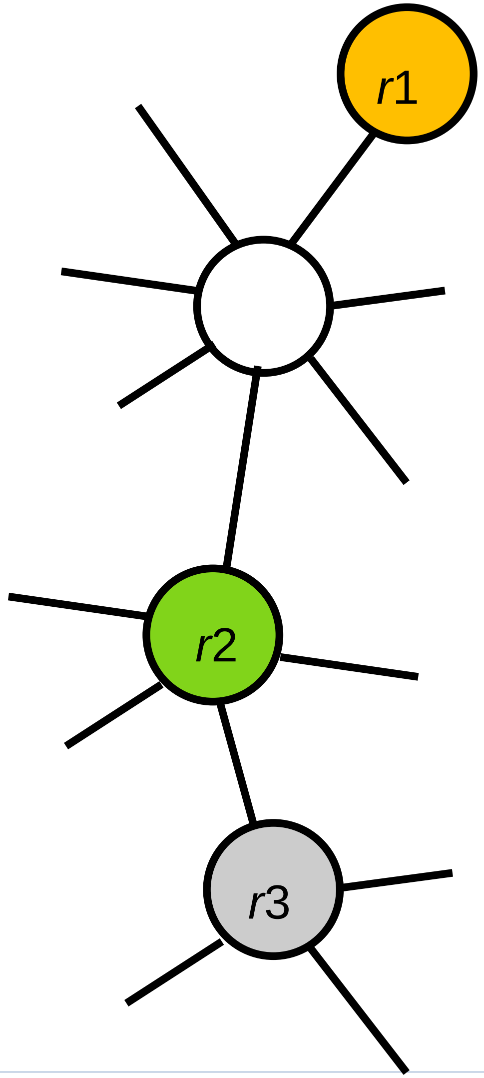
VISIBILITY: 1-HOP – PACK ALGORITHM

- Leader:
  - Can only move to an unvisited vertex. When it wants to move, it
  - shows the direction: setting the DIR color, and
  - it waits until its successor allows to move by setting its CONF color. During the movement, the Leader shows the MOV color.
  - When its successor sets CONF color, the chain is in Packed state.



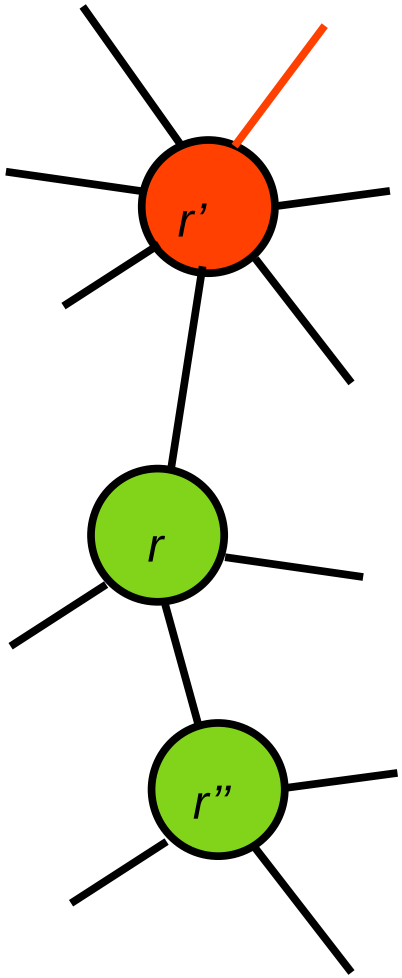
VISIBILITY: 1-HOP – PACK ALGORITHM

- Leader:
  - Can only move to an unvisited vertex. When it wants to move, it
  - shows the direction: setting the DIR color, and
  - it waits until its successor allows to move by setting its CONF color. During the movement, the Leader shows the MOV color.
  - When its successor sets CONF color, the chain is in Packed state.



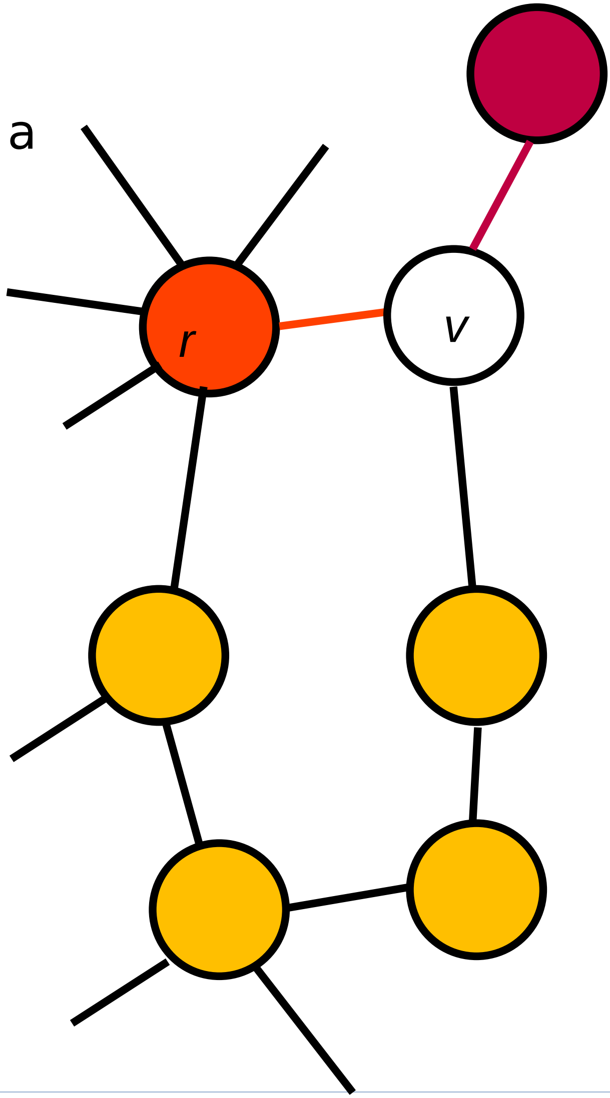
VISIBILITY: 1-HOP – PACK ALGORITHM

- Follower: Follows its predecessor.
- Follower *r* sets the CONF color if and only if
  - 1) predecessor of *r* is showing its direction, and
  - 2) successor *r''* of *r* – if exists – has already set its CONF color (i.e. *r''* knows in which direction *r* will move).
  - This allows the predecessor *r'* of *r* to move to its destination knowing:
    - 1) all the robots behind *r'* have set CONF color, and
    - 2) the robots behind *r'* will not move until *r'* moved.
    - When *r'* is the Leader, the chain is in Packed state.



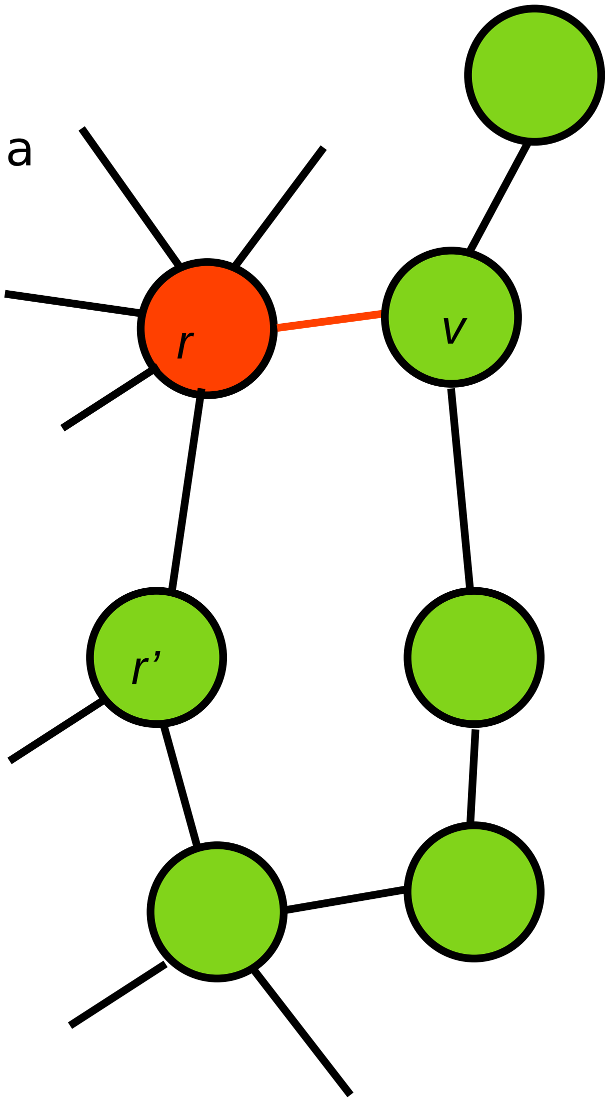
VISIBILITY: 1-HOP – PACK ALGORITHM

- Leader target change:
  - It might happen that the Leader *r* chooses a target *v*, which is unoccupied when *r* performs its Look operation.



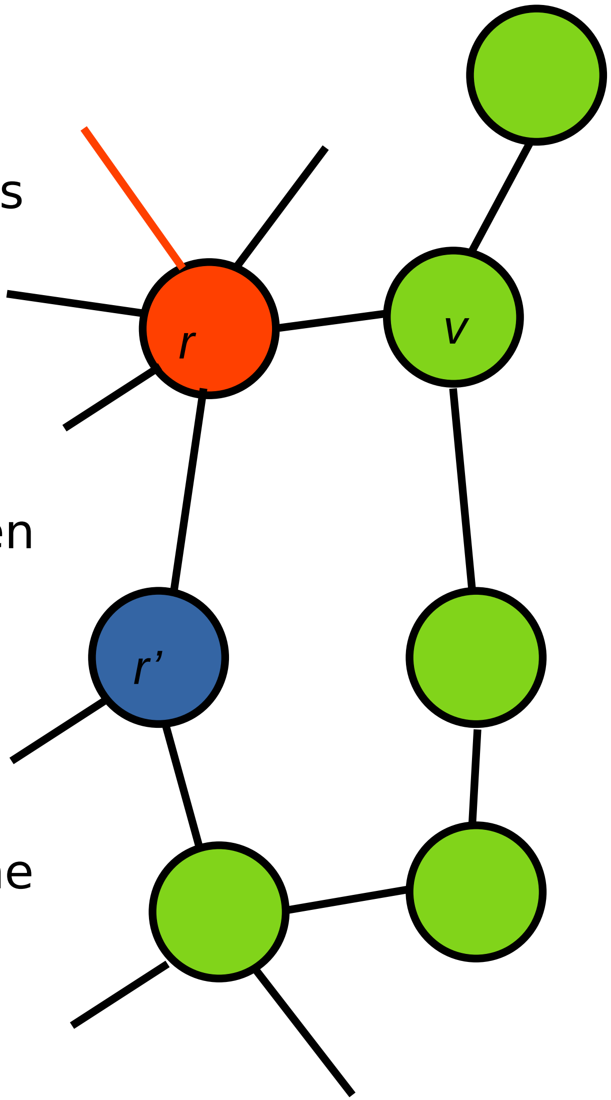
VISIBILITY: 1-HOP – PACK ALGORITHM

- Leader target change:
  - It might happen that the Leader *r* chooses a target *v*, which is unoccupied when *r* performs its Look operation.
  - When the successor *r'* of *r* sets the CONF color, another robot already moved to *v*.



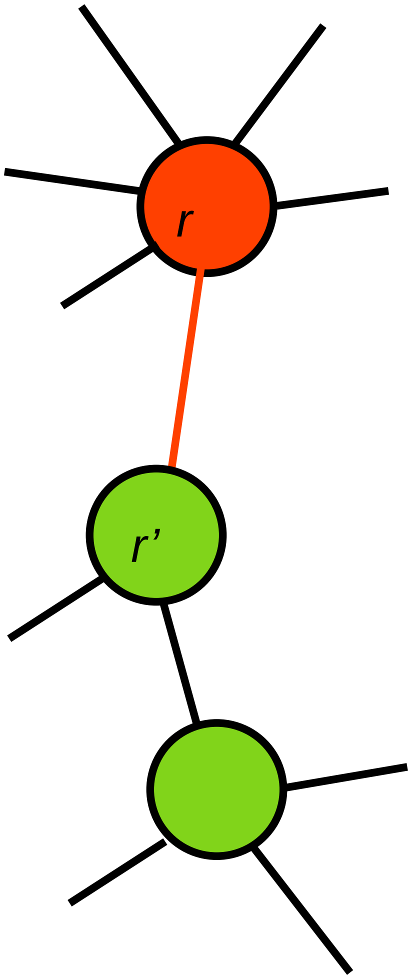
VISIBILITY: 1-HOP – PACK ALGORITHM

- Leader target change:
  - It might happen that the Leader *r* chooses a target *v*, which is unoccupied when *r* performs its Look operation.
  - When the successor *r'* of *r* sets the CONF color, another robot already moved to *v*.
  - If *r* has an unoccupied neighboring vertex then
    - *r* sets the new DIR color and waits until its successor sets the CONF2 color.
    - *r* moves to the target.
  - Otherwise, *r* switches to Finished state and the Leadership is taken by *r'*.



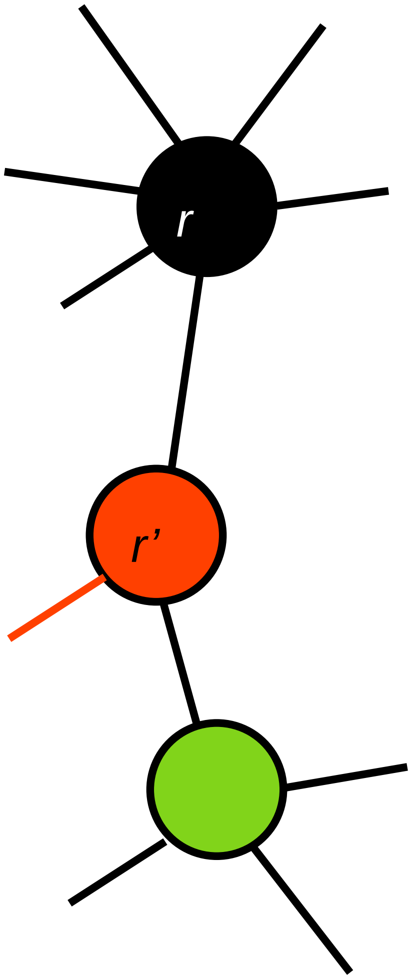
VISIBILITY: 1-HOP – PACK ALGORITHM

- Taking the Leadership:
  - When the Leader *r* cannot move anymore, its successor has to become the new Leader.
  - *r* sets its DIR color to Δ.
    - color Δ indicates that
      - the Leader cannot move anymore,
      - wants to switch to Finished state,
      - leadership must be taken by its successor.
  - successor *r'* of r sets its CONF color, waits for the previous Leader to turn off its light.
  - Then *r'* becomes the Leader.
    r' tries to move to an unvisited vertex.



VISIBILITY: 1-HOP – PACK ALGORITHM

- Taking the Leadership:
  - When the Leader *r* cannot move anymore, its successor has to become the new Leader.
  - *r* sets its DIR color to Δ.
    - color Δ indicates that
      - the Leader cannot move anymore,
      - wants to switch to Finished state,
      - leadership must be taken by its successor.
  - successor *r'* of r sets its CONF color, waits for the previous Leader to turn off its light.
  - Then *r'* becomes the Leader.
    r' tries to move to an unvisited vertex.



PACK – ANALYSIS

- **Lemma 1.** Leader only moves to unvisited vertices.
- **Lemma 2.** There can be at most one Leader at any time.
- **Lemma 3.** Robots can not collide.
- **Lemma 4.** PACK fills the area represented by a connected graph.

PACK – ANALYSIS

- **Theorem 1**. Algorithm PACK fills a connected graph in the ASYNC model by robots having a
  - visibility range of 1 hop,
  - O(log Δ) bits of persistent storage, and
  - Δ + 4 colors, including the color when the light is off.
  - PACK runs in O(n^2) asynchronous rounds.

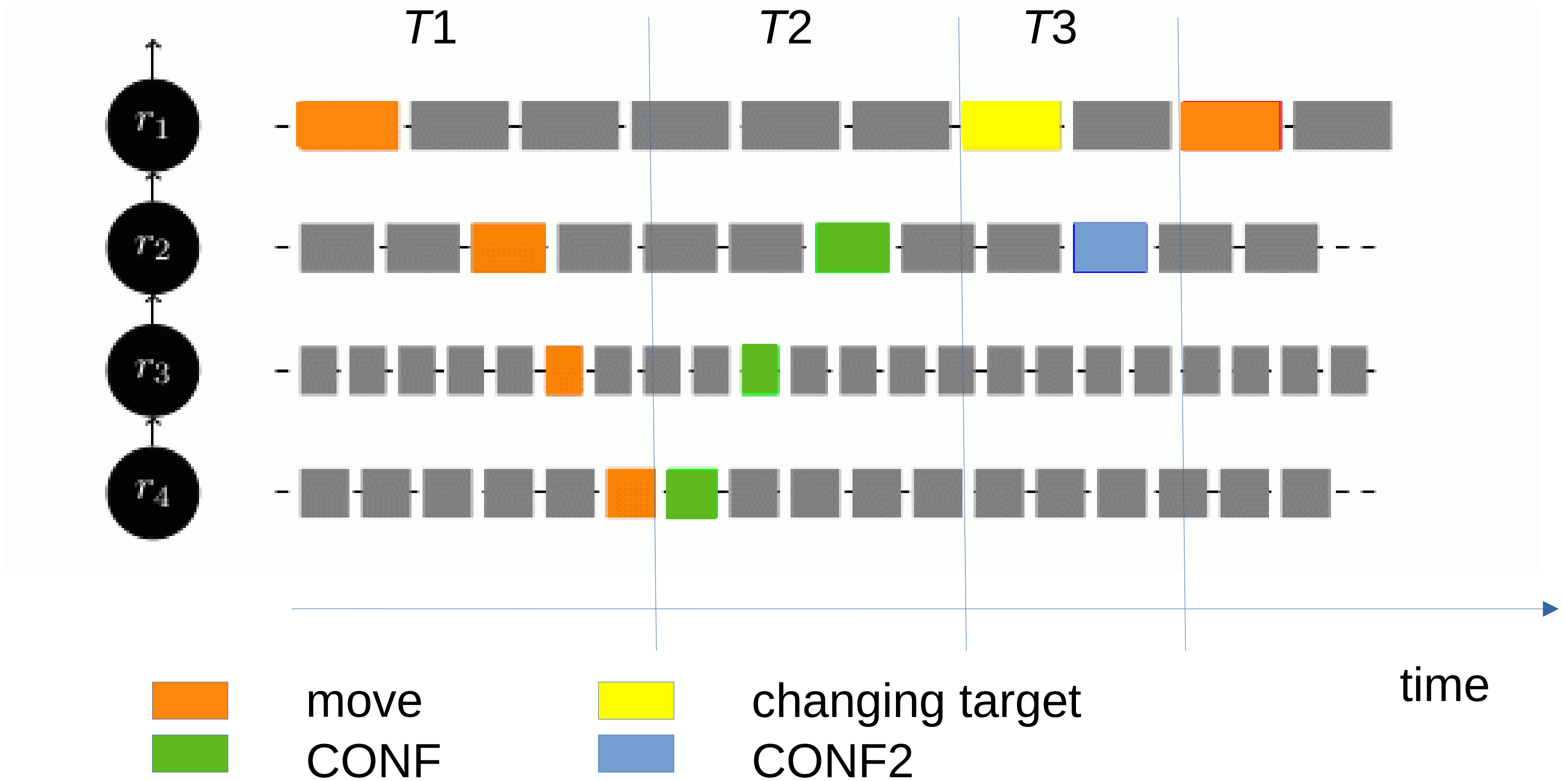
PACK – ANALYSIS

Proof idea for the running time:

T : time until occupying a new unvisited vertex

a) Leader has an unoccupied neighboring vertex

$T = T1 + T2 + T3 \leq 2i + 2 \leq 2n$ rounds



Attila Hideg & Tamás Lukovszki

PACK – ANALYSIS

Proof idea for the running time:

b) Leader has no unoccupied neighboring vertex
Taking the leadership : ≤ 5 rounds

Each vertex can take the leadership only once.
Time for all leadership taking: ≤ $5n$ rounds.

Overall time for PACK: ≤ $n\ (2n + 5) = O(n^2)$ rounds.

O(1) COLORS – MOD-PACK ALGORITHM

- Idea:
  - Encode the $L = \Delta + 4$ colors by a sequence of $\lceil \log L \rceil$ bits and
  - transmit this sequence by emulating the Alternating Bit Protocol (ABP), also referred to as Stop-and-wait ARQ

- **Theorem 2:** The modified PACK algorithm fills a connected graph in the ASYNC model by robots having a
  - visibility range of 1 hop,
  - $O(\log \Delta)$ bits of persistent storage, and
  - $O(1)$ colors.
  - The algorithm needs $O(n^2 \log \Delta)$ asynchronous rounds.

2-HOP VISIBILITY – BLOCK ALGORITHM

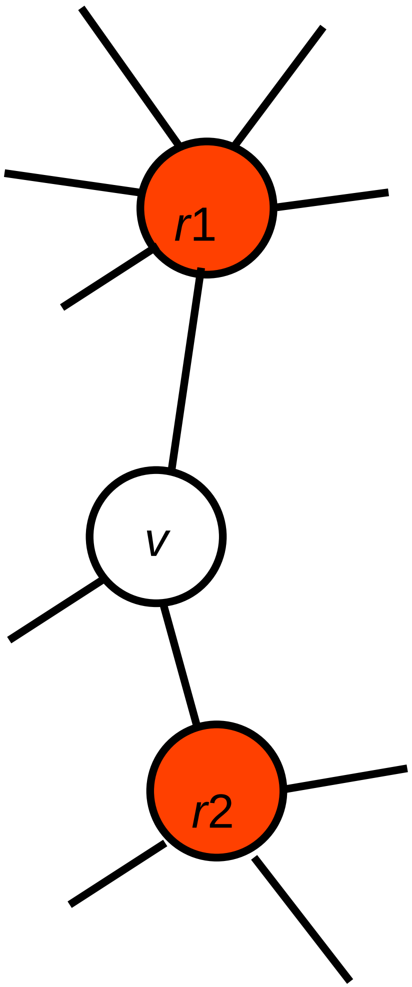
- **Idea:**
  - The Leader *r* sees all robots, that could move to the same target
  - The Leader only chooses a vertex *v* as the target, if the 1 hop neighborhood of *v* does not contain any other robot with the light turned on
    - except when the light showing direction Δ (wants to switch to Finished state)
  - A vertex neighboring to a robot with its light on (except the color Δ) is considered as blocked vertex for the Leader.

VISIBILITY: 2-HOP – BLOCK ALGORITHM

- **Theorem 3:** Algorithm BLOCK fills the area represented by a connected graph in the ASYNC model by robots having a
  - visibility range of 2 hops,
  - $O(\log \Delta)$ bits of persistent storage, and using
  - $\Delta + 4$ colors, including the color when the light is off.

MULTIPLE DOORS – *K*-DOOR-BLOCK ALGORITHM

- *k* Doors, $k \geq 2$
- *k* chains
- Assume, robots entering from different doors have distinct colors.
- Priority protocol:
  - We define a strict total order between these colors, called priority order.
  - Taking leadership:
    new Leader takes the color of the old (Finished) Leader



Attila Hideg & Tamás Lukovszki

MULTIPLE DOORS – K-DOOR-BLOCK ALGORITHM

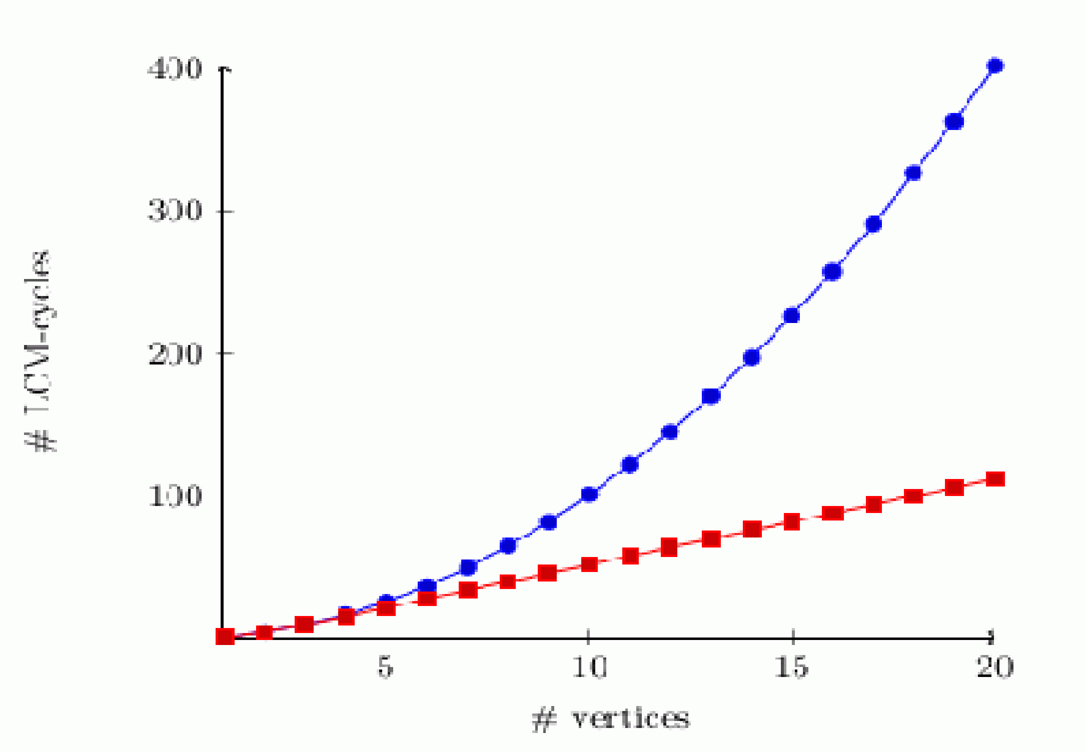
- **Lemma 5.** Leader only moves to unvisited vertices.
- **Lemma 6.** There can be at most *k* Leader at a time (one Leader per Door).
- **Lemma 7.** Robots can not collide.
- **Lemma 8.** BLOCK fills the area represented by a connected graph.

MULTIPLE DOORS – K-DOOR-BLOCK ALGORITHM

- **Theorem 4:** Algorithm BLOCK extended with the Priority protocol solves the *k*-Door Filling problem, $k \geq 2$, in the ASYNC model with robots having a
  - visibility range of 2 hops,
  - $O(\log \Delta)$ bits of persistent memory and using
  - $\Delta + k + 4$ colors including the color when the light is off.
  - BLOCK needs $O(n)$ asynchronous rounds.

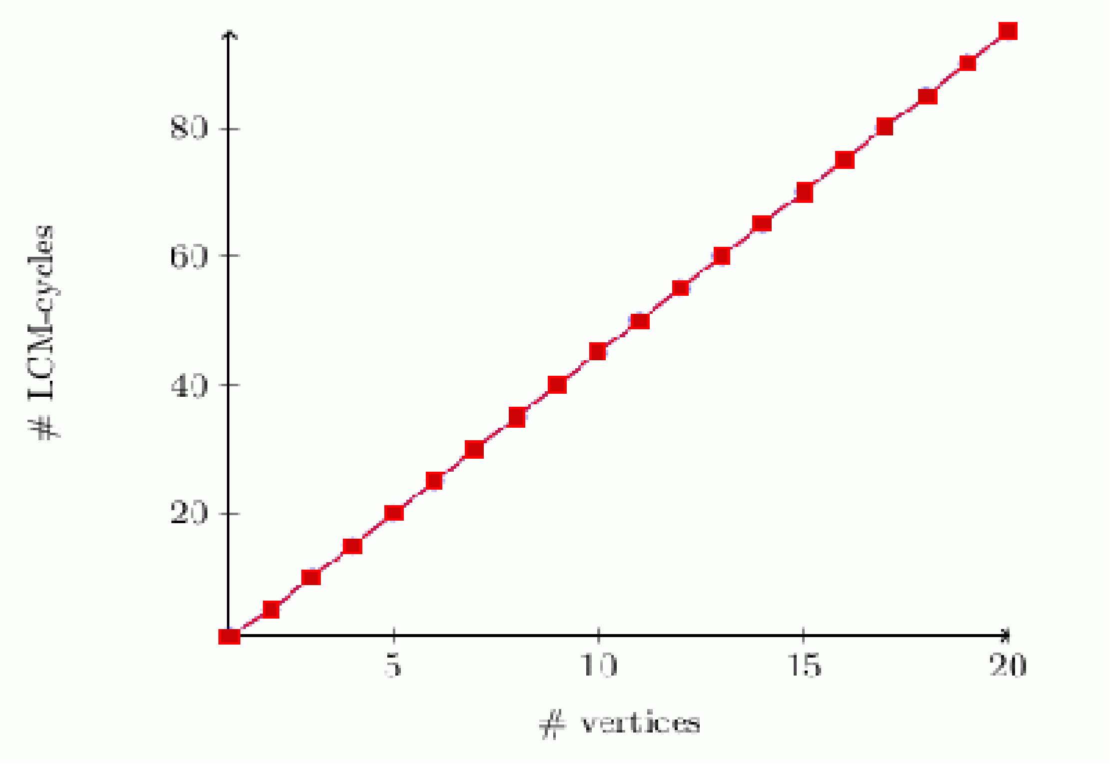
SIMULATIONS

Synchronous scheduler
Line graph, n = 1,…,20



SIMULATIONS

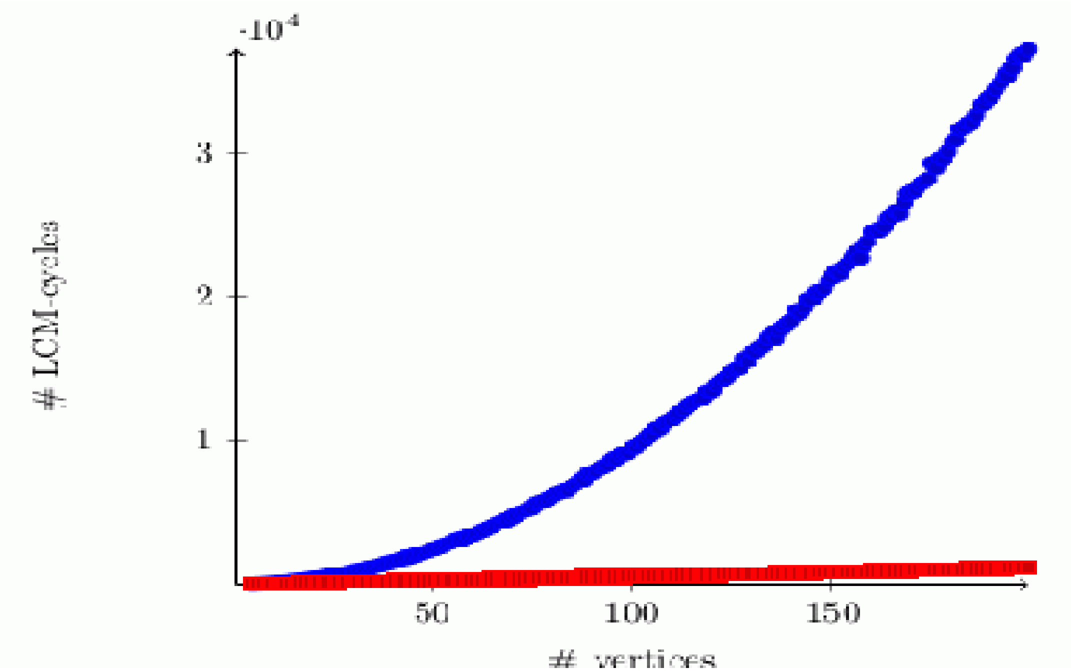
Synchronous scheduler
Star graph, n = 1,…,20



Attila Hideg & Tamás Lukovszki

SIMULATIONS

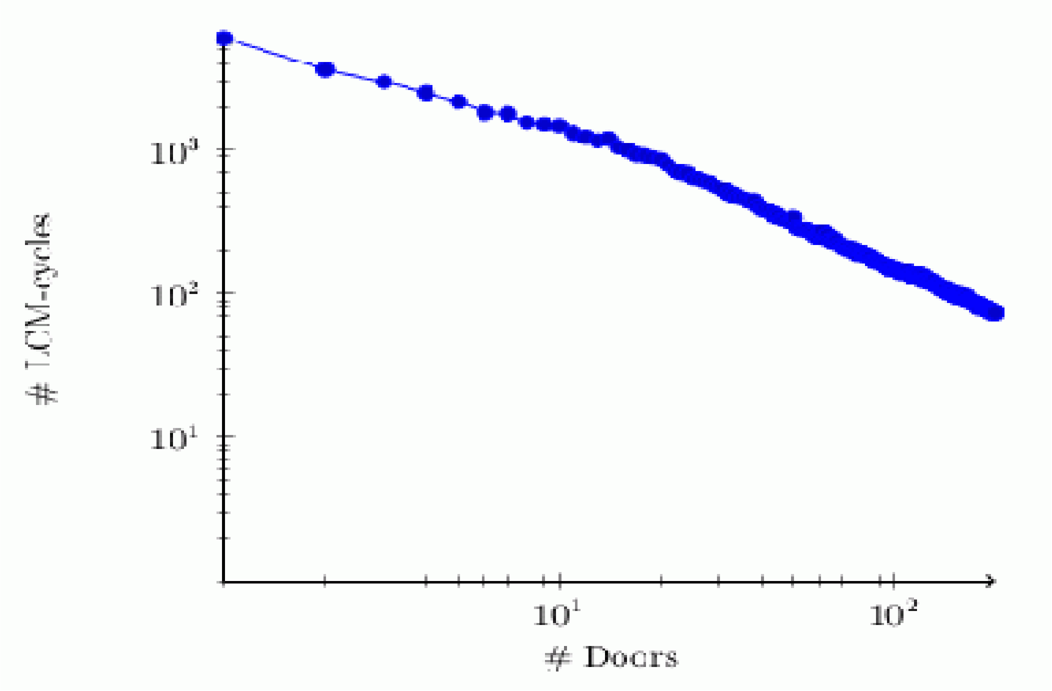
Synchronous scheduler
Delaunay graph, vertices distributed uniformly at random in $[0,1]^2$
$n = 3,\ldots,200$
50 runs for each n



Attila Hideg & Tamás Lukovszki

SIMULATIONS

Synchronous scheduler
Delaunay graph, vertices distributed uniformly at random in $[0,1]^2$
n = 1000
k = 1,…,200
50 runs for each k



Attila Hideg & Tamás Lukovszki

SUMMARY

Method	FSYNC/ ASYNC	Doors	Viewing range	Runtime #async rounds	Persistent memory bits	Colors	Area (Orthogonal/ Arbitrary)
PACK	ASYNC	Single	1	$O(n^2)$	$O(\log \Delta)$	$\Delta+4$	A
Mod-PACK	ASYNC	Single	1	$O(n^2 \log \Delta)$	$O(\log \Delta)$	$O(1)$	A
BLOCK	ASYNC	Single	2	$O(n)$	$O(\log \Delta)$	$\Delta+4$	A
k-Door-BLOCK	ASYNC	Multiple	2	$O(n)$	$O(\log(\Delta+k))$	$\Delta+k+4$	A

First asymptotic bounds for filling in the ASYNC model.

Only termination in finite time has been proven in previous works.

SUMMARY

Method	FSYNC/ ASYNC	Doors	Viewing range	Runtime #async rounds	Persistent memory bits	Colors	Area (Orthogonal/ Arbitrary)
PACK	ASYNC	Single	1	$O(n^2)$	$O(\log \Delta)$	$\Delta+4$	A
Mod-PACK	ASYNC	Single	1	$O(n^2 \log \Delta)$	$O(\log \Delta)$	$O(1)$	A
BLOCK	ASYNC	Single	2	$O(n)$	$O(\log \Delta)$	$\Delta+4$	A
k-Door-BLOCK	ASYNC	Multiple	2	$O(n)$	$O(\log(\Delta+k))$	$\Delta+k+4$	A

First asymptotic bounds for filling in the ASYNC model.

Only termination in finite time has been proven in previous works.

Open question:

- Can the runtime be reduced for robots with visibility range 1?

Thank you for your attention!