Rectangle Enclosed Triangular Grid Exploration with Myopic Luminous Robots

Raja Das
Department of Mathematics
Jadavpur University
Kolkata, West Bengal, India
rajad.math.rs@jadavpuruniversity.in

Pritam Goswami
Department of Computer Science and Engineering
Sister Nivedita University
Kolkata, West Bengal, India
pgoswami.cs@gmail.com

Brati Mondal
Department of Mathematics
Jadavpur University
Kolkata, West Bengal, India
bratim.math.rs@jadavpuruniversity.in

Buddhadeb Sau
Department of Mathematics
Jadavpur University
Kolkata, West Bengal, India
buddhadeb.sau@jadavpuruniversity.in

Abstract

One of the fundamental problems of distributed systems that has been extensively studied is the exploration of different network topologies. In exploration, each node of the graph network has to be visited by at least one robot within a finite time. Existing literature typically assumes on various networks like lines, rings, tori, arbitrary networks, and rectangular grids. To the best of our knowledge, no existing work has addressed the exploration problem considering triangular grid. In this work the exploration problem is considered on *rectangle enclosed triangular grid* (RETG) with myopic robots, where the visibility of the robot is limited to a specific distance, denoted as ϕ, since infinite visibility becomes impractical for a very large network. The robots are luminous and work under FSYNC scheduler. Firstly the cases where the perpetual RETG exploration is not possible are discussed in three impossibility results. Then two algorithms are provided to solve the exploration problem on RETG. The first algorithm requires two robots without common chirality, $\phi = 1$, and three colors of light. The second algorithm requires two robots with common chirality, $\phi = 2$, and two colors of light. Using luminous robots to decrease both visibility and the number of robots is a nice trade-off from the implementation's point of view as increasing visibility is more expensive than using robots with a light having finite colors.

CCS Concepts

• **Theory of computation** → **Distributed algorithms**.

Keywords

Myopic robot, Autonomous robots, Robot with lights, Triangular grid, Perpetual exploration, Distributed algorithms

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).
ICDCN 2025, January 04–07, 2025, Hyderabad, India
© 2025 Copyright held by the owner/author(s).
ACM ISBN 979-8-4007-1062-9/25/01
https://doi.org/10.1145/3700838.3700841

ACM Reference Format:
Raja Das, Pritam Goswami, Brati Mondal, and Buddhadeb Sau. 2025. Rectangle Enclosed Triangular Grid Exploration with Myopic Luminous Robots. In *26th International Conference on Distributed Computing and Networking (ICDCN 2025), January 04–07, 2025, Hyderabad, India.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3700838.3700841

1 Introduction

In the realm of distributed systems, there has been substantial and active research on swarm robot algorithms over the past two decades. A swarm of robots typically comprises multiple simple and cost-effective units that operate autonomously to fulfill specific tasks, such as gathering at a point, forming patterns, exploring a network, and many others. The appeal of this research area lies in its numerous real-world applications, such as patrolling areas inaccessible to humans, cleaning large surfaces, maintaining networks, etc.

In most of the works in this field, the robots are considered to have no unique identifiers i.e., they are *anonymous*. They are also considered to be physically indistinguishable and run the same algorithm i.e., they are *identical* and *homogeneous*. The robots operate in a Look-Compute-Move cycle also known as LCM cycle. A robot has two states *idle* state and *non-idle* state. *Activation* of a robot denotes the transformation from its idle state to the non-idle state. The non-idle state has three phases called Look phase, Compute phase and Move phase. In the Look phase, a robot takes a snapshot of its visible surroundings and gets information about other robots it can see, according to its own local coordinate system. In the Compute phase the robot runs an algorithm with the information from the Look phase as input and as an output it gets a location where it moves during the Move phase. After completing these phases, the robot reverts to the idle state until the next activation, constituting the recurring LCM cycle.

The activation moment and duration of a robot's activity have an impact on the snapshots of other robots in a swarm robot system. Therefore, the manner in which robots are activated plays a crucial role. It is presumed that a *scheduler* is responsible for activating the robots and managing the time they spend executing the LCM cycle. In existing literature, three types of schedulers are identified: *fully* *synchronous* (FSYNC), *semi-synchronous* (SSYNC), and *asynchronous* (ASYNC).

In FSYNC and SSYNC schedulers, time is partitioned into rounds of equal length. Robots are activated at the start of a round and synchronously go through the Look, Compute, and Move phases once in that round. The time allocated for each phase is uniform for all robots. The key distinction between FSYNC and SSYNC schedulers lies in the number of robots activated in a round. In an FSYNC scheduler, all robots are activated in each round, while in an SSYNC scheduler, only a nonempty subset of robots may be activated in a round.

In contrast, in an ASYNC scheduler, there is no concept of rounds. At any given moment, a robot could be in its Look phase, while others might be in an idle state, Compute phase, or executing the Move phase. Furthermore, the time duration for each of the Look, Compute, and Move phases is not consistent for robots in an ASYNC scheduler.

1.1 Background and Motivation

In this work, we are focusing on the problem of exploration in graph networks where each node has to be visited by at least one robot within a finite time. There are two variations of exploration, *perpetual* and *terminating*. In perpetual exploration, every node needs to be visited infinitely often while in terminating exploration the robots terminate within a finite time after each node is visited at least once by some robot. These fundamental problems have been extensively investigated in the past two decades across various network topologies, including rings (terminating exploration in [11, 14, 15] and perpetual exploration in [1]) and finite rectangular grids (terminating in [8, 9] perpetual in [2]). Other than these, terminating exploration is also studied in various other network topologies such as lines ([13]), trees ([12]), tori ([10]), and arbitrary networks ([5]). Existing research typically assumes unlimited visibility for each robot, meaning that each robot can perceive all other robots in the network. However, this assumption becomes impractical for very large networks. Consequently, recent works have shifted focus to myopic robots, where a robot's visibility is limited to a specific distance, denoted as ϕ. Terminating exploration on rings with $\phi = 1$ and $\phi = 2, 3$ was studied by Datta et al. in [6] and [7] respectively. The limitation of myopic robots without persistent memory and communication capabilities made exploration impossible in various scenarios. To address this, many studies propose the use of *Luminous* robots that are equipped with persistent lights, serving as both constant memory and a communication medium for a robot.

Exploration of infinite grids was explored by Bramas et al. ([3]) involving myopic luminous and non-luminous robots. In ([4]), perpetual exploration of a finite rectangular grid was first investigated using myopic luminous and non-luminous robots, assuming common chirality (i.e., all robots agreeing on a common clockwise direction) under the FSYNC model. The same problem was then studied in ([16]), assuming robots do not share common chirality.

In this work, we have considered the problem of exploration of a *rectangle enclosed triangular grid* (RETG) with myopic robots under the luminous model. Informally, a *rectangle enclosed triangular grid* is the part of an infinite triangular grid that is on or inside a rectangle embedded in the infinite triangular grid in such a way that a pair of parallel sides of the rectangle aligns with a pair of parallel sides of the infinite triangular grid.

To the best of our knowledge, no existing work has addressed the exploration problem within a triangular grid. The triangular grid has recently garnered attention owing to its diverse applications in programmable matters. Additionally, when viewed from an area coverage standpoint, the maximum area is efficiently covered by a network of n sensors arranged in a triangular grid ([17]). Thus considering the exploration of a triangular grid is naturally of practical interest.

Note that we can not apply the algorithms of the rectangular grid in RETG directly. The corner nodes of a rectangular grid are of degree two and the boundary nodes of a rectangular grid are of degree three but the corner nodes of a RETG can be of degree two or degree three and the boundary nodes of a RETG can be of degree three or degree five or degree four. In fact a robot with one hop visibility at a corner node of degree three has the same view if the robot is at a boundary node of degree three in a RETG. We have overcome these challenges in this work.

1.2 Our Contribution

In this work, we have studied the problem of perpetual exploration of a *rectangle enclosed triangular grid* with myopic luminous robots under an FSYNC scheduler. Without assuming common chirality we have provided one algorithm A_{123}^{VRL}. The algorithm, A_{123}^{VRL} solves the perpetual exploration using two luminous robots with three colors having visibility $\phi = 1$. Here, using luminous robots we got the opportunity to decrease both visibility and the number of robots which is a nice trade-off as increasing visibility is more expensive than equipping a robot with a light having finite colors from the implementation's standpoint. Assuming common chirality, we have provided another algorithm A_{222}^{VRL} for perpetual exploration using two robots equipped with a light having two colors and $\phi = 2$. We provide a network comparison table (Table 1) to compare finite rectangular grid and *rectangle enclosed triangular grid* (RETG) in terms of perpetual exploration. From this table, it can be observed that,

(1) Without chirality agreement, two luminous robots having any finite colors and $\phi = 1$ can not explore a rectangular grid perpetually but two luminous robots with just three colors and $\phi = 1$ can explore a RETG perpetually.
(2) With chirality agreement the best-known algorithm to explore a rectangular grid perpetually requires two luminous robots having two colors and visibility $\phi = 2$. This is also sufficient to explore a RETG perpetually.

Thus our results are also kind of an indication that RETG is more practical in terms of exploration than a rectangular grid.

2 Model, Definitions and preliminaries

Consider an infinite triangular grid i.e. a graph with infinite nodes (say N) and each node is of degree six and each face of the graph is identical to an equilateral triangle of unit-length sides. Note that an infinite triangular grid has three families of parallel sides. Now we embed a rectangle on the infinite triangular grid in such a way that a pair of parallel sides of the rectangle aligns with a pair of

Network	ϕ	# Robots	# Colors	Chirality	Algorithm
Rectangular Grid	1	2	Finite	No	Impossible ([16])
Rectangular Grid	1	3	3	No	$Vone_3^3$ ([16])
RETG	1	2	3	No	A_{123}^{VRL}[This work]
Rectangular Grid	2	2	2	Yes	$Vtwo_1^2$([4])
RETG	2	2	2	Yes	A_{222}^{VRL}[This work]

Table 1: Network Comparison Table

parallel sides of the infinite triangular grid. Let N' be the subset of N and all the nodes which are on or inside the rectangle are elements of N'. The subgraph induced by N' is called a *rectangle enclosed triangular grid* (RETG). In this work we have considered all possible RETGs (see Fig 1) depending on the degree of the nodes.

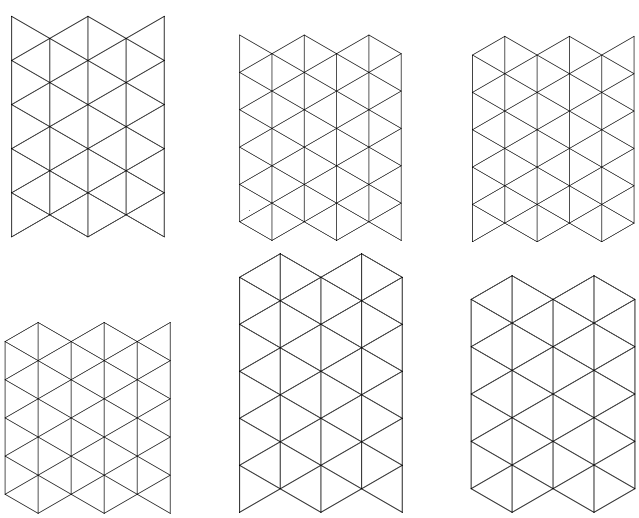

Figure 1: Different type of RETGs

In this work, we have considered a fully synchronous scheduler and luminous robots with one and two hop visibility.

In this work, the robots are fully disoriented. We have given an algorithm where the robots have consistent chirality i.e. the clockwise sense remains the same in each round. In another algorithm, the robots have a common chirality.

Definition 2.1 (Column). The straight lines of a *rectangle enclosed triangular grid* (RETG) are called columns if they are parallel to a side of the rectangle.

The columns of a RETG are denoted as C_i, where $1 \leq i \leq k$. C_i being the i-th column from "left". This "left" is from a global perspective, required only for the proof. The robots do not agree in any direction.

In this work, we have considered the RETGs with at least four columns and each column with at least four nodes.

Definition 2.2 (Diagonal lines). The straight lines of a *rectangle enclosed triangular grid* (RETG) are called diagonal lines if they are not parallel to any side of the rectangle.

In a RETG all nodes that are topmost in some column, together form the *upper boundary* of the RETG. Similarly, all the lowest nodes of each column form the *lower boundary*, nodes on the leftmost column form the *left boundary*, and nodes on the rightmost column form the *right boundary*. By *Corner nodes*, we mean the topmost and lowest nodes of both C_1 and C_k.

Our aim is to provide some algorithms, which will be executed by the robots presented on the nodes of a RETG so that the robots will visit each node of the RETG infinitely many times without any collision. The robots must form some specific configuration at the beginning of each algorithm. A_{ijk}^{VRL} denotes algorithm with i hop visibility, j robots, and k colors of light. In A_{123}^{VRL} the robots have consistent chirality i.e. the clockwise sense remains the same in each round. In A_{222}^{VRL} the robots have common chirality.

3 Impossibility results

In this section, we have given three impossibility results. Here we have given the sketches of the proofs.

THEOREM 3.1. *The perpetual RETG exploration is not possible with only one robot, for any number of light and any finite visibility range.*

Sketch of the proof: We have considered sufficient large RETG such that if a robot stays in the middle of the RETG, the robot can not see the boundary even after one hop movement in any direction. Since the robot is inconsistent, the adversary can set the local coordinate system in each round in such a way that the robot will move between two nodes only.

THEOREM 3.2. *The perpetual RETG exploration is not possible with only two robots, for one light and two hop visibility range.*

Sketch of the proof: We have considered sufficient large RETG such that if two robots stay at one hop distance from the middle of the RETG, the robots can not see boundary even after a finite number of one hop movement in any direction. Since the robots are inconsistent, adversary can set the local coordinate system in each round in such a way that the robot will move around a few nodes only or they become separated from each other's visibility range and Theorem 3.1 follows.

THEOREM 3.3. *The perpetual RETG exploration is not possible with only two robots, for any number of light and zero visibility range.*

Sketch of the proof: Since the robots are inconsistent and the visibility range is zero, adversary can set the local coordinate systems in each round in such a way that a robot will collide with the other robot within finite rounds irrespective of initial positions.

4 Algorithm A_{123}^{VRL}

In this section we have introduced the algorithm A_{123}^{VRL} for two robots with one hop visibility and three colors of light. We have given some definitions and preliminaries first. Then we have given the description of the algorithm A_{123}^{VRL}.

4.1 Definitions and preliminaries

Definition 4.1 ($InitA^{VRL}_{123}$). A configuration with two robots that are on adjacent nodes of a RETG and one robot (say r_1) with color L or R and another robot (say r_2) with color F is called an $InitA^{VRL}_{123}$ configuration and denoted as (r_1, r_2).

A^{VRL}_{123} algorithm is initiated assuming the initial configuration to be an $InitA^{VRL}_{123}$. Also since the scheduler is fully synchronous the algorithm A^{VRL}_{123} ensures that the configuration always remains an $InitA^{VRL}_{123}$ configuration.

Definition 4.2 (Erect configurations). A configuration with two robots that are on the same column and are on adjacent nodes is called an erect configuration.

In an erect configuration the robot which is above the other robot on some column C_i from a global perspective is called the *upper robot* and the other robot is called the *lower robot*. By $E^{clr_1}_{clr_2}$ we denote an erect configuration where the upper robot has color clr_1 and the lower robot has color clr_2. In our algorithm **??** there can be four types of erect configurations. These are E^L_F, E^F_R, E^R_F and E^F_L. Note that, the robots can not distinguish which one is upper and which one is lower robot as they do not have any directional agreement. The definition here is only for the purpose of proof.

Definition 4.3 (E_1-type erect configurations). E^L_F, E^F_R configurations on C_1; E^R_F, E^F_L configurations on C_k and E^L_F, E^F_R, E^R_F, E^F_L configurations on C_i for $2 \le i \le k-1$ are called E_1-type erect configurations.

Definition 4.4 (E_2-type erect configurations). E^R_F, E^F_L configurations on C_1 and E^L_F, E^F_R configurations on C_k are called E_2-type erect configurations.

Definition 4.5 (Diagonal configurations). A configuration consisting of two robots that are on the adjacent nodes and which is not an erect configuration is called a diagonal configuration.

Definition 4.6 (d_1-type diagonal configurations). Let v be a corner node of a RETG such that $\delta(v) = 3$. Then a configuration will be called a d_1-type diagonal configuration if one of the four following conditions is satisfied.

(1) v is the upper boundary node of C_1. A robot with color L is on v and another robot with color F is on the upper boundary node of C_2.
(2) v is the lower boundary node of C_1. A robot with color R is on v and another robot with color F is on the lower boundary node of C_2.
(3) v is the upper boundary node of C_k. A robot with color R is on v and another robot with color F is on the upper boundary node of C_{k-1}.
(4) v is the lower boundary node of C_k. A robot with color L is on v and another robot with color F is on the lower boundary node of C_{k-1}.

Definition 4.7 (d_2-type diagonal configurations). Any diagonal configuration which is not a d_1-type diagonal configuration is called a d_2-type diagonal configuration.

4.2 Description of A^{VRL}_{123}

At the beginning of A^{VRL}_{123} two robots can be anywhere on a RETG provided they are on adjacent nodes and two robots with different colors. Without this assumption, the problem can not be solved because if two robots have the same color either they will be in a livelock situation or the visibility graph will be disconnected. So one of the robots (say r_1) is with color L or R and another robot (say r_2) is with color F. r_1 is the leader and r_2 is the follower. The color L indicates *Left*, R indicates *Right*, and F indicates *follower*. The robot r_1 with color L or R can switch between these two colors. The robot r_2 with color F never changes color and always follows r_1. By definition 4.1 these initial configurations are denoted as $InitA^{VRL}_{123}$. Since the scheduler is fully synchronous A^{VRL}_{123} ensures $InitA^{VRL}_{123}$ configuration remains $InitA^{VRL}_{123}$ configuration after each round.

Now we present a high level idea of the algorithm A^{VRL}_{123}. We first ensure that if the configuration is an $E_1 - type$ erect configuration then the perpetual exploration of the RETG can be done by the robots. On the other hand if the initial configuration is not an $E_1 - type$ erect configuration, the primary goal of our algorithm is to form an $E_1 - type$ erect configuration first.

4.2.1 Formation of $E_1 - type$ erect configuration : If (r_1, r_2) is not E_1-type erect then (r_1, r_2) must be anyone among E_2-type erect, d_1-type diagonal and d_2-type diagonal. The first target of the algorithm is to make the configuration E_1-type erect. If $InitA^{VRL}_{123}$ is a d_2-type diagonal configuration then the configuration will become E_1-type erect. If $InitA^{VRL}_{123}$ is a d_1-type diagonal configuration then the configuration will become E_1-type erect. If $InitA^{VRL}_{123}$ is a E_2-type erect configuration then the configuration will become E_1-type erect.

4.2.2 Exploration of the RETG from $E_1 - type$ erect configuration : In E_1-type erect configuration the robots belong to the same column. Then the robots move along the column until r_1 reaches the boundary maintaining the E_1- type erect configuration. After that the robots go to the next column with E_1- type erect configuration and start moving in the opposite direction i.e. if the robots were going downwards through the previous column then they will move upwards through the current column or if the robots were going upwards through the previous column then they will move downwards through the current column.

Suppose the robots move upwards through C_i where $2 \le i \le k-2$ with r_1 having color R then after reaching the top most node of C_i the robots will enter C_{i+1} and start moving downwards with r_1 having color L. After reaching the bottom most node of C_{i+1} the robots will enter C_{i+2} and start moving upwards with r_1 in color R. In this way, the robots explore each column from C_{i+1} to C_{k-1} and will reach C_k. After exploring C_k the robots will enter C_{k-1} or C_{k-2} depending on the degree of the last visited node of C_k. If the degree is two then the robots will enter C_{k-1} and if the degree is three then the robots will enter C_{k-2}. The color of r_1 will be the color which helps the robots to explore columns from higher indices to lower indices. It may happen that the robots miss to explore C_{k-1} now but after returning from C_1 they will explore C_{k-1}.

Similarly, it may happen that the robots miss to explore C_2 after exploring C_1 but C_2 is already explored just before the exploration of C_1 or C_2 will be explored when the robots will return from C_k.

We can observe when C_1 or C_k is explored twice, all columns will be explored. Thus after finite rounds, all the nodes of RETG will be explored.

5 Algorithm A_{222}^{VRL}

In this section, we have introduced the algorithm A_{222}^{VRL} for two robots with two hop visibility and two colors of light.

Definition 5.1 ($InitA_{222}^{VRL}$). A configuration with two robots that are on the same column or diagonal line of a RETG and are within two hop distance from each other and one robot (say r_1) with color L and another robot (say r_2) with color F is called an $InitA_{222}^{VRL}$ configuration and denoted as (r_1, r_2).

The idea of A_{222}^{VRL} is similar to the idea of A_{123}^{VRL}. In A_{123}^{VRL} the *leader* robot can be of two colors i.e., L or R and the *follower* robot is of color F. In A_{222}^{VRL} the *leader* robot is of color L and the *follower* robot is of color F. To represent the *leader* robot L of A_{222}^{VRL} as the *leader* robot L of A_{123}^{VRL} the distance between two robots in A_{123}^{VRL} has been made two hop and to represent the *leader* robot L of A_{222}^{VRL} as the *leader* robot R of A_{123}^{VRL} the distance between two robots in A_{222}^{VRL} has been made one hop.

6 Conclusion

In this work, two different algorithms are given to explore a RETG perpetually. Three impossible results have been proven to show the algorithms are optimal with respect to certain parameters. Executing A_{123}^{VRL} two robots can explore a RETG perpetually with one hop visibility and three colors of light. Executing A_{222}^{VRL} two robots can explore a RETG perpetually with two hop visibility and two colors of light. In A_{123}^{VRL}, $'V'$ denotes the visibility range of a robot and it is one hop distance. $'R'$ denotes the number of robots needed for the algorithm and it is two. $'L'$ denotes the number of colors used for the light of a robot and it is three. A_{123}^{VRL} is optimal with respect to the number of robots if we fix $V = 1$ and $L = 3$. A_{123}^{VRL} is optimal with respect to visibility if we fix $R = 2$ and $L = 3$. A_{222}^{VRL} is optimal with respect to the number of robots if we fix $V = 2$ and $L = 2$. A_{222}^{VRL} is optimal with respect to the number of lights if we fix $V = 2$ and $R = 2$. As a future direction, it would be interesting to provide algorithms under semisynchronous scheduler and asynchronous scheduler.

Acknowledgement: The first author is supported by Council of Scientific & Industrial Research (CSIR), Govt. of India. The second author is supported by University Grants Commission (UGC), Govt. of India. The third author is supported by University Grants Commission (UGC), Govt. of India. The fourth author is supported by Science and Engineering Research Board (SERB), Govt. of India.

References

[1] Lélia Blin, Alessia Milani, Maria Potop-Butucaru, and Sébastien Tixeuil. 2010. Exclusive Perpetual Ring Exploration without Chirality. In *Distributed Computing, 24th International Symposium, DISC 2010, Cambridge, MA, USA, September 13-15, 2010. Proceedings (Lecture Notes in Computer Science, Vol. 6343)*, Nancy A. Lynch and Alexander A. Shvartsman (Eds.). Springer, 312–327. https://doi.org/10.1007/978-3-642-15763-9_29

[2] François Bonnet, Alessia Milani, Maria Potop-Butucaru, and Sébastien Tixeuil. 2011. Asynchronous Exclusive Perpetual Grid Exploration without Sense of Direction. In *Principles of Distributed Systems - 15th International Conference, OPODIS 2011, Toulouse, France, December 13-16, 2011. Proceedings (Lecture Notes in Computer Science, Vol. 7109)*, Antonio Fernández Anta, Giuseppe Lipari, and Matthieu Roy (Eds.). Springer, 251–265. https://doi.org/10.1007/978-3-642-25873-2_18

[3] Quentin Bramas, Stéphane Devismes, and Pascal Lafourcade. 2020. Infinite Grid Exploration by Disoriented Robots. In *Networked Systems - 8th International Conference, NETYS 2020, Marrakech, Morocco, June 3-5, 2020, Proceedings (Lecture Notes in Computer Science, Vol. 12129)*, Chryssis Georgiou and Rupak Majumdar (Eds.). Springer, 129–145. https://doi.org/10.1007/978-3-030-67087-0_9

[4] Quentin Bramas, Pascal Lafourcade, and Stéphane Devismes. 2023. Optimal exclusive perpetual grid exploration by luminous myopic opaque robots with common chirality. *Theor. Comput. Sci.* 977 (2023), 114162. https://doi.org/10.1016/J.TCS.2023.114162

[5] Jérémie Chalopin, Paola Flocchini, Bernard Mans, and Nicola Santoro. 2010. Network Exploration by Silent and Oblivious Robots. In *Graph Theoretic Concepts in Computer Science - 36th International Workshop, WG 2010, Zarós, Crete, Greece, June 28-30, 2010 Revised Papers (Lecture Notes in Computer Science, Vol. 6410)*, Dimitrios M. Thilikos (Ed.). 208–219. https://doi.org/10.1007/978-3-642-16926-7_20

[6] Ajoy Kumar Datta, Anissa Lamani, Lawrence L. Larmore, and Franck Petit. 2013. Ring Exploration by Oblivious Agents with Local Vision. In *IEEE 33rd International Conference on Distributed Computing Systems, ICDCS 2013, 8-11 July, 2013, Philadelphia, Pennsylvania, USA*. IEEE Computer Society, 347–356. https://doi.org/10.1109/ICDCS.2013.55

[7] Ajoy Kumar Datta, Anissa Lamani, Lawrence L. Larmore, and Franck Petit. 2013. Ring Exploration by Oblivious Robots with Vision Limited to 2 or 3. In *Stabilization, Safety, and Security of Distributed Systems - 15th International Symposium, SSS 2013, Osaka, Japan, November 13-16, 2013. Proceedings (Lecture Notes in Computer Science, Vol. 8255)*, Teruo Higashino, Yoshiaki Katayama, Toshimitsu Masuzawa, Maria Potop-Butucaru, and Masafumi Yamashita (Eds.). Springer, 363–366. https://doi.org/10.1007/978-3-319-03089-0_31

[8] Stéphane Devismes, Anissa Lamani, Franck Petit, Pascal Raymond, and Sébastien Tixeuil. 2021. Terminating Exploration Of A Grid By An Optimal Number Of Asynchronous Oblivious Robots. *Comput. J.* 64, 1 (2021), 132–154. https://doi.org/10.1093/COMJNL/BXZ166

[9] Stéphane Devismes, Anissa Lamani, Franck Petit, Pascal Raymond, and Sébastien Tixeuil. 2021. Terminating Exploration Of A Grid By An Optimal Number Of Asynchronous Oblivious Robots. *Comput. J.* 64, 1 (2021), 132–154. https://doi.org/10.1093/COMJNL/BXZ166

[10] Stéphane Devismes, Anissa Lamani, Franck Petit, and Sébastien Tixeuil. 2019. Optimal torus exploration by oblivious robots. *Computing* 101, 9 (2019), 1241–1264. https://doi.org/10.1007/S00607-018-0595-8

[11] Stéphane Devismes, Franck Petit, and Sébastien Tixeuil. 2013. Optimal probabilistic ring exploration by semi-synchronous oblivious robots. *Theor. Comput. Sci.* 498 (2013), 10–27. https://doi.org/10.1016/J.TCS.2013.05.031

[12] Paola Flocchini, David Ilcinkas, Andrzej Pelc, and Nicola Santoro. 2010. Remembering without memory: Tree exploration by asynchronous oblivious robots. *Theor. Comput. Sci.* 411, 14-15 (2010), 1583–1598. https://doi.org/10.1016/J.TCS.2010.01.007

[13] Paola Flocchini, David Ilcinkas, Andrzej Pelc, and Nicola Santoro. 2011. How many oblivious robots can explore a line. *Inf. Process. Lett.* 111, 20 (2011), 1027–1031. https://doi.org/10.1016/J.IPL.2011.07.018

[14] Paola Flocchini, David Ilcinkas, Andrzej Pelc, and Nicola Santoro. 2013. Computing Without Communicating: Ring Exploration by Asynchronous Oblivious Robots. *Algorithmica* 65, 3 (2013), 562–583. https://doi.org/10.1007/S00453-011-9611-5

[15] Anissa Lamani, Maria Gradinariu Potop-Butucaru, and Sébastien Tixeuil. 2010. Optimal Deterministic Ring Exploration with Oblivious Asynchronous Robots. In *Structural Information and Communication Complexity, 17th International Colloquium, SIROCCO 2010, Sirince, Turkey, June 7-11, 2010. Proceedings (Lecture Notes in Computer Science, Vol. 6058)*, Boaz Patt-Shamir and Tínaz Ekim (Eds.). Springer, 183–196. https://doi.org/10.1007/978-3-642-13284-1_15

[16] Arthur Rauch, Quentin Bramas, Stéphane Devismes, Pascal Lafourcade, and Anissa Lamani. 2021. Optimal Exclusive Perpetual Grid Exploration by Luminous Myopic Robots Without Common Chirality. In *Networked Systems - 9th International Conference, NETYS 2021, Virtual Event, May 19-21, 2021, Proceedings (Lecture Notes in Computer Science, Vol. 12754)*, Karima Echihabi and Roland Meyer (Eds.). Springer, 95–110. https://doi.org/10.1007/978-3-030-91014-3_7

[17] Honghai Zhang and Jennifer C. Hou. 2005. Maintaining Sensing Coverage and Connectivity in Large Sensor Networks. *Ad Hoc Sens. Wirel. Networks* 1, 1-2 (2005), 89–124. http://www.oldcitypublishing.com/journals/ahswn-home/ahswn-issue-contents/ahswn-volume-1-number-1-2-2005/ahswn-1-1-2-p-89-124/