

Brief Announcement: Universal Dancing by Luminous Robots Under Sequential Schedulers

Caterina Feletti ✉ 

Department of Computer Science, Università degli Studi di Milano, Italy

Paola Flocchini ✉ 

School of Electrical Engineering and Computer Science, University of Ottawa, Canada

Debasish Pattanayak ✉ 

Department of Computer Science and Engineering, Indian Institute of Technology Indore, India

Giuseppe Prencipe ✉ 

Department of Computer Science, Università di Pisa, Italy

Nicola Santoro ✉ 

School of Computer Science, Carleton University, Ottawa, Canada

Abstract

The **Dancing** problem requires a swarm of n autonomous mobile robots to form a sequence of patterns, aka perform a *choreography*. Existing work has proven that some crucial restrictions on choreographies and initial configurations (e.g., on repetitions of patterns, periodicity, symmetries, contractions/expansions) must hold so that the **Dancing** problem can be solved under certain robot models. Here, we prove that these necessary constraints can be dropped by considering the *LUMI* model (i.e., where robots are endowed with a light whose color can be chosen from a constant-size palette) under the quite unexplored *sequential scheduler*. We formalize the class of **Universal Dancing** problems which require a swarm of n robots starting from any initial configuration to perform a (periodic or finite) sequence of arbitrary patterns, only provided that each pattern consists of n vertices (including multiplicities). However, we prove that, to be solvable under *LUMI*, the length of the feasible choreographies is bounded by the compositions of n into the number of colors available to the robots. We provide an algorithm solving the **Universal Dancing** problem by exploiting the peculiar capability of sequential robots to implement a distributed counter mechanism. Even assuming non-rigid movements, our algorithm ensures spatial homogeneity of the performed choreography.

2012 ACM Subject Classification Theory of computation → Distributed algorithms

Keywords and phrases Luminous Robots, Sequence of Patterns, Pattern Formation, Sequential Scheduler

Digital Object Identifier 10.4230/LIPIcs.DISC.2025.56

Related Version *Full Version*: <https://doi.org/10.48550/arXiv.2508.15484> [10]

Funding This work was partly supported by NSERC through the Discovery Grant program.

Acknowledgements This research was done while Caterina Feletti was visiting Profs. Flocchini and Santoro.

1 Introduction

The *Look-Compute-Move* (LCM) model is a theoretical model used to study distributed systems of mobile robots, aka swarms [12]. Generally, a swarm is modeled as a set of punctiform mobile robots $\mathcal{R} = \{r_1, \dots, r_n\}$ acting on the Euclidean plane $\mathbb{R}^2$; they are assumed to be *anonymous* and *indistinguishable* (i.e., no internal nor external ids), *autonomous* (no central control), and *homogeneous* (they execute the same deterministic algorithm). Any



© Caterina Feletti, Paola Flocchini, Debasish Pattanayak, Giuseppe Prencipe, and Nicola Santoro; licensed under Creative Commons License CC-BY 4.0

39th International Symposium on Distributed Computing (DISC 2025).

Editor: Dariusz R. Kowalski; Article No. 56; pp. 56:1–56:7



Leibniz International Proceedings in Informatics

LIPICs Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

robot, idle by default, is repeatedly activated by a scheduler. As soon as activated, the robot performs an LCM cycle: it takes the snapshot of the system (*Look*), it executes the given algorithm calculating a position (*Compute*), and it moves straight towards the computed position (*Move*). By repeating the LCM cycle infinitely, the swarm performs a given task (aka solves a problem). Different LCM (sub) models, usually referred to as X^S , have been considered to study the solvability of a problem: $X = \text{OBLOT}$ represents the most limited model where robots are both oblivious and silent, while $X = \text{LUMI}$ represents the model where robots are embedded with a $O(1)$ -size light whose color is used both as an internal state and a communication means. Instead, S refers to the synchronization scheduler setting: **FSYNCH**, **SSYNCH**, and **ASYNCH** are the most considered ones, while the *sequential* setting (**SEQ**), which activates only one robot at each round assuming the fairness condition, has remained nearly unexplored until a few years ago [3, 7, 8, 11, 15].

Interest in **SEQ** has been driven by recent research on the well-known **Pattern Formation** problem [1, 2, 9, 13, 14, 17, 18, 19]. **Pattern Formation** for a given pattern Π (i.e., a multiset of positions in $\mathbb{R}^2$) requires the robots to rearrange themselves into a configuration similar to Π , and terminate. Indeed, the features of the model, particularly its X and S , characterize the class of patterns that a swarm can form [20]. A question arose: under which model assumptions can a swarm solve the **Universal Pattern Formation** (UPF) problem, namely form *any pattern starting from any initial configuration*? An answer is given in [11] which shows that, except for point formation, UPF is solvable under $\text{OBLOT}^{\text{SEQ}}$ without any additional assumption: consequently, UPF, with point formation, is solvable under LUMI^{SEQ} .

Recognizing the power of **SEQ**, the same interest has turned towards the quite unexplored **Dancing** problem, which asks a swarm to form a periodic or finite sequence of patterns, aka a *choreography* [4, 5, 6]. In [6] it was shown that under $\text{OBLOT}^{\text{SSYNCH}}$ robots can perform choreographies with several constraints (on patterns and choreography periodicity), assuming chirality agreement among robots; in [5], it has shown that, under LUMI and with a global chirality, the feasible choreographies are subject to fewer constraints, even assuming **ASYNCH** and non-rigid movements. Now, the question is similar: under which model assumptions can a swarm solve the **Universal Dancing** (UD) problem, namely perform *a sequence of arbitrary patterns starting from any initial configuration*?

This paper answers this question. We provide an algorithm that solves UD under LUMI^{SEQ} and considering robots that are *completely disoriented* (i.e., no agreement on origin, unit distance, axes, chirality) and suffering from *variable disorientation* (i.e., their local coordinate system may change from one LCM cycle to the next one). Robots have *strong multiplicity detection*, i.e., they can detect the number of all the robots – and their colors – lying in the same point (aka forming a multiplicity). We assume *non-rigid* movements; thus, any robot may be stopped unpredictably after having traveled at least a non-null constant distance δ , unknown to robots. Despite non-rigidity, our solution guarantees an interesting *spatial property* which provides homogeneity to the whole choreography performance: the smallest circles enclosing the patterns of the choreography (except for patterns with only two or three points) formed by the swarm are concentric. We prove that, to be solvable under LUMI , the length/period of the feasible choreographies of UD is bounded by the compositions of n into the number of colors available to the robots. Our contribution is summarized here:

► **Theorem 1.** *The Universal Dancing problem can be solved under LUMI^{SEQ} by a swarm of n disoriented robots with $k \geq 4$ colors, assuming that the choreographies have length/period $q \leq \binom{n+k-7}{k-4}$, even assuming non-rigid movements.*

2 Preliminaries

Patterns and choreographies. A pattern Π is a multiset of positions in $\mathbb{R}^2$, called *vertices*. We denote with $\text{shape}(\Pi)$ the set containing the unique positions in Π . We distinguish between four classes of patterns, namely **Point**, **TwoPoints**, **ThreePoints**, and **NPoints**, according to $|\text{shape}(\cdot)|$, which is respectively 1, 2, 3, and more than 3. We denote with $\mathcal{P}(\Pi)$ the set of patterns similar¹ to Π . A swarm *forms* Π if it stops in a configuration whose multiset of positions is in $\mathcal{P}(\Pi)$.

A *choreography* is a sequence of patterns $\mathcal{S} = (\Pi_0, \dots, \Pi_{q-1})^x$ where $\Pi_{i+1} \notin \mathcal{P}(\Pi_i)$ and with $x \in \{1, \infty\}$. If $\mathcal{S}$ is *periodic* (i.e., if $x = \infty$), then we assume that $\Pi_0 \notin \mathcal{P}(\Pi_{q-1})$ and that the sequence $\Pi_0, \dots, \Pi_{q-1}$ cannot be written as $(\Pi_0, \dots, \Pi_{h-1})^{\frac{q}{h}}$ for any $h < q$. We say that q is the *length* (*period*, resp.) of $\mathcal{S}$ if $x = 1$ ($x = \infty$, resp.). A swarm $\mathcal{R}$ *performs* $\mathcal{S}$ if, for any $j = 1, \dots, x$, there exists a series of increasing finite times $\{t_{i+(j-1)q}\}_{i=0, \dots, q-1, j=1, \dots, x}$ such that $\mathcal{R}$ forms Π_i at time $t_{i+(j-1)q}$. If $x = 1$, then $\mathcal{R}$ must remain still after time t_{q-1} .

The Dancing problems. Dancing refers to the generic problem of performing choreographies. Multiple constraints can be applied to the set of choreographies in order to make them performable (aka *feasible*). Different constraints define specific **Dancing** problems, each one formalized as the triple $\mathfrak{D} = \langle \Sigma(n), \mathcal{I}(n), \phi \rangle$, where:

- $\Sigma(n)$ is a set of patterns with n vertices;
 - $\mathcal{I}(n)$ is a set of initial configurations for a swarm of n robots;
 - ϕ is a predicate on choreographies (e.g., limiting their length, pattern repetitions, etc.).
- We say that $\mathcal{S} = (\Pi_0, \dots, \Pi_{q-1})^x$ is a *feasible* choreography for $\mathfrak{D}$ if $\Pi_i \in \Sigma(n)$ for any $i \in [0, q-1]$ and $\mathcal{S}$ satisfies ϕ . For $\mathfrak{D}$ to be well-defined, each pattern in $\Sigma(n)$ must appear in at least one feasible choreography. An algorithm $\mathcal{A}$ solves $\mathfrak{D}$ if, for any swarm $\mathcal{R}$ starting from a configuration in $\mathcal{I}(n)$, it makes $\mathcal{R}$ perform any feasible choreography for $\mathfrak{D}$. A **Dancing** problem $\langle \Sigma(n), \mathcal{I}(n), \phi \rangle$ is said to be **Universal** if $\Sigma(n) = (\mathbb{R}^2)^n$ (i.e., any pattern with n vertices), $\mathcal{I}(n) = (\mathbb{R}^2 \times \{\text{off}\})^n$ (i.e., set of all possible initial configurations where robots are off-colored), and ϕ can only bound the length/period of the choreographies. In this paper, we solve the **Universal Dancing** under a specific bound ϕ . We prove that if ϕ is a tautology $\top$, then the problem is unsolved under our model.

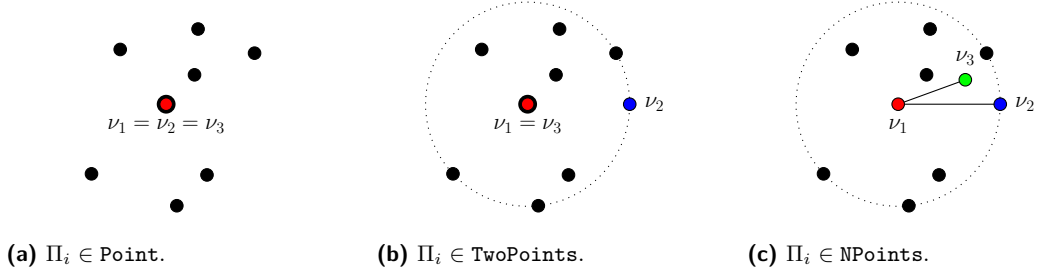
► **Theorem 2.** *The $\langle (\mathbb{R}^2)^n, (\mathbb{R}^2 \times \{\text{off}\})^n, \top \rangle$ problem cannot be solved under $\mathcal{LUMI}^{SEQ}$ even if robots have rigid movements and share a global coordinate system.*

Proof. (*sketch*) A contradiction is achieved considering $\mathcal{S} = (P_1, \Pi_1, \dots, P_z, \Pi_z, P_{z+1}, \Pi_{z+1})$, where $z = \binom{n+k-1}{k-1}$ is the number of color-compositions in a n -size swarm using k colors, and where $P_i \in \text{Point}$ for any $i \in [1, z+1]$ and $\Pi_i \neq \Pi_j$ for any $i \neq j \in [1, z+1]$. Then, there will be two times during the performance of $\mathcal{S}$ in which the swarm has the same configuration (same colors, forming P_a and P_b , with $a < b$). This may lead the swarm to perform in a loop the sub-sequence $(\Pi_a, \dots, \Pi_{b-1}P_b)^\infty$, thus never completing the choreography. ◀

3 Algorithm Outline

We here outline the algorithm that proves Theorem 1. Our algorithm uses a totally-ordered palette $\mathcal{L} = \{\ell_1, \dots, \ell_{k-3}, L_1, L_2, L_3\}$ of $k \geq 4$ colors which allows a n -size swarm $\mathcal{R}$ to perform any feasible choreography $\mathcal{S} = (\Pi_0, \dots, \Pi_{q-1})^x$ with $x \in \{1, \infty\}$ that satisfies $q \leq \binom{n+k-7}{k-4}$. By convention, $\ell_1 = \text{off}$, which is the default color to which all robots are initialized.

¹ We consider non-degenerate similarities.



■ **Figure 1** Chiral angles in different scenarios.

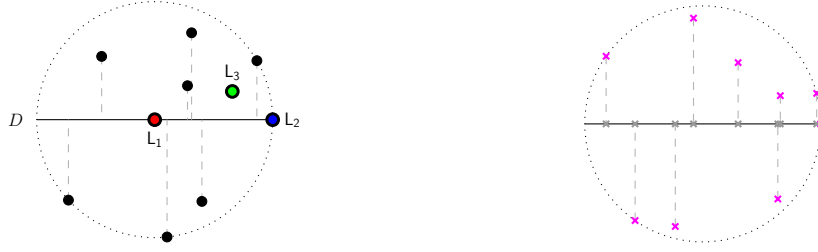
Techniques

Our algorithm adopts peculiar techniques which combine both the power of $\mathcal{LUMI}^{\text{Seq}}$ and some concepts taken and adapted from other fields.

- **Universal orientation.** The **SEQ** setting allows us to elect three leaders, *starting from any configuration*. These leaders, colored with the colors L_1 , L_2 , and L_3 , will be destined to form the *chiral angle*, a special triangle $\triangle \nu_1 \nu_2 \nu_3$ which globally defines a coordinate system for the swarm, and the position of the *actual pattern*, i.e., the pattern on whose vertices the robots must arrange. The construction of the chiral angle occurs before the formation of each pattern Π_i of the choreography and strictly depends on it; moreover, it is built to maintain spatial homogeneity to the choreography performance (e.g., ν_1 is always the same point for any Π_i). Notably, if $\Pi_i \notin \text{NPoints}$, then the vertices of the chiral angle² exactly define the points where the robots have to arrange themselves. Instead, if $\Pi_i \in \text{NPoints}$, ν_1 sets the origin $(0, 0)$, ν_2 sets the coordinate $(0, 1)$ thus the x -axis and the unit distance, and ν_3 sets the clockwise direction³. The actual pattern $\Pi \in \mathcal{P}(\Pi_i)$ is univocally defined by the chiral angle: ν_2 and ν_3 are two specific vertices of Π , and $\text{SEC}(\Pi)$ will be centered in ν_1 , so that ν_2 lies on its boundary. See Figure 1.
- **Robot-vertex matching.** The formation of each pattern is based on a technique that makes the robots match their target vertices, exploiting properties of Dyck words on the Boolean alphabet.
- **Distributed counter.** We exploit the power of $\mathcal{LUMI}^{\text{Seq}}$ by implementing a distributed *counter* to keep track of the pattern to form along the choreography. Specifically, the $n - 3$ non-leader robots, using the colors $\ell_1, \dots, \ell_{k-3}$, can encode an integer in the range $[0, z - 1]$ where $z = \binom{n+k-7}{k-4}$ is the number of weak-compositions of $n - 3$ robots with $k - 3$ colors. For this purpose, we exploit the Gray Code for compositions in [16]: that method allows us to order all the composition vectors $X_0, \dots, X_{z-1}$, so that each $X_i = (x_1, \dots, x_{k-3})$ indicates the number of robots x_j of color ℓ_j , for any $j \in [1, k - 3]$. The vector X_i thus encodes the counter value i . The Gray code ensures that X_0 corresponds to the vector where all the $n - 3$ robots are *off*, and that X_i and X_{i+1} differ at exactly two indices, say a, b , so that $X_{i+1}[a] = X_i[a] + 1$ and $X_{i+1}[b] = X_i[b] - 1$: thus, to increment the counter, an ℓ_b -colored robot must turn into ℓ_a .

² Degenerate if $\Pi_i \in \text{Point} \cup \text{TwoPoints}$.

³ When the vertices in Π_i are not aligned.



(a) Non-leader robots and their projections on D . (b) Projections of the target vertices, $\Pi_\perp$.

■ **Figure 2** PHASE 2, case $\Pi \in \text{NPoints}$.

Phases

The core of our algorithm is composed of three phases, i.e., PHASE 1, PHASE 2, PHASE 3, which are repeated cyclically until the choreography ends. For each $\Pi_i \in \mathcal{S}$, PHASE 1 aims at setting the chiral angle; PHASE 2 aims at making robots form Π_i ; PHASE 3 aims at updating the counter value. Before entering the loop, robots execute an initial phase PHASE 0.

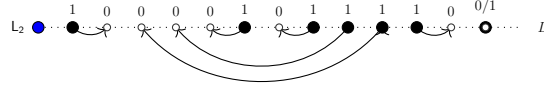
Phase 0 - Election of leaders. This phase is executed only once, starting from the initial configuration, and terminates when the three leaders have properly elected themselves by setting the dedicated colors L_1 , L_2 , and L_3 . No robot moves. The counter value now is 0.

Phase 1 - Chiral angle setup. Let i be the counter's value, and let Π_i be the corresponding pattern in the choreography. In this phase, the three leaders L_1 , L_2 , L_3 compute the chiral angle $\Delta\nu_1\nu_2\nu_3$ for Π_i , and sequentially reach the relative vertices. No other robot moves.

Phase 2 - Pattern Formation. This phase starts when the leaders have formed the chiral angle for Π_i and ends as soon as each vertex of $\Pi \in \mathcal{P}(\Pi_i)$, unequivocally defined by the chiral angle, is covered by a robot. We describe PHASE 2 for the case $\Pi \in \text{NPoints}$.

Let Ω be the circle centered in L_1 and such that L_2 lies on its boundary. By construction, all the robots are enclosed in Ω , and they will form Π so that $SEC(\Pi) = \Omega$. Note that L_2 and L_3 already lie on two vertices of Π , thus they will not move. Let D be the diameter of Ω starting from L_2 . Given a vertex v of Π , we indicate with $v_\perp$ its projection on D . Robots arrange on Π in four sub-phases:

1. **Migration towards D .** Each non-leader robot reaches D moving along a perpendicular trajectory (see Figure 2a).
2. **Arrangement on D .** Once all $m = n - 3$ non-leader robots have reached D , they compute $\Pi_\perp = \{v_\perp \mid v \in \Pi\}$, i.e., the multiset of all the projections of Π (see Figure 2b), and remove from it the projections of the three vertices intended for the leaders. Let r be an activated robot on D . Thus, r computes the boolean string $w \in \{0, 1\}^{\leq 2m}$ that represents the ordered arrangement of unmatched robots (1s) and uncovered projections (0s) along D . Note that multiplicities of robots (vertices, resp.) on D are treated by unrolling and representing them through factors of adjacent 1s (0s, resp.). Then, w can be factorized uniquely into Dyck words (or their reverses), which univocally defines the robot-projection matching (see Figure 3). Thus, each non-leader robot moves along D to reach the matching target projection. The properties of Dyck words ensure that each of the m non-leader robots will reach a vertex projection on D in finite time, even considering non-rigid movements.



■ **Figure 3** Robot-projection matching through Dyck words, where $w = 100001011110$.

3. **Back to the vertices.** In this sub-phase, all the non-leader robots travel perpendicularly to D and reach the corresponding vertices of Π . Let r be a non-leader robot on D : it unambiguously chooses the closest uncovered vertex of Π whose projection corresponds to its current position on D , and for which there does not yet exist a robot moving towards it. Note that multiple vertices (belonging to both the semi-circles cut by D) can have the same projection; in this case, r unambiguously elects the target vertex by considering the clockwise orientation of the plane. After the vertex choice, r starts traveling perpendicularly to D towards the computed vertex. Robot r may be stopped before reaching its vertex. Let h be the segment perpendicular to D which starts from D , contains r , and ends at Ω . Let $v_1, \dots, v_s$ be the list of vertices along h so that $d(v_i, D) \geq d(v_{i+1}, D)$. Then r moves to v_j where $j \in [1, s]$ is the greatest index such that v_j is not covered and r lies on the segment $[v_j, v_{j+1}]$.
4. **L_1 towards its vertex.** Lastly, L_1 moves to reach the last missing vertex of the pattern Π , which has been properly selected by the swarm.

Phase 3 - Counter Update. This phase starts as soon as each vertex of Π is covered by a robot. Let the current counter vector be $X_i = (x_1, \dots, x_{k-3})$. We distinguish three actions:

- **Termination.** If $i = q - 1$ and $x = 1$, then the swarm has terminated the choreography, and then it does not update the counter: this causes the swarm to freeze in Π_{q-1} .
- **Increment.** If $i < q - 1$, then the swarm must increment the counter from i to $i + 1$. Let X_{i+1} be the next vector. From the Gray code construction, there are exactly two differences between X_i and X_{i+1} corresponding to one decrement value and one increment value. Let a and b be the two indices that change between X_i and X_{i+1} . Thus, as soon as a robot with color ℓ_a gets activated, it changes its color to ℓ_b .
- **Resetting.** If $i = q - 1$ and $x = \infty$, then the swarm must reset the counter to the initial value X_0 , i.e., where all the non-leader robots are off-colored. For an unambiguous resetting, we make the leader L_3 set its color to L_2 : the presence of two L_2 robots signals non-leader robots that they have to reset the counter. Once all non-leader robots have turned into off, then one of the two L_2 robots turns into L_3 : if it is possible to establish which was the original L_3 , the leaders maintain their former roles.

References

- 1 Serafino Cicerone, Gabriele Di Stefano, and Alfredo Navarra. Asynchronous arbitrary pattern formation: The effects of a rigorous approach. *Distributed Computing*, 32(2):91–132, 2019. doi:10.1007/S00446-018-0325-7.
- 2 Serafino Cicerone, Gabriele Di Stefano, and Alfredo Navarra. Solving the pattern formation by mobile robots with chirality. *IEEE Access*, 9:88177–88204, 2021. doi:10.1109/ACCESS.2021.3089081.
- 3 Stefano Clemente and Caterina Feletti. Fault detection and identification by autonomous mobile robots. In *Proc. of 4th Symposium on Algorithmic Foundations of Dynamic Networks (SAND)*, pages 10:1–10:20, 2025. doi:10.4230/LIPIcs.SAND.2025.10.

- 4 Shantanu Das, Paola Flocchini, Giuseppe Prencipe, and Nicola Santoro. Synchronized dancing of oblivious chameleons. In *Proc. of 7th International Conference on Fun with Algorithms (FUN)*, volume 8496, pages 113–124. Springer, 2014. doi:10.1007/978-3-319-07890-8_10.
- 5 Shantanu Das, Paola Flocchini, Giuseppe Prencipe, and Nicola Santoro. Forming sequences of patterns with luminous robots. *IEEE Access*, 8:90577–90597, 2020. doi:10.1109/ACCESS.2020.2994052.
- 6 Shantanu Das, Paola Flocchini, Nicola Santoro, and Masafumi Yamashita. Forming sequences of geometric patterns with oblivious mobile robots. *Distributed Computing*, 28(2):131–145, 2015. doi:10.1007/s00446-014-0220-9.
- 7 Xavier Défago, Maria Gradinariu, Stéphane Messika, and Philippe Raipin Parvédy. Fault-tolerant and self-stabilizing mobile robots gathering. In *Proc. of 20th International Symposium on Distributed Computing (DISC)*, volume 4167, pages 46–60. Springer, 2006. doi:10.1007/11864219_4.
- 8 Xavier Défago, Maria Potop-Butucaru, and Sébastien Tixeuil. *Fault-Tolerant Mobile Robots*. Chapter 10 of [12], pages 234–251, 2019. doi:10.1007/978-3-030-11072-7_10.
- 9 Yoann Dieudonné, Franck Petit, and Vincent Villain. Leader election problem versus pattern formation problem. In *Proc. of 24th International Symposium on Distributed Computing (DISC)*, pages 267–281, 2010. doi:10.1007/978-3-642-15763-9_26.
- 10 Caterina Feletti, Paola Flocchini, Debasish Pattanayak, Giuseppe Prencipe, and Nicola Santoro. Universal dancing by luminous robots under sequential schedulers, 2025. arXiv:2508.15484.
- 11 Paola Flocchini, Alfredo Navarra, Debasish Pattanayak, Francesco Piselli, and Nicola Santoro. Oblivious robots under sequential schedulers: Universal pattern formation. In *Proc. 32nd International Colloquium of Structural Information and Communication Complexity (SIROCCO)*, volume 15671, pages 297–314. Springer, 2025. doi:10.1007/978-3-031-91736-3_18.
- 12 Paola Flocchini, Giuseppe Prencipe, and Nicola Santoro, editors. *Distributed Computing by Mobile Entities: Current Research in Moving and Computing*, volume 11340. Springer, 2019. doi:10.1007/978-3-030-11072-7.
- 13 Paola Flocchini, Giuseppe Prencipe, Nicola Santoro, and Peter Widmayer. Hard tasks for weak robots: The role of common knowledge in pattern formation by autonomous mobile robots. In *Proc. of 10th International Symposium on Algorithms and Computation (ISAAC)*, pages 93–102, 1999. doi:10.1007/3-540-46632-0_10.
- 14 Paola Flocchini, Giuseppe Prencipe, Nicola Santoro, and Peter Widmayer. Arbitrary pattern formation by asynchronous, anonymous, oblivious robots. *Theoretical Computer Science*, 407(1):412–447, 2008. doi:10.1016/j.tcs.2008.07.026.
- 15 Fabian Frei and Koichi Wada. Brief announcement: Distinct gathering under round robin. In *Proc. of 38th International Symposium on Distributed Computing (DISC)*, volume 319, pages 48:1–48:8, 2024. doi:10.4230/LIPIcs.DISC.2024.48.
- 16 Paul Klingsberg. A Gray code for compositions. *Journal of Algorithms*, 3(1):41–44, 1982. doi:10.1016/0196-6774(82)90006-2.
- 17 Giuseppe Prencipe. Pattern formation. In *Chapter 3 of [12]*, volume 11340, pages 37–62. Springer, 2019. doi:10.1007/978-3-030-11072-7_3.
- 18 Ichiro Suzuki and Masafumi Yamashita. Distributed anonymous mobile robots: Formation of geometric patterns. *SIAM Journal on Computing*, 28(4):1347–1363, 1999. doi:10.1137/S009753979628292X.
- 19 Ramachandran Vaidyanathan, Gokarna Sharma, and Jerry Trahan. On fast pattern formation by autonomous robots. *Information and Computation*, 285:104699, 2022. doi:10.1016/j.ic.2021.104699.
- 20 Masafumi Yamashita and Ichiro Suzuki. Characterizing geometric patterns formable by oblivious anonymous mobile robots. *Theoretical Computer Science*, 411(26-28):2433–2453, 2010. doi:10.1016/j.tcs.2010.01.037.