

Computational Power of Opaque Robots

Caterina Feletti¹, Lucia Mambretti, Carlo Mereghetti¹, and Beatrice Palano¹

¹Dipartimento di Informatica, Università degli Studi di Milano, Milan, Italy
¹{caterina.feletti, carlo.mereghetti, beatrice.palano}@unimi.it

Abstract

In the field of distributed computing by robot swarms, the research comprehends manifold models where robots operate in the Euclidean plane through a sequence of *look-compute-move* cycles. Models under study differ for (i) the possibility of storing constant-size information, (ii) the possibility of communicating constant-size information, and (iii) the synchronization mode. By varying features (i,ii), we obtain the noted four base models: *OBLLOT* (silent and oblivious robots), *FSTA* (silent and finite-state robots), *FCOM* (oblivious and finite-communication robots), and *LUMI* (finite-state and finite-communication robots). Combining each base model with the three main synchronization modes (*fully synchronous*, *semi-synchronous*, and *asynchronous*), we obtain the well-known 12 models. Extensive research has studied their *computational power*, proving the hierarchical relations between different models. However, only *transparent* robots have been considered.

In this work, we study the taxonomy of the 12 models considering *collision-intolerant opaque* robots. We present six witness problems that prove the majority of the computational relations between the 12 models. In particular, the last witness problem depicts a peculiar issue occurring in the case of *obstructed visibility* and asynchrony.

Keywords— Mobile robots, Look-Compute-Move, Computational complexity, Opaque robots, Distributed Computing, Obstructed visibility, Collision intolerance

1 Introduction

In the far-ranging field of distributed computing, a significant area concerns *computing by mobile entities* [15, 16], where tasks are required to be solved by multiple simple and limited entities (also called *robots*) that can move in the environment. In this realm, manifold theoretical models have been introduced to formalize realistic scenarios (e.g. sensor or drone swarms, dynamic networks, software agents). One of the most studied is the *look-compute-move* model [15, 16], where robots, once activated, execute a *cycle* of three steps: they *look* at the environment, they *compute* the next position executing a distributed algorithm, and they *move* to the computed position.

Under the umbrella of the *look-compute-move* macro-model, a vast combination of models has been proposed to formalize different robot capabilities and to study how model settings affect its computational power. In this respect, robots are assumed to possess very limited and restricted features, in order to find the minimal sets of capabilities which are required to achieve a given task. Accordingly, robots are assumed to be *autonomous*, *indistinguishable*, *anonymous*, and *homogeneous*: namely, they act without any central control, they cannot distinguish themselves by external appearance or by ids, they possess the same features, they execute the same algorithm in a decentralized way. Moreover, most of the literature considers *punctiform* robots which cannot communicate with other robots (*silent*), without any persistent memory (*oblivious*), without any agreement on a global coordinate system, or chirality, or a unit measure (*disoriented*). Besides robot capabilities, different model *environments* have been proposed to study diverse scenarios. The existing models can be mainly divided into two groups: the models where robots act on the Euclidean plane [1, 13, 17, 22], and the models where robots act on discrete spaces (generally graphs, rings, or lattices) [7, 8, 11, 23]. According to the *synchronization mode*, robots may be synchronized (time is globally divided into rounds) or not. Specifically, literature proposes three main modes: the *fully synchronous* mode (**FULLY**), where all robots execute each step of

the *look-compute-move* cycle synchronously in one round, the *semi-synchronous* mode (SEMI), where at each round a random subset of robots act synchronously, and the *asynchronous* mode (ASYNCH), where robots act without any synchronization assumption.

The traditional problems studied for swarms of mobile entities include **Pattern Formation** [1, 12, 13, 17, 25, 28, 29, 30], **Gathering** [5, 7, 11, 14, 21], **Scattering** [20, 24], **Flocking** [4]. A common goal of the algorithmic investigation is to reduce the model capabilities required to solve a given problem or to prove the impossibility of solving it under a certain set of capabilities. This approach has led to describing the *computational power* of a given model (i.e. the set of problems it can solve) and outlining the hierarchical relations (dominance, equivalence, or orthogonality) among different models. In the last decade, multiple works [2, 6, 8, 9, 10, 18] have inspected and compared the computational power of different models which differ in robot features and synchronization mode. According to the robot features, they have investigated how the communication and storage capabilities affect the computational power of the robots. Starting from the classical model where robots are both *oblivious* and *silent* (i.e. without any means of storage or communication), researchers have investigated how the possession of a persistent memory or communication means changes the power of such models. To characterize these extra properties, they proposed to add a *constant-size light* to each robot which can assume a color chosen among a constant and fixed set of colors. Such light is *persistent* (so the color is maintained until the next update), it can be updated at the beginning of a move step, and it can be internally or externally visible. Specifically, the literature focuses on four classes of robots: the *OBLLOT* class, where robots are assumed to be *oblivious* and *silent*, the *FSTA* class, where each robot is embedded with an *internal light* (visible just to the robot, thus providing a persistent memory), the *FCOM* class, where each robot is embedded with an *external light* (visible just to the other robots, thus providing communication means), and the *LUMI* class, where each robot is embedded with an external and internal light. According to the synchronization mode, each class has been studied under the three settings: FULLY, SEMI, and ASYNCH.

Besides some trivial relations between a pair of models that only differ because the first one enjoys a capability that the second one lacks, other model relations may not be obvious to identify. This is especially true for models characterized by completely different capabilities, so it may be difficult to understand which combination of capabilities is more powerful. In these cases, the literature has attempted to illustrate some *simulators* to prove the equivalence between models, or some *witness problems* to prove their strict dominance or orthogonality. Specifically, in [2, 6, 18, 19], the authors study the computational power of transparent robots that can move on the Euclidean plane, assuming multiple robots can occupy the same positions (*multiplicity*). In [8, 9, 10], the authors make the same effort but for robots acting on graphs. In [3], the authors consider *energy-constraint* robots, i.e. robots that necessitate an idle round to restore the needed energy to perform a new cycle.

Related works and our contributions. Our work is inspired by the papers [2, 6, 18, 19] where the authors exhibit the complete taxonomy of the 12 models of robots that can freely move on the Euclidean plane. Such models vary for the synchronization mode and for the possibility to memorize and communicate. However they are assumed to be transparent, thus always guaranteeing complete visibility for the swarm, and collision-tolerant, thus allowing robots to occupy the same position at the same time.

In this paper, we investigate the computational power of *opaque robots*, i.e. robots that cannot see beyond a collinear robot. Opacity introduces a remarkable difficulty in the design of correct algorithms to solve some classical problems [1, 12, 13]. In fact, the *obstructed visibility* leads to critical issues to be addressed in the algorithmic strategies: robots may not be aware of the cardinality of the swarm, robots may not be aware if there are some moving robots in the ASYNCH mode, robots may not know the complete topology of the current configuration, robots may compute the next action based on partial information. As a matter of fact, *ad hoc* techniques are needed to cope with this visibility limitation [26, 27].

Besides the *opacity* feature, our model differs from [2, 6, 18, 19] since robots do not tolerate collisions (so we drop the *multiplicity* assumption). The reason behind this choice is twofold, and it is coherent with the related literature [1, 12, 13, 26, 27]. Firstly, assuming collision intolerance leads to the formalization and analysis of more realistic models, as does assuming robot opacity. Secondly, dropping the multiplicity assumption is coherent with the hypothesis of obstructed visibility in the case of collinearity. As a matter of fact, a multiplicity of two robots forms a “degenerate” collinearity with any other robot of the swarm, for which it would be unnatural to state the visibility relation in this special case. In this respect, some witness problems introduced in [2, 19] cannot be applied under our model, which needs a new study with specific witness problems.

In the first part of this work, we expose a preliminary study of the relations between transparent

and opaque models. Intuitively, a transparent model seems to computationally dominate the same model but with opaque robots. In Section 3 we formally prove this strict dominance: endowing a model with transparency increases its computational power, allowing it to solve more problems. As a consequence, this result highlights that constant-size (internal or external) lights are not always sufficient to compensate for robot obstructed visibility.

In the second part of this work (Section 4), we present six witness problems showing the majority of the hierarchical relations among models of *collision-intolerant opaque robots*, thus providing a first overview of their computational taxonomy. For the sake of space, all relations proved in this work will be compactly shown in the theorems in Section 4 (i.e. without splitting them in multiple corollaries). See Appendix A for the proofs of such theorems.

2 Preliminaries

2.1 Models

This work compares 12 robot models that differ in some features. We here introduce in detail all the *core features* that such models share, and the *variable features* under study.

Core features. We investigate swarms of autonomous computational mobile robots, which act in the Euclidean plane \mathbb{R}^2 . Robots are *indistinguishable* (they cannot be distinguished by external appearance), *anonymous* (they are not provided with any id), *homogeneous* (they execute the same algorithm), and *punctiform* entities. We consider *opaque* robots so that in the case of three collinear robots p, q, r , the endpoint robots p, r cannot see each other. We assume robots are in the worst condition about orientation: they are *completely disoriented* so that they do not share a global common coordinate system (i.e. no agreement on origin, axis direction, chirality, or unit distance). Moreover, we assume that the local coordinate system of any robot may change from one activation to another (*variable disorientation*).

All the robots in the swarm are provided with the same deterministic algorithm, which is executed every time the robot is activated. At each time, a robot can be either *idle* or *active*, according to the scheduler. When activated, a robot executes a *Look-Compute-Move* cycle: it takes the snapshot of its visible area (*look*), it executes the algorithm using the sole snapshot as input (*compute*), and it travels straight towards the computed destination (*move*). If the destination position is equal to the current one, the robot is said to perform a *null movement*. After the *move* step, the robot becomes idle again. We consider *rigid* models, i.e. no adversary can stop the motion of a robot¹.

We deal with a *collision-intolerant model* meaning that it does not tolerate either multiplicity (i.e. no robot can occupy the same location as another robot at the same time) or overlapping trajectories (robots r and s have overlapping trajectories if (i) r is moving from a to a' , (ii) s is moving from b to b' , and (iii) the segments $\overline{aa'}$ and $\overline{bb'}$ have points in common). We refer to both multiplicity and overlapping trajectories as *collisions*.

Variable features. Regarding the *memory and communication* features of robots, we consider the four models mainly proposed in the literature. In the *OBLLOT* model, robots are assumed to be *oblivious* (i.e. they do not have any persistent memory to store data about past cycles) and *silent* (i.e. they do not have any means to communicate with other robots). In the *FSTA* model, robots are provided with a persistent *internal light* which can assume a color chosen from a constant-size set. Such internal light plays the role of a constant-size persistent memory. In the *FCOM* model, robots are equipped with a persistent *external light* visible only to other robots, which can assume a color chosen in a constant-size set of colors. Indeed, external lights can be exploited by the swarm to communicate some messages to the visible robots. Lastly, the *LUMI* model gather the features of both *FSTA* and *FCOM*. This model assumes *luminous* robots, which are equipped with a light that can be colored using a constant-size set of colors. Such light is both visible to the robot itself (working as an internal state) and visible to the other robots (working as an external communication means).

Regarding the *activation and synchronization* of robots, we consider the three modes mainly studied in the literature. In the *fully synchronous* mode (**FULLY**), time is split into atomic rounds, within which all robots are activated together and execute their look-compute-move steps completely synchronously. The

¹In [2, 6, 18, 19], the authors consider both rigid and non-rigid models. In the next model comparisons (transparent vs opaque), we consider only rigid models.

semi-synchronous mode (SEMI) differs from FULLY just for the fact that at each round a random subset of the swarm is activated. In the *asynchronous* mode (ASYNCH), every robot is activated independently from the others, and every cycle step lasts a finite but unpredictable amount of time. For the SEMI and ASYNCH modes, robots do not know which are the activated robots at each instant. Moreover, we always assume the *fairness condition*: for each time t and for each robot r , there exists a time $t' > t$ such that r is activated. This condition allows us to compute time complexity considering the number of *epochs*, where an epoch is a minimal time frame within which each robot is activated at least once. The selection of the subset of robots activated at every time is made by an adversarial *scheduler*. Formally, let $\mathcal{R} = \{r_1, \dots, r_n\}$ be a swarm of n robots, and let \mathcal{T} be a time domain which could be discrete $\mathbb{N}_{\geq 0}$ (in FULLY and SEMI) or continuous $\mathbb{R}_{\geq 0}$ (in ASYNCH). An *activation scheduling* is a function $S : \mathcal{T} \rightarrow 2^{\mathcal{R}}$ defining the subset of the swarm that is activated at a specific time.

Notation. We use the notation \overline{X}^Y to indicate a model for opaque robots that possess all the above core features and that has X as communication-storage setting and Y as synchronization mode, where $X \in \{\text{OBLOT}, \text{FSTA}, \text{FCOM}, \text{LUMI}\}$ and $Y \in \{\text{F}, \text{S}, \text{A}\}$ (FULLY, SEMI, ASYNCH, resp.). Consistently with the notation used in [2, 6, 18, 19], we indicate with X^Y the same model as \overline{X}^Y but considering transparent robots which tolerate collisions. We refer to these two classes of models as the *opaque* and *transparent framework*.

2.2 Problems

Robot swarms are distributed systems that are aimed at solving problems. Since in these models robots can just move in the plane, the literature studies problems requiring a swarm to form (a sequence of) geometric patterns, and/or to travel along specific trajectories. Formally, let us assume a swarm of n robots $\mathcal{R} = \{r_1, \dots, r_n\}$ on the Euclidean plane. When no ambiguity arises, we indicate with r_i both the robot and the point on the plane where r_i is located. Given an absolute coordinate system Z on \mathbb{R}^2 , we define the *configuration* of the swarm at time t as the set $C_t = \{(x_1, l_1), \dots, (x_n, l_n)\}$ where $x_i \in \mathbb{R}^2$ is the position of r_i according to Z , and l_i is the light color of r_i , at time t . In the *OBLOT* model, we always assume $l_i = \text{off}$ for every $r_i \in \mathcal{R}$. A configuration is *valid* if no collision occurs on it. We define \mathcal{C} as the set of all the valid configurations for \mathcal{R} . We say that a configuration C guarantees *complete visibility* if there are no collinearities among robots.

A *problem* P for a swarm of robots is defined² as a sequence $(\phi_0, \tau_0, \phi_1, \tau_1, \dots, \phi_m, \tau_m \dots)$ where each ϕ_i is a condition on the configuration of the swarm, and where τ_i is a condition on the intermediate configurations that the swarm is allowed to assume to reach a new configuration for which ϕ_{i+1} holds. We call such sequence the *request of the problem* P . The initial condition ϕ_0 must include the clause stating that $l_i = \text{off}$ for every $r_i \in \mathcal{R}$. Except for this clause, since P might be solved without lights and under any synchronization mode, ϕ_i, τ_i must not impose any conditions on light colors or the number of cycles, for each i .

Starting from an initial configuration C_0 for which ϕ_0 is true, P is said to *be solved* under a scheduling mode if, for each scheduling under the given mode, there exists an algorithm \mathbb{A} through which the swarm forms a sequence of configurations (C_1, \dots, C_m, \dots) such that ϕ_i holds in C_i for each $i \geq 1$, and such that τ_{i-1} holds during the formation of C_i starting from C_{i-1} . If the request of the problem is finite, the last condition τ_m requires the swarm to stay still after having satisfied the last condition ϕ_m of the request.

Given an initial configuration C_0 , a scheduling on a time domain \mathcal{T} and an algorithm \mathbb{A} solving P , we define the sequence $\{C(t)\}_{t \in \mathcal{T}}$ as the *evolution* of \mathbb{A} , where $C(t)$ is the configuration reached at time t executing \mathbb{A} according to the scheduling.

2.3 Computational Relations

Given a model M , we indicate with $\mathcal{P}(M)$ the set of problems solved under M , i.e. the *computational power* of M . Given two models M_1, M_2 , we define the following relations:

- M_1 is *computationally not less powerful* than M_2 , formally $M_1 \geq M_2$, if $\mathcal{P}(M_1) \supseteq \mathcal{P}(M_2)$, i.e any problem solvable in M_2 is solvable in M_1 ;

²For our purposes.

- M_1 is *computationally more powerful* than M_2 , formally $M_1 > M_2$, if $\mathcal{P}(M_1) \supset \mathcal{P}(M_2)$, i.e any problem solvable in M_2 is solvable in M_1 and there exists a problem solvable in M_1 that is not solvable in M_2 ;
- M_1 is *computationally orthogonal* to M_2 , formally $M_1 \perp M_2$, if $\mathcal{P}(M_1) \setminus \mathcal{P}(M_2) \neq \emptyset$ and $\mathcal{P}(M_2) \setminus \mathcal{P}(M_1) \neq \emptyset$, i.e there exists a problem solvable in M_1 (M_2 , resp.) that is not solvable in M_2 (M_1 , resp.);
- M_1 is *computationally equivalent* to M_2 , formally $M_1 \equiv M_2$, if $\mathcal{P}(M_1) = \mathcal{P}(M_2)$, i.e M_1 and M_2 solve the same set of problems.

The following relations trivially follow from the definitions of the models:

$$\mathcal{LUMI}^Y \geq \mathcal{FSTA}^Y \geq \mathcal{OBLLOT}^Y \quad \text{and} \quad \mathcal{LUMI}^Y \geq \mathcal{FCOM}^Y \geq \mathcal{OBLLOT}^Y$$

$$X^F \geq X^S \geq X^A$$

where $Y \in \{F, S, A\}$ and $X \in \{\mathcal{OBLLOT}, \mathcal{FSTA}, \mathcal{FCOM}, \mathcal{LUMI}\}$. Indeed, the same relations hold in the opaque framework.

3 Transparent vs opaque robots

Theorem 1. Let P be a problem solved in \overline{X}^Y . Then P is solved under X^Y .

Proof. Let $\overline{\mathbb{A}}$ be an algorithm solving P under \overline{X}^Y . We can easily construct an algorithm \mathbb{A} solving P under X^Y . Given a robot r and given in input its snapshot σ of all the robots, \mathbb{A} computes $\mathbb{A}(\sigma) := \overline{\mathbb{A}}(\overline{\sigma})$ where $\overline{\sigma}$ is the snapshot obtained by σ removing all the robots which would be hidden from r in case of opaqueness. \mathbb{A} perfectly simulates $\overline{\mathbb{A}}$, thus correctly solving P for transparent robots. \square

Corollary 1. For each $Y \in \{F, S, A\}$ and $X \in \{\mathcal{OBLLOT}, \mathcal{FSTA}, \mathcal{FCOM}, \mathcal{LUMI}\}$,

$$\overline{X}^Y \leq X^Y.$$

Problem 1 (Line-Stretch). Let us consider an initial configuration where $n > 3$ robots are equally spaced along the same line, say γ . Let d be the distance between two adjacent robots. The problem asks the endpoint robots to move away from their adjacent robot and stop in order to form a new distance $d + \frac{d}{n}$ with them. They are allowed to travel only along γ . The other robots must stay still. See Figure 1.



Figure 1: Line-Stretch.

Lemma 1. Line-Stretch is solved under \mathcal{OBLLOT}^A .

Proof. The problem is solved under the weakest model of the transparent framework. In fact, the endpoint robots can compute and head to their destination since they can count all the robots and at least two internal robots fix d . The final configuration is stable. \square

Lemma 2. Line-Stretch cannot be solved under $\overline{\mathcal{LUMI}}^F$.

Proof. The problem cannot be solved under the strongest model of the opaque framework. Since the n robots are always collinear by request, they cannot count themselves and so the endpoint robots will never accomplish the task. Moreover, lights would be inefficient for keeping a swarm counter, due to their constant size. \square

Theorem 2. For each $Y \in \{F, S, A\}$ and $X \in \{\mathcal{OBLLOT}, \mathcal{FSTA}, \mathcal{FCOM}, \mathcal{LUMI}\}$,

$$\overline{X}^Y < X^Y.$$

Proof. The result derives by combining Corollary 1 with Lemma 1 and Lemma 2. In fact, it holds that $\text{Line-Stretch} \in \mathcal{P}(X^Y)$ whereas $\text{Line-Stretch} \notin \mathcal{P}(\overline{X^Y})$ for any X, Y . \square

Theorem 3. Let P be a problem solved by an algorithm \mathbb{A} under X^Y always avoiding collisions, such that P is defined for a swarm with fixed cardinality, say k . If, given any evolution of \mathbb{A} , every robot can see k robots, then the problem can be solved even in $\overline{X^Y}$.

Proof. Since at any activation, each robot is aware it sees the whole swarm, it can compute its next action by executing \mathbb{A} . This computation results in the solution of the problem considering opaque robots. \square

4 Taxonomy of opaque models

We present our witness problems to prove some strict dominance ($>$) and orthogonality (\perp) relations among opaque models. Thanks to Theorem 1 and Theorem 3, one of the witness problems presented in [2] can be used to prove some hierarchical relations to hold in our opaque framework too. However, other witness problems in [2, 19] are not compliant with our collision-intolerant models; thus, we present specific problems that fit our assumptions.

4.1 Weakness of $\overline{\text{OBL\O{T}}$

Problem 2 (Triangle Round-Trip). Let C be a configuration where 3 robots are placed so that two of them lay on the vertices of an equilateral triangle (let a be the empty vertex), while the third robot lays on the triangle center. From C , the robot in the center has to move to a , forming the new configuration C' . Then, robots have to form C again, where a is again the empty vertex. See Table 1.

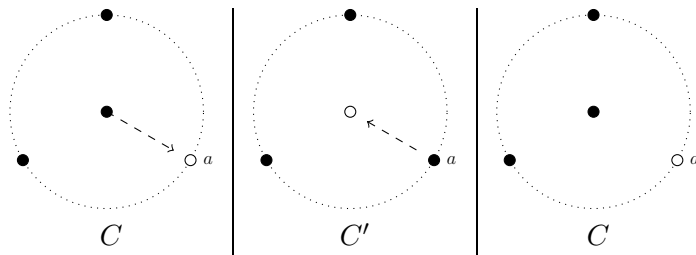


Table 1: Configurations in Triangle Round-Trip.

Triangle Round-Trip is a sub-case of the problem N -gon Round-Trip defined in [2] (see Definition 1).

Lemma 3. $\text{Triangle Round-Trip} \notin \mathcal{P}(\overline{\text{OBL\O{T}}^F})$.

Proof. The problem has been shown to not belong to $\text{OBL\O{T}}^F$ (see Lemma 3 in [2]). In fact, using oblivious and silent robots, there is no way to identify the former empty vertex a due to the full symmetry of C' . By the contrapositive of Theorem 1, the result holds. \square

Lemma 4. $\text{Triangle Round-Trip} \in \left(\mathcal{P}(\overline{\text{FSTA}}^A) \cap \mathcal{P}(\overline{\text{FCOM}}^A) \right)$.

Proof. The problem has been shown to be solved in FSTA^A and FCOM^A (see Lemma 4-5 in [2]). Since in this version of the problem the cardinality of the swarm is fixed and the robots never create collinearities or collisions, we can apply Theorem 3 to state that Triangle Round-Trip can be solved both in $\overline{\text{FSTA}}^A$ and $\overline{\text{FCOM}}^A$. \square

Theorem 4. Given the schedulers $Y_1 = F$, $Y_2 = S$, $Y_3 = A$, it holds

$$\begin{aligned} \overline{\text{FSTA}}^{Y_i} &> \overline{\text{OBL\O{T}}^{\{Y_j\}_{j \geq i}}} \\ \overline{\text{FCOM}}^{Y_i} &> \overline{\text{OBL\O{T}}^{\{Y_j\}_{j \geq i}}} \\ \overline{\text{LUMI}}^{Y_i} &> \overline{\text{OBL\O{T}}^{\{Y_j\}_{j \geq i}}}. \end{aligned}$$

4.2 Orthogonality between $\overline{\mathcal{FSTA}}$ and $\overline{\mathcal{FCOM}}$

Problem 3 (Flip-Flop-Flip). Let p, q and r be three robots forming a strictly isosceles triangle so that $\text{dist}(p, r) = \text{dist}(q, r)$. Let γ be the perpendicular bisector to the line segment \overline{pq} passing through the point $b \in \overline{pq}$. Let γ' (γ'' , resp.) be the semi-line of γ starting from b and which contains (does not contain, resp.) r . The problem requires r to perpetually perform three subsequent actions (see Table 2), in an infinite loop: (i) r must reach a point on $\gamma'' \setminus \{b\}$; (ii) r must reach a different point on γ'' in order to move away from p, q ; (iii) r must reach a point on $\gamma' \setminus \{b\}$. The problem requires r to never leave γ and to never stop so that p, q, r form an equilateral triangle. Robots p, q must stay still.

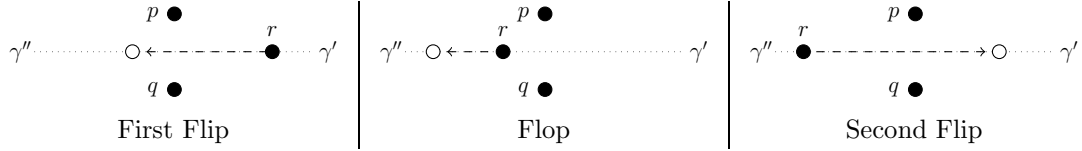


Table 2: Configurations in Flip-Flop-Flip.

Lemma 5. $\text{Flip-Flop-Flip} \in \left(\mathcal{P} \left(\overline{\mathcal{FSTA}}^{\text{A}} \right) \cap \mathcal{P} \left(\overline{\mathcal{FCOM}}^{\text{F}} \right) \right)$.

Proof. We solve the problem in these two models using three colors (**flip1**, **flop** and **flip2**), assuming w.l.o.g. all robots start with the color **flip1**. The problem request guarantees that each robot can recognize its role by geometric conditions. In $\overline{\mathcal{FSTA}}^{\text{A}}$, r moves along γ changing its internal color following the perpetual scheme $(\text{flip1} - \text{flop} - \text{flip2})^\infty$, so that at each activation, r knows which is the current action to be performed. The robots p, q do not need to change their colors. In the $\overline{\mathcal{FCOM}}^{\text{F}}$ model, all the robots synchronously update their external colors following the above scheme, so that at each round each robot knows what actions (color setting and move step) have to be accomplished. \square

Lemma 6. $\text{Flip-Flop-Flip} \notin \left(\mathcal{P} \left(\overline{\mathcal{OBL\O T}}^{\text{F}} \right) \cup \mathcal{P} \left(\overline{\mathcal{FCOM}}^{\text{S}} \right) \right)$.

Proof. Flip-Flop-Flip cannot be solved under an $\overline{\mathcal{OBL\O T}}$ model since r would not have any means to understand which movement it has to perform. Indeed, any strategy encoding the action of r into the distances with p, q fails. Suppose for example to use $u = \text{dist}(p, q)$ as a fixed measure unit, and let k, h be two fixed values, with $0 < k < h$. Suppose the algorithm implements this strategy: if $\text{dist}(p, r) < ku$, then r must execute the first flip, traveling to a position $r' \in \gamma''$ such that $ku \leq \text{dist}(p, r') < hu$. Otherwise, if $ku \leq \text{dist}(p, r) < hu$, then r must execute the flop, moving to a position $r' \in \gamma''$ such that $\text{dist}(p, r') \geq hu$. Lastly, if $\text{dist}(p, r) \geq hu$, then r must execute the second flip, moving to a position $r' \in \gamma'$ such that $\text{dist}(p, r') < ku$. Yet, since r could be placed at any position on γ' in the initial configuration, any distance encoding results inefficient for the solution of the problem.

Flip-Flop-Flip cannot be solved under the $\overline{\mathcal{FCOM}}^{\text{S}}$ model too. By contradiction, suppose that the problem is solved by a certain algorithm A . Let S be a SEMI activation scheduling under which A solves the problem. We show that there exists a SEMI activation scheduling S' such that Flip-Flop-Flip is not solved by A . Let t be the first round in S where r executes the first flip. Let S' be a scheduling such that $S'(t') = S(t'), \forall t' \leq t$. Clearly, r executes its first flip at the t -th round under S' . Suppose that, in the $(t+1)$ -th activation round under S' , r is the only one that gets activated, namely $S'(t+1) = \{r\}$. Yet, r has no memory of the previous activation rounds. As a consequence, r makes again a flip. Contradiction. \square

Theorem 5.

$$\begin{aligned} \overline{\mathcal{LUMI}}^{\text{A}} &> \overline{\mathcal{FCOM}}^{\text{A}} \\ \overline{\mathcal{LUMI}}^{\text{S}} &> \overline{\mathcal{FCOM}}^{\text{S,A}} \\ \overline{\mathcal{LUMI}}^{\text{F}} &> \overline{\mathcal{FCOM}}^{\text{S,A}} \\ \overline{\mathcal{FCOM}}^{\text{F}} &> \overline{\mathcal{FCOM}}^{\text{S,A}}. \end{aligned}$$

Problem 4 (Newcomer Introducing). Consider $n + 2$ robots, with $n \geq 7$. Let n robots be placed on the same circle whose ray length is ρ . Let c be a robot lying in the center of the circle. Let s be a robot external to the circle so that s can see c . The problem requires sequentially forming two configurations. First, s must travel along the line \overline{sc} and stop on the boundary of the circle. Second, c must travel along the radius defined by s and stop in a position c' so that $\text{dist}(s, c') = \frac{1}{2}\rho$. All the other robots must stay still. See Table 3.

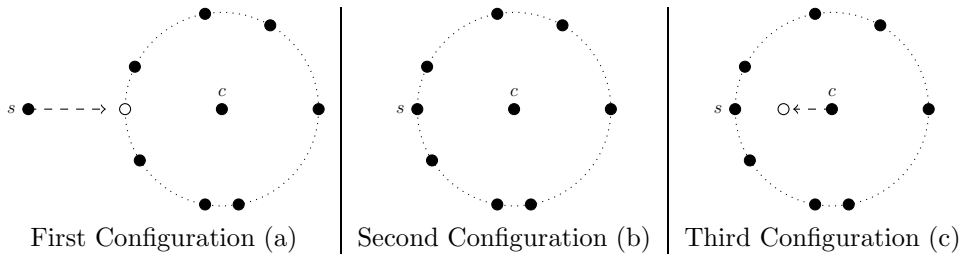


Table 3: Configurations in Newcomer Introducing.

Lemma 7. Newcomer Introducing $\notin \mathcal{P}(\overline{\mathcal{FSTA}}^F)$.

Proof. The impossibility of solving the problem with just internal lights derives from the fact that starting from the second configuration (see Table 3.b) c has no way to recognize which robot is s . Since s can be anywhere in the disposition of the $n + 1$ robots on the circle, a constant set of colors would not be sufficient to store robot indices. \square

Lemma 8. Newcomer Introducing $\in \mathcal{P}(\overline{\mathcal{FCOM}}^A)$.

Proof. We show a possible $\overline{\mathcal{FCOM}}^A$ algorithm solving Newcomer Introducing with two colors: off and s. All the robots are initially set to color off. Each robot can determine its role by the geometry of the configurations (c sees $n \geq 7$ robots equidistant from itself and an external robot, s sees at least four robots forming a circle with a robot on its center, while the other robots can see they lay on a circle with at least other $n - 2 \geq 5$ robots). When s is activated, it sets its light to s and starts to move. This color is maintained also in its next activations. When c is activated, if it sees a robot s on the circle, it can compute its destination correctly. The last configuration is stable: no other robot will move. \square

Theorem 6. Given the schedulers $Y_1 = F$, $Y_2 = S$, $Y_3 = A$,

$$\overline{\mathcal{LUMI}}^{Y_i} > \overline{\mathcal{FSTA}}^{\{Y_j\}_{j \geq i}}.$$

Theorem 7.

$$\overline{\mathcal{FSTA}}^{F,S,A} \perp \overline{\mathcal{FCOM}}^{S,A}.$$

4.3 Power of FULLY

Problem 5 (Spinning). The problem is defined recursively, without any stop conditions. Consider a configuration C where $n \geq 5$ robots $\{r_0, \dots, r_{n-1}\}$ are located on a circle centered in O . Let a_0, \dots, a_{n-1} be the related positions of the robots such that it is possible to establish a global clockwise direction (e.g. the one going from a_0 to a_2 , passing through a_1). Let α be the angle $a_0 \hat{O} a_1$, which is the minimum angle in $\{a_i \hat{O} a_{i+1}\}_{0 \leq i \leq n-1}$. The problem requires the given configuration to form a new configuration C' by rotating each r_i from a_i to a'_i of an angle $\frac{\alpha}{2}$, following the agreed clockwise direction. Robots are required only stop on the target points lying on the circumference. Recursively, the problem demands the same request starting from C' . See Table 4.

Lemma 9. Spinning $\in \left(\mathcal{P}(\overline{\mathcal{OBLCT}}^F) \cap \mathcal{P}(\overline{\mathcal{LUMI}}^A) \right)$.

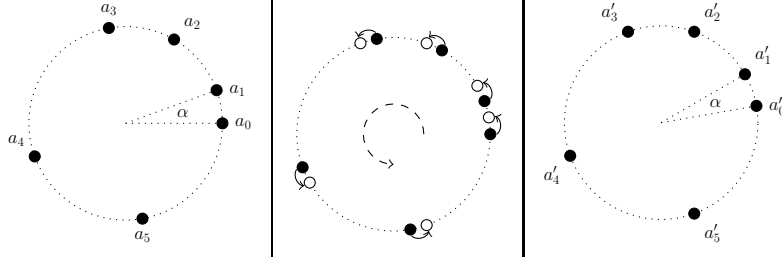


Table 4: Configurations in Spinning.

Proof. The problem is solvable in $\overline{\text{OBL\O T}}^{\text{F}}$: each robot always has complete visibility of the swarm, so it is able to determine the rotation center and the rotation angle. The FULLY mode guarantees that all the robots agree on the same rotation-angle, at each round.

The problem is solvable under $\overline{\text{LUMT}}^{\text{A}}$, by using these colors: off, a0, a1, moving0, moving1, m0, m1, moving, moved, end. The algorithm solving the problem executes the same sub-routine perpetually. This sub-routine implements a complete circle rotation of the swarm. At the beginning of each circle rotation, all robots are off. In the first epoch, the robots r_0 and r_1 set their lights as a0 and a1, respectively. After this setting, robot a0 (a1, resp.) computes its destination position, sets its light to moving0 (moving1, resp.) and starts moving. If a robot r , which is not moving0 or moving1 colored, sees a moving0 or moving1 robot, r does nothing. When a moving0 (moving1, resp.) robot is activated, it just updates its light to m0 (m1, resp.). Once the rotation angle through m0 and m1 has been fixed, the other robots can start their rotation. If an off robot r sees both m0 and m1 on the circle, it sets its light as moving and starts its rotation. When a moving robot is activated, it sets its light to moved. When a robot sees only m0, m1, moved, or end robots, then it updates its color to end. In the last phase of the sub-routine, if an end robot can see only end or off robots, it resets its color to off. Once all robots are off, the circle rotation is ready to restart. \square

Lemma 10. $\text{Spinning} \notin \left(\mathcal{P} \left(\overline{\text{FSTA}}^{\text{S}} \right) \cup \mathcal{P} \left(\overline{\text{FCOM}}^{\text{S}} \right) \right)$.

Proof. Spinning is not solvable under $\overline{\text{FSTA}}^{\text{S}}$ since an activated robot r cannot know what movements other robots have already made, thus it cannot determine the rotation-angle.

Spinning is not even solvable under model $\overline{\text{FCOM}}^{\text{S}}$. Suppose that, by contradiction, there exists an algorithm \mathbb{A} solving Spinning. In particular, the problem is solved under an activation scheduler S . Let r_0 be the robot in position a_0 . Let t_1 be the activation time, under S , of the first round during which r_0 performs a non-null movement. Let S' be another scheduling, such that

$$S'(t) := S(t) \quad \forall t < t_1 \quad \text{and} \quad S'(t_1) = S'(t_1 + 1) := \{r_0\}$$

If \mathbb{A} is executed under S' , then the execution is the same as S until time $t_1 - 1$. At time t_1 , robot r_0 behaves in the same way as it did under scheduling S but, as no other robot has been activated, then there is no way to keep track of the fact that r_0 has already moved. At time $t_1 + 1$ robot r_0 is activated again but it cannot understand from geometric conditions that it must stay still. Contradiction. \square

Theorem 8.

$$\begin{aligned} \overline{\text{OBL\O T}}^{\text{F}} &> \overline{\text{OBL\O T}}^{\text{S,A}} \\ \overline{\text{FSTA}}^{\text{F}} &> \overline{\text{FSTA}}^{\text{S,A}} \\ \overline{\text{FCOM}}^{\text{F}} &> \overline{\text{FCOM}}^{\text{S,A}} \\ \overline{\text{OBL\O T}}^{\text{F}} &\perp \overline{\text{FCOM}}^{\text{S,A}} \\ \overline{\text{OBL\O T}}^{\text{F}} &\perp \overline{\text{FSTA}}^{\text{S,A}}. \end{aligned}$$

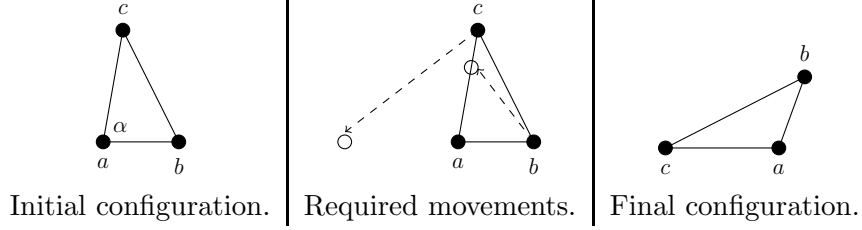


Table 5: **Angle-Shift**.

Problem 6 (Angle-Shift). Consider an initial configuration with three robots forming an acute and scalene triangle. Let a, b, c be the three robots, where a is placed on the greatest angle, say α , whereas c is placed on the smallest angle. Fixing a as the rotation center and following the direction given by a, b, c , the problem requires b to rotate of α and c to rotate of $\pi - \alpha$. The robots are not allowed to stop anywhere else on the plane. Afterwards, the robots must stay still. See Table 5.

Lemma 11. $\text{Angle-Shift} \in \left(\mathcal{P} \left(\overline{\text{OBL\O T}}^{\text{F}} \right) \setminus \mathcal{P} \left(\overline{\text{LUMI}}^{\text{S}} \right) \right)$.

Proof. **Angle-Shift** is solvable under any FULLY model: if b and c perform their cycles at the same time, then they correctly compute their target position. The final configuration is stable since it always forms an obtuse triangle (terminal condition).

Instead, the swarm can suffer from information loss in SEMI, making **Angle-Shift** unsolvable even under $\overline{\text{LUMI}}^{\text{S}}$. In fact, suppose that in the initial configuration only b is activated. After b 's movement, the three robots turn out to be collinear in the reached configuration. As a result, c has no means to recompute α , whether c uses the geometry of the configuration or uses constant-size lights. The same happens even if only c is activated. \square

Theorem 9.

$$\begin{aligned} \overline{\text{LUMI}}^{\text{F}} &> \overline{\text{LUMI}}^{\text{S,A}} \\ \overline{\text{OBL\O T}}^{\text{F}} &\perp \overline{\text{LUMI}}^{\text{S,A}} \\ \overline{\text{FSTA}}^{\text{F}} &\perp \overline{\text{LUMI}}^{\text{S,A}}. \end{aligned}$$

4.4 Opaqueness and asynchrony

We now introduce the Pseudo-Polygon problem which shows a peculiar issue occurring in case of obstructed visibility and asynchrony.

Definition 1. Given a regular n -gon \mathcal{N} , for any $n \geq 4$, a *pseudo-polygon* \mathcal{Q} is a subset of vertices of \mathcal{N} , such that $|\mathcal{Q}| \geq \frac{n}{2} + 1$. We call \mathcal{N} the *associated polygon* with respect to \mathcal{Q} .

Given a pseudo-polygon \mathcal{Q} , it is always possible to determine the associated polygon, which is unique. In fact, as \mathcal{Q} contains at least three vertices, the circumscribed circle is univocally defined. Moreover, since \mathcal{Q} contains more than half of the vertices of the associated n -gon, there always exist at least two vertices that are adjacent in \mathcal{N} . So, it is always possible to univocally establish the associated polygon from a pseudo-polygon.

Definition 2. A *safe zone* of a regular polygon is the locus of all points x in the plane such that:

- x is external to the regular polygon;
- x is not aligned with any of the two vertices of the associated polygon;
- x does not lie on the bisector of any edge of the associated polygon (equivalently, x is not equally distanced from any two adjacent vertices);
- if ℓ is the length of the edge of the polygon, then the distance between x and any vertex of the polygon is at least ℓ .

Figure 2 depicts the (complement of the) safe zone of a square.

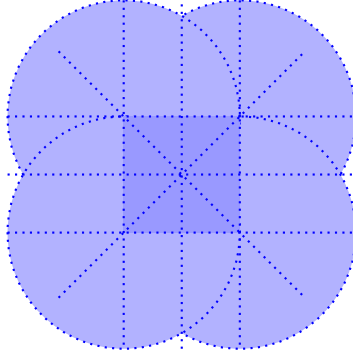


Figure 2: The safe zone of the square comprehends all the points not belonging to the blue-colored (infinite) lines and zones.

Problem 7 (Pseudo-Polygon). Let \mathcal{N} be a regular n -gon with $n \geq 6$ vertices. Let \mathcal{Q} be a pseudo-polygon of $m \geq \frac{n}{2} + 2$ vertices, associated with \mathcal{N} . Consider a swarm of $m + 1$ robots, where m robots lay on \mathcal{Q} and let the last robot, w , lay in the safe zone of \mathcal{N} . Let a be the farthest robot from w . Let b, c be the first two found robots, starting from a and following both directions on the perimeter of the associated polygon, one per each direction taken. Assume $\text{dist}(b, w) > \text{dist}(c, w)$. The problem requires a to move away from b towards a point x such that (i) x belongs to the safe zone of \mathcal{N} , (ii) x belongs to the halfplane delimited by the line \bar{bc} that does not contain a , and (iii) x must not be on any line passing by the position of w and any other robot on \mathcal{Q} . Note that requests (i,iii) are imposed in order to have x visible by every robot. See Figure 3.

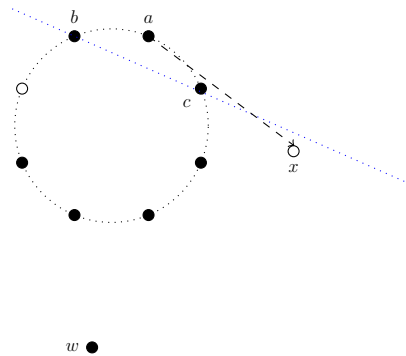


Figure 3: The Pseudo-Polygon problem associated with an octagon.

Lemma 12. Pseudo-Polygon $\notin \mathcal{P}(\overline{\mathcal{FSTA}}^A)$.

Proof. Pseudo-Polygon cannot be solved in the ASYNCH mode, only using internal lights. Let us consider the problem instance given by Figure 3 where the pseudo-polygon of the initial configuration is composed of $\frac{n}{2} + 3$ vertices, with $n = 8$. Let us assume b is activated for the first time during the movement of a , when a is hidden by c (i.e. b, c, a are collinear). When b looks at its snapshot, it recognizes a feasible initial configuration (it sees a pseudo-polygon with $\frac{n}{2} + 2$ robots, and the robot w). According to this configuration, b erroneously elects itself as the robot that has to move away from the pseudo-polygon. It has no means to understand if a exists or not. On the other hand, a has no means to know if b has updated its internal light to memorize it is not the elected robot to move. \square

False election. The impossibility of solving Pseudo-Polygon in the asynchronous modes with just internal lights derives from a critical issue that is typical of swarms with obstructed visibility. This critical issue can be described as the *false election* phenomenon. Such phenomenon can be informally described as follows: from a stable configuration, the given problem requires the use of a leader election routine to elect the unique robot (the *true leader*) which has to execute a non-null movement to reach the next configuration. All the other robots have to stay still. In the ASYNCH mode, a robot r executes its

look step while the true leader is moving and is hidden from r . However, r cannot deduct the presence of the true leader from its snapshot. So, applying the same leader election routine, r elects itself as the (*false*) leader, thus starting an unrequested movement.

The false election phenomenon must be examined when trying to transpose a **SEMI** algorithm in the **ASYNCH** mode. In particular, the use of lights must be considered as a possible method to avoid false elections. As we have shown in Lemma 12 for **Pseudo-Polygon**, internal lights are not sufficient to cope with them. Instead, the next lemma proves that external lights are required (and sufficient) to correctly solve the **Pseudo-Polygon** problem in the **ASYNCH** mode.

Lemma 13. $\text{Pseudo-Polygon} \in \left(\mathcal{P} \left(\overline{\text{OBLOT}}^{\text{S}} \right) \cap \mathcal{P} \left(\overline{\text{FCOM}}^{\text{A}} \right) \right)$.

Proof. **Pseudo-Polygon** is solvable in $\overline{\text{OBLOT}}^{\text{S}}$ (i.e. in any synchronous model), since complete visibility is guaranteed at any activation time and all the movements (null and non-null) are univocally determined by geometric conditions. In fact, each robot can determine \mathcal{Q} , the watcher w , and the robot a (the farthest from w). The robot a can compute its final destination and move there. If a robot is not the farthest from the watcher, or if it sees two robots that are not part of the pseudo-polygon, then it stands still.

Pseudo-Polygon needs at least external lights to be solvable in the **ASYNCH** mode. We show here an algorithm that needs 4 colors: **off** (default), **on**, **a**, **b**. In the first epoch, every robot updates its color according to its role: robot a turns into **a**, robot b turns into **b**, whereas the remainder turns into **on**. Afterward, let r be an activated robot that sees no **off** robots and that notes there is only one robot (the watcher) out of the pseudo-polygon. Let V_r be the set of colors r can see.

- if $V_r = \{\mathbf{a}, \mathbf{b}, \mathbf{on}\}$, r turns into **on** and stays still;
- if $V_r = \{\mathbf{a}, \mathbf{on}\}$, r turns into **b** and stays still;
- if $V_r = \{\mathbf{b}, \mathbf{on}\}$, and if r is the farthest robot from w , it turns into **a** and starts moving;
- if $V_r = \{\mathbf{on}\}$, it means r is **b** and stays still (robot a is hidden).

If a robot r sees two robots not belonging to the pseudo-polygon, then r does not move (the final configuration is already formed or is about to be formed). □

Theorem 10.

$$\begin{aligned} \overline{\text{OBLOT}}^{\text{S}} &> \overline{\text{OBLOT}}^{\text{A}} \\ \overline{\text{FSTA}}^{\text{S}} &> \overline{\text{FSTA}}^{\text{A}} \\ \overline{\text{FSTA}}^{\text{A}} &\perp \overline{\text{OBLOT}}^{\text{S}}. \end{aligned}$$

5 Relation map

Table 6 summarizes the results proved in this work, showing the relations ($>$, $<$, \perp , and \equiv) that hold between the pairs of models in our opaque framework. The map shows also which of the six witness problems (**TRT** for **Triangle Round-Trip**, **FFF** for **Flip-Flop-Flip**, **NWC** for **Newcomer Introducing**, **SPIN** for **Spinning**, **ASH** for **Angle-Shift**, **PSE** for **Pseudo-Polygon**) have been used to prove such relations. For some pairs of models (gray cells), the knowledge about what kind of relation holds is still now incomplete. E.g. between $\overline{\text{FSTA}}^{\text{F}}$ and $\overline{\text{FCOM}}^{\text{F}}$ two possible relations ($<$ or \perp) can exist: so far we have built **Newcomer Introducing** as witness problem proving that $\text{Newcomer Introducing} \in \left(\mathcal{P} \left(\overline{\text{FCOM}}^{\text{F}} \right) \setminus \mathcal{P} \left(\overline{\text{FSTA}}^{\text{F}} \right) \right)$. To prove the orthogonality relation, we should find a witness problem B such that $B \in \left(\mathcal{P} \left(\overline{\text{FSTA}}^{\text{F}} \right) \setminus \mathcal{P} \left(\overline{\text{FCOM}}^{\text{F}} \right) \right)$. Instead, to prove the strict dominance relation, we should find that any problem in $\overline{\text{FSTA}}^{\text{F}}$ can be solved also under $\overline{\text{FCOM}}^{\text{F}}$.

For the pairs of models where the relation is unknown in the opaque framework, we have reported the relation holding in the transparent framework in red.

Table 6: Relation map.

\uparrow	\overline{LUMI}^f	\overline{FCOM}^f	\overline{FSTA}^f	\overline{OBLOT}^f	\overline{LUMI}^s	\overline{FCOM}^s	\overline{FSTA}^s	\overline{OBLOT}^s	\overline{LUMI}^A	\overline{FCOM}^A	\overline{FSTA}^A
\overline{OBLOT}^A	< TRT	< TRT	< TRT	< SPIN	< TRT	< TRT	< TRT	< PSE	< TRT	< TRT	< TRT
\overline{FSTA}^A	< NWC	< or \perp , < NWC,	< SPIN	\perp TRT, SPIN	< NWC	\perp NWC, FFF	< PSE	\perp PSE, TRT	< NWC	\perp NWC, FFF	
\overline{FCOM}^A	< FFF	< FFF	\perp FFF, NWC	\perp NWC, SPIN	< FFF	< or \equiv , < FFF, NWC	\perp FFF, NWC	> or \perp , \perp NWC,	< FFF		
\overline{LUMI}^A	< ASH	< or \perp , < ASH,	\perp ASH, NWC	\perp ASH, TRT	< or \equiv , \equiv	> or \perp , > FFF,	> or \perp , > NWC,	> or \perp , > TRT,			
\overline{OBLOT}^s	< TRT	< TRT	< TRT	< SPIN	< TRT	< TRT	< TRT				
\overline{FSTA}^s	< NWC	< or \perp , < NWC,	< SPIN	\perp TRT, SPIN	< NWC	\perp NWC, FFF					
\overline{FCOM}^s	< FFF	< FFF	\perp FFF, NWC	\perp SPIN, NWC	< FFF						
\overline{LUMI}^s	< ASH	< or \perp , < ASH,	\perp ASH, NWC	\perp ASH, TRT							
\overline{OBLOT}^f	< TRT	< TRT	< TRT								
\overline{FSTA}^f	< NWC	< or \perp , < NWC,									
\overline{FCOM}^f	< or \equiv , \equiv										

6 Conclusions

We have investigated the computational power of the 12 models of collision-intolerant opaque robots, thus presenting the taxonomy of the problems solved in such framework. We have taken inspiration from [2, 6, 18, 19] where the authors provide the complete map of the relations held by the same 12 models but considering collision-tolerant transparent robots.

Thus far, the relations proved here in our opaque framework are the same as in the corresponding transparent framework. The natural question that arises from this observation is whether the relation map of the opaque models is completely identical to the relation map of the transparent models. To answer this question, future works should find the missing relations among the twelve opaque models in order to obtain the complete hierarchy in the opaque framework. Among the others, it is worth mentioning the yet unknown relation between \overline{LUMI}^s and \overline{LUMI}^A . In the transparent framework, the two models have proved to be computationally equivalent [6] through the design of a simulator which, with the help of extra light colors, simulates any SEMI algorithm in the ASYNCH mode. This simulator is not adequate to prove the same relation considering opaque robots, precisely because of their obstructed visibility. With the Pseudo-Polygon problem, we have presented the *false election* phenomenon whose formalization and investigation will be preparatory to answer this interesting open question: is it possible to simulate a \overline{LUMI}^s algorithm in the ASYNCH mode, thus proving that \overline{LUMI}^s and \overline{LUMI}^A are two equivalent models also in the opaque framework? Are constant-size lights sufficient to always avoid the phenomenon of false elections? In addition, it would be necessary to formalize and study all the critical issues caused by obstructed visibility: such formalizations may be essential for the correct investigation of the missing relations.

References

- [1] Kaustav Bose, Manash Kumar Kundu, Ranendu Adhikary, and Buddhadeb Sau. Arbitrary pattern formation by asynchronous opaque robots with lights. *Theor. Comput. Sci.*, 849:138–158, 2021.
- [2] Kevin Buchin, Paola Flocchini, Irina Kostitsyna, Tom Peters, Nicola Santoro, and Koichi Wada. Autonomous mobile robots: Refining the computational landscape. In *35th International Parallel and Distributed Processing Symposium Workshops, IPDPS*, pages 576–585. IEEE, 2021.
- [3] Kevin Buchin, Paola Flocchini, Irina Kostitsyna, Tom Peters, Nicola Santoro, and Koichi Wada. On the computational power of energy-constrained mobile robots: Algorithms and cross-model analy-

- sis. In *29th International Colloquium on Structural Information and Communication Complexity, SIROCCO*, volume 13298 of *Lecture Notes in Computer Science*, pages 42–61. Springer, 2022.
- [4] Davide Canepa and Maria Gradinariu Potop-Butucaru. Stabilizing flocking via leader election in robot networks. In Toshimitsu Masuzawa and Sébastien Tixeuil, editors, *9th International Symposium on Stabilization, Safety, and Security of Distributed Systems, SSS*, volume 4838 of *Lecture Notes in Computer Science*, pages 52–66. Springer, 2007.
- [5] Gianlorenzo D’Angelo, Gabriele Di Stefano, Ralf Klasing, and Alfredo Navarra. Gathering of robots on anonymous grids and trees without multiplicity detection. *Theor. Comput. Sci.*, 610:158–168, 2016.
- [6] Shantanu Das, Paola Flocchini, Giuseppe Prencipe, Nicola Santoro, and Masafumi Yamashita. Autonomous mobile robots with lights. *Theor. Comput. Sci.*, 609:171–184, 2016.
- [7] Shantanu Das, Riccardo Focardi, Flaminia L. Luccio, Euripides Markou, and Marco Squarcina. Gathering of robots in a ring with mobile faults. *Theor. Comput. Sci.*, 764:42–60, 2019.
- [8] Mattia D’Emidio, Daniele Frigioni, and Alfredo Navarra. Synchronous robots vs asynchronous lights-enhanced robots on graphs. In *16th Italian Conference on Theoretical Computer Science, ICTCS*, pages 169–180. Elsevier, 2015.
- [9] Mattia D’Emidio, Daniele Frigioni, and Alfredo Navarra. Characterizing the computational power of anonymous mobile robots. In *36th International Conference on Distributed Computing Systems, ICDCS*, pages 293–302. IEEE Computer Society, 2016.
- [10] Mattia D’Emidio, Gabriele Di Stefano, Daniele Frigioni, and Alfredo Navarra. Characterizing the computational power of mobile robots on graphs and implications for the euclidean plane. *Inf. Comput.*, 263:57–74, 2018.
- [11] Stefan Dobrev, Paola Flocchini, Giuseppe Prencipe, and Nicola Santoro. Multiple agents rendezvous in a ring in spite of a black hole. In *7th International Conference on Principles of Distributed Systems, OPODIS*, volume 3144 of *Lecture Notes in Computer Science*, pages 34–46. Springer, 2003.
- [12] Caterina Feletti, Carlo Mereghetti, and Beatrice Palano. Uniform circle formation for swarms of opaque robots with lights. In *20th International Symposium on Stabilization, Safety, and Security of Distributed Systems, SSS*, volume 11201 of *Lecture Notes in Computer Science*, pages 317–332. Springer, 2018.
- [13] Caterina Feletti, Carlo Mereghetti, and Beatrice Palano. $O(\log n)$ -time uniform circle formation for asynchronous opaque luminous robots. In *27th International Conference on Principles of Distributed Systems, OPODIS*, volume 286 of *LIPICs*, pages 5:1–5:21, 2023.
- [14] Paola Flocchini. Gathering. In Paola Flocchini, Giuseppe Prencipe, and Nicola Santoro, editors, *Distributed Computing by Mobile Entities, Current Research in Moving and Computing*, volume 11340 of *Lecture Notes in Computer Science*, pages 63–82. Springer, 2019.
- [15] Paola Flocchini, Giuseppe Prencipe, and Nicola Santoro. *Distributed Computing by Oblivious Mobile Robots*. Synthesis Lectures on Distributed Computing Theory. Morgan & Claypool Publishers, 2012.
- [16] Paola Flocchini, Giuseppe Prencipe, and Nicola Santoro, editors. *Distributed Computing by Mobile Entities, Current Research in Moving and Computing*, volume 11340 of *Lecture Notes in Computer Science*. Springer, 2019.
- [17] Paola Flocchini, Giuseppe Prencipe, Nicola Santoro, and Giovanni Viglietta. Distributed computing by mobile robots: uniform circle formation. *Distributed Comput.*, 30(6):413–457, 2017.
- [18] Paola Flocchini, Nicola Santoro, Yuichi Sudo, and Koichi Wada. On asynchrony, memory, and communication: Separations and landscapes. In *27th International Conference on Principles of Distributed Systems, OPODIS*, volume 286 of *LIPICs*, pages 28:1–28:23, 2023.
- [19] Paola Flocchini, Nicola Santoro, and Koichi Wada. On memory, communication, and synchronous schedulers when moving and computing. In *23rd International Conference on Principles of Distributed Systems, OPODIS*, volume 153 of *LIPICs*, pages 25:1–25:17, 2019.
- [20] Taisuke Izumi, Daichi Kaino, Maria Gradinariu Potop-Butucaru, and Sébastien Tixeuil. On time complexity for connectivity-preserving scattering of mobile robots. *Theor. Comput. Sci.*, 738:42–52, 2018.

- [21] Sayaka Kamei, Anissa Lamani, Fukuhito Ooshita, Sébastien Tixeuil, and Koichi Wada. Gathering on rings for myopic asynchronous robots with lights. In *23rd International Conference on Principles of Distributed Systems, OPODIS*, volume 153 of *LIPICs*, pages 27:1–27:17, 2019.
- [22] Peter Kling and Friedhelm Meyer auf der Heide. Continuous protocols for swarm robotics. In Paola Flocchini, Giuseppe Prencipe, and Nicola Santoro, editors, *Distributed Computing by Mobile Entities, Current Research in Moving and Computing*, volume 11340 of *Lecture Notes in Computer Science*, pages 317–334. Springer, 2019.
- [23] Giuseppe Antonio Di Luna. Mobile agents on dynamic graphs. In Paola Flocchini, Giuseppe Prencipe, and Nicola Santoro, editors, *Distributed Computing by Mobile Entities, Current Research in Moving and Computing*, volume 11340 of *Lecture Notes in Computer Science*, pages 549–584. Springer, 2019.
- [24] Moumita Mondal and Sruti Gan Chaudhuri. Uniform scattering of robots on alternate nodes of a grid. In *23rd International Conference on Distributed Computing and Networking*, pages 254–259. ACM, 2022.
- [25] Giuseppe Prencipe. Pattern formation. In Paola Flocchini, Giuseppe Prencipe, and Nicola Santoro, editors, *Distributed Computing by Mobile Entities, Current Research in Moving and Computing*, volume 11340 of *Lecture Notes in Computer Science*, pages 37–62. Springer, 2019.
- [26] Gokarna Sharma, Ramachandran Vaidyanathan, and Jerry L. Trahan. Constant-time complete visibility for robots with lights: The asynchronous case. *Algorithms*, 14(2):56, 2021.
- [27] Gokarna Sharma, Ramachandran Vaidyanathan, Jerry L. Trahan, Costas Busch, and Suresh Rai. Complete visibility for robots with lights in $O(1)$ time. In *18th International Symposium on Stabilization, Safety, and Security of Distributed Systems, SSS*, pages 327–345, 2016.
- [28] Kazuo Sugihara and Ichiro Suzuki. Distributed algorithms for formation of geometric patterns with many mobile robots. *J. Field Robotics*, 13(3):127–139, 1996.
- [29] Ichiro Suzuki and Masafumi Yamashita. Distributed anonymous mobile robots: Formation of geometric patterns. *SIAM J. Comput.*, 28(4):1347–1363, 1999.
- [30] Masafumi Yamashita and Ichiro Suzuki. Characterizing geometric patterns formable by oblivious anonymous mobile robots. *Theor. Comput. Sci.*, 411(26-28):2433–2453, 2010.

A Proofs of theorems

The proofs of the following theorems hold combining the previously stated lemmas and by transitivity. We use the compacted notation $\{\overline{X}_1, \dots, \overline{X}_m\}^{Y_1, \dots, Y_n}$ to indicate all the models in $\{\overline{X}_i^{Y_j}\}_{\substack{1 \leq i \leq m \\ 1 \leq j \leq h}}$ where $X_i \in \{\overline{OBLOT}, \overline{FSTA}, \overline{FCOM}, \overline{LUMI}\}$ and $Y_j \in \{\mathbf{F}, \mathbf{S}, \mathbf{A}\}$.

Theorem 4. Given the schedulers $Y_1 = \mathbf{F}$, $Y_2 = \mathbf{S}$, $Y_3 = \mathbf{A}$, it holds

$$\begin{aligned} \overline{FSTA}^{Y_i} &> \overline{OBLOT}^{\{Y_j\}_{j \geq i}} \\ \overline{FCOM}^{Y_i} &> \overline{OBLOT}^{\{Y_j\}_{j \geq i}} \\ \overline{LUMI}^{Y_i} &> \overline{OBLOT}^{\{Y_j\}_{j \geq i}}. \end{aligned}$$

Proof. **Triangle Round-Trip** cannot be solved under $\overline{OBLOT}^{\mathbf{F}, \mathbf{S}, \mathbf{A}}$ (by Lemma 3) but it can be solved under $\{\overline{FSTA}, \overline{FCOM}, \overline{LUMI}\}^{\mathbf{A}, \mathbf{S}, \mathbf{F}}$ (by Lemma 4). Combining the results, we obtain that \overline{OBLOT} is strictly dominated by \overline{FSTA} and \overline{FCOM} for a given synchronization mode $Y_i \in \{\mathbf{F}, \mathbf{S}, \mathbf{A}\}$. The other strict dominances are derived by transitivity. \square

Theorem 5.

$$\begin{aligned} \overline{LUMI}^{\mathbf{A}} &> \overline{FCOM}^{\mathbf{A}} \\ \overline{LUMI}^{\mathbf{S}} &> \overline{FCOM}^{\mathbf{S}, \mathbf{A}} \\ \overline{LUMI}^{\mathbf{F}} &> \overline{FCOM}^{\mathbf{S}, \mathbf{A}} \\ \overline{FCOM}^{\mathbf{F}} &> \overline{FCOM}^{\mathbf{S}, \mathbf{A}}. \end{aligned}$$

Proof. Flip-Flop-Flip is solved under \overline{FCOM}^F and $\overline{LUMI}^{A,S,F}$ (by Lemma 5) but it cannot be solved under $\overline{FCOM}^{S,A}$ (by Lemma 6). Combining the results, the strict dominance relations follow. \square

Theorem 6. Given the schedulers $Y_1 = F$, $Y_2 = S$, $Y_3 = A$, it holds

$$\overline{LUMI}^{Y_i} > \overline{FSTA}^{\{Y_j\}_{j \geq i}}.$$

Proof. By Lemma 8, **Newcomer Introducing** is solved under $\overline{LUMI}^{A,S,F}$. By Lemma 7, **Newcomer Introducing** cannot be solved under $\overline{FSTA}^{F,S,A}$. Combining the results, the strict dominance relations follow. \square

Theorem 7.

$$\overline{FSTA}^{F,S,A} \perp \overline{FCOM}^{S,A}.$$

Proof. By Lemma 5 and Lemma 6, Flip-Flop-Flip is solved in $\overline{FSTA}^{F,S,A}$ but not in $\overline{FCOM}^{S,A}$. By Lemma 8 and Lemma 7, **Newcomer Introducing** is solved in $\overline{FCOM}^{S,A}$ but not in $\overline{FSTA}^{F,S,A}$. Combining the results, the orthogonality relations follow. \square

Theorem 8.

$$\begin{aligned} \overline{OBLOT}^F &> \overline{OBLOT}^{S,A} \\ \overline{FSTA}^F &> \overline{FSTA}^{S,A} \\ \overline{FCOM}^F &> \overline{FCOM}^{S,A} \\ \overline{OBLOT}^F &\perp \overline{FCOM}^{S,A} \\ \overline{OBLOT}^F &\perp \overline{FSTA}^{S,A}. \end{aligned}$$

Proof. The above relations hold combining the previous lemmas and by transitivity:

- the strict dominance of \overline{X}^F over $\overline{X}^{S,A}$ derives from Lemma 9 and Lemma 10, for each $X \in \{\overline{OBLOT}, \overline{FSTA}, \overline{FCOM}\}$. In fact, **Spinning** is solved in $\{\overline{OBLOT}, \overline{FSTA}, \overline{FCOM}\}^F$ but it is not solved in $\{\overline{OBLOT}, \overline{FSTA}, \overline{FCOM}\}^{S,A}$;
- the orthogonality between \overline{OBLOT}^F over $\overline{FCOM}^{S,A}$ holds since **Spinning** is solved in \overline{OBLOT}^F but not in $\overline{FCOM}^{S,A}$, and since **Newcomer Introducing** is solved in $\overline{FCOM}^{S,A}$ but not in \overline{OBLOT}^F (by Lemma 8, Lemma 7);
- the orthogonality between \overline{OBLOT}^F over $\overline{FSTA}^{S,A}$ holds since **Spinning** is solved in \overline{OBLOT}^F but not in $\overline{FSTA}^{S,A}$, and since **Triangle Round-Trip** is solved in $\overline{FSTA}^{S,A}$ but not in \overline{OBLOT}^F (by Lemma 4, Lemma 3).

\square

Theorem 9.

$$\begin{aligned} \overline{LUMI}^F &> \overline{LUMI}^{S,A} \\ \overline{OBLOT}^F &\perp \overline{LUMI}^{S,A} \\ \overline{FSTA}^F &\perp \overline{LUMI}^{S,A}. \end{aligned}$$

Proof. The above relations hold combining the previous lemmas and by transitivity:

- the strict dominance of \overline{LUMI}^F over $\overline{LUMI}^{S,A}$ straightforwardly derives from Lemma 11. In fact, **Angle-Shift** is solved in \overline{LUMI}^F but it is not solved in $\overline{LUMI}^{S,A}$;
- the orthogonality between \overline{OBLOT}^F over $\overline{LUMI}^{S,A}$ holds since **Angle-Shift** is solved in \overline{OBLOT}^F but not in $\overline{LUMI}^{S,A}$, and since **Triangle Round-Trip** is solved in $\overline{LUMI}^{S,A}$ but not in \overline{OBLOT}^F (by Lemma 4, Lemma 3);
- the orthogonality between \overline{FSTA}^F over $\overline{LUMI}^{S,A}$ holds since **Angle-Shift** is solved in \overline{FSTA}^F but not in $\overline{LUMI}^{S,A}$, and since **Newcomer Introducing** is solved in $\overline{LUMI}^{S,A}$ but not in \overline{FSTA}^F (by Lemma 8, Lemma 7).

□

Theorem 10.

$$\begin{aligned}\overline{OBLOT}^S &> \overline{OBLOT}^A \\ \overline{FSTA}^S &> \overline{FSTA}^A \\ \overline{FSTA}^A &\perp \overline{OBLOT}^S.\end{aligned}$$

Proof. The above relations hold combining the previous lemmas and by transitivity:

- for each $X \in \{OBLOT, FSTA\}$, \overline{X}^S strictly dominates \overline{X}^A since **Pseudo-Polygon** can be solved in \overline{X}^S but not in \overline{X}^A (by Lemma 13 and Lemma 12);
- the orthogonality between \overline{FSTA}^A and \overline{OBLOT}^S holds since **Pseudo-Polygon** is solved in \overline{OBLOT}^S but not in \overline{FSTA}^A , and since **Triangle Round-Trip** is solved in \overline{FSTA}^A but not in \overline{OBLOT}^S (by Lemma 4 and Lemma 3).

□