On Memory, Communication, and Synchronous Schedulers When Moving and Computing

Paola Flocchini
EECS, University of Ottawa, Canada
pflocchi@uottawa.ca

Nicola Santoro
School of Computer Science, Carleton University, Canada
santoro@scs.carleton.ca

Koichi Wada
Faculty of Science and Engineering, Hosei University, Japan
wada@hosei.ac.jp

Abstract

We investigate the computational power of distributed systems whose autonomous computational entities, called robots, move and operate in the 2-dimensional Euclidean plane in synchronous *Look-Compute-Move* (*LCM*) cycles. Specifically, we focus on the power of persistent memory and that of explicit communication, and on their computational relationship.

In the most common model, $\mathcal{OBLOT}$, the robots are oblivious (no persistent memory) and silent (no explicit means of communication). In contrast, in the $\mathcal{LUMI}$ model, each robot is equipped with a constant-sized persistent memory (called *light*), visible to all the robots; hence, these luminous robots are capable in each cycle of both remembering and communicating. Since luminous robots are computationally more powerful than the standard oblivious one, immediate important questions are about the individual computational power of persistent memory and of explicit communication. In particular, which of the two capabilities, memory or communication, is more important? in other words, is it better to remember or to communicate ?

In this paper we address these questions, focusing on two sub-models of $\mathcal{LUMI}$: $\mathcal{FSTA}$, where the robots have a constant-size persistent memory but are silent; and $\mathcal{FCOM}$, where the robots can communicate a constant number of bits but are oblivious. We analyze the relationship among all these models and provide a complete exhaustive map of their computational relationship. Among other things, we prove that communication is more powerful than persistent memory under the fully synchronous scheduler FSYNCH, while they are incomparable under the semi-synchronous scheduler SSYNCH.

2012 ACM Subject Classification Theory of computation → Distributed algorithms

Keywords and phrases Look-Compute-Move, Oblivious mobile robots, Robots with lights, Memory versus Communication, Moving and Computing

Digital Object Identifier 10.4230/LIPIcs.OPODIS.2019.25

Acknowledgements This research was partly supported by NSERC through the Discovery Grant program, by Prof. Flocchini's University Research Chair, by JSPS KAKENHI No. 17K00019, and by Japan Science and Technology Agency (JST) SICORP Grant#JPMJSC1806.

1 INTRODUCTION

1.1 Background and Motivation

The computational issues of autonomous mobile entities operating in an Euclidean space in *Look-Compute-Move* (*LCM*) cycles have been the object of much research in distributed computing. In the *Look* phase, an entity, viewed as a point and usually called *robot*, obtains a snapshot of the space; in the *Compute* phase it executes its algorithm (the same for all

23rd International Conference on Principles of Distributed Systems (OPODIS 2019).
Editors: Pascal Felber, Roy Friedman, Seth Gilbert, and Avery Miller; Article No. 25; pp. 25:1–25:17
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

robots) using the snapshot as input; it then moves towards the computed destination in the *Move* phase. Repeating these cycles, the robots are able to collectively perform some tasks and solve some problems. The research interest has been on determining the impact that *internal* capabilities (e.g., memory, communication) and *external* conditions (e.g. synchrony, activation scheduler) have on the solvability of a problem.

In the most common model, $\mathcal{OBLOT}$, in addition to the standard assumptions of *anonymity* and *uniformity* (robots have no IDs and run identical algorithms), the robots are *oblivious* (no persistent memory to record information of previous cycles) and *silent* (without explicit means of communication). Computability in this model has been the object of intensive research since its introduction in [27]. Extensive investigations have been carried out to clarify the computational limitations and powers of these robots for basic coordination tasks such as Gathering (e.g., [1, 2, 4, 6, 7, 8, 15, 21, 27]), Pattern Formation (e.g., [16, 18, 27, 30, 31]), Flocking (e.g., [5, 19, 26]); for a recent account of the state of the art on some of these problems, see [13] and the chapters therein. Clearly, the restrictions created by the absence of persistent memory and the incapacity of explicit communication severely limits what the robots can do and renders complex and difficult for them to perform the tasks they can do.

A model where robots are provided with some (albeit limited) persistent memory and communication means is the $\mathcal{LUMI}$ model, formally defined and analyzed in [9, 10], following a suggestion in [24]. In this model, each robot is equipped with a constant-sized memory (called *light*), whose value (called *color*) can be set during the *Compute* phase. The light is visible to all the robots and is persistent in the sense that it is not automatically reset at the end of a cycle. Hence, these luminous robots are capable in each cycle of both remembering and communicating a constant number of bits. There is a lot of research work on the design of algorithms and the feasibility of problems for luminous robots (e.g., [3, 10, 11, 17, 20, 22, 23, 25, 28, 29]); for a recent survey, see [12].

As for the computational relationship between $\mathcal{OBLOT}$ and $\mathcal{LUMI}$, the availability of both persistent memory and communication, however limited, clearly renders luminous robots more powerful than oblivious robots (e.g., [10]). This immediately raises important questions about the individual computational power of the two internal capabilities: memory and communication. In particular,

- if the robots were endowed with a constant number of bits of persistent memory but were still unable to communicate explicitly, what problems could they solve ?
- If the robots could communicate a constant number of bits in each cycle, but were oblivious, what would be their computational power then ?
- Which of the two capabilities, memory or communication, is more important? or, in other words, *is it better to remember or to communicate ?*

Helpful in this regards are two sub-models of $\mathcal{LUMI}$. In the first model, $\mathcal{FSTA}$, the light of a robot is visible only by that robot, while in the second model, $\mathcal{FCOM}$, the light of a robot is visible only to the other robots. Thus in $\mathcal{FSTA}$ the color merely encodes an internal state; hence the robots are *finite-state* and *silent*. On the contrary, in $\mathcal{FCOM}$, a robot can communicate to the other robots through its colored light but forgets the content of its transmission by the next cycle; that is, robots are *finite-communication* and *oblivious*.

This means that some answers to the above questions, as well as others, can be provided by exploring and determining the computational power within these four models, $\mathcal{OBLOT}$, $\mathcal{FSTA}$, $\mathcal{FCOM}$, and $\mathcal{LUMI}$ and with respect to each other. This is the focus of this paper.

When studying computability within a model of *LCM* robots, two interrelated external factors play a crucial role: *time* and *activation schedule*. With respect to these factors, there are two fundamentally different settings: *asynchronous* and *synchronous*.

In the *asynchronous* setting (Asynch), first studied in [14], there is no common notion of time, each robot is activated independently of the others, the duration of each phase is finite but unpredictable and might be different in different cycles.

In the *synchronous* setting (Ssynch), also called semi-synchronous and first studied in [27], time is divided into discrete intervals, called *rounds*; in each round some (possibly all) robots are activated, perform their LCM cycle simultaneously, and terminate by the end of the round. The selection of which robots are activated at a round is made by the adversarial scheduler, constrained to be fair. A special synchronous setting which plays an important role is the *fully-synchronous* setting (Fsynch) where every robot is activated in every round; that is, the activation scheduler has no adversarial power.

Returning to the focus of this paper, which is to understand the computational power within each model, the amount of available knowledge is rather limited. In particular, it is known that, within $\mathcal{OBLOT}$, robots in Fsynch are strictly more powerful than those in Ssynch: there are problems solvable in Fsynch but unsolvable in Ssynch [27]. It is also known that, within $\mathcal{LUMI}$, robots have in Asynch the same computational power as in Ssynch [10]. As for the relationship between different models, it has been shown that asynchronous luminous robots are strictly more powerful than oblivious synchronous robots [10]. The $\mathcal{FCOM}$ and $\mathcal{FSTA}$ models have been studied only in the context of *Rendezvous*, which cannot be solved in Ssynch in the $\mathcal{OBLOT}$ model, while it has been shown to be solvable in both $\mathcal{FCOM}$ and $\mathcal{FSTA}$ [17]. In this paper we investigate these questions, focusing on synchronous schedulers.

1.2 Contributions

We analyze the relationship among all these models and provide a complete exhaustive map of their computational relationship, summarized in Tables 1-3, where: $\mathcal{X}^Y$ denotes model $\mathcal{X}$ under scheduler Y; F and S stand for Fsynch and Ssynch respectively, $A > B$ indicates that model A is computationally more powerful than model B, $A \equiv B$ denotes that they are computationally equivalent, $A \perp B$ denotes that they are computationally incomparable.

We first examine the computational relationship within each scheduler. Among other things, we prove that the answer to the question *"is it better to remember or to communicate ?"* depends on the type of scheduler. More precisely, communication is more powerful than persistent memory if the scheduler is fully synchronous; on the other hand, the two models are incomparable under the semi-synchronous scheduler.

We then focus on the relationship between Fsynch and Ssynch. In addition to the expected dominance results, we prove some interesting orthogonality results. In fact, we show that, on one hand, both $\mathcal{FSTA}^S$ and $\mathcal{FCOM}^S$ are incomparable with $\mathcal{OBLOT}^F$, on the other $\mathcal{LUMI}^S$ is incomparable with $\mathcal{FSTA}^F$, $\mathcal{FCOM}^F$, and even with $\mathcal{OBLOT}^F$. We also close an open problem of [10].

Table 1 Relationships within Fsynch.

	$\mathcal{FCOM}^F$	$\mathcal{FSTA}^F$	$\mathcal{OBLOT}^F$
$\mathcal{LUMI}^F$	$\equiv$ (Th.2)	$>$ (Th.2,6)	$>$ (Th.2,6,10)
$\mathcal{FCOM}^F$	–	$>$ (Th.6)	$>$ (Th.6,10)
$\mathcal{FSTA}^F$	–	–	$>$ (Th.10)

Table 2 Relationships within Ssynch.

	$\mathcal{FCOM}^S$	$\mathcal{FSTA}^S$	$\mathcal{OBLOT}^S$
$\mathcal{LUMI}^S$	$>$ (Th.17)	$>$ (Th.17)	$>$ (Th.15, 17)
$\mathcal{FCOM}^S$	–	$\perp$ (Th.14)	$>$ (Th.15)
$\mathcal{FSTA}^S$	–	–	$>$ (Th.15)

Table 3 Relationship between FSYNCH and SSYNCH.

	$\mathcal{LUMI}^S$	$\mathcal{FCOM}^S$	$\mathcal{FSTA}^S$	$\mathcal{OBLOT}^S$
$\mathcal{LUMI}^F$ $\equiv \mathcal{FCOM}^F$	$>$ (Th.20)	$>$ (Th.20)	$>$ (Th.6,20)	$>$ (Th.15,20)
$\mathcal{FSTA}^F$	$\perp$ (Th.26)	$\perp$ (Th.26)	$>$ (Th.20)	$>$ (Th.15,20)
$\mathcal{OBLOT}^F$	$\perp$ (Th.28)	$\perp$ (Th.25)	$\perp$ (Th.25)	$>$ (Th.20)

2 MODELS AND PRELIMINARIES

2.1 The Basics

The systems considered in this paper consist of a team $R = \{r_0, \cdots, r_{n-1}\}$ of computational entities moving and operating in the Euclidean plane $\mathbb{R}^2$. Viewed as points and called *robots*, the entities can move freely and continuously in the plane. Each robot has its own local coordinate system and it always perceives itself at its origin; there might not be consistency between these coordinate systems. A robot is equipped with sensorial devices that allows it to observe the positions of the other robots in its local coordinate system.

The robots are *identical*: they are indistinguishable by their appearance and they execute the same protocol. The robots are *autonomous*, without a central control.

At any point in time, a robot is either *active* or *inactive*. Upon becoming active, a robot r executes a *Look-Compute-Move* (*LCM*) cycle performing the following three operations:

1. *Look:* The robot activates its sensors to obtain a snapshot of the positions occupied by robots with respect to its own coordinate system[1].
2. *Compute:* The robot executes its algorithm using the snapshot as input. The result of the computation is a destination point.
3. *Move:* The robot moves to the computed destination[2]. If the destination is the current location, the robot stays still.

When inactive, a robot is idle. All robots are initially idle. The amount of time to complete a cycle is assumed to be finite, and the *Look* operation is assumed to be instantaneous.

Let $x_i(t)$ denote the location of robot r_i at time t in a global coordinate system (unknown to the robots), and let $X(t) = \{x_i(t) : 0 \le i \le n-1\} = \{x_0(t), x_1(t), \ldots, x_{m-1}(t)\}$; observe that $|X(t)| = m \le n$ since several robots might be at the same location at time t.

In this paper, we do not assume that the robots have a common coordinate system. If they agree on the same circular orientation of the plane (i.e., they do agree on "clockwise" direction), we say that there is *chirality*. Except when explicitly stated, we assume there is chirality.

2.2 The Models

Different models, based on the same basic premises defined above, have been considered in the literature and will be examined here.

[1] This is called the *full visibility* (or unlimited visibility) setting; restricted forms of visibility have also been considered for these systems

[2] This is called the *rigid mobility* setting; restricted forms of mobility (e.g., when the movement may be interrupted by an adversary) have also been considered for these systems

In the most common model, $\mathcal{OBLOT}$, the robots are *silent*: they have no explicit means of communication; furthermore they are *oblivious*: at the start of a cycle, a robot has no memory of observations and computations performed in previous cycles.

In the other common model, $\mathcal{LUMI}$, each robot r is equipped with a persistent visible state variable $Light[r]$, called *light*, whose values are taken from a finite set C of states called *colors* (including the color that represents the initial state when the light is off). The colors of the lights can be set in each cycle by r at the end of its *Compute* operation. A light is *persistent* from one computational cycle to the next: the color is not automatically reset at the end of a cycle; the robot is otherwise oblivious, forgetting all other information from previous cycles. In $\mathcal{LUMI}$, the *Look* operation produces a colored snapshot; i.e., it returns the set of pairs $(position, color)$ of the other robots[3]. Note that if $|C| = 1$, then the light is not used; thus, this case corresponds to the $\mathcal{OBLOT}$ model.

It is sometimes convenient to describe a robot r as having $k \geq 1$ lights, denoted $r.light_1, \ldots, r.light_k$, where the values of $r.light_i$ are from a finite set of colors C_i, and to consider $Light[r]$ as a k-tuple of variables; clearly, this corresponds to r having a single light that uses $\Pi_{i=1}^{k}|C_i|$ colors.

The lights provide simultaneously persistent memory and direct means of communication, although both limited to a constant number of bits per cycle. Two sub-models of $\mathcal{LUMI}$ have been defined and investigated, each offering only one of these two capabilities.

In the first model, $\mathcal{FSTA}$, a robot can only see the color of its own light; that is, the light is an *internal* one and its color merely encodes an internal state. Hence the robots are *silent*, as in $\mathcal{OBLOT}$; but are *finite-state*. Observe that a snapshot in $\mathcal{FSTA}$ is the same as in $\mathcal{OBLOT}$.

In the second model, $\mathcal{FCOM}$, the lights are *external*: a robot can communicate to the other robots through its colored light but forgets the color of its own light by the next cycle; that is, robots are *finite-communication* but *oblivious*. A snapshot in $\mathcal{FCOM}$ is like in $\mathcal{LUMI}$ except that, for the position x where the robot r performing the *Look* is located, $Light[r]$ is omitted from the set of colors present at x.

In all the above models, a *configuration* $C(t)$ at time t is the multi-set of the n pairs of the $(x_i(t), c_i(t))$, where $c_i(t)$ is the color of robot r_i at time t.

2.3 The Schedulers

With respect to the activation schedule of the robots, and the duration of their *Look-Compute-Move* cycles, the fundamental distinction is between the *asynchronous* and *synchronous* settings.

In the *asynchronous* setting (Asynch), first studied in [14], there is no common notion of time, each robot is activated independently of the others, the duration of each phase is finite but unpredictable and might be different in different cycles.

In the *synchronous* setting (Ssynch), also called semi-synchronous and first studied in [27], time is divided into discrete intervals, called *rounds*; in each round some robots are activated simultaneously, and perform their *LCM* cycle in perfect synchronization.

A popular synchronous setting which plays an important role is the *fully-synchronous* setting (Fsynch) where every robot is activated in every round; that is, the activation scheduler has no adversarial power.

[3] If (strong) multiplicity detection is assumed, the snapshot is a multi-set.

In all two settings, the selection of which robots are activated at a round is made by an adversarial *scheduler*, whose only limit is that every robot must be activated infinitely often (i.e., it is fair scheduler). In the following, for all synchronous schedulers, we use round and time interchangeably.

2.4 Computational Relationships

Let $\mathcal{M} = \{\mathcal{LUMI}, \mathcal{FCOM}, \mathcal{FSTA}, \mathcal{OBLOT}\}$ be the set of models under investigation, and $\mathcal{S} = \{$Fsynch, Ssynch$\}$ be the set of activation schedulers under consideration.

We denote by $\mathcal{R}$ the set of all teams of robots satisfying the core assumptions (i.e., they are identical, autonomous, and operate in *LCM* cycles), and $R \in \mathcal{R}$ a team of robots having identical capabilities (e.g., common coordinate system, persistent storage, internal identity, rigid movements etc.). By $\mathcal{R}_n \subset \mathcal{R}$ we denote the set of all teams of size n.

Given a model $M \in \mathcal{M}$, a scheduler $S \in \mathcal{S}$, and a team of robots $R \in \mathcal{R}$, let $Task(M, S; R)$ denote the set of problems solvable by R in M under adversarial scheduler S.

Let $M_1, M_2 \in \mathcal{M}$ and $S_1, S_2 \in \mathcal{S}$. We define the following relationships between model M_1 under scheduler S_1 and model M_2 under scheduler S_2:

- *computationally not less powerful* ($M_1^{S_1} \geq M_2^{S_2}$), if $\forall R \in \mathcal{R}$ we have $Task(M_1, S_1; R) \supseteq Task(M_2, S_2; R)$;
- *computationally more powerful* ($M_1^{S_1} > M_2^{S_2}$), if $M_1^{S_1} \geq M_2^{S_2}$ and $\exists R \in \mathcal{R}$ such that $Task(M_1, S_1; R) \setminus Task(M_2, S_2; R) \neq \emptyset$;
- *computationally equivalent* ($M_1^{S_1} \equiv M_2^{S_2}$), if $M_1^{S_1} \geq M_2^{S_2}$ and $M_2^{S_2} \geq M_1^{S_1}$;
- *computationally orthogonal* (or *incomparable*), ($M_1^{S_1} \bot M_2^{S_2}$), if $\exists R_1, R_2 \in \mathcal{R}$ such that $Task(M_1, S_1; R_1) \setminus Task(M_2, S_2; R_1) \neq \emptyset$ and $Task(M_2, S_2; R_2) \setminus Task(M_1, S_1; R_2) \neq \emptyset$.

For simplicity of notation, for a model $M \in \mathcal{M}$, let M^F and M^S denote M^{Fsynch} and M^{Ssync}, respectively; and let $M^F(R)$ and $M^S(R)$ denote $Task(M$,Fsynch;$R)$ and $Task(M$,Ssynch;$R)$, respectively.

Trivially, for any $M \in \mathcal{M}$, $M^F \geq M^S$; also, for any $S \in \mathcal{S}$, $\mathcal{LUMI}^S \geq \mathcal{FSTA}^S \geq \mathcal{OBLOT}^S$ and $\mathcal{LUMI}^S \geq \mathcal{FCOM}^S \geq \mathcal{OBLOT}^S$.

3 COMPUTATIONAL RELATIONSHIP IN Fsynch

In this section, we consider the fully synchronous scheduler Fsynch and we prove that, in this setting, it is better to communicate than to remember. Specifically, we prove that $\mathcal{FCOM}$ has the same power as $\mathcal{LUMI}$ and is strictly more powerful than $\mathcal{FSTA}$; furthermore, they are all strictly more powerful than $\mathcal{OBLOT}$.

3.1 $\mathcal{FCOM}^F \equiv \mathcal{LUMI}^F$

To prove that $\mathcal{FCOM}$ has the same power as $\mathcal{LUMI}$ in Fsynch, we first need to prove the following.

▶ **Lemma 1.** $\forall R \in \mathcal{R}, \mathcal{LUMI}^F(R) \subseteq \mathcal{FCOM}^F(R)$.

Proof. The proof is constructive. Our algorithm uses the following observation: if there is chirality, then there exists a unique circular ordering of the locations $X(t)$ occupied by the robots at that time [27]. Let `suc` and `pred` be the functions denoting the ordering and, without loss of generality, let $\texttt{suc}(x_i(t)) = x_{i+1 \bmod m}(t)$ and $\texttt{pred}(x_i)(t) = x_{i-1 \bmod m}(t)$ for $i \in \{0, 1, \ldots, m-1\}$. Even in absence of chirality, a circular arrangement can still be obtained, but there is no common agreement on suc and pred because the "clockwise" direction is not common to all robots and the notion of successor and predecessor is local, and possibly inconsistent among the robots. In this case, let $\texttt{neigh}(x_i(t))$ indicate the unordered pair of the two neighbouring locations of x_i: $\texttt{neigh}(x_i(t)) = \{x_{i+1 \bmod m}(t), x_{i-1 \bmod m}(t)\}$ for $i \in \{0, 1, \ldots, m-1\}$. When no ambiguity arises, we will omit the temporal indication.

We now describe an $\mathcal{FCOM}$ protocol, called LUbyFCinFSY, which, for any given $\mathcal{LUMI}$ protocol A, produces a fully-synchronous execution of A. The simulation algorithm is presented in Algorithm 1, where a robot r at location x uses three lights: $r.color$, indicating its own color, initially set to c_0, $r.\texttt{neigh}.color$, indicating the 2-element set of colors seen at $\texttt{suc}(x)$ and at $\texttt{pred}(x)$ taken from the set $2^{\mathcal{C}}$, where C is the set of colors used by algorithm A, initially set to $\{\{c_0\}, \{c_0\}\}$, and $r.step \in \{1, 2\}$, indicating the step of the algorithm, initially set to 1. It also uses variable $r.color.here$, initially set to $\{c_0\}$, indicating the set of colors visible by r at its own location. In the following, when no ambiguity arises, we will denote $\texttt{suc}(x)$ and $\texttt{pred}(x)$ by $\texttt{suc}(r)$ and $\texttt{pred}(r)$.

The algorithm simulates a single round of A with two rounds (or steps):

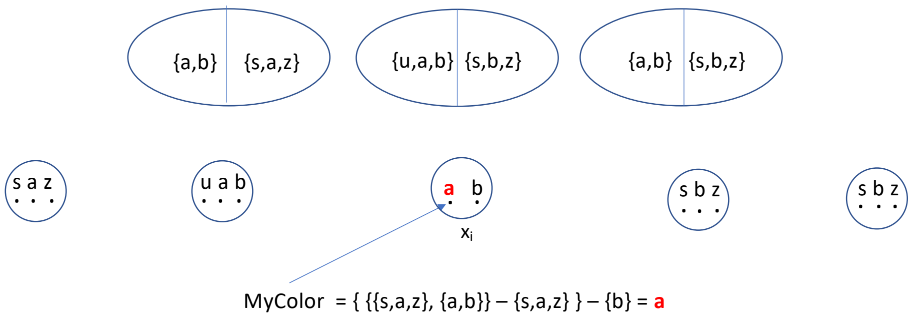

Figure 1 $\{\texttt{pred}(x).\texttt{neigh}.color - r'.color\} - r.color.here$.

1. **Copy Step:** ($r.step = 1$). In the Look phase, r determines $r.step = 1$ by observing the corresponding color of one of the neighbours (e.g., $\texttt{pred}(x).step$) and sets $r.step = 2$. It also observes the colors of the robots at its successor and predecessor and sets $r.\texttt{neigh}.color$ (notice that $r.\texttt{neigh}.color$ is the same for all robots at the same location). Robot r does not move.
2. **Execution Step:** ($r.step = 2$).
   Color Determination. After the Look phase, by looking at one of its neighbours ($\texttt{pred}(x)$) robot r discovers $r.step = 2$, as well as its own color. In fact, let $x' = \texttt{other}(\texttt{pred}(x))$ denote the other neighbour of r's predecessor, and let $r.color.here$ correspond to the set of colors seen by r at its own location x (note that, by definition, this set does not include r's color); then r's color is determined by letting *cand-set* be the element of $\texttt{pred}(x).\texttt{neigh}.color - \{x'.color\}$ and r's color be the element of *cand-set* $- r.color.here$, where "-" indicates the difference operator between sets (see Figure 1).
   Execution. Robot r executes the Compute and Move phases according to Algorithm A.

The correctness of Algorithm LUbyFCinFSY(A) follows easily from the fact that we are operating in FSYNCH and that the only difference between $\mathcal{LUMI}$ and $\mathcal{FCOM}$ is that in latter a robot does not see the color of its own light. This can however be determined as indicated in the protocol. In other words, LUbyFCinFSY(A) correctly simulate in FSYNCH algorithm A and Lemma 1 follows. ◀

Algorithm 1 `LUbyFCinFSY`(A) - for robot r at location x.

Phase Look
Observe, in particular, $\texttt{pred}(x).color$, $\texttt{suc}(x).color$, $\texttt{pred}(x).step$, $\texttt{other}(\texttt{pred}(x))$;
as well as $r.color.here$ (note that, for this, r cannot see its own color).

Phase Compute
if ($\texttt{pred}(x).step = 1$) **then** //step 1- Copy //
    $r.\texttt{neigh}.color \leftarrow \{\texttt{pred}(x).color, \texttt{suc}(x).color\}$,
    where $\texttt{pred}$(x).color=$\{\rho.color|\rho \in \texttt{pred}(x)\}$ and $\texttt{suc}$(x).color=$\{\rho.color|\rho \in \texttt{suc}(x)\}$
    $r.step \leftarrow 2$
    $r.des \leftarrow x$
else //step 2- Execution //
    $x' \leftarrow \texttt{other}(\texttt{pred}(x))$ // x' is the other neighbour of $\texttt{pred}(x)$ //
    $cand\text{-}set \leftarrow$ the element of $\texttt{pred}(x).\texttt{neigh}.color - \{x'.color\}$
    $r.color \leftarrow$ the element of $cand\text{-}set$ $-r.color.here$ // find my own color //
    Execute the `Compute` of $\mathcal{A}$ // with my color $r.color$, determining destination $r.des$ //

Phase Move
Move to $r.des$;

Since the reverse relation $\mathcal{FCOM}^F \leq \mathcal{LUMI}^F$ holds by definition, we can conclude:

▶ **Theorem 2.** $\mathcal{FCOM}^F \equiv \mathcal{LUMI}^F$.

3.2 $\mathcal{FCOM}^F > \mathcal{FSTA}^F$

We now turn our attention to the relationship between $\mathcal{FCOM}^{\mathcal{F}}$ and $\mathcal{FSTA}^F$. The following problem is used to show that $\mathcal{FCOM}^F > \mathcal{FSTA}^F$.

▶ **Definition 3.** ***Problem*** $\neg$***IL****: Three robots a, b, and c, starting from the initial configuration shown in Figure 2 (a), must form first the pattern of Figure 2 (b) and then move to form the pattern of Figure 2 (c).*

▶ **Lemma 4.** $\exists R \in \mathcal{R}_3$, $\neg IL \notin \mathcal{FSTA}^F(R)$,

Proof. In the initial pattern (a) of Figure 2, even if all the states of the robots are initially identical, each of them can uniquely distinguish its position in the pattern. Therefore, the three robots can easily form pattern (b) by having a move clockwise of 90 degrees. Assume that in pattern (b) the state of each robot is now different and indicates the full history of what the robot has done so far. Now the robots need to form pattern (c), which is asymmetric and requires b to move clockwise of 45 degrees. However, in pattern (b), even in presence of chirality, robot b cannot distinguish between the positions of a and c. This is true regardless of the information stored in the local state of robot b; so, after forming pattern (b), the robots cannot reach pattern (c). ◀

▶ **Lemma 5.** $\forall R \in \mathcal{R}_3$, $\neg IL \in \mathcal{FCOM}^S(R)$.

Proof. $\mathcal{FCOM}$ robots can easily solve ¬IL as follows: To form (b) from (a), robot a, which can easily distinguish its position, moves of 90 degrees clockwise and turns its light to `red`. To move from (b) to (c) robot b distinguishes a from c because of the external light and moves of 45 degrees clockwise to occupy the correct position. ◀

By Theorem 2 and Lemmas 4 and 5, we can conclude:

▶ **Theorem 6.** $\mathcal{FCOM}^F > \mathcal{FSTA}^F$.

3.3 $\mathcal{FSTA}^F > \mathcal{OBLOT}^F$

It is very easy to show that $\mathcal{FSTA}$ is strictly more powerful than $\mathcal{OBLI}$. To do that, we consider the *Oscillating Point Problem* defined in [10]

▶ **Definition 7.** ***Problem OSP (Oscillating Points) [10]:*** *Two robots, a and b, initially in distinct locations, alternately come closer and move further from each other. More precisely, let $d(t)$ denote the distance of the two robots at time t. The OSP problem requires the two robots, starting from an arbitrary distance $d(t_0) > 0$ at time t_0, to move so that there exists a monotonically increasing infinite sequence time instant $t_0, t_1, t_2, \dots$ such that :*

1. $d(t_{2i+1}) < d(t_{2i})$, *and* $\forall h', h" \in [t_{2i}, t_{2i+1}], h' < h", d(h") \leq d(h')$; *and*
2. $d(t_{2i}) > d(t_{2i-1})$, *and* $\forall h', h" \in [t_{2i-1}, t_{2i}], h' < h", d(h") \geq d(h')$.

Impossibility in $\mathcal{OBLOT}^F$ has been shown in [10]:

▶ **Lemma 8.** *[10]* $\exists R \in \mathcal{R}_2$, *OSP* $\notin \mathcal{OBLOT}^F(R)$.

On the other hand, possibility in $\mathcal{FSTA}^F$ is trivial because a robot can store in its local state whether in the previous round it was moving further or closer and successfully alternate movements. That is

▶ **Lemma 9.** $\forall R \in \mathcal{R}_2$, *OSP* $\in \mathcal{FSTA}^F(R)$.

By Lemmas 8 and 9, and the fact that $\mathcal{FSTA}^F \geq \mathcal{OBLOT}^F$ by definition, we have:

▶ **Theorem 10.** $\mathcal{FSTA}^F > \mathcal{OBLOT}^F$.

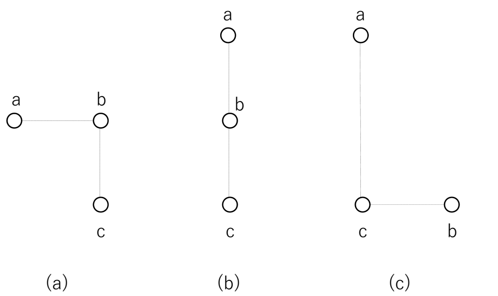

Figure 2 The configurations of problem ¬IL.

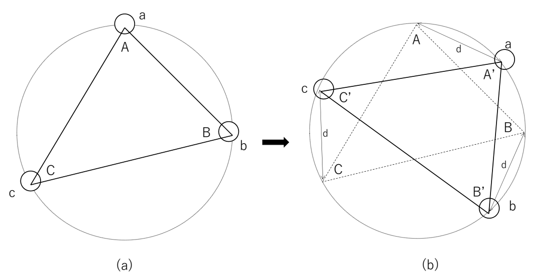

Figure 3 Illustration of TRIANGLE-ROTATION (TAR(d)).

4 COMPUTATIONAL RELATIONSHIP IN Ssynch

In this section, we examine the computational relationship of the models under the Semi-Synchronous scheduler.

4.1 Orthogonality of $\mathcal{FSTA}^S$ and $\mathcal{FCOM}^S$

▶ **Definition 11.** ***Problem TAR(d) (Triangle Rotation):*** *Let a, b, c be three robots forming a triangle ABC, let $\mathcal{C}$ be the circumscribed circle, and let d be a value known to the three robots. The $TAR(d)$ problem requires the robots to move so to form a new triangle $A'B'C'$ with circumscribed circle $\mathcal{C}$, and where $dis(A, A') = dis(B, B') = dis(C, C') = d$ (see Figure 3).*

▶ **Lemma 12.** *$\exists R \in \mathcal{R}_3$, $TAR(d) \notin \mathcal{FCOM}^S(R)$.*

Proof. (Sketch) By contradiction, let **A** be a correct solution protocol in $\mathcal{FCOM}^S$. Consider an initial configuration C_0 where the three robots a, b, and c, form a scalene triangle ABC with $AB \neq d$, $BC \neq d$, $CA \neq d$, and with all lights `off` (see Figure 3(a)). Consider now an execution $\mathcal{E}$ of **A** where all three robots are activated in each round, starting from C_0, until one or more robots move, say at round k. Let r be a robot that performed a non-null move in that round after observing configuration C_{k-1}. Consider now another execution $\mathcal{E}'$ of **A** where the first $k-1$ rounds are exactly the same, but in round k robot r is the only one activated. Robot r would move to a new location possibly changing color. Now the schedule activates again only robot r. If the previous move resulted in a scalene triangle, the robot cannot distinguish this situation from the one it observed at the previous round and thus it would perform the same type of movement, losing any information on the original triangle; if the previous move resulted in an equilateral or isosceles triangle, robot r would know it has already moved (even without having access to its light), but it still would not know from which location. In both cases the information on the original triangle cannot be reconstructed and the problem cannot be solved, contradicting the correctness of **A**. ◀

▶ **Lemma 13.** *$\forall R \in \mathcal{R}_3$, $TAR(d) \in \mathcal{FSTA}^S(R)$.*

Proof. The problem is easily solvable with $\mathcal{FSTA}$ robots in SSYNCH. Let the robots have color A initially. The first time a robot is activated, it moves to the desired position and changes its light to B. Whenever a robot is activated, if its light is B, it does not move. ◀

By Lemmas 4-5 and 12-13, we can conclude:

▶ **Theorem 14.** *$\mathcal{FCOM}^S \perp \mathcal{FSTA}^S$.*

4.2 Dominance of $\mathcal{FSTA}^S$ and $\mathcal{FCOM}^S$ over $\mathcal{OBLOT}^S$

The dominance of $\mathcal{FSTA}^S$ and $\mathcal{FCOM}^S$ over $\mathcal{OBLOT}^S$ follows directly from existing results on the rendezvous problem (RDV), which prescribes two robots to occupy exactly the same location, not known in advance.

▶ **Theorem 15.** *$\mathcal{FSTA}^S > \mathcal{OBLOT}^S$ and $\mathcal{FCOM}^S > \mathcal{OBLOT}^S$.*

Proof. It is well known that RDV cannot be solved in SSYNCH (see [27], whose proof uses chirality and trivially holds when movements are rigid). On the other hand, it can be solved in $\mathcal{FCOM}$ and $\mathcal{FSTA}$ in SSYNCH [17]. ◀

4.3 Dominance of $\mathcal{LUMI}^S$ over $\mathcal{FSTA}^S$ and $\mathcal{FCOM}^S$

To conclude the study of SSYNCH, we consider the OSP problem already employed in Section 3.3. also to show that $\mathcal{LUMI}^S > \mathcal{FSTA}^S(\mathcal{FCOM}^S)$.

▶ **Lemma 16.**
- $\exists R \in \mathcal{R}_2$, $OSP \notin \mathcal{FCOM}^S(R) \cup \mathcal{FSTA}^S(R)$.
- $\forall R \in \mathcal{R}_2$, $OSP \in LUMI^S(R)$.

Proof. The possibility in $\mathcal{LUMI}^S$ is proven in [10]. Let us then prove the impossibility in $\mathcal{FCOM}$ and $\mathcal{FSTA}$. Let a and b be the two robots with initial lights `off`. First note that if an activated robot performs a null move at the first round, the adversarial scheduler would activate both (making them change lights in the same way). The scheduler continues to activate them both until the first round t when the color of the light would make them do a non-null move. At this point, the scheduler changes strategy.

In the case of $\mathcal{FCOM}$, the scheduler activates only robot a in the two consecutive rounds t and $t+1$. At round $t+2$, robot a is activated again. Robot a will repeat (incorrectly) the same move at round $t+2$, not being able to distinguish the current situation from the previous, and regardless of the movement taken in round t.

In the case of $\mathcal{FSTA}$, the scheduler activates only robot a for 3 consecutive rounds $t, t+1, t+2$ and both robots at round $t+3$. In the first 3 activations robot a can use its internal light to correctly alternate a move going closer to b, one moving further and the third moving closer again. At round $t+3$, robot a will necessarily move further from b continuing this alternating pattern (as nothing has changed in its perceived view of the universe), but robot b is now in the same state robot a was at round t and will therefore take the same action taken by a at that round (i.e., moving closer to a). This lack of synchronization makes the robots incorrectly maintain their distance during round $t+3$. ◀

We can conclude that:

▶ **Theorem 17.** *$\mathcal{LUMI}^S > \mathcal{FSTA}^S$ and $\mathcal{LUMI}^S > \mathcal{FCOM}^S$.*

5 COMPUTATIONAL RELATIONSHIP BETWEEN Fsynch AND Ssynch

In this section we examine the computational relationship of fully synchronous and semi-synchronous models.

5.1 Dominances of Fsynch over Ssynch

The following problem prescribes the robots to perform a sort of "expansion" of the initial configuration with respect to their center of gravity; specifically, each robot must move away from the center of gravity (c_x, c_y) to the closest integral position corresponding to doubling its distance from it. More precisely:

▶ **Definition 18.** ***Problem CGE (Center of Gravity Expansion)**: Let R be a set of robots. The CGE problem requires each robot $r_i \in R$ to move from its initial position (x_i, y_i) directly to $(f(x_i, c_x), f(y_i, c_y))$, where $f(a, b) = \lfloor 2a - b \rfloor$ and (c_x, c_y) is the center of gravity of the initial configuration.*

▶ **Lemma 19.** *$CGE \in \mathcal{FSTA}^F$ and $CGE \notin \mathcal{LUMI}^S$.*

Proof. (Sketch) It is easy to see that $CGE \in \mathcal{FSTA}^F$ since all robots can simultaneously reach their destination in one step and change color to indicate termination. We now show that $CGE \notin \mathcal{LUMI}^S$. By contradiction. Consider an execution $\mathcal{E}$ of a solution algorithm where a single robot r is activated at the first time step. The robot moves correctly to its destination point and possibly changes its color. After this movement, regardless of the distance traveled, the center of gravity of the new configuration is different from the one of the initial configuration, with respect to which all the other robots must move. At the next activation, any robot different from r must move to its target location; however, this cannot be done because the robot cannot reconstruct the exact position of the original center of gravity. This is due to the fact that there are infinite combinations of coordinates from where r could have feasibly moved and the reconstruction of the original CoG cannot be done just on the basis of a light that can carry finite information. ◀

As a consequence, we have that:

▶ **Theorem 20.**

1. $\mathcal{LUMI}^F > \mathcal{LUMI}^S$
2. $\mathcal{FSTA}^F > \mathcal{FSTA}^S$
3. $\mathcal{FCOM}^F > \mathcal{LUMI}^S > \mathcal{FCOM}^S$
4. $\mathcal{OBLOT}^F > \mathcal{OBLOT}^S$

Proof.

1. It follows from Lemma 19, Theorem 2, and Theorem 6.
2. It follows from Lemma 19 and Theorem 17.
3. It follows immediately from Theorem 2, Theorem 17, and Theorem 20.
4. The RDV problem can be trivially solved in $\mathcal{OBLOT}^F$ but it cannot be solved in $\mathcal{OBLOT}^S$[27]. ◀

5.2 Incomparabilities between Fsynch and Ssynch

5.2.1 Orthogonality of $\mathcal{OBLOT}^F$ with $\mathcal{FCOM}^S$ and $\mathcal{FSTA}^S$

Consider the following problem:

▶ **Definition 21.** ***Problem SRO (Shrinking Rotation)****: Two robots a and b are initially placed in arbitrary distinct points (forming the initial configuration C_0), The two robots uniquely identify a square (initially Q_0) whose diagonal is given by the segment between them*[4]*. Let a_0 and b_0 indicate the initial positions of the robots, d_0 the segment between them, and $length(d_0)$ its length. Let a_i and b_i be the positions of a and b in configuration C_i ($i \geq 0$). The problem consists of moving from configuration C_i to C_{i+1} in such a way that Condition* ***C3*** *is verified and so is one of* ***C1*** *and* ***C2****:*

C1. *d_{i+1} is a 90 degree clockwise rotation of d_i and thus $length(d_{i+1}) = length(d_i)$,*

C2. *d_{i+1} is a "shrunken" 45 degree clockwise rotation of d_i such that $d_{i+1} = \frac{d_i}{\sqrt{2}}$,*

C3. *a_{i+1} and b_{i+1} must be included in the square Q_{i-1}, where Q_{-1} is the infinite square.*

[4] By square, we means the entire space delimited by the four sides.

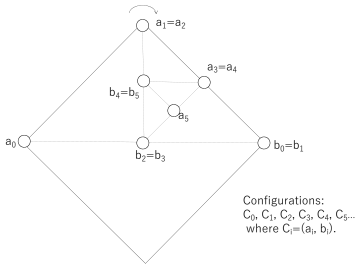

Figure 4 Illustration of SHRINKING ROTATION (SRO).

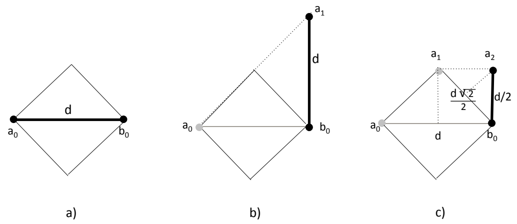

Figure 5 Proof of Lemma 23: *a*) Initial configuration; *b*) after the movement of robot a in Case (1); *c*) after two consecutive movements of robot a in Case (2).

▶ **Lemma 22.** $\forall R \in \mathcal{R}_2, SRO \in \mathcal{OBLOT}^F(R)$

Proof. The proof is by construction: Each robot rotates clockwise of 90 degrees with respect to the midpoint between itself and the other robot. Since the schedule is FSYNCH, it allows consecutive simultaneous activation of the two robots. So, there is only one possible type of executions under FSYNCH with two robots: a perpetual activation of both robots in each round. In this case, the problem is clearly solved by the algorithm stated above, because the robots keep rotating of 90 degrees clockwise around their mid-point, fulfilling **C1** and **C3**. Note that **C2** never happens under FSYNCH. Then SRO can be solved with $\mathcal{OBLOT}$ in FSYNCH. ◀

▶ **Lemma 23.** $\exists R \in \mathcal{R}_2, SRO \notin \mathcal{FCOM}^S(R) \cup \mathcal{FSTA}^S(R)$

Proof. First note that if an activated robot performs a null move at the first round, the schedule would activate both (making them change lights in the same way). The scheduler continues to activate them both until the first round i when the color of the light would make them do a non-null move. At this point, the scheduler changes strategy.

Consider first the case of $\mathcal{FCOM}^S$ and consider an execution where a robot, say a, is activated (alone) twice consecutively starting from configuration C_i. In the following, we show that, under this activation schedule, either C_{i+1} or C_{i+2} would violate **C3** (which states that a_{i+1} and b_{i+1} must be included in the square Q_{i-1}) (see Figure 4).
In fact, let robot a located at a_i be activated from a configuration C_i. Since b is not activated in C_i, the light of b at b_i and at b_{i+1} are the same. Then a at a_i and at a_{i+1} observe the same light on b. Since the coordinate systems of the robot can be chosen so that they have the same view of the universe, a at a_{i+1} performs the same action as it would perform at a_i, and this action must either fulfill **C1** or **C2** (as well as **C3** in either case).
Case (1). Let us consider first the situation when **C1** is fulfilled with a single movement of a: the only possibility would be for a to rotate clockwise of 90 degree with respect to b; this movement, however, would immediately violate **C3** because the new position a_{i+1} would be outside of the square Q_i (and thus also outside Q_{i-1}) (see Figure 5 from *a*) to *b*)).
Case (2). Let us consider now the case when **C2** is fulfilled with a single movement of a: the only possibility would be for a to move clockwise of 90 degrees with respect to the midpoint between a and b reaching a feasible configuration C_{i+1}. When robot a is activated again at the next round, it will perform the same action on C_{i+1}, now violating **C3** (see Figure 5 from *a*) to *c*)).

Therefore, this problem cannot be solved with $\mathcal{FCOM}$ in SSYNCH. The case of $\mathcal{FSTA}^S$ can be shown in a similar way, because the availability of internal lights cannot prevent - in SSYNCH - the consecutive activation of the same single robot and the impossibility argument described above would still hold. ◀

Moreover, we have:

▶ **Lemma 24.** $\forall R \in \mathcal{R}_2, SRO \in \mathcal{LUMI}^S(R)$

Proof. It is rather straightforward to see that in $\mathcal{LUMI}^S$ the two robots can be synchronized with 3 colors so to enforce a fully synchronous execution. ◀

We have seen that SRO can be solved in $\mathcal{OBLOT}^F$ but cannot be solved in $\mathcal{FCOM}^S$ and $\mathcal{FSTA}^S$. On the other hand, $\neg IL$ and TAR(d) can be solved in $\mathcal{FCOM}^S$ and $\mathcal{FSTA}^S$, respectively, but cannot be solved in $\mathcal{OBLOT}^F$. We can conclude that:

▶ **Theorem 25.** *$\mathcal{OBLOT}^F \perp \mathcal{FCOM}^S$ and $\mathcal{OBLOT}^F \perp \mathcal{FSTA}^S$.*

5.2.2 Orthogonality of $\mathcal{LUMI}^S$ with $\mathcal{FSTA}^F$ and $\mathcal{OBLOT}^F$

▶ **Theorem 26.** *$\mathcal{LUMI}^S \perp \mathcal{FSTA}^F$ and $\mathcal{FCOM}^S \perp \mathcal{FSTA}^F$.*

Proof. Problem ¬IL can be solved in $\mathcal{FCOM}^S$ (and thus in $\mathcal{LUMI}^S$) but not in $\mathcal{FSTA}^F$ (Lemmas 4 and 5). Problem CGE can be solved in $\mathcal{FSTA}^F$, but not in $\mathcal{LUMI}^S$ (Lemma 19). ◀

▶ **Definition 27.** ***Problem CGE* (Perpetual Center of Gravity Expansion).*** *This is the same as CGE, where however after each expansion, the robots have to repeat the same process from the new configuration.*

▶ **Theorem 28.** *$\mathcal{LUMI}^S \perp \mathcal{OBLOT}^F$.*

Proof. Problem OSP can be solved in $\mathcal{LUMI}^S$ (Lemma 16), but not in $\mathcal{OBLOT}^F$ (Lemma 8). Problem CoG* can be trivially solved in $\mathcal{OBLOT}^F$, but not in $\mathcal{LUMI}^S$ (Lemma 19). ◀

Let us remark that, since $\mathcal{LUMI}^s \equiv \mathcal{LUMI}^A$, the result of Theorem 28 answers the open question on the relationship between $\mathcal{LUMI}^A$ and $\mathcal{OBLOT}^F$ posed in [10].

6 CONCLUDING REMARKS

In this paper, we have investigated the computational power of communication versus persistent memory in mobile robots by studying the relationship among $\mathcal{LUMI}$, $\mathcal{FCOM}$, $\mathcal{FSTA}$ and $\mathcal{OBLOT}$ models, and we have shown that their relationship depends of the scheduler under which the robots operate. We considered the two classical synchronous schedulers, FSYNCH and SSYNCH, establishing several results. In particular, we proved that communication is more powerful than persistent memory if the scheduler is fully synchronous; on the other hand, the two models are incomparable under the semi-synchronous scheduler. For an overall panorama of the established relationship among the models, see Figure 6.

Several problems are still open. An outstanding open problem is the study of the relationship among these models in ASYNCH, where there is no notion of rounds and the cycles of the robots are executed independently.

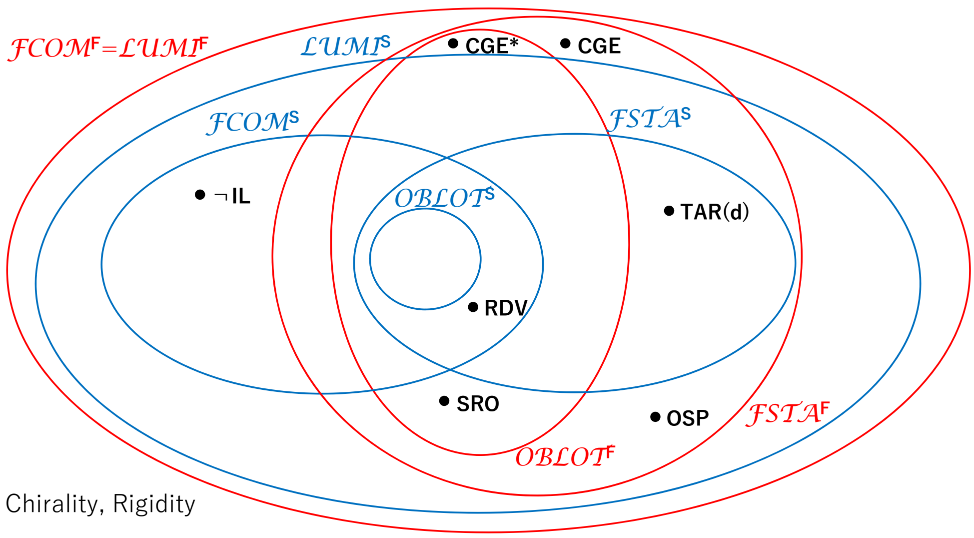

Figure 6 Relationship among $\mathcal{LUMI}$, $\mathcal{FCOM}$, $\mathcal{FSTA}$ and $\mathcal{OBLOT}$ in FSYNCH, and SSYNCH assuming chirality and rigidity.

Another open problem is whether there exists a scheduler S' ("weaker" than FSYNCH but stronger than SSYNCH) such that each model under S' would be computationally equivalent to the same model under FSYNCH.

Finally, most of the results of this paper hold assuming chirality and rigidity (exceptions are the RDV-algorithms, the OSP-algorithms, and the simulation algorithm, Algorithm 1, which do not require either). It is an open question to characterize the inclusions among all the various models in the case of disoriented robots with non-rigid movement.

References

1. N. Agmon and D. Peleg. Fault-tolerant gathering algorithms for autonomous mobile robots. *SIAM Journal on Computing*, 36(1):56–82, 2006.
2. H. Ando, Y. Osawa, I. Suzuki, and M. Yamashita. A distributed memoryless point concergence algorithm for mobile robots with limited visivility. *IEEE Transactions on Robotics and Automation*, 15(5):818–828, 1999.
3. S. Bhagat and K. Mukhopadhyaya. Optimum algorithm for mutual visibility among asynchronous robots with lights. In *Proc. of the 19th International Symposium on Stabilization, Safety, and Security of Distributed Systems (SSS)*, pages 341–355, 2017.
4. Z. Bouzid, S. Das, and S. Tixeuil. Gathering of mobile robots tolerating multiple crash Faults. In *the 33rd Int. Conf. on Distributed Computing Systems (ICDCS)*, pages 334–346, 2013.
5. D. Canepa and M. Gradinariu Potop-Butucaru. Stabilizing flocking via leader election in robot networks. In *Proc. of the 10th International Symposium on Stabilization, Safety, and Security of Distributed Systems (SSS)*, pages 52–66, 2007.
6. S. Cicerone, Di Stefano, and A. Navarra. Gathering of robots on meeting-points. *Distributed Computing*, 31(1):1–50, 2018.

7 M. Cieliebak, P. Flocchini, G. Prencipe, and N. Santoro. Distributed computing by mobile robots: Gathering. *sicomp*, 41(4):829–879, 2012.

8 R. Cohen and D. Peleg. Convergence properties of the gravitational algorithms in asynchronous robot systems. *sicomp*, 34(15):1516–1528, 2005.

9 S. Das, P. Flocchini, G. Prencipe, N. Santoro, and M. Yamashita. The power of lights: synchronizing asynchronous robots using visible bits. In *Proc. of the 32nd International Conference on Distributed Computing Systems (ICDCS)*, pages 506–515, 2012.

10 S. Das, P. Flocchini, G. Prencipe, N. Santoro, and M. Yamashita. Autonomous mobile robots with lights. *Theoretical Computer Science*, 609:171–184, 2016.

11 G.A. Di Luna, P. Flocchini, S.G. Chaudhuri, F. Poloni, N. Santoro, and G. Viglietta. Mutual visibility by luminous robots without collisions. *Information and Computation*, 254(3):392–418, 2017.

12 G.A. Di Luna and G. Viglietta. *Robots with Lights*. Chapter 11 of [13], pages 252–277, 2019.

13 P. Flocchini, G. Prencipe, and N. Santoro (Eds). *Distributed Computing by Mobile Entities*. Springer, 2019.

14 P. Flocchini, G. Prencipe, N. Santoro, and P. 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.

15 P. Flocchini, G. Prencipe, N. Santoro, and P. Widmayer. Gathering of asynchronous robots with limited visibility. *tcs*, 337(1–3):147–169, 2005.

16 P. Flocchini, G. Prencipe, N. Santoro, and P. Widmayer. Arbitrary pattern formation by asynchronous oblivious robots. *tcs*, 407:412–447, 2008.

17 P. Flocchini, N. Santoro, G. Viglietta, and M. Yamashita. Rendezvous with Constant Memory. *tcs*, 621:57–72, 2016.

18 N. Fujinaga, Y. Yamauchi, H. Ono, S. Kijima, and M. Yamashita. Pattern formation by oblivious asynchronous mobile robots. *sicomp*, 44(3):740–785, 2015.

19 V. Gervasi and G. Prencipe. Coordination without communication: The case of the flocking problem. *Discrete Applied Mathematics*, 144(3):324–344, 2004.

20 A. Hériban, X. Défago, and S. Tixeuil. Optimally gathering two robots. In *Proc. of the 19th Int. Conference on Distributed Computing and Networking (ICDCN)*, pages 1–10, 2018.

21 T. Izumi, S. Souissi, Y. Katayama, N. Inuzuka, X. Défago, K. Wada, and M. Yamashita. The gathering problem for two oblivious robots with unreliable compasses. *sicomp*, 41(1):26–46, 2012.

22 T. Okumura, K. Wada, and X. Défago. Optimal Rendezvous $\mathcal{L}$-Algorithms for Asynchronous Mobile Robots with External-Lights. In *Proc. of the 22nd Int. Conference on Principles of Distributed Systems (OPODIS)*, pages 24:1–24:16, 2018.

23 T. Okumura, K. Wada, and Y. Katayama. Brief Announcement: Optimal asynchronous Rendezvous for mobile robots with lights. In *Proc. of the 19th Int. Symposium on Stabilization, Safety, and Security of Distributed Systems (SSS)*, pages 484–488, 2017.

24 D. Peleg. Distributed coordination algorithms for mobile robot swarms: New directions and challenges. In *Proc. of 7th International Workshop on Distributed Computing (IWDC)*, pages 1–12, 2005.

25 G. Sharma, R. Alsaedi, C. Bush, and S. Mukhopadyay. The complete visibility problem for fat robots with lights. In *Proc. of the 19th International Conference on Distributed Computing and Networking (ICDCN)*, pages 21:1–21:4, 2018.

26 S. Souissi, T. Izumi, and K. Wada. Oracle-based flocking of mobile robots in crash-recovery model. In *Proc. of the 11th International Symposium on Stabilization, Safety, and Security of Distributed Systems (SSS)*, pages 683–697, 2009.

27 I. Suzuki and M. Yamashita. Distributed anonymous mobile robots: Formation of geometric patterns. *sicomp*, 28:1347–1363, 1999.

28 S. Terai, K. Wada, and Y. Katayama. Gathering problems for autonomous mobile robots with lights. *arXiv.org*, cs(ArXiv:1811.12068), 2018.

29 G. Viglietta. Rendezvous of two robots with visible bits. In *10th International Symposium on Algorithms and Experiments for Sensor Systems, Wireless Networks and Distributed Robotics (ALGOSENSORS)*, pages 291–306, 2013.

30 M. Yamashita and I. Suzuki. Characterizing geometric patterns formable by oblivious anonymous mobile robots. *tcs*, 411(26–28):2433–2453, 2010.

31 Y. Yamauchi, T. Uehara, S. Kijima, and M. Yamashita. Plane formation by synchronous mobile robots in the three-dimensional euclidean space. *Journal of the ACM (JACM)*, 64:3(16):16:1–16:43, 2017.