IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 22 December 2022, accepted 5 February 2023, date of publication 14 February 2023, date of current version 21 February 2023.

*Digital Object Identifier 10.1109/ACCESS.2023.3244844*

RESEARCH ARTICLE

# Molecular Oblivious Robots: A New Model for Robots With Assembling Capabilities

**SERAFINO CICERONE[1], ALESSIA DI FONSO[1], GABRIELE DI STEFANO[1], AND ALFREDO NAVARRA[2]**

[1] Department of Information Engineering, Computer Science and Mathematics, University of L'Aquila, 67100 L'Aquila, Italy
[2] Department of Mathematics and Computer Science, University of Perugia, 06123 Perugia, Italy

Corresponding author: Serafino Cicerone (serafino.cicerone@univaq.it)

This work was supported in part by the Italian National Group for Scientific Computation-Istituto Nazionale di Alta Matematica (GNCS-INdAM).

**ABSTRACT** Research in theoretical swarm robotics focuses on models that assign to robots a minimal set of capabilities. One of the models well investigated is certainly $\mathcal{OBLOT}$, addressing the case of distributed robots that are, anonymous, without means of communication, and *oblivious*. Here we propose $\mathcal{MOBLOT}$, an extension of $\mathcal{OBLOT}$ that allows to resolve a larger spectrum of cases. $\mathcal{MOBLOT}$ stands for *molecular oblivious robots*: like atoms combine themselves to form molecules, in $\mathcal{MOBLOT}$ simple robots can bond with each other in order to create possibly bigger computational units with more intrinsic capabilities with respect to robots (called *molecules* also in the model); like in nature, molecules can further bond to create more complex structures (e.g., the matter), the $\mathcal{MOBLOT}$ version of molecules can exploit their own capabilities to accomplish new tasks or simply to arrange themselves to form any shape defined according to some compositional properties. In order to better understand the potentials of $\mathcal{MOBLOT}$, we introduce a new problem called *matter formation* (MF). We do provide a necessary condition for the solvability of MF, in general. This relies on the 'amount of symmetries' arising by the disposal of the robots. In practice, we show how molecules can break certain symmetries that cannot be broken in $\mathcal{OBLOT}$. Finally, as a case study of $\mathcal{MOBLOT}$, we consider a representative problem derived from the general MF problem along with a distributed resolution algorithm and show its correctness.

**INDEX TERMS** Distributed algorithms, swarm and modular robotics, self-organizing systems, oblivious robots, pattern formation.

## I. INTRODUCTION

Robotics is an active area of research that includes many computer science and engineering disciplines. Dealing with robotics concerns the design, the construction, the operation, and the use of robots. In particular, two main research areas have been deeply investigated in robotics: *modular robotics* (eg, see [2]) and *swarm robotics* (eg, see [3], [4]).

Concerning modular robots, those refer to robotic systems where interconnected individual (electro-mechanical) modules can recover from failures or rearrange in order to better adapt to their task-environment (eg, see [5]). Their main objective is to achieve robotic systems that are more versatile, affordable, and robust than their standard counterparts, at the cost of a probable reduced efficacy for specific tasks. This concept was first introduced in the late 1980's as cellular robotic systems by T. Fukuda, later physically realized in the CEBOT modular robot by Fukuda and Kawauchi [6]. Since then, the field has been renamed modular robotics, and various robotic architectures have been proposed [7], [8].

Concerning swarm robotics, they differ from modular robotics as single robots in the system do not need to maintain the connection with each other at all times, but they are usually mobile units with full autonomy (e.g., Kilobot [9]). In such a context, the interaction among robots in some specific form should lead to a desired collective behavior. This approach emerged in the field of artificial swarm intelligence, as well as the biological studies of insects, ants and other fields in nature, where swarm behavior occurs. Researchers in the field of swarm robotics mainly follow

The associate editor coordinating the review of this manuscript and approving it for publication was Theofanis P. Raptis.

a theoretical approach that considers robot systems in the abstract, where the complexity and capabilities of the robots induced by the underlying model are often reduced to their minimum. Representative models in this context are the well-investigated Amoebot model [10], [11], and the more recent models Silbot [12], [13], [14], and Pairbot [15]. One of the main issue faced when dealing with such models, is that they help, in general, to rigorously analyze the designed algorithms, hence providing new theoretical insights that subsequently also extend the practical aspect of the studied systems.

One of the models well investigated in swarm robotics is certainly $\mathcal{OBLOT}$ (see, e.g., [16]), addressing the case of distributed robots moving in some environment. The $\mathcal{OBLOT}$ model can be considered as a sort of framework within which many different settings can be manipulated, each implied by specific choices among a range of possibilities, with respect to fundamental components like time synchronization as well as other important elements, such as memory, orientation, mobility and communication. Such settings are often maintained at minimum, thus defining very weak computational entities: robots are *oblivious* (no memory about past activities), *identical* (indistinguishable from their appearance), *anonymous* (no distinct identities that can be used during the computation), *autonomous* (they operate without a central control or external supervision), *homogeneous* (they all execute the same algorithm), *silent* (they have no means of direct communication with each other), and *disoriented* (each robot operates in its own local coordinate system). When a robot is activated, it enters in a so-called `Look`–`Compute`–`Move` cycle: it acquires a snapshot of the current global configuration (`Look`) as the disposal of the robots with respect to its own coordinate system. Successively, it decides whether to move toward a specific target or not (`Compute`), and in the positive case it moves (`Move`).

From a practical view point, the technology required to implement algorithms designed within $\mathcal{OBLOT}$ does not rely on special sensors nor actuators. Hence, cheap hardware might be used and experimented. One of the main peculiarities that makes $\mathcal{OBLOT}$ really appealing is the absence of direct communications among entities. Everything is computed on the base of stigmergic information (i.e., concerning to the relative positions of the entities only). An example of real robots working this way can be found in [17]. In [18], a robotic system governed by the $\mathcal{OBLOT}$ model has been realized by means of standard educational robots, solving the Gathering problem, that is the task to bring all the robots as much close to each other as possible. Similarly, [19], [20] deal with Gathering of robots moving on a ring, whereas in [21] robots gather on dense graphs.

In general, robot's capabilities are maintained as weak as possible so as to understand what is the limit for the feasibility of the problems. Moreover, the less assumptions are made, the more a resolution algorithm is robust with respect to possible disruptions.

Here we propose $\mathcal{MOBLOT}$, a new theoretical model concerning swarm robotics that extends $\mathcal{OBLOT}$ to address a larger spectrum of problems, including those more related to modular robotics. $\mathcal{MOBLOT}$ stands for *Molecular OBLivious robOTs*. The inspiration comes from nature: like atoms combine themselves to form molecules, in $\mathcal{MOBLOT}$ simple robots can bond with each other in order to create possibly bigger computational units with more intrinsic capabilities with respect to robots (called *molecules* also in the model). A molecule is specified by a pattern. When robots move so as to be correctly positioned according to the pattern they firmly bond to make a molecular robot; like in nature, molecules can further bond to create more complex structures (e.g., the matter), the $\mathcal{MOBLOT}$ version of molecules can exploit their own capabilities to accomplish new tasks or simply to arrange themselves (with movements specific to the molecules rather than to the single robots) to form any shape defined according to some compositional properties. Once a task is accomplished (e.g., a target shape is formed) in case of other inputs/stimuli, the molecules can start a new task (e.g., rearrange or self-reconfigure their positions to modify the shape). This last observation highlights how also problems arising in modular robotics contexts might be approached by means of the $\mathcal{MOBLOT}$ paradigm.

It is worth noting that one of the most studied problem within $\mathcal{OBLOT}$ model is certainly the Pattern Formation (PF) [22], [23], [24], [25], [26], [27], [28], [29]: given a team of robots and a geometric pattern in terms of points in the plain with respect to an ideal coordinate system (not known to the robots), the goal is to design a distributed algorithm that works for each robot to guide it so that eventually all robots together form the pattern, if possible.

Determining which patterns are formable under what conditions has been the subject of extensive studies in a variety of settings. A crucial role is played by the notion of *symmetricity* of a set of points, which informally measures the amount of rotational symmetries of the set. Indeed, for a pattern to be formable, its symmetricity must be divided by the one of the starting configuration (i.e., initial placement of the robots in the plane). The obtained results indicate that, in most settings, the robots can form single arbitrary geometric patterns in spite of their weak capabilities. In other words, apart from the inevitable limitations due to symmetry, it is worth noting how obliviousness, silence, and all other minimal assumptions are not limiting factors to form a single pattern.

An extension of Pattern Formation within $\mathcal{MOBLOT}$ is what we call the *Matter Formation* (MF) problem: given a team of robots, the definition of a set of molecules (ideally small patterns composed by a few of robots), and a geometric pattern in terms of molecules (possibly with some extent) in the plane with respect to an ideal coordinate system (not known to the robots nor to the molecules), the aim is to provide an algorithm that works for each single robot and for each composed molecule, in a distributed way, to guide them so that eventually the pattern if formed, if possible.

Intuitively, MF can be thought as a representative problem within $\mathcal{MOBLOT}$, like the PF is for $\mathcal{OBLOT}$. Hence, in order to start to investigate on the peculiarities of the $\mathcal{MOBLOT}$ model, it is worth considering MF.

### A. OUR RESULTS

In this paper, we formally define the new $\mathcal{MOBLOT}$ model by which both swarm and modular robotics can be approached. It is easy to see that $\mathcal{MOBLOT}$ extends $\mathcal{OBLOT}$ as the case where a molecule is considered to be composed by just one robot, without extra capabilities, clearly coincides with the $\mathcal{OBLOT}$ model.

As $\mathcal{MOBLOT}$ represents an extension of $\mathcal{OBLOT}$ in terms of possible applications, we introduce the Matter Formation (MF) problem as a natural extension to the classical Pattern Formation problem studied within $\mathcal{OBLOT}$.

For the problem, in general, we establish a condition about the solvability of MF which is necessary. Such a condition, in fact, relies on *symmetricity*. We show the power of dealing with molecules as in some cases they can resolve the symmetry breaking issue that would be unsolvable within $\mathcal{OBLOT}$.

Furthermore, we consider as a case study a MF problem obtained by specifying the possible molecules to be formed as well as their compositional properties in order to form the matter. The considered problem, called *Hexagonal Matter Formation* problem (HexMF for short), has been selected to specifically highlight some symmetry breaking abilities of the robots that cannot be exploited within $\mathcal{OBLOT}$. We then provide a distributed resolution algorithm for HexMF. In this respect, we show how the formal methodology thought for $\mathcal{OBLOT}$ in [30] to design resolution algorithms along with the corresponding correctness proofs can be still applied to manage the $\mathcal{MOBLOT}$ environment where also molecules, not only robots, must be managed.

### B. OUTLINE

The paper is organized as follows. Since the $\mathcal{MOBLOT}$ model is an extension of $\mathcal{OBLOT}$, in the next section we recall the latter model along with the general Pattern Formation problem. $\mathcal{MOBLOT}$ and the new Matter Formation problem are then formally introduced in Section III. In Section IV, in order to appreciate the capabilities of the $\mathcal{MOBLOT}$ model, we define a case study by introducing and solving HexMF, a specific matter formation problem. In particular, we first formalize this MF problem, then we motivate its definition according to some necessary conditions, and finally we provide an overview of the resolution strategy. In Sections V and VI we respectively provide all the details regarding the provided resolution algorithm and its correctness. Finally, in Section VII we conclude the paper by formally comparing the $\mathcal{OBLOT}$ and $\mathcal{MOBLOT}$ models and by highlighting some final remarks.

## II. OBLIVIOUS ROBOTS

A robotic system within $\mathcal{OBLOT}$ (cf. [16]) is represented by a set $R = \{r_1, r_2, \ldots, r_n\}$ of $n$ entities, called robots, that live and operate in a connected spatial universe $\mathcal{U} \subseteq \mathbb{R}^d$, $d \geq 1$, in which they can move. Robots are considered to be **dimensionless**, i.e., as points in $\mathbb{R}^d$. Moreover, at the same time, the same location can be occupied by more than one robot, in which case we say a **multiplicity** occurs.

Very minimal assumptions are considered for the robots: they are **identical** (they are indistinguishable by their external appearance), **anonymous** (they do not have distinct identities that can be used during the computation), **autonomous** (they operate without a central control or external supervision), **homogeneous** (they all have and execute the same protocol, or algorithm), **silent** (they have no means of direct communication of information to other robots), and **disoriented** (each robot has its own local coordinate system - LCS). A robot is capable of observing $\mathcal{U}$ hence determining the positions (expressed in its own LCS) of all the robots. It follows that the only means of interaction between robots is given by observations and movements, that is, communication is *stigmergic*. The behavior of a robot follows four sequential states:

- `Wait`. The robot is idle. A robot cannot stay indefinitely idle.
- `Look`. The robot observes $\mathcal{U}$ by activating its sensors which return a snapshot of the positions of all other robots expressed in its own LCS. If robots are capable of observing the entire universe $\mathcal{U}$, then robots are said to admit **unlimited visibility**; if robots can observe $\mathcal{U}$ only within a visibility range then robots have **limited visibility**.
- `Compute`. The robot performs a local computation according to a deterministic algorithm $\mathcal{A}$ (we also say that the robot executes $\mathcal{A}$). The algorithm is the same for all robots, and the result of the `Compute` phase is a destination point along with the trajectory to be followed.
- `Move`. The robot moves toward the computed destination; if the destination is the current location, the robot stays still, performing a *nil* movement.

Such states form a **computational cycle** of a robot. Robots are **oblivious**: at the beginning of each computational cycle, the robots have no memory of past actions, and the computation is based only on what determined in their current cycle. In particular, from the snapshot acquired during the `Look` phase, a robot can elaborate what later is called its **view**. This is a data structure containing all the information deductible by a robot during its `Look` phase. Since each robot is subject to a local coordinate system, the view cannot exploit absolute measurements but it is based on relative angles and positions of robots. Details about the view can be found in Section V-B.

### A. VARYING THE COMPONENTS OF THE SYSTEM

The $\mathcal{OBLOT}$ model is based on the fundamental properties recalled so far. Anyway, it can address a larger spectrum of situations by varying some additional components.

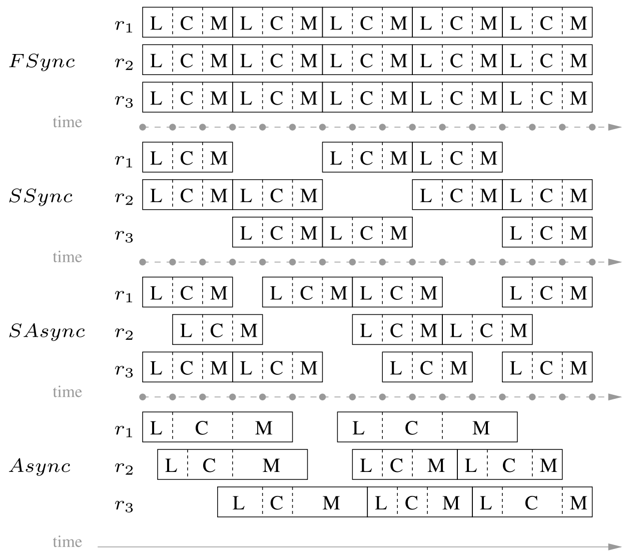

**FIGURE 1.** **The execution model of computational cycles for each of** FSync**,** SSync**,** SAsync**, and** Async **robots. The inactivity of robots is implicitly represented by empty time periods.**

### 1) TIME SCHEDULER

Among the most important features that can greatly vary the computational power of robots there is the *time scheduler*. Four schedulers are in general used:

- In the *Semi-synchronous* (SSync) scheduler, the activations of the robots is logically divided into global rounds; in each round, one or more robots are activated and obtain the same snapshot; based on that snapshot, they compute and perform their move, ending their cycle by the next round. The choice of which robots are activated in a given round is assumed to be made by the time scheduler.
- The *Fully-synchronous* (FSync) scheduler is an extreme case of SSync where all the robots are activated in every round;
- *Semi-Asynchronous* (SAsync): Robots are activated independently. Like in FSync or SSync, the duration of each phase is assumed to be always the same. Differently from FSync or SSync, two activated robots can be in different phases even though phases are synchronized.
- *Asynchronous* (Async): The robots are activated independently, and the duration of each phase is finite but unpredictable. In other words, robots do not have a common notion of time. As a result, computations can be made based on totally obsolete observations, taken arbitrarily far in the past. Moreover, they can be seen while moving, and computations can be made based on obsolete information about positions.

In schedulers different from FSync, it is necessary to guarantee that the time scheduler is **fair**: for every robot $r$ and time $t$, there exists a time $t' \geq t$ at which $r$ is activated; that is, every robot is activated infinitely often.

### 2) ORIENTATION

Concerning the orientation, we have already recalled that, in general, robots are assumed to be disoriented: each of them has its own LCS and its unit of measure. It is possible to customize the system by assuming that all robots agree on the direction and orientation of $k$ axes $(1 \leq k \leq d)$. Moreover, robots may have *chirality*, that is they agree on a cyclic orientation (e.g., clockwise) of the plane.

### 3) MOBILITY

The actual movement of a mobile robot is controlled by an external *mobility scheduler*. The scheduler decides how fast the robot moves toward its destination point, and it may even interrupt its movement before the destination point is reached. Two variants can be defined: *rigid* (or unlimited) mobility, where all robots always reach their destinations when performing `Move`; *non-rigid*, where the distance traveled within a move is neither infinite nor infinitesimally small. More precisely, the mobility scheduler has also the power to stop a moving robot before it reaches its destination, but there exists an unknown constant $\delta > 0$ such that if the destination point is closer than $\delta$, the robot will reach it, otherwise the robot will be closer to it of at least $\delta$.

### 4) EXTENT

In the standard model, robots are viewed as points; i.e., they are dimensionless. This property can be varied by assuming robots with a physical dimension, that is, entities with an extent. These robots are called *solid* (or *fat* as in [31], [32], and [33]) and are viewed as opaque circular disks of a fixed diameter (hence they are assumed to have a common unit distance).

### 5) MEMORY, APPEARANCE AND COMMUNICATION

Robots can be endowed with a persistent and externally visible state variable, called *visible light*, that can assume values from a finite set of colors. The light can be set in each cycle by the robot at the end of its `Compute` operation. It is externally visible in the sense that its color at time $t$ is visible to all robots in its visibility radius that perform a `Look` operation at that time. It is persistent in the sense that, while the robot is oblivious and forgets all information from previous cycles, the variable is not automatically reset at the end of a cycle. The color a robot sees is used as input during the computation. It follows that luminous robots can be thought as robots with persistent information that can be used not only to remember but also to communicate. Moreover, their appearance is also affected by the colors, i.e., only robots associated with the same color maintain the property to be identical.

### 6) ADVERSARY

The mobility scheduler as well as the time scheduler are both managed by an ideal *adversary*. In fact, such schedulers are completely out of the control of the robots. This does not mean this is a centralized environment but simply that any possible event can occur. However, the occurrence of an event is not just thought as a random process, since otherwise one may infer some properties with high probability. The adversarial technique, instead, is a way to keep in mind the worst-case scenario.

### B. THE PATTERN FORMATION PROBLEM

Regardless of the adversary, the activations of the robots determine specific ordered time instants. Let $R(t)$ be the configuration observed by some robots at time $t$ during their `Look` phase, and let $\{t_i : i = 0, 1, \ldots\}$, with $t_i < t_{i+1}$, be the set of all time instances at which at least one robot takes the snapshot $R(t_i)$. Since the information relevant for the computing phase of each robot is the order in which the different snapshots occur and not the exact time in which each snapshot is taken, then, without loss of generality we can assume $t_i = i$ for all $i = 0, 1, \ldots$. It follows that an *execution* of an algorithm $\mathcal{A}$ from an initial configuration $R$ is a sequence of configurations $\mathbb{E} : R(0), R(1), \ldots$, where $R(0) = R$ and $R(t+1)$ is obtained from $R(t)$ by moving some robots according to the result of the `Compute` phase as implemented by $\mathcal{A}$. Moreover, given an algorithm $\mathcal{A}$, there exist many different executions from $R(0)$ depending on the activation and the movement of the robots, controlled by the adversary.

In $\mathcal{OBLOT}$ systems, one of the most studied problem is certainly the Pattern Formation (PF). Given a team of robots $R$ and a geometric pattern $F$ in terms of (a multiset of) points in the universe $\mathcal{U}$ with respect to an ideal coordinate system, the goal is to design a distributed algorithm $\mathcal{A}$ that works for each robot to guide it so that eventually all robots together form the pattern, if possible. As the global coordinate system is usually unknown to the robots, a pattern is declared formed as soon as robots are disposed *similarly* to the input pattern, that is regardless of translations, rotations, reflections, uniform scalings. Formally, $\mathcal{A}$ *forms* the pattern $F$ from a configuration $R$ if for each execution $\mathbb{E} : R = R(0), R(1), R(2), \ldots$, there exists a time instant $t' > 0$ such that $R(t')$ is similar to $F$ and $R(t) = R(t')$ for each time $t \geq t'$.

### C. SYMMETRIC CONFIGURATIONS AND SYMMETRICITY

Let $d()$ be the function computing the Euclidean distance between points in the plane, and let $\varphi$ any map from points to points in the plane: $\varphi$ is called an *isometry* if $d(\varphi(a), \varphi(b)) = d(a, b)$ for any $a, b \in \mathbb{R}^2$. Examples of isometries in the plane are rotations and reflections. An isometry $\varphi$ is a rotation if there exists a unique point $x$ such that $\varphi(x) = x$ (and $x$ is called *center of rotation*); it is a reflection if there exists a line $\ell$ such that $\varphi(x) = x$ for each point $x \in \ell$ (and $\ell$ is called *axis of symmetry*).

An isometry $\varphi$ may induce an isometry for a configuration of robots: if a robot $r$ is mapped into a robot $r'$ then $\varphi$ maps the location of $r$ into the location of $r'$. This implies that a multiplicity of $k \geq 1$ robots is always mapped into a multiplicity with the same value. The isometries for configurations are the identity, rotations, reflections and their compositions. If $R$ admits only the identity, then $R$ is said **asymmetric**, otherwise it is said **symmetric** (i.e., $R$ admits rotations or reflections). Notice that any configuration that admits a multiplicity is symmetric.

In a symmetric configuration $R$, consider any subset of pairwise symmetric robots: such robots are in fact **equivalent** as their symmetry gives no rise to any means for an algorithm to distinguish among them. This is better explained in the following remark.

*Remark 2.1: Let $R$ be a symmetric configuration. It can be observed that there exists a set of local coordinate systems for robots located in $R$ that is symmetric with respect to $c(R)$ or to the axis of symmetry. Hence, if we consider any subset $R'$ of pairwise equivalent robots in $R$, being the robots identical and homogeneous, then any move planned by any algorithm $\mathcal{A}$ for any robot in $R'$ will be applied to* all *the robots in $R'$. It follows that the adversary may force all the robots in $R'$ to perform symmetric movements and hence it results to be impossible to break the symmetry among the robots in $R'$.*

Consider now a symmetric configuration $R$ composed of asynchronous robots. According to Remark 2.1, no algorithm can avoid that two (or more) equivalent robots in $R$ start the computational cycle simultaneously. In such a case, there might occur a so called **pending move**, that is, one of the two robots performs its entire computational cycle while the other has not started or not yet finished its `Move` phase. Formally, a robot $r$ aims to perform a pending move in a configuration $R(t)$, if at time $t$ robot $r$ is active, has taken a snapshot $R(t') \neq R(t)$ with $t' < t$, and is planning to move or is moving with a non-*nil* trajectory. Clearly, any other robot $r'$ is not aware whether $r$ aims to perform a pending move, that is it cannot deduce such an information from the snapshot acquired in the `Look` phase. This fact greatly increases the difficulty to devise algorithms for Async robots in symmetric configurations. Notice that all such difficulties are removed if an algorithm is able to produce always **stationary configurations**: a configuration $R(t)$ is called *stationary* if there are no pending robots in $R(t)$. Notice that when an algorithm makes just a single robot moving then the obtained configuration is stationary by definition.

#### 1) SYMMETRICITY

Let $P$ be a set of points in the Euclidean plane, $C(P)$ be the smallest circle enclosing all the points in $P$, and $c(P)$ be the center of $C(P)$. We consider a decomposition of $P$ into regular $m$-gons with a common center, where one point forms a regular 1-gon with an arbitrary center, and two points form a regular 2-gon with the center being the midpoint. Then, we consider the maximum value of $m$. In [29], this maximum value $m$ is called the **symmetricity** of $P$ and it is denoted by $\rho(P)$. Accordingly, given a set $P$ with $n$ points, the concept of symmetricity induces a unique partition $\mathcal{P} = \{P_1, P_2, \ldots, P_k\}$ of $P$ into $k$ subsets, where each subset $P_i$ forms a $\rho$-gon and $k = n/\rho(P)$. Notice that, when $\rho(P) > 1$ the symmetricity represents the rotational symmetry of $P$, and the center of the symmetry coincides with $c(P)$. See Figure 2 for some examples about $\rho(P)$.

All the above concepts and notation can be directly applied to any configuration by considering the points in the plane occupied by the robots in $R$. Hence, we can use notation as $c(R)$ and $C(R)$ and the concept of symmetricity of $R$ as well (i.e., $\rho(R)$ corresponds to the symmetricity of the points in the plane occupied by all robots in $R$).

We can now recall the characterization about formable patterns in the $\mathcal{OBLOT}$ model when robots share a common chirality, hence according to the notion of symmetricity.

*Theorem 2.2 [29]: Let $R$ be an initial configuration of $n \geq 3$ robots and $F$ be a pattern. $F$ is formable from $R$ by* FSync *or* SSync *robots with chirality if and only if $\rho(R)$ divides $\rho(F)$.*

This result states that the solvability of the PF problem highly depends on the symmetricity of both $R$ and $F$, even for FSync robots.

Notice that a symmetric configuration $R$ with $\rho(R) = 1$ (cf. Figure 2.c) can be easily modified into an asymmetric one by just moving the central robot away from $c(R)$.

## III. THE $\mathcal{MOBLOT}$ MODEL

It is very common to find self-organizing structure in the physical world. For instance, atoms (the smallest units of the matter) first combine with other atoms to form molecules (special kind of atom compounds) and then the molecules combine with each other to form some kind of matter, like for instance a crystal. We use this *matter formation paradigm* to present the $\mathcal{MOBLOT}$ model.

In a $\mathcal{MOBLOT}$ system, the smallest units correspond to the robots of the $\mathcal{OBLOT}$ model. Each robot can be thought as an atom. As in nature there exist different types of atoms, in $\mathcal{MOBLOT}$ this can be modeled by one of the variants of $\mathcal{OBLOT}$ discussed in [34], in which robots are colored. Colors specify the type of atom the robot represents.

For instance, the robot resembling the smallest units to form water could be modeled by *white* and *red* robots corresponding to hydrogen and oxygen atoms, respectively.

In a $\mathcal{MOBLOT}$ system, the algorithmic task for robots is to form **molecules**. A molecule $\mu$ is specified by a **fixed pattern** defined with respect to the same universe $\mathcal{U}$ where the robots move. For instance the water molecule is composed by two *white* robots and one *red* robot, where the *white* robots form a 104.5° angle with the *red* robot, and each *red*-*white* pair is at distance about 0.096 *nm* (cf. Figure 3).

The minimal ball enclosing a molecule $\mu$ is denoted as $B(\mu)$, and its diameter is denoted as $diam(B(\mu))$. For the sake of simplicity, we assume $B(\mu)$ to represent the extent of molecule $\mu$. Actually, different shapes rather than just a ball might be assumed.

It is possible that in $\mathcal{U}$ there are robots that must form different molecules, hence we denote as $\mathcal{M} = \{\mu_1, \mu_2, \ldots, \mu_m\}$ the set containing all kinds of molecules that can be potentially formed. In order to make the model fair enough and as much general/weak as possible, we impose some constraints. The first one concerns a **displacement constraint** for robots in any initial configuration:

$\mathcal{C}_1$: *In any initial configuration $R$, each pair of robots is at distance greater than $D = \max\{diam(B(\mu)) \mid \mu \in \mathcal{M}\}$.*

This constraint is introduced to avoid the accidental formation of molecules in an initial configuration.

Assume that an algorithm $\mathcal{A}$ is able to move robots from an initial configuration $R$ so that at a given time $t$ some molecules in $\mathcal{M}$ are formed. In the $\mathcal{MOBLOT}$ system we assume that each robot $r$ performing the `Look` phase at time $t$ is able to detect not only all the other robots but also any formed molecule $\mu$. Notice that $r$ perceives both $B(\mu)$ and the robots inside the ball, that is $B(\mu)$ is "transparent". According to the ability of perceiving possible formed molecules, and being the molecules expressed as fixed patterns, robots are implicitly assumed to share a common **unit of length** (for measuring distances, angles, etc).

A molecule $\mu$ is formed as soon as there are robots that form the pattern describing $\mu$, but there are some **molecule formation constraints**:

$\mathcal{C}_2$: In $B(\mu)$, there are only the robots necessary to form $\mu$ suitably placed with respect to the pattern defining $\mu$;

$\mathcal{C}_3$: For each $\mu'$ already formed or that could be formed at the same time of $\mu$, then $B(\mu) \cap B(\mu') = \emptyset$;

$\mathcal{C}_4$: Assume a robot $r$ is moving along a trajectory $\tau$ toward a target $t$ and there is a position $p \neq t$ along $\tau$ such that a molecule $\mu$ is formed once $r$ in on $p$; if $\mu$ can be formed (i.e., all the previous constraints are fulfilled), then $r$ is automatically stopped at $p$ and the molecule is formed;

$\mathcal{C}_5$: As soon as a molecule $\mu$ is formed, each robot forming $\mu$ is no longer an independent computational unit (i.e., it stops executing its algorithm and acts as a part of the molecule).

*Remark 3.1: Constrains $C_2$, $C_3$ are introduced to avoid ambiguities during the formation of molecules. Let us consider a very small molecule formed by two robots at a distance $D$. As a first scenario, if three robots move synchronously so as to reach the vertices of an equilateral triangle with side $D$ at the same time, then in accordance with constraints $\mathcal{C}_2$ and $\mathcal{C}_3$, it is clear that none of the three possible molecules is formed. As a second scenario, consider two robots at distance $2D$ and a third one moving between the two at distance $D$ from both. Again no molecules are formed. However, if just one of the two external robots moves, a molecule can be formed by the other two stationary robots.*

*Remark 3.2: If robots move toward each other without forming any molecule they may collide. If that happens, no algorithm can guarantee to separate them anymore. In fact, as the robots are identical and occupy the same location, the adversary can prevent them from making different moves. Then, any algorithm must avoid collisions (i.e., undesired multiplicities) among robots.*

Once a molecule is formed, it constitutes a new computational entity with a physical dimension, i.e., it is solid. The

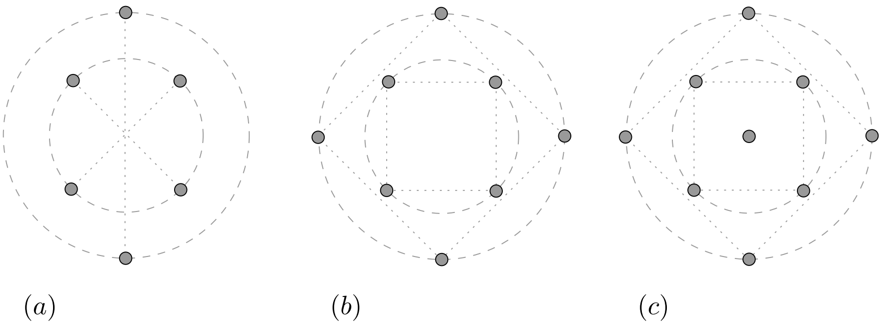

**FIGURE 2.** Examples of symmetricity of a set of points $P$. In $(a)$, $\rho(P) = 2$; in $(b)$, $\rho(P) = 4$; in $(c)$, $\rho(P) = 1$.

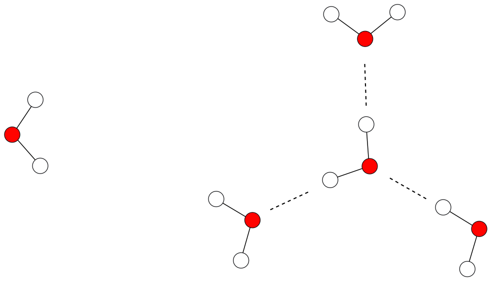

**FIGURE 3.** A classic representation of the $H_2O$ molecule.

basic properties of such new entities can still be modeled as in $\mathcal{OBLOT}$ systems (and its variants), with the main exception that a molecule not only can move along any trajectory but it also may rotate with respect to its center. [1] Being solid, any other element in $\mathcal{U}$ (robot or molecule) can touch the external surface of $B(\mu)$ – but cannot penetrate inside. This leads to a **movement constraint** for the model:

$\mathcal{C}_6$: If the trajectory of a moving robot $r$ intersects $B(\mu)$ for some molecule $\mu$, then $r$ stops its movement when it reaches the boundary of $B(\mu)$. Similarly, a moving molecule $\mu$ stops its movement as soon as it touches any robot $r$ or molecule.

Each type of molecule in $\mathcal{M}$ is provided as input to the algorithm, and the algorithm is responsible to assemble all the molecules so that a more complex structure (i.e., the **matter**) is formed. Also the matter to be formed must be given as input to robot/molecules and it can be defined in a very general way according to some **adjacency properties**. These properties just define as the molecules must be close to (or in contact with) each other, and accordingly the final form may vary. For instance, when forming the water, two molecules are connected when their external surfaces share a point $p$ and the robots in each molecule closest to $p$ have different colors; moreover, each molecule must be adjacent to another one so that the final structure must be connected. According to this definition, the molecules may form the matter by forming one among many possible patterns (by pattern we mean any configuration obtained by placing in the universe the molecules so that their positions fulfill the adjacency properties). We use symbol $\mathcal{F}$ to denote the set containing all the possible patterns describing the matter.

### A. The Matter Formation Problem

In Section II-B we have presented PF as one of the most general and studied problem with respect to the $\mathcal{OBLOT}$ model. Similarly, with respect to the defined $\mathcal{MOBLOT}$ model, the general Matter Formation (MF) problem can be defined as follows: Given a team of robots $R$, a set of different patterns $\mathcal{M}$ representing the possible molecules that can be formed, and a set $\mathcal{F}$ of possible patterns describing the matter to form by combining molecules from $\mathcal{M}$, the goal is to design a distributed algorithm $\mathcal{A}$ that works for each robot (and each composed molecule) so that eventually they form any pattern in $\mathcal{F}$, if possible. Formally, $\mathcal{A}$ *solves* MF if for each execution $\mathbb{E} : R = R(0), R(1), R(2), \ldots$, there exists a time instant $t' > 0$ such that, each robot belongs to a molecule, $R(t')$ is similar to some $F \in \mathcal{F}$ and $R(t) = R(t')$ for each time $t \geq t'$.

We have already recalled how the notion of symmetricity has been used in the literature to provide a necessary and sufficient condition for solving the PF problem with respect to FSync and SSync robots (cf. Theorem 2.2). In the remainder of this section we give a similar result by providing a necessary condition for the solvability of the MF problem. We assume that the spatial universe $\mathcal{U}$ in which robots and molecules move is confined to the Euclidean plane (even though the result can be easily extended to higher dimensions or different environments).

Let $R$, $\mathcal{M}$, and $\mathcal{F}$ be the elements forming an instance of MF, and assume the general case in which $R$ is composed by colored robots. A symmetry for $R$ is an isometry $\varphi$ for $R$ (cf. Section II-C) that preserves the color of robots. A symmetry for $F \in \mathcal{F}$ is an isometry for all the robots involved in $F$ that preserves the molecules, that is if $\{r_{i_1}, r_{i_2}, \ldots, r_{i_t}\}$ is the set of robots forming a molecule $\mu \in F$, then robots in $\{\varphi(r_{i_1}), \varphi(r_{i_2}), \ldots, \varphi(r_{i_t})\}$ form a molecule equal to $\mu$. It follows that rotational symmetries induce to the concept of symmetricity also for $F$. As an example, Figure 4 shows two patterns where the symmetricity is 1 (on the left side) and 3 (on the right side), respectively. Hence, in the remainder,

[1] These are basic assumptions for the molecules that can be suitably variated and managed with respect to the specific application. For instance, it is possible to variate the shape and/or the movement.

by $\rho(R)$ and $\rho(F)$ we denote, respectively, the symmetricity of a configuration of (possibly non-identical) robots, and of a pattern of a desired matter to be formed. We also use $\rho(\mu)$, with $\mu \in \mathcal{M}$, but in this case we simply mean the symmetricity of the set of robots that form $\mu$. Moreover, for a given pattern $F \in \mathcal{F}$, let $Mol(F)$ denote the set of molecules that form $F$. Clearly, $Mol(F) \subseteq \mathcal{M}$.

We are now ready to provide a necessary condition for solving the MF problem.

*Theorem 3.3: Let R be an initial configuration of robots, given an instance of MF, if there exists an algorithm $\mathcal{A}$ able to form the matter, i.e., a pattern $F \in \mathcal{F}$, then*

1) *$\rho(R)$ divides $\rho(F)$, or*
2) *there exists $\mu \in Mol(F)$ such that $\rho(R)$ divides $\rho(\mu)$.*

*Proof:* Assume that $\mathcal{A}$ is able to form $F$ without using the molecules (i.e., all the molecules are formed at the same time and they are arranged to form the matter; formally, there exists a time $t > 0$ such that $R(t)$ is similar to $F$ and no molecule is formed in $R(t')$ for each $t' < t$). In this case, by Theorem 2.2, we get that property (1) holds.

In what follows we assume that $\mathcal{A}$ must create and move some molecules to form $F$. We also assume $\rho(R) > 1$, otherwise both properties (1) and (2) are trivially verified. Notice that for each possible execution $\mathbb{E} : R = R(0), R(1), \dots$ of $\mathcal{A}$, according to Remark 2.1, the adversary may force $\rho(R(0))$ pairwise equivalent robots to move synchronously. Let $R(t)$, $t > 0$ be the first configuration containing molecules.

If $R(t)$ contains more than one molecule, according to the synchronous moves and to the symmetricity of $R$, then $(i)$ in $R(t)$ there are $\rho(R(0))$ molecules, $(ii)$ the molecules in $R(t)$ are all equal, and $(iii)$ $\rho(R(t)) = \rho(R(0))$. Then, from $R(t)$ on, each move planned by $\mathcal{A}$ may be forced by the adversary to maintain at least the same symmetricity $\rho(R(0))$ until $F$ is formed. Then $\rho(R(0))$ divides $\rho(F)$ and property (1) holds.

If $R(t)$ contains just one molecule $\mu$, then it must be formed around the center of the configuration. Even in this case, the adversary forces $\rho(R(0))$ equivalent robots to move synchronously, and then $\rho(\mu)$ must be a multiple of $\rho(R(0))$, i.e., property (2) holds. □

This theorem shows that if an algorithm $\mathcal{A}$ is able to form some pattern $F \in \mathcal{F}$ when the first condition does not hold, $\mathcal{A}$ must create a molecule $\mu$ in the center of the configuration and then move it to modify the symmetricity of the configuration. This resembles the mechanism used in $\mathcal{OBLOT}$ of creating an asymmetric configuration starting from a symmetric one with symmetricity 1.

Theorem 3.3 can be seen as the property of 'conservation' of the symmetry that is, the initial symmetry either is still present in the final pattern or it is encapsulated in at least one molecule composing the final pattern.

## IV. A CASE STUDY: THE HexMF PROBLEM

In this section, in order to appreciate the capabilities of the $\mathcal{MOBLOT}$ model, we define a case study by introducing and solving a specific matter formation problem. In particular, we first formalize the *Hexagonal Matter Formation* problem (HexMF for short), then we motivate its definition according to the necessary conditions given by Theorem 3.3, and finally we provide an overview of the resolution strategy.

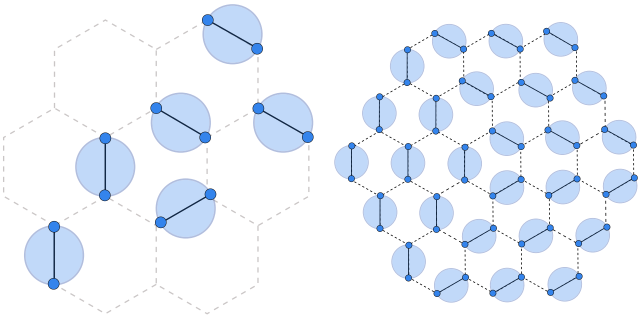

**FIGURE 4.** **(left) matter composed by 6 molecules; (right) three full levels of the matter. The hexagonal grid emphasizes the relative positions of the molecules composing the matter.**

### A. PROBLEM FORMALIZATION

HexMF is a specific variant of the general MF problem. In this problem we assume identical and Async robots endowed with chirality, that is a common handedness. We remind that robots also share a common unit of measure and are aware of the quantity $D$ necessary to form molecules. HexMF considers a single type of molecule composed of just two robots. Formally:

- $\mathcal{M} = \{\mu\}$, where $\mu$ is defined by two robots at a fixed distance $D$.

*Remark 4.1: For simplicity, in any initial configuration, the mutual distance for each pair of robots is at least 2D. This hypothesis guarantees that a robot won't make unwanted molecules while moving in a trajectory between two points.* Notice that, in the HexMF problem, not only robots but also molecules are assumed to be identical, Async and endowed with chirality.

Concerning the disposal of molecules for forming the matter, we consider a hexagonal tessellation, that is a regular tiling of the Euclidean plane in which three hexagons meet at each vertex. This tessellation induces a 'tessellation graph', that is an infinite graph embedded into the Euclidean plane induced by the vertices and sides of the hexagons forming the tessellation [35]. The tessellation graph induced by any hexagonal tessellation is simply called **hexagonal grid** and denoted as $G_H$. We consider the matter formed when each molecule is disposed on some edges of a hexagonal grid of side $D$ (see Figure 4). According to the general definition of matter given in Section III, we now specify the adjacency property for the molecules.

- Consider one hexagon of $G_H$ as the core of the matter where three non-adjacent edges of the hexagon correspond to three places where the molecules should lie. These edges form the "first level" of the matter.

Then, the six hexagons surrounding the core represent the place where the second level of the matter would be formed. That is, the non-adjacent edges of the second level, not shared with the first level and parallel to those where the first three molecules are posed, correspond to the second level of edges where molecules can be moved to form the matter, as soon as the first level is full.

The $i$-th level will be formed by the non-adjacent edges of the hexagons surrounding the $(i-1)$-th level, not shared with the $(i-1)$-th level and parallel to those where the molecules of the $(i-1)$-th level are posed. Figure 4 (right) shows three complete levels of the matter. Actually, as in Figure 4.left, the last level of the matter can be not fully occupied.

It follows that $\mathcal{F}$ contains all the patterns of molecules that satisfy the above definition of matter.

Notice that the defined matter looks like a ''polycyclic aromatic hydrocarbon'', that is a chemical compound containing only carbon and hydrogen that is composed of multiple aromatic rings [36].

To complete the definition of HexMF, we assume that any initial configuration (i.e., configurations considered as input for the problem) consists of a set $R$ of robots, with $|R| = 2m$, $m > 3$. The goal is to design a distributed algorithm $\mathcal{A}$ that works for each robot and molecule so that eventually $m$ molecules are disposed as an element $F \in \mathcal{F}$. We remark that is responsibility of $\mathcal{A}$ to coordinate the molecules (once they are formed) to find how the hexagonal grid $G_H$ is embedded into the plane. Of course, this task is made difficult by the fact that both robots and molecules do not share a common reference system.

### B. MOTIVATIONS

Problem HexMF has been defined in order to appreciate the capabilities of the $\mathcal{MOBLOT}$ model. To this end, we have defined the problem so that it constitute a minimal and yet well defined example requiring to explore all the conditions of Theorem 3.3. In fact, for each specific matter $F$ contained in $\mathcal{F}$, it can be observed that $\rho(F) = 1$ or $\rho(F) = 3$ holds. Moreover, given the unique molecule $\mu$ defined in HexMF, then $\rho(\mu) = 2$. Hence, according to the necessary conditions expressed by Theorem 3.3, we get that any initial configuration $R$ from which HexMF can be solved must guarantees that the following relationship holds:

$$1 \leq \rho(R) \leq 3 \tag{1}$$

Now, according to Theorem 3.3, assume that there exists an algorithm $\mathcal{A}$ able to solve HexMF starting from any configuration $R$ fulfilling the relationship in Equation 1. If $\rho(R) = 1$, then the first condition of Theorem 3.3 applies regardless of the matter $F$ to be formed. When $\rho(R) = 2$, the second condition of Theorem 3.3 holds, since in this case robots can form just a single occurrence of $\mu$ and then this molecule could be used to ''reduce'' the whole symmetry of the configuration to 1. Then, starting from the obtained configuration with symmetricity 1, again any pattern $F$ in $\mathcal{F}$ compatible with the size of $R$ could be formed. Note that, in $\mathcal{OBLOT}$, if $\rho(R) = 2$ then only configurations such that $\rho(F) = 2k$ can be realized.

Finally, when $\rho(R) = 3$, the first condition of Theorem 3.3 holds again. Note, however, that in this case Remark 2.1 applies, and therefore the adversary can force three robots at a time to move synchronously; of course, this necessarily leads to have three molecules formed at the beginning. It follows that a pattern $F$ with $\rho(F) = 3$ is the only pattern that $\mathcal{A}$ could create.

### C. OVERVIEW OF THE RESOLUTION STRATEGY

Here we provide a high-level description of the algorithm `FormHexMatter`, which is specifically designed for solving the HexMF problem.

Note that, even though constraint $\mathcal{C}_1$ holds according to Remark 4.1, robots can be very close each other, and this could make difficult to correctly create the molecules. So, the first step of the algorithm is to modify the configuration so that there is enough space between the robots before they start forming the matter. Such space is realized by moving almost all robots onto a circle $C^*$ having a sufficiently large radius. This implies to have a reference point (i.e., a point known to all robots) for the center of $C^*$. To define this point, one or three molecules (it depends on $\rho(R)$) are preliminarily formed around $c(R)$. Once these initial molecules are formed, the reference point for $C^*$ is given by the ''center'' of the formed molecules. By keeping these initial molecules still, all the remaining robots can reach $C^*$. The diameter of $C^*$ is defined so that all the robots can be disposed on it by defining a rotational symmetry equal to that of the molecules already formed.

Once $C^*$ is formed, its center - $c(C^*)$ - defines a reference point for creating (i.e., initializing) the matter. In particular, the previously formed molecules could be disposed by using $c(C^*)$ to define a unique embedding on the plane of the hexagonal grid $G_H$. Nonetheless, as we will show in the following, for the sake of correctness of the algorithm, it is required that before initializing the matter, some molecules must be formed on $C^*$. Consequently, once these additional molecules are formed on $C^*$, the internal molecules are moved on the first level of the hexagonal grid to initialize the matter; $c(C^*)$ is chosen as a reference for the center of the hexagonal grid. After the matter is initialized, the algorithm proceeds by repeating the following two steps: first, molecules on $C^*$ are correctly moved and added to the formed matter; then, new molecules are formed on $C^*$.

## V. THE RESOLUTION ALGORITHM

In this section we provide all the details concerning the provided resolution algorithm. In particular, we first give some general notation and data structures, then we describe how the methodology proposed in [30] is used to break down the general problem into a set of well-defined tasks where each task can be performed by robots/molecules, and finally we provide the formalization of `FormHexMatter`.

### A. GENERAL NOTATION

Here we summarize general concepts and notation used by `FormHexMatter`. Additional and specific notation will be introduced later.

Given any configuration $R$, the algorithm often uses circles on the plane, mainly centered in $c(R)$. As an example, we have already introduced $C^*$ as the circle in which all the robots, except those used for creating the initial molecules, are moved to have enough space.

In addition to $C^*$, other circles are defined. Notice that for any defined circle $C$, we use $r(C)$ to denotes its radius. Consider any configuration $R$ composed of robots located in distinct points of the plane. By $C_1^R, C_2^R, C_3^R, \ldots$ we denote all the circles centered on $c(R)$ such that on each of them it is located at least one robot of $R$. Of course, such circles generate a partition of $R$ where each set of the partition contains all the robots located on the same circle. Such circles are ordered according to their radius: given two circles $C_i^R$ and $C_j^R$, $i < j$ if and only if $r(C_i^R) < r(C_j^R)$. The algorithm uses an additional and special circle: it is denoted by $C_m$ and it is defined as the circle of radius $D/2$ centered in $c(R)$. Each half-line starting from $c(R)$ is called *ray* of $C(R)$.

Sometimes, the algorithm needs to order $n$ robots in $R$ according to their distance from $c(R)$. Of course, this generates a partial order, but when we say *"the distance from* $c(R)$ *identifies exactly* $k$ *robots"*, for any $1 \leq k < n$, we mean that such distance allows us to order the robots as follows: $r_{i_1} \leq r_{i_2} \leq \ldots \leq r_{i_k} < r_{i_{k+1}} \leq r_{i_{k+2}} \leq \ldots \leq r_{i_n}$.

Concerning molecules and matter, we denote by $Mol$ the set containing all the molecules formed in a configuration and by $Mat \subseteq Mol$ the subset of molecules forming the matter. Notice that, according to the $\mathcal{MOBLOT}$ model, robots can directly detect $Mol$ during the `Look` phase, whereas $Mat$ depends on the resolution algorithm.

The subsequent paragraph introduces further notation referred to the data acquired by robots during the `Look` phase.

### B. ROBOTS' VIEW

In this section we formalize the concept of *view of a robot* belonging to a configuration considered as input for the HexMF problem. It is a data structure used by `FormHexMatter` in which each robot encodes all the information acquired during the `Look` phase. Of course, such information refer to the configuration perceived with respect to its own LCS. Actually, sometimes a robot is asked to evaluate also what would be the view of other robots, hence it is convenient that the view does not depend on the current LCS, as this might be completely different from cycle to cycle and from robot to robot. Hence, unless further knowledge is provided to the robots, the view should exploit only the information that all robots can equally perceive, like those concerning relative distances and angles among robots' positions. It follows that in general, in a symmetric configuration there are robots with the same view.

Given two distinct points $u$ and $v$ in the Euclidean plane, let $line(u, v)$ denote the straight line passing through these points, and let $(u, v)$ ($[u, v]$, resp.) denote the open (closed, resp.) segment containing all points in $line(u, v)$ that lie between $u$ and $v$. The half-line starting at point $u$ (but excluding the point $u$) and passing through $v$ is denoted by $hline(u, v)$. We denote by $\sphericalangle(u, c, v)$ the angle centered in $c$ obtained by rotating clockwise $hline(c, u)$ until overlapping $hline(c, v)$. The angle $\sphericalangle(u, c, v)$ is measured from $u$ to $v$ in clockwise direction and the measure is always meant as positive.

Let $P$ be a generic set of points not including $c = c(P)$. For $p \in P$, we denote by $V(p)$ the view of $P$ computed from $p$. This is a sequence of pairs (angle, distance) defined as follows: first $(0, d(c, p))$ then, in order from the farthest to the closest point to $c$, all pairs $(0, d(c, p'))$ for any $p' \neq p$ in $hline(c, p)$, and successively all pairs $(\sphericalangle(p, c, p'), d(c, p'))$ arising from all other rays processed in clockwise order and points $p'$ from the farthest to the closest ones to $c$, for each ray. Strings associated to all points in $P$ can be ordered lexicographically. If $p = c(P)$ then $p$ is said the point in $P$ of *minimum view*, otherwise any $p = argmin\{V(p') : p' \in P\}$ is said of **minimum view** in $P$.

These definitions naturally extend to any configuration $R$ of robots and to any set of robots forming pattern $F \in \mathcal{F}$ as well. In fact, if $r$ is a robot in $R$, by $V(r)$ we mean the view obtained from the point in which $r$ is located. Moreover, observe that, as we are dealing with robots endowed with chirality, the clockwise direction used in the definition of the view is well-defined.

It follows that, in any symmetric configuration $R$, symmetric robots have the same view. Whereas, if $R$ is asymmetric, each robot can be associated with a unique view.

### C. DESCRIPTION OF THE ALGORITHM

The algorithm has been designed according to the methodology proposed in [30]. In the following, we describe how that methodology allowed us to break down the general problem into a set of well-defined tasks where each task can be performed by robots/molecules.

In general, each single robot or molecule has rather weak capabilities with respect to the general problem it is asked to solve along with other robots/molecules (we recall that such units of computation have no direct means of communication). For this reason, any resolution algorithm should be based on a preliminary decomposition approach: the problem should be divided into a set of sub-problems so that each sub-problem is simple enough to be thought of as a "task" to be performed by (a subset of) robots/molecules. This subdivision could require different steps before obtaining the definition of such simple tasks, thus generating a sort of hierarchical structure.

Following this approach and according to the overview of the strategy described in Sec. IV-C, HexMF is ini-

**TABLE 1. The hierarchical decomposition of** HexMF **into tasks.**

| *problem* | *sub-problem* | | *task* |
|---|---|---|---|
| HexMF | *FIM* | $FIM_1$ | $T_1$ |
| | | $FIM_2$ | $T_2$ |
| | *MRA* | | $T_3$ |
| | *FM1* | | $T_4$ |
| | *IM* | $IM_1$ | $T_5$ |
| | | $IM_2$ | $T_6$ |
| | *FM2* | | $T_7$ |
| | *MM* | | $T_8$ |
| | *MD* | | $T_9$ |

tially divided into the following sub-problems and tasks (cf. Table 1):

- *Formation of the Initial Molecules*, shortly denoted as *FIM*. It refers to the problem of forming the first molecules that will define the center of $C^*$. It is associated to two distinct tasks, according to the symmetricity of the configuration. In particular, Task $T_1$ is applied when the symmetricity is either 1 or 2 (sub-problem $FIM_1$), whereas $T_2$ is applied when the symmetricity is 3 (sub-problem $FIM_2$).
- *Move Robots Away*, shortly denoted as *MRA*. It corresponds to a single task $T_3$ which is responsible for moving all robots (not forming the initial molecules) on $C^*$ in order to make enough space for creating the requested matter.
- *Forming Molecules (auxiliary case)*, shortly denoted as *FM1*. It corresponds to a single task $T_4$ devoted to create some molecules on $C^*$ before the "matter initialization" (which is realized by using the initial molecules located on the center of $C^*$). It is considered as an auxiliary task exploited to guarantee the correct evolution of the algorithm from $T_3$ to the subsequent tasks that complete the matter formation. In fact, the formation of the matter without creating the molecules handled by $T_4$ may result in a modification of the definition of $C^*$.
- *Initialization of the Matter*, *IM* for short. This sub-problem is devoted to start the composition of the matter by opportuney moving the molecules formed during Tasks $T_1$ or $T_2$. According whether there are one or three molecules inside $C^*$, this sub-problem is subdivided into Task $T_5$ or Task $T_6$, respectively.
- *Forming Molecules*, *FM2* for short. In this sub-problem new molecules on $C^*$ are formed (that later on will be moved to make the matter growing). It is associated to a single Task $T_7$.
- *Moving Molecules*, *MM* for short. This sub-problem concerns the movement of all the molecules formed on $C^*$ (by means of tasks $T_4$ or $T_7$) to grow the matter. It is associated to a single Task $T_8$.
- *Matter Done*, *MD* for short. It refers to the problem of letting molecules to detect the matter has been formed, hence no more movements are required. Task $T_9$ is designed for this purpose.

In the remainder of the section we provide details for each designed task. In particular, we highlight the computation made by robots/molecules and formalize the outcoming moves.

### 1) TASK $T_1$: FORMATION OF ONE INITIAL MOLECULE

During task $T_1$, two robots denoted as $r_1$ and $r_2$ are identified and moved to form an initial molecule close to the center $c(R)$. Such robots move differently according whether $c(R)$ is occupied or not. The resulting move is denoted as $m_1$ and formalized as follows.

- **Move** $m_1$:
  - If there is a robot $r$ on $c(R)$, then $r$ radially moves toward $C_m$ in any direction that does not meet any robot (we recall that $C_m$ is the circle of radius $D/2$ centered in $c(R)$).
  - If $c(R)$ is not occupied, two robots $r_1$ and $r_2$ are selected and moved in order to form an initial molecule.

    Let $r_1$ be the robot closest to $c(R)$ (of minimum view in case of ties), $\ell_1$ be the line passing through $r_1$ and $c(R)$, $\ell_2$ be the line orthogonal to $\ell_1$ and passing through $c(R)$, $\mathcal{H}_1$ and $\mathcal{H}_2$ be the half-planes defined by $\ell_2$ with $r_1$ contained in $\mathcal{H}_1$. Associate to each robot a pair $(d_1, d_2)$, where $d_1$ is the distance from $c(R)$ and $d_2$ is the distance form $\ell_1$, and consider such pairs as lexicographically ordered. Now, consider the set $S$ containing all robots different from $r_1$ and with minimum associated pair $(d_1, d_2)$. If $S$ has robots in $\mathcal{H}_2$, then select $r_2$ as the robot in $\mathcal{H}_2$ with minimum view, otherwise $r_2$ is the robot in $\mathcal{H}_1$ with minimum view.

    The selected robots $r_1$ and $r_2$ move as follows: if $r_1$ is not on $C_m$, it moves radially on $C_m$, when $r_1$ is on $C_m$, $r_2$ moves radially toward $C_m$. Since $r_1$ and $r_2$ reduce their distance while moving toward $C_m$ eventually a molecule with robots $r_1$ and $r_2$ is formed. The distance $2D$ between each robot in any initial configuration guarantees there is no chance to form any other molecule except the one between $r_1$ and $r_2$ while they move.

Notice that, according to the definition of $r_1$ and $r_2$, when the symmetricity of the initial configuration is 2, then $r_1$ and $r_2$ result to be antipodal (as in Figure 5.left). In this case, the two robots are equivalent and they move concurrently toward each other (it can be observed that the molecule is finally formed even in case of asynchronous moves).

### 2) TASK $T_2$: FORMATION OF THREE INITIAL MOLECULES

During task $T_2$, three robots denoted as $r_1$, $r_2$, and $r_3$ are identified and moved toward other three robots denoted as $r'_1$, $r'_2$, and $r'_3$ in order to form three initial and distinct molecules. The moving robots are selected among the most internal

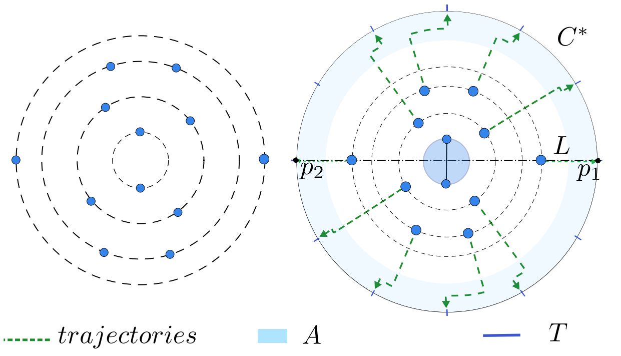

**FIGURE 5.** Configurations belonging to tasks $T_1$ (left) and $T_3$ (right). Since $R$ contains 12 robots, in the configuration belonging to $T_3$ the radius of $C^*$ should be of at least $12D$; notice that in the figure the relative distances have not been maintained in favor of the readability.

robots of $C(R)$, so the resulting molecules will be close to the center $c(R)$.

Two perform this task on an initial configuration $R$, robots must recognize that $\rho(R) = 3$. Unfortunately, due to possible asynchronous movements, before the requested molecules are formed, some intermediate configurations with symmetricity different from three can be created and hence observed by some robots. This observation imposes to provide robots with some *pre-conditions* that allows them to correctly recognize not only the initial but also each possible intermediate configuration created during the task execution.

Since the algorithm considers two different strategies for creating the molecules, two different Boolean pre-conditions are defined.

iM3$'$: It is considered true when all the following properties hold (cf. Figure 6):

a) the distance from $c(R)$ identifies exactly $r_1, r_2, r_3$;
b) if $R' = R \setminus \{r_1, r_2, r_3\}$, then $\rho(R') = 3x$, $x > 0$;
c) at least one robot among $r_1, r_2, r_3$ is not part of a molecule;
d) the rays passing through $r_1$, $r_2$, $r_3$, resp., and rotating clockwise, meet three robots $r'_1, r'_2, r'_3$ on $C_1^{R'}$ at $120^o$ each other;
e) $r_1, r_2$, and $r_3$ are on the same circle *or* their projections on $C_1^{R'}$ coincide with $r'_1, r'_2, r'_3$, respectively.

iM3$''$: It is considered true when all the following properties hold (cf. Figure 7):

a) $C_1^R$ contains more than three robots;
b) there exist $r_1, r_2, r_3$ on $C_1^R$ such that: their distance to the next (clockwise) robots is minimum and their rotation toward the next (clockwise) robots generates a configuration $R'$ with $\rho(R') = 3$;
c) at least one robot among $r_1, r_2, r_3$ is not part of a molecule.

The resulting move is denoted as $m_2$ and formalized as follows.

- **Move** $m_2$:
  - If iM3$'$ holds, robots $r_1$, $r_2$, and $r_3$ first rotate clockwise along $C_1^R$ without ever reaching distance $D$ from the next robot, until they are all aligned with $r'_1$, $r'_2$ and $r'_3$ lying on $C_1^{R'}$. If they reach such an alignment without creating molecules, then they move radially toward $r'_1, r'_2$ and $r'_3$, resp., until forming three molecules.
  - If iM3$''$ holds, then robots $r_1$, $r_2$, and $r_3$ rotate clockwise along the circle, until three molecules are formed.

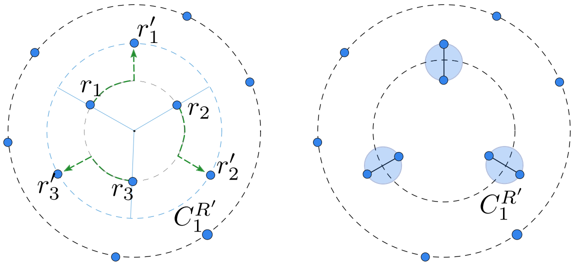

**FIGURE 6.** (left) configuration in $T_2$ where pre-condition iM3$'$ holds; (right) configuration obtained at the end of $T_2$ and belonging to $T_3$.

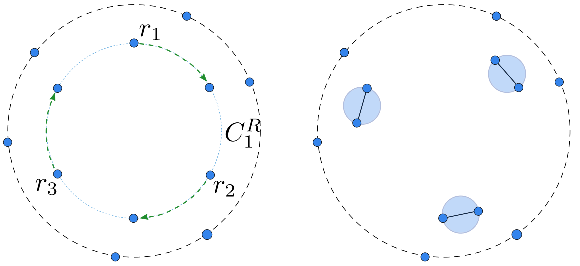

**FIGURE 7.** (left) configuration in $T_2$ where pre-condition iM3$''$ holds; (right) the configuration obtained at the end of $T_2$ and belonging to $T_3$.

Note that when applying the move, the same pre-condition between iM3$'$ and iM3$''$ that was true at the beginning of the task remains valid until the task is completed.

### 3) TASK $T_3$: MOVING ROBOTS AWAY

The only molecule formed in $T_1$ or the three molecules formed in $T_2$ are used in this task to detect a center from which $C^*$ is identified. Let $R$ be a configuration obtained after task $T_1$ or $T_2$. Circle $C^*$ is defined as follows:

- In case only one molecule $\mu$ is formed in $R$, then $C^*$ is the circle with the same center as the ball $B(\mu)$, including all the robots, with radius given by the integer

$$\rho_1 = \min\{d \mid d \geq 2mD \ \wedge \ d \ \textit{is a multiple of}\ 2m\}$$

  admitting an annulus $A$ delimited by $C^*$ and by a circle of radius $r(C^*) - 3D$ where at most one robot resides.
- In case three molecules are formed and are included in a minimum enclosing circle $C$ of radius $x$, then $C^*$ is the circle with the same center as $C$, including all the robots, with radius given by the integer

$$\rho_3 = \min\{d \mid d \geq 2mD + x \ \wedge \ d \ \textit{is a multiple of}\ 2m\}$$

admitting an annulus $A$ delimited by $C^*$ and by a circle of radius $r(C^*) - 3D$ where at most three robots reside, one for each sector of $C^*$ (the partition into sectors of the area enclosed by $C^*$ is defined later in this section, just before the formal definition of move $m_3$).

The annulus used in this definition can be observed in Figure 5.right.

The aim of this task is to move all the robots not forming molecules so that the following property holds:

FarC: All robots (excluding those forming molecules) are correctly positioned on $C^*$.

Task $T_3$ is characterized by the following pre-condition.

cM: It is considered true when all the following properties hold:

a) $\neg$iM3$'$
b) $\neg$iM3$''$
c) $|Mol| = 1$ *or* ($|Mol| = 3$ *and* $\rho(Mol) = 3$)
d) $\neg$FarC

In particular, Properties cM.0a and cM.0b ensure that the initial molecules have been already formed, Property cM.0c assures that the formed molecules have the right symmetricity, and Property cM.0d states that the task is not yet completed.

In order to move the robots on $C^*$, we need to define a suitable set $T$ of target points on $C^*$. Notice that Figure 5.right provides an example of the following concepts and notation necessary to define $T$. When $|Mol| = 1$, let $L$ be the line passing through $c(C^*)$ and orthogonal to the segment between the two robots forming the molecule, and let $P = \{p_1, p_2\}$ with $p_1$ and $p_2$ being the intersections of $L$ with $C^*$. When $|Mol| = 3$ instead, let $L_1, L_2$ and $L_3$ be the radii of $C^*$ passing through the center of each molecule, then $P = \{p_1, p_2, p_3\}$ with $p_1$, $p_2$ and $p_3$ being the intersections of $L_1$, $L_2$ and $L_3$, resp., with $C^*$. The set $T$ is defined by all the points at a distance multiple of $\pi r(C^*)/m$ from points in $P$ in the clockwise direction on $C^*$. Being $C^*$ of radius multiple of $2m$, the points of $T$ are $2m$, including those in $P$, and are equally distributed on $C^*$.

The area enclosed by $C^*$ can be partitioned into $\rho(Mol)$ sectors as follows: when $\rho(Mol) = 3$ the sectors are defined by the radii $L_1, L_2$ and $L_3$; when $\rho(Mol) = 2$ they are defined by the half-lines forming $L$; finally, when $\rho(Mol) = 1$, the whole area inside $C^*$ forms just one sector.

The resulting move to lead all the robots on the target in $T$ is denoted as $m_3$ and formalized as follows.

- **Move** $m_3$:
  - Robots are moved on $C^*$ so as to not create undesired molecules. For each sector, and in a coordinated 3-steps way, the robot furthest from $c(C^*)$ is first moved radially until distance $\frac{3}{2}D$ from $C^*$ (that is in the exact middle of $A$), then it rotates clockwise until being on the radius of $C^*$ passing through the *first unoccupied target*, and finally moves radially to the target.

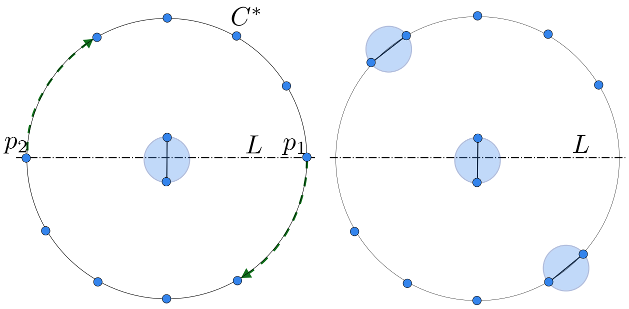

FIGURE 8. Configurations in tasks $T_4$ (left) and $T_5$ (right). Note that in $T_4$, the robots that move are those on $p_1$ and $p_2$.

Note that there might be at most three robots moving concurrently. The use of annulus $A$ is to be sure that the moving robots do not create molecules accidentally while moving. In fact the width of $A$ is $3D$ and robots move in the middle of $A$, that is at distance at least $1.5D$ from any other robot.

### 4) TASK $T_4$: FORMING MOLECULES-AUXILIARY CASE

The task is needed when $T_3$ is finished and the matter can be formed (by means of the subsequent tasks). In particular, $T_4$ can be thought as an auxiliary task exploited to guarantee the evolution of the system from $T_3$ to $T_5$ or $T_6$. In fact, the formation of the matter without creating the molecules handled by $T_4$ may result in a modification of the definition of $C^*$.

We now introduce some additional notation. Let $Mol'$ be the set of molecules inside $C^*$. We recall that in the description of task $T_3$ there are defined points $p_1$ and $p_2$ on $C^*$ when $|Mol'| = 1$, whereas such points became $p_1, p_2$, and $p_3$ when $|Mol'| = 3$. Now, if $|Mol'| = 1$, let $X = \{r_1, r_2\}$ be the first two robots that are met from $p_1$ and $p_2$, respectively, in the clockwise direction. Conversely, if $|Mol'| = 3$, let $X = \{r_1, r_2, r_3\}$ be the first three robots that are met from $p_1$, $p_2$, and $p_3$, respectively, in the clockwise direction. Once $X$ is defined, let $R^+ = R \setminus X$. Figure 8.left shows a configuration processed by Task $T_4$

This notation can now be used to define the pre-condition that characterizes task $T_4$.

nM1: It is considered true when all the following properties hold:

a) $|Mat| = 0$
b) $|Mol'| = 1$ *or* $|Mol'| = 3$
c) FarC
d) the number of molecules on $C^*$ is less than $\rho(R^+)$

The move performed in this task is denoted as $m_4$ and formalized as follows.

- **Move** $m_4$:
  - If $|X| = 3$ or ($|X| = 2$ and $\rho(R^+) = 2$), then all robots in $X$ rotate clockwise, otherwise among $r_1$ and $r_2$ the farthest from $L$ rotates clockwise.

According to this move, one, two or three molecules are formed on $C^*$ depending on the possible initial

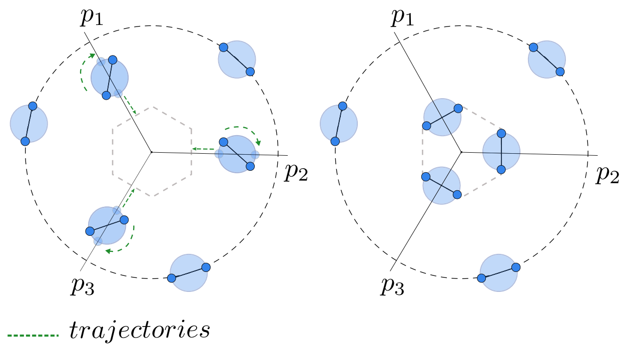

**FIGURE 9.** Movement of molecules in Task $T_6$.

symmetry deduced from $\rho(R^+)$. Such molecules along with the positioning of the other robots on $C^*$ allow the internal molecules to move to create the core of the matter made by either one or three molecules. Figure 8.right shows a configuration obtained at the end of Task $T_4$ when $\rho(R) = 2$.

### 5) TASKS $T_5$: ONE MOLECULE INITIALIZES THE MATTER

This task is devoted to start the "initialization of the matter" by correctly positioning the unique molecule formed in task $T_1$. In other words, an embedding of the hexagonal grid $G_H$ into the plane is defined, and the unique molecule located close to the center of $C^*$ is moved on an edge of the first level of $G_H$. Notice that in the previous tasks, $C^*$ was commonly identified by robots by using the internal molecules formed during $T_1$ or $T_2$. Now, since we move the internal molecule, here $C^*$ is recognized as *the circle containing all the robots not forming molecules and two robots forming one molecule and with radius equal to* $\rho_1$ (for $\rho_1$ see Section V-C3). Of course, with such a definition, internal molecules can freely move without changing the identification of $C^*$.

A preliminary embedding of $G_H$ is given by matching its center with the center of $C^*$. The exact embedding will be defined once the molecule $\mu$ is moved to a position consistent with the first level of $G_H$. Figure 8.right provides an example of a configuration where this task must be applied.

This task is applied only when the following pre-condition holds.

nM2: It is considered true when all the following properties hold:

a) $|Mat| = 0$
b) FarC
c) $2 \leq |Mol| \leq 3$: 1 or 2 molecules are on $C^*$ and 1 internal
d) the number of molecules on $C^*$ no less than $\rho(R)$

Notice that all these properties remain valid during the movement of $\mu$, while the first one becomes false as soon as $\mu$ reaches the target. The move performed in this task is denoted as $m_5$ and formalized as follows.

- **Move** $m_5$:
  - The unique internal molecule $\mu$ radially moves along $L$ until reaching a position consistent with the first level of the defined embedding of $G_H$.

Note that if the initial configuration was admitting symmetricity equal to 2, after $T_5$ the configuration becomes asymmetric. This symmetry breaking is impossible in the $\mathcal{OBLOT}$ model.

### 6) TASKS $T_6$: THREE MOLECULE INITIALIZE THE MATTER

This task is similar to $T_5$, but here three molecules instead of one are used to "initialize the matter". As in the previous task, $C^*$ is recognized as *the circle containing all the robots not forming molecules along with six robots forming three molecules, and with radius equal to* $\rho_3$ (for $\rho_3$ see Section V-C3). Of course, with such a definition, internal molecules can freely move without changing the identification of $C^*$. Again $G_H$ is embedded by matching its center with the center of $C^*$. The exact embedding will be defined once the moved molecules will be located close to the center of $G_H$ in a position consistent with the first level of $G_H$. Figure 9.left provides an example of a configuration where this task must be applied.

This task is applied only when the following pre-condition holds.

nM3: It is considered true when all the following properties hold:

a) $0 \leq |Mat| < 3$
b) FarC
c) $|Mol| = 6$: 3 molecules are on $C^*$ and 3 internal

Notice that all these properties remain valid during the movement of the three internal molecules, while the first one becomes false as soon all the moving molecules reach the target. The move performed in this task is denoted as $m_6$ and formalized as follows.

- **Move** $m_6$:
  - The three internal molecules first rotate clockwise with respect to their center until the rays of $C^*$ passing through their centers become orthogonal to the segments joining the two robots forming each molecule. Then, they radially move until reaching the right positioning in order to become part of the matter with respect to $c(C^*)$.

Once this task terminates, the matter is suitably initialized and the configuration admits a symmetricity of three.

### 7) TASK $T_7$: FORMING NEW MOLECULES

This task is responsible of forming new molecules on $C^*$ that later will be moved to make the matter growing. It starts when the matter is initialized by at least three molecules and no new molecules on $C^*$ exists. The difficulty here is mainly due to detecting how many molecules on $C^*$ need to be formed and selecting and moving the robots responsible for assembling the required new molecules to $C^*$.

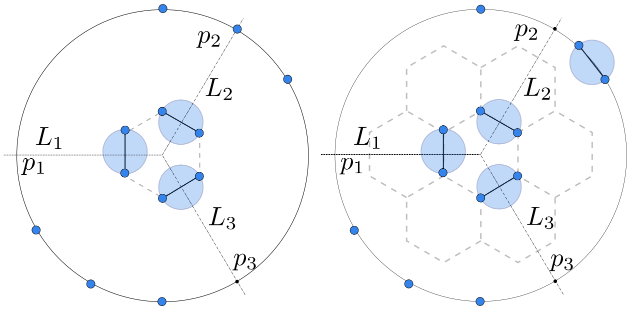

**FIGURE 10.** Configurations in tasks $T_7$ (left) and $T_8$ (right).

For this purpose, we refer again to the lines $L_1$, $L_2$, and $L_3$ introduced in the description of task $T_3$ (cf. Section V-C3), and to the points $p_1$, $p_2$, and $p_3$ induced by such lines on $C^*$. In the move planned for this task, the algorithm selects the robots $r_1$, $r_2$ and $r_3$ as the first three robots met from $p_1$, $p_2$, and $p_3$, respectively, in the clockwise direction.

The move performed in this task is denoted as $m_7$ and formalized as follows.

- **Move** $m_7$:
  - If the matter is currently composed by at least three molecules and the first three robots $r_1$, $r_2$, $r_3$ met from $p_1$, $p_2$, and $p_3$, respectively, in the clockwise direction, are distinct and $\rho(R \setminus \{r1, r2, r3\}) = 3$, then $r_1$, $r_2$ and $r_3$ rotate along $C^*$ until creating three molecules. In any other case, the robot on $C^*$ closest to the successive one in the clockwise direction, with minimum view in case ot ties, rotates along $C^*$ until forming a new molecule.

Notice that this move builds either one or three molecules on $C^*$. We conlcude by observing that this task is applied only when the following pre-condition holds.

M1: It is considered true when all the following properties hold:

a) FarC
b) $|Mat| > 0$
c) $|Mol \setminus Mat| < 3$

Notice that all these properties remain valid during the movement of the robots in $C^*$, while the third one becomes false if three molecules are formed on $C^*$. In Section VI, we will see that even though M1 may remain true once $T_7$ is over, the subsequent Task $T_8$ has a higher priority, so it will be responsible to correctly detect such an occurrence (see, e.g., Figures 10 and 11).

### 8) TASK $T_8$: MOVING MOLECULES

This task moves all the molecules – previously formed on $C^*$ by means of tasks $T_4$ or $T_7$ – to grow the matter. As in the previous task, here we still refer to the three lines $L_1$, $L_2$, and $L_3$ introduced in the description of task $T_3$, and to the points $p_1$, $p_2$, and $p_3$ induced by such lines on $C^*$. Hence, robots $r_1$, $r_2$ and $r_3$ are defined as the first three robots on $C^*$ met from $p_1$, $p_2$, and $p_3$, respectively, in the clockwise direction.

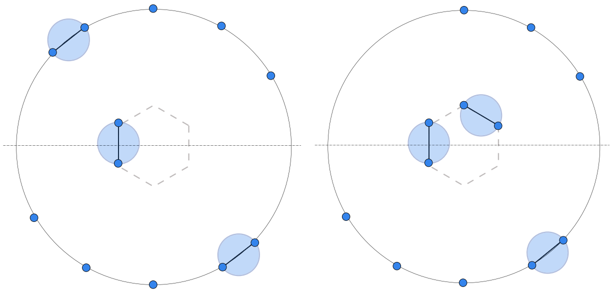

**FIGURE 11.** Two successive configurations belonging to task $T_8$.

Here the algorithm exploits an extension of the notion of symmetricity, since this notion is also applied to the matter formed so far. In particular, by $\rho(Mat)$ we denote the symmetricity of the molecules already embedded in the hexagonal grid $G_H$. For instance, in Figure 11 $\rho(Mat) = 1$, in Figure 10 $\rho(Mat) = 3$, in Figure 4 $\rho(Mat) = 1$ and $\rho(Mat) = 3$ on the left and right side, respectively.

$T_8$ is applied when the following pre-condition holds in the observed configuration.

M2: It is considered true when all the following three properties hold:

1) $|Mat| > 0$
2) The following does not hold: there are three molecules on $C^*$ and $|Mol' \setminus Mat| > 0$
3) One of the following properties holds:
   a) $|Mol \setminus Mat| > 0$ *and* $\rho(Mat) = 1$
   b) $|Mol \setminus Mat| = 1$ *and* $\rho(Mat) = 3$ *and* $\rho(R \setminus \{r_1, r_2, r_3\}) = 1$
   c) $|Mol \setminus Mat| = 3$

Concerning Property 1, it is required in order to be sure that the matter has been previously correctly initialized. Moreover, notice that it remains true during the whole task since it is not affected by the move (the matter can only increase).

Concerning Property 2, it is responsible for recognizing whether $T_6$ is over (i.e., all the three molecules inside $C^*$ have been moved in order to correctly initialize the matter) or not. In fact, as it will be clarified in Section VI, since the precondition of $T_8$ is evaluated before that of $T_6$, here it is necessary to test whether there are three molecules on $C^*$ and, at the same time, there are molecules internal to $C^*$ (i.e., molecules in $Mol'$) which are not yet correctly added to the current matter. If this condition holds, then it is clear that $T_6$ is not completed.

Let us now consider Property 3. This is due to the fact that, similarly to $T_6$, in the algorithm this pre-condition is evaluated before that of $T_7$ and hence it is responsible of recognizing whether $T_7$ is over (i.e., no more molecules must be formed on $C^*$) or not. It is based on three different sub-properties:

- Property 3a captures configurations where there are molecules on $C^*$ and $\rho(Mat) = 1$. If $T_8$ is just starting, then there are one or two molecules on $C^*$, no more molecules must be formed on $C^*$ and $T_8$ can proceed; otherwise, $\rho(Mat) = 1$ could be due to some molecule just added and also in this case it is correct to start again $T_8$ (it may happen that there are other two molecules to be moved).

- Property 3b captures configurations where there is one molecule on $C^*$ and $\rho(Mat) = 3$. Condition $\rho(R \setminus \{r_1, r_2, r_3\}) = 1$ allows the algorithm to recognize that the current configuration must not be processed by $T_7$ (no more molecules to be created on $C^*$). Hence, the unique molecule on $C^*$ must be added to the matter.
- Property 3c is simple: here there are three molecules on $C^*$, and since $T_7$ builds at most three molecules, it is evident that $T_8$ must be started.

The move performed in this task is denoted as $m_8$ and formalized as follows.

- **Move** $m_8$:
  - Each molecule moves toward the closest available position of the last level of the matter not yet filled, in the clockwise direction in case of ties, while possibly rotating with respect to its center (e.g., see Figure 11).

### 9) TASK $T_9$: MATTER DONE

It refers to the requirement of letting molecules to detect the matter has been formed, hence no more movements are required. The corresponding precondition is defined as follows:

Mf: All robots form molecules *and* matter is completed

Clearly, only *nil* movements are allowed and it is not possible to switch to any other task (e.g., see Figure 4).

## D. FORMALIZATION

As already remarked, FormHexMatter is based on a strategy that decomposes HexMF into tasks $T_1, T_2, \ldots, T_9$. Such tasks have been detailed throughout Sections V-C1–V-C9. There, for each task we have provided a detailed description for (1) the concept and notation specifically needed by the task, (2) the pre-condition that must be verified in order to accomplish the task, and (3) the move performed by robots/molecules for accomplishing the task. For the sake of convenience, Table 2 summarizes all the Boolean variables introduced in order to define the tasks' preconditions.

According to the LCM model and to the robots' obliviousness, during the Compute phase each robot must be able to recognize the task to be performed just according to the input provided to HexMF: the configuration perceived during the Look phase, the unique molecule $\mu$, and the matter defined in $\mathcal{F}$. This recognition can be performed by providing FormHexMatter with a *predicate* $P_i$ for each task $T_i$. Given the perceived configuration $R$, $\mu$ and $\mathcal{F}$, the predicate $P_i$ that results to be true reveals to robots that the corresponding task $T_i$ is the task to be performed. This approach requires that the designed predicates must guarantee some properties:

- $\mathsf{Prop}_1$*: each $P_i$ must be computable on the configuration perceived in each* Look *phase;*
- $\mathsf{Prop}_2$*: $P_i \wedge P_j =$* false*, for each $i \neq j$;*
- $\mathsf{Prop}_3$*: for each possible perceived configuration there must exists a predicate $P_i$ evaluated as true.*

Concerning how to define the predicates, we have already remarked in the previous section that each task can be accomplished only when its pre-condition is fulfilled. For the sake of simplicity, denote as $\mathtt{pre}_i$ the pre-condition defined for $T_i$ for each $2 \leq i \leq 9$ (cf. Table 2), and define $\mathtt{pre}_1 = true$. Then, $P_i$ can be formally defined as follows:

$$P_i = \mathtt{pre}_i \wedge \neg \bigvee_{j>i} \mathtt{pre}_j \tag{2}$$

Notice that $P_i$ can be true only when the pre-conditions of all the tasks with higher index than $i$ are false. This definition leads to the following lemma:

*Lemma 5.1: Predicates $P_i$ fulfill both Properties* $\mathsf{Prop}_2$ *and* $\mathsf{Prop}_3$.

*Proof:* $\mathsf{Prop}_2$ is directly implied by Equation 2. $\mathsf{Prop}_3$ is implied by pre-condition $\mathtt{pre}_1$ and predicates $P_i$. □

Table 3 formalizes the proposed algorithm: the first two (general) columns recall the hierarchical decomposition described in the previous section, the third column associates tasks names to sub-problems, and the fourth column defines precondition $\mathtt{pre}_i$ for each task $T_i$. These preconditions must be used to define each predicate $P_i$ according to Equation 2. The fifth column of Table 3 contains the name of the move defined for each task. The last column specifies the **transitions** $T_i \rightarrow T_j$ that can occur from each task $T_i$, that is any possible task $T_j$ to be performed on the obtained configurations once $T_i$ is terminated. For example, the table states that from $T_4$ only transitions $T_4 \rightarrow T_5$ and $T_4 \rightarrow T_6$ can occur. Figure 12 shows a directed graph representing all the transitions among tasks.

According to the definitions of $P_i$ given in Equation 2, in the Compute phase, each robot evaluates – with respect to the perceived configuration and the provided input $\mu$ and $\mathcal{F}$ – the preconditions starting from $P_9$ and proceeding in the reverse order until a true precondition is found. In case all predicates $P_9, P_8, \ldots, P_2$ are evaluated false, then task $T_1$, whose precondition is simply true, is performed.

### 1) EVALUATING PREDICATES: AN EXAMPLE

In this paragraph we provide an example of how robots, during the execution of FormHexMatter, detect the task to be performed. For this purpose, we consider distinct initial configurations, those shown in Figures 5.left, 6 and 7.

The initial configuration in Figure 5.left is such that $\rho(R) = 2$ and belongs to $T_1$. In fact, since there are no molecules formed, Mf, M2, M1, nM3, nM2, nM1, and cM are false, that is the configuration is not in $T_9, \ldots T_3$, respectively. Concerning iM3$''$, circle $C_1^R$ contains only two robots, hence the predicate is false. Concerning iM3$'$, the distance from $c(R)$ does not identify exactly three robots, hence the predicate is false too, that is the configuration is not in $T_2$ and then belongs to

**TABLE 2.** Summary of all the Boolean variables defined in the description of tasks $T_1, T_2, \ldots, T_9$ (cf. Sections V-C1–V-C9), used to define the tasks' preconditions $\texttt{pre}_2, \ldots, \texttt{pre}_9$.

| *prec.* | *var* | *definition* |
|---|---|---|
| $*$ | $\texttt{FarC}$ | True when all robots (excluding those forming molecules) are correctly positioned on $C^*$. |
| $\texttt{pre}_2$ | $\texttt{iM3}'$ | True when all the following properties hold:<br>1) the distance from $c(R)$ identifies exactly $r_1$, $r_2$, $r_3$;<br>2) if $R' = R \setminus \{r_1, r_2, r_3\}$, then $\rho(R') = 3x$, $x > 0$;<br>3) at least one robot among $r_1$, $r_2$, $r_3$ is not part of a molecule;<br>4) the rays passing through $r_1$, $r_2$, $r_3$, resp., and rotating clockwise, meet three robots $r'_1$, $r'_2$, $r'_3$ on $C^{R'}_1$ at $120^o$ each other;<br>5) $r_1$, $r_2$, and $r_3$ are on the same circle *or* their projections on $C^{R'}_1$ coincide with $r'_1$, $r'_2$, $r'_3$, respectively. |
| | $\texttt{iM3}''$ | True when all the following properties hold:<br>1) $C^R_1$ contains more than three robots;<br>2) there exist $r_1$, $r_2$, $r_3$ on $C^R_1$ such that: their distance to the next (clockwise) robots is minimum and their rotation toward the next (clockwise) robots generates a configuration $R'$ with $\rho(R') = 3$;<br>3) at least one robot among $r_1$, $r_2$, $r_3$ is not part of a molecule. |
| $\texttt{pre}_3$ | $\texttt{cM}$ | True when all the following properties hold:<br>1) $\neg\texttt{iM3}'$<br>2) $\neg\texttt{iM3}''$<br>3) $\lvert Mol\rvert = 1$ *or* ($\lvert Mol\rvert = 3$ *and* $\rho(Mol) = 3$)<br>4) $\neg\texttt{FarC}$ |
| $\texttt{pre}_4$ | $\texttt{nM1}$ | True when all the following properties hold:<br>1) $\lvert Mat\rvert = 0$<br>2) $\lvert Mol'\rvert = 1$ *or* $\lvert Mol'\rvert = 3$<br>3) $\texttt{FarC}$<br>4) the number of molecules on $C^*$ is less than $\rho(R^+)$ |
| $\texttt{pre}_5$ | $\texttt{nM2}$ | True when all the following properties hold:<br>1) $\lvert Mat\rvert = 0$<br>2) $\texttt{FarC}$<br>3) $2 \leq \lvert Mol\rvert \leq 3$: 1 or 2 molecules are on $C^*$ and 1 internal<br>4) the number of molecules on $C^*$ no less than $\rho(R)$ |
| $\texttt{pre}_6$ | $\texttt{nM3}$ | True when all the following properties hold:<br>1) $0 \leq \lvert Mat\rvert < 3$<br>2) $\texttt{FarC}$<br>3) $\lvert Mol\rvert = 6$: 3 molecules are on $C^*$ and 3 internal |
| $\texttt{pre}_7$ | $\texttt{M1}$ | True when all the following properties hold:<br>1) $\texttt{FarC}$<br>2) $\lvert Mat\rvert > 0$<br>3) $\lvert Mol \setminus Mat\rvert < 3$ |
| $\texttt{pre}_8$ | $\texttt{M2}$ | True when all the following properties hold:<br>1) $\lvert Mat\rvert > 0$<br>2) The following does not hold: there are three molecules on $C^*$ and $\lvert Mol' \setminus Mat\rvert > 0$<br>3) One of the following properties holds:<br>a) $\lvert Mol \setminus Mat\rvert > 0$ *and* $\rho(Mat) = 1$<br>b) $\lvert Mol \setminus Mat\rvert = 1$ *and* $\rho(Mat) = 3$ *and* $\rho(R \setminus \{r_1, r_2, r_3\}) = 1$<br>c) $\lvert Mol \setminus Mat\rvert = 3$ |
| $\texttt{pre}_9$ | $\texttt{Mf}$ | Matter formed. |

$T_1$. During $T_1$ the two most internal robots move toward each other according to $m_1$, hence the same considerations as above hold until their distance reduces to $D$ and a molecole is formed, see Figure 5.right.

The reached configuration in Figure 5.right belongs to $T_3$. In fact, $\texttt{Mf}$ is clearly false, that is the configuration is not in $T_9$. As $\texttt{FarC}$ is false, then $\texttt{M1}$, $\texttt{nM3}$, $\texttt{nM2}$, and $\texttt{nM1}$ are false, that is the configuration in not in $T_7, \ldots, T_4$, respectively. For the same reason, the embedding of $G_H$ cannot be defined and hence $|Mat| = 0$ and the configuration is not in $T_8$. Concerning $\texttt{iM3}''$, circle $C_R$ contains only two robots, hence the predicate is false. Concerning $\texttt{iM3}'$, the distance from $c(R)$ does not identify exactly three robots, hence the predicate is false too. Since $|Mol| = 1$ and not all the robots are on $C^*$ then $\texttt{cM}$ is true, that is the configuration belongs to $T_3$. In Figure 5 also the trajectories traced by the robots are shown during $T_3$ and the above Boolean values hold until the last robot reaches $C^*$. In particular, for $\texttt{iM3}'$ it is possible that at some point the distance from $c(R)$ identifies exactly three robots, however, in that case $\rho(R') < 3$.

The reached configuration in Figure 8.left belongs to $T_4$. In fact, here $Mat = \emptyset$, hence $\texttt{Mf}$, $\texttt{M2}$, $\texttt{M1}$, are false and the configuration is not in $T_9$, $T_8$, nor $T_7$. As $|Mol| = 1$ then $\texttt{nM3}$ and $\texttt{nM2}$ are false, that is the configuration does not belong to $T_6$ nor $T_5$. Since $|Mat| = 0$, $\texttt{FarC}$ is true, $|Mol'| = 1$ and there are no molecules on $C^*$ then $\texttt{nM1}$ is true and the configuration is in $T_4$. The corresponding move $m_4$ makes robots on $p_1$ and $p_2$ rotate clockwise on $C^*$ until forming two molecules. During the movements it is possible that one

**TABLE 3. Algorithm `FormHexMatter` designed to solve the HexMF problem. To task $T_i$ is associated a predicate $P_i$ as shown in Equation (2). To recognize the task to perform, each robot evaluates the predicates starting from $P_9$ and proceeding in the reverse order until a true precondition is found.**

| *problem* | *sub-problem* | | *task* | *precondition* | *move* | *transitions* |
|---|---|---|---|---|---|---|
| HexMF | *FIM* | *FIM*$_1$ | $T_1$ | *true* | $m_1$ | $T_3, T_4$ |
| | | *FIM*$_2$ | $T_2$ | $\texttt{iM3}' \vee \texttt{iM3}''$ | $m_2$ | $T_3, T_4$ |
| | *MRA* | | $T_3$ | `cM` | $m_3$ | $T_4$ |
| | *FM1* | | $T_4$ | `nM1` | $m_4$ | $T_5, T_6$ |
| | *IM* | *IM*$_1$ | $T_5$ | `nM2` | $m_5$ | $T_8$ |
| | | *IM*$_2$ | $T_6$ | `nM3` | $m_6$ | $T_8$ |
| | *FM2* | | $T_7$ | `M1` | $m_7$ | $T_8$ |
| | *MM* | | $T_8$ | `M2` | $m_8$ | $T_7, T_9$ |
| | *MD* | | $T_9$ | `Mf` | *nil* | $T_9$ |

molecule appears before the other but this does not affect the truth value of the above predicates.

The reached configuration in Figure 8.right belongs to $T_5$. In fact, here $Mat = \emptyset$, hence `Mf`, `M2`, `M1`, are false and the configuration is not in $T_9$, $T_8$, nor $T_7$. As $|Mol| = 3$ then `nM3` is false, that is the configuration does not belong to $T_6$. Since $|Mat| = 0$, `FarC` is true, $|Mol| = 3$ with 1 or 2 molecules on $C^*$ and 1 internal, and the number of molecules on $C^*$ is no less than $\rho(R)$ then `nM2` is true and the configuration is in $T_5$. Here the internal molecule radially moves along $L$ reaching the side of an hexagon of side $D$ centered in $c(C^*)$, hence making $|Mat| = 1$. During the movement, the truth value of the above predicates is not affected.

The reached configuration in Figure 11 belongs to $T_8$. In fact, here `Mf` is false, that is the configuration is not in $T_9$. Since $|Mat| > 0$, $|Mol \setminus Mat| = 2$, and $\rho(Mat) = 1$, then `M2` holds and the configuration is in $T_8$. Move $m_8$ involves the two external molecules, one by one, and leads them to be part of the matter. During the movement and after the first molecule arrives, $|Mol \setminus Mat| = 1$ and $\rho(R \setminus \{r_1, r_2, r_3\}) = 1$, hence the truth value of the above predicates is not affected.

The reached configuration in Figure 10.left belongs to $T_7$. In fact, here `Mf` is false, that is the configuration is not in $T_9$. $|Mol| \setminus |Mat| = 0$, that is `M2` is false the configuration is not in $T_8$. Since $|Mat| > 0$, $|Mol \setminus Mat| < 3$ and `FarC` is true then `M1` holds and the configuration is in $T_7$. By alternating tasks $T_7$ and $T_8$ the final configuration in Figure 4.left is achieved. According to precondition `Mf`, the final configuration belongs to Task $T_9$, where only the *nil* movement is performed. It is worth noting that according to Theorem 3.3, the reached configuration admits symmetricity 1 whereas the initial configuration of Figure 5.left has symmetricity 2. In fact, this is possible since $\rho(\mu) = 2$.

By considering the configurations shown in Figures 6 and 7 (both referring to configurations of symmetricity three) it is also possible to simulate the evaluation of all the predicates to see that in such configurations $P_9, P_8, \ldots, P_3$ are all false while $P_2$ holds (thanks to $\texttt{iM3}'$ and $\texttt{iM3}''$, respectively). During the execution of the algorithm, these configurations will be processed by tasks $T_3$ and $T_4$ until reaching the configuration shown in Figure 9. It can be observed that, in such a configuration, predicates $P_9$, $P_8$, and $P_7$ are all false while $P_6$ holds.

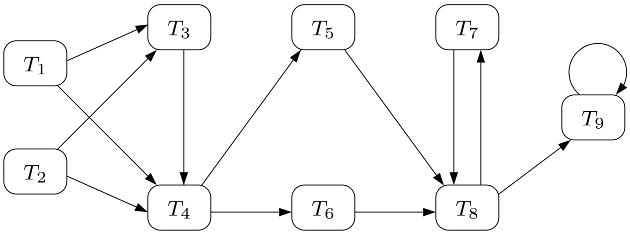

**FIGURE 12. The directed graph representing all the possible transitions among tasks.**

According to the definition of all the Boolean variables given in Sections V-C1–V-C9 and to the above examples, we can make the following remark.

*Remark 5.2: Algorithm* `FormHexMatter` *fulfills Property* Prop$_1$.

### 2) USING THE ALGORITHM IN THE COMPUTE PHASE

According to Lemma 5.1 and Remark 5.2, we get that all properties Prop$_1$, Prop$_2$, and Prop$_3$ hold. As a consequence, `FormHexMatter` can be used in the `Compute` phase as follows:

– *if any robot $r$ (or molecule $\mu$) executing algorithm* `FormHexMatter` *detects that predicate $P_i$ holds, then $r$ (or $\mu$) simply performs the move $m_i$ associated with task $T_i$.*

## VI. CORRECTNESS

In this section we formally prove that algorithm `FormHexMatter` solves the HexMF problem. According to the methodology proposed in [30], the correctness of the proposed algorithm can be obtained by proving that all the following properties hold:

$H_1$: *The algorithm never generates unsolvable configurations. According to Theorem 3.3, this implies that each configuration $R(t)$, $t > 0$, generated by the algorithm fulfills $\rho(R(t)) \leq 3$.*

$H_2$: *The movement of each robot is collision-free (cf. Remark 3.2).*

$H_3$: *For each task $T_i$, the transitions from $T_i$ to any other task are "exactly" those declared in Table 3; moreover, all such transitions lead to stationary configurations.*

$H_4$: *Each transition in Table 3 occurs after a finite number of cycles. This means that the generated configurations can remain in the same task only for a finite number of cycles.*

Since these properties must be proved for each transition/move, then in the following we provide a specific lemma for each task. Notice that Property $H_3$ does not directly implies that robots/molecules "complete" each task in a finite amount of time. In fact, there is a cycle created by transitions between tasks $T_7$ and $T_8$. Anyway, a final theorem will assess the correctness of `FormHexMatter` by making use of all the proved properties $H_1$–$H_4$ for each task and by also showing that there is a finite number of transitions between tasks $T_7$ and $T_8$.

*Lemma 6.1: Let $R$ be a stationary configuration in $T_1$. From $R$,* `FormHexMatter` *eventually leads to a stationary configuration belonging to $T_3$ or $T_4$.*

*Proof:* Let us analyze properties $H_i$, for $1 \leq i \leq 4$, separately.

$H_1$: In this task, configurations with only symmetricity of 1 or 2 are processed. Notice that in the particular case in which there is a robot in $c(R)$, that robot is moved away in order to create an asymmetric configuration. As a consequence, there are always two robots $r_1$ and $r_2$ detectable, and such robots are those closest to $c(R)$. Robots $r_1$ and $r_2$ are moved toward the circle $C_m$ so that exactly one molecule is created, eventually. Hence, the symmetricity always remains at most 2 until the end of the task.

$H_2$: There are at most two robots $r_1$ and $r_2$ moving toward $C_m$. When their reciprocal distance becomes $D$, if a molecule is formed no collision can occur. When they are at distance $D$, a molecule is not formed only if either constraint $\mathcal{C}_2$ or constraint $\mathcal{C}_3$ are not satisfied (cf. Section III). In these cases the two robots could potentially reach the same point on $C_m$ and collide. However, condition $\mathcal{C}_2$ cannot occur since there are no further robots close to $c(R)$ and the starting distance between robots is more than $2D$. Regarding condition $\mathcal{C}_3$, there should be a third robot $r_3$ at distance $D$ from either $r_1$ or $r_2$ when $d(r_1, r_2) = D$. In this case, it is not difficult to prove that $r_3$ is closer to $c(R)$ than $r_2$, and this is a contradiction for the definition of $r_1$ and $r_2$.

$H_3$: We show that each configuration generated from $R$ remains in $T_1$ until the moving robots $r_1$ and $r_2$ have reached their targets. Since $R$ belongs to $T_1$, then the precondition of $T_i$, for each $i > 1$, is false with respect to $R$. In particular, since the precondition of $T_8$ is false, then in the considered configuration $R$ there are no molecules. As a consequence, since all preconditions of $T_9, T_8, \ldots, T_3$ require formed molecules to hold, they remain false until at least a molecule if formed. Concerning $T_2$, since its precondition is false in $R$, then it remains as such during the movements of $r_1$ and $r_2$. In particular, `iM3`$''$ is false because when the two robots start moving there remain less than 3 robots on $C_1^R$, and `iM3`$''$ is incompatible with the kind of movement performed by $r_1$ and $r_2$.

We now show that when $r_1$ and $r_2$ reach their targets, a stationary configuration $R'$ belonging to either $T_3$ or $T_4$ is generated. In fact, when $r_1$ and $r_2$ reach their targets, preconditions of $T_9, T_8, \ldots, T_5$ do not hold because they require $|Mat| > 0$ or $|Mol| > 1$, against the presence of just one molecule. The membership $R'$ depends on `FarC` only: if `FarC` is false, then the obtained configuration is stationary in $T_3$ because there is only one molecule and both `iM3`$'$ and `iM3`$''$ are false for the same reasons above; when `FarC` holds, since there are no molecules on $C^*$, then $R'$ is a stationary configuration in $T_4$.

$H_4$: As long as the configuration remains in $T_1$, the distance of each moving robot from $C_m$ decreases. Hence, within a finite number of computational cycles, the robots create a molecule.

□

*Lemma 6.2: Let $R$ be a stationary configuration in $T_2$. From $R$,* `FormHexMatter` *eventually leads to a stationary configuration belonging to $T_3$ or $T_4$.*

*Proof:* Let us analyze properties $H_i$, for $1 \leq i \leq 4$, separately.

$H_1$: In this task, there are always three robots detectable, in such a way that without them the configuration admits a symmetricity of 3. The defined move guarantees that $r_1, r_2, r_3$ are the closest robots to the targets that accomplish the task. Moreover, while such robots move, they decrease their distances from their target and hence they are always distinguishable. It follows that the symmetricity never overcomes 3.

$H_2$: When `iM3`$'$ holds, $r_1$, $r_2$, and $r_3$ rotate clockwise until either they create three molecules, or they become aligned with robots $r'_1$, $r'_2$, and $r'_3$, resp., located on $C_2^R$. In this second case, they radially move toward $r'_1$, $r'_2$, and $r'_3$, again creating three molecules. Hence, no collisions are possible. When `iM3`$''$ holds, $r_1$, $r_2$, $r_3$ rotate clockwise until creating three molecules with three robots located on $C_1^R$ or $C_2^R$. Again, since the starting distance between robots is more than $2D$, by such a movement no collisions are possible. Furthermore, the moving robots cannot meet other molecules along their trajectory as the configuration did not contain any molecule when the task started.

$H_3$: We show that each configuration generated from $R$ remains in $T_2$ until all the three moving robots $r_1$, $r_2$ and $r_3$ have reached their targets. Since $R$ belongs to $T_2$, then the precondition of $T_i$, for each $i > 2$, is false with respect to $R$. In particular, since the precondition of $T_8$ is false, then there are no molecules in $R$. As a consequence, since all preconditions of $T_9, T_8, \ldots, T_3$ require the presence of molecules to hold, they remain all false until at least one molecule if formed. Since `iM3`$'$ and `iM3`$''$ remain valid until $r_1$, $r_2$, and $r_3$ are all part of a molecule, then each obtained configuration remains in $T_2$ until one molecule is formed.

Let $R'$ be any configuration observed in the interval of time in which one or two molecules are formed. During such an interval, the following properties hold in $R'$: (1) variable `FarC` is false (this can be easily observed since the center of the minimum circle enclosing all the formed molecules does not coincide with $c(R)$, as requested by the definition of $C^*$), (2) $|Mat| = 0$ (the formed molecules are not positioned as the definition of

matter requires), and (3) both `iM3`$'$ and `iM3`$''$ remain valid. As a consequence of these properties, $R'$ is evaluated as belonging to $T_2$.
Let $R''$ be any configuration observed as soon as the three moving robots reach their targets. $R''$ does not belong to $T_9$, $T_8$ and $T_7$ since otherwise $|Mat| > 0$ (and this is false since the three formed molecules are not relatively positioned as the definition of matter requires). It does not belong to $T_6$, since it requires $|Mol| = 6$ against the only three formed molecules. $R''$ is not in $T_5$, since it requires molecules formed on $C^*$ against the only three molecules formed close to $c(R)$. Finally, the membership of $R''$ depends on `FarC` only: if `FarC` is false, then $R''$ is a stationary configuration in $T_3$ because there are only three molecules and both `iM3`$'$ and `iM3`$''$ are false since $r_1$, $r_2$, and $r_3$ all belong to molecules; if `FarC` holds, since there are no molecules on $C^*$, then $R''$ is a stationary configuration in $T_4$.

$H_4$: As long as the configuration remains in $T_3$, the distance of each moving robot from its target decreases. Hence, within a finite number of computational cycles, each moving robot reaches its target.

□

*Lemma 6.3: Let R be a stationary configuration in* $T_3$. *From R,* `FormHexMatter` *eventually leads to a stationary configuration belonging to* $T_4$.

*Proof:* During task $T_3$, we observe the following properties concerning move $m_3$. In the first part, the robots move radially outward, so the mutual distances of the robots inside $C^*$ can only increase. In the second part of the movement, the robots rotate clockwise on a circle in the center of the ring but remaining within different sectors; being at a distance greater than $D$ from $C^*$ and in different sectors, these robots cannot form molecules either during rotation or during movement toward the target.

We can now analyze properties $H_i$, for $1 \leq i \leq 4$, separately.

$H_1$: As there are one or three molecules defining $c(C^*)$ and as no further molecules are created according to the above observation, then $\rho(R)$ can be at most 3.

$H_2$: Each moving robot $r$ is tracing a trajectory suitably defined to not incur in collisions nor create molecules as described in the above observation.

$H_3$: We show that each configuration generated from $R$ remains in $T_3$ until all the moving robots have reached the targets. Notice that in $R$, `pre`$_3$ = `cM` holds whereas `pre`$_i$, $i > 3$, does not hold. This implies that `FarC` is false (derived from `cM` true) and $|Mat| = 0$ (derived from `pre`$_8$ = `M2` false). Since the value of these two variables is not affected by the robots' movement, then each configuration $R'$ obtained before all robots reach the targets does not belong to $T_9, T_8, \ldots, T_4$. Moreover, since it can be easily observed that `cM` still holds in $R'$, then $R'$ remains in $T_3$.
Let $R''$ be the configuration obtained at the time in which all the moving robots reach the targets. Since the molecules in $R$ did not change their position, then still $|Mat| = 0$ in $R''$; moreover, non new molecules have been created in $R''$. This implies that $R''$ is not in $T_9, T_8, \ldots, T_5$. Finally, since `FarC` became true in $R''$, then $R''$ results to be a stationary configuration in $T_4$.

$H_4$: As long as the configuration remains in $T_3$, the distance of each moving robot from its target decreases. Hence, within a finite number of computational cycles, the robot reaches its target.

□

*Lemma 6.4: Let R be a stationary configuration in* $T_4$. *From R,* `FormHexMatter` *eventually leads to a stationary configuration belonging to* $T_5$ *or* $T_6$.

*Proof:* Let us analyze properties $H_i$, for $1 \leq i \leq 4$, separately.

$H_1$: During the robots' movement there are only one or three molecules inside $C^*$ defining $c(C^*)$, hence $\rho(R)$ can be at most 3.

$H_2$: Each moving robot $r$ is going along $C^*$ toward the next closest robot $r'$ in the clockwise direction. Hence, conditions $\mathcal{C}_2$ and $\mathcal{C}_3$ cannot occur since no robots are in between $r$ and $r'$ and once $d(r, r') = D$ a molecule is created.

$H_3$: Let $R'$ be any configuration observed during the robots' movement and before all moving robots reach their targets. Notice that in both $R$ and $R'$ we have $|Mat| = 0$ (as remarked in the proofs of the previous lemmas, the molecules inside $C^*$ are not positioned as the definition of matter requires). This implies that `Mf`, `M2` and `M1` remain false and hence $R'$ is not in $T_9, T_8, T_7$. Notice that during the movement, predicates `nM2` and `nM3` are false due to the non-consistent numbers of molecules inside and on $C^*$. Hence, $R'$ remains in $T_4$.
Let $R''$ be the configuration observed as soon as all the moving robots reach their targets. In $R''$ there can be three, two or one molecule on $C^*$. If three, it means there are three molecules inside $C^*$ and predicate `nM3` becomes true, i.e., the configuration is in $T_6$ and it is stationary. If one or two, there is only one molecule inside $C^*$ and predicate `nM2` becomes true, whereas `nM3` is false. Hence, the configuration is in $T_5$ and it is stationary.

$H_4$: As long as the configuration remains in $T_4$, the distance of each moving robot from its target decreases. Hence, within a finite number of computational cycles, each moving robot reaches its target.

□

*Lemma 6.5: Let R be a stationary configuration in* $T_5$. *From R,* `FormHexMatter` *eventually leads to a stationary configuration belonging to* $T_8$.

*Proof:* Let us analyze properties $H_i$, for $1 \leq i \leq 4$, separately.

- $H_1$: As soon as the molecule inside $C^*$ moves, $\rho(R)$ can only be equal to 1.
- $H_2$: Since there is only one molecule inside $C^*$, no collisions among robots are possible, when the molecule is moving. Similarly, no other molecules are met.
- $H_3$: Let $\mu$ be the moving molecule, and let $R'$ be any configuration observed during the movement of $\mu$ and before $\mu$ reaches its target. Of course, $|Mat| = 0$ in $R'$. Hence, both `M1` and `M2` are false in $R'$ and hence $R'$ is not in $T_9, T_8, T_7$. $R'$ is not in $T_6$ because `nM3` is false (there is only $\mu$ inside $C^*$). Since `nM2` remains true, $R'$ is in $T_5$. Let $R''$ be the configuration observed as soon as $\mu$ reaches the target. Notice that in this configuration we have $|Mat| = 1$. As a consequence, $R''$ is not in $T_9$. Notice that `M2` holds in $R''$ ($|Mat| = 1$, the matter formed does not admit a rotation, and $|Mol \setminus Mat| > 0$). Since there is at least one molecule on $C^*$, then $R''$ results to be a stationary configuration in $T_8$.
- $H_4$: As long as the configuration remains in $T_5$, the distance of the moving molecule from its target decreases. Hence, within a finite number of computational cycles, the molecule reaches its target.

□

*Lemma 6.6: Let R be a stationary configuration in $T_6$. From R,* `FormHexMatter` *eventually leads to a stationary configuration belonging to $T_8$.*

*Proof:* Let us analyze properties $H_i$, for $1 \leq i \leq 4$, separately.

- $H_1$: Since there are three molecules on $C^*$, $\rho(R)$ can only be 1 or 3.
- $H_2$: Since there are only molecules moving, collisions among robots cannot occur. Moreover, by move $m_6$, molecules always move on radial trajectories toward $c(C^*)$ without ever touching each other.
- $H_3$: Let $R'$ be any configuration observed during the molecules' movement and before they all reach their targets. $R'$ is clearly not in $T_9$ since there are still molecules to be added to the matter. Moreover, Property 2 in the definition of `M2` is false and hence $R'$ does not belong to $T_8$. Since there are three molecules on $C^*$, then $|Mol \setminus Mat| > 3$ and hence the last property of `M2` does not hold: hence, $R'$ is not in $T_7$. Since `nM3` is clearly not affected by the movements, then $R'$ remains in $T_6$.

  Let $R''$ be the configuration observed as soon as each molecule reaches its target. Again, $R''$ is not in $T_9$ since there are still molecules to be added to the matter. Notice that `M2` is true in $R''$ (in fact, $|Mol \setminus Mat| = 3$), and hence $R''$ results to be a stationary configuration in $T_8$.
- $H_4$: As long as the configuration remains in $T_6$, the distance of each moving molecule from its target decreases. Hence, within a finite number of computational cycles, each molecule reaches its target.

□

*Lemma 6.7: Let R be a configuration with $|Mat| > 0$. Then, $\rho(R)$ can be either 1 or 3.*

*Proof:* By construction, the molecules constituting the first level of the matter are three at most. If they are less than three then $\rho(R) = 1$. Else, if they are three, then $\rho(R)$ cannot be larger than three nor equal to two. □

*Lemma 6.8: Let R be a stationary configuration in $T_7$. From R,* `FormHexMatter` *eventually leads to a stationary configuration belonging to $T_8$.*

*Proof:* Let us analyze properties $H_i$, for $1 \leq i \leq 4$, separately.

- $H_1$: Follows directly from Lemma 6.7.
- $H_2$: Each moving robot $r$ is going along $C^*$ toward the next closest robot $r'$ in the clockwise direction. Hence, conditions $\mathcal{C}_2$ and $\mathcal{C}_3$ cannot occur since no robots are in between $r$ and $r'$ and once $d(r, r') = D$ a molecule is created.
- $H_3$: Let $R'$ be any configuration observed during the robots' movement and before they all reach their targets. $R'$ is clearly not in $T_9$ since there are still molecules to be added to the matter. If only one robot is allowed to move, then `M2` remains false until one molecule is created. If three robots are allowed to move, it means that the matter admits a rotation and $\rho(R' \setminus \{r_1, r_2, r_3\}) = 3$ while $|Mol \setminus Mat| < 3$, hence again `M2` is false. Summarizing, in both cases (one or three robots moving) we get that $R'$ belongs to $T_7$.

  Let $R''$ be the configuration observed as soon as each robot reaches its target. Again, $R''$ is not in $T_9$ since there are still molecules to be added to the matter. Notice that `M2` is true in $R''$. In fact, either $|Mol \setminus Mat| = 3$ or $|Mol \setminus Mat| = 1$; in the latter case either the matter does not admit a rotation or $\rho(R \setminus \{r_1, r_2, r_3\}) = 1$. Hence, $R''$ is a stationary configuration in $T_8$.
- $H_4$: As long as the configuration remains in $T_7$, the distance of each moving robot from its target decreases. Hence, within a finite number of computational cycles, the robot reaches its target.

□

*Lemma 6.9: Let R be a stationary configuration in $T_8$. From R,* `FormHexMatter` *eventually leads to a stationary configuration belonging to $T_7$ or $T_9$.*

*Proof:* Move $m_8$ aims to bring molecules formed on $C^*$ to join the matter. Hence, during this task, predicates `FarC` and $|Mat| > 0$ remain true. Let us analyze properties $H_i$, for $1 \leq i \leq 4$, separately.

- $H_1$: Follows directly from Lemma 6.7.
- $H_2$: Since there are only molecules moving, collisions among robots cannot occur. Moreover, by move $m_8$, molecules always move on free trajectories.
- $H_3$: Let $R'$ be any configuration observed during the molecules' movement and before they all reach their targets. In $R'$, `M2` remains trivially true as long as $|Mol \setminus Mat| = 3$. Otherwise `M2` remains true since, if $|Mol \setminus Mat| = 1$, then there is exactly one molecule inside $C^*$

and hence $\rho(R \setminus \{r_1, r_2, r_3\}) = 1$; if $|Mol \setminus Mat| = 2$, then $|Mat| \mod 3 = 2$, and hence the matter does not admit a rotation. It follows that $R'$ remains in $T_8$.

Let $R''$ be the configuration observed as soon as each molecule reaches its target. In $R''$, it can be easily observed `M2` becomes false (because $|Mol \setminus Mat| = 0$) and `M1` becomes true. Then, if no further robots are on $C^*$, then $R''$ is a stationary configuration in $T_9$. Otherwise, if there are still robots on $C^*$, then $R''$ is a stationary configuration in $T_7$.

$H_4$: As long as the configuration remains in $T_8$, the distance of each moving molecule from its target decreases. Hence, within a finite number of computational cycles, the molecule reaches its target.

□

*Theorem 6.10: Let $m > 3$ be an integer and let $R$ be any initial configuration composed of $|R| = 2m$ asynchronous robots moving on the plane. If robots in $R$ have pairwise distance greater than $2D$, then* `FormHexMatter` *correctly solves the HexMF problem.*

*Proof:* Lemmata 6.1-6.9 ensure that properties $H_1$, $H_2$, $H_3$, and $H_4$ hold for each task $T_1, T_2, \ldots, T_9$. Then $\rho(R(t))$, for $t > 0$, is always less than or equal to 3; the moves of the robots are all collision-free; all the transitions are those reported in Table 3 and visualized in the graph of Figure 12; and the generated configurations can remain in the same task only for a finite number of cycles. Lemmata 6.1-6.9 also show that from a given task only subsequent tasks can be reached, or `Mf` eventually holds (and hence HexMF is solved). The only exception is the cycle among tasks $T_7$ and $T_8$ (cf. Figure 12). However, in this case, at the end of $T_8$, the number of molecules composing the matter increases, and since no molecule is moved away from the matter, task $T_9$ is reached from $T_8$ after a finite number of transitions between $T_7$ and $T_8$. This formally implies that, for each initial configuration $R$ and for each execution $\mathbb{E} : R = R(0), R(1), R(2), \ldots$ of `FormHexMatter`, there exists a finite time $t' > 0$ such that $R(t')$ is similar to the matter to be formed in the HexMF problem and $R(t) = R(t')$ for each time $t \geq t'$. □

## VII. CONCLUSIVE REMARKS AND FUTURE WORK

In this paper, we have proposed $\mathcal{MOBLOT}$, a new theoretical model in the context of swarm and modular robotics that extends the well-known $\mathcal{OBLOT}$ model. $\mathcal{MOBLOT}$ concerns two levels of computational entities: robots and molecules. Robots can be very weak entities like in the $\mathcal{OBLOT}$ model; molecules are usually more complex entities with an extent. Ideally, robots and molecules are guided by two different distributed algorithms: the former is used to form molecules, the latter to manage molecules, e.g., to assemble them to obtain some complex structure (the matter). Once molecules have accomplished their first task (e.g., the matter is formed), a new task could be further approached by molecules, e.g., rearrange (self-reconfigure) their positions to get a different shape for the matter.

In order to highlight some potentials of the $\mathcal{MOBLOT}$ model, we have introduced the Matter Formation problem. We have proven that there is some necessary condition for forming the matter. According to Theorem 3.3, such a condition states that the symmetricity of the initial configuration of robots must divide either the symmetricity of some molecule, or the symmetricity of the matter to be formed. Interestingly, this implies that the matter could be formed even when the symmetricity of the input configuration is unrelated to that of the matter (in such cases, the molecules play a decisive role). In this respect, we have specialized the definition of Matter Formation to what we have called HexMF, where only one type of molecule made of just two robots is considered.

Actually, by exploiting the definition of the HexMF problem provided in the case study, we can formally prove that $\mathcal{MOBLOT}$ is an extension of $\mathcal{OBLOT}$. To this aim, we first introduce some notation. Given two robot models $\mathcal{M}_1$ and $\mathcal{M}_2$, inequality $\mathcal{M}_1 \geq \mathcal{M}_2$ means that any problem that can be solved in $\mathcal{M}_2$ is also solvable in $\mathcal{M}_1$ (i.e., $\mathcal{M}_1$ is not less powerful than $\mathcal{M}_2$). Inequality $\mathcal{M}_1 > \mathcal{M}_2$ means that $\mathcal{M}_1 \geq \mathcal{M}_2$ holds and there exists a problem that can be solved in $\mathcal{M}_1$ but not in $\mathcal{M}_2$ (i.e., $\mathcal{M}_1$ is more powerful than $\mathcal{M}_2$). We can prove the following result.

*Theorem 7.1: $\mathcal{MOBLOT} > \mathcal{OBLOT}$.*

*Proof:* We first show that $\mathcal{MOBLOT} \geq \mathcal{OBLOT}$. To this aim, observe that in case each molecule in $\mathcal{M}$ is constituted by a single robot without any extent nor extra capabilities, then $\mathcal{MOBLOT}$ reduces to $\mathcal{OBLOT}$.

We now show that $\mathcal{MOBLOT} > \mathcal{OBLOT}$. Consider HexMF, and let $\mathcal{I}$ be the set of all possible instances $(R, \mathcal{M}, \mathcal{F})$ for HexMF fulfilling the conditions in Theorem 3.3.

Let us analyze whether there exists an algorithm $\mathcal{A}'$ defined according to the $\mathcal{OBLOT}$ model and able to solve HexMF for each instance in $\mathcal{I}$. In other words, we are asking whether $\mathcal{A}'$ is able to form some pattern in $\mathcal{F}$ by moving robots from $R$ *always assuming them as single units, that is without exploiting the capabilities of molecules in* $\mathcal{M}$. It is well known from [29] that "a pattern $F$ cannot be formed from a configuration $R$ when $\rho(R)$ does not divide $\rho(F)$". Since there are configurations $(R, \mathcal{M}, \mathcal{F}) \in \mathcal{I}$ such that $\rho(R) = 2$ and $\rho(F) \in \{1, 3\}$ for each $F \in \mathcal{F}$, then $\mathcal{A}'$ cannot solve the problem. Hence, HexMF cannot be solved in the $\mathcal{OBLOT}$ model.

Consider now `FormHexMatter`, the algorithm described in Section V and formalized in Section VI. Theorem 6.10 shows that `FormHexMatter` is able to solve HexMF for each instance in $\mathcal{I}$ by suitably exploiting the molecules' capabilities. In fact, the molecules have the ability to break the symmetry of the original configuration, provided that at least one molecule shows this symmetry. In particular, when the first property of Theorem 3.3 does not hold but the second does, algorithm `FormHexMatter` creates a single molecule close to the center of the configuration and after moves it to break the initial symmetry. □

There are many directions for future research in the proposed model. The most natural one is to investigate about a (complete) characterization on the solvability of the general Matter Formation problem according to the assumed capabilities for robots and molecules. Others concern the formalization of possible self-reconfigurable matter problems, as well as problems related to the matter movement. Completely different tasks can also be thought, perhaps by adding further capabilities to the molecules.

Note that the proposed model can also be used when robots and molecules are limited to move in a grid-based terrain where movements are only allowed along the grid lines and only toward an adjacent grid point in each step, as it often happens in real life robotic navigation systems. From an algorithmic point of view, the restrictions imposed by the environment on movements should make it more difficult to solve problems that were easy in the continuous environment. It would also be interesting to investigate whether and how the conditions given in Theorem 3.3 are affected by the underlying grid-based environment. A first work in this respect can be found in [37] for robots moving in square grids.

## ACKNOWLEDGMENT

A earlier version of this paper was presented at the 20th International Conference on Autonomous Agents and Multiagent Systems (AAMAS '21) [DOI: 10.5555/3463952.3463998].

## REFERENCES

[1] S. Cicerone, A. Di Fonso, G. Di Stefano, and A. Navarra, "MOBLOT: Molecular oblivious robots," in *Proc. 20th Int. Conf. Auton. Agents MultiAgent Syst.*, F. Dignum, A. Lomuscio, U. Endriss, and A. Nowé, Eds., May 2021, pp. 350–358.

[2] A. L. Christensen, "Self-reconfigurable robots—An introduction," *Artif. Life*, vol. 18, no. 2, pp. 237–240, 2012, doi: 10.1162/artl_r_00061.

[3] M. Amir and A. M. Bruckstein, "Minimizing travel in the uniform dispersal problem for robotic sensors," in *Proc. 18th Int. Conf. Auton. Agents MultiAgent Syst. (AAMAS)*, 2019, pp. 113–121.

[4] E. Sahin, "Swarm robotics: From sources of inspiration to domains of application," in *Proc. Int. Workshop Swarm Robot.*, in Lecture Notes in Computer Science, vol. 3342, E. Sahin and W. M. Spears, Eds. Santa Monica, CA, USA: Springer, Jul. 2004, pp. 10–20, doi: 10.1007/978-3-540-30552-1_2.

[5] M. Yim, W.-M. Shen, B. Salemi, D. Rus, M. Moll, H. Lipson, E. Klavins, and G. Chirikjian, "Modular self-reconfigurable robot systems," *IEEE Robot. Autom. Mag.*, vol. 14, no. 1, pp. 43–52, Mar. 2007.

[6] T. Fukuda and Y. Kawauchi, "Cellular robotic system (CEBOT) as one of the realization of self-organizing intelligent universal manipulator," in *Proc. Int. Conf. Robot. Autom.*, Cincinnati, OH, USA, May 1990, pp. 662–667, doi: 10.1109/ROBOT.1990.126059.

[7] H. Ahmadzadeh, E. Masehian, and M. Asadpour, "Modular robotic systems: Characteristics and applications," *J. Intell. Robot. Syst.*, vol. 81, nos. 3–4, pp. 317–357, Mar. 2016, doi: 10.1007/s10846-015-0237-8.

[8] T. Tucci, B. Piranda, and J. Bourgeois, "A distributed self-assembly planning algorithm for modular robots," in *Proc. 17th Int. Conf. Auton. Agents Multiagent Syst.*, E. André, S. Koenig, M. Dastani, and G. Sukthankar, Eds. Richland, SC, USA, 2018, pp. 550–558.

[9] M. Rubenstein, C. Ahler, and R. Nagpal, "Kilobot: A low cost scalable robot system for collective behaviors," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2012, pp. 3293–3298, doi: 10.1109/ICRA.2012.6224638.

[10] Z. Derakhshandeh, R. Gmyr, T. Strothmann, A. R. Bazzi, W. A. Richa, and C. Scheideler, "Leader election and shape formation with self-organizing programmable matter," in *Proc. 21st Int. Conf. Comput. Mol. Program.*, in Lecture Notes in Computer Science, vol. 9211. Cambridge, MA, USA, 2015, pp. 117–132, doi: 10.1007/978-3-319-21999-8_8.

[11] J. J. Daymude, K. Hinnenthal, A. W. Richa, and C. Scheideler, "Computing by programmable particles," in *Distributed Computing by Mobile Entities* (Lecture Notes in Computer Science), vol. 11340, P. Flocchini, G. Prencipe, and N. Santoro, Eds. Cham, Switzerland: Springer, 2019, pp. 615–681, doi: 10.1007/978-3-030-11072-7_22.

[12] G. D'Angelo, M. D'Emidio, S. Das, A. Navarra, and G. Prencipe, "Asynchronous silent programmable matter achieves leader election and compaction," *IEEE Access*, vol. 8, pp. 207619–207634, 2020.

[13] G. D'Angelo, M. D'Emidio, S. Das, A. Navarra, and G. Prencipe, "Leader election and compaction for asynchronous silent programmable matter," in *Proc. 19th Int. Conf. Auton. Agents MultiAgent Syst.*, 2020, pp. 276–284.

[14] A. Navarra, G. Prencipe, S. Bonini, and M. Tracolli, "Scattering with programmable matter," in *Proc. 37th Int. Conf. Adv. Inf. Netw. Appl. (AINA)*, 2023.

[15] Y. Kim, Y. Katayama, and K. Wada, "Pairbot: A novel model for autonomous mobile robot systems consisting of paired robots," 2020, *arXiv:2009.14426*.

[16] P. Flocchini, G. Prencipe, and N. Santoro, "Moving and computing models: Robots," in *Distributed Computing by Mobile Entities* (Lecture Notes in Computer Science), vol. 11340, P. Flocchini, G. Prencipe, N. Santoro, Eds. Cham, Switzerland: Springer, 2019, pp. 3–14, doi: 10.1007/978-3-030-11072-7_1.

[17] M. Salman, D. Garzón Ramos, K. Hasselmann, and M. Birattari, "Phormica: Photochromic pheromone release and detection system for stigmergic coordination in robot swarms," *Frontiers Robot. AI*, vol. 7, Dec. 2020, Art. no. 591402, doi: 10.3389/frobt.2020.591402.

[18] A. Pásztor, "Gathering simulation of real robot swarm," *Tehnicki Vjesnik*, vol. 21, no. 5, pp. 1073–1080, 2014.

[19] S. Das, R. Focardi, F. L. Luccio, E. Markou, and M. Squarcina, "Gathering of robots in a ring with mobile faults," *Theor. Comput. Sci.*, vol. 764, pp. 42–60, Apr. 2019, doi: 10.1016/j.tcs.2018.05.002.

[20] G. D'Angelo, A. Navarra, and N. Nisse, "A unified approach for gathering and exclusive searching on rings under weak assumptions," *Distrib. Comput.*, vol. 30, no. 1, pp. 17–48, Feb. 2017.

[21] S. Cicerone, G. Di Stefano, and A. Navarra, "Gathering robots in graphs: The central role of synchronicity," *Theor. Comput. Sci.*, vol. 849, pp. 99–120, Jan. 2021.

[22] S. Cicerone, "Breaking symmetries on tessellation graphs via asynchronous robots," in *Proc. Int. Conf. Inf. Commun. Technol. Competitive Strategies*, vol. 2756, G. Cordasco, L. Gargano, and A. Rescigno, Eds., 2020, pp. 122–136.

[23] S. Cicerone, A. Di Fonso, G. Di Stefano, and A. Navarra, "Arbitrary pattern formation on infinite regular tessellation graphs," *Theor. Comput. Sci.*, vol. 942, pp. 1–20, Jan. 2023.

[24] S. Cicerone, G. D. Stefano, and A. Navarra, "Solving the pattern formation by mobile robots with chirality," *IEEE Access*, vol. 9, pp. 88177–88204, 2021.

[25] S. Cicerone, G. Di Stefano, and A. Navarra, "Asynchronous arbitrary pattern formation: The effects of a rigorous approach," *Distrib. Comput.*, vol. 32, no. 2, pp. 91–132, Apr. 2019.

[26] S. Cicerone, G. Di Stefano, and A. Navarra, "Embedded pattern formation by asynchronous robots without chirality," *Distrib. Comput.*, vol. 32, no. 4, pp. 291–315, Aug. 2019.

[27] P. Flocchini, G. Prencipe, N. Santoro, and P. Widmayer, "Arbitrary pattern formation by asynchronous, anonymous, oblivious robots," *Theor. Comput. Sci.*, vol. 407, nos. 1–3, pp. 412–447, Nov. 2008.

[28] P. Flocchini, G. Prencipe, and N. Santoro, "Distributed computing by mobile entities," in *Distributed Computing by Mobile Entities Current Research in Moving and Computing* (Lecture Notes in Computer Science), vol. 11340. Cham, Switzerland: Springer, 2019, pp. 37–62, doi: 10.1007/978-3-030-11072-7_3.

[29] I. Suzuki and M. Yamashita, "Distributed anonymous mobile robots: Formation of geometric patterns," *SIAM J. Comput.*, vol. 28, no. 4, pp. 1347–1363, Jan. 1999.

[30] S. Cicerone, G. Di Stefano, and A. Navarra, "A structured methodology for designing distributed algorithms for mobile entities," *Inf. Sci.*, vol. 574, pp. 111–132, Oct. 2021, doi: 10.1016/j.ins.2021.05.043.

[31] C. Agathangelou, C. Georgiou, and M. Mavronicolas, "A distributed algorithm for gathering many fat mobile robots in the plane," in *Proc. ACM Symp. Princ. Distrib. Comput.*, Jul. 2013, pp. 250–259.

[32] J. Czyzowicz, L. Gasieniec, and A. Pelc, "Gathering few fat mobile robots in the plane," *Theor. Comput. Sci.*, vol. 410, nos. 6–7, pp. 481–499, Feb. 2009.

[33] S. G. Chaudhuri and K. Mukhopadhyaya, "Leader election and gathering for asynchronous fat robots without common chirality," *J. Discrete Algorithms*, vol. 33, pp. 171–192, Jul. 2015.

[34] Z. Liu, Y. Yamauchi, S. Kijima, and M. Yamashita, "Team assembling problem for asynchronous heterogeneous mobile robots," *Theor. Comput. Sci.*, vol. 721, pp. 27–41, Apr. 2018, doi: 10.1016/j.tcs.2018.01.009.

[35] E. J. Ionascu, "Half domination arrangements in regular and semi-regular tessellation type graphs," 2012, *arXiv:1201.4624*.

[36] *Polycyclic Aromatic Hydrocarbon*. Accessed: Jul. 4, 2021. [Online]. Available: https://en.wikipedia.org/wiki/Polycyclic_aromatic_hydrocarbon

[37] S. Cicerone, A. Di Fonso, G. Di Stefano, and A. Navarra, "Molecular robots with chirality on grids," in *Proc. 18th Int. Symp. Algorithmics Wireless Netw.*, in Lecture Notes in Computer Science, Germany, U.K.: Springer, Sep. 2022, pp. 45–59.

**SERAFINO CICERONE** received the degree in computer science from the University of L'Aquila, in 1993, and the Ph.D. degree from the Sapienza University of Rome, in 1998. He is currently an Associate Professor with the Department of Information Engineering, Computer Science and Mathematics, University of L'Aquila. His research interests include specification, design, verification, and implementation of efficient algorithms. His specific areas of interests include distributed algorithms, combinatorial optimization, algorithm engineering, algorithmic graph theory, spatial, and geometric data.

**ALESSIA DI FONSO** is currently pursuing the Ph.D. degree in engineering and information science with the Department of Information Engineering, Computer Science and Mathematics, University of L'Aquila. Her research interests include graph algorithms and distributed algorithms for mobile robots.

**GABRIELE DI STEFANO** received the Ph.D. degree from the Sapienza University of Rome, in 1992.

He is currently a Full Professor with the Department of Information Engineering, Computer Science and Mathematics, University of L'Aquila. He is the (co)author of more than 120 publications in journals and international conferences. He had key-participations in several EU funded projects. Among them: MILORD (AIM 2024), COLUMBUS (IST 2001-38314), AMORE (HPRN-CT-1999-00104), ARRIVAL (IST FP6-021235-2), and GEOSAFE (H2020-691161). His current research interests include network algorithms, combinatorial optimization, algorithmic graph theory, and distributed computing.

**ALFREDO NAVARRA** received the Ph.D. degree in computer science from the Sapienza University of Rome, Italy, in 2004. He is currently a Full Professor with the Department of Mathematics and Computer Science, University of Perugia, Italy. Before joining the University of Perugia, in 2007, he has been with various international research institutes, such as the INRIA of Sophia Antipolis, France; the Department of Computer Science, University of L'Aquila, Italy; and the LaBRI, University of Bordeaux, France. His research interests include algorithms, computational complexity, distributed computing, and networking.

Open Access funding provided by 'Università degli Studi dell'Aquila' within the CRUI CARE Agreement