Symmetry Preservation in Swarms of Oblivious Robots with Limited Visibility

Raphael Gerlach
Paderborn University, Institute of Mathematics, Germany

Sören von der Gracht
Paderborn University, Institute of Mathematics, Germany

Christopher Hahn
University of Hamburg, Department of Informatics, Germany

Jonas Harbig
Paderborn University, Heinz Nixdorf Institute, Germany

Peter Kling
University of Hamburg, Department of Informatics, Germany

Abstract

In the *general pattern formation* (GPF) problem, a swarm of simple autonomous, disoriented robots must form a given pattern. The robots' simplicity imply a strong limitation: When the initial configuration is rotationally symmetric, only patterns with a similar symmetry can be formed [34]. The only known algorithm to form large patterns with limited visibility and without memory requires the robots to start in a *near-gathering* (a swarm of constant diameter) [22]. However, not only do we not know any near-gathering algorithm guaranteed to preserve symmetry but most natural gathering strategies trivially increase symmetries [5].

Thus, we study near-gathering without changing the swarm's rotational symmetry for disoriented, oblivious robots with limited visibility (the $\mathcal{OBLOT}$-model, see [15]). We introduce a technique based on the theory of dynamical systems to analyze how a given algorithm affects symmetry and provide sufficient conditions for symmetry preservation. Until now, it was unknown whether the considered $\mathcal{OBLOT}$-model allows for *any* non-trivial algorithm that always preserves symmetry. Our first result shows that a variant of Go-To-The-Average *always* preserves symmetry but may sometimes lead to multiple, unconnected near-gathering clusters. Our second result is a symmetry-preserving near-gathering algorithm that works on swarms with a convex boundary (the outer boundary of the unit disc graph) and without "holes" (circles of diameter 1 inside the boundary without any robots).

2012 ACM Subject Classification Theory of computation → Self-organization

Keywords and phrases Swarm Algorithm, Swarm Robots, Distributed Algorithm, Pattern Formation, Limited Visibility, Oblivious

Digital Object Identifier 10.4230/LIPIcs.OPODIS.2024.23

Related Version *Full Version*: https://arxiv.org/abs/2409.19277 [19]

Funding *Sören von der Gracht and Jonas Harbig*: This work was partially supported by the German Research Foundation (DFG) under the project numbers ME 872/14-1 & 453112019.

1 Introduction

The study of large *robot swarms* that consist of simple (and thus cheap) mobile, autonomous robots has grown into an important and active research area in recent years. For example, when used for exploration or rescue missions in hazardous environments (like the deep sea or outer space), such swarms can be more robust and economical compared to a small group of more capable but expensive high-end units [24]. At a completely different scale, precision medicine explores how to use swarms of nanobots (with inherently limited capabilities due to their small size) to, e.g., deliver drugs in a more targeted manner [31].

A key question in this area is to what extent one can make up for the robots' lack of individual capabilities by clever coordination, possibly even reaching emergent behavior of the swarm as a whole. Examples for the high potential of such an approach can be readily found in nature, where ant colonies or bee hives show a remarkable degree of complexity built on rather simple interaction rules between a vast number of very simple entities [30, 25]. Similar to many other branches of distributed computing, breaking or avoiding *symmetries* presents a major challenge for such systems.

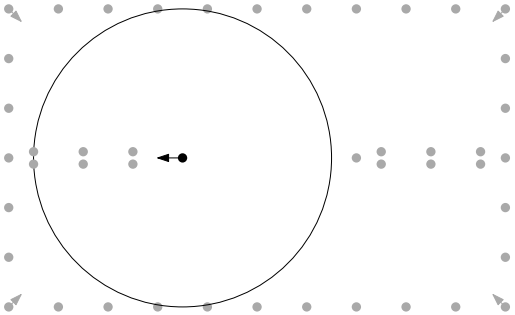
General Pattern Formation in the $\mathcal{OBLOT}$ Model. Our work studies the role of symmetries in solving one of the most important problems in the theory of swarm robots, the *general pattern formation* (GPF) problem. Here, a swarm of $n \in \mathbb{N}$ autonomous, mobile robots must form a pattern $P \subseteq \mathbb{R}^2$ of $|P| = n$ coordinates (in an arbitrary rotation/translation). We assume the well-known $\mathcal{OBLOT}$ model [15] for deterministic point robots that are *oblivious* (have no memory), *anonymous* (have no identities and cannot be distinguished), *homogeneous* (execute the same algorithm), and *disoriented* (perceive their surroundings in coordinate systems that may be arbitrarily rotated/translated compared to other robots). We also assume that the robots have a limited, constant *viewing range*, such that they cannot observe anything beyond that range. Moreover, we assume the unit disc graph to be connected, while the viewing range may be different from the unit distance. Robots act in discrete time steps in which each of them performs a full LCM-cycle consisting of a Look- (observe surroundings), a Compute- (calculate target), and a Move- (move to target) phase. This is often referred to as the *fully-synchronous* ($\mathcal{F}$SYNC) time model (other models allow the phases of different robots to be arbitrarily interleaved). We consider only *rigid* movements (robots always reach their designated target).

Symmetries in GPF. Symmetries play a key role in GPF, since – even if the robots are not oblivious and have an unlimited viewing range – a swarm can only form patterns whose *symmetricity* (a measure of a pattern's symmetry, see Definition 3.1) is a multiple of the swarm's initial symmetricity [32, 18]. In the case of oblivious robots with an unlimited viewing range, even in an asynchronous setting, this *symmetricity condition* characterizes *exactly* those patterns that can be formed [34, 18]. A good overview of the symmetricity condition's role in various synchronous and asynchronous settings is given in [36].

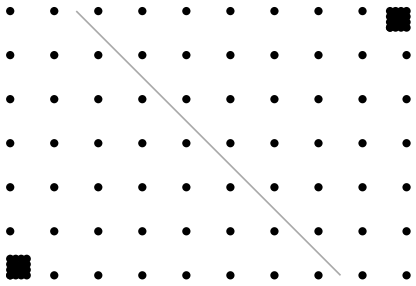
If the viewing range is limited but robots are not oblivious, [37] gave a protocol that forms a scaled-down version of a target pattern P adhering to the symmetricity condition. That protocol first forms a *Near-Gathering* (a formation whose diameter is at most the robots' viewing range) and then uses the de facto global view to form a small version of P with the algorithm from [34]. Because of the symmetry constrains of the GPF algorithm, the Near-Gathering must preserve the symmetry, i.e. have the same symmetricity as the initial configuration. The authors show that if movements are *non-rigid* (i.e., an adversary can stop robots during their movement), robots must be non-oblivious to preserve symmetricity.

Under the light of these results, it remains an open question whether oblivious robots with a limited viewing range can form arbitrary connected patterns P (ideally in its original size) under the above symmetricity condition. Very recently, [22] introduced a protocol that achieves this *if the robots start from a Near-Gathering*. This opens the possibility to solve GPF from *any* formation by first (similiar to [37]) forming a Near-Gathering and then (once the robots observe $|P|$ peers in the Near-Gathering) form P using the protocol from [22].

Unfortunately, while there are algorithms for local, oblivious robots that achieve a Near-



(a) Example with hole. Algorithm 3: Near-Gathering but introduces new symmetries.[2] Go-To-The-Average: Near-Gathering with symmetry preservation.[3]



(b) Example with convex boundary and without hole. Algorithm 3: Near-Gathering with symmetry preservation. Go-To-The-Average: Keeps symmetry but looses connectivity (swarm splits at the gray line).[4]

Figure 1 Visualization of example configurations where either algorithm fails.

Gathering [5, 28], all of them might (and typically do) *increase* the initial symmetry. As a result, it might be impossible to form the target pattern from the resulting Near-Gathering, even if the swarm's original symmetry met the symmetricity condition. The authors of [22] leave as a central open question *"whether there is a suitable Near-Gathering protocol that preserves the initial symmetricity"* for oblivious robots with *rigid* movement.[1] If the answer were positive, one could show that also for oblivious robots with a limited viewing range in the synchronous model, the patterns that can be formed are characterized by the symmetricity condition.

Our Contribution. In this work, we initiate the systematical study of when and how a swarm of oblivious robots with limited viewing range can perform global tasks like Near-Gathering without increasing the swarm's initial symmetricity. We derive a mathematical framework based on methods from the theory of dynamical systems. In particular, we formulate the following simple but useful theorem (see Section 3) that provides sufficient properties for a given swarm protocol (represented by its evolution function $F\colon \mathbb{R}^{2n} \to \mathbb{R}^{2n}$ that describes the configuration $\mathbf{z}^+ \coloneqq F(\mathbf{z})$ after one protocol step on a given configuration $\mathbf{z} \in \mathbb{R}^{2n}$ of n robots in the Euclidean plane) to preserve symmetricity. For the precise definitions of the used terms, like a configuration $\mathbf{z}$'s symmetries and (local) invertibility, we refer to Section 3.

▶ **Theorem 1.1.** *Consider the dynamics of an arbitrary swarm protocol with evolution function $F\colon \mathbb{R}^{2n} \to \mathbb{R}^{2n}$. Assume that F is (locally) invertible. Then, any configuration $\mathbf{z} \in \mathbb{R}^{2n}$ and its successor configuration $\mathbf{z}^+ \coloneqq F(\mathbf{z})$ have the same symmetries $G_{\mathbf{z}^+} = G_{\mathbf{z}}$.*

[1] Note that [37] already proved that this is impossible for oblivious robots with *non-rigid* movement. They use the robots' memory to remember their movements, which allows them to maintain their original position and, thus, symmetricity (symmetric robots in the Near-Gathering can be distinguished by their original position).

[2] The black robot does not observe robots on the outer rectangle (circle depicts viewing range). It assumes it is a boundary-robot and moves accordingly. All other robots in the inside of the rectangle do not move, because they see enough robots on the outer rectangle to detect, that they are inner-robots.

[3] We simulated this example with the shown viewing range.

[4] The distance between grid points is $1/\sqrt{2}$ and the viewing range $2+\sqrt{2}$. We simulated it with 400 robots at each cluster.

The framework of dynamical systems provides a clean mathematical basis to formulate the symmetries of a given configuration and how they are affected by a protocol step. To prove the usefulness of our framework, we provide two example protocols that, under certain conditions, achieve a Near-Gathering without increasing the swarms symmetricity.

The first protocol (see Section 4) is a variant of the well known Go-To-The-Average protocol. This protocol is known to *not* always preserve the swarm's initial connectivity (see Figure 1b for an example), but if it does, it leads to Near-Gathering.[5] Our framework easily implies that it preserves the swarm's initial symmetry.

Our second protocol (see Section 5) is an adaption of the well known Go-to-the-Middle strategy. It restricts movements to robots close to the swarm's boundary and coordinates the movement of nearby robots to ensure that no symmetries are created. While this protocol always preserves connectivity and leads to Near-Gathering (Remark 5.22), it is not always symmetry preserving (see Figure 1a for an example). We can prove that its evolution function is locally invertible (and, thus, the protocol symmetry preserving) for configurations that contain no "holes" and are convex (we formalize this requirement in Section 5).

Outline. This work is outlined as follows: First, further related work is reviewed in Section 2. In Section 3, we give a formal description of the problem and formulate a sufficient condition for the preservation of symmetries based on the theory of dynamical systems. We present and analyze our two algorithms in Section 4 and Section 5. Finally, we discuss limitations and possible generalizations of both algorithms in Section 6.

2 Further Related Work

The general pattern formation problem has been considered in numerous further settings and variants whose focus or assumptions differ from our setting and the work described in the Section 1. For example, [2] considers pattern formation for oblivious robots with unlimited visibility on an infinite grid, showing that an initially asymmetric swarm can form any pattern (basically thanks to the grid enabling partial axis agreement). The authors of [3] assume unlimited but obstructed visibility (i.e., robots can obstruct each others view) with partial axis agreement and luminous robots (that can communicate via a constant number of lights). The role of partial axis agreement and disorientation was also studied in [17, 8], proving several possibility and impossibility results depending on how much of their coordinate systems the robots agree. In [10], the authors considered when oblivious robots with unlimited visibility can form (cyclic) sequences of patterns. Note that the impossibility results in most of these works typically rely to a large degree on assuming an asynchronous model (where the robots' LCM-cycles may be arbitrarily interleaved) and/or non-rigid movements (where an adversary can stop robots mid-motion). For further results on general pattern formation we refer to the survey [35].

There are also several results on forming specific patterns, like a point [11, 5, 4] (gathering), an arbitrarily tight near-gathering [9, 26] (convergence), and a uniform circle [33, 27, 16]. A rather up-to-date and good overview can be found in [14].

In dynamical systems theory, much of the related literature considers *consensus* or *synchronization* problems that share characteristics with gathering or near-gathering, e.g., by identifying robot positions with "opinions" the agents want to find a consensus on. A good overview over this research branch can be found in [29] or in the slightly more recent

[5] Robots on the swarm's convex hull move inwards in every step.

surveys [6, 12]. Extensions of these methods to general pattern formation (in our sense) have been proposed as well (see e.g., [1]). However, in this area the focus is typically on time-continuous systems—in the form of differential equations—and the models are not as strictly restricted as necessary in our context, e.g., allowing global communication range.

3 Preliminaries & Notation

This section introduces some formal notation and definitions we use throughout the rest of the paper. In particular, we use some tools from the theory of dynamical systems to formulate sufficient conditions for swarm protocols that preserve symmetries.

Basic Notions & Notation. We consider a swarm of robots in the $\mathcal{OBLOT}$ model as described in Section 1. When discussing the configuration (state) of the swarm in some round $t \in \mathbb{N}$, we typically take the perspective of an external observer who passively observes the motion of the robot swarm. In particular, given a swarm of n robots we specify the robots' positions $z_1^t = (x_1^t, y_1^t), \dots, z_n^t = (x_n^t, y_n^t)$ in round t in an arbitrary *global coordinate system* for $\mathbb{R}^2$. The *configuration* of the entire swarm in round t can then be expressed as the (column) vector $\mathbf{z}^t = (z_i^t)_{i=1}^n \in (\mathbb{R}^2)^n \equiv \mathbb{R}^{2n}$. During the analysis, we sometimes identify a robot with its position (e.g., saying that robot z_i moves to its target point).

The *unit disc graph* of a configuration $\mathbf{z}$ is an undirected graph $G = (V, E)$ with $V = \{1, \dots, n\}$ and $\{i, j\} \in E$ if and only if $\|z_i - z_j\| \leq 1$. A *Near-Gathering* is a configuration whose unit disc graph is a clique (i.e., any two robots see each other). We say a configuration is *connected* if its unit disc graph is connected. The viewing range of the robots can differ from the unit distance. In Section 4 we assume a viewing range of 1 for the first protocol while the second protocol in Section 5 needs a viewing range of $2 + \sqrt{2}$.

The protocol definition given in Section 5 relies on the *Connectivity-Boundary* of the swarm's configuration, which we define as the outer boundary of the (in general non-convex) polygon enclosed by the unit disc graph. In our analysis we sometimes identify the Connectivity-Boundary with the set of robots that lie on it. A figure showing the Connectivity-Boundary can be foud in Appendix C.1.

3.1 Characterizing Protocols via their Evolution Function

We now formalize how to model the effect of a given protocol in the language of dynamical systems, which we will use to analyze the protocol's effect on the swarm's symmetry.

Remember that robots move autonomously in the plane in discrete synchronous rounds according to the same protocol. Mathematically, from round t to $t+1$, the i-th robot moves from its old position z_i^t to its new position z_i^{t+1}. This new position z_i^{t+1} depends on its own as well as on (potentially) *all other robots'* positions (e.g., if the robot currently sees the whole swarm and the protocol states to move towards the center of gravity of visible robots). In the most general formulation, we can describe such a dynamics as

$$z_i^{t+1} = f(z_i^t; \mathbf{z}^t) \quad i = 1, \dots, n \tag{1}$$

for some function $f \colon \mathbb{R}^2 \times \mathbb{R}^{2n} \to \mathbb{R}^2$. Note that all robots execute the same protocol, so the function f does not depend on i. However, since all robots have their own coordinate system they must to be able to distinguish their own position from all other positions in the swarm. This is reflected in the explicit first argument given to f.

With this, we can describe the evolution of the entire configuration as

$$\mathbf{z}^{t+1} = F(\mathbf{z}^t) = \begin{pmatrix} f(z_1^t; \mathbf{z}^t) \\ \vdots \\ f(z_n^t; \mathbf{z}^t) \end{pmatrix}. \tag{2}$$

We refer to F as the *evolution function* of the protocol.

Whenever the specific value of t is irrelevant (e.g., when we investigate an arbitrary round) we drop the superscript and abbreviate $z_i^+ = f(z_i; \mathbf{z})$ where $z_i^+ \in \mathbb{R}^2$ indicates the "next" position of robot i. Similarly, we use the notation $\mathbf{z}^+ = F(\mathbf{z})$ to indicate the "next" configuration if the evolution function F is applied to some configuration $\mathbf{z}$.

3.2 Preserving Symmetries via Local Invertibility

As explained in the introduction, we seek Near-Gathering strategies that do not increase the swarm's *symmetricity* (because of the *symmetricity condition* for pattern formation, see [18, Theorem 1]). The symmetricity measures the rotational symmetricity of a finite set $P \subseteq \mathbb{R}^2$ (in our case the set of robot positions) and can be defined as follows:

▶ **Definition 3.1** (Symmetricity [18]). *Consider a finite set $P \subseteq \mathbb{R}^2$ whose smallest enclosing circle is centered at $c \in \mathbb{R}^2$. An* m-regular *partition of P is a partition of P into $k = |P|/m$ regular m-gons with common center c. The* symmetricity *of P is defined as* $\mathrm{sym}(P) := \max\{\, m \in \mathbb{N} \mid$ *there is an m-regular partition of P* $\}$.

Since here we consider the configuration typically in the global coordinate system of the external observer (which can be chosen arbitrarily), we assume (without loss of generality) that the swarm's center c is the origin.

▶ Remark 3.2. In Definition 3.1, a single point is considered a 1-gon with an arbitrary center. Thus, any P has a 1-regular partition. If the center c is an element of P, then $\mathrm{sym}(P) = 1$. However, in terms of symmetry preservation it is more relevant to consider the actual rotational symmetry of P which is not influenced by any element at the center. In our paper you may as well ignore any element at center c while computing the symmetricity of P.

We now formalize the notion of a *symmetry* in such a way that we can apply the theory of dynamical systems to argue how a protocol's evolution function influences those symmetries. For a swarm of symmetricity $m > 1$ there are exactly m rotations $\rho\colon \mathbb{R}^2 \to \mathbb{R}^2$ around the origin (center) under which the *set* of robot positions is invariant (i.e., $\{\, \rho(z_1), \ldots, \rho(z_n) \,\} = \{\, z_1, \ldots, z_n \,\}$). We represent configurations as tuples instead of sets, which is more typical in the context of dynamical systems. Thus, we define a symmetry of a configuration $\mathbf{z} \in (\mathbb{R}^2)^n$ as follows (using a permutation to "relabel" the tuple suitably after the rotation).

▶ **Definition 3.3** (Symmetry of a Configuration 1). *Consider a configuration $\mathbf{z} = (z_1, \ldots, z_n)^{\mathrm{T}}$ and a rotation $\rho\colon \mathbb{R}^2 \to \mathbb{R}^2$ centered at the origin. Then ρ is a* symmetry of the configuration $\mathbf{z}$ *if and only if there exists a permutation $\kappa\colon \{\, 1, \ldots, n \,\} \to \{\, 1, \ldots, n \,\}$ such that $\rho(z_i) = z_{\kappa(i)}$ for all $i \in \{\, 1, \ldots, n \,\}$.*

To lift the rotation ρ and permutation κ from Definition 3.3 to the entire configuration $\mathbf{z} \in (\mathbb{R}^2)^n \equiv \mathbb{R}^{2n}$, we use the following block matrices M_ρ and M_κ ($n \times n$ matrices with

entries in $\mathbb{R}^{2\times 2}$) implied by ρ and κ:

$$M_\rho = \begin{pmatrix} \rho & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \rho & \ddots & \vdots \\ \vdots & \ddots & \ddots & \mathbf{0} \\ \mathbf{0} & \cdots & \mathbf{0} & \rho \end{pmatrix} \quad \text{and} \quad (M_\kappa)_{ij} = \begin{cases} \mathbf{1}_2, & \text{if } \kappa(i) = j \\ \mathbf{0}, & \text{otherwise.} \end{cases} \tag{3}$$

Here, $\mathbf{1}_l$ denotes for the $l \times l$ identity matrix and $\mathbf{0}$ the zero matrix of suitable dimensions. Without further mention, we identify both matrices with their $\mathbb{R}^{2n\times 2n}$ counterparts.

The matrices above allows us to reformulate the condition from Definition 3.3 as $M_\rho \mathbf{z} = M_\kappa \mathbf{z}$. Since M_κ is furthermore invertible with $M_\kappa^{-1} = M_{\kappa^{-1}}$ (and the inverse κ^{-1} is also a permutation), we can reformulate Definition 3.3:

▶ **Definition 3.4** (Symmetry of a Configuration (alternative)). *Consider a configuration $\mathbf{z} \in \mathbb{R}^{2n}$ and a rotation $\rho\colon \mathbb{R}^2 \to \mathbb{R}^2$, both centered at the origin. Then ρ is a* symmetry of the configuration $\mathbf{z}$ *if and only if there exists a permutation $\kappa\colon \{1,\dots,n\} \to \{1,\dots,n\}$ such that $M_\kappa M_\rho \mathbf{z} = \mathbf{z}$.*

We denote the set of all *potential symmetries* as

$$G = \left\{ M_\kappa M_\rho \mid \kappa\colon \{1,\dots,n\} \to \{1,\dots,n\} \text{ permutation}, \rho\colon \mathbb{R}^2 \to \mathbb{R}^2 \text{ rotation} \right\}. \tag{4}$$

and the subset of *(actual) symmetries* of a configuration $\mathbf{z}$ as

$$G_\mathbf{z} = \left\{ M_\kappa M_\rho \in G \mid M_\kappa M_\rho \mathbf{z} = \mathbf{z} \right\}. \tag{5}$$

With this formalization at hand, we can study how a protocol (via its evolution function) influences a configurations actual symmetries. In fact, the classical theory of equivariant dynamics [21, 7] immediately yields that a protocol can never cause the *loss* of symmetries.[6] Our Theorem 1.1 states that if the evolution function F is additionally invertible, then we also cannot *gain* symmetries. Its proof is given in Appendix A.2.

▶ **Theorem 1.1** (restated). *Consider the dynamics of an arbitrary swarm protocol with evolution function $F\colon \mathbb{R}^{2n} \to \mathbb{R}^{2n}$. Assume that F is (locally) invertible. Then, any configuration $\mathbf{z} \in \mathbb{R}^{2n}$ and its successor configuration $\mathbf{z}^+ := F(\mathbf{z})$ have the same symmetries $G_{\mathbf{z}^+} = G_\mathbf{z}$.*

Note that the assumption of invertibility is required on the level of the configuration. This means there exists a function $F^{-1}\colon \mathbb{R}^{2n} \to \mathbb{R}^{2n}$ such that $F(F^{-1}(\mathbf{z})) = F^{-1}(F(\mathbf{z})) = \mathbf{z}$ for all $\mathbf{z} \in \mathbb{R}^{2n}$. In general, F^{-1} is not the evolution function of a distributed protocol. In particular, a robot *does not* have to be able to determine its previous position based on its local information (f is not invertible). Informally speaking, from our perspective as an external observer we must always be able to determine the swarm's configuration in the previous round. The term "locally" in Theorem 1.1 is a mathematical characterization of the inverse function F^{-1}. It makes the statement stronger, since the inverse function needs to be defined only locally, meaning for all configurations that are sufficiently similar to a given one. This is, for example, central for the analysis of the averaging strategy from Section 4. For full details and precise definitions see Definition A.5 in Appendix A below.

Moreover, it is important to emphasize that the theorem does not require any regularity of the evolution function F. In particular, it remains true even if F is non-continuous, which will be essential for the protocol presented in Section 5 below.

[6] For example, robots forming an identical n-gon have identical views and, thus, perform the same local calculations. Thus, the swarm would be forever trapped in a, possibly scaled, n-gon formation.

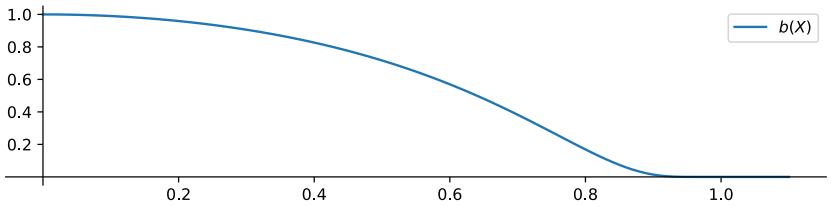


Figure 2 Graph of bump function $b(X)$. It decreases monotonically from 1 (for $X = \|z_i - z_j\|^2 = 0$; i.e., when j is at the same position as i) to 0 (for $X = 1$; i.e., when j is at the brink of being invisible).

4 Protocol 1: Preserving Symmetries via Averaging

This section presents a protocol that preserves the symmetries of a configuration (which we prove via Theorem 1.1). However, it does not always achieve Near-Gathering as certain initial configurations may result in several clusters of Near-Gatherings. Our proposed protocol ε-Go-To-The-Average does the following in each round:

1. Observes positions of visible neighbors (viewing range 1).
2. Calculate the *target point* as the *weighted* (see below) average of all visible robot positions.
3. Move an ε-fraction towards the target point for an $\varepsilon \in (0, 1)$.

The weights of the i-robot at position $\mathbf{z}_i$ for a visible neighbor at position $\mathbf{z}_j$ is derived via a monotonically decreasing *bump function* of their squared distances $X := \|z_i - z_j\|^2$ defined via

$$b(X) = \begin{cases} \exp\left(-\frac{X^2}{1-X^2}\right) & \text{if } X \in [0,1] \\ 0 & \text{if } X > 1. \end{cases} \tag{6}$$

whose graph is shown in Figure 2. With the bump function, we can model the computation of the target point of a robot in position $z_i \in \mathbb{R}^2$ of a configuration $\mathbf{z} \in \mathbb{R}^{2n}$ as

$$T(z_i; \mathbf{z}) = \frac{1}{n} \sum_{j=1}^{n} b(\|z_i - z_j\|^2)(z_j - z_i). \tag{7}$$

For simplicity of the mathematical proof below, we assume that the global information n, i.e., the total number of robots, is known to all robots. It can, however, be generalized by replacing n with $n(z_i; \mathbf{z}) = \#\{\text{robots within viewing range of robot } i\}$ (which is always at least 1). Note that the weights (values of the bump function) for robots outside of the viewing range of i are 0 and, hence, do not affect the computation. With this, we can describe

the evolution of position z_i in the form of Equation (1) by a (robot-independent) function $f\colon \mathbb{R}^2 \times \mathbb{R}^{2n} \to \mathbb{R}^2$ defined by

$$z_i^+ = f(z_i; \mathbf{z}) = z_i + \varepsilon \cdot T(z_i; \mathbf{z}) = z_i + \frac{\varepsilon}{n} \sum_{j=1}^{n} b(\|z_i - z_j\|^2) \cdot (z_j - z_i) \tag{8}$$

for some fixed $\varepsilon \in (0, 1)$. This yields our protocol's evolution function F as specified in Equation (2).

The next lemma states that if ϵ is chosen small enough, F is locally invertible. As a direct consequence of Theorem 1.1, this implies that ε-Go-To-The-Average preserves symmetries. The proof of the following lemma can be found in Appendix B.

▶ **Lemma 4.1.** *Consider the evolution function F of the ε-Go-To-The-Average protocol for $\varepsilon < \frac{n}{27(n-1)}$. Then F is locally invertible.*

Unfortunately, as with the standard Go-To-The-Average protocol, we cannot guarantee that connectivity is preserved by this adaptation. Thus, in general we might not end up in a Near-Gathering, but in several clusters at distance greater than 1, each of which can be seen as a "separate" Near-Gatherings. However, if we start in a configuration for which Go-To-The-Average maintains connectivity (like highly regular meshes), we achieve Near-Gathering without increasing the symmetries. It remains an open question to characterize such configurations more clearly (cf. Section 6).

5 Protocol 2: Preserving Symmetries via Contracting Waves

In this section, we will provide a symmetry preserving algorithm that yields Near-Gathering. However, it only works on a subset of initial configurations and it needs a larger viewing range.

Requirements of the algorithm. Remember the Connectivity-Boundary of a configuration defined in Section 3. A *δ-hole* of a configuration is a circular area inside the Connectivity-Boundary with a diameter of δ that contains no robot.

We require that the swarm starts in a configuration that contains no 1-hole and that has a convex[7] Connectivity-Boundary. The robots have a viewing range of $2 + \sqrt{2}$.

Usually, algorithms with limited visibility allow initial configurations with a connected unit disc graph. Our requirements only allow a subset of these configurations. However, the subset still allows a high variety of initial configurations of n robots with arbitrary low entropy and a diameter in $\mathcal{O}(n)$.

Overview. The robots on the Connectivity-Boundary (we call them *boundary-robots*) will perform the ϵ-Go-to-the-Middle algorithm, where robots move towards the midpoint between their two neighbors. The advantage of this algorithm is that it is invertible (and therefore connectivity preserving) and a gathering algorithm. The robots near to the boundary will move with the boundary-robots, one may get the impression they are being pushed to the inside by the boundary like a wave (we call them *wave-robots*). We will carefully construct the way wave-robots move such that this movement is invertible. All other robots (called

[7] Note, that we do not consider it to be strictly convex. It may contain multiple collinear robots.

inner-robots) do not move. The boundary will contract, more and more inner-robots will become wave-robots and are further pushed inwards until a Near-Gathering is reached.

We split the description of the algorithm in three subsections. In Section 5.1 we define the boundary-robots and their algorithm. We prove, that boundary-robots will remain a convex set during the execution of their algorithm (Lemma 5.5) and that their algorithm is invertible (Lemma 5.4). In Section 5.2 we define an area around the boundary-robots called *Wave*. All robots in this area are wave-robots. We define their algorithm and prove that their movement is invertible assuming the Wave is known. Both algorithms are combined in Section 5.3. We depict the execution of one round in Figure 3a.

5.1 Boundary Algorithm

▶ **Definition 5.1** (Boundary-Robots)**.** *Robots that are part of the Connectivity-Boundary are called* boundary-robots*. We denote the boundary robots in round t by $\mathbf{b}^t = (b_0^t, \cdots, b_k^t)$. We assume that robots are enumerated counterclockwise with b_0^t chosen arbitrarily.*

We define the following algorithm based on Go-to-the-Middle [13] for boundary-robots. Afterwards, we prove that it is invertible.

Algorithm 1 Boundary-algorithm: ϵ-Go-to-the-Middle

$$\epsilon\text{-GtM}(b_k^t, \mathbf{b}^t) := \epsilon \cdot \frac{b_{k-1}^t + b_{k+1}^t}{2} + (1-\epsilon) \cdot b_k^t$$

▶ Remark 5.2. The algorithm is in general not executable in our model. In general, robots cannot decide locally, whether they are boundary-robots. In Lemma 5.17, we prove that it is executable in swarms meeting our requirements.

▶ Remark 5.3. If not stated otherwise, we assume that $b_k^{t+1} = \epsilon\text{-GtM}(b_k^t, \mathbf{b}^t)$. Theoretically, the robots on the Connectivity-Boundary may change during the execution, depending on where other robots move. However, in Lemma 5.16, we prove that the set of boundary-robots does not change during the execution of our algorithm.

▶ **Lemma 5.4.** *If $\epsilon \in [0, 0.5)$, ϵ-Go-to-the-Middle is invertible for a global observer.*

Note, that while ϵ-GtM is invertible it is only symmetry preserving when considering the neighborhood relation as part of the symmetry.

▶ **Lemma 5.5.** *If $\mathbf{b}^t$ is convex, $\mathbf{b}^{t+1}$ is convex as well. Let $area(\mathbf{b}^t)$ denote the area enclosed by $\mathbf{b}^t$. Then, $area(\mathbf{b}^{t+1}) \subseteq area(\mathbf{b}^t)$.*

The proofs of Lemmas 5.4 and 5.5 can be found in Appendix C.

5.2 Wave-Algorithm

In this subsection we will define the set of wave-robots and construct Algorithm 2. In Lemma 5.5 we have shown, that the area enclosed by $\mathbf{b}^{t+1}$ is inside the area enclosed by $\mathbf{b}^t$. The goal of the wave-algorithm is, to remove all inner robots that are outside of the area of $\mathbf{b}^{t+1}$ inside that area such that $\mathbf{b}^{t+1}$ is the Connectivity-Boundary in round $t+1$. We first define this area formally. Because this process is similar to a wave front that pushes the robots inwards we use the terminology wave to describe it and call the area *Wave*.

▶ **Definition 5.6** (Wave). *Let area*$(\mathbf{b}^t)$ *denote the area enclosed by* $\mathbf{b}^t$*. We define* $\texttt{wave}^t$ *as* $area(\mathbf{b}^t) \setminus area(\mathbf{b}^{t+1})$ *(see Appendix C.1 for a visualization).*

▶ **Definition 5.7** (Wave-robot). *We call robots in* $\texttt{wave}^t$ *and* $\texttt{wave}^{t+1}$ *at time* t wave-robots.

▶ **Definition 5.8** (Wave-segment). *We cut* $\texttt{wave}^t$ *by cutting from* b_k^t *to* b_k^{t+1} *for all* $1 \leq k \leq n$*. This leaves us with* n Wave-Segments*. We call the segment with corners* $b_k^t, b_{k+1}^t, b_k^{t+1}, b_{k+1}^{t+1}$ *the* k*-th wave-segment of* $\texttt{wave}^t$ *or* $\texttt{seg}_k^t$*.*

We will design Algorithm 2 such that all robots from $\texttt{seg}_k^t$ move into $\texttt{seg}_k^{t+1}$. The robots in $\texttt{seg}_k^{t+1}$ move inside their segment as well, to prevent collisions with incoming robots.

Preliminary Statements To use wave-segments as a base for our algorithm, we need to make sure that they partition the robots unambiguously. In the following we prove that segments do not overlap and are not twisted. But there are configurations of $\mathbf{b}^t$ (in case of collinear robots in $\mathbf{b}^t$) where the quadrilateral is degenerated, i.e. partly without area or just a line without any area. We prove that a segment in $\texttt{wave}^t$ will not become more degenerated in $\texttt{wave}^{t+1}$.

▶ **Lemma 5.9.** *Let* $\mathbf{b}^t$ *be a convex set of robots. The segments of* $\texttt{wave}^t$ *are not twisted quadrilaterals, i.e., two sides do not intersect on a single point, and do not overlap.*

▶ **Lemma 5.10.** *We call a quadrilateral where all four corners are not collinear* non-degenerated*, a* partially degenerated *quadrilateral has 3 collinear corners and a* fully degenerated *has 4 collinear corners.*

1. *If* $\texttt{seg}_k^t$ *is a non-degenerated quadrilateral,* $\texttt{seg}_k^{t+1}$ *is also a non-degenerated quadrilateral.*
2. *If* $\texttt{seg}_k^t$ *is a partially degenerated segment,* $\texttt{seg}_k^{t+1}$ *is a non-degenerated segment.*

▶ **Lemma 5.11.** *Robots in* $\texttt{seg}_k^t$ *and* $\texttt{seg}_k^{t+1}$ *have a distance of* $\leq 1 + \epsilon^2/2$ *to* b_k^t *and* b_{k+1}^t*.*

The proofs of Lemmas 5.9–5.11 can be found in Appendix C.

Definition of the Algorithm The algorithm uses a bijective mapping from the area $\texttt{seg}_k^t \cup \texttt{seg}_k^{t+1}$ onto $\texttt{seg}_k^{t+1}$. For this mapping we define a normalized two-dimensional coordinate system inside each segment from ranging from $(0,0)$ to $(1,1)$ (Definition 5.12). In rectangular segments these are simple vertical and horizontal lines and the x- and y unit-distance is scaled accordingly, for other shapes the lines are adjusted to follow the boundaries of the shape. Algorithm 2 is a simple mapping between the areas of the segments based on the normalized coordinates. Wave robots from $\texttt{seg}_k^t$ move inside the outer half of $\texttt{seg}_k^{t+1}$ while the robots inside $\texttt{seg}_k^{t+1}$ move into the inner half of $\texttt{seg}_k^{t+1}$. See Figure 3a and Appendix C.1 for an example. In the end we prove that the wave algorithm is invertible (Lemma 5.13).

▶ **Definition 5.12** (normalized coordinate-system inside a quadrilateral). *Let* A, B, C, D *be the corners of the quadrilateral, we denote the line-segments between the corners by* $\overline{AB}$*. For* $x \in [0,1]$ *we define* $(x,0) := A + x \cdot (D - A)$ *and* $(x,1) := B + x \cdot (C - B)$ *(in particular* $(0,0) = A; (1,0) = B; (1,1) = C$ *and* $(0,1) = D$*). If the quadrilateral is convex, for* $y \in [0,1]$ *we equally distribute* (x,y) *on the straight line between* $(x,0)$ *and* $(x,1)$*. For non-convex quadrilaterals we assume w.l.o.g. that* C *is the concave corner. For* $x \in [0,1]$ *we connect* $(x,0)$ *and* $(x,1)$ *with a parallel to* $\overline{AB}$ *starting at* $(x,0)$ *and a parallel to* $\overline{CD}$ *starting at* $(x,1)$*. For* $y \in [0,1]$ *we equally distribute* (x,y) *on this connection. See Figure 3b and the figures in Appendix C.1 for an example.*

For wave-segment $\texttt{seg}_k^t$ *we define* $A = b_k^t, B = b_{k+1}^t, C = b_{k+1}^{t+1}$ *and* $D = b_k^{t+1}$*. To denote the position* (x,y) *inside* $\texttt{seg}_k^t$ *we use the notation* $\texttt{seg}_k^t(x,y)$*.*

Algorithm 2 Wave-Algorithm

This algorithm gets the positions of boundary-robots $\mathbf{b}^t$ as input. It is used to compute $\texttt{wave}^t$ and $\texttt{wave}^{t+1}$. The algorithm then determines whether z_i^t is in $\texttt{seg}_k^t$ or $\texttt{seg}_k^{t+1}$ for some k and the coordinates x and y according to Definition 5.12.

$$\textsc{Wave-Algorithm}(z_i^t, \mathbf{b}^t) = \begin{cases} \texttt{seg}_k^{t+1}(x/2, y), & \text{if } z_i^t = \texttt{seg}_k^t(x, y) \\ \texttt{seg}_k^{t+1}(1/2 + x/2, y), & \text{if } z_i^t = \texttt{seg}_k^{t+1}(x, y) \end{cases}$$

▶ **Lemma 5.13.** *Assuming $\mathbf{b}^t$ and the wave-robots in round t are fixed and known. After executing Algorithm 2 with all wave-robots, we can compute the positions of the wave-robots in round t.*

Proof. All robots from $\texttt{seg}_k^t$ and $\texttt{seg}_k^{t+1}$ moved inside $\texttt{seg}_k^{t+1}$ for $0 \leq k < |\mathbf{b}^t|$. The segments do not overlap (Lemma 5.9), therefore the movement is unambiguous. If $\texttt{seg}_k^t$ is non-degenerated, then $\texttt{seg}_k^{t+1}$ is also non-degenerated (Lemma 5.10). In both segments, the coordinates are unambiguous. Therefore, Definition 5.12 yields a bijective mapping.

If $\texttt{seg}_k^t$ is partly degenerated, there may exist $(x, y) \neq (x', y')$ with $\texttt{seg}_k^t(x, y) = \texttt{seg}_k^t(x', y')$. But $\texttt{seg}_k^{t+1}$ is non-degenerated (Lemma 5.10), therefore no two robots move onto the same positions in $\texttt{seg}_k^{t+1}$. Therefore, the origins for all robots is unambiguous. If $\texttt{seg}_k^{t+1}(x, y)$ is fully-degenerated (is only a line), it follows from Lemma 5.10 that $\texttt{seg}_k^t(x, y)$ must be fully-degenerated as well. But because both segments are only a line without area in this case, the mapping is bijective. ◀

5.3 Main Algorithm

We first define *inner-robots*, that are the remaining robots beside boundary- and wave-robots. Afterwards we state the main algorithm formally and prove its correctness as well as that it is symmetry preserving.

▶ **Definition 5.14** (inner-robots). *We call robots in $\texttt{wave}^q, q > t + 1$ at time t* inner-robots.

Algorithm 3 Main Algorithm

Based on $\mathbf{z}^t$ the positions of boundary-robots $\mathbf{b}^t$ can be observed. $\mathbf{b}^t$ is used to compute $\texttt{wave}^t$ and $\texttt{wave}^{t+1}$ to determine wave-robots and inner-robots.

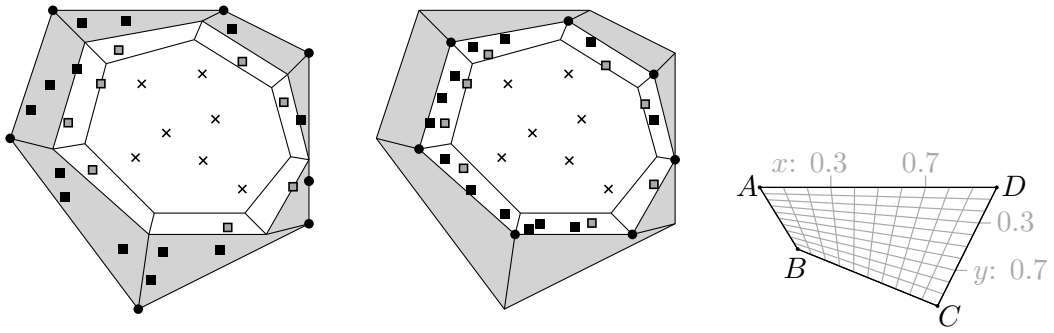
$$f(z_k^t, \mathbf{z}^t) = \begin{cases} \epsilon\text{-GTM}(z_k^t, \mathbf{b}^t) & \text{if } z_k^t \text{ is a boundary-robot (Definition 5.1),} \\ \textsc{Wave-Algorithm}(z_k^t, \mathbf{b}^t) & \text{if } z_k^t \text{ is a wave-robot (Definition 5.7),} \\ z_k^t & \text{if } z_k^t \text{ is an inner-robot (Definition 5.14).} \end{cases}$$

See Figure 9 and the figures in Appendix C.1 for an example.

▶ Remark 5.15. To execute Algorithm 3 as it is written, robots must observe $\mathbf{b}^t$. They are not able to observe $\mathbf{b}^t$ fully because of the limited visibility in out model. But we prove in Lemma 5.17 that they are able to observe the locally relevant part of $\mathbf{b}^t$ to compute Algorithm 3 locally.

▶ **Lemma 5.16.** *If $\mathbf{b}^t$ is convex and the Connectivity-Boundary of a swarm without 2.24-holes that executes Algorithm 3 than*

$$\mathbf{b}^{t+1} = \epsilon\text{-GTM}(\mathbf{b}^t)$$



(a) Execution of Algorithm 3 in round t with boundary-robots(•), wave-robots (□) and inner-robots (×). $\texttt{wave}^t$ (gray) and $\texttt{wave}^{t+1}$ are partitioned in segments. More details in Appendix C.1.

(b) Shows a grid inside a segment according to Definition 5.12.

Figure 3 Visualization of Algorithm 3 and Definition 5.12.

Proof. From Lemma 5.5 we know, that ϵ-GTM($\mathbf{b}^t$) is convex. Neighboring positions in ϵ-GTM($\mathbf{b}^t$) have distance ≤ 1. Therefore, ϵ-GTM($\mathbf{b}^t$) would be the Connectivity-Boundary of the new configuration, if all other robots are within $area(\epsilon\text{-GTM}(\mathbf{b}^t))$. We defined $\texttt{wave}^t$ as $area(\mathbf{b}^t) \setminus area(\epsilon\text{-GTM}(\mathbf{b}^t))$. Algorithm 2 is designed such that all robots from $\texttt{wave}^t$ move into $\texttt{wave}^{t+1}$, therefore moving inside $area(\epsilon\text{-GTM}(\mathbf{b}^t))$. Only robots from coordinates $\texttt{seg}_k^t(x, 0)$ move onto coordinates $\texttt{seg}_k^{t+1}(x, 0)$ (see Algorithm 2). On coordinates $\texttt{seg}_k^t(x, 0)$ are by definition only boundary-robots. Therefore, ϵ-GTM($\mathbf{b}^t$) is the Connectivity-Boundary of the swarm in round $t + 1$. ◀

▶ **Lemma 5.17.** *In a configuration with a convex Connectivity-Boundary and no* 2.24*-holes, Algorithm 3* ***is executable*** *for $\mathcal{OBLOT}$-robots with a viewing range of* $2 + \sqrt{2}$.

▶ **Lemma 5.18.** *In a configuration with a convex Connectivity-Boundary Algorithm 3* ***does not lead to collisions****.*

▶ **Lemma 5.19.** *In a configuration with convex Connectivity-Boundary and initially no* 1*-holes, Algorithm 3* ***does not create*** 2.24***-holes****.*

The proofs of Lemmas 5.17–5.19 can be found in Appendix C.

▶ **Lemma 5.20.** *In a configuration with a convex Connectivity-Boundary, Algorithm 3* ***is locally invertible****.*

Proof. Let us assume $\mathbf{b}^t$ is convex and the Connectivity-Boundary of the swarm in round t. For the initial configuration this is true by definition. We will show, that we can compute the swarm in round t from round $t + 1$. From Lemma 5.5 we know that $\mathbf{b}^{t+1}$ is convex. From Lemma 5.16 we know, that $\mathbf{b}^{t+1}$ is the Connectivity-Boundary in round $t + 1$. The Connectivity-Boundary can easily be identified by a global observer, therefore $\mathbf{b}^{t+1}$ is known. By Lemma 5.4 we know that ϵ-GTM is invertible. Therefore, we can compute $\mathbf{b}^t$ and $\texttt{wave}^t$ as well as $\texttt{wave}^{t+1}$. We know that all robots that were in round t not in $\texttt{wave}^t$ or $\texttt{wave}^{t+1}$ are inner-robots and have not moved (Definition 5.14). Therefore, all robots in $\texttt{wave}^{t+1}$ in round $t + 1$ must have been wave- or boundary-robots in round t. Because all wave- and boundary-robots in round t move into $\texttt{wave}^{t+1}$ the robots in $\texttt{wave}^{t+1}$ in round $t + 1$ are the complete set of wave- and boundary-robots in round $t + 1$. The boundary-robots are already identified, therefore we know the set of wave-robots. This allows us to apply Lemma 5.13 to compute the positions of the wave-robots in round t. All other robots are inner robots, therefore they do not move in round t. ◀

▶ **Theorem 5.21.** *We assume robots according the $\mathcal{OBLOT}$ model and $\mathcal{F}$SYNC scheduler with a viewing range of $2+\sqrt{2}$ in a swarm with a convex Connectivity-Boundary and no 1-holes. Algorithm 3 leads to* Near-Gathering *and does not change the symmetry.*

Proof. The algorithm is executable in the assumed model (Lemma 5.17). From [13] we know, that ϵ-Go-to-the-Middle converges towards gathering. From Lemma 5.16 we know, that $\mathbf{b}^{t+1} = \epsilon\text{-GtM}(\mathbf{b}^t)$. Therefore, $\mathbf{b}^t$ will converge towards gathering for $t \leftarrow \infty$. Because $\mathbf{b}^t$ is the Connectivity-Boundary all robots are within $area(\mathbf{b}^t)$. At the same time, each robot keeps its own position (Lemma 5.18). Therefore, eventually a Near-Gathering will be reached. Because Algorithm 3 is locally invertible (Lemma 5.20) we can follow from Theorem 1.1 that the symmetries are not changed. ◀

▶ Remark 5.22. The algorithm presented above works for general swarms. Even if the Connectivity-Boundary is not convex, the robots at the convex hull of the swarm can always be detected as boundary-robots and move inwards, such that the convex hull shrinks monotonically. There may be movements at other parts of the swarm, especially if the swarm contains holes, but ϵ-GtM does never result in connectivity loss. For situations where a robot is only connected to one other robots (e.g. robots form a line) ϵ-GtM must be slightly altered (a robot considers one neighbor as its left and right neighbor at the same time in this situations). While a Near-Gathering is always reached, the symmetry preservation cannot be guarantied in general.

6 Discussion & Conclusion

In our paper we partially solved the Near-Gathering problem with symmetry preservation. On one hand, we gave a variant of Go-To-The-Average that preserves symmetry for all initial configurations and even reaches Near-Gathering for certain configurations. On the other hand, Algorithm 3 solves Near-Gathering for all configurations but preserves symmetry only for a subset of configurations. This opens a few questions.

Can we loosen the restrictions of Algorithm 3? A central part of our analysis is that the robots on the Connectivity-Boundary perform the same linear function during the execution. This is only the case, if the set of boundary-robots never changes. If we allow for a non-convex Connectivity-Boundary, robots move towards the outside of the swarm in concave sections. So, eventually two robots of the Connectivity-Boundary that initially have a distance > 1 will reach a distance ≤ 1. One of these robots is longer part of the Connectivity-Boundary in the next round. One could fix this by introducing memory, but with memory symmetry preserving Near-Gathering is already solved [37]. Another possible fix is to not move robots on concave parts of the Connectivity-Boundary. But eventually, these parts become convex and robots start to move according to ϵ-GtM which, again, changes the linear function. The second restriction are holes. Robots at the boundary of large holes cannot distinguish their location locally from the Connectivity-Boundary (not even with memory). Therefore, they will start with a Near-Gathering that eventually leads to a configuration, where no hole exists anymore. But this configuration can have another pre-image where the inner robots already had the position of the not-anymore-hole. Therefore, the algorithm is not invertible in this case.

Can we create a framework for a more general strategy from Algorithm 3? The proofs in Section 5 can be generalized for a class of strategies. Our core idea is to split the robots

into layers (e.g. boundary, wave, inner). The outermost layer performs a Near-Gathering algorithm that is symmetry preserving. All other layers perform algorithms, such that they stay inside the outermost layer. The outermost layer is always distinguishable and invertible. The inner layers are distinguishable and invertible, if the outer layers are known. The advantage of these class of algorithms is, that you can reduce the problem of an invertible Near-Gathering algorithm to a restricted set of robots. However, even for a restricted set of robots like those on the Connectivity-Boundary it turns out to be a major challenge to find a connectivity preserving algorithm.

For which class of configurations does Go-To-The-Average lead to Near-Gathering? The configurations must contain evenly spread robots. One example is a square (or triangle or hexagon) that is filled with a regular grid with distance $<$ viewing range. Also perturbations of such configurations lead to Near-Gathering.

When do Near-Gathering algorithms from [5] increase symmetry? In [5], a class of Near-Gathering functions is introduced. These functions do not in general preserve the symmetry. A simple example is the Go-To-The-Center algorithm where robots move onto the center of the smallest enclosing circle of their local neighborhood. Let us assume a configuration $\mathbf{z}^t$ with symmetry. After performing Go-To-The-Center the new configuration $\mathbf{z}^{t+1}$ must be symmetric as well. The smallest enclosing circle is dependent only on robots exactly on the circle. We fix all robots that are on these circles and perturb all other robots such that the configuration is asymmetric. Let this be configuration $\breve{\mathbf{z}}^t$. After performing Go-To-The-Center the new configuration is $\breve{\mathbf{z}}^{t+1}$. Because the smallest enclosing circles have not changed, the robots in $\breve{\mathbf{z}}^t$ move onto the same positions as the robots in $\mathbf{z}^t$. Therefore, with $\breve{\mathbf{z}}^{t+1} = \mathbf{z}^{t+1}$ we introduced symmetries.

6.1 Conclusion

We presented in Section 4 the first known non-trivial algorithm for local robots that preserves symmetry. This in itself is no small achievement because the local capabilities (especially limited visibility) make it impossible for robots to observe the current symmetry and to act accordingly. Additionally, we presented a Near-Gathering algorithm that preserves the symmetry for a large subset of initial configurations. The algorithm is based on ϵ-GtM, a linear function that leads to the gathering of swarms connected in a chain. The advantage of a linear function is that we can analyze the invertibility using well-known properties from linear algebra. Other gathering algorithms like Go-To-The-Center contain conditions which robots in the viewing range to consider for the computation (GtC ignores all but three robots). Such properties make it hard to analyze their invertibility to apply our analysis method. As we have argued in the discussion, the initial configuration limitation stems from the problem of getting a consistent chain of robots (in our case, the Connectivity-Boundary) that executes ϵ-GtM while having no memory and no global knowledge.

References

1 Muhammad Zaki Almuzakki and Bayu Jayawardhana. Cooperative nearest-neighbor control of multi-agent systems: Consensus and formation control problems. *IEEE Control Systems Letters*, 7:1873–1878, 2023. doi:10.1109/LCSYS.2023.3282423.

2 Kaustav Bose, Ranendu Adhikary, Manash Kumar Kundu, and Buddhadeb Sau. Arbitrary pattern formation on infinite grid by asynchronous oblivious robots. *Theor. Comput. Sci.*,

815:213–227, 2020. URL: https://doi.org/10.1016/j.tcs.2020.02.016, doi:10.1016/J.TCS.2020.02.016.

3 Kaustav Bose, Manash Kumar Kundu, Ranendu Adhikary, and Buddhadeb Sau. Arbitrary pattern formation by asynchronous opaque robots with lights. *Theor. Comput. Sci.*, 849:138–158, 2021. URL: https://doi.org/10.1016/j.tcs.2020.10.015, doi:10.1016/J.TCS.2020.10.015.

4 Jannik Castenow, Matthias Fischer, Jonas Harbig, Daniel Jung, and Friedhelm Meyer auf der Heide. Gathering anonymous, oblivious robots on a grid. *Theor. Comput. Sci.*, 815:289–309, 2020. URL: https://doi.org/10.1016/j.tcs.2020.02.018, doi:10.1016/J.TCS.2020.02.018.

5 Jannik Castenow, Jonas Harbig, Daniel Jung, Peter Kling, Till Knollmann, and Friedhelm Meyer auf der Heide. A unifying approach to efficient (near)-gathering of disoriented robots with limited visibility. In Eshcar Hillel, Roberto Palmieri, and Etienne Rivière, editors, *26th International Conference on Principles of Distributed Systems, OPODIS 2022, December 13-15, 2022, Brussels, Belgium*, volume 253 of *LIPIcs*, pages 15:1–15:25. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2022. URL: https://doi.org/10.4230/LIPIcs.OPODIS.2022.15, doi:10.4230/LIPICS.OPODIS.2022.15.

6 Yao Chen, Jinhu Lu, Xinghuo Yu, and David J. Hill. Multi-agent systems with dynamical topologies: Consensus and applications. *IEEE Circuits and Systems Magazine*, 13(3):21–34, 2013. doi:10.1109/MCAS.2013.2271443.

7 Pascal Chossat and Reiner Lauterbach. *Methods in Equivariant Bifurcations and Dynamical Systems*, volume 15 of *Advanced series in nonlinear dynamics*. World Scientific, 2000.

8 Serafino Cicerone, Gabriele Di Stefano, and Alfredo Navarra. Asynchronous arbitrary pattern formation: the effects of a rigorous approach. *Distributed Comput.*, 32(2):91–132, 2019. URL: https://doi.org/10.1007/s00446-018-0325-7, doi:10.1007/S00446-018-0325-7.

9 Reuven Cohen and David Peleg. Convergence properties of the gravitational algorithm in asynchronous robot systems. *SIAM J. Comput.*, 34(6):1516–1528, 2005. doi:10.1137/S0097539704446475.

10 Shantanu Das, Paola Flocchini, Nicola Santoro, and Masafumi Yamashita. Forming sequences of geometric patterns with oblivious mobile robots. *Distributed Comput.*, 28(2):131–145, 2015. URL: https://doi.org/10.1007/s00446-014-0220-9, doi:10.1007/S00446-014-0220-9.

11 Bastian Degener, Barbara Kempkes, Tobias Langner, Friedhelm Meyer auf der Heide, Peter Pietrzyk, and Roger Wattenhofer. A tight runtime bound for synchronous gathering of autonomous robots with limited visibility. In Rajmohan Rajaraman and Friedhelm Meyer auf der Heide, editors, *SPAA 2011: Proceedings of the 23rd Annual ACM Symposium on Parallelism in Algorithms and Architectures, San Jose, CA, USA, June 4-6, 2011 (Co-located with FCRC 2011)*, pages 139–148. ACM, 2011. doi:10.1145/1989493.1989515.

12 Florian Dörfler and Francesco Bullo. Synchronization in complex networks of phase oscillators: A survey. *Automatica*, 50(6):1539–1564, 2014. URL: https://www.sciencedirect.com/science/article/pii/S0005109814001423, doi:10.1016/j.automatica.2014.04.012.

13 Miroslaw Dynia, Jaroslaw Kutylowski, Pawel Lorek, and Friedhelm Meyer auf der Heide. Maintaining communication between an explorer and a base station. In Yi Pan, Franz J. Rammig, Hartmut Schmeck, and Mauricio Solar, editors, *Biologically Inspired Cooperative Computing, IFIP 19th World Computer Congress, TC 10: 1st IFIP International Conference on Biologically Inspired Computing, August 21-24, 2006, Santiago, Chile*, volume 216 of *IFIP*, pages 137–146. Springer, 2006. doi:10.1007/978-0-387-34733-2_14.

14 Paola Flocchini, Giuseppe Prencipe, and Nicola Santoro, editors. *Distributed Computing by Mobile Entities, Current Research in Moving and Computing*, volume 11340 of *Lecture Notes in Computer Science*. Springer, 2019. doi:10.1007/978-3-030-11072-7.

15 Paola Flocchini, Giuseppe Prencipe, and Nicola Santoro. Moving and computing models: Robots. In Paola Flocchini, Giuseppe Prencipe, and Nicola Santoro, editors, *Distributed Com-*

puting by Mobile Entities, Current Research in Moving and Computing, volume 11340 of *Lecture Notes in Computer Science*, pages 3–14. Springer, 2019. doi:10.1007/978-3-030-11072-7_1.

16 Paola Flocchini, Giuseppe Prencipe, Nicola Santoro, and Giovanni Viglietta. Distributed computing by mobile robots: uniform circle formation. *Distributed Comput.*, 30(6):413–457, 2017. URL: https://doi.org/10.1007/s00446-016-0291-x, doi:10.1007/S00446-016-0291-X.

17 Paola Flocchini, Giuseppe Prencipe, Nicola Santoro, and Peter Widmayer. Arbitrary pattern formation by asynchronous, anonymous, oblivious robots. *Theor. Comput. Sci.*, 407(1-3):412–447, 2008. URL: https://doi.org/10.1016/j.tcs.2008.07.026, doi:10.1016/J.TCS.2008.07.026.

18 Nao Fujinaga, Yukiko Yamauchi, Hirotaka Ono, Shuji Kijima, and Masafumi Yamashita. Pattern formation by oblivious asynchronous mobile robots. *SIAM J. Comput.*, 44(3):740–785, 2015. doi:10.1137/140958682.

19 Raphael Gerlach, Sören von der Gracht, Christopher Hahn, Jonas Harbig, and Peter Kling. Symmetry preservation in swarms of oblivious robots with limited visibility, 2024. URL: https://arxiv.org/abs/2409.19277, arXiv:2409.19277.

20 Semyon Aronovich Geršgorin. Über die Abgrenzung der Eigenwerte einer Matrix. *Izvestija Akademii Nauk SSSR, Serija Matematika*, 7(6):749–754, 1931.

21 Martin Golubitsky, Ian Stewart, and David G. Schaeffer. *Singularities and Groups in Bifurcation Theory*, volume 69 of *Applied mathematical sciences*. Springer, 1988. doi:10.1007/978-1-4612-4574-2.

22 Christopher Hahn, Jonas Harbig, and Peter Kling. Forming large patterns with local robots in the OBLOT model. In Arnaud Casteigts and Fabian Kuhn, editors, *3rd Symposium on Algorithmic Foundations of Dynamic Networks, SAND 2024, June 5-7, 2024, Patras, Greece*, volume 292 of *LIPIcs*, pages 14:1–14:20. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2024. URL: https://doi.org/10.4230/LIPIcs.SAND.2024.14, doi:10.4230/LIPICS.SAND.2024.14.

23 Morris W. Hirsch and Stephen Smale. *Differential equations, dynamical systems, and linear algebra*, volume 60 of *Pure and applied mathematics*. Acad. Press, [nachdr.] edition, 1988. Smale, Stephen (VerfasserIn).

24 Chang-kwon Kang, Farbod Fahimi, Rob Griffin, D Brian Landrum, Bryan Mesmer, Guangsheng Zhang, Taeyoung Lee, Hikaru Aono, Jeremy Pohly, Jesse McCain, et al. Marsbee-swarm of flapping wing flyers for enhanced mars exploration. Technical report, NASA, 2019.

25 Anaïs Khuong, Guy Theraulaz, Christian Jost, Andrea Perna, and Jacques Gautrais. A computational model of ant nest morphogenesis. In Tom Lenaerts, Mario Giacobini, Hugues Bersini, Paul Bourgine, Marco Dorigo, and René Doursat, editors, *Advances in Artificial Life: 20th Anniversary Edition - Back to the Origins of Alife, ECAL 2011, Paris, France, August 8-12, 2011*, pages 404–411. MIT Press, 2011. URL: http://mitpress.mit.edu/sites/default/files/titles/alife/0262297140chap61.pdf.

26 David G. Kirkpatrick, Irina Kostitsyna, Alfredo Navarra, Giuseppe Prencipe, and Nicola Santoro. Separating bounded and unbounded asynchrony for autonomous robots: Point convergence with limited visibility. In Avery Miller, Keren Censor-Hillel, and Janne H. Korhonen, editors, *PODC '21: ACM Symposium on Principles of Distributed Computing, Virtual Event, Italy, July 26-30, 2021*, pages 9–19. ACM, 2021. doi:10.1145/3465084.3467910.

27 Moumita Mondal and Sruti Gan Chaudhuri. Uniform circle formation by swarm robots under limited visibility. In Dang Van Hung and Meenakshi D'Souza, editors, *Distributed Computing and Internet Technology - 16th International Conference, ICDCIT 2020, Bhubaneswar, India, January 9-12, 2020, Proceedings*, volume 11969 of *Lecture Notes in Computer Science*, pages 420–428. Springer, 2020. doi:10.1007/978-3-030-36987-3_28.

28 Linda Pagli, Giuseppe Prencipe, and Giovanni Viglietta. Getting close without touching: near-gathering for autonomous mobile robots. *Distributed Comput.*, 28(5):333–349, 2015. URL: https://doi.org/10.1007/s00446-015-0248-5, doi:10.1007/S00446-015-0248-5.

29 Arkadij Pikovskij, Michael Rosenblum, and Jürgen Kurths. *Synchronization*, volume 12 of *Cambridge nonlinear science series*. Cambridge Univ. Press, 1st paperback ed., repr edition, 2003. URL: http://www.loc.gov/catdir/description/cam041/2003283137.html.

30 Thomas D. Seeley, P. Kirk Visscher, Thomas Schlegel, Patrick M. Hogan, Nigel R. Franks, and James A. R. Marshall. Stop Signals Provide Cross Inhibition in Collective Decision-Making by Honeybee Swarms. *Science*, 335(6064):108–111, January 2012. URL: https://www.science.org/doi/10.1126/science.1210361, doi:10.1126/science.1210361.

31 Fernando Soto, Jie Wang, Rajib Ahmed, and Utkan Demirci. Medical Micro/Nanorobots in Precision Medicine. *Advanced Science*, 7(21), November 2020. URL: https://onlinelibrary.wiley.com/doi/10.1002/advs.202002203, doi:10.1002/advs.202002203.

32 Ichiro Suzuki and Masafumi Yamashita. Distributed anonymous mobile robots: Formation of geometric patterns. *SIAM J. Comput.*, 28(4):1347–1363, 1999. doi:10.1137/S009753979628292X.

33 Giovanni Viglietta. Uniform circle formation. In Paola Flocchini, Giuseppe Prencipe, and Nicola Santoro, editors, *Distributed Computing by Mobile Entities, Current Research in Moving and Computing*, volume 11340 of *Lecture Notes in Computer Science*, pages 83–108. Springer, 2019. doi:10.1007/978-3-030-11072-7_5.

34 Masafumi Yamashita and Ichiro Suzuki. Characterizing geometric patterns formable by oblivious anonymous mobile robots. *Theor. Comput. Sci.*, 411(26-28):2433–2453, 2010. URL: https://doi.org/10.1016/j.tcs.2010.01.037, doi:10.1016/J.TCS.2010.01.037.

35 Yukiko Yamauchi. A survey on pattern formation of autonomous mobile robots: asynchrony, obliviousness and visibility. *Journal of Physics: Conference Series*, 473:012016, December 2013. URL: https://iopscience.iop.org/article/10.1088/1742-6596/473/1/012016, doi:10.1088/1742-6596/473/1/012016.

36 Yukiko Yamauchi. Symmetry of anonymous robots. In Paola Flocchini, Giuseppe Prencipe, and Nicola Santoro, editors, *Distributed Computing by Mobile Entities, Current Research in Moving and Computing*, volume 11340 of *Lecture Notes in Computer Science*, pages 109–133. Springer, 2019. doi:10.1007/978-3-030-11072-7_6.

37 Yukiko Yamauchi and Masafumi Yamashita. Pattern formation by mobile robots with limited visibility. In Thomas Moscibroda and Adele A. Rescigno, editors, *Structural Information and Communication Complexity - 20th International Colloquium, SIROCCO 2013, Ischia, Italy, July 1-3, 2013, Revised Selected Papers*, volume 8179 of *Lecture Notes in Computer Science*, pages 201–212. Springer, 2013. doi:10.1007/978-3-319-03578-9_17.

A Dynamical Systems Proofs for Section 3

In this appendix, we provide some additional details on the mathematical dynamical systems and symmetries perspective and, in particular, prove Theorem 1.1.

A.1 Symmetries of a protocol

We begin by investigating the interplay of the symmetries defined in Section 3 and the dynamics induced by a protocol. The robots do not know the global coordinate system that we choose as the external observer. Hence, their computation and movement is insensitive to rotations of the global coordinates in the sense that collectively rotating all robots positions by ρ causes each robot to move to a rotated target point in each round.

▶ Remark A.1. In principle, the same holds true for translations of the global coordinate systems. However, since we fixed the origin, we typically omit these transformations without loss of generality.

On the other hand, recall that the robots are indistinguishable. That means if a robot observes another robot in a certain position, it does not know which label this robot has.

More precisely, the computations and movement of every single robot depend on the set of the other robots' positions rather than their ordered tuple. Once again, this can be recast by stating that computation and movement are insensitive to arbitrary permutations of all robots positions.

Summarizing these observations and reformulating them in terms of the function governing the dynamics of all robots f, we obtain

▶ **Lemma A.2.** *Let $\eta \in \mathbb{R}^2$ and $\boldsymbol{\zeta} = (\zeta_1, \dots, \zeta_n)^{\mathrm{T}} \in \mathbb{R}^{2n}$ be arbitrary. The function governing the dynamics of all robots f has the following* symmetry properties*:*

i $f(\rho\eta; \rho\zeta_1, \dots, \rho\zeta_n) = \rho f(\eta; \zeta)$ *for all rotations* $\rho\colon \mathbb{R}^2 \to \mathbb{R}^2$*;*

ii $f(\eta; \zeta_{\kappa(1)}, \dots, \rho\zeta_{\kappa(n)}) = f(\eta; \zeta)$ *for all permutations* $\kappa\colon \{1, \dots, n\} \to \{1, \dots, n\}$*.*

These can be restated using matrices (3) *as*

i $f(\rho\eta; M_\rho\zeta) = \rho f(\eta; \zeta)$ *for all rotations* $\rho\colon \mathbb{R}^2 \to \mathbb{R}^2$*;*

ii $f(\eta; M_\kappa\zeta) = f(\eta; \zeta)$ *for all permutations* $\kappa\colon \{1, \dots, n\} \to \{1, \dots, n\}$*.*

From the global perspective, considering the collective evolution of the entire formation, these symmetry properties imply that the evolution is insensitive to arbitrary rotations and arbitrary permutations of the robots. More precisely, it does not matter if first all robots compute/move and then rotate/permute or do it the other way around. More precisely, we may even replace rotate/permute by rotate and permute which gives us the combined transformations in G.

▶ **Proposition A.3.** *The evolution function F is symmetric—or* equivariant*—with respect to all potential symmetries of formations $M_\kappa M_\rho \in G$:*

$$F \circ (M_\kappa M_\rho) = (M_\kappa M_\rho) \circ F. \tag{A.1}$$

Proof. We claim that it suffices to prove that F is equivariant as in (A.1) with respect to M_κ for all κ and M_ρ for all ρ individually to prove the statement. In fact, if this holds true we immediately see

$$\begin{aligned} F \circ (M_\kappa M_\rho) &= F \circ (M_\kappa \circ M_\rho) \\ &= (F \circ M_\kappa) \circ M_\rho \\ &= (M_\kappa \circ F) \circ M_\rho \\ &= M_\kappa \circ (F \circ M_\rho) \\ &= M_\kappa \circ (M_\rho \circ F) \\ &= (M_\kappa M_\rho) \circ F. \end{aligned}$$

Hence, we prove the claim using the symmetry properties in Lemma A.2. To that end let $\boldsymbol{\zeta} = (\zeta_1, \dots, \zeta_n)^{\mathrm{T}} \in \mathbb{R}^{2n}$ be an arbitrary point in configuration space, $\kappa\colon \{1, \dots, n\} \to \{1, \dots, n\}$ an arbitrary permutation, and $\rho\colon \mathbb{R}^2 \to \mathbb{R}^2$ an arbitrary rotation. Then, we

compute

$$\begin{aligned}(F \circ M_\kappa)(\boldsymbol{\zeta}) &= F(M_\kappa \boldsymbol{\zeta}) \\ &= \begin{pmatrix} f((M_\kappa \boldsymbol{\zeta})_1, M_\kappa \boldsymbol{\zeta}) \\ \vdots \\ f((M_\kappa \boldsymbol{\zeta})_n, M_\kappa \boldsymbol{\zeta}) \end{pmatrix} \\ &= \begin{pmatrix} f(\zeta_{\kappa(1)}, M_\kappa \boldsymbol{\zeta}) \\ \vdots \\ f(\zeta_{\kappa(n)}, M_\kappa \boldsymbol{\zeta}) \end{pmatrix} \\ &= \begin{pmatrix} f(\zeta_{\kappa(1)}, \boldsymbol{\zeta}) \\ \vdots \\ f(\zeta_{\kappa(n)}, \boldsymbol{\zeta}) \end{pmatrix} \\ &= \begin{pmatrix} F(\boldsymbol{\zeta})_{\kappa(1)} \\ \vdots \\ F(\boldsymbol{\zeta})_{\kappa(n)} \end{pmatrix} \\ &= M_\kappa F(\boldsymbol{\zeta}) \\ &= (M_\kappa \circ F)(\boldsymbol{\zeta}), \end{aligned} \tag{A.2}$$

where the fourth equality holds due to Lemma A.2. Similarly, we obtain

$$\begin{aligned}(F \circ M_\rho)(\boldsymbol{\zeta}) &= F(M_\rho \boldsymbol{\zeta}) \\ &= \begin{pmatrix} f((M_\rho \boldsymbol{\zeta})_1, M_\rho \boldsymbol{\zeta}) \\ \vdots \\ f((M_\rho \boldsymbol{\zeta})_n, M_\rho \boldsymbol{\zeta}) \end{pmatrix} \\ &= \begin{pmatrix} f(\rho\zeta_1, M_\rho \boldsymbol{\zeta}) \\ \vdots \\ f(\rho\zeta_n, M_\rho \boldsymbol{\zeta}) \end{pmatrix} \\ &= \begin{pmatrix} \rho f(\zeta_1, \boldsymbol{\zeta}) \\ \vdots \\ \rho f(\zeta_n, \boldsymbol{\zeta}) \end{pmatrix} \\ &= M_\rho F(\boldsymbol{\zeta}) \\ &= (M_\rho F)(\boldsymbol{\zeta}) \end{aligned} \tag{A.3}$$

again using Lemma A.2. This completes the proof of the claim. ◀

By the previous proposition, F commutes with all elements of G. Hence, whenever we refer to an arbitrary symmetry without the need to specify rotation and permutation separately, we use $M, M', \ldots \in G$ from now on.

A.2 Proof of Theorem 1.1

Proposition A.3 places our mathematical framework in the context of *equivariant dynamics*, for which there exists a well developed theory to investigate the interplay of dynamics and symmetries (e.g. [21, 7]). It allows us to prove Theorem 1.1. We do so in the following three propositions, which combined give the statement of the theorem.

▶ **Proposition A.4.** *Consider the dynamics of a configuration according to an arbitrary protocol* (2)*. Then the configuration after one round cannot have fewer symmetries than the initial one:* $G_{\mathbf{z}} \subset G_{\mathbf{z}^+}$.

Proof. Let $\mathbf{z} \in \mathbb{R}^{2n}$ be some arbitrary configuration and $M \in G_{\mathbf{z}}$. Consider the evolution $\mathbf{z}^+ = F(\mathbf{z})$ in one round. Then

$$M\mathbf{z}^+ = MF(\mathbf{z}) = F(M\mathbf{z}) = F(\mathbf{z}) = \mathbf{z}^+,$$

where we have exploited the symmetry of F (Proposition A.3) as well as the fact that M leaves $\mathbf{z}$ unchanged. In particular, this implies $M \in G_{\mathbf{z}^+}$ proving the statement. ◀

In a very similar manner we may prove that a configuration cannot gain any symmetries during the temporal evolution. This statement, however, is only true in general if the evolution function F is invertible.

▶ **Definition A.5** ((Local) invertibility)**.** *An evolution function* $F\colon \mathbb{R}^{2n} \to \mathbb{R}^{2n}$ *is called* invertible *if there exists an* inverse function $F^{-1}\colon \mathbb{R}^{2n} \to \mathbb{R}^{2n}$ *such that* $F(F^{-1}(\mathbf{z})) = F^{-1}(F(\mathbf{z})) = \mathbf{z}$ *for all* $\mathbf{z} \in \mathbb{R}^{2n}$*. The evolution function is called* locally invertible *if for any configuration* $\mathbf{z}$ *there are open subsets* $U_{\mathbf{z}}$ *and* $V_{F(\mathbf{z})}$ *containing* $\mathbf{z}$ *and* $F(\mathbf{z})$*, respectively, such that* $F_{\mathbf{z}}\colon U_{\mathbf{z}} \to V_{F(\mathbf{z})}$ *is invertible, i.e., there exists a* local inverse function $F_{\mathbf{z}}^{-1}\colon V_{\mathbf{z}} \to U_{F(\mathbf{z})}$*) such that* $F_{\mathbf{z}}(F_{\mathbf{z}}^{-1}(\mathbf{y})) = \mathbf{y}$ *for all* $\mathbf{y} \in U_{F(\mathbf{z})}$ *and* $F_{\mathbf{z}}^{-1}(F_{\mathbf{z}}(\mathbf{y})) = \mathbf{y}$ *for all* $\mathbf{y} \in U_{\mathbf{z}}$.

▶ Remark A.6. The (local) inverse function is not the evolution function of a distributed protocol, in general, i.e., it is not of the form (2).

▶ **Proposition A.7.** *Consider the dynamics of a configuration according to an arbitrary protocol* (2)*. Assume that the evolution function* $F\colon \mathbb{R}^{2n} \to \mathbb{R}^{2n}$ *is invertible. Then the configuration after one round cannot have more symmetries than the initial one. In particular,* $G_{\mathbf{z}} = G_{\mathbf{z}^+}$.

Proof. The setup for the proof is the same as in the previous. Let $\mathbf{z} \in \mathbb{R}^{2n}$ be some arbitrary configuration. Consider the evolution $\mathbf{z}^+ = F(\mathbf{z})$ in one round. By assumption, F is invertible and we may restate $\mathbf{z} = F^{-1}(\mathbf{z}^+)$. It can readily be seen that the inverse F^{-1} has the same symmetry properties as F:

$$F \circ M = M \circ F \iff F \circ M \circ F^{-1} = M \iff M \circ F^{-1} = F^{-1} \circ M$$

for any $M \in G$.

In particular, for $M \in G_{\mathbf{z}^+}$ we may apply Proposition A.4 to F^{-1} to obtain $M \in G_{\mathbf{z}}$. This implies $G_{\mathbf{z}^+} \subset G_{\mathbf{z}}$, which in combination with Proposition A.4 proves the claim. ◀

The previous proposition requires the existence of a *global* inverse F^{-1} to F. However, the weaker notion of *local* invertibility is sufficiently strong to draw the conclusions of Proposition A.7.

▶ **Proposition A.8.** *Consider the dynamics of a configuration according to an arbitrary protocol* (2)*. Assume that the evolution function* $F\colon \mathbb{R}^{2n} \to \mathbb{R}^{2n}$ *is* locally *invertible. Then, the configuration after one round cannot have more symmetries than the initial one. In particular,* $G_{\mathbf{z}} = G_{\mathbf{z}^+}$.

Proof. As we have a local inverse for every configuration $\mathbf{z} \in \mathbb{R}^{2n}$, we may compare $G_{\mathbf{z}}$ and $G_{\mathbf{z}^+}$ using the local inverse as before. To that end, let $\mathbf{z} \in \mathbb{R}^{2n}$ be an arbitrary configuration, $\mathbf{z}^+ = F(\mathbf{z})$, $F_{\mathbf{z}}^{-1}$ the local inverse, and $M \in G_{\mathbf{z}^+}$. Then

$$\begin{aligned}\mathbf{z} &= F_{\mathbf{z}}^{-1}(\mathbf{z}^+)\\ &= F_{\mathbf{z}}^{-1}(M\mathbf{z}^+)\\ &= F_{\mathbf{z}}^{-1}(MF(F_{\mathbf{z}}^{-1}(\mathbf{z}^+)))\\ &= F_{\mathbf{z}}^{-1}(F(MF_{\mathbf{z}}^{-1}(\mathbf{z}^+)))\\ &= MF_{\mathbf{z}}^{-1}(\mathbf{z}^+) = M\mathbf{z}.\end{aligned}$$

All applications of the local inverse are well-defined, as M leaves $\mathbf{z}^+$ unchanged. In particular, we have shown that $M \in G_{\mathbf{z}}$ proving the necessary inclusion as in Proposition A.7. ◀

▶ Remark A.9. One readily confirms, that the set of all potential symmetries G has the algebraic structure of a group, i.e., it contains the identity, is closed under products, and contains the inverse of every element—fully justifying the reference to equivariant dynamics. Furthermore, the subset of symmetries of a configuration $G_{\mathbf{z}}$ is a subgroup, i.e., a subset that is a group itself. Consequently, we may refer to G as the *symmetry group* and to $G_{\mathbf{z}}$ as the *isotropy subgroup*. One may further compute that all matrices in G have determinant 1. Hence, G is a subgroup of the special orthogonal group $SO(2n)$ consisting of all rotations of R^{2n}.

▶ Remark A.10. The equivariant dynamical systems formalism immediately offers simple means to deduce features of the collective dynamics. In fact, any subspace of R^{2n} that is pointwise mapped to itself by any of the potential symmetries in G cannot be left by the collective dynamics. This can for example be used to show the invariance of certain formations. However, since this is not the focus of this paper, we omit any details at this point.

B Invertibility of Protocol 1

Here we prove Lemma 4.1, namely that our first protocol (Section 4) satisfies the assumptions of Theorem 1.1. In fact, we need to confirm, that the induced evolution function is indeed locally invertible. However, as this proof is technical and tedious but not very enlightening, we omit most of the technicalities and provide a sketch instead. The main ingredients of the proof are

- the inverse function theorem (e.g., [23]), which states that that a continuously differentiable function is locally invertible at every point where its Jacobian is an invertible matrix, and
- the Gershgorin circle theorem [20], which states that all eigenvalues of a matrix $A = (a_{i,j})_{i,j=1}^n$ are contained in the union of the circles $\{z \in \mathbb{C} \mid |z - a_{i,i}| \leq \sum_{j=1, j\neq i}^n |a_{i,j}|\}$.

We first fix some notation. Recall from (8) that

$$f(z_i, \mathbf{z}) = z_i + \frac{\varepsilon}{n} \sum_{j=1}^{n} b(\|z_i - z_j\|^2)(z_j - z_i),$$

which is smooth—and thus continuously differentiable in particular—by construction. Furthermore, it is a two-dimensional expression. The collection of these expressions for $i = 1, \ldots, n$

is the evolution function F. To specify the x- and y-directions separately, we denote

$$F(\mathbf{z}) = \begin{pmatrix} f(z_1, \mathbf{z}) \\ \vdots \\ f(z_n, \mathbf{z}) \end{pmatrix} = \begin{pmatrix} F_1^x(\mathbf{z}) \\ F_1^y(\mathbf{z}) \\ \vdots \\ F_n^x(\mathbf{z}) \\ F_n^y(\mathbf{z}) \end{pmatrix}. \tag{B.1}$$

We use the representation (B.1) to compute the Jacobian $DF(\mathbf{z})$ at an arbitrary point $\mathbf{z} = ((x_1, y_1), \dots, (x_n, y_n)) \in \mathbb{R}^{2n}$. It is of the form

$$DF(\mathbf{z}) = (D_{i,j})_{i,j=1}^n \quad \text{with} \quad D_{i,j} = \begin{pmatrix} \partial_{x_j} F_i^x(\mathbf{z}) & \partial_{y_j} F_i^x(\mathbf{z}) \\ \partial_{x_j} F_i^y(\mathbf{z}) & \partial_{y_j} F_i^y(\mathbf{z}) \end{pmatrix}.$$

The $2n$ Gershgorin circles of the Jacobian $DF(\mathbf{z})$ are centered at $\partial_{x_i} F_i^x(\mathbf{z})$ and $\partial_{y_i} F_i^y(\mathbf{z})$ for $i = 1, \dots, n$. Their radii are given by

$$R_i^x(\mathbf{z}) = \sum_{j \neq i} |\partial_{x_j} F_i^x(\mathbf{z})| + \sum_{j=1}^n |\partial_{y_j} F_i^x(\mathbf{z})| \quad \text{and} \quad R_i^x(\mathbf{z}) = \sum_{j \neq i} |\partial_{y_j} F_i^y(\mathbf{z})| + \sum_{j=1}^n |\partial_{x_j} F_i^y(\mathbf{z})|$$

respectively.

We compute the partial derivatives as

$$\partial_{x_j} F_i^x(\mathbf{z}) = \begin{cases} 2\frac{\varepsilon}{n} b'(\|z_i - z_j\|^2)(x_i - x_j)^2 + \frac{\varepsilon}{n} b(\|z_i - z_j\|^2), & j \neq i, \\ 1 - 2\frac{\varepsilon}{n} \sum_{j \neq i} (b'(\|z_i - z_j\|^2)(x_i - x_j)^2 + b((\|z_i - z_j\|^2)), & j = i, \end{cases}$$

$$\partial_{y_j} F_i^x(\mathbf{z}) = \begin{cases} 2\frac{\varepsilon}{n} b'((\|z_i - z_j\|^2))(x_i - x_j)(y_i - y_j), & j \neq i, \\ 2\frac{\varepsilon}{n} \sum_{j \neq i} b'(\|z_i - z_j\|^2)(x_i - x_j)(y_i - y_j), & j = i, \end{cases}$$

$$\partial_{x_j} F_i^y(\mathbf{z}) = \begin{cases} 2\frac{\varepsilon}{n} b'((\|z_i - z_j\|^2))(x_i - x_j)(y_i - y_j), & j \neq i, \\ 2\frac{\varepsilon}{n} \sum_{j \neq i} b'(\|z_i - z_j\|^2)(x_i - x_j)(y_i - y_j), & j = i, \end{cases}$$

$$\partial_{y_j} F_i^y(\mathbf{z}) = \begin{cases} 2\frac{\varepsilon}{n} b'(\|z_i - z_j\|^2)(y_i - y_j)^2 + \frac{\varepsilon}{n} b(\|z_i - z_j\|^2), & j \neq i, \\ 1 - 2\frac{\varepsilon}{n} \sum_{j \neq i} (b'(\|z_i - z_j\|^2)(y_i - y_j)^2 + b((\|z_i - z_j\|^2)), & j = i. \end{cases}$$

Omitting any details, using these expressions it can be shown that for

$$\varepsilon < \frac{n}{27(n-1)} \tag{B.2}$$

one has

$$R_i^x(\mathbf{z}) < |\partial_{x_i} F_i^x(\mathbf{z})| \quad \text{and} \quad R_i^x(\mathbf{z}) < |\partial_{x_i} F_i^x(\mathbf{z})|. \tag{B.3}$$

This estimate essentially only requires the triangle inequality and the specific form of the bump function b (see (6)). In particular, (B.3) shows that none of the Gershgorin circles contains 0. Therefore, 0 cannot be an eigenvalue and the Jacobian $DF(\mathbf{z})$ is invertible for any $\mathbf{z} \in \mathbb{R}^{2n}$. By the inverse function theorem, F is therefore locally invertible at every $\mathbf{z} \in \mathbb{R}^{2n}$.

▶ Remark B.1. The estimate (B.2) is not sharp but sufficient to guarantee the estimate of the radii (B.3).

C Additional Proofs and Figures for Protocol 2

▶ **Lemma 5.4** (restated). *If $\epsilon \in [0, 0.5)$, ϵ-Go-to-the-Middle is invertible for a global observer.*

Proof. ϵ-Go-to-the-Middle can be described as $\mathbf{z}^{t+1} = F(\mathbf{z}^t)$ with

$$F = \begin{pmatrix} 1-\epsilon & \frac{\epsilon}{2} & 0 & 0 & 0 & \cdots & 0 & \frac{\epsilon}{2} \\ \frac{\epsilon}{2} & 1-\epsilon & \frac{\epsilon}{2} & 0 & 0 & \cdots & 0 & 0 \\ 0 & \frac{\epsilon}{2} & 1-\epsilon & \frac{\epsilon}{2} & 0 & \cdots & 0 & 0 \\ 0 & 0 & \frac{\epsilon}{2} & 1-\epsilon & \frac{\epsilon}{2} & \cdots & 0 & 0 \\ \cdots & & & & & & & \\ \frac{\epsilon}{2} & 0 & 0 & 0 & 0 & \cdots & \frac{\epsilon}{2} & 1-\epsilon \end{pmatrix}.$$

For $\epsilon < 0.5$, F is strictly diagonally dominant. By [20], F is invertable.

◀

▶ **Lemma 5.5** (restated). *If $\mathbf{b}^t$ is convex, $\mathbf{b}^{t+1}$ is convex as well. Let $area(\mathbf{b}^t)$ denote the area enclosed by $\mathbf{b}^t$. Then, $area(\mathbf{b}^{t+1}) \subseteq area(\mathbf{b}^t)$.*

Proof. Let us assume $\mathbf{b}^t$ is a convex set. We consider three neighboring robots $b^t_{k-1}, b^t_k, b^t_{k+1}$ in $\mathbf{b}^t$ that are not collinear. They form a triangle. Let τ be the target point of b^t_k executing ϵ-GTM with $\epsilon = 1$. The point τ is on the line l between b^t_{k-1} and b^t_{k+1} (black in Figure 4).

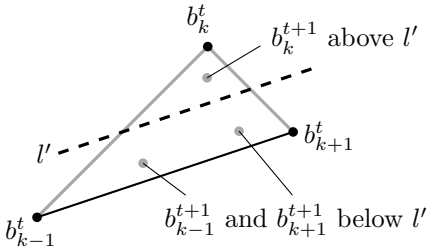


Figure 4 Shows that a convex corner in $\texttt{wave}^t$ is convex in $\texttt{wave}^{t+1}$ as well (Lemma 5.5).

The position of robot b^{t+1}_k is the target point of ϵ-GTM with $\epsilon < 0.5$. It can be constructed by moving τ factor ϵ towards b^t_k which is more than half the way. Therefore, it must lie above l', the parallel line to l move half way towards b^t_k (dotted in Figure 4).

Let τ' be the target point of b^t_{k+1} executing ϵ-GTM with $\epsilon = 1$. Because $\mathbf{b}^t$ is a convex set, b^t_{k+2} must lie below l. The midpoint between b^t_k and b^t_{k+2} cannot lie above l'. Moving τ' towards b^t_{k+1} cannot move it above l'. Therefore, b^{t+1}_{k+1} and (analogous) b^{t+1}_{k-1} lies below l'. Therefore, the robots b^{t+1}_{k-1}, b^{t+1}_k and b^{t+1}_{k+1} form a convex corner in $\mathbf{b}^{t+1}$.

If $b^t_{k-1}, b^t_k, b^t_{k+1}$ are collinear, they might move onto target points on the same line. But with analog arguments as above, it is easy to see that they cannot move onto positions that form a concave corner.

Therefore, the set $\mathbf{b}^{t+1}$ is still convex.

In the proof above we have shown, that b^{t+1}_k lies inside the triangle $b^t_{k-1}, b^t_k, b^t_{k+1}$. Therefore, the area surrounded by $\mathbf{b}^{t+1}$ is a subset of the area surrounded by $\mathbf{b}^t$.

◀

▶ **Lemma 5.9** (restated). *Let $\mathbf{b}^t$ be a convex set of robots. The segments of $\texttt{wave}^t$ are not twisted quadrilaterals, i.e., two sides do not intersect on a single point, and do not overlap.*

Proof. From the arguments of Lemma 5.5 follows, that each position b_k^{t+1} lies in the triangle $b_k^t, \frac{b_k^t+b_{k-1}^t}{2}, \frac{b_k^t+b_{k+1}^t}{2}$. Figure 5 shows these triangles with gray dashed lines. It is easy to see, that quadrilaterals with corners in these triangles cannot be overlapping or twisted.

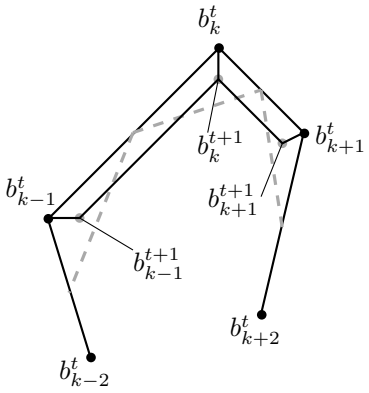


Figure 5 Shows that a convex corner in $\texttt{wave}^t$ is convex in $\texttt{wave}^{t+1}$ as well (Lemma 5.5).

◀

▶ **Lemma 5.10** (restated). *We call a quadrilateral where all four corners are not collinear* non-degenerated*, a* partially degenerated *quadrilateral has 3 collinear corners and a* fully degenerated *has 4 collinear corners.*

1. *If* $\texttt{seg}_k^t$ *is a non-degenerated quadrilateral,* $\texttt{seg}_k^{t+1}$ *is also a non-degenerated quadrilateral.*
2. *If* $\texttt{seg}_k^t$ *is a partially degenerated segment,* $\texttt{seg}_k^{t+1}$ *is a non-degenerated segment.*

Proof. (1) can be followed from Lemma 5.5.

(2) If b_k^t, b_{k+1}^t and b_{k+2}^t are collinear but b_{k-1}^t is not, $b_{k+1}^{t+1} = b_{k+1}^t$ will not move but $b_k^{t+1} \neq b_k^t$. Therefore, b_k^{t+1}, b_{k+1}^{t+1} and b_{k+2}^{t+1} are not anymore collinear and $\texttt{seg}_k^{t+1}$ is non-degenerated. ◀

▶ **Lemma 5.11** (restated). *Robots in* $\texttt{seg}_k^t$ *and* $\texttt{seg}_k^{t+1}$ *have a distance of* $\leq 1 + \epsilon^2/2$ *to* b_k^t *and* b_{k+1}^t.

Proof. We compute the distances between the corners of $\texttt{seg}_k^t$ and $\texttt{seg}_k^{t+1}$.

In the equations below we estimate $|b_k^t - b_{k+1}^t| \leq 1$, $|b_k^t - b_{k+2}^t| \leq 2$ and $\left|b_k^t - b_{k+3}^t\right| \leq 3$.

$$\begin{aligned}\left|b_k^t - b_{k+1}^{t+1}\right| &= \left|b_k^t - \frac{\epsilon}{2}b_k^t - \frac{\epsilon}{2}b_{k+2}^t - (1-\epsilon)b_{k+1}^t\right| \\ &= \frac{\epsilon}{2}\left|b_k^t - b_{k+2}^t\right| + (1-\epsilon)\left|b_k^t + b_{k+1}^t\right| \\ &\leq \epsilon + (1-\epsilon) = 1\end{aligned}$$

Analog is $\left|b_{k+1}^t - b_k^{t+1}\right| \leq 1$. It is clear, that $\left|b_k^t - b_k^{t+1}\right| \leq \epsilon$ and $\left|b_{k+1}^t - b_{k+1}^{t+1}\right| \leq \epsilon$.

$$\begin{aligned}
\left|b_k^t - b_{k+1}^{t+2}\right| &= \left|b_k^t - (1-\epsilon)b_{k+1}^{t+1} - \frac{\epsilon}{2}b_k^{t+1} - \frac{\epsilon}{2}b_{k+2}^{t+1}\right| \\
&\leq (1-\epsilon)\cdot 1 + \frac{\epsilon}{2}\cdot\epsilon + \frac{\epsilon}{2}\left|b_k^t - \frac{\epsilon}{2}b_{k+1}^t - \frac{\epsilon}{2}b_{k+3}^t - (1-\epsilon)b_{k+2}^t\right| \\
&\leq (1-\epsilon) + \frac{\epsilon^2}{2} + \frac{\epsilon}{2}(\frac{\epsilon}{2} + 3\frac{\epsilon}{2} + 2(1-\epsilon)) \\
&= (1+\epsilon)(1-\epsilon) + \frac{3}{2}\epsilon^2 \\
&= 1 + \frac{\epsilon^2}{2}
\end{aligned}$$

Analog is $\left|b_{k+1}^t - b_k^{t+2}\right| \leq 1 + \frac{\epsilon^2}{2}$. It is clear, that $\left|b_k^t - b_k^{t+2}\right| \leq 2\epsilon$ and $\left|b_{k+1}^t - b_{k+1}^{t+2}\right| \leq 2\epsilon$.

Any robot inside $\mathtt{seg}_k^t$ or $\mathtt{seg}_k^{t+1}$ as a smaller or equal distance to b_k^t and b_{k+1}^t than the farthest corner of the wave-segments. Therefore, the distance is $\leq 1 + \frac{\epsilon^2}{2}$. ◀

▶ **Lemma 5.17** (restated). *In a configuration with a convex Connectivity-Boundary and no* 2.24*-holes, Algorithm 3* ***is executable*** *for $\mathcal{OBLOT}$-robots with a viewing range of $2+\sqrt{2}$.*

Proof. Decide robot state locally (boundary, wave, inner). With no 2.24-holes, a robot is on the Connectivity-Boundary if it is on the border of the convex hull of its 2.24-surrounding. This can be determined with a viewing range of 2.24. A robot r inside $\mathtt{seg}_k^t$ or $\mathtt{seg}_k^{t+1}$ has a distance $\leq 1 + \epsilon^2/2 < 1.12$ to b_k^t and b_{k+1}^t (see Lemma 5.11). Therefore, r with a viewing range of $1.12 + 2.24 < 2 + \sqrt{2}$ can determine, whether the robot on b_k^t is a boundary-robot. To compute $\mathtt{seg}_k^t$ and $\mathtt{seg}_k^{t+1}$ robots on $b_{k-2}^t, \cdots, b_{k+3}^t$ must be observed. To b_k^t and b_{k+1}^t these have a maximal distance of up to 2. Because the robot on b_k^t could already be identified, it is known where the outside of the Connectivity-Boundary is. We know that the Connectivity-Boundary is connected with a distance ≤ 1. Therefore, the 1-surrounding around a known boundary-robot is sufficient to identify its next neighbor along the boundary. With a viewing range of $2+\sqrt{2}$ the 1-surrounding of b_k^t and b_{k+1}^t as well as from b_{k-1}^t and b_{k+2}^t is observable. This allows to identify robots on $b_{k-2}^t, \cdots, b_{k+3}^t$. Therefore, each robot in $\mathtt{seg}_k^t$ and $\mathtt{seg}_k^{t+1}$ can identify, that it is in the mentioned segment (wave-robots). All robots r not in these segments can either not find sufficient many boundary-robots or can compute that they are not in segments of $\mathtt{wave}^t$ or $\mathtt{wave}^{t+1}$ adjacent to the observed boundary-robots. Both is sufficient to decide, that r is an inner robot.

Compute the algorithm locally. For boundary robots (ϵ-Go-to-the-Middle) and inner robots (do not move) this is trivial. The wave-segments are not overlapping. The target position is computed based on the segment a robot is in. Therefore, a robot not on the borders of a segment can unambiguously compute its target positions. A robot on the border of a segment computes for both segments the same target position, hence, the computation is unambiguous as well. The movement distance is ≤ 1, therefore the computed move can be executed. ◀

▶ **Lemma 5.18** (restated). *In a configuration with a convex Connectivity-Boundary Algorithm 3* ***does not lead to collisions****.*

Proof. We consider round t. Robot within $\mathtt{wave}^{t'}, t' \geq t+2$ (inner-robots) cannot have collisions with each other, because they do not move. They can also have no collisions with other robots, because boundary- and wave-robots move within $\mathtt{wave}^t$ and $\mathtt{wave}^{t+1}$. Boundary robots are essentially wave robots with the special case that they are on coordinates $\mathtt{seg}_k^t(x, 0)$. From the proof of Lemma 5.13 follows, that Algorithm 2 is collision-free. ◀

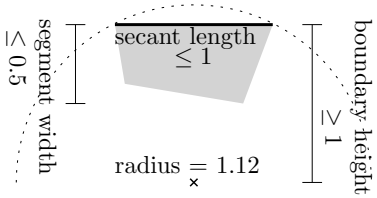


Figure 6 Distance between $\texttt{seg}_k^t$ (gray) and the midpoint of a 2.24-hole.

▶ **Lemma 5.19** (restated). *In a configuration with convex Connectivity-Boundary and initially no* 1*-holes, Algorithm 3* ***does not create*** 2.24***-holes****.*

Proof. All robots not inside $\texttt{wave}^t$ have been inner robots until now and have not moved at all. Therefore, holes that have no overlap with $\texttt{wave}^t$ must have existed initially and can only have diameter < 1. Let us consider a 2.24-hole that overlaps with $\texttt{wave}^t$. See the construction of $\texttt{seg}_k^t$ that overlaps with a 2.24-hole in Figure 6. $\texttt{seg}_k^t$ cannot lie completely within the hole, because that would mean the boundary robots are inside the hole. But the boundary of $\texttt{seg}_k^t$ can be a secant with length ≤ 1 of the hole. The height of such a secant of a circle with radius 1.12 is ≥ 1. A wave-segment can have a maximal width of 0.5, therefore all points of $\texttt{seg}_k^t$ have a distance ≥ 0.5 to the midpoint of the hole. Therefore, 1-hole with the same midpoint does not overlap with $\texttt{wave}^t$ and must have existed initially. ◀

C.1 Visualization of the Algorithms and Definitions

In this subsection, we provide a more detailed and split version of Figure 3a. The goal is to visualize how robots decide their role and compute their movement.

Connectivity-Boundary and Holes

Figure 7 (left) depicts a configuration with a non-convex Connectivity-Boundary. The small dotted lines are the unit-disc-graph; the Connectivity-Boundary is the outer boundary of this graph. Note that our proof does not apply to this configuration because the Connectivity-Boundary is not convex. Figure 7 (right) depicts the same configuration scaled by factor 0.35, such that the Connectivity-Boundary is convex. It also shows the largest hole. Because the unit-disc-graph in this example is close to a complete graph, we omitted it.

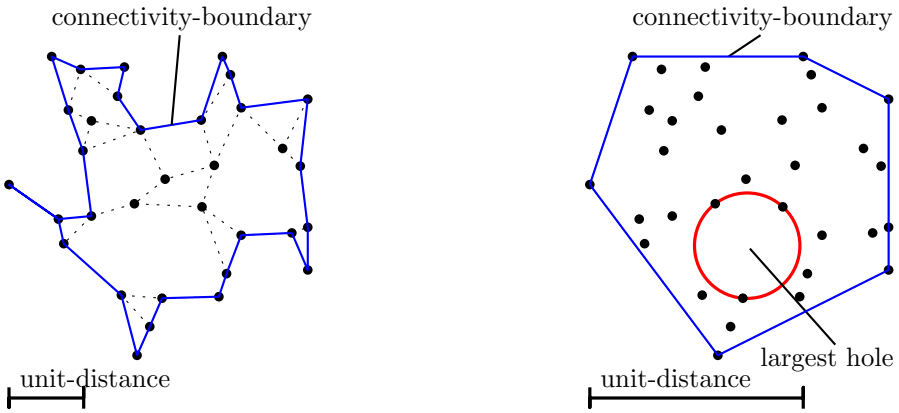


Figure 7 Left: a configuration with a non-convex Connectivity-Boundary and its unit-disc-graph. Right: A configuration with a convex Connectivity-Boundary and its largest hole.

Construction of the Wave

The waves are computed solely from the robots on the Connectivity-Boundary. Figure 8 (left) shows how the wave segments are computed via the midpoints of neighboring robots (Algorithm 1). For the point b_0^{t+1} the formula is written. Figure 8 (right) shows the naming of the waves (Definition 5.6), its segments (Definition 5.8) and the boundary (Definition 5.1).

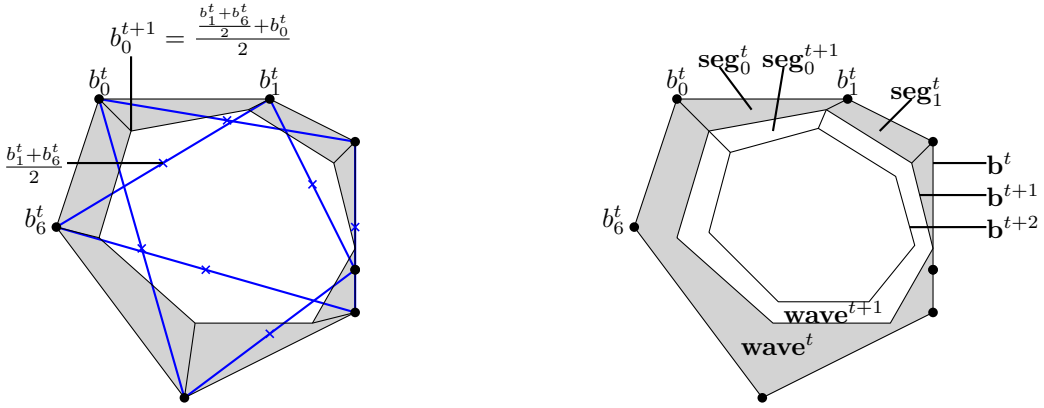


Figure 8 Left: computation of the wave. Right: the naming of waves, segments, and boundary.

Robots and Movement

Figure 9 (left) shows the partition of robots in boundary-robot (Definition 5.1), wave-robots (Definition 5.7) and inner-robots (Definition 5.14). Figure 9 (right) shows the new robot positions after execution Algorithm 3. The figure in the middle shows a close-up on $\mathtt{seg}_6^t$ and $\mathtt{seg}_6^{t+1}$ that demonstrates the computation of Algorithm 2, based on the coordinate-system defined in Definition 5.12.

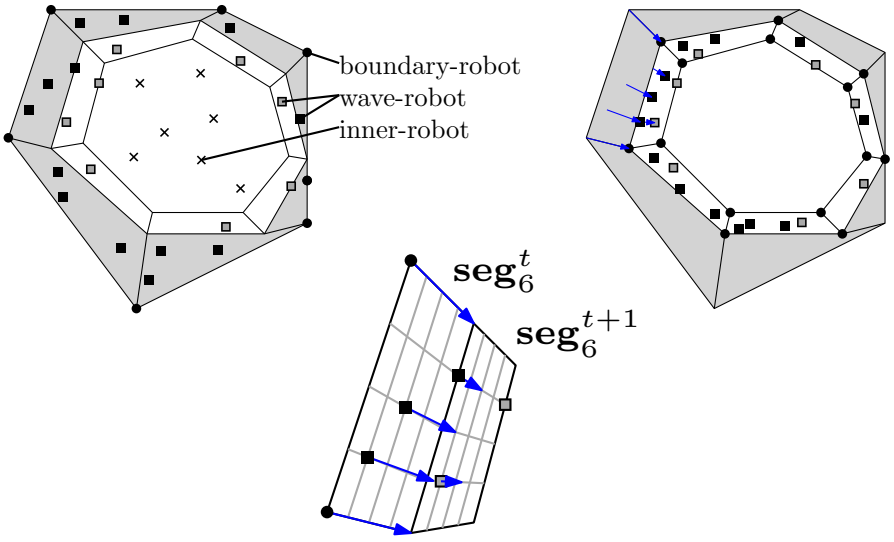


Figure 9 Left: robot classification, Right: new positions after executing Algorithm 3, Middle: closeup of the movement in $\mathtt{seg}_6^t$ and $\mathtt{seg}_6^{t+1}$.